# Krylov Subspace Methods in Finite Precision: A Unified Approach

Vom Promotionsausschuß der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
genehmigte Dissertation

von

**Jens–Peter Max Zemke**

aus Lüneburg

**2003**

1. Gutachter: Prof. Dr. S. M. Rump
2. Gutachter: Prof. Dr. H. Voß

Tag der mündlichen Prüfung: 14.05.2003

# Contents

# List of Figures

# List of Algorithms

# List of Symbols

**number sets, rings, fields**

| | |
|---|---|
| $\mathbb{N}$, $\mathbb{Z}$ | the set of natural numbers and the ring of integers |
| $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ | the fields of rational, real and complex numbers |
| $\mathbb{K}$ | one of $\mathbb{R}$ or $\mathbb{C}$, in Chapter 6 also $\mathbb{Q}$ or finite field |
| $\mathbb{F}$ | the set of IEEE 754 floating point numbers |

**spaces, subspaces**

| | |
|---|---|
| $\mathbb{R}^n$, $\mathbb{C}^n$, $\mathbb{K}^n$ | (column) vector space of dimension $n$ over $\mathbb{R}$, $\mathbb{C}$, $\mathbb{K}$ |
| $\mathbb{R}^{n\times m}$, $\mathbb{C}^{n\times m}$, $\mathbb{K}^{n\times m}$ | matrix space of dimension $n \times m$ over $\mathbb{R}$, $\mathbb{C}$, $\mathbb{K}$ |
| $\mathcal{U}$,$\mathcal{U}(n) \subset \mathbb{K}^{n\times n}$ | space of unitary / orthogonal matrices |
| $\mathcal{H}$,$\mathcal{H}(n) \subset \mathbb{K}^{n\times n}$ | set of unreduced Hessenberg matrices |
| $\mathcal{K}$, $\mathcal{K}_k \subset \mathbb{K}^n$ | Krylov subspace (of step $k$) |
| $\mathcal{K}_k\,(A, q) \subset \mathbb{K}^n$ | Krylov subspace spanned by $q, Aq, \ldots, A^{k-1}q$ |
| $\mathbb{P}$, $\mathbb{P}_k$ | space of polynomials and the subspace of polynomials with degree at most $k$ |
| $\mathbb{P}_k^k$, $\mathbb{P}_k^0$ | sets of monic polynomials and polynomials with constant term one of degree at most $k$ |
| $\mathcal{L}$, $\mathcal{L}_k \subset \mathbb{K}^n$ | space spanned by the column vectors of $L$, $L_k$ |
| $\mathcal{M}$, $\mathcal{M}_k \subset \mathbb{K}^n$ | space spanned by the column vectors of $M$, $M_k$ |
| $\operatorname{span}\{q_1, \ldots, q_k\}$ | space spanned by the vectors $q_1, \ldots, q_k$ |

**important (structured) matrices**

| | |
|---|---|
| $A \in \mathbb{K}^{n\times n}$ | sparse, large system matrix, maybe with special properties introduced in the context |
| $B, B_k \in \mathbb{K}^{k\times k}$ | bidiagonal matrix (dimension follows from context) |
| $D, D_k \in \mathbb{K}^{k\times k}$ | diagonal matrix (dimension follows from context) |
| $I, I_k \in \mathbb{K}^{k\times k}$ | identity matrix (dimension follows from context) |
| $J, J_k \in \mathbb{K}^{k\times k}$ | Jordan normal form (dimsion follows from context) |
| $L, L_k \in \mathbb{K}^{k\times k}$ | lower triangular matrix, with unit diagonal (dimension follows from context) |
| $M, M_k \in \mathbb{K}^{k\times k}$ | matrices with low rank, mostly with rank one (dimension follows from context) |
| $R, R_k \in \mathbb{K}^{k\times k}$ | upper triangular matrix, mostly regular (dimension follows from context) |
| $T, T_k \in \mathbb{K}^{k\times k}$ | tridiagonal matrix (dimension follows from context) |

**operations on matrices**

| | |
|---|---|
| $\det(A)$ | determinant of $A$ |

| | |
|---|---|
| $[q_1, \ldots, q_k]$ | matrix composed of the columns $q_1, \ldots, q_k$ |
| $A^H$ | conjugate transpose of $A$ |
| $A^T$ | transpose of $A$ |
| $C_k(A)$ | $k$-th compound matrix of $A$ |
| adj$(A)$ | adjugate or classical adjoint of $A$ |

**matrix inverses**

| | |
|---|---|
| $A^{-1}$ | (algebraic) inverse of $A$ |
| $A^{\eta}$ | $\eta$-inverse of $A$, $\eta \subset \{\mathrm{I}, \mathrm{II}, \mathrm{II}, \mathrm{IV}, \mathrm{V}, \mathrm{D}\}$ |
| $A^{\dagger} \equiv A^{\{\mathrm{I,II,III,IV}\}}$ | Moore-Penrose inverse of $A$, pseudo-inverse of $A$ |
| $A^{\{\mathrm{I}\}}$ | condition one inverse of $A$, g-inverse of $A$ |
| $A^{\{\mathrm{I,III}\}}$ | least-squares generalised inverse of $A$ |
| $A^{\{\mathrm{II,IV}\}}$ | minimum-norm generalised inverse of $A$ |
| $A^{\#} \equiv A^{\{\mathrm{I,II,V}\}}$ | group inverse of $A$ |
| $A^{D} \equiv A^{\{\mathrm{II,V,D}\}}$ | Drazin-inverse of $A$ |

**projected matrices and corresponding bases**

| | |
|---|---|
| $C, C_k \in \mathbb{K}^{k\times k}$ | computed condensed matrix, has form $T$, $H$ or $N$ |
| $N, N_k \in \mathbb{K}^{k\times k}$ | nilpotent matrix |
| $H, H_k \in \mathbb{K}^{k\times k}$ | Hessenberg matrix |
| $T, T_k \in \mathbb{K}^{k\times k}$ | tridiagonal matrix |
| $Q, Q_k \in \mathbb{K}^{n\times k}$ | (bi)orthogonal to $\hat{Q}$ ($\hat{Q}_k$), computed counterpart |
| $\hat{Q}, \hat{Q}_k \in \mathbb{K}^{n\times k}$ | (bi)orthogonal to $Q$ ($Q_k$), computed counterpart |

**eigenproblem matrices related to $A$**

| | |
|---|---|
| $V$ | matrix of right eigenvectors (principal vectors) of $A$ |
| $\hat{V} \equiv V^{-H}$ | special matrix of left eigenvectors of $A$ |
| $\check{V} \equiv V^{-T}$ | special matrix of left eigenvectors of $A$ |
| $J_\Lambda$ | Jordan normal form of $A$ |
| $\Lambda$ | diagonal matrix of eigenvalues of $A$ |

**eigenproblem matrices related to $C$**

| | |
|---|---|
| $S, S_k \in \mathbb{K}^{k\times k}$ | matrix of right eigenvectors (principal vectors) of $C$, $C_k$ |
| $\hat{S} \equiv S^{-H}$ | special matrix of left eigenvectors of $C$ |
| $\check{S} \equiv S^{-T}$ | special matrix of left eigenvectors of $C$ |
| $\hat{S}_k \equiv S_k^{-H} \in \mathbb{K}^{k\times k}$ | special matrix of left eigenvectors of $C_k$ |
| $\check{S}_k \equiv S_k^{-T} \in \mathbb{K}^{k\times k}$ | special matrix of left eigenvectors of $C_k$ |
| $J_\Theta$ | Jordan normal form of $C$ |
| $\Theta, \Theta_k \in \mathbb{K}^{k\times k}$ | eigenvalue matrix of $C$, $C_k$ |
| $Y \equiv QS$ | matrix of right Ritz vectors of $C$ |
| $\hat{Y} \equiv \hat{Q}\hat{S}$ | matrix of left Ritz vectors of $C$ |
| $Y_k \equiv Q_k S_k \in \mathbb{K}^{n\times k}$ | matrix of right Ritz vectors of $C_k$ |
| $\hat{Y}_k \equiv \hat{Q}_k \hat{S}_k \in \mathbb{K}^{n\times k}$ | matrix of left Ritz vectors of $C_k$ |

**matrices related to linear systems**

| | |
|---|---|
| $X, X_k \in \mathbb{K}^{n\times k}$ | matrix of approximate solutions |
| $R, R_k \in \mathbb{K}^{n\times k}$ | matrix of right residuals |

| | |
|---|---|
| $P, P_k \in \mathbb{K}^{n\times k}$ | matrix of right direction vectors |
| $\hat{R}, \hat{R}_k \in \mathbb{K}^{n\times k}$ | matrix of left residuals |
| $\hat{P}, \hat{P}_k \in \mathbb{K}^{n\times k}$ | matrix of left direction vectors |

**error matrices**

| | |
|---|---|
| $E, E_k, E_{\mathrm{loc}}, E_{\mathrm{rec}}$ | transformed error matrices |
| $F, F_k$ | recursion error matrices |

**submatrix-selection operators**

| | |
|---|---|
| $A(\alpha)$ | matrix $a_{ij}, i, j \in \alpha$ |
| $A(\alpha, \beta)$ | matrix $a_{ij}, i \in \alpha, j \in \beta$ |
| $A[\alpha, \beta]$ | matrix $a_{ij}, i \notin \alpha, j \notin \beta$ |
| $A[\alpha]$ | matrix $a_{ij}, i, j \notin \alpha$ |
| $A_{ij}$ | short form for $A[i, j]$ |
| $C_{l:k}$ | the matrix $C(l : k, l : k)$ |
| $\mathrm{diag}(A)$ | diagonal part of the matrix $A$ |
| $\mathrm{diag}(A, l)$ | $l$th diagonal of the matrix $A$ |
| $\mathrm{tridiag}(\gamma, \alpha, \beta)$ | tridiagonal matrix with diagonals $\gamma, \alpha, \beta$ |
| $\mathrm{tril}(A)$ | triangular upper part of the matrix $A$ |
| $\mathrm{tril}(A, l)$ | $l$th triangular upper part of the matrix $A$ |
| $\mathrm{triu}(A)$ | triangular lower part of the matrix $A$ |
| $\mathrm{triu}(A, l)$ | $l$th triangular lower part of the matrix $A$ |

**important vectors**

| | |
|---|---|
| $b$ | right hand side |
| $e$ | vector of all ones, $\mathrm{ones}(n, 1)$ |
| $e_k$ | unit vector with $k$-th component equal to one |
| $q, q_k$ | Krylov subspace vector |
| $r, r_k$ | residual vector |
| $s, s_j$ | left eigenvector of $C$ |
| $v, v_i$ | right eigenvector of $A$ |
| $x, x_k$ | solution to linear system |
| $y, y_j$ | left Ritz vector |

**important scalars**

| | |
|---|---|
| $\delta_{ij}$ | Kronecker delta, components of the identity matrix |
| $\lambda$ | eigenvalue of $A$ |
| $\theta$ | eigenvalue of $C, T, H$ |
| $i, i_1, i_l$ | indices for vectors and other quantities arising from $A$ |
| $j, j_1, j_p$ | indices for vectors and other quantities arising from $C$ |
| $k$ | step of the Krylov method, for non block size of subspace |
| $m$ | index reserved for the last step in execution of algorithm |
| $s_{lj}^{(k)}, s_{lj}$ | the $lj$ component of $S_k$, $l$th component of $s_j$ |

**important polynomials**

| | |
|---|---|
| $\chi = \chi_A$ | characteristic polynomial of $A$ |
| $\mu = \mu_A$ | minimal polynomial of $A$ |

| | |
|---|---|
| $\mu = \mu_{A,q}$ | minimal polynomial of $A$ with respect to $q$ |

**error analysis related**

| | |
|---|---|
| $\epsilon$ | machine precision, IEEE 754 double precision: $\epsilon \equiv 2^{-53}$ |
| $\mathbf{u}$ | unit roundoff, IEEE 754 double precision: $\mathbf{u} \equiv 2^{-52}$ |
| $\nu$ | smallest floating point number, IEEE 754 double precision: $\nu \equiv 2^{-1074}$ |
| $fl($`<term>`$)$ | floating point evaluation of `<term>` |
| `Inf` | IEEE 754 signal for 'Overflow' |
| `NaN` | IEEE 754 signal for 'Not A Number' |

**interval arithmetic related**

| | |
|---|---|
| $[a] \equiv [\underline{a}, \overline{a}]$ | interval in inf-sup arithmetic |
| $\langle a \rangle \equiv \langle a, r \rangle$ | interval in mid-rad arithmetic |
| $\mathrm{diam}([a])$ | diameter of interval $[a]$ |
| $\inf([a]) \equiv \underline{a}$ | infimum of interval $[a]$ |
| $\mathrm{mid}([a])$ | midpoint of interval $[a]$ |
| $\mathrm{rad}([a])$ | radius of interval $[a]$ |
| $\sup([a]) \equiv \overline{a}$ | supremum of interval $[a]$ |

**miscellaneous**

| | |
|---|---|
| $\kappa(\cdot)$ | condition number of $\cdot$ |
| $\{\ldots\}$ | set with elements ... |
| $\otimes$ | Kronecker product |
| $\overline{z}$ | conjugate complex of $z$ |
| $\underline{n}$ | the set $\{1, \ldots, n\}$ |
| $l : k$ | the vector $(\, l \quad l+1 \quad \cdots \quad k \,)^T$ |
| vec | vec operator |

# Chapter 1

# Preliminaries

The first chapter consists of a short introduction, defines notation and motivates why Krylov methods are, and will be, a necessary tool to accomplish the computational tasks of the 21st century. The lacking of simple error analyses of perturbed Krylov methods will become obvious in Chapter 4. This is compared to the state of the art in direct methods. To grasp the perfection that has been achieved in this area, we give an extensive listing of backward analyses, backward errors, condition numbers and expansions related to the solution process of linear systems of equations and to the algebraic eigenproblem. We give the basic results along with some generalizations.

We intend to show how the approach used in this thesis fits into the framework developed over the last five decades by a variety of authors to access the intrinsic properties of Krylov methods. We stress that it seems *impossible* to apply *standard error analysis* to understand the finite precision behaviour of Krylov subspace methods.

## 1.1 Introduction

Solving mathematical problems is the major area of scientific computation. Many of these mathematical problems arise in the engineering disciplines when modeling physical behaviour. Of particular interest are ever more precise models, increasing computational amount.

The solution process of nonlinear problems frequently is composed of iterated solution of linearized problems. Other linear problems arise as such. The two main tasks are the computation of

a) *the solution of a linear system* of equations and of

b) *eigenvalues and eigenvectors* of a linear transformation.

In other words, *matrices* play a star role in numerical computations.

There are *two ways* to solve problems of this type, *direct approaches* and *iterative approaches*. Even though the computation of eigenvalues has to be iterative, previous reductions to simpler form are mostly based on direct approaches. Direct approaches are more natural and have been used for a long time. Most direct methods used nowadays are stable and reliable.

*Large* matrix computations are often based on iterative approaches. A broad class of iterative methods is given by the class of Krylov subspace methods. Krylov subspace methods have a variety of favourable characteristics, at least in exact arithmetic:

*Krylov methods are direct methods.* To be more precise, they are coordinate free variants of some well-known matrix reduction and matrix decomposition algorithms.

*Krylov methods are optimal methods.* They compute the optimal solution in a subspace subject to method dependent constraints.

*Krylov methods are cheap methods.* When considered as iterative methods, Krylov methods tend to converge fast with mostly linear operation count and storage amount per step to the solution.

*Krylov methods are at the heart of numerical analysis.* Krylov methods are related to structured eigenvalue problems, to orthogonal polynomials, to rational approximation theory. Amongst others, this enables detailed convergence analysis.

*Krylov methods are closely related to each other.* In particular, a linear system solver can be used to extract eigenvalues, and vice versa.

This was the *infinite precision* part. This was the *good news.*

Now we switch to *bad* news. The bad news is the *finite precision* behaviour of Krylov methods. One main problem is the lack of *useful* error analyses, like results on backward stability. Another problem is the lack of *generality* of existing error analyses.

In finite precision, Krylov methods do not terminate after a finite number of steps. The solutions are not optimal in the subspace constructed. The connection to related areas used in the convergence analysis is lost. Nevertheless the methods compute useful results. Solely *part* of the matrix relations defining the methods in infinite precision have a finite precision counterpart. *Error analysis* has to be based on these relations. *Convergence analysis* has to be based on these relations.

A unified error analysis has to explain the behaviour of the finite precision Krylov methods and to include the results derived thus far. Our approach is considered with the understanding of the behaviour. Error bounds resulting from the analysis have to be based on the characteristics of the methods. For this reason we will not consider error bounds. The main result is that a *detoriation* is always the result of a *convergence* of a part of the *computed* quantities.

## 1.2 Notation

> "The important thing for an error analyst is to settle on a comfortable notation that does not hinder the thinking process."
>
> Nicholas J. Higham,
> Accuracy and Stability of Numerical Algorithms.

Notation becomes an important topic in any broad approach like ours. The presentation has to be understandable and self-explanatory, at the same time in a familiar and appropriate style. We use standard notation as developed by Wilkinson et.al. ([Wil65, Hou75, Par98, HJ85, HJ94]). A small portion is close to Matlab style pseudo-code notation.

Notation will be mainly introduced in the context. For better readability we collect the part of notation valid for the complete thesis or very important for our approach in this section. The notational conventions are violated sometimes, for instance when denoting matrix and vector entries. We tried to keep use of one symbol in different contexts at a minimum.

### The Basics

Capital roman letters stand for matrices, small roman letters stand for vectors, and small greek letters stand for scalars. Subspaces are denoted by capital calligraphic letters. Polynomials are denoted by small greek or roman letters. The letters $\mathbb{R}, \mathbb{C}$ are used to denote the real and complex fields. The letters $\mathbb{N}, \mathbb{Z}$ are used to denote the natural numbers and integers. The letter $\mathbb{K}$ is used to denote one of $\mathbb{R}, \mathbb{C}$.

Indices are denoted by small roman letters, starting with $i, j, k, \ldots$. For $n \in \mathbb{N}$ we define the set $\underline{n} \equiv \{1, \ldots, n\}$. For $z \in \mathbb{C}$ the complex conjugate is denoted by $\overline{z}$. The letter $I \equiv I_n$ denotes the identity matrix of dimension $n \in \mathbb{N}$. The columns of $I$ are denoted by $e_i$, $i \in \underline{n}$. The elements of $I$ are denoted by $\delta_{ij}$, $i, j \in \underline{n}$. The columns of $I$ are termed standard unit vectors. Whenever the notation $e_j$ (or similar) appears, the dimension should be obvious from the context.

The set of matrices of size $n$ with entries in $\mathbb{K}$ is denoted by $\mathbb{K}^{n\times n}$. The set of rectangular matrices is denoted by $\mathbb{K}^{n\times k}$. The set of vectors of length $n$ is denoted by $\mathbb{K}^n$. The sets $\mathbb{K}^{n\times n}, \mathbb{K}^n$ form an algebra and a vector space, respectively. The subset of unitary matrices will be denoted by $\mathcal{U}$. The set of unitary matrices of size $n$ is denoted by $\mathcal{U}(n)$.

### Problem Related

The letter $A$ is reserved for the system matrix of the eigenproblem and the linear system of equations. The dimension of $A$ is given by $n \in \mathbb{N}$. We suppose $A \in \mathbb{K}^{n\times n}$. The matrix $A$ is large and sparse. Further structure is introduced in the context. The matrix of right eigenvectors of $A$ is denoted by $V$, the diagonal matrix of eigenvalues is denoted by $\Lambda$ and the Jordan matrix is denoted by $J_\Lambda$. With this notation $AV = VJ_\Lambda$. We need to access the right eigenvectors separately. For this reason we define the columns of $V$ to be $v_i$, $i \in \underline{n}$. Note that some $v_i$ are principal vectors and no eigenvectors. The elements of $V$ are denoted by $v_{ji}$, $j, i \in \underline{n}$. We remark that the vector $v_i$ consists of the elements $v_{ji}$, $j \in \underline{n}$. We need to access the left eigenvectors. Since $V^{-1}$ is a matrix of left eigenvectors, we define $\hat{V}^H \equiv V^{-1}$, i.e., $\hat{V} \equiv V^{-H}$. With this notation

$$\hat{V}^H A = J_\Lambda \hat{V}^H \quad \text{and} \quad \hat{V}^H V = V^H \hat{V} = I.$$

In order to access the entries of $\hat{V}^H$ we define $\check{V} \equiv \overline{\hat{V}}$. With this notation

$$\check{V}^T A = J_\Lambda \check{V}^T \quad \text{and} \quad \check{V}^T V = V^T \check{V} = I.$$

This convention can be memorised as reflection on the real axis, turning *hat* to *vee* and vice versa. These notations are extended to any bi-orthogonal set.

We need to access some classes of submatrices of a given matrix. For given $A \in \mathbb{K}^{n\times n}$ we denote by $A_{ij}$, $i, j \in \underline{n}$ the matrix with $i$th row and $j$th column deleted. The notation $A_{ij}$ must not be confused with the element $a_{ij}$ in $i$th row and $j$th column,

$$\begin{array}{lll} a_{ij}, & \text{element of } A \text{ in} & i\text{th row, } j\text{th column,} \\ A_{ij}, & A \text{ with} & i\text{th row, } j\text{th column deleted.} \end{array}$$

The short-hand notation $A_i$ is used to denote $A_{ii}$.

### Method Related

Krylov methods compute smaller matrices, denoted by $C_m \in \mathbb{K}^{m\times m}$, $m \in \mathbb{N}$. The computed matrices form a sequence of principal submatrices. For this reason we

label the leading principal submatrices by size, $C_k \in \mathbb{K}^{k\times k}$, $k \in \underline{m}$. With this notation

$$C_m = \begin{pmatrix} C_k & \star \\ \star & \star \end{pmatrix}.$$

The matrix $C_m$ is structured. Mostly $H_m \equiv C_m$ is Hessenberg or even $T_m \equiv C_m$ tridiagonal. The set of all unreduced Hessenberg matrices is denoted by $\mathcal{H}$. The subset of unreduced Hessenberg matrices of size $m$ is denoted by $\mathcal{H}(m)$. Krylov methods compute the matrix $C_m$ using a basis expanded in every step. We denote the rectangular matrix whose columns span the basis by $Q_m$, $Q_m \in \mathbb{K}^{n\times m}$. The submatrices of $Q_m$ consisting of the first $k$ columns are denoted by $Q_k$, $Q_k \in \mathbb{K}^{n\times k}$, $k \in \underline{m}$. The columns of $Q_k$ are denoted by $q_j$, $j \in \underline{k}$.

**Polynomial Related**

We use the letter $\mathbb{P}$ to denote the vector space of all polynomials. The eigenvalues of any matrix $A$ are zeros of the characteristic polynomial $\chi_A(\lambda) \equiv \det(\lambda I - A)$. This definition ensures that the term $\lambda^n$ has the factor plus one. The minimal polynomial will be denoted by $\mu(\lambda) = \mu_A(\lambda)$.

The elements in Krylov subspaces are often described in terms of polynomials of degree less than the dimension of the space. The subspace of polynomials of degree less equal $k$ will be denoted by $\mathbb{P}_k$. Depending on the context, we need to distinguish two different normalisations. In the context of the eigenproblem we need monic polynomials, denoted by $\mathbb{P}_k^k$. With our definition $\chi_A \in \mathbb{P}_n^n$. In the context of linear systems we need polynomials with constant term equal one, denoted by $\mathbb{P}_k^0$.

For a more complete overview of the notation used we refer the reader to the list of symbols in the beginning of this thesis.

## 1.3 Motivation and History

This thesis is concerned with two problems of linear algebra:

a) find the solution of a linear system of equations

$$Ax = b,$$

and

b) find (partial) solutions to the algebraic eigenproblem

$$Av = v\lambda, \quad \hat{v}^H A = \lambda \hat{v}^H.$$

From an numerical analyst point of view it shows up to be very important to distinguish between *dense* and *sparse* systems. For dense systems the state of the art almost seems to have reached its final destination. Both roblems mentioned above are solved in a backward stable manner by a variety of well-known, well-understood algorithms.

Recent changes, modifications and enhancements are due to a better understanding, i.e., due to new results in *componentwise structured* and *relative* error analysis. Examples of such enhancements are Demmel's results on Cauchy matrices ([Dem00]) and Dhillon's *Relatively Robust Representations* ([Dhi97]).

Most of the methods for dense systems are included as black-box solvers in (freely) available software libraries like BLAS/LAPACK. A general start may be NETLIB ([Net]) and the overview of freely available software for linear algebra that has been compiled by Dongarra ([Don]).

The picture is not that rosy when we are looking at sparse systems. Most direct methods lead to storage problems due to fill-in and numerical instabilities due to

restrictions on the pivoting strategies. We stress that no *entirely* direct method for the eigenproblem can exist. Nevertheless we name a method that applies to the eigenproblem a *direct* method, when it is based on the *iterated application* of a direct method. An example of such a method is the QR-iteration.

The so-called *classical iterative* methods like Jacobi, Gauß–Seidel and SOR in general are converging to slowly to the solution to be of practical use. Krylov subspace methods are direct methods (they terminate after a finite number of steps, at least in theory) and improve over the classical iterative methods in the sense of being optimal. Krylov methods are frequently the method of choice for large sparse problems.

### Past

Krylov methods were developed in the early fifties. The first papers were Lanczos' papers published in 1950 and 1952 on his *method of minimised iterations* (cf. [Lan50, Lan52]), Arnoldi's paper published in 1951 based on Lanczos' ideas (cf. [Arn51]) and the joint paper by Hestenes and Stiefel published in 1952 (cf. [HS52]).

It was realized soon that the methods where not competitive to the other direct methods in terms of accuracy and stability. Due to a lack of better understanding they were abandoned, or only used in conjunction with complete reorthogonalisation, which made them less competitive. Nevertheless, the methods appeared in textbooks for the first time (cf. [Wil65, Hou75]).

In the seventies Reid was among the first to realize that CG interpreted as iterative method was superior to direct approaches for certain matrices, especially when used with *preconditioning* (cf. [Rei71]). At roughly the same time Paige gave the first detailed error analysis for the symmetric Lanczos method (cf. [Pai71]). The first public available code for Lanczos methods by Cullum and Willoughby dates back to the the end of the seventies (cf. [CW85a, CW85b]).

During the eighties a group consisting mainly of researchers around Parlett made use of Paige's results to obtain new error analyses and to develop more stable algorithms. This progress came in form of a series of Ph.D. theses (cf. [Sco78, Gre81, Grc81, Sim82, Day93]). Bai's analysis of the nonsymmetric Lanczos method (cf. [Bai94]) and Strakoš' work dates also to this period. Paige's analysis was contained for the first time in the republished version of Parlett's book (cf. [Par98]). Also contained is part of the work that builds upon this analysis.

The release of a textbook touching error analysis for Krylov subspace methods for the first time paved way for similar results to occur in a variety of textbooks (cf. [Gre97, Dem97, TB97, Meu99]). The topics covered are mainly results.

### State of the Art

Nowadays, Krylov methods have gained widespread recognition. They have even been elected one of the ten best algorithms ever, see ([TTA00]). In this list (ordered by date) they appear as third item,

> **3**. 1950: Krylov Subspace Iteration Method. A technique for rapidly solving the linear equations that abound in scientific computation.

Nevertheless, up to now, no black-box solver based on Krylov subspace methods exists. Packages for the solution of real-world problems refuse to contain Krylov methods, the reasons for this being the non-predictable behaviour with respect to success and, more seriously, time.

We think that the area of Krylov methods is approaching a state where it becomes useful to collect the results known to form a basis for future examination. This, in turn, might help to reach a state where the understanding and development of algorithms based on Krylov subspaces will also be satisfactory.

**Active Research and Future Work**

There are several areas of active research in the development and understanding of Krylov methods. In the authors opinion the main areas are given as follows:

**Preconditioning.** Krylov methods are only competitive when used with preconditioning. Since Reid published his paper (cf. [Rei71]) this is, maybe by far, the most important area.

**Restart, Truncation, Look-Ahead.** These new types of methods are introduced to fill the gap between fast and controllable Krylov methods. In the field of eigenproblem methods, implicit restart has become state of the art, as example we mention Lehoucq's and Sorensen's ARPACK (cf. [LSY98]).

**Generalisation.** New methods have been introduced recently that are very close to Krylov methods. These methods include the rational Krylov method, inner-outer iterations (which itself are just a form of preconditioning) and Galerkin approaches like Jacobi-Davidson (cf. [SvdV95, FSvdV96, SvdV95]).

**Inexact Methods.** In some applications accurate matrix vector products are very costly. It is currently an active line of research to find out *how* accurate the matrix vector products have to be in *which* stage of the algorithm to achieve the result up to a specified accuracy.

**Error Analysis.** The usual definition of a backward error is not appropriate for all Krylov methods. Backward error analysis has been applied successfully to GMRES by Rozložník (cf. [Roz97]). Progress towards a different type of backward error analysis has only been made for the symmetric Lanczos method and CG (cf. [Gre89, GS92]).

**Unification.** Approaches are seldom as general as they might have been. Many methods, their convergence theory, their error analysis, can be understood from a broader point of view.

This thesis is mainly concerned with the last two areas. The first two areas are developing too rapidly to be included. The major problem of *error analysis* lies in the non-normality of the matrices. The approaches that are the most promising are pseudospectra, the field of values and the degree of normality [TT94, TT96, Tre93, Tre99, Tre97].

Recent publications concerned with the last topic are the paper by Cullum and Greenbaum ([CG96]) on similarities of GMRES/FOM with BiCG/QMR, the paper by Hochbruck and Lubich ([HL98]) on Krylov methods in a nutshell and the work of Eiermann and Ernst ([EE99]) on the relations between MR and OR approaches. Another approach using an engineering viewpoint is the work of Schönauer and Weiss [SW95].

To our knowledge, no unified approach has been considered for the error analysis of Krylov subspace methods in finite precision. This will be our main concern.

## 1.4 Finite Precision and Error Analysis

We distinguish between three categories of error:

**Data Errors.** These types of error are due to imprecise measurement of the data or due to simplified models.

**Method Errors.** These types of error are due to discretisations and finite termination of infinite representations.

**Rounding Errors.** These types of error are due to the finite precision nature of computer arithmetic.

We consider only rounding errors as source of error, i.e., we assume that all calculations apart from the rounding errors occurring in the application of the methods are *exact*.

### 1.4.1 Floating Point Models

Computer arithmetic can only represent a *finite* subset of the natural, real and complex numbers. The representation of $\mathbb{N}$ and $\mathbb{Z}$ by $\mathbb{N}_n \equiv \mathbb{N}/n\mathbb{N}$ and $\mathbb{Z}_n \equiv \mathbb{Z}/n\mathbb{Z}$ introduces only *one* type of error, the so-called *wrap-around*. The computer representation inherits most algebraic properties of $\mathbb{N}$ and $\mathbb{Z}$.

The representation of $\mathbb{R}$ has been standardised in the eighties and is ruled by the ANSI/IEEE standards 754 and 854 ([IEE85b, IEE85c, IEE87]). The set $\mathbb{F}$ used to represent $\mathbb{R}$ consists of so-called *floating point* numbers. The field $\mathbb{C}$ is a subdivision algebra over $\mathbb{R}$ and is usually modelled by $\mathbb{F}^2$. A number in $\mathbb{F}$ consists of an array of fixed bit-length. One bit is reserved for the sign, a fixed number of bits represents the mantissa and the remaining bits are used to encode the exponent.

The mathematical result of an operation on floating point numbers only in trivial cases is part of $\mathbb{F}$. The *computer implementation* of the mathematical operations introduces new types of errors. When the true result is not part of $\mathbb{F}$, it has to be *rounded* to a number in $\mathbb{F}$. Numbers that are too small and too large to be representable are said to *underflow* and *overflow*, respectively.

To distinguish the true *mathematical* result form the computed *floating point* result we use the notation already used by Wilkinson (cf. [Wil63], section 7, page 4, equation (7.1)). Wilkinson used the notation $fl($`<term>`$)$ to denote the result of the floating point evaluation of `<term>`. Evaluation is assumed to be from left to right.

One part of theory is to develop rounding error models capturing the important features of the computer arithmetic in use, the other is to have models at hand that predict how solutions change when the data is perturbed according to these models. The very complicated behaviour of finite precision computations is modelled by making some assumptions on the distribution and size of absolute and relative errors.

#### ANSI/IEEE arithmetic

The ANSI/IEEE standards 754 and 854 define two formats of floating point numbers, four rounding modes and the return value of the four basic machine operations {`+`,`-`,`*`,`/`} and the square root. Furthermore, they introduce `Inf` and `NaN` (Not a Number) and exception handling to deal with the implications of underflow and overflow.

The formats are single precision with 32 bits and double precision with 64 bits. The rounding modes are round to nearest (nearest), round toward plus infinity (upward), round toward minus infinity (downward) and round to zero (chop). The four basic operations and the square root are defined to deliver the floating point number that is closest to the true result relative to the rounding mode.

The largest and smallest floating point number and the maximal relative distance between two floating point numbers are used to describe the behaviour of the floating point arithmetic. The smallest (denormalised) floating point number $\nu$ is given by

$$\begin{aligned} \nu_{\text{single}} &= 2^{-149} \approx 1.40130 \cdot 10^{-45}, \\ \nu_{\text{double}} &= 2^{-1074} \approx 4.94066 \cdot 10^{-324}, \end{aligned}$$

respectively. The relative distance is the distance between one and its floating point successor. This number is labelled *unit roundoff* and has a last bit in the mantissa equal to one, all other bits except the implicit one are equal to zero. The unit roundoff **u** is given by

$$\begin{aligned} \mathbf{u}_{\text{single}} &= 2^{-23} \approx 1.19209 \cdot 10^{-7}, \\ \mathbf{u}_{\text{double}} &= 2^{-52} \approx 2.22045 \cdot 10^{-16}. \end{aligned}$$

In standard mode the rounding is set to nearest. With this rounding mode half the unit roundoff **u** is the maximal relative error that can occur in a rounding operation. The resulting number

$$\begin{aligned} \epsilon_{\text{single}} &= 2^{-24} \approx 5.96046 \cdot 10^{-8}, \\ \epsilon_{\text{double}} &= 2^{-53} \approx 1.11022 \cdot 10^{-16}, \end{aligned}$$

is called the *machine precision*. When any other rounding mode is used, the resulting machine precision is equal to the unit roundoff.

Attention: in the open literature several contradictory definitions can be found. We use two of the most popular to give an example of the loss of the algebraic properties when computing in floating point:

**Example 1.1 (Loss of Algebraic Properties)** In floating point computations the usual algebraic properties no longer hold. As an example, consider the quantities defined as follows:

$$\begin{aligned} \tilde{\epsilon} &\equiv \max_{x \in \mathbb{F}} \{ fl(1+x) = 1 \} \\ \epsilon &= \min_{x \in \mathbb{F}} \{ fl(1+x-1) = x \}. \end{aligned}$$

With the above definitions $\tilde{\epsilon} \neq \epsilon$.

### Model accounting for underflow

The ANSI/IEEE standard is also an error *model*. This model is hard to use, since the complexity of the model and the arithmetic is essentially the same. Furthermore, the model consists of discrete points and is thus non-continuous. In order to get rid of this non-continuous character we consider the errors as unknowns and simply use the bounds on the basic operations that are implied by the ANSI/IEEE standard.

Let $\circ$ denote one of the five basic operations. The rounding errors fulfil the following:

$$fl(a \circ b) = (a \circ b)(1+\delta) + \mu, \qquad |\delta| \leq \epsilon, \quad |\mu| \leq \nu.$$

The additive term $\mu$ is only relevant in case an underflow occurs and is not necessary in case of addition and subtraction when considering ANSI/IEEE arithmetic, because the gradual underflow prevents the results in the denormalised range from being incorrect. This model is seldom used because of the additive term.

### Model not accounting for underflow

The model above is correct and useful, but not accounting for underflow makes it far more easy to develop error bounds, and underflow can be taken care of using exception handling. The new model looks as follows:

Let $\circ$ denote one of the five basic operations. As long as no underflow occurs, the rounding errors fulfil the following:

$$fl(a \circ b) = (a \circ b)(1+\delta_1) = \frac{(a \circ b)}{(1+\delta_2)}, \qquad |\delta_i| \leq \epsilon.$$

This model is the one that is mainly used in error analyses. In the following we act as if we have an extension valid also for complex arithmetic. Given an example of complex arithmetic it is easy to find such (slightly larger) constants.

**Using the error model**

Wilkinson is the main source of inspiration to numerical analysts. Wilkinson used a straight, *problem dependent* notation. Often he assumes that $n\epsilon < 1.01$ or similar. His style of error analysis gives very sharp results that very much depend on the actual algorithm and the machine used.

Another style of error analysis uses Landau's O–notation. This analysis is easier to follow and better to enhance in order to apply to different algorithms and machines. The drawback is that the hidden constants may become important. When a new type of algorithm or more sophisticated analysis is used, it may become necessary to redo any prior analysis.

Recent approaches to error analysis use different classes of bounded variables. A nice example of such an error analysis is given in Higham (cf. [Hig96]). From this three major styles that are used to capture the distribution of errors in more complex formula we chose the notation used in Higham.

Higham uses $\delta, \delta_i$ to denote quantities bounded by the machine precision, $|\delta_i| \leq \epsilon$. From the error model it is obvious that many expressions involving products of sums of one and $\delta$ occur. To bound the distance of the result from one, a new class of variables is defined by

$$\prod_{i=1}^{n} (1+\delta_i)^{\pm 1} \equiv 1 + \theta_n.$$

As long as $n\epsilon < 1$, the variables constructed this manner fulfil

$$|\theta_n| \leq \frac{n\epsilon}{1-n\epsilon} \equiv \gamma_n.$$

The class of variables is extended to *all* variables with this bound. Variables with this new bound that occur in expressions can be handled easily and fulfil simple algebraic rules:

**Lemma 1.2 ([Hig96], Lemma 3.3, page 74)** *For any positive integer $k$ let $\theta_k$ denote a quantity bounded according to $|\theta_k| \leq \gamma_k = k\epsilon/(1-k\epsilon)$. The following relations hold:*

$$\begin{aligned}
(1+\theta_k)(1+\theta_j) &= 1+\theta_{k+j}, \\
\frac{1+\theta_k}{1+\theta_j} &= \begin{cases} 1+\theta_{k+j}, & j \leq k, \\ 1+\theta_{k+2j}, & j > k, \end{cases} \\
\gamma_k \gamma_j &\leq \gamma_{\min(k,j)}, \\
i\gamma_k &\leq \gamma_{ik}, \\
\gamma_k + \gamma_j + \gamma_k\gamma_j &\leq \gamma_{k+j}.
\end{aligned}$$

The above results are an excerpt from Higham's book [Hig96].

### 1.4.2 Error Analysis

Suppose we want to compute $f(p) = x$, where $p$ is a given set of parameters. Instead of the true solution $x$ we compute $\tilde{x} = x + \Delta x$. The quantity $\Delta x$ is known as the *forward error*. The goal is to show that $\Delta x$ is small. The *error models* describe the errors in the atomic steps

$$x_0 = p, \quad f_i(x_0, \ldots, x_{i-1}) = x_i, \quad f(p) = x_k = x$$

every algorithm is composed of. These atomic steps are supposed to be given by the five standard operations.

There are essentially two approaches to determine a bound on the overall error $\Delta x$, namely *forward error analysis* and *backward error analysis* combined with *perturbation theory*. The advantage of splitting the error analysis into two parts is that the backward error analysis is *independent* of the sensitivity on the data, and the perturbation theoretic part is *independent* of the algorithm and has to be carried out only once for all algorithms.

#### Forward Error Analysis

Forward error analysis tries to keep track of the intermediate (local) errors $\Delta x_i$,

$$f_i(\tilde{x}_0, \ldots, \tilde{x}_{i-1}) = x_i + \Delta x_i,$$

to obtain a bound on the overall (global) error $\Delta x$. *First-order* forward error analysis is closely linked to bounding products of Jacobians of the composing functions $f_i$ (cf. [Hig96], pp. 82). The bounds obtained are usually pessimistic. An example of *successful application* of forward error analysis is the application to matrix inversion by von Neumann and Goldstine (cf. [vNG47]).

#### Backward Error Analysis

Backward error analysis takes another approach. Backward error analysis was introduced by Wilkinson and tries to *change the data* such that the *computed solution* is *exact* for the *modified problem*. The numerical analyst has to modify the parameters $p$ of the problem, that is, given $\Delta x$, determine any or all $\Delta p$ with

$$f(p + \Delta p) = x + \Delta x.$$

Mostly we seek $\Delta p$ with special structure, for example with minimal norm. The quantity $\Delta p$ is a *backward error*. *First-order* backward error analysis is closely linked to solve large (under-determined) systems (cf. [Hig96], pp. 82). Backward error analysis gives very successful results in the context of dense linear systems.

#### Stability of Algorithms

Closely linked to error analysis is the *stability* of the algorithm. An algorithm is *forward stable*, when it computes solutions that are close to the exact solution, i.e., when the forward error is small. An algorithm is *backward stable*, when it computes solutions that are exact solutions of a nearby problem, i.e., when the backward error is small. There are examples of algorithms that are forward stable without being backward stable.

We already noted that backward error analysis is independent of data sensitivity. In turn, backward error analysis gives no information directly on the forward error. This has to be done using *perturbation theory*.

### 1.4.3 Perturbation Theory

Perturbation theory tries to determine how the solution changes when the parameters are perturbed. The question we are interested in is the following: Given $\Delta p$, how does the deviation

$$\Delta x = f(p + \Delta p) - f(p)$$

look like?

In general, the task is the computation of measures when the perturbations stem from a *prescribed set*. We are interested in measuring the change in data, when we are perturbing the parameter according to a given set. The question is, given a set of parameter variations $P = \{\Delta p\}$ how does the set of solution variations $X = \{\Delta x : f(p + \Delta p) = x + \Delta x\}$ look like? There are two main approaches.

a) Algebra.

The idea is to insert a generic perturbation $\Delta p$ and then to determine resulting conditions on

$$\Delta x \equiv f(p + \Delta p) - f(p).$$

If this approach works, we usually obtain sharp bounds.

b) Calculus.

The idea is to use a parameterisation $p(t)$ of $p$, such that

$$p(0) = p, \quad p(\epsilon) = p + \Delta p = p(0) + \epsilon \dot{p}(0) + O(\epsilon^2).$$

We define the variation on $x$ as $\Delta x = \Delta x(\epsilon) = f(p(\epsilon)) - f(p(0))$. Computing derivatives with respect to $t$ gives the linearisation at zero,

$$\frac{d}{dt}\Delta x(t) = \frac{d}{dt} f(p(t)).$$

We estimate the variation $\Delta x = \Delta x(\epsilon)$ by the first term in the Taylor expansion, $\epsilon \Delta \dot{x}(0)$. The approach based on calculus is more often applicable, but naturally delivers only first-order bounds.

We are often only interested in bounding the *amplification factor* between a change in parameter and resulting change in the solution. This task is accomplished by condition numbers.

### Condition Numbers

A condition number is a *bound* on the set of changes computed using perturbation theory. A variety of useful condition numbers can be defined, see Wilkinson (cf. [Wil63], section 37, p.29). We define the *normwise condition number* with respect to scalar $z$ to be

$$\kappa_z(x) = \inf_{\epsilon > 0} \sup \left\{ \frac{\|\Delta x\|}{\|x\|} \epsilon : f(p + \Delta p) = x + \Delta x, \ \|\Delta p\| \leq \epsilon z \right\}.$$

Accordingly we define the *componentwise condition number* with respect to non-negative $z$ to be

$$\mathrm{cond}_z(x) = \inf_{\epsilon > 0} \sup \left\{ \max_i \frac{|\Delta x|_i}{|x|_i} \epsilon : f(p + \Delta p) = x + \Delta x, \ |\Delta p| \leq \epsilon z \right\}.$$

A condition number is called *absolute*, when $z = 1$, and *relative*, when $z = \|p\|$, $z = |p|$. If $f$ is differentiable, a condition number can be obtained from its derivative. Combining backward error analysis with perturbation theory gives information on the forward error $\Delta x$. This relation is usually stated in informal manner as a rule of thumb:

$$\text{forward error} \quad \leq \quad \text{condition number} \quad \times \quad \text{backward error.}$$

When the condition number is small, *backward* stable algorithms are also *forward* stable. A problem is termed *ill-conditioned* when the condition number is *large*, and *ill-posed* when the condition number is *infinite*. Ill-posed problems can not be solved in finite precision.

## 1.5 Solution of Linear Systems

In this section, the linear system under consideration will be denoted by

$$Ax = b,$$

where we assume that $A \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$ are given and we seek a solution $x \in \mathbb{K}^n$. This solution is *unique* when $A$ is *regular*. For regular $A$, the entries $x_i$ of the solution $x$ depend analytically on the entries of $A$ and $b$. This is known as *Cramer's rule*,

$$x_i = \frac{\det(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n)}{\det(A)}.$$

This relation is merely of theoretical interest.

The error analysis for linear systems of equations splits up into backward error analysis and application of results from perturbation theory, i.e., derivation of condition numbers. We start with stating the main results from perturbation theory. The results presented can be found in the textbook of Chaitin-Chatelin and Frayssé and in the textbook of Higham (cf. [CCF96, Hig96]).

### 1.5.1 Perturbation Theory

The linear system and the related perturbed system will be denoted by

$$Ax = b \quad \text{and} \quad \tilde{A}\tilde{x} = \tilde{b},$$

respectively. We suppose again that $A \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$ are given, and we seek $x \in \mathbb{K}^n$. For notational purposes we define the differences

$$\Delta A = \tilde{A} - A, \qquad \Delta x = \tilde{x} - x \qquad \text{and} \qquad \Delta b = \tilde{b} - b.$$

We interpret $\tilde{x}$ as an approximate solution of $Ax = b$ and denote the corresponding residual by $r = b - A\tilde{x}$. With these notations established, we are enabled to state a sequence of well-known results:

**Lemma 1.3 (Rigal, Gaches)** *The normwise backward error of an approximate solution $\tilde{x}$ of a linear system $Ax = b$ can be expressed as*

$$\begin{aligned} \eta_{\alpha,\beta}(\tilde{x}) &\equiv \min\{\epsilon : (A + \Delta A)\tilde{x} = b + \Delta b,\ \|\Delta A\| \le \epsilon\alpha,\ \|\Delta b\| \le \epsilon\beta\} \\ &= \frac{\|b - A\tilde{x}\|}{\alpha\|\tilde{x}\| + \beta}. \end{aligned}$$

*A perturbation that achieves equality is given by*

$$\Delta A = \frac{-\alpha}{\alpha\|x\| + \beta} r v^H, \qquad \Delta b = \frac{\beta}{\alpha\|x\| + \beta} r,$$

*where $v$ is a vector dual to $\tilde{x}$, i.e., chosen such that $v^H\tilde{x} = \|\tilde{x}\|$ and $\|v\|_d = 1$.*

**Proof.** The proof can be found in the original paper of Rigal and Gaches, dating back to 1967 (cf. [RG67]), and in the textbook of Chaitin-Chatelin and Frayssé (cf. [CCF96], p. 74). Here $\|\cdot\|_d$ is the dual norm to $\|\cdot\|$, defined by

$$\|x\|_d = \max_{v\neq 0} \frac{|v^H x|}{\|v\|}.$$

□

For the particular choice $\alpha = \|A\|$ and $\beta = \|b\|$ we obtain the *normwise relative backward error*.

**Lemma 1.4 (Turing)** *The normwise condition number of the linear system $Ax = b$ can be expressed as*

$$\begin{aligned}\kappa_{\alpha,\beta}(A,x) &\equiv \inf_{\epsilon>0}\sup\left\{\frac{\|\Delta x\|}{\|x\|}\epsilon : \tilde{A}\tilde{x} = \tilde{b},\ \|\Delta A\| \le \epsilon\alpha,\ \|\Delta b\| \le \epsilon\beta\right\} \\ &= \frac{\|A^{-1}\|(\alpha\|x\| + \beta)}{\|x\|}.\end{aligned}$$

**Proof.** This condition number has its origins in the works of Turing. The proof of the lemma can be found in the textbook of Chaitin-Chatelin and Frayssé (cf. [CCF96], p. 50). □

The same approach works componentwise. Here, we suppose that we are given a nonnegative matrix $E \in \mathbb{R}$ and a nonnegative vector $f \in \mathbb{R}$ that are the entrywise bounds for the perturbations in $A$ and $b$, respectively. Then the following can be shown to hold true:

**Lemma 1.5 (Oettli, Prager)** *The componentwise backward error of an approximate solution $\tilde{x}$ of a linear system $Ax = b$ can be expressed as*

$$\begin{aligned}\omega_{E,f}(x) &\equiv \min\{\epsilon : (A+\Delta A)\tilde{x} = b + \Delta b,\ |\Delta A| \le \epsilon E,\ |\Delta b| \le \epsilon f\} \\ &= \max_i \frac{|b - Ax|_i}{(E|x| + f)_i}\end{aligned}$$

*A perturbation that achieves equality is given by*

$$\begin{aligned}\Delta A &= \operatorname{diag}\left(\frac{(E|\tilde{x}| + f)_i}{r_i}\right) E \operatorname{diag}\left(\frac{\overline{\tilde{x}}_i}{|x|_i}\right), \\ \Delta b &= -\operatorname{diag}\left(\frac{(E|\tilde{x}| + f)_i}{r_i}\right) f.\end{aligned}$$

**Proof.** This result is due to Oettli and Prager and dates back to 1964 (cf. [OP64]). The proof again may be found in the textbook of Chaitin-Chatelin and Frayssé (cf. [CCF96], p. 74). □

Similar to the normwise analysis we define

$$\omega_{|A|,|b|}(x) = \max_i \frac{|b - Ax|_i}{(|A||x| + |b|)_i}$$

as the *componentwise relative backward error*.

**Lemma 1.6 (Rohn)** *The componentwise condition number of the linear system $Ax = b$ can be expressed as*

$$\begin{aligned}cond_{E,f}(A,x) &\equiv \inf_{\epsilon>0}\sup\left\{\max_i \frac{|\Delta x_i|}{|x_i|}\epsilon : \tilde{A}\tilde{x} = \tilde{b},\ |\Delta A| \le \epsilon E,\ |\Delta b| \le \epsilon f\right\} \\ &= \max_i \frac{\left(\left|A^{-1}\right|(E|x| + f)\right)_i}{|x|_i}.\end{aligned}$$

**Proof.** For a proof we refer the reader to the textbook of Chaitin-Chatelin and Frayssé (cf. [CCF96], p. 50). □

The componentwise condition number originates from work by Rohn. All backward errors and condition numbers do *not* depend on any structure. We briefly remark that the condition number and the backward error with respect to *structured* perturbations may be significantly smaller than indicated by the general results.

### 1.5.2 Decompositions and Error Analysis

The naïve solution of a linear system $Ax = b$ by application of Cramer's rule *directly* to the matrix $A$ and the left-hand side $b$ involves too much work. As preceeding, or better, *pre-processing* step the linear system is transformed to a *couple* of linear systems which are simpler to solve, i.e., we first *reformulate* the equation $Ax = b$. The idea that turned out to be very successful is to *decompose* or *factor* the matrix $A$. Whenever determinants are cheaply computable, Cramer's rule becomes applicable. We seek a factorisation as a sequence of triangular or mixed triangular and orthogonal (unitary) matrices.

#### Triangular Decompositions

The oldest idea for the solution of a linear system consists of replacing $A$ by a product of triangular matrices. The decompositional approach with triangular or block triangular matrices is known as *Gaussian elimination*. We distinguish between Gaussian elimination (GE), GE with partial pivoting (GEPP) and GE with complete pivoting (GECP),

$$A = LU, \qquad PA = LU, \qquad PAQ = LU.$$

The matrix $L$ is lower triangular, the matrix $U$ is upper triangular, $P$ and $Q$ are permutation matrices. To fix the number of equations and the number of unknowns, usually the diagonal elements of $L$ are set to one.

All results on decompositions are based on the following lemma:

**Lemma 1.7 ([Hig96], Lemma 8.4, page 154)** *If $y = (c - \sum_{i=1}^{k-1} a_i b_i)/b_k$ is evaluated in floating point arithmetic, then, no matter what the order of evaluation,*

$$b_k \tilde{y}(1 + \theta_k^{(k)}) = c - \sum_{i=1}^{k-1} a_i b_i (1 + \theta_k^{(i)}),$$

*where $\tilde{y}$ is the computed solution and $|\theta_k^{(i)}| \leq \gamma_k$ for all $i$. If $b_k = 1$, so there is no division, then $|\theta_k^{(i)}| \leq \gamma_{k-1}$ for all $i$.*

The backward error analysis is the same for all forms of GE. First, permute the system matrix $\hat{A} \leftarrow PAQ$. Then suppose we have already permuted $A$, such that $A = LU$ holds true. We have the following bound regardless of the GE variant used. The following holds true:

**Lemma 1.8 ([Hig96], Theorem 9.3, page 175)** *Suppose we have computed triangular factors $\tilde{L}$ and $\tilde{U}$. Then the following backward bound holds true:*

$$A + \Delta A = \tilde{L}\tilde{U}, \quad |\Delta A| \leq \gamma_n |\tilde{L}||\tilde{U}|.$$

For the *solution of linear systems by GE* one can prove the following bound:

**Lemma 1.9 ([Hig96], Theorem 9.4, page 175)** *Suppose we have computed triangular factors $\tilde{L}$ and $\tilde{U}$. Suppose further that we have used forward and backward substitution to compute an approximate solution $\tilde{x}$. Then we obtain the backward bound*

$$(A + \Delta A)\tilde{x} = b, \quad |\Delta A| \leq \gamma_{3n} |\tilde{L}||\tilde{U}|.$$

How are these *a posteriori* bounds that involve the *a priori* unknown quantities $\tilde{L}$ and $\tilde{U}$ related to the original data? When the computed triangular factors satisfy

$$|\tilde{L}||\tilde{U}| \approx |\tilde{L}\tilde{U}| \approx |A|\,,$$

the algorithm is backward stable. When the LU decomposition is carried out in infinite precision, we can measure the so-called growth factor, defined as

$$\rho_n(A) \equiv \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

The upper index $k$ in $a_{ij}^{(k)}$ is used to denote the elements of $A$ in the $k$th step of the LU decomposition. Using the definition of the growth factor, Wilkinson proved the following theorem:

**Lemma 1.10 ([Hig96], Theorem 9.5, page 176)** *Let the growth factor be defined as above. Then the approximate solution $\tilde{x}$ to the linear system $Ax = b$ obtained by GE and forward and backward substitution satisfies*

$$(A + \Delta A)\tilde{x} = b, \quad \|\Delta A\|_\infty \leq 2n^2\gamma_n\rho_n\|A\|_\infty.$$

This lemma uses the element growth in an *exact computation* to bound the errors in a *finite precision computation*. We refer to the comment in Higham's book (cf. [Hig96], page 176).

GE, GEPP and GECP applied to a general matrix have growth factors with

$$\rho_n^{\text{GE}} \leq \infty, \quad \rho_n^{\text{GEPP}} \leq 2^{n-1}, \quad \rho_n^{\text{GECP}} \leq n^{1/2}(2 \cdot 3^{1/2} \cdots n^{1/n-1}),$$

respectively. The first two bounds are sharp.

### Special Forms of Triangular Decomposition

If the LU decomposition of $A$ without pivoting exists, i.e., when *all leading principal minors are nonzero*, we define the LDMT and LDMH decompositions

$$A = LDM^T, \qquad A = LDM^H,$$

where $L$ and $M$ are lower triangular matrices with unit diagonal, $D$ is a diagonal matrix. When $A$ is *symmetric* or *Hermitian*, this decomposition reduces to the LDLT or LDLH decomposition,

$$A = LDL^T, \qquad A = LDL^H.$$

When $A$ is *symmetric positive definite* (SPD) or *Hermitian positive definite* (HPD), we can choose $C = L\sqrt{D}$ resulting in the Cholesky decomposition, given by

$$A = CC^T, \qquad A = CC^H.$$

The LDLT decomposition can be interpreted as a generalisation of the Cholesky decomposition. This decomposition is sometimes termed the rational Cholesky decomposition because of its suitability in fields other than $\mathbb{R}$ or $\mathbb{C}$.

When $A$ is symmetric *indefinite*, there exist at least two well-known methods that compute a generalised LDLT decomposition, namely Aasen's method and the Bunch-Kaufmann decomposition. The error analysis for these decompositions is similar to the error analysis of GE. We only state one well-known backward result:

**Theorem 1.11 ([Hig96], Theorem 10.3, page 206)** *If Cholesky factorisation applied to the symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ runs to completion then the computed factor $\tilde{C}$ satisfies*

$$\tilde{C}\tilde{C}^T = A + \Delta A, \qquad |\Delta A| \leq \gamma_{n+1}|\tilde{C}||\tilde{C}^T|.$$

*Furthermore, the right-hand side can be bounded normwise using*

$$\| \, |\tilde{C}||\tilde{C}^T| \, \| \leq n(1 - n\gamma_{n+1})^{-1}\|A\|$$

*and componentwise using*

$$|\tilde{C}||\tilde{C}^T| \leq (1 - \gamma_{n+1})^{-1} dd^T, \quad d_i = \sqrt{a_{ii}}.$$

**Proof.** All results can be found in Higham's book. The normwise bound can be found on page 206, the componentwise bound in the proof of Theorem 10.5, page 207. □

Bounds similar to the corresponding ones in the analysis of GE hold true for the solution of linear systems using Cholesky decomposition. We note that the growth factor in GE for $A$ SPD is one.

### Triangular Decomposition for Matrices with Special Structure

Special structure has impacts on the bounds of the growth factor. This may be the non-zero pattern of the matrix or other properties. The matrices we are mainly concerned with are Hessenberg and tridiagonal matrices. We collect some useful results in the following lemma:

**Lemma 1.12** *Let $A \in \mathbb{C}^{n\times n}$ be diagonally dominant by rows or columns. Then GE works and the growth factor is bounded by $\rho_n \leq 2$. GEPP requires no row interchanges.*

*Let $H \in \mathbb{C}^{n\times n}$ be an upper Hessenberg matrix. Then for the growth factor $\rho_n^p$ in GEPP $\rho_n^p \leq n$ holds true.*

*Let $T \in \mathbb{R}^{n\times n}$ be a nonsingular tridiagonal matrix. If any of the following conditions holds, then $T$ has an $LU$ factorisation and $|L||U| = |LU|$:*

*(a) $T$ is symmetric positive definite;*

*(b) $T$ is totally nonnegative, or equivalently, $L \geq 0$ and $U \geq 0$;*

*(c) $T$ is an M-matrix, or equivalently, $L$ and $U$ have positive diagonal elements and nonpositive off-diagonal elements;*

*(d) $T$ is sign equivalent to a matrix of type (a)–(c), that is, $T = S_1\tilde{T}S_2$, where $|S_1| = |S_2| = I$.*

*Suppose further that the unit roundoff $\mathbf{u}$ is sufficiently small. Then GE for solving $Tx = b$ succeeds and the computed solution $\tilde{x}$ satisfies*

$$(T + \Delta T)\tilde{x} = b, \qquad |\Delta T| \leq \frac{4\mathbf{u} + 3\mathbf{u}^2 + \mathbf{u}^3}{1 - \mathbf{u}}|T|.$$

**Proof.** These results are an excerpt of theorems in Higham's book ([Hig96]). The result on diagonal dominance is Theorem 9.8, page 181, the Hessenberg result is Theorem 9.9, page 182, the conditions for $|L||U| = |LU|$ for tridiagonals are collected in Theorem 9.11, page 184 and the bound on the computed solution is Theorem 9.13, page 185. □

LR decomposition is very stable for so-called *M-matrices*. M-matrices enjoy very special properties. These properties can, to some extent, generalised from the set

of M-matrices to the so-called *H-matrices*. H-matrices are defined with the aid of Ostrowski's comparison matrix $\mathcal{M}(A)$,

$$\mathcal{M}(A) \equiv \text{abs}(D) - \text{abs}(A - D), \qquad D \equiv \text{diag}(A).$$

A matrix $A$ is an H-matrix when $\mathcal{M}(A)$ is an M-matrix. The set of equimodular matrices $\omega(A)$ to a given M-matrix $A$ is defined by

$$\omega(A) \equiv \{H,\ \mathcal{M}(H) = A\}.$$

For H-matrices special bounds for the growth factor exist.

**Orthogonal Decompositions**

We can write any matrix as product of an orthogonal or unitary matrix and a triangular matrix, i.e.,

$$A = QR \quad \text{or} \quad A = LQ.$$

The matrix $R$ is upper triangular, $L$ is lower triangular, $Q$ is unitary. The first is known as QR decomposition, the latter as LQ decomposition. The first $Q$ in general is not equal to the second $Q$. There is no need for $A$ to be square, so in the following we assume $A \in \mathbb{K}^{m\times n}$, $m \geq n$. The formerly triangular matrices $L$ and $R$ are sometimes termed *trapezoidal* matrices.

There are several ways to obtain the QR decomposition of a given matrix $A$. The QR decomposition can be computed by means of Householder reflectors (Householder QR), by means of Givens rotators (Givens QR), by means of Givens-Kahan rotators or by means of classical or modified Gram-Schmidt on the columns of $A$.

We state some results along the lines of Higham (cf. [Hig96]). First we state the result on the stability of the QR decomposition using Householder reflectors and the backward error analysis of a subsequent solution of a linear system.

**Lemma 1.13 ([Hig96], Theorem 18.4, page 368)** *Let $\tilde{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ obtained via the Householder QR algorithm. Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$A + \Delta A = Q\tilde{R},$$

*where $\|\Delta A\|_F \leq n\gamma_{cm}\|A\|_F$ and $|\Delta A| \leq mn\gamma_{cm}G|A|$, with $\|G\|_F = 1$. The matrix $Q$ is given explicitly as $Q = (P_nP_{n-1}\dots P_1)^T$, where $P_k$ is the Householder matrix that corresponds to the exact application of the kth step of the algorithm to $\tilde{A}_k$.*

**Lemma 1.14 ([Hig96], Theorem 18.5, page 369)** *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. Suppose we solve the system $Ax = b$ with the aid of a QR factorisation computed by the Householder algorithm. The computed $\tilde{x}$ satisfies*

$$(A + \Delta A)\tilde{x} = b + \Delta b,$$

*where*

$$|\Delta A| \leq n^2\gamma_{cn}G|A|, \quad |\Delta b| \leq n^2\gamma_{cn}G|b|, \qquad \|G\|_F = 1.$$

Very similar bounds hold true, when the Givens QR variant is used.

**Lemma 1.15 ([Hig96], Theorem 18.9, page 375)** *Let $\tilde{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ obtained via the Givens QR algorithm, with any standard choice and ordering of rotations. Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$A + \Delta A = Q\tilde{R},$$

*where* $\|\Delta A\|_F \leq \gamma_{c(m+n)}\|A\|_F$ *and* $|\Delta A| \leq m\gamma_{c(m+n)}G|A|$, *with* $\|G\|_F = 1$. *The matrix* $Q$ *is a product of Givens rotations, the* $k$*th of which corresponds to the exact application of the* $k$*th step of the algorithm to* $\tilde{A}_k$.

QR decomposition via Householder reflections or Givens rotations is normwise backward stable. These methods usually fail to be componentwise backward stable, because of the occurring matrices $G$ with $\|G\|_F = 1$. Classical Gram–Schmidt, short CGS, is numerically unstable. The only thing one can prove is that the computed QR decomposition has a small backward error:

**Lemma 1.16 ([Hig96], page 381)** *Let* $\tilde{Q}$ *and* $\tilde{R}$ *be the computed factors of the QR decomposition. For CGS the following bound holds:*

$$A + \Delta A = \tilde{Q}\tilde{R}, \qquad \|\Delta A\|_2 \leq c\mathbf{u}\|A\|_2,$$

*where* $c$ *depends only on the dimensions* $m, n$.

Paige and Björk worked on the correspondence of MGS-QR to Householder-QR for an augmented matrix (cf. [BP92]). The correspondence holds in infinite *as well as in finite precision* and is given by

$$A = QR, \quad P^T \begin{pmatrix} 0 \\ A \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where the Householder reflectors are given by the matrices

$$I - p_k p_k^T, \quad p_k = \begin{pmatrix} -e_k \\ q_k \end{pmatrix}.$$

This result can be used to obtain a backward error result on MGS-QR, which shows that the $R$ factor is as stable as the $R$ factor in a Householder or Givens computation, only the computed $Q$ factor deviates. In this paper, also an example how to obtain a *better* $Q$ factor from the *computed* $Q$ factor is given.

**Lemma 1.17 ([Hig96], Theorem 18.12, page 379)** *Suppose that the MGS method is applied to* $A \in \mathbb{R}^{m\times n}$ *of rank* $n$, *yielding computed matrices* $\tilde{Q} \in \mathbb{R}^{m\times n}$ *and* $\tilde{R} \in \mathbb{R}^{n\times n}$. *Then there are constants* $c_i \equiv c_i(m, n)$ *such that*

$$\begin{aligned} A + \Delta A_1 = \tilde{Q}\tilde{R}, \quad \|\Delta A_1\|_2 \leq c_1\mathbf{u}\|A\|_2, \\ \|\tilde{Q}^T\tilde{Q} - I\|_2 \leq c_2\mathbf{u}\kappa_2(A) + O((\mathbf{u}\kappa_2(A))^2), \end{aligned}$$

*and there exists an orthonormal matrix* $Q$ *such that*

$$A + \Delta A_2 = Q\tilde{R}, \quad |\Delta A_2| \leq c_3\mathbf{u}G|A|, \quad \|G\|_F = 1.$$

As already mentioned, for some methods the computation of QR decomposition is normwise backward stable, but not componentwise. To better understand the sensitivity of QR decompositions, Stewart considered normwise, Zha and Sun considered componentwise sensitivity analysis of perturbed QR decompositions,

$$A = QR, \qquad A + \Delta A = (Q + \Delta Q)(R + \Delta R).$$

In the normwise case, for sufficiently small $\Delta A$,

$$\max\left\{\|\Delta Q\|_F, \frac{\|\Delta R\|_F}{\|R\|_F}\right\} \leq c_n\kappa_F(A)\frac{\|\Delta A\|_F}{\|A\|_F}$$

holds true, where $c_n$ is a constant. In the componentwise case the perturbation $\Delta A$ is assumed to be bounded by $|\Delta A| \leq \epsilon G|A|$, $G \geq 0$, $\|G\|_\infty = 1$. Then, for sufficiently small $\epsilon$,

$$\max\left\{\|\Delta Q\|_\infty, \frac{\|\Delta R\|_\infty}{\|R\|_\infty}\right\} \leq c_{m,n}\mathrm{cond}(R^{-1}) + O(\epsilon^2)$$

holds true, where $c_{m,n}$ is a constant depending on $m$ and $n$ and $\mathrm{cond}(A) = \| \, |A^{-1}||A| \, \|_\infty$. The componentwise condition number of the factors of the QR decomposition is defined to be

$$\phi(A) \equiv \mathrm{cond}(R^{-1}) = \| \, |R||R^{-1}| \, \|_\infty.$$

These results are stated in [Hig96], pages 382–383.

### Structured Error Analysis

Many algorithms make extensive use of known structure, simply to perform better. This introduces only errors that are of the same structure. But the preceeding analyses only took the norm or the componentwise absolute value into account. We remark that for many classes of matrices like symmetric, skew-symmetric, Toeplitz, Hankel, Bezout, banded matrices suitable condition numbers and backward errors can be defined and bounded or even explicitly computed.

### Generalised Inverses

The inverse of a matrix is only defined as long the matrix is regular. This regularity necessarily implies that the matrix is moreover *square*. Under various circumstances *generalisations* taking the place of the matrix inverse exist. All these will be termed *generalised inverses*. Frequently, given the equation $Ax = b$, we are interested in a linear mapping $X$, such that $x = Xb$ is a solution *whenever* a solution exists, i.e., when $b \in \mathrm{range}(A)$. This is equivalent to the fulfilment of the condition $AXA = A$. The most prominent among all generalised inverses $X$ is the classical *Moore-Penrose inverse* or *pseudo-inverse*. The pseudo-inverse $A^\dagger$ of $A \in \mathbb{K}^{m\times n}$ is given by the matrix $X$ *uniquely* defined by the following four axioms:

$$\begin{array}{llcll} \text{(I)} & AXA & = & A, & \qquad \text{(III)} \quad (AX)^H = AX, \\ \text{(II)} & XAX & = & X, & \qquad \text{(IV)} \quad (XA)^H = XA. \end{array}$$

When the matrix is square, also the following condition is of interest:

$$\text{(V)} \quad XA = AX$$

Any matrix $X$ that fulfils a subset $\eta$ of the equations (I–V) is termed an $\eta$–inverse and is denoted by $A^\eta$. The Moore-Penrose pseudo-inverse is thus given by $A^\dagger = A^{\{\mathrm{I,II,III,IV}\}}$. A matrix fulfilling the important condition (I) is also known as a *g–inverse* or *condition one* inverse. Any matrix $A^{\{\mathrm{I,III}\}}$ is a *least–squares* generalised inverse and any matrix $A^{\{\mathrm{II,IV}\}}$ is a *minimum norm* generalised inverse. The so-called *group inverse* $A^{\{\mathrm{I,II,V}\}}$ exists and is uniquely defined when $\mathrm{rank}(A^2) = \mathrm{rank}(A)$. The group inverse is denoted by $A^\#$.

When $A$ is square, but not necessarily $\mathrm{rank}(A^2) = \mathrm{rank}(A)$, we can still find a generalised inverse that fulfils the conditions (II), (V) and

$$\text{(D)} \quad A^{k+1}X = A^k$$

where $k$ is the *index* $\mathrm{ind}(A)$ of the matrix $A$, i.e., the smallest non-negative integer $k$ such that $\mathrm{rank}(A^{k+1}) = \mathrm{rank}(A^k)$. This matrix is denoted by $A^D$ and is termed the *Drazin inverse* of $A$. The Drazin inverse is *uniquely* defined by these axioms,

but in general does not fulfil condition (I), i.e., is *not* a g–inverse. The group inverse is merely a special case of the Drazin inverse, where $\mathrm{ind}(A) = 1$. This can be seen since in this case conditions (D) and (V) imply condition (I). The Drazin inverse is important in areas where algebraic structure (group structure) has to be taken into account.

## 1.6 The Algebraic Eigenproblem

The *algebraic eigenvalue problem*, or shorter and more precise, the *algebraic eigenproblem*, is the following. Given a matrix $A \in \mathbb{K}^{n\times n}$, we seek a scalar $\lambda$ and a (non-zero) vector $v$, such that

$$Av = v\lambda$$

holds true. This is a *partial* eigenproblem. The scalar $\lambda$ is termed an *eigenvalue*, the corresponding vector $v$ is termed the *right eigenvector* or simply *eigenvector*. Sometimes we also seek vectors $\hat{v}^H$ or $\check{v}^T$ such that with the value $\lambda$ as above

$$\hat{v}^H A = \lambda \hat{v}^H \quad \text{or} \quad \check{v}^T A = \lambda \check{v}^T$$

holds true. In this case, the vectors $\hat{v}^H$ and $\check{v}^T$ are termed *right eigenvectors*. The algebraic eigenvalue problem is *not* a *linear* problem, since it involves a product of two unknowns. Nevertheless, it is a *linear algebra problem* since it clarifies the properties of a linear operator. Depending on the domain of application, we might be interested in different quantities. We might want to compute *all* or only *a few* eigenvalues, only the *eigenvalues* or also the corresponding *right*, and maybe even the *left* eigenvectors.

We start with the properties of the eigenvalues. All eigenvalues are due to

$$\begin{aligned} Av = v\lambda &\Leftrightarrow (\lambda I - A)v = 0 \\ &\Rightarrow \chi_A(\lambda) \equiv \det(\lambda I - A) = 0 \end{aligned}$$

roots of the characteristic polynomial $\chi = \chi_A$ of $A$. The eigenvalues depend analytically on the matrix entries. Counting with multiplicity, every matrix $A \in \mathbb{K}^{n\times n}$ has $n$ eigenvalues. If the root of the characteristic polynomial is simple, the eigenvalue is termed *simple*, and *multiple* otherwise. The multiplicity of an eigenvalue interpreted as *root* of the characteristic polynomial is termed its *algebraic multiplicity*. The *set* of all eigenvalues is the *spectrum* of $A$ and denoted by $\Lambda = \Lambda(A)$. The value

$$\rho(A) \equiv \max_{\lambda\in\Lambda(A)} |\lambda|$$

is the *spectral radius* of the matrix $A$.

Now we switch to the eigenvectors. When $\lambda$ is an eigenvalue of $A$, the preceeding computations also show that $\lambda I - A$ is rank deficient. Every non-zero vector in the nullspace is an eigenvector to eigenvalue $\lambda$. Since the nullspace is at least one-dimensional, this implies that for every eigenvalue $\lambda$ there must exist at least *one* right eigenvector $v$ (and *one* left eigenvector $\hat{v}^H$). The dimension of the nullspace of $\lambda I - A$ is termed the *geometric multiplicity* of the eigenvalue $\lambda$. The geometric multiplicity is less or equal to the algebraic multiplicity.

With these definitions at hand, we can classify eigenvalues in more detail. An eigenvalue $\lambda$ is termed *derogatory* when the geometric multiplicity is larger than one and else *non-derogatory*. When the geometric multiplicity is strictly less than the algebraic multiplicity, the eigenvalue is *defective*. When the geometric multiplicity is equal to the algebraic multiplicity but greater than one, the eigenvalue is *semi-simple*. These terms extend naturally to matrices.

### 1.6.1 Related Decompositions and Their Uniqueness

As already mentioned, we may be interested only in eigenvalues, or in both eigenvalues and eigenvectors. We may be interested in eigenvalues and in a stable basis of the corresponding invariant subspaces. These tasks are solved by the computation of the *Jordan* and *Schur* decompositions. To introduce these *normal forms*, we observe that a so-called *similarity transformation*

$$\mathbb{K}^{n\times n} \to \mathbb{K}^{n\times n} \qquad A \mapsto A_X \equiv X^{-1}AX$$

does not alter the characteristic polynomial (and thus the eigenvalues),

$$\det(\lambda I - A) = \det(X^{-1})\det(\lambda I - A)\det(X) = \det(\lambda I - A_X)$$

and changes the eigenvectors in prescribed ways,

$$A_X(X^{-1}v) = X^{-1}Av = \lambda(X^{-1}v).$$

By definition of the eigenvalues it is obvious that the eigenvalues of an upper (or lower) *triangular* matrix appear on the diagonal. When the matrix is overmore *diagonal*, we explicitly know the eigenvectors. So we seek for a similarity transformation that maps the given matrix to a matrix that is 'as close as possible' to a diagonal matrix. This results in the Jordan decomposition. When we impose the restriction that the similarity transformation is a unitary one, we obtain the Schur decomposition.

#### Jordan Decomposition

This decomposition is for *theoretical* purposes of *eminent* interest. As example, it can be used to express the solutions to some ordinary value problems algebraically. In contrast, it is almost *non-computable* numerically because of inherent instability and thus only of minor interest when it comes to scientific computing. The Jordan normal form reveals that 'as close as possible' to a diagonal matrix in general is a merely bi-diagonal matrix with ones and zeros on the off-diagonal.

**Theorem 1.18 (The Jordan Normal Form)** *Every matrix $A \in \mathbb{K}^{n\times n}$ is* similar *to a so-called* Jordan matrix

$$J_\Lambda = V^{-1}AV.$$

*This relation can be stated alternatively as*

$$A = VJ_\Lambda V^{-1} \quad \Leftrightarrow \quad AV = VJ_\Lambda \quad \Leftrightarrow \quad V^{-1}A = J_\Lambda V^{-1}.$$

*The Jordan matrix $J_\Lambda = J_{\lambda_1} \oplus \cdots \oplus J_{\lambda_l}$ is composed of a direct sum of smaller* Jordan blocks

$$J_{\lambda_i} = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix} \equiv \lambda_i I_{\ell_i} + N_{\ell_i}.$$

*The matrices $N_{\ell_i}$ are nilpotent, where $\ell_i$ is not only the dimension, but also the smallest integer $k$ such that $N_{\ell_i}^k = 0$.*

*The Jordan matrix is* unique *up to permutations of the Jordan blocks. There may occur multiple Jordan blocks of different sizes to one value $\lambda$.*

**Proof.** The proof can be found in most textbooks on linear algebra, see for instance the textbook by Horn and Johnson (cf. [HJ85], Theorem 3.1.11, page 126) or the treatise of the eigenproblem by Chatelin (cf. [Cha93], Theorem 1.6.7, page 25). □

By inspection, the scalars $\lambda_i, i \in \underline{l}$ are the eigenvalues. To *every Jordan block* corresponds *one right* and *one left* eigenvector given by a column of $V$ or $\hat{V}$, defined by $\hat{V} \equiv V^{-1}$, respectively. The columns of $V$ or $\hat{V}$ that are *no* eigenvectors are termed *generalised eigenvectors* or *principal vectors*. Despite this fact we refer to the matrix $V$ as the matrix of eigenvectors or simply as *eigenmatrix*. The index of the matrix $N_\ell$ corresponds to the dimension of the largest Jordan block to eigenvalue $\lambda$ and will be termed the *ascent* of $\lambda$.

The Jordan decomposition, when written in the form

$$A = \sum_\ell V_\ell(\lambda_\ell I_\ell + N_\ell)\hat{V}_\ell^H,$$

where $V_\ell$ and $\hat{V}_\ell$ consist of the columns of $V$ and $\hat{V}$ corresponding to *distinct* $\lambda_\ell$, is known as the *spectral decomposition* of $A$. Sometimes this notion is used only in conjunction with Hermitian matrices (cf. [SS90], page 19), sometimes also in the general case (cf. [Cha93], Theorem 1.7.1, page 27). The matrices $N_\ell$ are nilpotent, $N_\ell^\ell = 0$, where $\ell$ is the ascent of the eigenvalue $\lambda$, i.e., the size of the largest Jordan block to eigenvalue $\lambda$.

We denote the diagonal of the Jordan matrix by $\Lambda$. When $A$ is diagonalisable, the Jordan matrix is diagonal, i.e., $\Lambda = J_\Lambda$. $A$ is diagonalisable when all eigenvalues are semi-simple, and $A$ is also termed *semi-simple*. $A$ is *not* diagonalisable when $A$ is *defective*, i.e., when there exists at least one defective eigenvalue. Diagonalisable matrices are also called *non-defective*.

The Jordan *matrix* is determined uniquely up to permutations of the Jordan *blocks*. In contrast, the *eigenmatrix* that transforms $A$ to Jordan normal form is *not uniquely defined*. We derive a theorem on the class of transformations that map a given eigenmatrix to another eigenmatrix.

**Theorem 1.19 (Non-Uniqueness of the Eigenmatrix)** *Eigenmatrices are* not uniquely defined. *Let $V$ be an eigenmatrix that maps $A$ to Jordan normal form $V^{-1}AV = J_\Lambda$. Denote by $\mathcal{T} = \mathcal{T}(J_\Lambda)$ the set of regular matrices that commute with the Jordan normal form $J_\Lambda$ of $A$. Denote by $\mathcal{V}$ the* set of eigenmatrices *that map $A$ to the* same *Jordan matrix.*

*The set of eigenmatrices $\mathcal{V}$ has the representation*

$$\mathcal{V} \;=\; V\mathcal{T} \;=\; \{VT,\; T \in \mathcal{T}\}.$$

*The set $\mathcal{T}$ can be described explicitly. It is a subset of the set of* regular block matrices *partitioned according to the Jordan blocks $J_i$, $i \in \underline{k}$ in the Jordan matrix $J = J_\Lambda$,*

$$J \;=\; \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_k \end{pmatrix}, \qquad T \;=\; \begin{pmatrix} T_{11} & \cdots & T_{1k} \\ \vdots & \ddots & \vdots \\ T_{k1} & \cdots & T_{kk} \end{pmatrix}.$$

*The matrices $T_{ij}$ on the off-diagonals may be rectangular. The subset is then determined by choosing the block entry $T_{ij}$ as a* zero *matrix of appropriate dimensions when $J_i$ and $J_j$ correspond to* different *eigenvalues, and* upper triangular Toeplitz

*else. When $T_{ij}$ is rectangular, we mean by* upper triangular *a matrix of type*

$$T_{ij} = \begin{pmatrix} 0 & \cdots & 0 & \star & \cdots & \star \\ \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \star \end{pmatrix} \quad \text{or} \quad T_{ij} = \begin{pmatrix} \star & \cdots & \star \\ 0 & \ddots & \vdots \\ \vdots & \ddots & \star \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}.$$

*Furthermore, we can decompose every matrix $T \in \mathcal{T}$ using a block LR-type decomposition with partial pivoting into*

$$PT = LR \quad \Leftrightarrow \quad T = P^T LR.$$

*Here, $L$ is* block lower triangular *with identity matrices on the diagonal blocks, $R$ is* block upper triangular. *The upper triangular blocks of $R$ and the strict lower triangular blocks of $L$ are again upper triangular Toeplitz. The matrix $P$ is a* block permutation *that mixes* only *rows corresponding to* equal sized *Jordan blocks to the* same *eigenvalue.*

**Proof.** This theorem is based on some ideas of Zemke (cf. [Zem97]). Two different eigenmatrices $V$, $\tilde{V}$ to the same Jordan matrix must necessarily fulfil

$$V^{-1} J_\Lambda V = \tilde{V}^{-1} J_\Lambda \tilde{V} \quad \Rightarrow \quad J_\Lambda V \tilde{V}^{-1} = V \tilde{V}^{-1} J_\Lambda. \tag{1.1}$$

This proves that the matrix $T \equiv V\tilde{V}^{-1}$ commutes with the Jordan matrix and is regular. We write a generic $T$ as block matrix. Multiplication by a block-diagonal matrix from the *left* is a multiplication of the block *columns*, multiplication from the *right* is a multiplication of the block *rows*. A typical block equation resulting from equation (1.1) thus has the form

$$J_i T_{ij} = T_{ij} J_j \quad \Leftrightarrow \quad J_i T_{ij} - T_{ij} J_j = 0.$$

The Sylvester equation has a unique zero solution when the eigenvalues of $J_i$ and $J_j$ are different. In the other case, observe that $\lambda I_i T_{ij} - T_{ij} \lambda I_j$ is in the kernel and can be removed,

$$(\lambda I_i + N_i) T_{ij} = T_{ij} (\lambda I_j + N_j) \quad \Leftrightarrow \quad N_i T_{ij} = T_{ij} N_j.$$

The matrices $N_i$ and $N_j$ in the remaining part of the equation can be interpreted as *up-shift* and *right-shift* operator, respectively. Thus the matrix $T_{ij}$ has to be *Toeplitz*. Since the shifts introduce zeros in the *last row* and *first column*, respectively, the lower part is filled with zeros and we have described our set. In this representation it is not easy to see *which* block matrices $T$ are regular. This fault is removed by the LR-type decomposition which we introduce now.

We consider block Gaussian elimination. A block elimination step like

$$\begin{pmatrix} I & 0 \\ -T_{i+1,i} T_{ii}^{-1} & I \end{pmatrix} \begin{pmatrix} T_{ii} & T_{i,i+1} \\ T_{i+1,i} & T_{i+1,i+1} \end{pmatrix} = \begin{pmatrix} T_{ii} & T_{i,i+1} \\ 0 & T_{i+1,i+1} - T_{i+1,i} T_{ii}^{-1} T_{i,i+1} \end{pmatrix}$$

is only possible when the diagonal block $T_{ii}$ is regular. Since the blocks are upper triangular Toeplitz, $T_{ii}$ is regular when the element on the diagonal is non-zero. When all other blocks to the same eigenvalue are of smaller size, the regularity of $T$ implies the regularity of $T_{ii}$. This is best seen with the aid of a little picture. Figure 1.1 plots a small example of a matrix $T$ of the class $\mathcal{T}$ for a Jordan matrix that has only *one* eigenvalue $\lambda$ and *four* Jordan blocks, *one large* sized, *two equally*

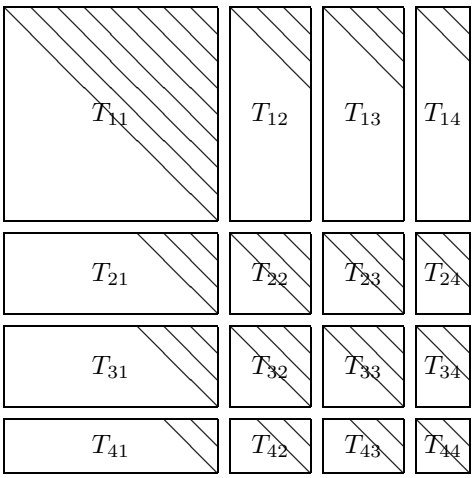


Figure 1.1: A typical member of the class $\mathcal{T}$

sized and *one small* sized. The matrix $T$ in the figure may be considered a typical example of one diagonal block occurring in a more general matrix $T$ where all other blocks in the row and column are zero. When looking at the first column, obviously $T_{11}$ has to be regular, since otherwise the matrix $T$ would contain a zero column. This, in turn, is impossible, since $T$ is *regular*. When looking at the first columns of the following block columns, we observe that we can eliminate the entries in the first row. When the next block would have again been larger in size, this would imply that also $T_{22}$ would be regular, and so forth.

Regular matrices can be used in a block elimination step. The rectangular matrices can be embedded into larger (singular) matrices and are no problem in the elimination steps. We only have to show that the *structure* is not destroyed, i.e., it remains to show that *upper triangular Toeplitz* matrices form a *group*. The sum of two upper triangular Toeplitz matrices is again an upper triangular Toeplitz matrix. We identify upper triangular Toeplitz matrices with there first row, i.e., we define a mapping

$$\mathbb{K}^{n\times n} \to \mathbb{K}^n, \qquad T = \begin{pmatrix} t_{11} & \cdots & t_{1n} \\ & \ddots & \vdots \\ & & t_{nn} \end{pmatrix} \mapsto \begin{pmatrix} t_{11} \\ \vdots \\ t_{1n} \end{pmatrix} \equiv \begin{pmatrix} \tau_0 \\ \vdots \\ \tau_{n-1} \end{pmatrix} = \tau.$$

The numbering scheme is chosen such that the matrix entries $t_{ij}$ correspond to the vector entries $\tau_{j-i}$. The matrix product $T_p = T_1 T_2$ of two upper triangular Toeplitz matrices $T_1$, $T_2$ can be written as

$$t_{ij}^{(p)} = \sum_{k=1}^{n} t_{ik}^{(1)} t_{kj}^{(2)} = \sum_{k=i}^{j} t_{ik}^{(1)} t_{kj}^{(2)} = \sum_{k=i}^{j} \tau_{k-i}^{(1)} \tau_{j-k}^{(2)} = \sum_{k=0}^{j-i} \tau_k^{(1)} \tau_{(j-i)-k}^{(2)} = \tau_{j-i}^{(p)}.$$

Thus, the product again is an upper triangular Toeplitz matrix. As mentioned, the inverse exists, provided $\tau_0$ is non-zero.

Now, it is easy to see that the inverse of an upper triangular matrix contains the inverses of arbitrary *diagonal blocks* in the *same position*. Since a Toeplitz matrix may be characterised by the property that *every* diagonal block of the *same size* has the *same elements*, obviously the inverse of an (invertible) upper triangular Toeplitz matrix is an upper triangular Toeplitz matrix. Thus these matrices form a group.

In our example in figure 1.1, there are two Jordan blocks of the *same size* to the *same eigenvalue*. The regularity of $T$ implies that the first column of the second and third block column are linearly independent. After elimination of the elements in the first row, we observe that the linear independence is equivalent to the assumption that the matrix $T_D$, formed out of the elements $\tau_0^{ij}$ in the diagonals of the matrices $T_{ij}$,

$$T_D = \begin{pmatrix} \tau_0^{(22)} & \tau_0^{(23)} \\ \tau_0^{(32)} & \tau_0^{(33)} \end{pmatrix},$$

has to be *regular*. When it is regular, it can be decomposed with application of GEPP. So, at least *one* of $\tau_0^{(22)}$ or $\tau_0^{(32)}$ must be non-zero. This implies that also at least one of $T_{22}$ or $T_{32}$ must be non-singular and can be moved by a block perturbation to take place in the diagonal. This naturally extends to higher dimensions, i.e., *several* equal sized Jordan blocks. In any case, we can block-eliminate the matrices in the lower triangular and succeed with the next submatrix in the same manner. This finishes the proof. □

When we take a Jordan decomposition, we can easily obtain what we will refer to as a *partial Jordan decomposition* by stripping off some columns and corresponding rows of the Jordan blocks and the corresponding columns of $V$. The resulting equation should have the form

$$AV_p = V_p J_p, \tag{1.2}$$

where $J_p \in \mathbb{K}^{m\times m}$ is a *smaller sized* Jordan matrix and $V_p \in \mathbb{K}^{n\times m}$ spans the corresponding invariant subspace. We may ask for conditions such that the converse also holds true. That is, given an equation of type (1.2), is it possible to construct a matrix $V$ such that equation (1.2) *is* a *corresponding* partial Jordan decomposition? The following example clarifies that this is *not* possible in every case.

**Example 1.20 (Non-Extendable Partial Decomposition)** The matrix $A$, given explicitly by

$$A = \begin{pmatrix} \lambda & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 1 \\ 0 & 0 & \lambda & 1 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 1 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda \end{pmatrix},$$

*has* a partial Jordan decomposition. Here, the matrix $V_p$ is defined by the first five standard unit vectors, the Jordan matrix $J_p$ is the leading five by five submatrix of $A$. This partial Jordan decomposition *can not* be extended to a complete Jordan decomposition without altering some columns of $V_p$. To see this, assume the contrary. Obviously, the complete Jordan decomposition has two Jordan blocks, one of size two, one of size four. Let the complete Jordan decomposition be given by

$$A \begin{pmatrix} V_p & w \end{pmatrix} = \begin{pmatrix} V_p & w \end{pmatrix} \begin{pmatrix} J_2 & \\ & J_4 \end{pmatrix}.$$

Evaluating the last column, to be more precise, the second entry, we can conclude that the last entry of $w$ must be *zero*. The fifth entry shows that the last entry must be *one*. This is impossible.

Nevertheless, there exist partial Jordan decompositions in *the space spanned by*

$V_p$. One such partial Jordan decomposition results when we set a *new* $V_p$ to

$$V_p \equiv [e_1, e_2, e_3, e_4 + e_1, e_5 + e_2] = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

This partial Jordan decomposition can be extended to a complete Jordan decomposition by adding the sixth standard unit vector $e_6$.

The example clearly reveals that partial Jordan decompositions exist that can not be extended without altering columns. The *second* part of the example raises new questions. Is it always possible to find another basis of $V_p$ such that an extendable partial Jordan decomposition results? When it is possible, how many columns have to be altered? The next example will shed some more light on these questions.

**Example 1.21 (Non-Extendable Partial Decomposition, II)** Let us focus once again on Example 1.20. We choose the partial Jordan decomposition that is defined by the third to fifth standard unit vectors.

Then a short computation shows that it is not possible to expand this decomposition to a complete Jordan decomposition without adding information about the first two standard unit vectors.

Next, we state a theorem on the *general* case. It will show up that the previous two examples are precisely the *interesting special* cases.

**Theorem 1.22** *Let $A \in \mathbb{K}^{n\times n}$. Let $J_p$, $V_p$, be a* right partial *Jordan decomposition*

$$AV_p = V_p J_p, \qquad J_p \in \mathbb{K}^{m\times m}, \quad V_p \in \mathbb{K}^{n\times m} \quad \textit{full rank}$$

*of $A$. Then this decomposition* can be expanded *to form a complete Jordan decomposition*

$$AV = VJ, \qquad J \in \mathbb{K}^{n\times n}, \quad V \in \mathbb{K}^{n\times n} \quad \textit{regular},$$

*such that the columns of $V_p$ form a subset of the columns of $V$ in case:*

*a) All eigenvalues of $J_p$ are distinct from the remaining eigenvalues of $A$.*

*b) All Jordan blocks of $J_p$ to eigenvalues that are also eigenvalues of $A$ are the* largest Jordan blocks *of $A$.*

*When* all *Jordan blocks of $A$ are contained in $J_p$, and all but the* maximal one *have already maximal size, a partial Jordan decomposition in the space spanned by $V_p$ can be found that can be extended. When the non-maximal one is not the largest Jordan block, a size-drop by* one *leads to the same conclusion.*

*In general, the extension to a complete Jordan decomposition is* impossible *without altering the space spanned by $V_p$, i.e., without adding information from the orthogonal complement.*

*An analogue holds true for a* left partial *Jordan decomposition.*

**Proof.** Obviously, b) implies a). Nevertheless, we will prove the statements in the order they occur in the theorem. Let $V_z$ denote a matrix whose columns form a basis of the orthogonal complement of $V_p$. The requirement that the pair $(J_p, V_p)$ is a right partial Jordan decomposition can be stated in the following form:

$$A \begin{pmatrix} V_p & V_z \end{pmatrix} = \begin{pmatrix} V_p & V_z \end{pmatrix} \begin{pmatrix} J_p & X \\ 0 & B \end{pmatrix}. \tag{1.3}$$

When the eigenvalues of $J_p$ and of $B$ are distinct, the Sylvester equation $J_pY - YB = X$ has a unique solution $Y$ and the transformation

$$(\, V_p \quad V_z^{\,\mathrm{new}} \,) = (\, V_p \quad V_z \,) \begin{pmatrix} I & -Y \\ 0 & I \end{pmatrix} = (\, V_p \quad V_z - V_pY \,) .$$

results in the new relation

$$A\,(\, V_p \quad V_z^{\,\mathrm{new}} \,) = (\, V_p \quad V_z^{\,\mathrm{new}} \,) \begin{pmatrix} J_p & 0 \\ 0 & B \end{pmatrix} .$$

This proves statement a). From now on, we assume w.l.o.g. that $A$ has only *one* eigenvalue. Let the Jordan decomposition of $B$ be given by $J_z = W^{-1}BW$. Then we can go on to transform equation (1.3) via the change of basis

$$(\, V_p \quad V_z^{\,\mathrm{new}} \,) = (\, V_p \quad V_z \,) \begin{pmatrix} I & 0 \\ 0 & W \end{pmatrix} = (\, V_p \quad V_zW \,) , \qquad X^{\,\mathrm{new}} = XW$$

to the simpler form

$$A\,(\, V_p \quad V_z^{\,\mathrm{new}} \,) = (\, V_p \quad V_z^{\,\mathrm{new}} \,) \begin{pmatrix} J_p & X^{\,\mathrm{new}} \\ 0 & J_z \end{pmatrix} .$$

In other words, we may assume that the trailing block matrix $B$ in equation (1.3) already *is* in Jordan normal form. The proof of b) is based on adding new columns to the partial Jordan decomposition without altering previously computed columns too much.

For sake of simplicity, we assume that $J = J_p$ is only one single Jordan block and set $V = V_p$, $\tilde{v} = V_z(:, 1)$ and $\alpha = X(:, 1)$. With this notation,

$$A\,(\, V \quad \tilde{v} \,) = (\, V \quad \tilde{v} \,) \begin{pmatrix} J & \alpha \\ 0 & \lambda \end{pmatrix} \tag{1.4}$$

holds true. When $\alpha$ is identically zero, we have found a second eigenvector to the eigenvalue $\lambda$. This increases the number of Jordan blocks to the same eigenvalue by one and we split the blocks $X_{ij}$ to deal with this new situation. Otherwise, there is a unique largest integer $j$ such that $\alpha_j \neq 0$. When we choose $V^{\,\mathrm{new}} = VT$, where $T$ is the *regular* upper triangular Toeplitz matrix

$$T = \begin{pmatrix} \alpha_j & \cdots & \alpha_1 & 0 & \cdots & 0 \\ & \alpha_j & \cdots & \alpha_1 & \ddots & \vdots \\ & & \ddots & \ddots & \ddots & 0 \\ & & & \alpha_j & \cdots & \alpha_1 \\ & & & & \ddots & \vdots \\ & & & & & \alpha_j \end{pmatrix}$$

defined by the entries of the vector $\alpha$, we obtain the new form

$$A\,(\, V^{\,\mathrm{new}} \quad \tilde{v} \,) = (\, V^{\,\mathrm{new}} \quad \tilde{v} \,) \begin{pmatrix} J & e_j \\ 0 & \lambda \end{pmatrix}$$

of equation (1.4). When $j$ equals the *last* entry of $\alpha$ we have expanded the Jordan block by one. When $j$ is not equal to the last entry, we can form $\tilde{v}^{\,\mathrm{new}} = \tilde{v} - v_{j+1}^{\mathrm{new}}$, which is *non-zero* because the columns of $V$ and $\tilde{v}$ are linearly independent, and an *eigenvector*, since

$$A\tilde{v}^{\,\mathrm{new}} = A(\tilde{v} - v_{j+1}^{\mathrm{new}}) = \lambda(\tilde{v} - v_{j+1}^{\mathrm{new}}) + v_j^{\mathrm{new}} - v_j^{\mathrm{new}} = \lambda\tilde{v}^{\,\mathrm{new}}.$$

In the first case, we expand the Toeplitz matrix $T$ by one column and row and invert it to recover the original $V$, in the second case we simply invert $T$ to recover the original $V$. This proves that we can transform equation (1.4) to take the form

$$A\begin{pmatrix} V & \tilde{v}^{\,\mathrm{new}} \end{pmatrix} = \begin{pmatrix} V & \tilde{v}^{\,\mathrm{new}} \end{pmatrix} \begin{pmatrix} J & \delta_{jk} e_k \\ 0 & \lambda \end{pmatrix},$$

where $e_k$ is the $k$th unit vector of length $k$, where $k$ is the last entry of $\alpha$, $\delta_{jk}$ denotes Kronecker delta. In a similar manner we can show that in case of *several* Jordan blocks in $J_p$, lets say $l$ Jordan blocks, we can transform the equation

$$A\begin{pmatrix} V & \tilde{v} \end{pmatrix} = \begin{pmatrix} V & \tilde{v} \end{pmatrix} \begin{pmatrix} J_p & \alpha \\ 0 & \lambda \end{pmatrix} \tag{1.5}$$

by setting

$$V^{\,\mathrm{new}} = V \begin{pmatrix} T_1 & & \\ & \ddots & \\ & & T_l \end{pmatrix}$$

to take the form

$$A\begin{pmatrix} V^{\,\mathrm{new}} & \tilde{v} \end{pmatrix} = \begin{pmatrix} V^{\,\mathrm{new}} & \tilde{v} \end{pmatrix} \begin{pmatrix} J_p & c \\ 0 & \lambda \end{pmatrix}, \tag{1.6}$$

where the vector $c$ has *at most one non-zero component* in the set of entries corresponding to the $l$ distinct Jordan blocks. As before, we assume that they are scaled to equal one. The ones in the *last* entries of the partitioned vector can be removed by subtracting a portion of $V^{\,\mathrm{new}}$ from $\tilde{v}$. We assume w.l.o.g. that these entries are already removed. We remark that this form is the form that was used as Example 1.20.

We assume that the Jordan blocks appear sorted by *ascending* size. This ensures that the transformation of the class $\mathcal{T}$ defined by

$$V^{\,\mathrm{mixed}} = V^{\,\mathrm{new}} \begin{pmatrix} I^{(1)} & & & E^{(1,l)} \\ & \ddots & & \vdots \\ & & I^{(l-1)} & E^{(l-l,l)} \\ & & & I^{(l)} \end{pmatrix}, \tag{1.7}$$

where the matrices $E^{(i,l)}$ are 'shifted' identity matrices,

$$E^{(i,l)} = \begin{pmatrix} 0 & I^{(i)} \end{pmatrix} \in \mathbb{K}^{k_i \times k_l},$$

where $k_i$ is the size of the $i$th Jordan block, maps equation (1.6) to the form

$$A\begin{pmatrix} V^{\,\mathrm{mixed}} & \tilde{v} \end{pmatrix} = \begin{pmatrix} V^{\,\mathrm{mixed}} & \tilde{v} \end{pmatrix} \begin{pmatrix} J_p & e_k \\ 0 & \lambda \end{pmatrix}.$$

This, in turn, implies that the largest Jordan block is extended by one. When we are in situation b), this is impossible, thus in this case the vector $c$ must be identical zero. Induction proves that we do not need *any* transformations of the form (1.7), i.e., we do not have to mix up invariant subspaces of Jordan blocks.

When we have only *one* Jordan block that has to be extended, we know that we only add information from the space we already have constructed. This, of course, is always the case when we only have to extend the maximal Jordan block. If the block to be extended is not the maximal one, but we have to extend only *once*, we have not gained any information on the orthogonal complement and thus mix only *known* information.

In general, we might have to mix information on the orthogonal complement gained in such a step with the information in $V_p$ when we construct the next column. This finishes the proof. □

Theorem 1.19 might be of use when one wants to investigate *optimal* scalings of the eigenmatrix in the *non-diagonalisable* case. In the diagonalisable case it is well-known that a scaling such that left and right eigenvectors have unit length is nearly optimal. Theorem 1.22 shows that the a partial Jordan decomposition, i.e., a restriction to a subspace, might result in a *different* condition number of the columns of the eigenmatrix.

Theorem 1.22 also shows that the Jordan normal form may be constructed iteratively by adding columns like in the proof of the theorem. The proof clearly reveals that the Jordan normal form is by no means continuous. When we use only unitary (orthogonal) similarity transformations, this fault is removed. The resulting *Schur decomposition* is more stable than the Jordan decomposition and to be preferred in finite precision.

### Schur Decomposition

When we restrict the class of similarity transformations to unitary matrices, we still can find a similar *upper triangular* matrix. This matrix is known as a *Schur normal form*.

**Theorem 1.23 (The Schur Normal Form)** *Any matrix $A \in \mathbb{K}^{n\times n}$ is* similar *to an* upper triangular matrix*,*

$$A = URU^{-1} = URU^H.$$

*Here, $U$ is unitary. The diagonal elements of $R$ are the eigenvalues of $A$ and may appear in* any given order *along the diagonal. A* change *in the ordering usually changes the strictly upper triangular part of $R$ and the columns of $U$, the so-called* Schur vectors*. Every such decomposition is known as a* Schur decomposition *of $A$.*

The Jordan normal form and the Schur normal from are closely related to each other. In important special cases these normal forms coincide. One such class of matrices is the class of *normal matrices*. Normal matrices are matrices that fulfil $AA^H = A^HA$, i.e., they commute with their conjugate transpose (this includes selfadjoint and Hermitian matrices). The theorem can be used to prove that *normal matrices* can be decomposed as

$$A = V\Lambda V^H$$

with $\Lambda$ diagonal and $V^HV = I$, that is, $V$ is unitary. This follows by inserting the Schur decomposition and using the fact that $A$ and $A^H$ commute.

The most prominent members of the class of normal matrices are the Hermitian matrices, defined by $A \in \mathbb{K}^{n\times n}$, $A = A^H$, and the real symmetric matrices, defined by $A \in \mathbb{R}^{n\times n}$, $A = A^T$. These matrices are related to quadratic forms and have only *real* eigenvalues. This is the basis for the following theorem, taken from [HJ85] (page 179, Theorem 4.2.11):

**Theorem 1.24 (Courant-Fischer)** *Let $A \in \mathbb{K}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, and let $k \in \underline{n}$. Let $\mathcal{S}_j \subset \mathbb{C}^n$ denote a subspace of $\mathbb{C}^n$ of dimension $j$. Then*

$$\begin{aligned} \lambda_k &= \min_{\mathcal{S}_k} \quad \max_{x\in\mathcal{S}_k} \quad \frac{x^HAx}{x^Hx} \\ &= \max_{\mathcal{S}_{n-k+1}} \quad \min_{x\in\mathcal{S}_{n-k+1}} \quad \frac{x^HAx}{x^Hx} \end{aligned} \tag{1.8}$$

*The cases $k = 1$ and $k = n$ are also known as the* Rayleigh-Ritz theorem.

As already mentioned, the Schur form is more stable than the Jordan normal form and should be computed when we are interested *only* in the eigen*values* and the corresponding *invariant subspaces* and not explicitely in the eigen*vectors*. The Jordan normal form has advantages when we wish to express polynomials, e.g. powers of $A$ in an easy to follow way.

The Schur and Jordan normal forms are *shift-invariant*, because of the similarity transformation. When we drop the property of shift-invariance, i.e., the constraint that the matrix is obtained by the application of a similarity transformation, we can compute *unitary* bases for the left and right space *such* that the matrix is mapped to a *diagonal* form. This is known as the *singular value decomposition* of $A$.

### Singular Value Decomposition

When we force unitary bases of the left and right space *and* a diagonal matrix in the middle, we obtain the *singular value decomposition*.

**Theorem 1.25 (Singular value decomposition)** *Every $A \in \mathbb{K}^{n\times m}$ can be decomposed into*

$$A = U\Sigma V^H,$$

*where $U$ and $V$ are unitary and $\Sigma$ is diagonal. The diagonal elements of $\Sigma$ are non-negative and often assumed sorted according to magnitude,*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min_{n,m}} \geq 0.$$

*The values $\sigma_j$ in the diagonal of $\Sigma$ are the so-called* singular values.

The singular vales contain a lot of information about the matrix. The smallest singular value, denoted by $\sigma_{\min}$, is the distance to singularity (in $\|\cdot\|_2$). More general, the distance to the space of matrices with the same dimensions and rank less $k$ is given by $\sigma_k$.

When $A \in \mathbb{K}^{n\times n}$ is symmetric positive definite (SPD) or Hermitian positive definite (HPD), i.e., when

$$x^H Ax > 0 \quad \forall\, x \neq 0 \in \mathbb{K}^n,$$

the SVD, Schur and Jordan form coincide. When $A$ is symmetric or Hermitian indefinite, we obtain the SVD from the Schur/Jordan normal form by moving minus signs from the diagonal to the columns of one side.

In any case, the singular value decomposition is closely connected to the eigendecomposition of selfadjoint (and thus normal) matrices, since we have that

$$\begin{array}{rclcl} A^HA & = & V\Sigma U^H U\Sigma V^H & = & V\Sigma^H\Sigma V^H, \\ AA^H & = & U\Sigma V^H V\Sigma U^H & = & U\Sigma^H\Sigma U^H. \end{array}$$

The squared singular values (and some additional zeros) are the eigenvalues of the Hermitian positive semi-definite matrices $A^HA$ and $AA^H$, the left and right singular vectors are the eigenvectors. This interpretation is based on the *normal equations* and was first used by Beltrami. Another connection that was noted for the first time by Jordan is the so-called *Jordan-Wielandt* form (here only for square $A$)

$$\begin{pmatrix} 0 & A \\ A^H & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} U & U \\ V & -V \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} U & U \\ V & -V \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{pmatrix}.$$

In this way, the SVD can be interpreted as the generalisation of the spectral decomposition to general *rectangular* matrices.

Thus, in view of Theorem 1.24, it appears natural to interpret the singular values as maximal, respectively, minimal amplification factors on subspaces.

**Theorem 1.26** *Let $A \in \mathbb{K}^{n\times m}$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p$, where $p = \min\{n, m\}$. Let $k \in \underline{p}$. Let $\mathcal{S}_j \subset \mathbb{C}^m$ denote a subspace of $\mathbb{C}^m$ of dimension $j$. Then the singular values can be characterised by*

$$\begin{aligned} \sigma_k &= \min_{\mathcal{S}_k} \quad \max_{x\in\mathcal{S}_k} \quad \frac{\|Ax\|_2}{\|x\|_2} \\ &= \max_{\mathcal{S}_{m-k+1}} \quad \min_{x\in\mathcal{S}_{m-k+1}} \quad \frac{\|Ax\|_2}{\|x\|_2} \end{aligned} \tag{1.9}$$

## 1.6.2 Necessary Definitions

In this section we introduce the working horses of perturbation theory for the algebraic eigenproblem. The main tools are *spectral projectors*, the *separation of matrices* and the *(reduced) resolvent*. We start with the definition of the spectral projectors. They will be of use in the investigation of the stability of the eigenproblem. The investigation is usually based on a splitting of the entire spectrum into two distinct parts. This parts may consist of one single eigenvalue or a cluster of close eigenvalues, and the remaining part. Usually both sets have, in some sense, to be well separated in order to apply perturbation theory. This is where the separation of matrices comes into play. Our representation is based to some extent on the treatise of Chatelin (cf. [Cha93]) and on the joint paper of Bai, Demmel and McKenney (cf. [BDM91]).

### Spectral Projectors

Based on the Jordan normal form we define the *spectral projectors* onto the subspace associated with an eigenvalue $\lambda$. We assume w.l.o.g. that the eigenvalue $\lambda$ we are interested in corresponds to the *leading* blocks of the Jordan matrix.

**Definition 1.27 (Spectral Projector I)** The spectral projectors or *Frobenius covariants* are given by the matrices

$$P_\lambda \equiv P \equiv V \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \hat{V}^H.$$

More general, we can think of the spectral projector as the matrix $A$ in Jordan normal form where we replace the Jordan matrix by a diagonal matrix that has ones in places of the eigenvalue $\lambda$ and zeros elsewhere. Naturally, the following holds true:

**Lemma 1.28** *Let $P_\lambda$ denote the spectral projector to eigenvalue $\lambda$. Then*

$$\sum_{\lambda=\lambda(A)} P_\lambda = I, \qquad P_\lambda^k = P_\lambda \quad \forall\ k, \qquad P_{\lambda_i} P_{\lambda_j} = 0 \quad \forall\ \lambda_i \neq \lambda_j.$$

We additionally denote by $V_\lambda$ the subset of the columns of $V$ that spans the right invariant subspace to $\lambda$. The subset of the rows of the inverse of $V$, $\hat{V}^H = V^{-1}$ that spans the adjoint basis of the left invariant subspace to $\lambda$ is denoted by $\hat{V}_\lambda^H$. These notions are extended to the case of a cluster.

The spectral projectors can alternatively be defined using the Schur form. The Schur form is advantageous when we are interested in actually *computing* the spectral projectors.

**Definition 1.29 (Spectral Projector II)** Suppose $A$ has the (partial) Schur form

$$A = U \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} U^H.$$

Here, the desired eigenvalue (or the cluster under investigation) is assumed to be contained in the block $A_{11}$. Let $R$ be the solution of the *Sylvester equation*

$$A_{11}R - RA_{22} = A_{12}.$$

This equation has a *unique* solution when the spectra of $A_{11}$ and $A_{22}$ are *disjoint*. The right and left invariant subspace associated with $A_{11}$ are given by

$$U \begin{pmatrix} I \\ 0 \end{pmatrix}, \qquad (\, I \quad R \,) U^H.$$

The spectral projector is given by the outer product of left and right invariant subspace,

$$P = U \begin{pmatrix} I \\ 0 \end{pmatrix} (\, I \quad R \,) U^H = U \begin{pmatrix} I & R \\ 0 & 0 \end{pmatrix} U^H.$$

**Lemma 1.30** *Definition 1.27 and Definition 1.29 are equivalent.*

**Proof.** As most textbooks use only *one* of these definitions we give the short proof for their equivalence. We can assume that the Jordan normal form of the partial block Schur form is given by

$$\begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} X_1 & Z \\ 0 & X_2 \end{pmatrix} = \begin{pmatrix} X_1 & Z \\ 0 & X_2 \end{pmatrix} \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix}.$$

This implies that the Jordan normal forms of the diagonal blocks are given by

$$A_{11}X_1 = X_1J_1, \qquad A_{22}X_2 = X_2J_2, \quad \Rightarrow \quad X_2^{-1}A_{22} = J_2X_2^{-1}.$$

This knowledge can be used to derive an explicit expression for the matrix $Z$ in terms of the matrix $R$ previously defined,

$$A_{11}Z + A_{12}X_2 = ZJ_2 \quad \Rightarrow \quad A_{11}ZX_2^{-1} + A_{12} = ZJ_2X_2^{-1} = ZX_2^{-1}A_{22}.$$

By comparison with the definition of $R$ we observe that $R = -ZX_2^{-1}$. The inverse of the block eigenvector matrix is given by

$$\begin{pmatrix} X_1 & Z \\ 0 & X_2 \end{pmatrix}^{-1} = \begin{pmatrix} X_1^{-1} & -X_1^{-1}ZX_2^{-1} \\ 0 & X_2^{-1} \end{pmatrix} = \begin{pmatrix} X_1^{-1} & X_1^{-1}R \\ 0 & X_2^{-1} \end{pmatrix}.$$

Thus, $A$ has the Jordan normal form

$$A = U \begin{pmatrix} X_1 & Z \\ 0 & X_2 \end{pmatrix} \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix} \begin{pmatrix} X_1^{-1} & X_1^{-1}R \\ 0 & X_2^{-1} \end{pmatrix} U^H$$

and the spectral projector is given by

$$\begin{aligned} P &= U \begin{pmatrix} X_1 & Z \\ 0 & X_2 \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X_1^{-1} & X_1^{-1}R \\ 0 & X_2^{-1} \end{pmatrix} U^H \\ &= U \begin{pmatrix} X_1 \\ 0 \end{pmatrix} (\, X_1^{-1} \quad X_1^{-1}R \,) U^H. \end{aligned}$$

This finishes the proof of the equivalence. □

Later on, the norm of the spectral projector is used as a measure of the condition of an eigenvalue (or cluster). In case of the 2-norm we can express this norm by means of *angles* between the invariant subspaces and their orthogonal complements.

Angles between subspaces are defined by choosing orthogonal bases $U$ and $V$ and computing the singular values of the product $U^H V$.

We denote the *right* invariant subspace we are interested in by $\mathcal{R}$, its complement by $\mathcal{R}_c$, the *left* invariant subspace by $\mathcal{L}$ and its complement by $\mathcal{L}_c$. Then, it is easy to show that the Euclidian norm of the spectral projector may be expressed as

$$\begin{aligned} \|P\|_2 &= \csc(\theta_{\min}(\mathcal{R}, \mathcal{R}_c)) &&= \csc(\theta_{\min}(\mathcal{L}, \mathcal{L}_c)) \\ &= \sec(\theta_{\max}(\mathcal{R}, \mathcal{L})) &&= \sec(\theta_{\max}(\mathcal{R}_c, \mathcal{L}_c)), \end{aligned}$$

where we have defined the angles $\theta$ using the singular values $\sigma$ previously mentioned:

$$\theta_{\max} \equiv \operatorname{acos}(\sigma_{\min}), \qquad \theta_{\min} \equiv \operatorname{acos}(\sigma_{\max}).$$

The interpretation using angles between subspaces can better be memorised.

Apart from the spectral projectors, we additionally define the matrix

$$D = (A - \lambda I)P = V \begin{pmatrix} J(0) & 0 \\ 0 & 0 \end{pmatrix} \hat{V}^H.$$

This matrix may have several Jordan blocks to eigenvalue zero, one for every Jordan block to eigenvalue $\lambda$ of the original matrix $A$. The matrix is nilpotent, where $\ell$ is the smallest integer such that $D^\ell = 0$. This is the *ascent* of the eigenvalue $\lambda$.

### The Separation of Matrices

In the definition of the spectral projector based on the Schur form we already observed the dependency on the solution of a Sylvester equation. Obviously the norm of the spectral projector is large, when this solution is a large-normed solution. For this reason we introduce the separation of matrices:

**Definition 1.31 (The Separation of Matrices)** Let matrices $A \in \mathbb{K}^{n\times n}$, $B \in \mathbb{K}^{m\times m}$ and a norm $\|\cdot\|$ be given. Then the separation of the matrices $A$ and $B$ in the norm $\|\cdot\|$ is defined to be

$$\operatorname{sep}(A, B) \equiv \min_{X \in \mathbb{K}^{n\times m}} \frac{\|AX - XB\|}{\|X\|}.$$

In case of the Frobenius norm, $\|\cdot\| = \|\cdot\|_F$, the separation can be expressed in terms of singular values of a larger matrix:

$$\operatorname{sep}_F(A, B) = \sigma_{\min}(I \otimes A - B^T \otimes I).$$

Following Chatelin (cf. [Cha93], page 77), we express the solution $X$ of the Sylvester equation $AX - XB = R$ with right-hand side $R$ by

$$X = (A, B)^{-1} R.$$

Using this terminology, the separation might be expressed as $\operatorname{sep}(A, B) = \|(A, B)^{-1}\|$. When $m = 1$ or $n = 1$, this expression collapses to the so-called *resolvent*. The resolvent has some very interesting features and deserves an extra section.

### The Resolvent

The *resolvent* of $A$ is defined for any $z \in \mathbb{C}$, $z \notin \Lambda(A)$ by

$$R(z) = (zI - A)^{-1}.$$

The matrix $R(z)$ is also known as *spectral transformation*. When $z$ comes close to an eigenvalue, the norm of the resolvent grows. We will show later on that the norm of the resolvent is closely connected to the backward error of an approximate eigenvalue.

The resolvent is not defined when we choose $z = \lambda$ as an eigenvalue. In this case, we still can compute the *Drazin inverse*

$$S = S_\lambda \equiv (\lambda I - A)^D$$

The resulting matrix is known as the *reduced resolvent*. When we refrain from the usage of the Drazin inverse, we can derive the reduced resolvent with the aid of the spectral decomposition. It is easy to show that

$$S = S_\lambda = V \begin{pmatrix} 0 & 0 \\ 0 & (\lambda I - J_2)^{-1} \end{pmatrix} \hat{V}^H \equiv V_2 (\lambda I - J_2)^{-1} \hat{V}_2^H,$$

where $J_2$ is the Jordan matrix of the Jordan blocks to the other eigenvalues. The reduced resolvent has the following properties:

**Lemma 1.32** *Let $S$ be the* reduced resolvent *as defined above. Let $\|\cdot\|$ denote an operator norm. Then the norm of $S$ can be bounded as follows:*

$$\operatorname{dist}(\lambda, \Lambda \setminus \{\lambda\})^{-1} = \rho(S) \leq \|S\| \leq \|V_2\| \|\hat{V}_2\| \|(\lambda I - J_2)^{-1}\|. \tag{1.10}$$

*Furthermore,* equality *holds true when the matrix $A$ is normal and $\|\cdot\| = \|\cdot\|_2$.*

This lemma proves that the norm of $S$ can not be large, when the eigenvalue $\lambda$ is well separated form the remaining part of the spectrum and the eigenvector matrix is not too ill-conditioned.

The resolvent is a holomorphic function in $\operatorname{res}(A)$, where $\operatorname{res}(A)$ is defined to be $\mathbb{C} \setminus \Lambda(A)$. This enables us to expand the resolvent into different series.

**Theorem 1.33 (Expansions of the Resolvent)** *The resolvent has a* Taylor expansion *in the circle $|z - z_0| \leq \rho(R(z_0))^{-1}$,*

$$R(z) = \sum_{k=0}^{\infty} (z_0 - z)^k \left[R(z_0)\right]^{k+1}. \tag{1.11}$$

*The resolvent has a* Taylor expansion at $\infty$ *for all $|z| > \rho(A)$,*

$$R(z) = \frac{1}{z} \sum_{k=0}^{\infty} \left[\frac{A}{z}\right]^k. \tag{1.12}$$

*We are putting these expansions together to obtain a Laurent series. In the* neighbourhood *of an* eigenvalue $\lambda$ *the resolvent can be expanded in a* Laurent series at $\lambda$*,*

$$R(z) = \frac{P}{z - \lambda} + \sum_{k=1}^{\ell - 1} \frac{D^k}{(z - \lambda)^{k+1}} + \sum_{k=0}^{\infty} (\lambda - z)^k S^{k+1}. \tag{1.13}$$

*As a by-product we obtain that for* all $z$ in *res(A),*

$$R(z) = \sum_{\lambda} \left[ \frac{P_\lambda}{z - \lambda} + \sum_{k=1}^{\ell_\lambda - 1} \frac{D_\lambda^k}{(z - \lambda)^{k+1}} \right]. \tag{1.14}$$

**Proof.** Suppose that $z, z_0$ are not contained in the spectrum of $A$. Then the identity $(z_0 - z)I = (z_0 I - A) - (zI - A)$ proves the so-called *first resolvent equation*

$$R(z) - R(z_0) = (z_0 - z)R(z_0)R(z).$$

By algebraic transformations we arrive at the formal expression

$$R(z) = R(z_0)\left[I - (z_0 - z)R(z_0)\right]^{-1}.$$

The use of the (formal) inverse on the right-hand side is justified when $|z - z_0| \leq \rho(R(z_0))^{-1}$. This enables us to insert a Neumann series for the matrix inverse, resulting in

$$R(z) = R(z_0)\sum_{k=0}^{\infty}\left[(z_0 - z)R(z_0)\right]^k,$$

which is just the Taylor expansion given as equation (1.11). The second expansion follows by the observation that for any $|z| > \rho(A)$ the series

$$\tilde{R}(z) = \frac{1}{z}\sum_{k=0}^{\infty}\left[\frac{A}{z}\right]^k$$

is convergent, and, furthermore, $\tilde{R}(z)(zI - A) = I$, i.e., $\tilde{R}(z) = R(z)$. The other two expansions follow by a simple re-writing of the resolvent. First, we write the resolvent as $R(z) = R(z)P + R(z)(I - P)$. The term $R(z)P$ can be written as

$$R(z)P = V\begin{pmatrix}(zI - J_1)^{-1} & 0\\ 0 & 0\end{pmatrix}\hat{V}^H.$$

We substitute $\tilde{z} = z - \lambda$. Then the projected resolvent becomes

$$R(z)P = V\begin{pmatrix}\left[(z-\lambda)I - (J_1 - \lambda I)\right]^{-1} & 0\\ 0 & 0\end{pmatrix}\hat{V}^H = (\tilde{z}P - D)^{\dagger}.$$

When $|z - \lambda| = |\tilde{z}| > \rho(D) = 0$, i.e., when $z \neq \lambda$, we can apply the second expansion and obtain the expression

$$R(z)P = \frac{1}{z-\lambda}\sum_{k=0}^{\infty}\left[\frac{D}{z-\lambda}\right]^k = \frac{P}{z-\lambda} + \sum_{k=1}^{\ell-1}\frac{D^k}{(z-\lambda)^{k+1}}. \tag{1.15}$$

The second term $R(z)(I - P)$ can be expressed in terms of a perturbed reduced resolvent,

$$R(z)(I-P) = V\begin{pmatrix}0 & 0\\ 0 & (zI - J_2)^{-1}\end{pmatrix}\hat{V}^H \quad \Rightarrow \quad \lim_{z\to\lambda}\left(R(z)(I-P)\right) = S_\lambda.$$

This shows that when $|z - \lambda| \leq \rho(S_\lambda)^{-1}$, this term can be handled by the first expansion,

$$R(z)(I-P) = \sum_{k=0}^{\infty}(\lambda - z)^k\left[S_\lambda\right]^{k+1}.$$

The last expansion follows simply by writing the resolvent as a sum of projections onto the different eigenspaces, $R(z) = \sum R(z)P_\lambda$, and plugging in the result stated in equation (1.15). This finishes the proof. □

All the results and proofs may also be found in the textbook of Chatelin (cf. [Cha93]). Note that we have defined the resolvent in a slightly different way, thus

some signs differ. The first expansion is Proposition 2.2.2, page 64. Equation (1.13) is stated in Theorem 2.2.10, page 70 and equation (1.14) is stated in Corollary 2.2.12, page 71. In Chapter 2 we *algebraically* prove the expansion stated in equation (1.14).

Depending on the application we have in mind, different expansions can be used in different contexts. These expansions can be used to investigate the limiting behaviour of matrix powers, the behaviour of operators when we are close to an eigenvalue, or the dependence of the sensitivity of one eigenvalue on the other eigenvalues, i.e., the cross-sensitivity of the whole spectrum.

The resolvent can be used to express the spectral projectors:

$$P = \frac{1}{2\pi i} \int_\Gamma R(z)\, dz.$$

Here, $\Gamma$ is any closed Jordan curve enclosing the eigenvalue of interest that does not contain any other points of the spectrum. This integral is known as a *Taylor-Dunford* integral. With this notation the last result can be interpreted as the residue theorem applied to the meromorphic function $\mathrm{res}(z)$.

The definition of the spectral projector and the reduced resolvent is based on knowledge on the *right and left* invariant subspaces. We can define a *partial inverse* that presupposes only knowledge about the *right* invariant subspace $V_p$. We only need an adjoint basis of $V_p$, denoted by $V_a$. Then the partial inverse with respect to $V_a$ is defined by

$$\Sigma = V_p(V_a^H A V_p - \lambda I)^{-1} V_a^H.$$

This is of importance when one wishes to *explicitely* compute condition numbers of eigenspaces and only a basis for the right invariant subspace is at hand (cf. [Cha93], page 76).

Most important is the special case of an orthonormal basis. Let $(Q_p, Q_a)$ be an orthonormal basis of $\mathbb{K}^{n\times n}$, such that $Q_p$ is an orthonormal basis of $V_p$ and $Q_a$ is an orthonormal basis of the orthogonal complement $V_p^\perp$. Then, a superscript $^\perp$ is used to denote the partial inverse

$$\Sigma^\perp \equiv Q_a(Q_a^H A Q_a - \lambda I)^{-1} Q_a^H.$$

This partial inverse occurs in the bound of the variation of an eigenspace.

The notation established thus far allows only a distinction into one eigenvalue on the one hand and the remaining part of the spectrum on the other. This fault is removed by a block wise approach. Chatelin defines a *block reduced resolvent* and a *partial block inverse* (cf. [Cha93], pages 76–79). These notions are of importance in case we are interested in the perturbation of a cluster of eigenvalues.

### 1.6.3 Perturbation Theory

Perturbation becomes rather complicated, since the problem is non-linear and we can define several *residuals*, depending on the quantities we want to turn into exact solutions of a perturbed problem. Suppose that the unperturbed and the corresponding perturbed system are given by

$$Av = v\lambda \qquad \text{and} \qquad \tilde{A}\tilde{v} = \tilde{v}\tilde{\lambda}.$$

We consider the case that we are given $\tilde{\lambda}$ as an approximate eigenvalue and $\tilde{v}$ as corresponding approximate eigenvector to $A$. We define in analogy to the linear system case the residual $r = \tilde{v}\tilde{\lambda} - A\tilde{v}$.

### Backward Errors

Expressions for the backward errors can easily be derived. The equation $Av = v\lambda$ is nothing but a linear system of equations with a special right-hand side. Thus, we can apply the backward error theory of linear systems of equations also in the eigenproblem case. In strict analogy to the linear system of equations case the following holds true:

**Lemma 1.34** *Let $A \in \mathbb{K}^{n\times n}$. Denote by $\tilde{\lambda}$ an approximate eigenvalue and by $\tilde{v}$ an approximate eigenvector.*

*Then the backward error of an* approximate eigenpair $(\tilde{\lambda}, \tilde{v})$ *is given by*

$$\begin{aligned} \eta(\tilde{\lambda}, \tilde{v}) &\equiv \min\{\epsilon : (A + \Delta A)\tilde{v} = \tilde{v}\tilde{\lambda},\ \|\Delta A\| \leq \epsilon\alpha\} \\ &= \frac{\|A\tilde{v} - \tilde{v}\tilde{\lambda}\|}{\alpha\|\tilde{v}\|} = \frac{\|r\|}{\alpha\|\tilde{v}\|}. \end{aligned} \tag{1.16}$$

*The backward error of an* approximate eigenvector $\tilde{v}$ *is given by*

$$\begin{aligned} \eta(\tilde{v}) &\equiv \min\{\epsilon : (A + \Delta A)\tilde{v} = \tilde{v}\lambda,\ \|\Delta A\| \leq \epsilon\alpha,\ \exists\lambda \in \mathbb{K}\} \\ &= \eta(\mathrm{R}(\tilde{v}; A), \tilde{v}). \end{aligned} \tag{1.17}$$

*Here,* $\mathrm{R}(\tilde{v}; A)$ *denotes the* Rayleigh quotient *of the vector $\tilde{v}$, defined by*

$$\mathrm{R}(\tilde{v}) \equiv \mathrm{R}(\tilde{v}; A) \equiv \frac{\tilde{v}^H A\tilde{v}}{\tilde{v}^H\tilde{v}}.$$

*The backward error of an* approximate eigenvalue $\tilde{\lambda}$ *is given by*

$$\begin{aligned} \eta(\tilde{\lambda}) &\equiv \min\{\epsilon : (A + \Delta A)v = v\tilde{\lambda},\ \|\Delta A\| \leq \epsilon,\ \exists v \neq 0 \in \mathbb{K}^n\} \\ &= \|(\tilde{\lambda}I - A)^{-1}\|^{-1} = \|R(\tilde{\lambda})\|^{-1}. \end{aligned} \tag{1.18}$$

*All expressions involving approximate eigen*vectors *are easy to compute, the backward error of an approximate eigen*value *involves the inversion of a shifted matrix. Moreover, when the approximation is fairly close, this shifted matrix is close to a singular matrix.*

**Proof.** The results may be found in the textbook by Chaitin-Chatelin and Frayssé (cf. [CCF96]). Equation (1.16) follows upon application of the backward error result of Lemma 1.3 to the linear system $Av = v\lambda$. Here, we have set the perturbation in the right-hand side equal to zero. The result (1.17) is obvious, since $\eta(\tilde{v}) = \min_{\lambda\in\mathbb{K}} \eta(\lambda, \tilde{v})$. This minimisation,

$$\mathrm{R} = \arg\min_{\lambda\in\mathbb{K}} \|A\tilde{v} - \tilde{v}\lambda\|,$$

is solved by the *Rayleigh quotient* R. The last equation (1.18) holds true, when $\tilde{\lambda}$ is an eigenvalue of $A$. Otherwise, we use the property that

$$\eta(\tilde{\lambda}) = \min_{v\in\mathbb{K}^n} \eta(\tilde{\lambda}, v).$$

Upon inversion of $A - \tilde{\lambda}I$, the result (1.18) follows by definition of the operator norm. □

We note that when we measure the backward error in the 2-norm, the singular values of shifted $A$ come into play, because $\|R(\tilde{\lambda})^{-1}\|_2^{-1} = \sigma_{\min}(\tilde{\lambda}I - A)$.

All of this is not helpful when we ask for the minimal perturbation in case of an *eigentriple*. Obviously it is possible to have *two* perturbations of sizes that can be measured by application of the lemma, but the perturbations to achieve equality may not coincide. The backward error for an eigentriple is given by the following theorem due to Kahan, Parlett and Jiang:

**Lemma 1.35 ([KPJ82], Theorem 2')** *Let $A$ and two unit vectors $\tilde{w}^H$ and $\tilde{v}$, with $\tilde{w}^H\tilde{v} \neq 0$, be given. For any scalar $\tilde{\lambda}$ define residual vectors*

$$r = \tilde{v}\tilde{\lambda} - A\tilde{v} \qquad \text{and} \qquad \hat{r}^H = \tilde{\lambda}\tilde{w}^H - \tilde{w}^H A.$$

*Then the 2-norm backward error of the eigentriple is given by*

$$\begin{aligned} \eta_2(\tilde{\lambda}, \tilde{v}, \tilde{w}^H) &\equiv \min\{\epsilon : \tilde{A}\tilde{v} = \tilde{v}\tilde{\lambda},\ \tilde{w}^H\tilde{A} = \tilde{\lambda}\tilde{w}^H,\ \|\Delta A\|_2 \leq \epsilon\} \\ &= \max\{\|r\|_2, \|\hat{r}\|_2\} \end{aligned}$$

*and the Frobenius norm backward error of the eigentriple is given by*

$$\begin{aligned} \eta_F(\tilde{\lambda}, \tilde{v}, \tilde{w}^H) &\equiv \min\{\epsilon : \tilde{A}\tilde{v} = \tilde{v}\tilde{\lambda},\ \tilde{w}^H\tilde{A} = \tilde{\lambda}\tilde{w}^H,\ \|\Delta A\|_F \leq \epsilon\} \\ &= \sqrt{\|r\|_2 + \|\hat{r}\|_2 - (\tilde{\lambda}\tilde{w}^H\tilde{v} - \tilde{w}^H A\tilde{v})^2}. \end{aligned}$$

*The Frobenius norm backward error is achieved by* exactly one *perturbation $\Delta A$.*

The obvious choice for $\tilde{\lambda}$ is the Rayleigh quotient

$$\tilde{\lambda} \ = \ \mathrm{R}(\tilde{v}, \tilde{w}^H) \ \equiv \ \frac{\tilde{w}^H A\tilde{v}}{\tilde{w}^H\tilde{v}}.$$

This choice leads to a simple structure of the perturbation that achieves the minimum for both norms, we have that

$$\Delta A = r\tilde{v}^H + \tilde{w}\hat{r}^H.$$

In contrast to the single-sided (including the Hermitian) case, this perturbation is *not the minimal one*. Nevertheless, the authors show that the *minimal value* is not far away from the Rayleigh quotient when the vectors $\tilde{v}$ and $\tilde{w}$ are close to eigenvectors.

A very special case is the formulation of Lemma 1.35 for (nonsymmetric) tridiagonal matrices that are leading (or trailing) submatrices of a larger tridiagonal matrix.

**Lemma 1.36 ([KPJ82], Corollary 2)** *Let $(\theta, s, \hat{s}^H)$ be an eigentriple of the tridiagonal matrix $T_j \in \mathbb{K}^{j\times j}$ that is the leading block of a tridiagonal matrix $T_k \in \mathbb{K}^{k\times k}$, with $\hat{s}^H s = 1$, where*

$$T_j = \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{j-1} \\ & & \beta_{j-1} & \alpha_j \end{pmatrix}, \qquad T_k = \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{pmatrix}.$$

*Then, for all $k > j$,*

$$\left(\theta, \begin{pmatrix} s \\ 0 \end{pmatrix}, \begin{pmatrix} \hat{s} \\ 0 \end{pmatrix}^H\right)$$

*is an eigentriple of $\tilde{T}_k \equiv T_k + \Delta T_k$,*

$$\tilde{T}_k \begin{pmatrix} s \\ 0 \end{pmatrix} = \begin{pmatrix} s \\ 0 \end{pmatrix}\theta, \qquad \begin{pmatrix} \hat{s} \\ 0 \end{pmatrix}^H \tilde{T}_k = \theta \begin{pmatrix} \hat{s} \\ 0 \end{pmatrix}^H,$$

*where*

$$-\Delta T_k \ \equiv \ \frac{\beta_j s_j}{\|s\|_2^2} e_{j+1} \begin{pmatrix} s \\ 0 \end{pmatrix}^H + \frac{\gamma_j \check{s}_j}{\|\hat{s}\|_2^2} \begin{pmatrix} \hat{s} \\ 0 \end{pmatrix} e_{j+1}^T.$$

*Moreover*

$$\|\Delta T_k\|_2 \;=\; \max\left\{\frac{|\beta_j s_j|}{\|s\|_2}, \frac{|\gamma_j \check{s}_j|}{\|\hat{s}\|_2}\right\}, \quad \|\Delta T_k\|_F \;=\; \sqrt{\frac{|\beta_j s_j|^2}{\|s\|_2^2} + \frac{|\gamma_j \check{s}_j|^2}{\|\hat{s}\|_2^2}}.$$

*Especially, both norms are independent of $k > j$.*

The lemma shows that eigentriples will *persist* when the matrix is enlarged whenever the *left and right* residual is small.

We shortly mention that it is possible to give an expression for the *componentwise backward error* of an *approximate eigenpair*. This follows by application of the Oettli-Prager theorem to the eigenequation $Av = v\lambda$ interpreted as linear system. Again the perturbation in the right-hand side is zero.

**Lemma 1.37** *Let $A \in \mathbb{K}^{n\times n}$. Denote by $\tilde{\lambda}$ an approximate eigenvalue and by $\tilde{v}$ an approximate eigenvector. Let $r = \tilde{v}\tilde{\lambda} - A\tilde{v}$ denote the residual.*

*Then the* componentwise *backward error of an* approximate eigenpair $(\tilde{\lambda}, \tilde{v})$ *is given by*

$$\begin{aligned} c(\tilde{\lambda}, \tilde{v}) &\equiv \min\left\{\epsilon : (A + \Delta A)\tilde{v} = \tilde{v}\tilde{\lambda},\ |\Delta A| \le \epsilon E\right\} \\ &= \max_i \frac{|r|_i}{(E|\tilde{v}|)_i}. \end{aligned}$$

*Accordingly, componentwise backward errors $c(\tilde{v})$ and $c(\tilde{\lambda})$ for single quantities may be derived by minimisation of $c(\tilde{\lambda}, \tilde{v})$ over all $\tilde{\lambda}$ and $\tilde{v}$.*

**Proof.** The proof follows by Lemma 1.5, when setting the perturbation in the right-hand side of $A\tilde{v} = \tilde{v}\tilde{\lambda}$ to zero. □

### Eigenvalue and Eigenspace Expansions

Well-known examples show that the eigenvalues of a Jordan block of size $n$ subject to a generic perturbation of size $\epsilon$ move in the worst-case with size comparable to $\sqrt[n]{\epsilon}$. For large values of $n$, this makes perturbation theory unappealing. The situation changes when we consider the arithmetic mean of the eigenvalues. We remark that the arithmetic mean of the eigenvalues is given by the trace of the matrix divided by size. Thus, usually the arithmetic mean of the eigenvalues is to be preferred. A remarkable exception is the Lidskii–Vishik–Lyusternik perturbation theory, where the explicit computation of the leading terms of the eigenvalue expansion in terms of Schur complements of submatrices of the perturbation matrix $\Delta A$ is considered. This perturbation theory may be found in the paper by Moro, Burke and Overton (cf. [MBO97]).

The non-linearity of the eigenproblem makes it hard to obtain other expressions than expansions of perturbed quantities in terms of known quantities. We set $A(t) = A + t\Delta A$. Then, $A(0) = A$ and $A(1) = \tilde{A} = A + \Delta A$ holds true. We try to find expansions in powers of $t$ that describe the variation of invariant subspaces, (clusters of) eigenvalues and the spectral projector. The most famous amongst these expansions are the so-called *Rellich-Kato* and *Rayleigh-Schrödinger* expansions. The following results are excerpts from the book of Chatelin (cf. [Cha93], pages 85–89).

The Rellich-Kato expansion expresses the arithmetic mean of a cluster of eigenvalues of a perturbed matrix and the associated spectral projector in terms of the perturbation, the arithmetic mean of the unperturbed matrix and associated spectral projector. The Rellich-Kato expansion is merely of theoretical interest.

**Lemma 1.38 (Rellich-Kato Expansion)** *Let $P$ represent the spectral projection of $A \in \mathbb{K}^{n\times n}$ associated with the eigenvalue $\lambda$ of index $\ell$ and multiplicity $m$. Let $\Gamma$ be a Jordan curve drawn in res$(A)$ and isolating $\lambda$. Define formally*

$$R(t,z) \;\equiv\; (zI - A(t))^{-1}, \qquad P(t) = \frac{1}{2i\pi}\int_\Gamma R(t,z)\,dz.$$

*Then, for every $t$ with $|t| < \max_{z\in\Gamma} \rho(\Delta A R(z))^{-1}$ we have the expansions*

$$P(t) = P - \sum_{k=2}^{\infty} t^{k-1} \sum_{\star} S^{(p_1)}\Delta A S^{(p_2)} \cdots \Delta A S^{(p_k)}$$

*and*

$$\lambda_c(t) = \lambda + \frac{1}{m}\sum_{k=1}^{\infty}\frac{1}{k}\sum_{\star} \operatorname{trace}\left[\Delta A S^{(p_1)}\Delta A S^{(p_2)} \cdots \Delta A S^{(p_k)}\right]$$

*where $\lambda_c(t)$ is the arithmetic mean of the perturbed eigenvalues and*

$$\star = \left\{ p_i \geq -\ell + 1, \quad i \in \underline{k}, \quad \sum_{i=1}^{k} p_i = k-1 \right\},$$
$$S^{(0)} = P, \quad S^{(-p)} = D^p, \quad S^{(p)} = S^p, \qquad \text{when} \quad p > 0$$
$$D = (A - \lambda I)P \qquad \text{and} \qquad S = \lim_{z\to\lambda} R(z)(I-P).$$

This lemma is included just to show that it is possible to expand the arithmetic mean and the spectral projector of *any* perturbed eigenvalue, including even the derogatory and defective cases. The next expansion is computationally of more use. It shows how to expand an eigenspace and a corresponding basis. This expansion is known as the Rayleigh–Schrödinger expansion:

**Lemma 1.39 (Rayleigh–Schrödinger Expansion)** *Let $A \in \mathbb{K}^{n\times n}$ be given. Let a subspace relation be given by $AV = VB$, where $B \in \mathbb{K}^{m\times m}$. Let $(V, V_c)$ be a basis for $\mathbb{K}^{n\times n}$. Now, we consider the family of matrices $A(t) = A + t\Delta A$. Suppose that the eigenspace can be expanded. Let the formal expansion be given by*

$$V(t) = \sum_{k=0}^{\infty} Z_k t^k, \qquad B(t) = \sum_{k=0}^{\infty} C_k t^k.$$

*Then the coefficients fulfil formally the sets of equations*

$$C_0 = B, \qquad C_k = W^H(AZ_k + \Delta A Z_{k-1}),$$
$$Z_0 = V, \qquad Z_k = \Sigma\,(\sum_{i=1}^{k-1} Z_{k-i}C_i - \Delta A Z_{k-1}).$$

*Here, $(W, W_c)$ is an adjoint basis of $(V, V_c)$, where $W$ not necessarily is the left invariant subspace, and $\Sigma$,*

$$\Sigma = V_c(W^H A V, W_c^H A V_c)^{-1} W_c^H$$

*is a block partial inverse to the block of eigenvalues we are interested in.*

**Proof.** Let the subspace relation be given by $A(t)V(t) = V(t)B(t)$. We assume that $W^H V(t) = I$ holds true for all $t$, which is possible for $\Delta A$ small enough. Then

we can set $B(t) = W^H A(t)V(t)$. Inserting the expansion gives

$$\begin{aligned}(A + t\Delta A)\sum_{k=0}^{\infty} Z_k t^k &= \left(\sum_{k=0}^{\infty} Z_k t^k\right)\left(\sum_{i=0}^{\infty} C_i t^i\right)\\ AZ_0 + \sum_{k=1}^{\infty}(AZ_k + \Delta A Z_{k-1})\,t^k &= \sum_{k=0}^{\infty}\left(\sum_{i=0}^{k} Z_{k-i}C_i\right)t^k.\end{aligned}$$

The proof proceeds by comparing coefficients of $t^k$. □

The two expansions are excerpts form the book of Chatelin. She shows that the disc $|t| \leq \rho^{-1}$ belongs to the domain of analyticity of $B(t)$ and $V(t)$, where $\rho$ is obtained by

$$\rho = \frac{|\Gamma|}{\pi}\|\Pi\|_2\|\Delta A\|_2 \max_{z\in\Gamma}\|R(z)\|_2.$$

Here, $\Pi = VW^H$ is the (oblique) projector defined by the adjoint spaces $V$ and $W$. When $W = \hat{V}$ spans the right invariant subspace, we obtain the spectral projector, $\Pi = P$. When $\rho < 1$, i.e., when the perturbation $\Delta A$ is sufficiently small, we have that the invariant subspace $\tilde{V} = V(1)$ and subspace representation $\tilde{B}$ of the perturbed matrix $\tilde{A}$ are given by

$$\tilde{V} = \sum_{k=0}^{\infty} Z_k, \qquad \tilde{B} = \hat{V}^H \tilde{A}\tilde{V}.$$

She also shows that the coefficients in the expansion decay geometrically, i.e., like

$$\|Z_k\|_2 \leq \alpha q^k, \qquad \|C_k\|_2 \leq \beta q^k$$

where the rate $q$ is a constant strictly larger than $\rho$ and (naturally) strictly less than one (cf. [Cha93], page 89).

### Complete Eigensystems

Without any knowledge on eigenvectors or eigenvalues we can only obtain very *crude* and in general *pessimistic* bounds on the variation of the eigenvalues. The classical theorems on the variation of the spectrum or single eigenvalues that do not need any information about the matrix $A$ are the theorems of Ostrowski and Elsner. These theorems may be found in the textbook of Stewart and Sun (cf. [SS90], pages 167–170).

*With* knowledge on the eigenvalues and eigenvectors or other properties of a 'near-by' problem one can do significantly better. Many results on the *entire* spectrum are based on instances of the following theorem:

**Theorem 1.40 (Bauer, Fike)** *Let $A \in \mathbb{K}^{n\times n}$. Let $\tilde{\lambda}$ be an eigenvalue of $\tilde{A} = A + \Delta A$. Then, for* any choice *of a regular matrix $Q \in \mathbb{K}^{n\times n}$ and* all *consistent norms $\|\cdot\|$ the following bound holds true:*

$$\|Q^{-1}(\tilde{\lambda}I - A)^{-1}Q\|^{-1} = \|Q^{-1}R(\tilde{\lambda})Q\|^{-1} \leq \|Q^{-1}\Delta A Q\|.$$

**Proof.** When $\tilde{\lambda}$ is an eigenvalue of $A$, there is nothing to prove. From now on we suppose that $\tilde{\lambda}$ is *no* eigenvalue of $A$. Let $\tilde{v}$ denote any eigenvector of $\tilde{A}$ to eigenvalue $\tilde{\lambda}$. This vector may not be uniquely defined, but at least *one* eigenvector *must* exist. Then $(\tilde{\lambda}I - A)\tilde{v} = \Delta A\tilde{v}$ holds true. We multiply this equation by $Q^{-1}$ and transform $\tilde{v} = QQ^{-1}\tilde{v}$ to obtain

$$Q^{-1}(\tilde{\lambda}I - A)QQ^{-1}\tilde{v} = Q^{-1}\Delta A Q Q^{-1}\tilde{v}.$$

Our assumption that $\tilde{\lambda}$ is not contained in the spectrum of $A$ implies that the matrix $Q^{-1}(\tilde{\lambda}I - A)Q$ is non-singular and thus can be inverted. Defining $y = Q^{-1}\tilde{v}$ and taking norms we obtain

$$\begin{aligned} \|y\| &\leq \|Q^{-1}(\tilde{\lambda}I - A)^{-1}QQ^{-1}\Delta AQ\|\|y\| \\ &\leq \|Q^{-1}(\tilde{\lambda}I - A)^{-1}Q\|\|Q^{-1}\Delta AQ\|\|y\|. \end{aligned}$$

Upon division by $\|(\tilde{\lambda}I - Q^{-1}AQ)^{-1}\|\|y\|$ we obtain the desired result. □

This theorem is more or less a variation of Theorem II in the original paper of Bauer and Fike (cf. [BF60]). To simplify subsequent steps we followed Stewart and Sun (cf. [SS90], Theorem 1.6, page 171) in adding the similarity transformation. Actually, the proof implies a slightly stronger, but less usable result:

**Lemma 1.41 ([BF60], Theorem I)** *Let $A$, $\Delta A$ and $R(\tilde{\lambda})$ be defined as in Theorem 1.40. Then for all consistent matrix norms the following holds true:*

$$\|R(\tilde{\lambda})\Delta A\|^{-1} \leq \rho(R(\tilde{\lambda})\Delta A)^{-1} \leq 1.$$

**Proof.** We observe that the vector $\tilde{v}$ is an *eigenvector* of the matrix $R(\tilde{\lambda})\Delta A$ to the eigenvalue *one*. The spectral radius is defined to be the *maximal* modulus of *all* eigenvalues. This finishes the proof. □

The general result of Theorem 1.40 leaves some freedom in the choice of the matrix $Q$ and the norm. Taking $Q$ as eigenvector matrix $V$, we obtain the following corollary:

**Corollary 1.42 ([BF60], Theorem IIIa)** *Let $A$, $\Delta A$ be defined as in Theorem 1.40. Let the Jordan decomposition of $A$ be given as $A = VJV^{-1}$. Then for all consistent matrix norms and all eigenvalues $\tilde{\lambda}$ of $\tilde{A} = A + \Delta A$ the following holds true:*

$$\|(\tilde{\lambda}I - J)^{-1}\|^{-1} \leq \|V^{-1}\Delta AV\| \leq \kappa(V)\|\Delta A\|. \tag{1.19}$$

*Let $\|\cdot\|_p$ be a* Hölder *norm. Then there exists an eigenvalue $\lambda$ of $A$ with index $m = m_\lambda$ such that*

$$\frac{|\tilde{\lambda} - \lambda|}{\sum_{i=0}^{m-1}|\tilde{\lambda} - \lambda|^{-i}} = \frac{|\tilde{\lambda} - \lambda|^m}{\sum_{i=0}^{m-1}|\tilde{\lambda} - \lambda|^i} \leq \|V^{-1}\Delta AV\|_p \leq \kappa(V)\|\Delta A\|_p \tag{1.20}$$

*holds true.*

*Let $A$ be* diagonalisable *and $\|\cdot\|$ be an* axis-oriented *norm. Then there exists an eigenvalue $\lambda$ of $A$ such that*

$$\min|\tilde{\lambda} - \lambda| \leq \|V^{-1}\Delta AV\| \leq \kappa(V)\|\Delta A\| \tag{1.21}$$

*holds true.*

**Proof.** Equation (1.19) follows directly from Theorem 1.40. To prove equation (1.20), we have to show that

$$\min_\lambda \frac{|\tilde{\lambda} - \lambda|}{\sum_{i=0}^{m-1}|\tilde{\lambda} - \lambda|^{-i}} = \min_\lambda \frac{|\tilde{\lambda} - \lambda|^m}{\sum_{i=0}^{m-1}|\tilde{\lambda} - \lambda|^i} \leq \|(\tilde{\lambda}I - J)^{-1}\|_p^{-1}$$

for all Hölder norms. We stress that $m$ depends on $\lambda$. First, note that for *all* block-diagonal matrices $B = B_1 \oplus \cdots \oplus B_k$, $\|B\|_p = \max_j\{\|B_j\|_p\}$. We partition

$x = x_1 \oplus \cdots \oplus x_k$ accordingly. One direction follows by definition,

$$\begin{aligned}
\|B\|_p^p &= \max_x \frac{\|Bx\|_p^p}{\|x\|_p^p} = \max_x \left\{ \frac{\|B_1 x_1\|_p^p}{\|x\|_p^p} + \cdots + \frac{\|B_k x_k\|_p^p}{\|x\|_p^p} \right\} \\
&= \max_x \left\{ \frac{\|B_1 x_1\|_p^p}{\|x_1\|_p^p} \frac{\|x_1\|_p^p}{\|x\|_p^p} + \cdots + \frac{\|B_k x_k\|_p^p}{\|x_k\|_p^p} \frac{\|x_k\|_p^p}{\|x\|_p^p} \right\} \\
&\leq \max_{\alpha_1+\cdots+\alpha_k=1,\ \alpha_j \geq 0} \left\{ \|B_1\|_p^p \alpha_1 + \cdots + \|B_k\|_p^p \alpha_k \right\} = \max_j \{\|B_j\|_p^p\}.
\end{aligned}$$

Now, suppose the maximal block is given by $B_j$. Equality is attained by choosing $x$ in the form

$$x = (\, 0 \quad \cdots \quad 0 \quad x_j^T \quad 0 \quad \cdots \quad 0 \,)^T,$$

where the vector $x_j$ attains equality for the block $B_j$, i.e., $x_j$ fulfils $\|B_j x_j\|_p = \|B_j\|_p \|x_j\|_p$. Let $J_\lambda$ be the largest Jordan block to the eigenvalue $\lambda$ of size $m$. The result (1.20) follows, since the norm of the matrix $(J_\lambda - \tilde{\lambda} I)^{-1}$ can be bounded using the triangle inequality and a Neumann series:

$$\begin{aligned}
\|(J_\lambda - \tilde{\lambda} I)^{-1}\|_p &= \|(\lambda - \tilde{\lambda})^{-1} (I + (\lambda - \tilde{\lambda})^{-1} N)^{-1}\|_p \\
&\leq \frac{1}{|\tilde{\lambda} - \lambda|} \sum_{i=0}^{m-1} \frac{\|N^i\|_p}{|\tilde{\lambda} - \lambda|^i} = \frac{1}{|\tilde{\lambda} - \lambda|} \sum_{i=0}^{m-1} \frac{1}{|\tilde{\lambda} - \lambda|^i}.
\end{aligned}$$

The proof for the validity of equation (1.21) is obvious. The results and proofs may also be found in the book by Stewart and Sun (cf. [SS90]) and in an early preprint of Li (cf. [Li85]). □

The result (1.21) is often quoted as 'the' Bauer–Fike theorem. The corollary shows that it may be possible that an eigenvalue of a perturbed matrix has its origin *not* in the *closest* eigenvalue of the original matrix, but in a Jordan block whose eigenvalue is further apart, but more sensitive to perturbations.

Instead of the Jordan form we can use the Schur form. For any norm we define the departure from normality $\nu(A)$ of $A$ by

$$\nu(A) \equiv \min_R \{\|R - \operatorname{diag}(R)\|,\ R \text{ is a Schur form of } A\}.$$

The Schur form leads to the following corollary, originally due to Henrici:

**Corollary 1.43 (Henrici)** *Let $\|\cdot\|$ be a norm on $\mathbb{C}^{n\times n}$ such that $\|C\| \geq \|C\|_2$ for all $C \in \mathbb{C}^{n\times n}$. Let $\nu(A)$ be the departure from normality measured in this norm. Then for every eigenvalue $\tilde{\lambda}$ of $\tilde{A}$ there is an eigenvalue $\lambda$ of $A$ such that*

$$\frac{\left(\nu(A)^{-1} |\tilde{\lambda} - \lambda|\right)^n}{\sum_{i=0}^{n-1} \left(\nu(A)^{-1} |\tilde{\lambda} - \lambda|\right)^i} \leq \nu(A)^{-1} \|\Delta A\|_2.$$

**Proof.** The proof proceeds analogously to the proof of Corollary 1.42 and can be found written out in the book of Stewart and Sun (cf. [SS90], Theorem 1.9, page 172). □

We can set $\Delta A$ to $A$ without diagonal and use the $\|\cdot\|_\infty$ norm in the Bauer-Fike theorem. This leads to the result first proven by Gerschgorin:

**Corollary 1.44 (Gerschgorin)** *For $A \in \mathbb{K}^{n\times n}$ let*

$$r_i = \sum_{j \neq i} |a_{ij}| \quad \text{and} \quad \mathcal{G}_i(A) = \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\}.$$

*Then*

$$\Lambda(A) \subset \bigcup_{i=1}^{n} \mathcal{G}_i(A).$$

*Moreover, if $m$ of the* Gerschgorin disks $\mathcal{G}_i(A)$ *are isolated from the other $n-m$ disks, then there are precisely $m$ eigenvalues of $A$ in their union.*

**Proof.** The first part follows directly from the Bauer-Fike theorem as mentioned above. That every connected component of $m$ circles contains precisely $m$ eigenvalues follows by a simple continuity argument. □

The Bauer–Fike approach can be generalised, which seems to be unknown yet.

**Theorem 1.45 ("Unsymmetric" Bauer-Fike Theorem)** *Let all notation be as defined in Theorem 1.40. Furthermore, let $W \in \mathbb{K}^{n\times n}$ be any regular matrix. Then the following holds true:*

$$\|W^{-1}(\tilde{\lambda}I - A)^{-1}Q\|^{-1} = \|W^{-1}R(\tilde{\lambda})Q\|^{-1} \leq \|Q^{-1}\Delta AW\|.$$

**Proof.** The proof is quite analogous to the proof of Theorem 1.40. The only difference is that we substitute $WW^{-1}\tilde{v}$ and not $QQ^{-1}\tilde{v}$ for $\tilde{v}$. □

This theorem can be used to bring the resolvent *or* the perturbation matrix into some canonical form *not necessarily related to similarity transformations.* The use of the *SVD of the (inverted) resolvent* for instance results in the following corollary:

**Corollary 1.46** *Let all notation be as in Theorem 1.45. Let $U\Sigma V^H = \tilde{\lambda}I - A$ be the singular value decomposition of the matrix $\tilde{\lambda}I - A$. Then the following holds true:*

$$\|\Sigma^{-1}\|^{-1} \leq \|U^H\Delta AV\| \leq \|U^H\|\|V\|\|\Delta A\|.$$

*When $\|\cdot\|$ is axis-oriented, this implies*

$$\sigma_{\min}(\tilde{\lambda}I - A) \leq \|U^H\Delta AV\| \leq \|U^H\|\|V\|\|\Delta A\|.$$

*Let $|||\cdot|||$ be any unitarily invariant norm. Then the following bound holds true:*

$$|||\Sigma^{-1}|||^{-1} \leq |||\Delta A|||$$

*For the special choice of the 2-norm, $|||\cdot||| = \|\cdot\|_2$, this implies that*

$$\sigma_{\min}(\tilde{\lambda}I - A) = \left(\sigma_{\max}(R(\tilde{\lambda}))\right)^{-1} \leq \|\Delta A\|_2$$

*holds true. This result, of course, is obvious by definition of the backward error for an approximate eigenvalue. For the special choice of the Frobenius norm, $|||\cdot||| = \|\cdot\|_F$, we obtain the new result*

$$\left(\sum_{i=1}^{n} \frac{1}{\sigma_i^2}\right)^{-1/2} \leq \|\Delta A\|_F.$$

**Proof.** The results follow upon taking $Q = U$ and $W = V$. □

When the matrix $A$ is normal or even Hermitian, we can do significantly better. We state the well-known theorems of Wielandt-Hoffman and Weyl.

**Theorem 1.47 (Wielandt, Hoffman)** *Let $A$ and $\tilde{A}$ be normal. Let $\pi$ denote a permutation of the set $\underline{n}$. Then the following holds true:*

$$\min_{\pi} \sqrt{\sum_{i} |\tilde{\lambda}_{\pi(i)} - \lambda_i|^2} \leq \|\Delta A\|_F$$

**Proof.** The proof may be found in volume one of the textbook by Horn and Johnson (cf. [HJ85], Theorem 6.3.5, page 368). □

The theorem is not true when the Frobenius norm is replaced by an arbitrary unitarily invariant norm. For *Hermitian* matrices this theorem can be generalised to arbitrary unitarily invariant norms with the aid of symmetric gauge functions and the following Theorem by Mirsky:

**Theorem 1.48 (Mirsky)** *Let $A \in \mathbb{K}^{n\times m}$, and let $\tilde{A} \in \mathbb{K}^{n\times m}$ be a perturbation of $A$, $\tilde{A} = A + \Delta A$. Let $A = U\Sigma V^H$ and $\tilde{A} = \tilde{U}\tilde{\Sigma}\tilde{V}^H$ denote the singular value decomposition of $A$ and $\tilde{A}$. Then for all unitarily invariant norms $\|\!|\cdot|\!\|$ the following holds true:*

$$\|\!|\tilde{\Sigma} - \Sigma|\!\| \leq \|\!|\Delta A|\!\|.$$

**Proof.** The proof may be found in the book by Stewart and Sun (cf. [SS90], Theorem 4.11, pages 204f). □

Because of the one-to-one correspondence between symmetric gauge functions (cf. [SS90], Definition 3.3, page 75) and unitarily invariant norms (cf. [SS90], Theorem 3.6, page 78) this implies the following theorem for Hermitian matrices:

**Theorem 1.49** *Let $A$ and $A$ be Hermitian. Then for all symmetric gauge functions*

$$\|\tilde{\Lambda} - \Lambda\|_\Phi \leq \|\Delta A\|_\Phi$$

*holds true. The special case of the 2-norm is also known as the* Theorem of Weyl*,*

$$\min |\tilde{\lambda} - \lambda| \leq \|\Delta A\|_2.$$

*The special case of the Frobenius norm improves over the Wielandt-Hoffman theorem for normal matrices,*

$$\sqrt{\sum_{i=1}^{n} |\tilde{\lambda} - \lambda|^2} \leq \|\Delta A\|_F,$$

*since now the ordering is* explicitely *given.*

**Proof.** The result may be found in the book of Stewart and Sun (cf. [SS90], Corollary 4.12, page 205). □

In case of Hermitian matrices the eigenvalues can be characterised using Theorem 1.24. This implies the stronger version of the Theorem of Weyl:

**Lemma 1.50 (Weyl)** *Let $A \in \mathbb{K}^{n\times n}$ and $\tilde{A} \in \mathbb{K}^{n\times n}$ be Hermitian with eigenvalues $\lambda_i$ and $\tilde{\lambda}_i$. Let $\epsilon_{\min}$ and $\epsilon_{\max}$ denote the minimal and maximal eigenvalue of the perturbation $\Delta A = \tilde{A} - A$. Then the following holds true:*

$$\tilde{\lambda}_i \in [\lambda_i + \epsilon_{\min}, \lambda_i + \epsilon_{\max}].$$

This lemma is a corollary of an even stronger result which may be found in the book by Stewart and Sun (cf. [SS90], Theorem 4.8, page 202).

**Semi-Simple Eigenvalues**

Let $\lambda$ be a semi-simple eigenvalue of $A$ with right eigenvector $v$ and left eigenvector $\hat{v}^H$ such that $\hat{v}^H v \neq 0$. Let $\tilde{A} = A + \Delta A$ be the perturbed matrix. A first order bound for the simple eigenvalue $\lambda$ is based on the observation that the Rayleigh quotient provides a first-order approximation to the perturbed eigenvalue $\tilde{\lambda}$,

$$\begin{aligned}\tilde{\lambda} &= \frac{\hat{v}^H(A+\Delta A)v}{\hat{v}^H v} + O(\|\Delta A\|^2) \\ &= \lambda + \frac{\hat{v}^H \Delta A v}{\hat{v}^H v} + O(\|\Delta A\|^2).\end{aligned}$$

This is proven algebraically in the textbook of Stewart and Sun (cf. [SS90], Theorem 2.3, page 183). The first-order behaviour shows that the derivatives of $\lambda$ with respect to the matrix entries are given by

$$\frac{\partial \lambda}{\partial a_{ij}} = \frac{\check{v}_i v_j}{\hat{v}^H v}.$$

This shows that the condition of the eigenvalues depends on the variation of *all* $n^2$ entries of $A$. As Wilkinson ([Wil65], page 89) pointed out, it might be necessary to observe *all* $n^2$ derivatives to unseal all the mysteries of eigenvalue perturbations. For all $n$ eigenvalues this would sum up to $n^3$ important data.

The first-order representation directly implies a condition number. A crude norm estimation gives the first-order bound

$$|\tilde{\lambda} - \lambda| \leq \kappa(\lambda)\|\Delta A\| + O(\|\Delta A\|^2), \qquad \kappa(\lambda) \equiv \frac{\|\hat{v}\|\|v\|}{|\hat{v}^H v|}.$$

The quantity $\kappa(\lambda)$ is known as the *Wilkinson condition number* of the simple eigenvalue $\lambda$. It measures the secant of the angle between $\hat{v}$ and $v$. Observe that in the class of non-derogatory matrices iff $\hat{v}$ and $v$ are eigenvectors to a non-simple Jordan block, they are orthogonal. Nevertheless, they may even be arbitrarily close to orthogonality for a semi-simple eigenvalue (cf. [SS90], page 185f).

When the norm is the 2-norm, the Wilkinson condition number $\kappa(\lambda)$ is closely related to the condition of a specially scaled eigenmatrix $V$ as follows:

**Lemma 1.51 (Nearly Optimal Scalings)** *Let $V \in \mathbb{K}^{n\times n}$ be an eigenmatrix of the matrix $A \in \mathbb{K}^{n\times n}$. Let $\|\cdot\|_p$ be a Hölder norm. Then the condition $\kappa_p(V) \equiv \|V\|_p\|V^{-1}\|_p$ of $V$ is bounded from below by the Wilkinson condition of the eigenvalues,*

$$\max_\lambda \kappa_p(\lambda) \leq \kappa_p(V). \tag{1.22}$$

*Now, let $\|\cdot\|_2$ be the 2-norm. Suppose that $V$ has been scaled by a diagonal scaling $V \leftarrow VD$ such that the $j$th column has unit length. Then, furthermore*

$$\max_\lambda \kappa_2(\lambda) \leq \kappa_2(V) \leq n \max_\lambda \kappa_2(\lambda) \tag{1.23}$$

*holds true. Next, consider the scaling such that the $j$th column has length $\sqrt{\kappa(\lambda_j)}$. With this particular scaling, the last bound can be sharpened to give*

$$\max_\lambda \kappa_2(\lambda) \leq \kappa_2(V) \leq \sum_\lambda \kappa_2(\lambda) \leq n \max_\lambda \kappa_2(\lambda). \tag{1.24}$$

**Proof.** The first inequality (1.22) follows from the observation that for all $j \in \underline{n}$, $\|\hat{v}_j^H\|_p \leq \|\hat{V}^H\|_p\|e_j\|_p = \|V^{-1}\|_p$ and $\|v_j\|_p \leq \|V\|_p\|e_j\|_p = \|V\|_p$ holds true. Thus,

the first equality *in general* holds true, i.e., *independent of scalings*. The other two inequalities hold true, since upon setting $\hat{V}^H \equiv V^{-1}$ we can conclude that

$$\|V\|_2\|\hat{V}\|_2 \leq \|V\|_F\|\hat{V}\|_F = \left(\sum_i \|v_i\|_2^2\right)^{\frac{1}{2}} \left(\sum_j \|\hat{v}_j\|_2^2\right)^{\frac{1}{2}} \tag{1.25}$$

holds true. In case $V$ has columns of unit length, i.e., in case of inequality (1.23), we observe that

$$\kappa_2(\lambda_i) = \frac{\|\hat{v}_i\|_2\|v_i\|_2}{|\hat{v}_i^H v_i|} = \|\hat{v}_i\|_2.$$

Hence, with $\|v_i\| = 1$, the last part of equation (1.25) can be bounded by

$$\sqrt{n}\sqrt{\sum_j \|\hat{v}_j\|_2^2} \leq n \max_j \|v_j\|_2 = n \max_j \kappa_2(\lambda_j).$$

When the columns of $V$ are scaled such that $\|v_i\|_2 = \sqrt{\kappa_2(\lambda_i)}$, we observe that also the rows of $\hat{V}^H \equiv V^{-1}$ are scaled this way, since

$$\kappa_2(\lambda_i) = \frac{\|\hat{v}_i\|_2\|v_i\|_2}{|\hat{v}_i^H v_i|} = \sqrt{\kappa_2(\lambda_i)}\|\hat{v}_i\|.$$

Thus, the columns of $V$ *and* the rows of $\hat{V}^H = V^{-1}$ are equilibrated, $\|v_i\| = \|\hat{v}_i^H\| = \sqrt{\kappa(\lambda_i)}$. In this case, the last part of equation (1.25) can be bounded by

$$\sqrt{\sum_i \kappa_2(\lambda_i)}\sqrt{\sum_i \kappa_2(\lambda_i)} = \sum_i \kappa_2(\lambda_i).$$

This finishes the proof. □

The last scaling was introduced by Wilkinson (cf. [Wil65], pages 88–89). Demmel invented the (simpler) scaling of $V$, $\|v_i\| = 1$ for all $i \in \underline{n}$. This scaling is used in tools like LAPACK and Matlab.

The Wilkinson condition number of a semi-simple eigenvalue $\lambda$ is closely linked to the limes of the resolvent, we have that

$$\frac{\|\hat{v}\|\|v\|}{|\hat{v}^H v|} = \lim_{z\to\lambda} \frac{|z-\lambda|}{\|R(z)\|}$$

This is readily verified when we bring to mind the Laurent expansion of the resolvent in a punctured neighbourhood of $\lambda$ of equation (1.13). Thus, the first-order bound for semi-simple eigenvalues is usually stated in terms of the norm of the associated spectral projector. We state the first-order bound and a global bound:

**Theorem 1.52** *Let $\lambda$ be a simple eigenvalue of $A \in \mathbb{K}^{n\times n}$. Let $P$ denote the spectral projector onto the eigenspace associated with $\lambda$. Let $\tilde{A} = A + \Delta A$. Then there exists an eigenvalue $\tilde{\lambda}$ of $\tilde{A}$, such that*

$$|\tilde{\lambda} - \lambda| \leq \|P\|_2\|\Delta A\|_2 + O(\|\Delta A\|_2^2)$$

*holds true. The first-order bound can be turned into a global bound by multiplication by the dimension $n$:*

$$|\tilde{\lambda} - \lambda| \leq n\|P\|_2\|\Delta A\|_2.$$

The proof for this theorem can be found in the paper by Bauer and Fike (cf. [BF60]). This is a slightly stronger result than the before mentioned one.

We switch to eigenvectors. An eigenvector is not uniquely defined, since with $v \in \mathbb{K}^n$ every $\alpha v$, $\alpha \in \mathbb{K}$ non-zero is an eigenvector to the same eigenvalue. The linear space spanned by an eigenvector is turned by an angle when the matrix $A$ is subject to a perturbation. When the corresponding eigenvalue is simple, we can find a normwise and mixed condition number and a bound on the variation of the angle between the unperturbed and the perturbed eigenvector.

The *relative normwise condition number* of a simple eigenvector is linked to a partial inverse, defined by

$$\Sigma = V \left(\lambda I - W^H A V\right)^{-1} W^H, \tag{1.26}$$

where $(v, V)$ and $(w, W)$ are adjoint bases of $\mathbb{K}^n$, where $v$ is the unperturbed eigenvector and $w$ is any vector such that $w^H v = 1$. Then, by direct calculation or by application of the Rayleigh-Schrödinger expansion, to first-order

$$\Delta v \approx \Sigma \Delta A v$$

holds true.

**Theorem 1.53** *Let $(\lambda, v)$ be a simple eigenpair of $A \in \mathbb{K}^{n\times n}$. Let $\tilde{A} = A + \Delta A$ be a perturbation of $A$. Let $\Sigma$ be defined as in equation (1.26). Then, the* normwise condition number *of $v$ is given by*

$$\kappa(v) = \alpha \|\Sigma\|,$$

*and the* mixed condition number *of $v$ is given by*

$$\kappa(v) = \frac{\| |\Sigma| E |v| \|_\infty}{\|v\|_\infty}.$$

The normwise condition number in full generality has its origins in the works of Chatelin (cf. [Cha93, CCF96]). The mixed condition number is due to Geurts (cf. [CCF96], page 63).

When the basis $(v, V)$ has been chosen unitary, the perturbation $\Delta v = \Sigma \Delta A v$ is such that $\|\Delta v\|_2 = \tan\theta$ gives the tangent of the angle between the space spanned by $v$ and $\tilde{v}$. This choice of a basis is due to Stewart and ensures that the condition of the eigenvalue and the eigenvector are not necessarily related. This is in contrast to the choice of Wilkinson, who proposed to use the *left* eigenvector as $w$ (cf. [CCF96], pages 63–64).

### Clustered Eigenvalues

More interesting is the case of a multiple eigenvalue or cluster and corresponding invariant subspace. We give a summary of the bounds that apply to clusters (including multiple eigenvalues) and invariant subspaces. When we are talking about a cluster of close eigenvalues, we have to be sure that no other eigenvalues come close when perturbing. So it seems natural to *restrict* the size of the perturbation $\Delta A$ in order to investigate *only* clusters perturbed such that no other eigenvalues interact significantly. Let $A$ be in partial Schur form like in Definition 1.29. Then this is the case, as long as the spectra of $A_{11}$ and $A_{22}$ remain disjoint, which is the case, when

$$\|\Delta A\|_F < \frac{\operatorname{sep}_F(A_{11}, A_{22})}{4\|P\|_2} \tag{1.27}$$

holds true. Here, $P$ is the spectral projector onto the eigenspace associated with the spectrum of $A_{11}$. The proof can be found in an article by Stewart (cf. [Ste73]). The expansion of the single eigenvalues in general would have to be based on fractional powers of the perturbation. In the following, $\lambda$ is the *arithmetic mean* of the eigenvalues that are part of the cluster. This average of the eigenvalues can be computed by means of the trace,

$$\lambda_c = \mathrm{trace}(A_{11})/m, \qquad \tilde{\lambda}_c = \mathrm{trace}(\tilde{A}_{11})/m.$$

Here, the cluster is assumed to be of size $m$ with the Schur forms of $A$ and $\tilde{A}$ chosen such that the correspond to the matrices $A_{11}$ and $\tilde{A}_{11}$, respectively.

**Theorem 1.54** *Let $A \in \mathbb{K}^{n\times n}$ be perturbed to $\tilde{A} = A + \Delta A$. Let $\lambda_c$ be the mean of the eigenvalues of the cluster we are interested in. Let $\tilde{\lambda}_c$ denote the mean of the perturbed cluster. Then, a first-order bound is given by*

$$|\tilde{\lambda}_c - \lambda_c| \leq \|P\|_2 \|\Delta A\|_2 + O(\|\Delta A\|_2^2).$$

*As long as condition (1.27) holds true, a global bound is given by*

$$|\tilde{\lambda}_c - \lambda_c| \leq 2\|P\|_2 \|\Delta A\|_2.$$

*In other words, this bound holds as long as the cluster remains disjoint from the remaining part of the spectrum.*

**Proof.** The proof for the first order bound may be found in the book by Kato (cf. [Kat66]). The proof for the global bound may be found in the aforementioned article (cf. [Ste73]). □

Similarly, we can obtain bounds for the variation of an associated invariant subspace:

**Theorem 1.55** *Let all notation be as in Theorem 1.54. Let $\mathcal{V}$ denote the invariant subspace associated with the cluster, and let $\tilde{\mathcal{V}}$ denote the invariant subspace associated with the perturbed cluster. Then, the maximal angle $\theta \equiv \theta_{\max}(\mathcal{V}, \tilde{\mathcal{V}})$ between $\mathcal{V}$ and $\tilde{\mathcal{V}}$ is bounded to first-order by*

$$\theta \leq \frac{2\|\Delta A\|_F}{\mathrm{sep}} + O(\|\Delta A\|_F^2).$$

*Moreover, as long as condition (1.27) holds true, we have the global bound*

$$\theta \leq \mathrm{atan}\left(\frac{2\|\Delta A\|_F}{\mathrm{sep} - 4\|P\|_2\|\Delta A\|_F}\right).$$

*Condition (1.27) ensures that the denominator on the right-hand side of the global bound is positive.*

**Proof.** The proofs may be found in an article by Demmel (cf. [Dem86]). □

All the results without proofs may be found in LAPACK working note 13 (cf. [BDM91]).

### 1.6.4 Subspaces and Projectors

Krylov methods are projection methods. Thus, we summarise the basic results on projectors. A matrix $P$ is a projector when $P^2 = P$ holds true. Therefore, we can

write $\mathbb{K}^n = \ker(P) \oplus \operatorname{ran}(P)$. In other words, a projector is defined by *two spaces*, $\mathcal{M} = \ker(P)$ and $\mathcal{S} = \operatorname{ran}(P)$. We say that the projector $P$ is *onto* $\mathcal{M}$,

$$Px \in \mathcal{M},$$

and *along (parallel)* to $\mathcal{S}$

$$x - Px \in \mathcal{S}.$$

This point of view is based on two spaces of (in general) different dimensions. When we switch to the orthogonal complement $\mathcal{L} = \mathcal{S}^\perp$, we obtain a description involving two spaces of the *same dimension*:

$$Px \in \mathcal{M}, \quad x - Px \perp \mathcal{L}.$$

We now say that the projector is onto $\mathcal{M}$ and *orthogonal* to $\mathcal{L}$. We can write down the projector in matrix form by choosing bases:

$$\mathcal{M} = \operatorname{span}(M), \quad \mathcal{L} = \operatorname{span}(L).$$

The matrix representation of $P$, if it is defined, is given by

$$P = M(L^H M)^{-1} L^H.$$

A projector is defined iff no vector in $\mathcal{M}$ is orthogonal to $\mathcal{L}$. When the spaces $\mathcal{M}$ and $\mathcal{L}$ are different, we talk of an *oblique projector*. A better choice of bases would be biorthogonal (adjoint) bases,

$$L^H M = I.$$

Then the oblique projector takes the simple form $P = ML^H$. When the spaces are equal and we have chosen an orthogonal basis,

$$M^H M = I,$$

we name the projector an *orthogonal projector*. An orthogonal projector is Hermitian, since $P = MM^H$. In contrast to an oblique projector, an orthogonal projector always exists, because no vector can be orthogonal to itself.

## 1.7 Miscellaneous

In this section we collect some material that is related to the field of Krylov methods and paves way for some of the results. Krylov methods are related to polynomials, more precise, *infinite* precision Krylov subspace methods may be defined using polynomials. Thus, we give some basic facts on polynomials.

Like Krylov methods, polynomials, especially *orthogonal* polynomials, or more general, *orthogonal functions*, are closely related to three-term recurrences, i.e., to finite difference equations. One of the most interesting phenomena, the deviation of a short-term recurrence from the infinite precision counterpart can already be observed in the simpler example of the so-called *Bessel labyrinth*.

### 1.7.1 Polynomials

The set of all polynomials is a linear vector space which we denote by $\mathbb{P}$. We heavily use three finite dimensional subsets of $\mathbb{P}$. First, we define the subspace of all polynomials of degree less $k$, denoted by $\mathbb{P}_{k-1}$. In context of Krylov methods for the solution of the algebraic eigenproblem we minimise certain expressions over the

set $\mathbb{P}_{k-1}^{k-1}$, the set of *monic polynomials* of degree less $k$. In context of the solution of linear systems this role is played by the set $\mathbb{P}_{k-1}^{0}$, the set of polynomials with constant term equal to one, $c_0 = 1$. These sets are no subspaces.

For simplicity we say that the zero polynomial, $p(\lambda) = 0$ has *all* values as root with infinite multiplicity. The degree of the zero polynomial is minus infinity. The *characteristic polynomial* of a matrix,

$$\chi(\lambda) = \chi_A(\lambda) \equiv \det(\lambda I - A),$$

is defined such that $\chi_A \in \mathbb{P}_n^n$. By Cayley-Hamilton $\chi_A(A) = 0$. The smallest polynomial with the same feature is the *minimal polynomial* of the matrix. The minimal polynomial is given by

$$\mu = \mu_A(\lambda) \equiv \prod (\lambda - \lambda_i)^{k_i},$$

where the product is over all distinct eigenvalues and the power $k_i$ is the size of the largest Jordan block to eigenvalue $\lambda_i$.

Now, we switch to orthogonal polynomials. Orthogonal polynomials are given by the constraint

$$\langle p_k, p_j \rangle = \delta_{kj},$$

where $\langle \cdot, \cdot \rangle$ is some scalar product in $\mathbb{P}$. Frequently, the scalar product is an integral

$$\langle p_k, p_j \rangle = \int_{\mathbb{R}} w(x) p_k(x) p_j(x)\, dx,$$

where $w$ is a non-negative *weight function*. A discrete evaluation is formally written in the same form,

$$\langle p_k, p_j \rangle = \sum_{i=1}^{n} w(x_i) p_k(x_i) p_j(x_i) = \int_{\mathbb{R}} w(x) p_k(x) p_j(x)\, du(x).$$

The occurring integral

$$I[f] \;=\; \int_{\mathbb{R}} f(x)\, du(x) \;=\; \int_{\mathbb{R}} f\, du$$

is known as the Stieltjes integral (or Riemann-Stieltjes integral) with the integrating function $u$.

The existence of orthogonal polynomials is equivalent to the existence of a three-term recurrence. This follows from the representation of $xp_k(x)$ in the basis of the orthogonal polynomials,

$$xp_k(x) = \sum_{j=0}^{k+1} c_{kj} p_j(x).$$

Upon application of the inner product $\langle \cdot, p_j \rangle$ we obtain that

$$\langle xp_k(x), p_j(x) \rangle = c_{kj}$$

holds true. By our orthogonality assumption,

$$c_{kj} = \langle xp_k(x), p_j(x) \rangle = \langle p_k(x), xp_j(x) \rangle = 0 \quad \forall\, j < k - 1,$$

since $xp_j(x)$ is a polynomial of degree less or equal $j + 1$. This leaves us with a recurrence where at most *three coefficients* are non-zero, i.e., we obtain a three-term recurrence

$$c_{k,k+1} p_{k+1}(x) = r_k(x) = (x - c_{kk})\, p_k(x) - c_{k,k-1} p_{k-1}(x).$$

The coefficients can be computed as

$$\begin{aligned} c_{k,k-1} &= \langle x p_k(x), p_{k-1}(x) \rangle, \\ c_{kk} &= \langle x p_k(x), p_k(x) \rangle \qquad \text{and} \\ c_{k,k+1} &= \|r_k\| \equiv \sqrt{\langle r_k(x), r_k(x) \rangle}. \end{aligned}$$

This Gram-Schmidt procedure in the linear space of polynomials is known as *Stieltjes procedure*. We just remark that orthogonal polynomials have connections to Gauss quadrature, to continued fractions and to Padé approximation.

### 1.7.2 Finite Difference Equations

Finite difference equations were the source of inspiration for Leibniz to invent differential calculus in 1715. In the works of Casorati (1835–1890) many analogies between differential and difference calculus are recognised, the most interesting among them being the notion of the Casoratian or Casorati determinant, the difference calculus counterpart to the Wronskian or Wronski determinant (cf. [Jar]). Finite difference equations are closely related to the matrices arising in Krylov subspace methods.

### 1.7.3 Short-Term Recurrences: An Example

Three-term recurrences (and coupled two term recurrences) form the core of short-term Krylov subspace methods like the Lanczos method and CG. Originally, three-term recurrences arise in the efficient computation of orthogonal functions, mostly polynomials, such as the Chebychev, Legendre or Laguerre polynomials.

It is well known that the straight-forward computation of orthogonal functions need not be stable. As an example we consider the so-called *Bessel labyrinth* ("Besselscher Irrgarten", cf. [DH93], pages 167–173). The *Bessel functions* are special functions that fulfil the three-term recurrence

$$J_{k+1}(x) = \frac{2k}{x} J_k(x) - J_{k-1}(x), \quad k \geq 1.$$

The *Neumann functions* fulfil the same three-term recurrence, and small rounding errors introduced in the computation lead to a deviation from the Bessel part to the Neumann part and vice versa. This can be seen in figure 1.2.

The given three-term recurrence is evaluated with two given starting values $J_0(x)$ and $J_1(x)$ at the point $x = 2.13$ up to $J_{23}(x)$. We evaluate the recurrence backward with the last two values. The sketched forward-backward evaluation is carried out *repeatedly* several times. The 'exact' Bessel functions $J_k$ are *decreasing* with growing $k$ and the Neumann functions are rapidly *increasing*. In a forward sweep, small rounding errors are blown up such that the *dominant* Neumann functions overlap the Bessel functions. In a backward sweep the Bessel functions overlap the Neumann functions. This causes the plot shown in figure 1.2. We have adopted the figure and the example from the book by Deuflhard and Hohmann, where the starting values and a detailed error analysis can be found (cf. [DH93], pages 167–169).

To understand the problem in full generality, we consider *all* solutions of the homogeneous three-term recurrence. The solution set is a subspace of all maps from $\mathbb{N}$ to $\mathbb{R}$. The solution set is two-dimensional, since any solution of the recurrence is determined *uniquely* by *two* starting values. In the example, a set of independent solutions is given by the Bessel and Neumann functions.

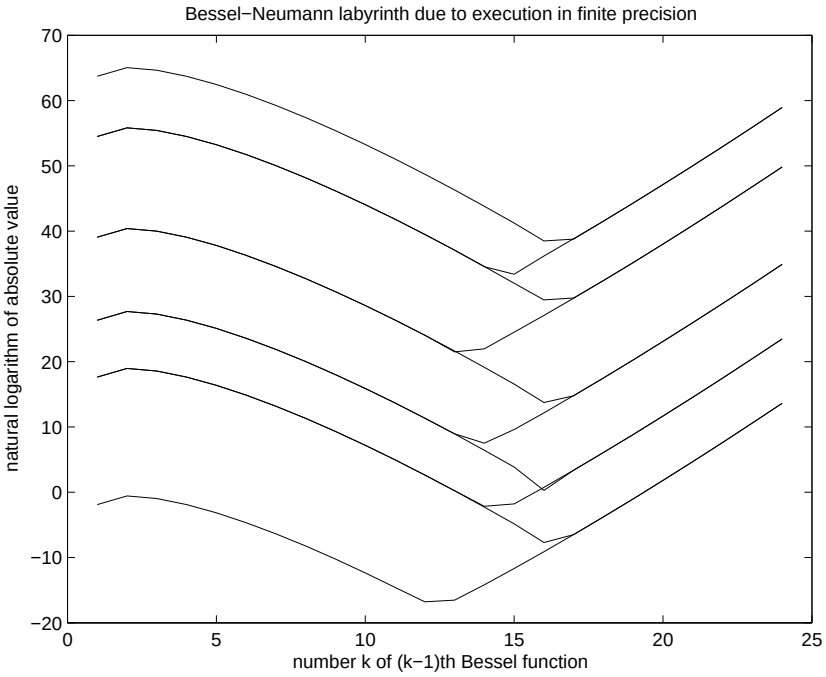


Figure 1.2: The Bessel labyrinth

We show how to construct two independent solutions. We think of the three-term recurrence in the form

$$xp_k(x) = a_k p_{k+1}(x) + b_k p_k(x) + a_{k-1} p_{k-1}(x), \qquad k \geq 1 \tag{1.28}$$

$$xp_0(x) = a_1 p_1(x) + b_0 p_0(x). \tag{1.29}$$

Equation (1.28) is merely a slightly re-written form of a three-term recurrence. The additional equation (1.29) fixes one degree of freedom. We suppose that all $a_k > 0$.

When we impose the starting value $p_0(x) = 1$, we have a unique solution $\{p_k\}_{k=0}^{\infty}$. The second solution will only fulfil equation (1.28). One such solution $\{r_k\}_{k=0}^{\infty}$ is determined by the initial conditions $r_0(x) = 0$ and $r_1(x) = a_0^{-1}$. The resulting polynomials could also have been obtained by equating the integral formula

$$r_k(x) = \int_{\mathbb{R}} \frac{p_k(x) - p_k(y)}{x - y}\, du(y).$$

The polynomials $\{r_k\}_{k=0}^{\infty}$ are the *associated polynomials*.

Whether two solutions to a three-term recurrence are *linear dependent* or not can be checked with the aid of the *Casorati determinant*. The Casorati determinant in *difference* calculus corresponds to the Wronski determinant in *differential* calculus. The Casorati determinant $C_{k,k+1}$ of $\{p_k\}_{k=0}^{\infty}$ and $\{r_k\}_{k=0}^{\infty}$ in step $k$ is defined by

$$C_{k,k+1}[p, r] = a_k \left(p_{k+1} r_k - p_k r_{k+1}\right).$$

Both sets $\{p_k\}_{k=0}^{\infty}$ and $\{r_k\}_{k=0}^{\infty}$ defined above fulfil equation (1.28). We multiply the equation for $p_k(x)$ with $r_k(y)$ and subtract the equation for $r_k(y)$ multiplied by $p_k(x)$. This results in

$$(x - y) p_k(x) r_k(y) = (a_k p_{k+1}(x) + b_k p_k(x) + a_{k-1} p_{k-1}(x)) r_k(y)$$

$$\begin{aligned} & - (a_k r_{k+1}(y) + b_k r_k(y) + a_{k-1} r_{k-1}(y)) p_k(x) \\ = \; & a_k (p_{k+1}(x) r_k(y) - r_{k+1}(y) p_k(x)) \\ - \; & a_{k-1} (p_k(x) r_{k-1}(y) - r_k(y) p_{k-1}(x)). \end{aligned}$$

Setting $y = x$ proves that the Casorati determinant does not depend on $k$ and is given by $C_{k,k+1}[p, r] = -1$. Replacing $r_k$ by $p_k$ and summing yields the famous *Christoffel-Darboux formula*

$$(x - y) \sum_{k=0}^{n-1} p_k(x) p_k(y) = a_{n-1} (p_n(x) p_{n-1}(y) - p_{n-1}(x) p_n(y)).$$

These results can be found in the lecture notes by Koelink for a SIAM Activity Group Summer School in Spain (cf. [Koe00], pages 9–11).

We merely remark that the coefficients occurring in equations (1.28) and (1.29) may be arranged in the form of a *Jacobi matrix*, such that the solution set $\{p_k\}_{k=0}^{\infty}$ corresponds (in some un-specified manner) to the whole matrix and the associated polynomial solution set $\{r_k\}_{k=0}^{\infty}$ corresponds (in the same un-specified manner) to the Jacobi matrix consisting of the previous one, where the first row and first column is left out. This has many similarities with the decision scheme invented by Cullum and Willoughby to identify spurious Ritz values in the finite precision symmetric Lanczos method.

The solution to a perturbed three-term recurrence, i.e., the solution of an *inhomogeneous* three-term recurrence can be computed using the convolution with a *discrete Green function*. The discrete Green function is defined similarly to its differential calculus analogue. The Kronecker delta $\delta_{jk}$ plays the role of a *discrete delta distribution*.

We think of a three-term recurrence for a fixed value $x$, and write the *general inhomogeneous recurrence* in the form

$$p_k = a_k p_{k-1} + b_k p_{k-2} + c_k, \qquad k > 1, \quad b_k \neq 0. \tag{1.30}$$

The discrete Green function is defined by

$$G(j, k) \equiv \begin{cases} G^-(j, k), & \text{if} \quad k \geq j \\ G^+(j, k), & \text{if} \quad k \leq j \end{cases},$$

where $G^-(j, k)$ and $G^+(j, k)$ are the solutions of the special inhomogeneous three-term recurrences

$$\begin{aligned} G^-(j, k) - a_k G^-(j, k-1) - b_k G^-(j, k-2) &= \delta_{jk} \\ G^+(j, k) - a_k G^+(j, k-1) - b_k G^+(j, k-2) &= -b_k \delta_{j,k-2}. \end{aligned}$$

The solution to the inhomogeneous equation (1.30) with starting values $p_0 = c_0$ and $p_1 = c_1$ is then given by the convolution

$$p_k = \sum_{j=0}^{k} c_j G(j, k) = \sum_{j=0}^{k} c_j G^-(j, k), \qquad k \in \mathbb{N}$$

with the Green function. This corresponds to the infinite dimensional (differential calculus) case.

Now we consider *perturbed* three-term recurrences. Let the perturbed recurrence be given by (relatively) perturbed starting values

$$\tilde{p}_0 = p_0(1 + \theta_0), \quad \tilde{p}_1 = p_1(1 + \theta_1)$$

and (relatively) perturbed coefficients

$$\tilde{a}_k = a_k(1+\alpha_k), \quad \tilde{b}_k = b_k(1+\beta_k), \quad k > 1.$$

Then the absolute error $\Delta p_k \equiv \tilde{p}_k - p_k$ fulfils the inhomogeneous three-term recurrence

$$\Delta p_k = a_k \Delta p_{k-1} + b_k \Delta p_{k-2} + E_k, \quad k > 1$$

where the error terms $E_k$ are defined by

$$E_k \equiv \alpha_k a_k \tilde{p}_{k-1} + \beta_k b_k \tilde{p}_{k-2} \approx \alpha_k a_k p_{k-1} + \beta_k b_k p_{k-2}.$$

The *absolute error* of a three-term recurrence can be computed by the convolution of the Green function with the *error terms*, i.e., by

$$\Delta p_k = \sum_{j=0}^{k} E_j G(j,k).$$

The discrete Green function gives information on the *absolute condition number* of the three-term recurrence subject to *relative perturbations*.

The relative errors $\theta_k \equiv \Delta p_k / p_k$ fulfil the inhomogeneous three-term recurrence

$$\theta_k = \frac{a_k p_{k-1}}{p_k}\theta_{k-1} + \frac{b_k p_{k-2}}{p_k}\theta_{k-2} + \epsilon_k, \quad k > 1$$

where the relative error terms $\epsilon_k$ are given by

$$\epsilon_k = \frac{E_k}{p_k} \approx \alpha_k \frac{a_k p_{k-1}}{p_k} + \beta_k \frac{b_k p_{k-2}}{p_k}.$$

Thus, we can express the relative error in the form of a convolution

$$\delta_k = \sum_{j=0}^{k} \epsilon_k R(j,k)$$

with some sort of 'relative' discrete Green function

$$R(j,k) = \frac{p_j}{p_k} G(j,k).$$

The 'relative' discrete Green function gives information on the *relative condition number* of the three-term recurrence subject to *relative perturbations*.

The solutions can be distinguished into *minimal* solutions and *dominant* solutions. A solution $\{p_k\}_{k=0}^{\infty}$ is called minimal, if

$$\lim_{k\to\infty} \frac{p_k}{q_k} = 0 \quad \forall\ q_k \ \text{ independent of } \ p_k.$$

All other independent solutions are termed dominant solutions. These dominant solutions can be computed stable by the three-term recurrence. The minimal solutions can not be computed in a stable way using the three-term recurrence (at least not using the straight-forward approach). The convergence of Krylov methods is linked to *minimal* solutions. The trick that *does* work for simple three-term recurrences (i.e., those where the coefficients are known *a priori*, in contrast to the coefficients in Krylov subspace methods, whose are computed *whilst executing the algorithm*) is known as *Miller's method* and is based on a *backward* evaluation of the three-term recurrence.

The definition of the discrete Green function and its use to express the absolute and relative errors of a perturbed three-term recurrence is an excerpt from (the German edition of) the book by Deuflhard and Hohmann (cf. [DH93], pages 164–166, 169–171).

# Chapter 2

# Krylov Subspace Methods and Matrix Structure

Krylov methods are iterative methods. In every step they expand a basis and corresponding matrix representation. These matrices have a nested structure. In most cases the old matrix will be a submatrix of the new one.

Krylov methods are iterative eigenproblem and linear system solvers. We have to ask ourselves how a bordering of a matrix changes intrinsic properties like eigenvalues and eigenvectors.

Krylov methods are a subset of the set of iterative methods. The matrix representations are restricted to the class of Hessenberg matrices. The matrix representations of the so-called short-term Krylov methods are even restricted to the class of tridiagonal matrices.

Execution in finite precision does not destroy the structure of the matrix representations. So we have a connection between Krylov methods and matrix structure. Knowledge on matrix structure is knowledge on (finite precision) Krylov methods and vice versa.

In this chapter we derive results which are necessary for our approach. To simplify and to shorten, we always examine the most general setting. We start at general matrices and stop at symmetric tridiagonals.

At the same time we aim at justifying part of mathematical folklore on the size of eigenvector entries and on the shape of eigenvectors.

We remark that various results can be extended to the classes of block Hessenberg, block tridiagonal and banded matrices.

## 2.1 An Algebraic Identity

In this section we pave the way for a variety of nice results that enlighten the relations between eigenvalues of a matrix and of its principal submatrices. Our approach is based on the adjugate and on the Jordan normal form.
Let $A \in \mathbb{K}^{n\times n}$. The adjugate, or (classical) adjoint, of $A$ is defined as the matrix of cofactors

$$B \equiv \operatorname{adj} A \quad \Leftrightarrow \quad b_{ij} \equiv (-1)^{i+j} \det A_{ji}.$$

Laplace expansion proves the important relation

$$\operatorname{adj} A\, A = A \operatorname{adj} A = \det A\, I. \tag{2.1}$$

We remark that the adjugate is defined by rational functions and exists even when $A$ is singular. The adjugate is more stable than the inverse, i.e., its condition number

in 2-norm is given by the ratio of the *second* smallest and largest singular value (cf. [Ste98]). It is well-known that the adjugate is connected to compound matrices.

Let the Jordan decomposition of $A$ been given by

$$AV = VJ_\Lambda \quad \Leftrightarrow \quad A = VJ_\Lambda V^{-1}.$$

We need to access the rows of $V^{-1}$ later on, and since these are conjugated left eigenvectors, we define the matrix

$$\hat{V}^H \equiv V^{-1}, \quad \text{i.e.,} \quad \hat{V} \equiv V^{-H}.$$

With this matrix we assure validity of the two relations

$$\hat{V}^H A = J_\Lambda \hat{V}^H \quad \text{and} \quad \hat{V}^H V = V^H \hat{V} = I.$$

In other words, the columns of $\hat{V}$ are left eigenvectors, scaled such that they form a *bi-orthonormal set* to the right eigenvectors of $A$.

Let $\lambda$ be not contained in the spectrum of $A$. Then the inverse of $\lambda I - A$ can be expressed in terms of the Jordan decomposition as

$$(\lambda I - A)^{-1} = VJ_{\lambda-\Lambda}^{-1}V^{-1} = VJ_{\lambda-\Lambda}^{-1}\hat{V}^H.$$

We use the Jordan normal form and equation (2.1) to express the adjugate of $\lambda I - A$,

$$\begin{aligned} \operatorname{adj}(\lambda I - A) &= \det(\lambda I - A)(\lambda I - A)^{-1} \\ &= \chi_A(\lambda)(\lambda I - A)^{-1} \\ &= V\left[\chi_A(\lambda)(\lambda I - J_\Lambda)^{-1}\right]\hat{V}^H. \end{aligned} \tag{2.2}$$

We take a closer look at the matrix in the middle. First we consider $(\lambda I - J_\Lambda)^{-1}$. The structure for one single Jordan block $J_{\lambda_i}$ of size $k$ is given by

$$(\lambda I - J_{\lambda_i})^{-1} = E_i \equiv \begin{pmatrix} (\lambda-\lambda_i)^{-1} & (\lambda-\lambda_i)^{-2} & \dots & (\lambda-\lambda_i)^{-k} \\ & (\lambda-\lambda_i)^{-1} & & \\ & & \ddots & \vdots \\ & & & (\lambda-\lambda_i)^{-1} \end{pmatrix}.$$

Let $S$ denote the sign matrix

$$S = S^{-1} = \begin{pmatrix} 1 & & & & \\ & -1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & (-1)^{n+1} \end{pmatrix}.$$

We express the adjugate with the aid of compound matrices,

$$\operatorname{adj} A \equiv SC_{n-1}(A^T)S.$$

Define $E \equiv \oplus_i E_i$. We transform equation (2.2) by moving the sign matrices,

$$\begin{aligned} P \equiv C_{n-1}(\lambda I - A^T) &= (SV)\,G\,(\hat{V}^H S) \\ &\equiv (SV)\,[\chi_A(\lambda)\,E]\,(\hat{V}^H S). \end{aligned} \tag{2.3}$$

The elements of the compound matrix $P$ are *polynomials* in $\lambda$ of the form

$$p_{ij} = p_{ij}(\lambda; A) \equiv \det L_{ji}, \quad \text{where} \quad L \equiv \lambda I - A. \tag{2.4}$$

The elements of $G$ are obviously given by *rational functions* in $\lambda$, since

$$G = \chi_A(\lambda) \cdot (\oplus_i E_i).$$

Many terms cancel, and the elements of $G$ turn out to be *polynomials* in $\lambda$.

The algebraic identity (2.3) relates polynomials of submatrices to polynomials of the matrix and left and right eigenvectors.

We remark that only the diagonal elements of the left-hand side are characteristic polynomials, of principal submatrices. We consider two interpretations of equation (2.3).

When $A$ has a multiple eigenvalue, one can raise the question whether a principal submatrix inherits this eigenvalue. We derive lower bounds on the algebraic multiplicity with the aid of equation (2.3).

Suppose $\lambda_1$ is fixed. Let $\alpha_j$ denote the algebraic multiplicity of eigenvalue $\lambda_j$. The entries inside the upper triangular part of the first Jordan blocks without sign matrices are given by

$$\chi_A(\lambda)(\lambda - \lambda_1)^{-\pi} = \prod_i (\lambda - \lambda_i)^{\alpha_i'}, \tag{2.5}$$

where

$$\alpha_i' = \begin{cases} \alpha_i & i \neq 1 \\ \alpha_1 - \pi & i = 1 \end{cases}.$$

The index $\pi$ is determined by the position in the matrix, i.e., we have $\pi = j - i + 1$ for position $(i, j)$. We determine the matrix position with $\pi$ maximal, and divide equation (2.3) by the factor

$$(\lambda - \lambda_1)^{\alpha_1 - \pi}.$$

After computing the limes $\lambda \to \lambda_1$ at least one element in the resulting matrix has non-zero value $c$ defined by

$$\chi_A^{(\alpha_1)}(\lambda_1) = \prod_{i \neq 1} (\lambda_1 - \lambda_i)^{\alpha_i} \equiv c.$$

The remaining entries are zero. All non-zero entries are in the $\pi - 1$th diagonal. The number of non-zero entries corresponds to the number of Jordan blocks that have maximal size given by $\pi$.

We have proven validity of the following theorem:

**Theorem 2.1** *Let $A \in \mathbb{K}^{n \times n}$. Let the Jordan decomposition of $A$ be given by $A = VJ_\Lambda V^{-1} \equiv VJ_\Lambda \hat{V}^H$. Suppose that the Jordan blocks are sorted first by eigenvalues and then by descending block size. Let $c = \chi_A^{(\alpha_1)}(\lambda_1)$, where $\alpha_1$ denotes the algebraic multiplicity of $\lambda_1$. Let $\pi$ denote the size of the first Jordan block. Let $P$ be the transposed matrix of minors of $\lambda I - A$.*

*Then the following relation holds true:*

$$\lim_{\lambda \to \lambda_1} (\lambda - \lambda_1)^{\pi - \alpha_1} P = (SV) \begin{pmatrix} M & & & \\ & \ddots & & \\ & & M & \\ & & & \mathbf{O} \end{pmatrix} (\hat{V}^H S), \tag{2.6}$$

*where*

$$M = \begin{pmatrix} 0 & \cdots & 0 & c \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & \ddots & \\ & & & \pm 1 \end{pmatrix},$$

*and $\mathbf{O}$ is a zero matrix of appropriate size.*

The number of blocks $M$ is the number of Jordan blocks of $\lambda_1$ of maximal size. The other Jordan blocks of $\lambda_1$ do not play a role in this equation.

The right-hand side of (2.6) is well-defined, thus the left-hand side is well-defined and the polynomial

$$(\lambda - \lambda_1)^{\alpha_1 - \pi} \;\Big|\; P$$

divides *all* entries of $P$. Depending on the right-hand side the factor may have higher multiplicity.

The entries interpreted as polynomials in $\lambda$ have $\lambda_1$ as root with multiplicity greater equal $\alpha_1 - \pi$.

**Remark 2.2** As interesting byproduct we note that the polynomials (2.4) in certain cases *must* have a multiple root $\lambda$. This is of general interest even only the diagonal elements are characteristic polynomials.

The polynomials (2.4) can be used to make propositions on the Jordan normal form based on the structure of $A$. In some cases we can restrict the class of possible Jordan normal forms to a subclass or in other cases exclude a subclass.

For certain structured matrices (for example many zeros) the degrees of the polynomials can be given explicitly. This observation is based on the Laplace expansion or the Leibniz formula for the determinant, or the representation of the determinant as a double sum of minors.

The next theorem is based on the observation that the $i$th diagonal element of $P$ is the characteristic polynomial $\chi_{A_i}(\lambda)$ of the $i$th principal submatrix of $A$ of size $n-1$.

**Theorem 2.3** *Let $A \in \mathbb{K}^n$. Let $\lambda$ be an eigenvalue of $A$ with $k$ Jordan blocks of sizes $n_j, j \in \underline{k}$.*

*Then all principal submatrices $A_i$, $i \in \underline{n}$ have $\lambda$ as an eigenvalue of algebraic multiplicity at least*

$$n_{\min} \equiv \left( \sum_{j=1}^{k} n_j \right) - n_{\max},$$

*where $n_{\max} = \max_j(n_j)$ is the size of the largest Jordan block.*

*Suppose we know the locations of the zero components of the eigenvectors. When some of components of the left and right eigenvectors are zero, the multiplicity of some eigenvalues of $A_i$ may be higher. If on the other hand the multiplicity is larger than predicted, the vectors must have zero components.*

We observe that the multiplicity is not only determined by the sizes and numbers of the Jordan blocks, but can be higher depending on possible zeros in the eigenvectors and principal vectors.

We now restrict ourselves to the case of no zero elements in the eigenvectors and principal vectors, i.e., we take a closer look at the generic case.

The algebraic multiplicity of an inherited eigenvalue with algebraic multiplicity $\alpha = \alpha(\lambda)$ and geometric multiplicity $\gamma = \gamma(\lambda)$ is given by $\alpha - \pi$, where $\pi$ is the size of the largest block in the Jordan decomposition.

The combinatorial evaluation of the

$$\binom{\alpha}{\gamma}$$

possible partitions of the eigenspace into Jordan blocks proves that when we restrict ourselves to the case that we only know the algebraic and geometric multiplicity

of an eigenvalue, we can only show that the algebraic multiplicity for a principal submatrix is at least

$$m_{\min} = \alpha - (\alpha - \gamma + 1) = \gamma - 1.$$

We formulate this result as

**Theorem 2.4** *Let $A \in \mathbb{K}^{n\times n}$. Let $\lambda$ be an eigenvalue of $A$ with geometric multiplicity $\gamma$.*

*Then all principal submatrices $A_i$, $i \in \underline{n}$ have $\lambda$ as an eigenvalue of algebraic multiplicity at least*

$$m_{\min} = \gamma - 1.$$

*Additional knowledge of the algebraic multiplicity can not be used to sharpen the bound any further.*

The intimate connection stated in Theorem 2.3 between zeros eigenvector components and multiple eigenvalues of submatrices can be extended to non-zero eigenvector components. In the next section we derive formulae shedding light on relations between eigenvalues and eigenvectors.

## 2.2 Eigenvalue – Eigenvector Relations

We have shown that some eigenvalues of $A$ are eigenvalues of principal submatrices. The proof rests upon equation (2.6). Up to now we neglected the non-zero eigenvector components. In this section we keep track of all eigenvector components.

In the sequel we will heavily use the notation

$$\hat{V} = V^{-H} \quad \text{i.e.,} \quad \hat{V}^H V = V^H \hat{V} = I.$$

We also access the entries of the left eigenvectors. For this reason we introduce the shorthand notation $\check{v} \equiv \overline{\hat{v}}$. This may be memorised as reflection on the real axis, turning *hat* to *vee* and vice versa. This notation corresponds to the matrix definition

$$\check{V} = \overline{\hat{V}} = \overline{V^{-H}} = V^{-T}, \quad \text{i.e.,} \quad \check{V}^T V = V^T \check{V} = I.$$

As consequence of (2.6) we have to consider a sum of rank one matrices which are built of the left and right eigenvectors. We remark that no principal vectors can occur in these equations. For ease of understanding we refer to the following clarifying picture.

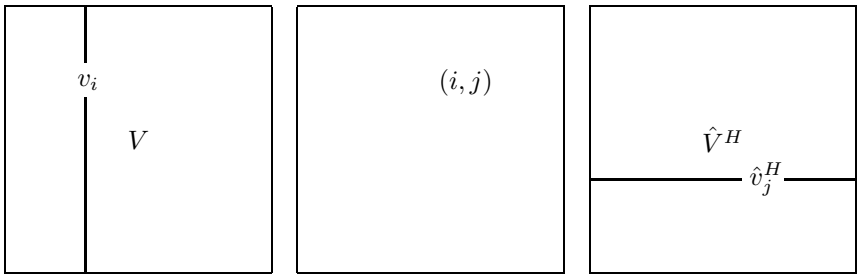


Figure 2.1: How vectors are selected from the eigensystem

We will derive a variety of results from equation (2.6). As first example we re-derive the well-known result (cf. [Tho66, TM68, CW80]) for a simple eigenvalue.

**Example 2.5 (simple eigenvalue)** Let $A \in \mathbb{K}^{n\times n}$. Let $\lambda_l$ be a simple eigenvalue of $A$. Let $\hat{v}_l^H$ and $v_l$ be the corresponding left and right eigenvectors with $\hat{v}_l^H v_l = 1$. Then the well-known relation

$$\begin{aligned} \operatorname{adj}(\lambda_l I - A) &= \chi_A'(\lambda_l) v_l \hat{v}_l^H \\ &= \prod_{s\neq l} (\lambda_l - \lambda_s)\, v_l \hat{v}_l^H \end{aligned} \tag{2.7}$$

holds true. This result could have been obtained using only equation (2.1) and the definition of the trace of a matrix.

This first example is the basis for more results. We multiply equation (2.7) from the left by $e_i^T$ and from the right by $e_i$ to obtain another well-known eigenvalue-eigenvector relation.

**Example 2.6 (same index)** Let $A \in \mathbb{K}^{n\times n}$. Let $\lambda_l$ be a simple eigenvalue of $A$. Let $\hat{v}_l^H$ and $v_l$ be the corresponding left and right eigenvectors with $\hat{v}_l^H v_l = 1$. Define $\nu_{s,i} \equiv \lambda_s(A_i)$. Then

$$v_{il}\check{v}_{il} = \frac{\prod_{\nu_{s,i}} (\lambda_l - \nu_{s,i})}{\prod_{s\neq l} (\lambda_l - \lambda_s)} = \frac{\prod_{\nu_{s,i}} (\lambda_l - \nu_{s,i})}{\prod_{\lambda_s\neq\lambda_l} (\lambda_l - \lambda_s)}$$

holds true. We observe that in the non-normal case the condition of the eigenvalue comes into play. In the normal case we can express the sizes of the eigenvector elements using only the eigenvalues of the matrix and some principal submatrices,

$$|v_{il}|^2 = \frac{\prod_{\nu_{s,i}} (\lambda_l - \nu_{s,i})}{\prod_{\lambda_s\neq\lambda_l} (\lambda_l - \lambda_s)}.$$

The next result uses different indices. It seems to be less known, maybe because of the unhandy polynomials (2.4).

**Example 2.7 (different indices)** Let $A \in \mathbb{K}^{n\times n}$. Let $\lambda_l$ be a simple eigenvalue of $A$. Let $\hat{v}_l^H$ and $v_l$ be the corresponding left and right eigenvectors with $\hat{v}_l^H v_l = 1$. Then

$$v_{jl}\check{v}_{il} = (-1)^{(j+i)} \frac{p_{ji}(\lambda_l; A)}{\prod_{s\neq l} (\lambda_l - \lambda_s)} = (-1)^{(j+i)} \frac{p_{ji}(\lambda_l; A)}{\prod_{\lambda_s\neq\lambda_l} (\lambda_l - \lambda_s)},$$

This example contains the polynomials $p_{ji}(\lambda_l; A)$.

Example 2.7 shows that it is possible to construct the eigenvector components iteratively. The computation involves the polynomials (2.4). A special case where this makes sense will be considered later on.

The next theorem is important for Hessenberg and tridiagonal matrices and applies to all non-derogatory eigenvalues.

**Theorem 2.8** *Let $A \in \mathbb{K}^{n\times n}$. Let $\lambda_l = \lambda_{l+1} = \ldots = \lambda_{l+k}$ be a geometrically simple eigenvalue of $A$. Let $k+1$ be the algebraic multiplicity of $\lambda$. Let $\hat{v}_{l+k}^H$ and $v_l$ be the corresponding left and right eigenvectors with appropriate normalisation. Then*

$$v_{jl}\check{v}_{i,l+k} = (-1)^{(j+i)} \frac{p_{ji}(\lambda_l; A)}{\prod_{\lambda_s\neq\lambda_l} (\lambda_l - \lambda_s)}$$

*holds true. Again the polynomials $p_{ji}(\lambda_l; A)$ are involved.*

*This setting matches every eigenvalue of non-derogatory $A$.*

**Remark 2.9** In the generic case all matrices are non-derogatory. This can been seen when comparing the dimension of all non-derogatory matrices with the dimension of the derogatory ones using Schur decomposition (cf. [Dem]).

If the matrix is diagonalisable (which is easy to see in the normal case, hard to see if the eigenvalues are merely separated, and mostly impossible to see if we have multiple eigenvalues) we can get rid of the messy polynomials.

**Example 2.10 (diagonalisable matrix)** Let $A \in \mathbb{K}^{n\times n}$ be diagonalisable. Suppose $\lambda_l = \lambda_{l+1} = \ldots = \lambda_{l+k}$ is a multiple eigenvalue with algebraic multiplicity $k+1$. Then

$$\sum_{t=l}^{k} v_{jt}\check{v}_{it} = (-1)^{(j+i)} \frac{p_{ji}^{(k)}(\lambda_l; A)}{\prod_{\lambda_s \neq \lambda_l}(\lambda_l - \lambda_s)}$$

holds true. We remark that the polynomial $p_{ji}(\lambda_l; A)$ has a $k$th root $\lambda_l$. Thus the $k$th derivative $p_{ji}^{(k)}(\lambda_l; A)$ is well-defined and non-zero.

The last theorem and example are just instances of a more general example with all Jordan blocks to eigenvalue $\lambda_l$ equal sized.

**Example 2.11 (equal sized Jordan blocks)** Let $A \in \mathbb{K}^{n\times n}$ have $p$ equal sized Jordan blocks to eigenvalue $\lambda_l = \lambda_{l+1} = \ldots = \lambda_{l+k}$ with algebraic multiplicity $k+1 = p \cdot q$. Then

$$\sum_{t=1}^{p} v_{j,l+(t-1)q}\check{v}_{i,l+tq-1} = (-1)^{(j+i)} \frac{p_{ji}^{(k+1-q)}(\lambda_l; A)}{\prod_{\lambda_s \neq \lambda_l}(\lambda_l - \lambda_s)}$$

holds true.

When $A$ is normal, the left and right eigenvectors are identical.

**Example 2.12 (the envelope)** Let $A \in \mathbb{K}^{n\times n}$ be normal. Let $\nu_{s,i} = \lambda_s(A_i)$. Then the *envelope of the eigenspace* associated with a multiple eigenvalue is given by

$$\sum_m |v_{im}|^2 = \frac{\prod_{\nu_{s,i} \neq \lambda_l}(\lambda_l - \nu_{s,i})}{\prod_{\lambda_s \neq \lambda_l}(\lambda_l - \lambda_s)}.$$

## 2.3 Hessenberg Matrices

An important class of matrices are so-called *upper Hessenberg matrices*. An upper Hessenberg matrix is a matrix having all elements equal to zero below the first sub-diagonal. Analogously lower Hessenberg matrices may be defined. We restrict ourselves to the case of upper Hessenberg matrices.

We term a Hessenberg matrix *unreduced* when the sub-diagonal compromises only of non-zero elements. A general Hessenberg matrix is the direct sum of unreduced Hessenberg matrices. We define $\mathcal{H}(m)$ to be the set of all unreduced upper Hessenberg matrices of size $m$, and the set of all unreduced upper Hessenberg matrices

$$\mathcal{H} = \cup_{m\in\mathbb{N}}\mathcal{H}(m).$$

Schur's Theorem states that every matrix $A \in \mathbb{K}^{n\times n}$ is unitarily similar to an upper triangular matrix,

$$Q^H A Q = R \equiv \Lambda + N.$$

Abel's Theorem implies that in general the Schur form *can not* be computed by means of a *finite sequence* of *algebraic transformations*, i.e., the Schur form can only be *approximated*.

Hessenberg matrices are close to triangular matrices. Indeed, every matrix $A \in \mathbb{K}^{n\times n}$ is unitary similar to the direct sum of unreduced Hessenberg matrices,

$$Q^H AQ = H \equiv \oplus_i H^{(i)}, \quad H^{(i)} \in \mathcal{H}.$$

This Hessenberg form may be computed by applying Householder reflectors or Givens rotators. This proves that a Hessenberg form *can* be computed by means of a *finite sequence* of *algebraic operations*.

This approach is norm-wise stable when implemented in finite precision due to the multiplicative use of orthogonal matrices.

We have gained a computable decomposition at the cost of one additional sub-diagonal. We remark that the Hessenberg form is not destroyed in a GR algorithm applied subsequently, and even accelerates computation.

Another approach is to compute $Q$ columnwise. This approach is known as Gram-Schmidt and is based on induction.

We re-state the unitarian similarity in an equivalent subspace form,

$$Q^H AQ = H \quad \Rightarrow \quad AQ = QH. \tag{2.8}$$

We consider the leading part of this subspace equation, such that the Hessenberg matrix is unreduced and $Q$ may be rectangular.

We interprete the last equation columnwise, i.e., we form (2.8) $e_k$ to obtain the relation

$$\begin{aligned} Aq_k &= \sum_{j=1}^{k+1} q_j h_{jk} \quad \Leftrightarrow \\ q_{k+1}h_{k+1,k} &= r_k \quad \equiv \quad Aq_k - \sum_{j=1}^{k} q_j h_{jk}. \end{aligned} \tag{2.9}$$

The *moments* $h_{jk}$ fulfil

$$h_{jk} = \langle q_j, Aq_k\rangle \quad \forall\ j \in \underline{k}, \qquad |h_{k+1,k}| = \|r_k\|. \tag{2.10}$$

We can compute the entries of $H$ and the columns of $Q$ using the relations (2.9+2.10). If we fix the sub-diagonal to be real positive, i.e., $h_{k+1,k} \equiv \|r_k\|$ we obtain $Q$ with $q$ as first column and an unreduced Hessenberg matrix $H$.

The process may break down before step $n$ in case a zero vector occurs. We can choose a new starting vector of unit length orthogonal to the computed columns of $Q$. This process is much less stable than the direct version. The next theorem states that the computed $Q$ (and $H$) is unique.

**Theorem 2.13 (Implicit $Q$ Theorem)** *Let $A \in \mathbb{K}^{n\times n}$. Let $q = q_1 \in \mathbb{K}^n$ have unit norm. This uniquely determines an orthonormal matrix $Q$ with $q_1$ as first column and $H \in \mathcal{H}$ with positive real sub-diagonal such that*

$$AQ = QH$$

*holds true. The* implicitly defined *$Q$ in general will be rectangular, $Q \in \mathbb{K}^{n\times m}$.*

**Proof.** The existence of orthonormal $Q$ and Hessenberg $H$ follows by induction. The starting vector $q_1$ is orthonormal by definition. Suppose the vectors $q_j$ computed by Gram-Schmidt are orthonormal for $j \in \underline{k}$. The vector $r_k$ and thus $q_{k+1}$ is orthogonal to the previous vectors, since

$$\langle q_j, r_k\rangle = \langle q_j, Aq_k\rangle - h_{jk} = 0.$$

The choice of $h_{k+1,k}$ ensures $q_{k+1}$ orthonormal. The resulting $H$ is unreduced upper Hessenberg.

The uniqueness follows by contradiction. Suppose

$$AQ = Q \begin{pmatrix} H & \star \\ 0 & \star \end{pmatrix}, \quad AP = P \begin{pmatrix} G & \star \\ 0 & \star \end{pmatrix}$$

with $H, G \in \mathcal{H}$ and $Q, P$ orthonormal holds true. Suppose w.l.o.g. that $H \in \mathcal{H}(m)$ and $G \in \mathcal{H}(l)$ with $m \leq l$. Then $W = P^H Q$ fulfils

$$\begin{pmatrix} G & \star \\ 0 & \star \end{pmatrix} W = P^H A Q = W \begin{pmatrix} H & \star \\ 0 & \star \end{pmatrix}.$$

The first column of $W$ is the first unit vector $e_1$. By induction on the columns of $W$ using

$$w_{k+1} h_{k+1,k} = \begin{pmatrix} G & \star \\ 0 & \star \end{pmatrix} w_k - \sum_{j=1}^{k} w_j h_{jk},$$

we observe that $W$ has the form

$$W = \begin{pmatrix} R & \star \\ 0 & \star \end{pmatrix}$$

with $R \in \mathbb{K}^{m \times m}$ upper triangular. By analogous induction on the columns of $W^H = Q^H P$, it follows that $W^H$ has the form

$$W^H = \begin{pmatrix} D & 0 & 0 \\ 0 & \tilde{R} & \star \\ 0 & 0 & \star \end{pmatrix}$$

with $D \in \mathbb{K}^{m \times m}$ diagonal and $\tilde{R} \in \mathbb{K}^{(m-l) \times (m-l)}$ upper triangular. $W$ is orthonormal and thus

$$\left| \begin{pmatrix} D & 0 \\ 0 & \tilde{R} \end{pmatrix} \right| = I_l.$$

We note that $l = m$, since $G$ is unreduced. The relation $DGD^{-1} = H$ together with the real positivity of the sub-diagonals proves uniqueness. □

In analogy to the *orthogonal projection* onto Hessenberg form we may consider an *oblique projection* onto Hessenberg form, namely

$$Q^{-1} A Q \equiv \hat{Q}^H A Q = H \quad \Rightarrow \quad AQ = QH.$$

There is no need for a *direct* method of this type. The extension of Gram-Schmidt to this setting makes perfect sense.

### 2.3.1 The Eigendecomposition

A Hessenberg matrix has a very special structure. This structure is reflected in the structure of the inverse and the eigendecomposition.

Unreduced Hessenberg matrices are non-derogatory, i.e., every eigenvalue corresponds to one single Jordan block. This becomes obvious when considering the rank deficiency of shifted $H \in \mathcal{H}(m)$,

$$\operatorname{rank}(H - \theta I) \geq m - 1,$$

since the first $m - 1$ columns are linearly independent. Whenever $H - \theta I$ becomes rank deficient, $\theta$ is an eigenvalue and has geometric multiplicity one.

We denote the Jordan decomposition of $H \in \mathcal{H}(m)$ by

$$HS = SJ_\Theta, \quad \text{i.e.,} \quad H = SJ_\Theta S^{-1}.$$

Like before we define

$$\hat{S}^H \equiv S^{-1}, \quad \text{i.e.,} \quad \hat{S} \equiv S^{-H}.$$

This may be considered as the unique left eigenvector matrix with the properties

$$\hat{S}^H H = J_\Theta \hat{S}^H, \qquad \hat{S}^H S = S^H \hat{S} = I.$$

The eigenvectors can not be arbitrary. This is proven in the next theorem.

**Theorem 2.14 (Non-zero components in eigenvectors)** *Let $H \in \mathcal{H}(m)$. Let $\theta$ be an eigenvalue of $H$. Let $s$ be the right eigenvector to eigenvalue $\theta$. Let $\hat{s}^H$ be the left eigenvector to eigenvalue $\theta$.*

*Then the last component of $s$ and the first component of $\hat{s}^H$ are both non-zero.*

**Proof.** The proof follows by contradiction. Suppose $s$ is a right eigenvector with last component zero,

$$Hs = s\theta, \quad e_m \perp s.$$

The last row of $H$ has to be orthogonal to $s$. Since $H$ is unreduced upper Hessenberg, this implies $e_{m-1} \perp s$. By induction all components are zero, and $s$ is the null-vector. The proof for the left eigenvector is analogous. □

From now on we focus on Hessenberg matrices related to another matrix $A$ by unitary or general similarity. We consider only the leading unreduced part of the subspace equations as constructed by Gram-Schmidt.

The eigenvalues and eigenvectors of the unreduced Hessenberg matrix are related to those of $A$ via

$$AQ = QH \quad \Rightarrow \quad AQS = QSJ_\Theta.$$

This indicates that we have computed *part* of the eigenvectors and eigenvalues of $A$. Let the eigendecomposition of $A$ been given by

$$A = VJ_\Lambda V^{-1} = VJ_\Lambda \hat{V}^H.$$

Suppose the Jordan decompositions of $A$ and $H$ can be chosen such that the vectors $QS$ are contained in the columns of $V$. Conditions when this is possible are stated in Theorem 1.22. We denote the submatrix compromising of the columns of a matrix $V$ in the index set $\mu$ by $V(\mu)$. Thus $QS = V(\mu)$ for some index set $\mu$ of cardinality $m$. We observe that

$$\hat{V}^H QS = \hat{V}^H V(\mu) = I(\mu)$$

This ensures that

$$\hat{V}^H Q = I(\mu)\hat{S}^H.$$

Finally we re-write $QS = V(\mu)$ as $S = \hat{Q}^H V(\mu)$. We have shown that in certain cases the relations

$$\hat{V}^H Q = I(\mu)\hat{S}^H \quad \text{and} \quad S = \hat{Q}^H V(\mu)$$

are valid. Interpreted componentwise we see that

$$\forall j \; \exists i \quad \hat{s}_j^H = \hat{v}_i^H Q \quad \text{and} \quad s_j = \hat{Q}^H v_i \tag{2.11}$$

holds true. In any case eigenvectors fulfilling these relations can be found.

We have proven that in some cases *in the iterative computation* of the unreduced Hessenberg matrix we *fix the leading entries of the eigenvectors* of ultimate $H$ in implicit form, i.e.,

$$\hat{s}_{kj} = \langle q_k, \hat{v}_i \rangle, \qquad s_{kj} = \langle \hat{q}_k, v_i \rangle. \tag{2.12}$$

Knowledge on *eigenvectors* of $A$ enables us to compute the leading entries of *eigenvectors* of $H$ without knowing the trailing part of $H$. In a later section we prove a similar result when we know the *eigenvalue* $\lambda$.

We need to access submatrices of Hessenberg matrices. For this reason we extend our notation.

### 2.3.2 Submatrices and Eigenvectors

We define the principal submatrices

$$\begin{aligned} H_{1:m} &\equiv H_m \equiv H, \qquad H \in \mathcal{H}(m) \\ H_{i:j} &\equiv H \begin{pmatrix} i, i+1, \ldots, j \\ i, i+1, \ldots, j \end{pmatrix}, \quad i \le j,\ i, j \in \underline{m}. \end{aligned}$$

We are interested in eigenvalues and eigenvectors. We denote the eigendecomposition of the submatrix $H_{i:j}$ by $H_{i:j} S_{i:j} = S_{i:j} J_{i:j}$. The left eigenvectors are defined by

$$\hat{S}_{i:j} \equiv S_{i:j}^{-H}, \quad \text{i.e.,} \quad \hat{S}_{i:j}^H H = J_{i:j} \hat{S}_{i:j}^H \quad \text{and} \quad \hat{S}_{i:j}^H S_{i:j} = S_{i:j}^H \hat{S}_{i:j} = I.$$

We can use prolonged versions of eigenvectors of submatrices as estimations for the true eigenvectors. The residuals of these approximate eigenvectors have a nice structure in two cases.

**Theorem 2.15** *Let $H \in \mathcal{H}(m)$. Split $H = H_{1:m}$ into*

$$H_{1:m} = \begin{pmatrix} H_{1:k} & \star \\ h_{k+1,k} e_1 e_k^T & H_{k+1:m} \end{pmatrix} \equiv \begin{pmatrix} H_{1:k} & \star \\ M & H_{k+1:m} \end{pmatrix}.$$

*Consider the* prolonged right eigenvectors of the leading part $H_{1:k}$ *as approximate right eigenvectors of $H$. The residual is given by the rank one matrix*

$$\begin{aligned} \begin{pmatrix} H_{1:k} & \star \\ M & H_{k+1:m} \end{pmatrix} \begin{pmatrix} S_{1:k} \\ 0 \end{pmatrix} &- \begin{pmatrix} S_{1:k} \\ 0 \end{pmatrix} J_{1:k} \\ &= h_{k+1,k} e_{k+1} e_k^T S_{1:k}. \end{aligned}$$

*The* prolonged left eigenvectors of the trailing part $H_{k+1:m}$ *have the rank-one residual*

$$\begin{aligned} \begin{pmatrix} 0 \\ \hat{S}_{k+1:m} \end{pmatrix}^H \begin{pmatrix} H_{1:k} & \star \\ M & H_{k+1:m} \end{pmatrix} &- J_{k+1:m} \begin{pmatrix} 0 \\ \hat{S}_{k+1:m} \end{pmatrix}^H \\ &= h_{k+1,k} \hat{S}_{k+1:m}^H e_1 e_k^T. \end{aligned}$$

Both cases are interesting for backward error analyses. When some part of the residual becomes small, it does not move any longer in backward sense. The caveat is that the condition of the eigenvectors might grow. Only when the condition growth is bounded, this result is useful.

We can extend the results slightly by applying the full eigenvector matrix from the other side. This results in the following.

**Theorem 2.16** *Let $H \in \mathcal{H}(m)$. Suppose $H$ partitioned as in the last theorem. Denote the eigendecompositions as before. The* inner product *of the* full left eigenvectors $\hat{S}^H_{1:m}$ *with the* prolonged leading right eigenvectors $S_{1:k}$ *fulfils*

$$\begin{aligned} J_{1:m}\hat{S}^H_{1:m}\begin{pmatrix} S_{1:k} \\ 0 \end{pmatrix} &- \hat{S}^H_{1:m}\begin{pmatrix} S_{1:k} \\ 0 \end{pmatrix} J_{1:k} \\ &= h_{k+1,k}\hat{S}^H_{1:m}e_{k+1}e_k^T S_{1:k}. \end{aligned}$$

*The* inner product *of the* prolonged left leading eigenvectors $\hat{S}^H_{k+1:m}$ *with the* full right eigenvectors $S_{1:m}$ *fulfils*

$$\begin{aligned} \begin{pmatrix} 0 \\ \hat{S}_{k+1:m} \end{pmatrix}^H S_{1:m}J_{1:m} &- J_{k+1:m}\begin{pmatrix} 0 \\ \hat{S}_{k+1:m} \end{pmatrix}^H S_{1:m} \\ &= h_{k+1,k}\hat{S}^H_{k+1:m}e_1e_k^T S_{1:m}. \end{aligned}$$

This theorem states in a compact form many useful eigenvalue - eigenvector relations based on submatrices. These relations clarify the dependencies between the quality of a prolonged eigenvector as approximation and the sizes of certain eigenvector components.

The distance between an eigenvalue $\theta_j^{(k)}$ of a leading submatrix $H_{1:k}$ and an eigenvalue $\theta_i^{(m)}$ of the full matrix $H_{1:m}$ in terms of eigenvector components for example is described by

$$\left(\theta_i^{(m)} - \theta_j^{(k)}\right)\left(\hat{s}_i^{(m)}\right)^H \begin{pmatrix} s_j^{(k)} \\ 0 \end{pmatrix} = h_{k+1,k}\check{s}_{k+1,i}^{(m)}s_{kj}^{(k)}.$$

There is some freedom in this relation. We might choose the eigenvalue of $H_m$ nearest to the eigenvalue of interest of $H_{1:k}$.

The next section is concerned with another approach to eigenvalue eigenvector relations.

### 2.3.3 Eigenvalue – Eigenvector Relations

We enhance the results on the connection between eigenvalues and eigenvectors obtained in the preceeding sections.

We have shown that there is a certain relation between eigenvalues and eigenvectors involving polynomials evaluated at eigenvalues.

To be more precise, let $H \in \mathcal{H}(m)$. Let $\theta$ be an eigenvalue of $H$ and $\hat{s}^H = \check{s}^T$ and $s$ the corresponding eigenvectors. Let the algebraic multiplicity of $\theta$ be $k+1$. Then we have shown in Theorem 2.8 that

$$(-1)^{j+i}\,\check{s}(i)s(j) = \frac{p_{ji}(\theta; H)}{\prod_{\theta_l \neq \theta}\theta - \theta_l}.$$

We adopted notation to Hessenberg matrices, $\hat{S} \equiv S^{-H}$ and $\check{S} \equiv S^{-T}$.

For Hessenberg matrices some polynomials $p_{ji}(\theta; H)$ can be evaluated explicitly. Whenever $i \leq j$ the new matrix inside the determinant is a block upper diagonal matrix whose determinant is simply the product of the determinants of the blocks, in this case the determinants of smaller Hessenberg matrices or upper triangular matrices.

We consider two important examples.

**Example 2.17 (omitting first row and last column)** Let $H \in \mathcal{H}(m)$. Let the matrix of polynomials $P$ be defined as in (2.4). Let $L = \theta I - H$. The polynomial

$p_{m1}$ is constant and can be evaluated as follows,

$$\begin{aligned} p_{m1}(\theta; H) &= \det L_{1m} \\ &= \begin{vmatrix} -h_{21} & \theta - h_{22} & \cdots & -h_{2,m-1} \\ 0 & -h_{32} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \theta - h_{m-1,m-1} \\ 0 & \cdots & 0 & -h_{m,m-1} \end{vmatrix} \\ &= \prod_{l=1}^{m-1} -h_{l+1,l} = (-1)^{m+1} \prod_{l=1}^{m-1} h_{l+1,l}. \end{aligned}$$

To be conforming with previous notation we used

$$(-1)^{j-i} = (-1)^{j-i} (-1)^{2i} = (-1)^{i+j}.$$

Whenever an upper triangular matrix occurs inside the determinant of the polynomial, the unknowns play no role. It turns out that we can evaluate explicitly half of the polynomials.

**Example 2.18 (row index less than column index)** Let $H \in \mathcal{H}(m)$. Let the matrix of polynomials $P$ be defined as in (2.4). Let $i \leq j$. Let $L = \theta I - H$. The polynomial $p_{ji}$ has degree $(i-1) + (m-j)$ and can be evaluated as follows,

$$\begin{aligned} p_{ji}(\theta; H) &= \det L_{ij} \\ &= \begin{vmatrix} \theta I - H_{1:i-1} & & \star \\ & R_{i+1:j-1} & \\ 0 & & \theta I - H_{j+1:m} \end{vmatrix} \\ &= \det(\theta I - H_{1:i-1}) \det(R_{i+1:j-1}) \det(\theta I - H_{j+1:m}) \\ &= (-1)^{i+j} \chi_{H_{1:i-1}}(\theta) \prod \operatorname{diag}(H_{i:j}, -1) \chi_{H_{j+1:m}}(\theta). \end{aligned}$$

Example 2.18 simplifies the relation between characteristic polynomials of submatrices, sub-diagonal elements and eigenvector entries.

**Theorem 2.19** *Let $H \in \mathcal{H}(m)$. Let $i \leq j$. Let $\theta$ be an eigenvalue of $H$ with multiplicity $k+1$. Let $s$ be the unique right eigenvector and $\hat{s}^H$ be the unique left eigenvector to eigenvalue $\theta$. Then*

$$\check{s}(i)s(j) = \left[ \frac{\chi_{H_{1:i-1}} \chi_{H_{j+1:m}}}{\chi^{(k+1)}_{H_{1:m}}}(\theta) \right] \prod_{l=i}^{j-1} h_{l+1,l} \tag{2.13}$$

*holds true.*

Among these relations of special interest is the case of index pairs $(i, m)$, $(1, m)$ and $(1, m)$, $(1, j)$. They can be used to derive relations between eigenvalues and a single eigenvector. The dependency on multiplicity is removed. The three index pairs of interest are

$$\check{s}(i)s(m) = \left[ \frac{\chi_{H_{1:i-1}}}{\chi^{(k+1)}_{H_{1:m}}}(\theta) \right] \prod_{l=i}^{m-1} h_{l+1,l}, \tag{2.14}$$

$$\check{s}(1)s(m) = \left[ \frac{1}{\chi^{(k+1)}_{H_{1:m}}}(\theta) \right] \prod_{l=1}^{m-1} h_{l+1,l}, \tag{2.15}$$

$$\check{s}(1)\, s(j) \quad = \quad \left[ \frac{\chi_{H_{j+1:m}}}{\chi^{(k+1)}_{H_{1:m}}}(\theta) \right] \prod_{l=1}^{j-1} h_{l+1,l}. \tag{2.16}$$

We remark that equation (2.15) provides a second proof that the last component of a right eigenvector and the first component of a left eigenvector are non-zero. The right-hand side is non-zero by assumption that $H$ is unreduced and is finite by construction. Thus the left hand side has to be non-zero.

Dividing equation (2.14) by equation (2.15) we obtain a useful characterisation of the *left eigenvector*.

**Theorem 2.20 (Construction of left eigenvectors)** *Let $H \in \mathcal{H}(m)$. Let $\theta$ be an eigenvalue of $H$. Then $\hat{s} = \overline{\check{s}}$ defined by non-zero $\check{s}(1)$ and the relations*

$$\frac{\check{s}(i)}{\check{s}(1)} = \frac{\chi_{H_{i-1}}(\theta)}{\prod_{l=1}^{i-1} h_{l+1,l}} \quad \forall\ i \in \underline{m},$$

*is (up to scaling) the unique left eigenvector of $H$ to eigenvalue $\theta$.*

Dividing equation (2.16) by equation (2.15) we obtain an analogous characterisation of the *right eigenvector*.

**Theorem 2.21 (Construction of right eigenvectors)** *Let $H \in \mathcal{H}(m)$. Let $\theta$ be an eigenvalue of $H$. Then $s$ defined by non-zero $s(m)$ and the relations*

$$\frac{s(j)}{s(m)} = \frac{\chi_{H_{j+1:m}}(\theta)}{\prod_{l=j+1}^{m} h_{l,l-1}} \quad \forall\ j \in \underline{m},$$

*is (up to scaling) the unique right eigenvector of $H$ to eigenvalue $\theta$.*

The last two theorems reflect the Hessenberg structure. Suppose the eigenvalue $\theta$ is given. Then the left eigenvector can be computed whilst computing the Hessenberg matrix column by column, whereas the right eigenvector depends from the beginning on on the whole matrix.

We apply the last two results to unreduced Hessenberg matrices that have been constructed by Gram-Schmidt.

Equation (2.11) shows the relations between components of eigenvectors of $H$ and eigenvectors of $A$. Inserted into the two theorems we obtain the following theorem:

**Theorem 2.22** *Let $A \in \mathbb{K}^{n\times n}$ fulfil the conditions of Theorem 1.22. Let $H \in \mathcal{H}(m)$. Let $AQ = QH$.*

*Then*

$$\langle \hat{v}_i, q_{k+1}\rangle = \frac{\chi_{H_{1:k}}(\lambda_i)}{\prod_{l=1}^{k} h_{l+1,l}} \langle \hat{v}_i, q_1\rangle \quad \text{and} \quad \langle v_i, \hat{q}_{k-1}\rangle = \frac{\chi_{H_{k:m}}(\lambda_i)}{\prod_{l=k}^{m} h_{l,l-1}} \langle v_i, \hat{q}_m\rangle$$

*hold true.*

Suppose $A$ is diagonalisable. We multiply the first set of quantities by the right eigenvectors,

$$\prod_{l=1}^{k+1} h_{l+1,l} v_i \langle \hat{v}_i, q_{k+1}\rangle = \chi_{H_k}(\lambda_i) v_i \langle \hat{v}_i, q_1\rangle.$$

After summing up the equations we obtain the shorter and more familiar expression

$$q_{k+1} \prod_{l=1}^{k+1} h_{l+1,l} = \chi_{H_k}(A) q_1.$$

We obtained an explicit expression for the recurrence vectors $q_j$.

### 2.3.4 Mathematical Folklore

This section is concerned with a bunch of observations and rules of thumb on Hessenberg matrices. More precisely, we consider the shape of the eigenvectors and the size of the eigenvector components.

What can be said about the absolute size of the entries of the eigenvectors? We first consider zero entries, and then switch to small entries.

When a zero component occurs in a left eigenvector $\hat{s}^H$, we have seen that the corresponding eigenvalue is also an eigenvalue of a smaller *leading* Hessenberg matrix. At the same time a zero in a right eigenvector implies that the corresponding eigenvalue is also an eigenvalue of the a smaller *trailing* Hessenberg matrix.

When switching to small components, we see that this remains true approximately. This has been considered in more detail in a previous section.

From now on we focus on the left eigenvectors. We have derived an explicit recurrence for the components of left eigenvectors, which uses only the leading part of the final Hessenberg matrix $H$ and the exact final eigenvalue $\theta$.

Suppose that $H$ is well-conditioned, i.e., close to normal. Suppose that $\theta$ is well-separated from the remaining part of the spectrum. Then the polynomials which form the basis for the recurrence are well-conditioned and prune to small perturbations in $\theta$.

In the generic case well-separated eigenvalues are well approximated by some eigenvalues of some submatrices. The preceeding considerations imply that the eigenvector components will be small when hitting the border of such a matrix.

The polynomial recurrences imply that the eigenvector components in the generic case will be distributed logarithmically. The convergence history is stored in the final eigenvector.

When we switch to clustered eigenvalues, the picture changes. We propose a dynamic interpretation. In the beginning none of the eigenvalues will be approximated well. When the first eigenvalue approximates the cluster, the eigenvector corresponding to the approximation $\theta$ will have small last components.

When the second eigenvalue becomes approximated, the polynomial will be very sensitive to small perturbations. Both eigenvectors will look nearly identical. In the beginning the size of the components will decrease until the polynomial becomes sensitive. Then the size of the components will start to increase again. When two approximations are close to the cluster, the polynomials become sensitive and so forth.

Also in this case the convergence will be stored in the eigenvector components. But in contrast to a simple eigenvalue, the point where the eigenvector has a maximum is the point where the leading submatrix will provide a good approximation to the cluster.

The same holds true for the backward interpretation and right eigenvectors.

## 2.4 Tridiagonal Matrices

Matrices that are upper *and* lower Hessenberg are tridiagonal matrices. All results obtained thus far apply to tridiagonal matrices.

The elements of tridiagonal matrices will be denoted by

$$T_k = \text{tridiag}(\beta, \alpha, \gamma) = \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{pmatrix}.$$

The dimension of $T$ will in general be given by $m$. We extend all notation developed for Hessenberg matrices to tridiagonals. In contrast to Hessenberg matrices we term a tridiagonal matrix *unreduced* when all elements along the sub- *and* super-diagonal are non-zero.

The eigendecomposition of general tridiagonal $T$ will be denoted by

$$TS = SJ_\Theta \quad \Leftrightarrow \quad T = SJ_\Theta S^{-1}.$$

In case of normal $T$ the eigenvector matrix is unitary. In case of diagonalisable $T$ the Jordan form is denoted by the diagonal matrix $\Theta$.

Like before we define

$$\hat{S} = S^{-H} \quad \text{with } \hat{S}^H T = J_\Theta \hat{S}^H \quad \text{and} \quad \hat{S}^H S = S^H \hat{S} = I$$

as special left eigenvector matrix.

We collect the knowledge on unreduced tridiagonals as corollary.

**Corollary 2.23** *Unreduced tridiagonals are upper and lower unreduced Hessenberg matrices. This implies the following:*

- *Tridiagonals are non-derogatory. This implies that* normal *tridiagonal matrices have* simple *eigenvalues.*
- *The last* and *first components of the unique eigenvectors are non-zero. When a component of an eigenvector s is zero, the trailing (leading) principal submatrix has the trailing (leading) part of s as eigenvector to the same eigenvalue.*
- *Knowing the final eigenvalue the eigenvectors can be computed with the aid of trailing* and *leading submatrices, i.e., we have forward* and *backward expressions for the eigenvector components.*

**Proof.** The results follow by application of Theorem 2.14, Theorem 2.20 and Theorem 2.21, since unreduced tridiagonals are upper and lower unreduced Hessenberg matrices. □

We considered the Gram-Schmidt process to construct a Hessenberg matrix $H$ and an orthonormal matrix $Q$. The so-called *two-sided Gram-Schmidt* process uses *two* sequences of vectors to project the matrix simultaneously onto an upper and a lower Hessenberg matrix, i.e., onto a tridiagonal matrix.

Two-sided Gram-Schmidt assumes the existence of a matrix $Q$ such that

$$Q^{-1}AQ = T \quad \Leftrightarrow \quad AQ = QT.$$

We define $\hat{Q}^H \equiv Q^{-1}$. We consider the leading part of $T$, i.e., we assume that $T = T_m$ is unreduced. We choose $\hat{q}_1$ and $q_1$ with $\langle \hat{q}_1, q_1 \rangle = 1$. We consider the iterations

$$\begin{aligned} q_{k+1}\beta_k &= r_k \equiv Aq_k - \alpha_k q_k - \gamma_{k-1} q_{k-1}, \\ \gamma_{k+1}\hat{q}_{k+1}^H &= \hat{r}_k^H \equiv \hat{q}_k^H A - \hat{q}_k^H \alpha_k - \hat{q}_{k-1}^H \beta_{k-1}, \\ &\quad \text{with} \quad \gamma_k \beta_k \equiv \langle \hat{r}_k, r_k \rangle. \end{aligned}$$

This iteration may fail to produce unreduced $T$. When $A$ and $T$ are self-adjoint this process collapses to a single recurrence and is always executable.

The residual of prolonged left and right eigenvectors has a nice structure when we use trailing or leading principal submatrices. A low rank structure occurs also when considering prolonged eigenvectors of principal submatrices *in the middle.*

Let $T_{i:j}$, $i \leq j$ be a principal submatrix somewhere in the middle. Let the eigendecomposition of $T_{i:j}$ be given by

$$T_{i:j} S_{i:j} = S_{i:j} J_{i:j}.$$

The residual when we apply the prolonged middle eigenvectors to $T_m$ is given by

$$\begin{pmatrix} T_{1:i-1} & \gamma_{i-1} e_{i-1} e_1^T & 0 \\ \beta_{i-1} e_1 e_{i-1}^T & T_{i:j} & \gamma_j e_{j-i+1} e_1^T \\ 0 & \beta_j e_1 e_{j-i+1}^T & T_{j+1:m} \end{pmatrix} \begin{pmatrix} 0 \\ S_{i:j} \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ S_{i:j} \\ 0 \end{pmatrix} J_{i:j}$$
$$= \gamma_{i-1} e_{i-1} e_1^T S_{i:j} + \beta_j e_{j-1} e_{j-i+1}^T S_{i:j}.$$

Essentially the same holds true for the prolonged left eigenvectors. We obtain in all cases a low-rank residual. In other words:

**Theorem 2.24** *Let $T = T_m \in \mathbb{K}^{m \times m}$ be unreduced tridiagonal. Let the eigendecomposition of $T$ and of a middle principal submatrix be defined as stated above.*

*The departure of the right partial eigensystem from the right exact one is given by the low-rank matrix*

$$\begin{aligned} T_m \begin{pmatrix} 0 \\ S_{i:j} \\ 0 \end{pmatrix} &- \begin{pmatrix} 0 \\ S_{i:j} \\ 0 \end{pmatrix} J_{i:j} \\ &= \gamma_{i-1} e_{i-1} e_1^T S_{i:j} + \beta_j e_{j-i+1} e_j^T S_{i:j}. \end{aligned}$$

*The departure of the left partial eigensystem form the left exact one is given by the low-rank matrix*

$$\begin{aligned} \begin{pmatrix} 0 \\ \hat{S}_{i:j} \\ 0 \end{pmatrix}^H T_m &- J_{i:j} \begin{pmatrix} 0 \\ \hat{S}_{i:j} \\ 0 \end{pmatrix}^H \\ &= \beta_{i-1} \hat{S}_{i:j}^H e_1 e_{i-1}^T + \gamma_j \hat{S}_{i:j}^H e_{j-i+1} e_{j+1}^T. \end{aligned}$$

*These results may be interpreted column-, row- and componentwise to give backward bounds on single eigenvector approximations.*

These results also give backward bounds on the eigensystem when the matrix is prolonged to either direction.

They can be used to describe the setting in case of multiple eigenvalues. The shape of the eigenvectors will reflect the number of eigenvalues close to each other. Together with the polynomial evaluations of the eigenvectors in terms of submatrices we observe that we can find a basis of eigenvectors that are non-zero only in a single region.

In the consideration of the residual of prolonged eigenvectors of Hessenberg matrices we obtained some useful eigenvector - eigenvalue relations. The complex relations stated in Theorem 2.24 multiplied from the other side by the full eigenvector matrix again result in eigenvalue - eigenvector relations. We obtain the following theorem.

**Theorem 2.25** *Let $T = T_m \in \mathbb{K}^{m \times m}$ be unreduced tridiagonal. Let $T_{i:j}$ be any principal submatrix with $i \leq j$. Let the eigendecomposition be given as $T_{i:j} S_{i:j} = J_{i:j} S_{i:j}$.*

*Then the following relations hold among the eigenvectors, the prolonged eigenvectors and the Jordan forms. First we state the relation in terms of the* full left

*eigenvector matrix and the* partial right *eigenvector matrix:*

$$\begin{aligned} & J_{1:m}\hat{S}^H_{1:m}\begin{pmatrix}0\\S_{i:j}\\0\end{pmatrix} - \hat{S}^H_{1:m}\begin{pmatrix}0\\S_{i:j}\\0\end{pmatrix}J_{i:j} \\ = \quad & \gamma_{i-1}\hat{S}^H_{1:m}e_{i-1}e_1^T S_{i:j} + \beta_j \hat{S}^H_{1:m}e_{j-i+1}e_j^T S_{i:j}. \end{aligned}$$

*The corresponding result involving the* partial left *eigenvector matrix and the* full right *eigenvector matrix is similar:*

$$\begin{aligned} & \begin{pmatrix}0\\\hat{S}_{i:j}\\0\end{pmatrix}^H S_{1:m}J_{1:m} - J_{i:j}\begin{pmatrix}0\\\hat{S}_{i:j}\\0\end{pmatrix}^H S_{1:m} \\ = \quad & \beta_{i-1}\hat{S}^H_{i:j}e_1 e_{i-1}^T S_{1:m} + \gamma_j \hat{S}^H_{i:j}e_{j-i+1}e_{j+1}^T S_{1:m}. \end{aligned}$$

*Again these relations may be interpreted colum-, row- and componentwise.*

### 2.4.1 Mathematical Folklore

The size of eigenvector components and the shape of eigenvectors is also of interest in case of tridiagonal matrices.

The residual bounds show that in case of small eigenvector components two matrices, a trailing and a leading have eigenvalues close to the corresponding eigenvalue. The polynomial representation together with the stability considerations of the polynomial evaluation ensures that there are eigenvectors of principal submatrices, that are candidates for eigenvectors with small residual.

Parlett was one of the researchers to find more rigid proofs. His work is restricted to the case of symmetric tridiagonals. He proved that it is sufficient to compute eigenvectors of some leading and trailing submatrices to obtain a nearly orthogonal basis for the subspace, and furthermore to have a small residual when these vectors are used as approximate eigenvectors.

He stated that it was better to use the bisectors as approximate eigenvectors. This is natural, since their shape is better matching with the eigenvector representation in terms of characteristic polynomials of leading, trailing and middle submatrices.

## 2.5 Rectangular Matrices

We compute candidates for invariant subspaces column by column. The basis vectors form rectangular matrices. In case of the methods using orthogonal bases the resulting matrices will have orthonormal columns, or at least locally orthonormal columns. Most methods result in basis vectors that have approximately unit length.

For reasons of completeness we state some basic results on relations between singular values and singular vectors that can easily be obtained.

### 2.5.1 Singular Value – Singular Vector Relations

We are interested in rectangular matrices $Q \in \mathbb{K}^{n\times k}$. Let the SVD of $Q$ be given as

$$Q = U\Sigma W^H, \qquad U \in \mathcal{U}(n), \quad \Sigma \in \mathbb{K}^{n\times k}, \quad W \in \mathcal{U}(k).$$

We denote the short SVD by

$$Q = U_0\Sigma_0 W^H, \qquad U_0 \in \mathbb{K}^{n\times k}, \quad \Sigma_0 \in \mathbb{K}^{k\times k}.$$

The short SVD is obtained by discarding the null-space.

Using the equation

$$\begin{pmatrix} 0 & Q \\ Q^H & 0 \end{pmatrix} \begin{pmatrix} U & U_0 \\ W_1 & -W \end{pmatrix} = \begin{pmatrix} U & U_0 \\ W_1 & -W \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & -\Sigma \end{pmatrix}$$

we could obtain relations between the eigenvalues, i.e., the singular values, and the eigenvectors, i.e., the left and right singular vectors. Here we have additionally defined the matrices

$$W_1 = \begin{pmatrix} W & 0 \end{pmatrix} \in \mathbb{K}^{k \times n} \quad \text{and} \quad \Sigma_1 = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{K}^{n \times n}.$$

This is possible and involved. There is a better approach which gives the same results.

Another way of embedding singular value decompositions into symmetric eigenvalue problems is given by

$$Q^H Q = W \Sigma^H U U^H \Sigma W^H = W \Sigma^2 W^H.$$

We set

$$A = Q^H Q, \quad V = W.$$

Since $A$ is selfadjoint, we observe that $\hat{V} = W$, and thus $\check{V} = \overline{W}$. The matrix of eigenvalues is given by $\Lambda = \Sigma^2$. Principal submatrices of $A$ are given by

$$A_j = \begin{pmatrix} Q_{1:j-1} & Q_{j+1:k} \end{pmatrix}^H \begin{pmatrix} Q_{1:j-1} & Q_{j+1:k} \end{pmatrix}$$

and correspond to erasing one column in $Q$. We denote the singular values of such $Q$ with $j$th column erased by $\sigma'_s$.

Proceeding this manner we obtain that for a simple singular value $\sigma_l$ the relation

$$|w_{jl}|^2 = \frac{\prod \sigma_l^2 - \sigma_s'^2}{\prod \sigma_l^2 - \sigma_s^2},$$

and in case of a multiple singular value the relation

$$\sum_t |w_{jt}|^2 = \frac{\prod_{\sigma'_s \neq \sigma_l} \sigma_l^2 - \sigma_s'^2}{\prod_{\sigma_s \neq \sigma_l} \sigma_l^2 - \sigma_s^2}$$

hold true. We do not obtain relations on changes in the left singular vectors, but they actually *do not change*. They change when we erase rows, and in this case we can use the other representation, namely

$$A = QQ^H = U \begin{pmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{pmatrix} U^H.$$

Here we have to be careful with the $n - k$ zero eigenvalues.

# Chapter 3

# Krylov Subspace Methods in Infinite Precision

In this chapter we re-derive the commonly known Krylov subspace methods. We introduce a general framework intended to fit the investigation of their finite precision behaviour.

Having defined simple and block Krylov subspaces we study the properties of different bases and representations. We introduce the fundamental notion of a Krylov subspace decomposition, which will form the basis for our unified approach. Then we turn attention to the actual methods.

We introduce Krylov methods divided according to their application. First we dicuss the methods for the algebraic eigenproblem. Then we focus on methods used in the solution process of sparse linear systems.

We restrict ourselves to the basic methods.

## 3.1 Krylov Subspaces

Krylov subspaces are named after Russian naval engineer Aleksei Nikolaevich Krylov (*1863 – †1945). In 1931 Krylov published a paper (cf. [Кры31]) on a method to compute the coefficients of the characteristic polynomial. In this paper Krylov subspaces were used for the first time.

For more information on Aleksei Nikolaevich Krylov see (cf. [Bot, oMa]).

**Definition 3.1** A *Krylov subspace*, short *Krylov space*, is a subspace of $\mathbb{K}^n$ spanned by a sequence of *Krylov vectors*, short *Krylov sequence*

$$A^{k-1}q, \quad A \in \mathbb{K}^{n\times n}, \quad q \in \mathbb{K}^n, \quad k \in \underline{m},$$

We denote this (sub)space by

$$\mathcal{K} \equiv \mathcal{K}(A,q) \equiv \mathcal{K}_m \equiv \mathcal{K}_m(A,q) \equiv \operatorname{span}\left\{q, Aq, A^2q, \ldots, A^{m-1}q\right\} \subset \mathbb{K}^n.$$

The Krylov vectors can be computed iteratively,

$$A^k q = A\left(A^{k-1}q\right), \quad k < m.$$

This leads to a sequence of nested subspaces

$$\operatorname{span}\{0\} \equiv \mathcal{K}_0 \subset \mathcal{K}_1 \subset \mathcal{K}_2 \subset \ldots \subset \mathcal{K}_m, \quad A\mathcal{K}_{k-1} \subset \mathcal{K}_k \quad \forall\, k \in \underline{m}.$$

The dimensions are bounded by

$$0 = \dim \mathcal{K}_0 \leq \ldots \leq \dim \mathcal{K}_{m-1} \leq \dim \mathcal{K}_m \leq m. \tag{3.1}$$

**Definition 3.2** We refer to the matrix

$$K \equiv K(A,q) \equiv K_m \equiv K_m(A,q) \equiv \left[q, Aq, A^2q, \ldots, A^{m-1}q\right] \in \mathbb{K}^{n\times m},$$

whose columns span the Krylov space as the *Krylov matrix*. The Krylov vectors are the columns of the Krylov matrix.

Krylov matrices characterise the vectors in Krylov subspaces. Every vector $x$ in a Krylov subspace $\mathcal{K}_k$ can be written as a linear combination of columns of $K_k$, that is

$$x \in \mathcal{K}_k \quad \Leftrightarrow \quad x = K_k z, \quad z \in \mathbb{K}^k. \tag{3.2}$$

The dimension of a Krylov subspace is given by the rank of the Krylov matrix,

$$\dim(\mathcal{K}_k) = \operatorname{rank}(K_k).$$

Krylov subspaces are furthermore connected to polynomial spaces, we have

$$x \in \mathcal{K}_k \quad \Leftrightarrow \quad x = p_{k-1}(A)q, \quad p_{k-1} \in \mathbb{P}_{k-1}. \tag{3.3}$$

The coefficients of the polynomial $p_{k-1}$ defined in equation (3.3) are the components of the vector $z$ defined in equation (3.2),

$$p_{k-1}(\theta) = z_1 + \cdots + z_k\theta^{k-1} = \sum_{j=1}^{k} z_j\theta^{j-1}. \tag{3.4}$$

**Lemma 3.3** *Suppose* $\dim(\mathcal{K}_k) = k$. *Then equation (3.4) defines an isomorphism*

$$\pi_k : \mathcal{K}_k \to \mathbb{P}_{k-1}$$

*between the spaces* $\mathcal{K}_k$, *the* $k$*th Krylov space, and* $\mathbb{P}_{k-1}$, *the space of polynomials of degree less* $k$.

The condition $\dim(\mathcal{K}_k) = k$ is valid for $k = 0$. The bound (3.1) reveals that there must be a first index $m \geq 0$ with

$$m = \dim(\mathcal{K}_m) = \dim(\mathcal{K}_{m+1}).$$

Then we have equivalence

$$\begin{aligned} A^m q \in \mathcal{K}_m \quad &\Leftrightarrow \quad \mathcal{K}_m = \mathcal{K}_{m+l} \quad \forall\, l \in \mathbb{N} \\ &\Leftrightarrow \quad A\mathcal{K}_m = \mathcal{K}_m. \end{aligned}$$

Thus $\mathcal{K}_m$ is an invariant subspace of $A$. Furthermore we formulate the characterisations using the Krylov matrix, (3.2), and the polynomial space, (3.3),

$$\begin{aligned} A^m q \in \mathcal{K}_m \quad &\Leftrightarrow \quad A^m q = K_m z \\ &\Leftrightarrow \quad K_{m+1}\begin{pmatrix} -z \\ 1 \end{pmatrix} = 0 \\ &\Leftrightarrow \quad p_m(A)\, q = 0, \quad p_m \in \mathbb{P}_m^m. \end{aligned}$$

We summarise the results obtained:

**Theorem 3.4** *Suppose* $m = \dim(\mathcal{K}_m)$. *The following conditions are equivalent:*

$$\begin{aligned} \dim(\mathcal{K}_m) = \dim(\mathcal{K}_{m+1}) \quad &\Leftrightarrow \quad A\mathcal{K}_m = \mathcal{K}_m \\ &\Leftrightarrow \quad p_m(A)\, q = 0, \quad p_m \in \mathbb{P}_m^m. \end{aligned} \tag{3.5}$$

*The index* $m$ *is the dimension of the smallest invariant subspace of* $A$ *that contains the starting vector* $q$.

There is a unique index $m$ with these properties. We will refer to $m$ as the maximal reachable dimension.

Krylov spaces are subspaces of $\mathbb{K}^n$, thus $m \leq n$. Equation (3.5) shows that the index $m$ can be estimated by finding monic polynomials of small degree with

$$p(A)q = 0 \quad \Leftarrow \quad p(A) = 0.$$

Well-known polynomials with the latter property are the characteristic polynomial $\chi_A$ of $A$ (by Cayley-Hamilton) and the minimal polynomial $\mu_A$ of $A$.

The stronger condition involving the vector $q$ is precisely the definition of $\mu_{A,q}$, the minimal polynomial of $A$ with respect to the vector $q$.

This gives us

**Theorem 3.5** *Every Krylov sequence ceases to expand the Krylov subspace after a finite number of steps. More exact, as long as*

$$k \leq m = \deg \mu_{A,q} \leq \deg \mu_A \leq \deg \chi_A = n,$$

*we have that*

$$k = \dim(\mathcal{K}_k) = \operatorname{rank}(K_k).$$

The Krylov spaces defined thus far will be called simple Krylov subspaces.

We will also consider block Krylov subspaces. Block Krylov subspaces are obtained in analogue to simple Krylov spaces, the only change being that $q$ is a block vector.

**Definition 3.6** A *block Krylov subspace* is a subspace of $\mathbb{K}^n$ spanned by a sequence of *block Krylov vectors*

$$A^{k-1}q, \quad A \in \mathbb{K}^{n\times n}, \quad q \in \mathbb{K}^{n\times b}, \quad k \in \underline{m}.$$

The number $b$ denotes the *block size*.

The block size usually remains fixed. Varying block sizes occur when deflation is used, i.e., when $A^{k-1}q$ becomes rank deficient.

The idea behind block methods is to capture multiple eigenvalues and clusters of eigenvalues, since it is well-known that in this case the matrix of moments is better conditioned than a single eigenvalue.

Many of the results obtained for simple Krylov subspaces can be extended to results on block Krylov subspaces. Special care has to be taken for the dimensions of matrices involved.

When speaking of a Krylov space we distinguish between simple and block Krylov spaces only when necessary.

Having defined the spaces, we have to choose a representation, i.e., we have to choose a basis for $\mathcal{K}_m$.

## 3.2 Krylov Subspace Bases

In the following $m = \deg \mu_{A,q}$ denotes maximal reachable dimension. The sizes of matrices given occur when considering simple Krylov spaces.

A natural basis for the Krylov space is given by the Krylov vectors, i.e., by the columns of the Krylov matrix.

In the sequel we will use somewhat sloppy the formulation '*the basis* $X$' as abbreviation of the more precise formulation '*the basis formed by the columns of* $X$'.

**Definition 3.7** We will refer to the basis $K_m$ as the *natural Krylov basis*. We call any basis $Q^{(m)}$ of the Krylov space $\mathcal{K}_m$ a *(general) Krylov basis*.

As we will see in the context of the power method, the first choice leads to a badly conditioned basis. There exist various other bases, each has its own advantages.

The relation between a general and the natural basis can be expressed as matrix equation

$$Q^{(m)}B^{(m)} = K_m, \quad B^{(m)} \in \mathbb{K}^{m\times m}.$$

The transformation matrix $B^{(m)}$ is a non-singular matrix and $Q^{(m)}$ and $K_m$ have both full (column) rank.

We additionally require any basis

- to be computed during execution of the (block) algorithm and
- to be such that the already computed basis (block) vectors remain unaltered.

The vectors spanning the general Krylov basis, i.e., the columns of $Q_m \equiv Q^{(m)}$, are denoted by $q_k, k \in \underline{m}$. We define the submatrices

$$Q_k \equiv [q_1, \ldots, q_k] \quad \forall\ k \in \underline{m}.$$

We are not free in choosing $Q_m$. We have to ensure

$$\mathcal{K}_k = \mathrm{span}(Q_k) \quad \forall\ k \in \underline{m}. \tag{3.6}$$

Suppose the relations between the bases $K_k$ and $Q_k$, $k \in \underline{m}$ are given by

$$Q_k B^{(k)} = K_k, \quad B^{(k)} \in \mathbb{K}^{k\times k}.$$

Equation (3.6) imposes restrictions on the structure of the basis transformations. Consider two subsequent basis transformations $B^{(k)}$, $B^{(k+1)}$. We know that

$$Q_{k+1}B^{(k+1)} = K_{k+1} = \left[K_k, A^k q\right] = \left[Q_k B^{(k)}, A^k q\right]. \tag{3.7}$$

Next partition $Q_{k+1}$ and $B^{(k+1)}$,

$$Q_{k+1}B^{(k+1)} = [Q_k, q_{k+1}] \begin{pmatrix} B_k^{(k+1)} & c_{k+1} \\ r_{k+1}^T & s_{k+1} \end{pmatrix}. \tag{3.8}$$

Ignoring the last column in equation (3.7)

$$\begin{aligned} Q_k B_k^{(k+1)} + q_{k+1} r_{k+1}^T &= Q_k B^{(k)}, \\ \Leftrightarrow \quad Q_k \left(B^{(k)} - B_k^{(k+1)}\right) &= q_{k+1} r_{k+1}^T. \end{aligned} \tag{3.9}$$

We partition the pseudo-inverse of $Q_{k+1}$ as follows,

$$Q_{k+1}^\dagger Q_{k+1} = \begin{pmatrix} \hat{Q}_k^H \\ \hat{q}_{k+1}^H \end{pmatrix} [Q_k, q_{k+1}] = I_{k+1}. \tag{3.10}$$

Applying $\hat{Q}_k^H$ to equation (3.9) leads to

$$I_k \left(B^{(k)} - B_k^{(k+1)}\right) = 0, \tag{3.11}$$

and as consequence

**Lemma 3.8** *The basis transformations $B^{(k)}$, $k \in \underline{m}$*

$$Q_k B^{(k)} = K_k, \quad \text{such that} \quad \mathcal{K}_k = \text{span}(Q_k),$$

*are leading principal submatrices of the basis transformation $B_m \equiv B^{(m)}$, i.e., with the notation of equation (3.8),*

$$B_k \equiv B_k^{(k+1)} = B^{(k)} \quad \forall\ k < m.$$

*The basis transformations are upper (block) triangular matrices, i.e.*

$$R_k \equiv B_k \quad \forall\ k \in \underline{m}.$$

**Proof.** The first part is just equation (3.11) re-written. The second part follows noting that the left-hand side of equation (3.9) vanishes, and thus

$$0 = q_{k+1} r_{k+1}^T.$$

Keep in mind that $q_{k+1} \neq 0$ and thus $r_{k+1} = 0$. □

From now on we will focus on iteratively computable bases, i.e., bases $Q_m$ satisfying

$$Q_m R_m = K_m,$$

with $R_m$ non-singular upper (block) triangular.

**Definition 3.9** A Krylov basis $Q_m$ fulfilling the relation

$$Q_m R_m = K_m$$

with $R_m$ non-singular upper (block) triangular will be called a *(block) Hessenberg basis* of $\mathcal{K}_m$.

The triangular structure is natural, since the $k$th basis vector $q_k$ can only be a combination of vectors available at step $k$. The proof, especially equation (3.10), helps to answer the following question:

How to choose triangular basis transformations when we do not want to compute the natural Krylov basis explicitly?

**Remark 3.10** Sufficient conditions to ensure triangular basis transformations are

a) An orthogonal (orthonormal) basis $Q_m$, i.e., $Q_m$ such that

$$Q_m^H Q_m = D_m \quad (Q_m^H Q_m = I_m)$$

holds true.

b) A biorthogonal (biorthonormal) basis $Q_m$, i.e., $Q_m$ such that with a second matrix $\hat{Q}_m$

$$\hat{Q}_m^H Q_m = D_m \quad (\hat{Q}_m^H Q_m = I_m)$$

holds true.

c) A basis $Q_m$, such that with a second matrix $\hat{Q}_m$

$$\hat{Q}_m^H Q_m = L_m$$

holds true where $L_m$ is lower triangular.

Biorthogonality is sometimes called duality for reasons of brevity.

The most common used and from a numerical viewpoint favourable bases are orthonormal bases.

**Definition 3.11** An orthonormal (block) Hessenberg basis $Q_m$,

$$Q_m R_m = K_m, \quad Q_m^H Q_m = I_m$$

is referred to as *(block) Arnoldi basis* of $\mathcal{K}_m$.

The orthonormal Arnoldi basis $Q_m$ can (in principle) be computed by means of (block) QR decomposition of $K_m$. This is of interest only for theoretical purposes.

The simple Arnoldi basis is unique up to sign. A block Arnoldi basis is unique up to the choice of an orthonormal basis for each block vector. These are direct consequences of the underlying QR decompositions.

The orthonormality condition

$$Q_m^H Q_m = I_m$$

reveals that the upper (block) triangular matrix $R_m$

$$K_m^H K_m = R_m^H Q_m^H Q_m R_m = R_m^H R_m$$

is the (block) Cholesky factor of the (block) Cholesky decomposition of $K_m^H K_m$. This relates existence and uniqueness as well as stability of the (block) Arnoldi basis to the existence, uniqueness and stability of a (block) Cholesky decomposition. An Arnoldi basis always exists.

The orthogonality condition

$$Q_m^H Q_m = D_m$$

in similar fashion is related to the LDLT decomposition

$$K_m^H K_m = R_m^H Q_m^H Q_m R_m = R_m^H D_m R_m.$$

We remark that in case of selfadjoint $A$ the matrix $K_m^H K_m$ is a *Hankel* matrix.

**Definition 3.12** A generalised orthonormal (block) Hessenberg basis $Q_m$,

$$Q_m R_m = K_m, \quad Q_m^H Z Q_m = I_m$$

is referred to as *generalised Arnoldi basis* of $\mathcal{K}_m$.

This generalised orthonormality will be of interest in the solution process of linear systems. Frequent choices for $Z$ include $Z = A^H A$, and if $A$ HPD, $Z = A$.

Construction of a generalised Arnoldi basis is related to the Cholesky decomposition

$$K_m^H Z K_m = R_m^H Q_m^H Z Q_m R_m = R_m^H R_m.$$

When we use two (block) Hessenberg bases $Q_m$, $\hat{Q}_m$,

$$Q_m R_m = K_m \qquad \hat{Q}_m \hat{R}_m = \hat{K}_m \equiv K_m(\hat{A}, \hat{q})$$

of two Krylov spaces $\mathcal{K}_m$, $\hat{\mathcal{K}}_m$,

$$\mathcal{K}_m = \mathrm{span}(K_m) \quad \text{and} \quad \hat{\mathcal{K}}_m = \mathrm{span}(\hat{K}_m),$$

with additional constraint that the two bases $Q_m$, $\hat{Q}_m$ are biorthonormal,

$$\hat{Q}_m^H Q_m = I_m$$

we see that the (block) triangular matrices $R_m, \hat{R}_m$ are just factors of a (block) LR decomposition of the matrix

$$\hat{K}_m^H K_m = \hat{R}_m^H \hat{Q}_m^H Q_m R_m = \hat{R}_m^H R_m.$$

This relates existence, uniqueness and stability of two bi-orthonormal (block) Hessenberg bases to the existence, uniqueness and stability of a (block) LR decomposition.

Merely bi-orthogonal bases $Q_m$, $\hat{Q}_m$,

$$\hat{Q}_m^H Q_m = D_m$$

lead to the LDMT decomposition

$$\hat{K}_m^H K_m = \hat{R}_m^H \hat{Q}_m^H Q_m R_m = \hat{R}_m^H D_m R_m.$$

We remark that when $\mathcal{K}_m = \mathcal{K}_m(A, q)$, and $\hat{\mathcal{K}}_m = \mathcal{K}_m(A^H, \hat{q})$, the matrix $\hat{K}_m^H K_m$ again is a *Hankel* matrix.

Bi-orthogonal Hessenberg bases are important for the non-symmetric Lanczos algorithms.

**Definition 3.13** Bi-orthogonal (block) Hessenberg bases $Q_m$, $\hat{Q}_m$,

$$Q_m R_m = K_m \qquad \hat{Q}_m \hat{R}_m = \hat{K}_m, \qquad \hat{Q}_m^H Q_m = I_m$$

of two Krylov spaces $\mathcal{K}_m$ and $\hat{\mathcal{K}}_m$ will be referred to as *(block) Lanczos bases*.

We see that bi-orthogonal Lanczos bases need not exist (because of occurrence of zero pivots in the decomposition, a so-called *serious breakdown*).

**Remark 3.14** The connection of Arnoldi and Lanczos bases to matrix decompositions gives insight in the methods of Arnoldi and Lanczos to be defined later on.

## 3.3 Krylov Subspace Decompositions

The sizes of matrices given in the following occur when considering simple Krylov spaces.

We know that $A$ maps the $k$th Krylov subspace $\mathcal{K}_k$ onto the $(k+1)$st Krylov subspace $\mathcal{K}_{k+1}$,

$$A\mathcal{K}_k \subset \mathcal{K}_{k+1}. \tag{3.12}$$

To be more precise,

$$[q, AK_k] = K_{k+1}.$$

Inserting any two Krylov bases (not necessarily $q_1^{(k+1)} = q$) gives

$$\begin{aligned} Q^{(k+1)} B^{(k+1)} &= \left[q, AQ^{(k)} B^{(k)}\right] \\ &= \left[q, AQ^{(k)}\right] \begin{pmatrix} I & 0 \\ 0 & B^{(k)} \end{pmatrix}, \end{aligned}$$

where the dimension of the identity matrix is given by the block size of the starting vector. Since $B^{(k)}$ is a basis transformation and thus non-singular we obtain the marvellous relation

$$\left[q, AQ^{(k)}\right] = Q^{(k+1)} B^{(k+1)} \begin{pmatrix} I & 0 \\ 0 & \left(B^{(k)}\right)^{-1} \end{pmatrix}. \tag{3.13}$$

This proves that we have computed a decomposition of the matrix $\left[q, AQ^{(k)}\right]$.

Furthermore we have expressed the equation as one step of a recurrence in the new basis vectors. The matrix representation of the subspace inclusion (3.12) in the chosen bases is given by

$$AQ^{(k)} = Q^{(k+1)}\underline{C}^{(k)}, \quad \underline{C}^{(k)} \in \mathbb{K}^{k+1\times k}.$$

Expressed in equivalent rank-one update formulation,

$$AQ^{(k)} = Q_k^{(k+1)}C^{(k)} + q_{k+1}^{(k+1)}\left(\underline{c}^{(k)}\right)^T, \quad C^{(k)} \in \mathbb{K}^{k\times k}, \quad \left(\underline{c}^{(k)}\right)^T \in \mathbb{K}^{1\times k}.$$

The matrices $\underline{C}^{(k)}$, $C^{(k)}$ and the (block) vector $\underline{c}^{(k)}$ are defined by the relations

$$\begin{pmatrix} c & \underline{C}^{(k)} \\ s & \end{pmatrix} = \begin{pmatrix} c & C^{(k)} \\ s & \left(\underline{c}^{(k)}\right)^T \end{pmatrix} = B^{(k+1)}\begin{pmatrix} I & 0 \\ 0 & \left(B^{(k)}\right)^{-1} \end{pmatrix}. \tag{3.14}$$

**Definition 3.15** Following Stewart ([Ste01]), we will call the subspace equation

$$AQ^{(k)} = Q^{(k+1)}\underline{C}^{(k)}$$

a *Krylov subspace decomposition*, short a *Krylov decomposition*.

The connection given by equation (3.14) between basis transformations and Krylov decompositions relates the existence, uniqueness and stability of the Krylov decompositions to the existence, uniqueness and stability of the basis transformations.

**Remark 3.16** Two Krylov bases uniquely define a Krylov decomposition.

Use again the partitioning

$$\left(Q^{(k+1)}\right)^\dagger Q^{(k+1)} = \begin{pmatrix} \left(\hat{Q}^{(k)}\right)^H \\ \left(\hat{q}^{(k+1)}\right)^H \end{pmatrix}\left[Q_k^{(k+1)}, q_{k+1}^{(k+1)}\right] = I_{k+1}$$

of the pseudo-inverse of $Q^{(k+1)}$.

The matrices $\underline{C}^{(k)}$ and $C^{(k)}$ can be expressed in terms of basis vectors and the system matrix $A$,

$$\begin{aligned} \underline{C}^{(k)} &= \begin{pmatrix} \left(\hat{Q}^{(k)}\right)^H \\ \left(\hat{q}^{(k+1)}\right)^H \end{pmatrix} AQ^{(k)}, \\ C^{(k)} &= \left(\hat{Q}^{(k)}\right)^H AQ_k^{(k)}. \end{aligned}$$

Matters simplify when we are restricting ourselves to Hessenberg bases.

**Lemma 3.17** *Choosing a (block) Hessenberg basis (with fixed block size)*

$$Q_m R_m = K_m$$

*leads to a Krylov decomposition*

$$\begin{aligned} AQ_m &= Q_m C_m, \\ AQ_k &= Q_{k+1}\underline{C}_k = Q_k C_k + q_{k+1}c_{k+1,k}e_k^T \quad \forall\ k < m, \end{aligned} \tag{3.15}$$

*where $\underline{C}_k \equiv \underline{C}^{(k)}$ and $C_k \equiv C^{(k)}$ are (block) Hessenberg matrices. Furthermore they are leading (principal) submatrices of $C_m$.*

**Proof.** Define

$$X_k = R_{k+1} \begin{pmatrix} I & 0 \\ 0 & (R_k)^{-1} \end{pmatrix} \quad \forall\ k < m.$$

The basis transformations $R_k$ of a Hessenberg basis are upper triangular matrices. Thus $X_k$ is upper triangular. The matrices $\underline{C}_k$ ($C_k$) are defined as $X_k$ with first column (and last row) missing. This shows they are (block) Hessenberg.

For the second part it suffices to show that $X_k$ is a principal submatrix of $X_{k+1}$. This is true, because for triangular matrices a) inversion as well as b) building products commutes with forming principal submatrices. □

**Definition 3.18** The Krylov decomposition (3.15) using a Hessenberg basis will be called a *Hessenberg decomposition*.

Equation (3.13) reveals that a Hessenberg decomposition is related to the (block) GR decomposition of a slightly extended matrix,

$$[q, AQ_k] = Q_{k+1} R_{k+1} \begin{pmatrix} I & 0 \\ 0 & (R_k)^{-1} \end{pmatrix}. \tag{3.16}$$

The characteristic polynomials of the matrices $C_k$ are of special interest in a general Hessenberg decomposition. They give the explicit relation between the starting vector $q$ and the basis vectors:

**Theorem 3.19** *The sequence of basis vectors $q_k$ of a simple Hessenberg decomposition*

$$\begin{aligned} AQ_m &= Q_m C_m, \\ AQ_k &= Q_{k+1}\underline{C}_k = Q_k C_k + c_{k+1,k} q_{k+1} e_k^T \quad \forall\ k < m, \end{aligned} \tag{3.17}$$

*can be expressed in terms of $A$, $q$ and the characteristic polynomials of $C_k$,*

$$q_{k+1} \prod_{j=1}^{k} c_{j+1,j} = \chi_{C_k}(A)\, q.$$

**Proof.** We use induction. Trivially

$$q_2 c_{21} = (A - c_{11} I)\, q_1 = \chi_{C_1}(A)\, q.$$

Assume

$$q_l \prod_{j=1}^{l-1} c_{j+1,j} = \chi_{C_{l-1}}(A)\, q \quad \forall\ l \in \underline{k}.$$

Forming (3.17) $e_k$ gives

$$Aq_k - \sum_{l=1}^{k} q_l c_{lk} = q_{k+1} c_{k+1,k}.$$

Using induction hypothesis, multiplying by

$$\prod_{j=1}^{k-1} c_{j+1,j} = \prod_{j=1}^{l-1} c_{j+1,j} \prod_{j=l}^{k-1} c_{j+1,j}$$

we have to show that

$$(A - c_{kk}I)\,\chi_{C_{k-1}}(A) - \sum_{l=1}^{k-1} \chi_{C_{l-1}}(A)\, c_{lk} \prod_{j=l}^{k-1} c_{j+1,j} \stackrel{!}{=} \chi_{C_k}(A).$$

But this is just the expression for the block-wise determinant of

$$I_k \otimes A - C_k^T \otimes I_n = \begin{pmatrix} A - c_{11}I & -c_{21}I & & \\ -c_{12}I & A - c_{22}I & \ddots & \\ \vdots & & \ddots & -c_{k,k-1}I \\ -c_{1k}I & -c_{2k}I & \cdots & A - c_{kk}I \end{pmatrix}$$

evaluated using Laplace expansion by last row. This block-wise determinant is well-defined since $A$ commutes with all polynomials in $A$. □

The question how to find a Hessenberg basis and thus a Hessenberg decomposition was already discussed in Remark 3.10. We will consider the first two variants.

Using the orthogonal Arnoldi basis

$$Q_m R_m = K_m, \quad Q_m^H Q_m = I_m$$

of the Krylov space we have

$$\underline{C}_k = Q_{k+1}^H A Q_k, \quad C_k = Q_k^H A Q_k \quad \forall\ k < m,$$

i.e., the matrices $C_k$ are principal submatrices of a matrix that is *unitarian* similar to $A$.

**Definition 3.20** The Krylov decomposition using the orthogonal Arnoldi basis will be called the *Arnoldi decomposition*.

Since the Arnoldi basis is a Hessenberg basis, we know that $C_k$ is a Hessenberg matrix. This matrix is unique up to sign (see also Theorem 2.13), because the GR decomposition (3.16) becomes a QR decomposition,

$$[q, AQ_k] = Q_{k+1} R_{k+1} \begin{pmatrix} I & 0 \\ 0 & (R_k)^{-1} \end{pmatrix}.$$

The QR decomposition is unique up to sign.

The bases $Q_m$ using merely $Z$-orthogonality,

$$Q_m R_m = K_m, \quad Q_m^H Z Q_m = I_m$$

lead to Krylov decompositions that look similar to a Arnoldi decomposition.

**Definition 3.21** A Krylov decomposition using a $Z$-orthogonal basis will be called a *generalised Arnoldi decomposition*.

This decomposition is unique provided $Z$ is HPD.

Using two bi-orthogonal Lanczos bases $Q_m$, $\hat{Q}_m$,

$$Q_k R_k = K_k, \quad \hat{Q}_k \hat{R}_k = \hat{K}_k,$$

of two Krylov spaces

$$\mathcal{K}_k = \operatorname{span}(K_k), \quad \hat{\mathcal{K}}_k = \operatorname{span}(\hat{K}_k),$$

we have

$$\underline{C}_k = \hat{Q}_{k+1}^H A Q_k, \quad C_k = \hat{Q}_k^H A Q_k.$$

The matrices $C_k$ are again principal submatrices of a matrix, but the similarity no longer is a unitary one.

Furthermore, when we span the second Krylov space by $A^H$, i.e.

$$\hat{\mathcal{K}}_k = \mathcal{K}_k(A^H, \hat{q}) = \text{span}(\hat{K}_k),$$

both $C_k$ and $C_k^H = Q_k^H A^H \hat{Q}_k$ are (block) Hessenberg, i.e., $C_k$ is (block) tridiagonal.

**Definition 3.22** The Krylov decomposition(s) using two bi-orthogonal or dual Lanczos bases will (both) be called *Lanczos decomposition*.

The Arnoldi, Hessenberg and Lanczos decompositions are often introduced in terms of spaces and projectors.

The Arnoldi decomposition is based on orthogonal projection. The orthogonal projector onto $\mathcal{K}_k$ will be denoted by $\mathcal{Q}_k$, i.e.

$$\begin{aligned} \mathcal{Q}_k x &\in \mathcal{K}_k \\ x - \mathcal{Q}_k x &\perp \mathcal{K}_k. \end{aligned}$$

The orthogonal projector $\mathcal{Q}_k$ of the Arnoldi decomposition is given by $Q_k Q_k^H$.

General Hessenberg decompositions are based on oblique projections. The oblique projector onto $\mathcal{K}_k$ orthogonal to $\mathcal{L}_k$ will be denoted by $\mathcal{P}_k$, i.e.

$$\begin{aligned} \mathcal{P}_k x &\in \mathcal{K}_k \\ x - \mathcal{P}_k x &\perp \mathcal{L}_k. \end{aligned}$$

For all Hessenberg decompositions it is possible to find a sequence of spaces $\mathcal{L}_k$. One could choose for example the spaces $\mathcal{L}_k$ spanned by the first $k$ rows of the pseudo-inverse of $Q_{k+1}$ ($Q_m$).

In case of a generalised Arnoldi decomposition the space $\mathcal{L}_k$ is given by $Z^H \mathcal{K}_k$. The corresponding oblique projector is given by $Q_k Q_k^H Z$.

In case of a Lanczos decomposition the space $\mathcal{L}_k$ is itself a Krylov space. The oblique projector $\mathcal{P}_k$ of a Lanczos decomposition is given by $Q_k \hat{Q}_k^H$.

**Remark 3.23** Our notation of orthogonal and oblique projectors is in diametrical opposition to the notation used in the book of Saad ([Saa92]).

Orthogonal projection chooses the minimal element, i.e.

$$\|x - \mathcal{Q}_k x\| = \min_{y \in \mathcal{K}_k} \|x - y\|.$$

Krylov subspace methods are based on Hessenberg decompositions. Hessenberg decompositions can be implemented using the abstract algorithm presented in Algorithm 3.1.

The computed Hessenberg decomposition is used to derive information on the eigendecomposition or to compute approximate solutions to linear systems.

Practical Krylov methods often differ from the abstract algorithm in mixing up the computation of the moments with purging the residual vector or only implicitly performing the Hessenberg decomposition.

Krylov linear system solvers frequently require the solution of small Hessenberg systems involving the computed $C_k$. A natural approach is to find a recurrence that directly computes a decomposition of $C_k$ instead of $C_k$.

**input** : $A$, $r_0$
**output**: $C_k$, $Q_k$ for all $k \in \mathbb{N}$

**for** $k \in \mathbb{N}$ **do**
  Normalise $r_{k-1}$:
  $q_k \leftarrow r_{k-1}/c_{k,k-1}$
  Expand the Hessenberg basis:
  $Q_k \leftarrow [Q_{k-1}, q_k]$
  Expand the Krylov subspace:
  $r_k \leftarrow Aq_k$
  **for** $j \in \underline{k}$ **do**
    Compute $c_{jk}$
  **end for**
  Purge $r_k$:
  $r_k \leftarrow r_k - \sum_j q_j c_{jk}$
**end for**

Algorithm 3.1: Generic Hessenberg decomposition

## 3.4 Krylov Subspace Methods

**Definition 3.24** A *Krylov subspace method*, or *Krylov method* for short, is a method that returns a sequence of approximations from a sequence of Krylov subspaces $\mathcal{K}_k, k \in \underline{m}$.

This definition is slightly more general than the commonly used one. Our definition includes the power method and the subspace iteration in the class of Krylov methods.

We can represent the approximations in any basis. Assume we have chosen to use the Hessenberg basis $Q_m$,

$$\begin{aligned} AQ_m &= Q_m C_m, \\ AQ_k &= Q_{k+1}\underline{C}_k = Q_k C_k + q_{k+1} c_{k+1,k} e_k^T \quad \forall\ k < m. \end{aligned}$$

The different Krylov methods result from different moments $c_{jk}$.

For the moment lets assume we denote the sequence of approximations by $y^{(k)}$, $k \in \underline{m}$. The approximation $y^{(k)}$ from the $k$th Krylov space $\mathcal{K}_k$ can be represented in the Hessenberg basis $Q_m$ as

$$y^{(k)} = Q_k s^{(k)} \quad \forall\ k \in \underline{m}.$$

Next we multiply the Hessenberg decomposition by $s^{(k)}$. This leads to

$$Ay^{(k)} = AQ_k s^{(k)} = Q_{k+1}\underline{C}_k s^{(k)} = Q_k C_k s^{(k)} + q_{k+1} c_{k+1,k} e_k^T s^{(k)}.$$

This equation relates the action of $A$ on $y^{(k)}$ to the action of $\underline{C}_k$ ($C_k$) on $s^{(k)}$.

For $s^{(k)}$ to be computable we need to impose additional restrictions. This is done by projection. Krylov subspace methods are composed of two components, i.e.

**Krylov method = Krylov decomposition + projection**

First we consider Krylov methods for the algebraic eigenvalue problem.

## 3.5 Krylov Methods for the Eigenproblem

The latest (block) vector in the Krylov basis tends to approximate the dominant eigenvector (invariant subspace). This feature is used to compute the dominant eigenspace. This is done in the power method and its block variant, the subspace iteration.

Part of the sequence of nested subspaces tends to approximate invariant subspaces of $A$. Efficient methods for extracting this information are the methods of Lanczos and Arnoldi. These methods are (bi)orthogonal projection methods.

The key idea in such projection methods is to solve a small eigenproblem obtained by the projection

$$(A, I) \to (AQ_k, Q_k) \to (\hat{Q}_k^H AQ_k, \hat{Q}_k^H Q_k) = (\hat{Q}_k^H Q_{k+1}\underline{C}_k, \hat{Q}_k^H Q_k)$$

of the pencil $(A, I)$. The first step of this projection is the result of restricting the space to $\mathcal{K}_k$ in the chosen Hessenberg basis.

The matrix $\hat{Q}_k$ of the second projection step is introduced to balance the number of unknowns and equations.

Sometimes a singular projection is used, this holds true especially in methods that use only the last vectors generated. In this case only the eigenvectors corresponding to non-zero eigenvalues of the projected eigenproblem are used.

We present an abstract formulation of a Krylov subspace method for the solution of the eigenproblem. We assume the Krylov space to be spanned by $A$ and the (unnormalised) starting vector $r_0$. The resulting algorithm is Algorithm 3.2.

**input** : $A$, $r_0$

**output**: $C_k$, $Q_k$, $S_k$, $J_\Theta^{(k)}$ for all $k \in \mathbb{N}$

**for** $k \in \mathbb{N}$ **do**

Compute Hessenberg decomposition:
$AQ_k = Q_k C_k + q_{k+1} c_{k+1,k} e_k^T$

Compute projection of pencil:
$\hat{Q}_k^H (A, I) Q_k = (\hat{Q}_k^H Q_{k+1}\underline{C}_k, \hat{Q}_k^H Q_k)$

Compute eigendecomposition:
$\hat{Q}_k^H Q_{k+1}\underline{C}_k S_k = \hat{Q}_k^H Q_k S_k J_\Theta^{(k)}$

Compute approximate eigenvectors:
$Y_k \leftarrow Q_k S_k$

**end for**

Algorithm 3.2: Generic Krylov eigensolver

Since in general the simple eigenproblem is easier to solve, we intend to use an orthonormal or bi-orthonormal projection matrix $\hat{Q}_k$. This leads to the simpler algorithm in Algorithm 3.3.
In most Hessenberg decompositions defined thus far we explicitly know how to choose $\hat{Q}_k$. In the Arnoldi decomposition we use $\hat{Q}_k = Q_k$, in a generalised Arnoldi decomposition we use $\hat{Q}_k = Z^H Q_k$, and in case of a Lanczos decomposition the bases $\hat{Q}_k$, $Q_k$ are bi-orthonormal by construction.

In any of these cases the projected pencil

$$(\hat{Q}_k^H AQ_k, \hat{Q}_k^H Q_k) = (\hat{Q}_k^H Q_{k+1}\underline{C}_k, \hat{Q}_k^H Q_k) = (C_k, I)$$

has a very simple structure. We have to solve a simple eigenproblem involving the matrix $C_k$ of the Hessenberg decomposition.

**input** : $A$, $r_0$

**output**: $C_k$, $Q_k$, $S_k$, $J_\Theta^{(k)}$ for all $k \in \mathbb{N}$

**for** $k \in \mathbb{N}$ **do**
  Compute Hessenberg decomposition:
  $AQ_k = Q_k C_k + q_{k+1} c_{k+1,k} e_k^T$
  Compute eigendecomposition:
  $C_k S_k = S_k J_\Theta^{(k)}$
  Compute approximate eigenvectors:
  $Y_k \leftarrow Q_k S_k$
**end for**

Algorithm 3.3: Bi-orthonormal Krylov eigensolver

In case of an orthogonal projection $C_k$ is the result of a Ritz-Galerkin method. For this reason we define

**Definition 3.25** The eigenvalues of $C_k$ resulting from (bi)orthogonal projection are named *Ritz values* and the prolonged eigenvectors are named *Ritz vectors*. The characteristic polynomials

$$\chi_{C_k}(\theta) \equiv \det(\theta I_k - C_k)$$

of the Hessenberg matrices $C_k$ are labelled *Ritz polynomials*.

The Ritz polynomials are sometimes also called *Lanczos*, *Hankel* or *Hadamard polynomials*.

Krylov subspace methods are polynomial methods. The polynomials chosen to built a new basis vector are just the characteristic polynomials of the Hessenberg matrices $C_k$. This obviously follows from the recurrence relation of the basis vectors.

### 3.5.1 The Power Method

The power method is one of the oldest methods known for computing eigenvalue estimates. It uses the natural Krylov basis with additional scaling to avoid underflow and overflow. We denote the scaled Krylov vectors by $q_k$, and the decomposition by

$$AQ_k = Q_{k+1}\underline{N}_k = Q_k N_k + \nu_k q_{k+1} e_k^T \quad \forall\ k \in \mathbb{N}. \tag{3.18}$$

The matrix $N_k$ takes care of the normalising factors $\nu_j$ and is given by

$$N_k = \begin{pmatrix} 0 & & & & 0 \\ \nu_1 & 0 & & & \\ & \nu_2 & \ddots & & \\ & & \ddots & 0 & \\ 0 & & & \nu_{k-1} & 0 \end{pmatrix}.$$

For regular $A$ and non-zero $q$ this recurrence can be carried out for all $k \in \mathbb{N}$.

We use a singular projection to obtain a smaller eigenproblem. We project the Hessenberg decomposition with the singular matrix

$$\hat{Q}_k^H = \Big[0, \ldots, 0, q_k\Big]^H \in \mathbb{K}^{k \times n}.$$

This leads to the small (singular) pencil

$$\begin{aligned}\left(\hat{Q}_k^H A Q_k, \hat{Q}_k^H Q_k\right) &= \left(\hat{Q}_k^H Q_{k+1}\underline{N}_k, \hat{Q}_k^H Q_k\right) \\ &= \left(\begin{pmatrix} 0 & 0 \\ q_k^H Q_k \underline{N}_{k-1} & \nu_k q_k^H q_{k+1}\end{pmatrix}, \begin{pmatrix} 0 & 0 \\ q_k^H Q_{k-1} & q_k^H q_k\end{pmatrix}\right)\end{aligned}$$

The only non-trivial solution is the last unit vector $s_k = e_k$. This leads to the approximate eigenvector

$$y_k = Q_k s_k = Q_k e_k = q_k.$$

The approximate eigenvalue is given by

$$\theta_k = \frac{\nu_k q_k^H q_{k+1}}{q_k^H q_k} = \frac{q_k^H A q_k}{q_k^H q_k},$$

which is just the Rayleigh quotient of $y_k = q_k$.

A variant of the power method is given with Algorithm 3.4. In this variant the recurrence vectors are normalised to have unit length. Another frequent choice is normalisation by largest entry.

**input** : $A$, $r_0$
**output**: $N_k$, $Q_k$ for all $k \in \mathbb{N}$

**for** $k \in \mathbb{N}$ **do**
  $\nu_{k-1} \leftarrow \|r_{k-1}\|$
  $q_k \leftarrow r_{k-1}/\nu_{k-1}$
  $r_k \leftarrow A q_k$
  $\theta_k \leftarrow \langle q_k, r_k \rangle$
**end for**

Algorithm 3.4: Power method

We assume $A$ not to be nilpotent and the starting vector not to be contained in the null-space of $A$. Once invoked, the power method will continue forever.

Given $A$ and $q_k$, the Rayleigh quotient $\theta_k$ of $q_k$ minimises the residual

$$A q_k - q_k \theta_k$$

with respect to Euclidian norm. Simultaneously the residual is orthogonal to the approximate eigenvector $q_k$. This is obvious by derivation, but is also an intrinsic property of the Rayleigh quotient.

We now focus on the rate of convergence to an eigenvector. Defining $q = q_1$ to be the normalised $r_0$, the recurrence of the basis vectors can be stated more condensed as

$$q_{k+1} \prod_{j=1}^{k} \nu_j = A^k q \tag{3.19}$$

The power method is a *monomial method*.

There are two ways to analyse the behaviour of the vectors $q_k$. The first way is to insert the Jordan normal form of $A$,

$$A = V J_\Lambda V^{-1} = V J_\Lambda \hat{V}^H,$$

into equation (3.19). This results in

$$q_{k+1} \prod_{j=1}^{k} \nu_j = V \left(J_\Lambda\right)^k V^{-1} q = V \left(J_\Lambda\right)^k \hat{V}^H q.$$

**Remark 3.26** We can choose the Jordan decomposition such that we have at most one Jordan block with non-zero part $\hat{V}^H q$ for every eigenvalue.

We first analyse the behaviour for a single Jordan block $J_\lambda \equiv J_\Lambda$. We assume w.l.o.g. that $k > l$, where $l$ is the size of the Jordan block. Define

$$Z^{(k)} \equiv (J_\lambda)^k .$$

The matrix $Z^{(k)}$ is upper triangular.

Suppose $s \geq i$. The components of the upper triangular part of $Z^{(k)}$ are given by

$$\begin{aligned} z^{(k)}_{j,j+i} &= \binom{k}{i}\lambda^{k-i} \\ &= \binom{k}{s}\lambda^{k-s}\frac{s(s-1)\cdot\dots\cdot(i+1)\lambda^{s-i}}{(k-i)(k-i-1)\cdot\dots\cdot(k-s+1)} \\ &= \binom{k}{s}\lambda^{k-s}O(k^{i-s}). \end{aligned}$$

Choose $s$ to be the largest index such that the corresponding part of $\hat{V}^H q$ is non-zero. This part behaves asymptotically like

$$\left(\binom{k}{s}\lambda^{k-s}\right)\begin{pmatrix} O(1) & O(k^{-1}) & \cdots & O(k^{-l}) \end{pmatrix}^T .$$

Apart from normalisation, this vector converges to the first unit vector $e_1$. The part corresponding to the subspace associated with the eigenvalue $\lambda$ converges to the unique eigenvector.

Next consider two Jordan blocks of two eigenvalues different in absolute value, w.l.o.g. labelled $|\lambda_1| > |\lambda_2|$,

$$\begin{aligned} \left(\binom{k}{s_1}\lambda_1^{k-s_1}\right)\left(\binom{k}{s_2}\lambda_2^{k-s_2}\right)^{-1} &= O(k^{s_1-s_2})\left(\frac{\lambda_1}{\lambda_2}\right)^{k-s_1}\left(\frac{1}{\lambda_2}\right)^{s_1-s_2} \\ &= O\left(k^{s_1-s_2}\left(\frac{\lambda_1}{\lambda_2}\right)^k\right). \end{aligned}$$

The part corresponding to the eigenvalue smaller in absolute value vanishes.

The vectors generated converge to a mixture of eigenvectors corresponding to maximal $|\lambda|$. If only one eigenvalue of maximal modulus exists, the method converges to the corresponding eigenvector.

When there are several distinct eigenvalues of maximal modulus,

$$|\lambda_1| = |\lambda_2| = \ldots = |\lambda_l|,$$

we often observe periodicity and quasi-periodicity in the computed vectors (cf. [KGL97]).

Another way is to insert the Schur decomposition,

$$q_{k+1}\prod_{j=1}^{k}\nu_j = U\left(\Lambda+N\right)^k U^H q = U\left(\sum_{l=0}^{k}\binom{k}{l}\Lambda^l N^{k-l}\right)U^H q.$$

The matrix $N$ is a nilpotent matrix which is zero in case of a normal matrix. Again we observe convergence to the eigenvector(s) of maximal modulus.

**Remark 3.27** We see that non-normality (i.e., huge $N^{k-l}$ compared to $\Lambda^l$) causes late convergence of the method and huge intermediate quantities.

If the eigenvalue largest in absolute value is *simple*, and there is a *substantial gap* between the first two eigenvalues, we observe *fast convergence*. When all eigenvalues are simple and well separated in absolute value we can use deflation to compute one eigenvalue after the other.

The power method can be applied to shifted and inverted $A$. In this case the power method is called *inverse iteration*. The shifts may vary from step to step since the eigenvectors remain unaltered by shifts. This idea is due to Wielandt and was analysed mainly by Wilkinson. Some history of Wielandt's and Wilkinson's work on the power method and inverse iteration may be found in three articles by Ipsen (cf. [Ips96a], [Ips96b], [Ips97]).

The power method delivers estimates to one eigenvalue. Yet the decomposition does contain more information on $A$. Assume the Hessenberg decomposition (3.18) is available. Assume further that $k \leq m$, such that $Q_k$ has full rank. Multiply (3.18) from the left by $Q_k^\dagger$ to obtain

$$Q_k^\dagger A Q_k = N_k + \nu_k Q_k^\dagger q_{k+1} e_k^T.$$

For simplicity assume further that we do not use any normalisation, i.e. $Q_k = K_k$. This leads to a unit lower diagonal in $N_k$ and thus $K_k^\dagger A K_k$ is a companion matrix.

**Remark 3.28** The coefficients of the characteristic polynomial of $K_k^\dagger A K_k$ are related to the Krylov basis via

$$\chi_{K_k^\dagger A K_k}(\theta) = \theta^k - a_1^{(k)}\theta^{k-1} - a_2^{(k)}\theta^{k-2} - \cdots - a_k^{(k)} \in \mathbb{P}_k^k,$$

where $a^{(k)} = K_k^\dagger A^k q$, i.e., the coefficients minimise $\|K_k a^{(k)} - A^k q\|$.

As we will see, the roots of these polynomials approximate eigenvalues of $A$. The pseudo-inverse becomes economic when using an orthonormal basis. The method we sketched here is the Arnoldi method. A related derivation of the Arnoldi method is given in the books of Demmel ([Dem97]) and Meurant (cf. [Meu99]). The latter grants it to C. Vuik.

### 3.5.2 Subspace Iteration

The power method works best for matrices with a large gap between largest and second largest eigenvalue. If this gap is small, a better alternative might be the block power method, better known as *subspace iteration*, also known as *simultaneous iteration*.

Subspace iteration performs block Hessenberg decomposition

$$A Q_k = Q_{k+1} \underline{N}_k = Q_k N_k + q_{k+1} \nu_k e_k^T \quad \forall\ k \in \mathbb{N}.$$

The Hessenberg decomposition computed in subspace iteration is similar to the Hessenberg decomposition computed in the power method, but now $N_k$ is a block matrix, the normaliser $\nu_k$ is a small matrix and the vectors $q_k$ are block vectors.

To maintain linear independence a factorisation of the block vectors is used. The first code developed in 1957 by Bauer is known as 'Treppeniteration' (staircase iteration) and was based on the LR decomposition. When a complete set of vectors spanning $\mathbb{K}^n$ is used, this code was superseded 1958 by the LR iteration by Rutishauser. The next step was to use *orthogonal* transformations, i.e., the QR decomposition, which resulted 1961 in Francis' and Kublanovskaya's QR iteration. Both the LR and QR iterations work on a full set of vectors and are considered to be direct methods.

The subspace iteration code used today, for example SRRIT (cf. [BS97]), performs a QR decomposition of the block vectors $A^{\text{iter}}q_k$,

$$q_{k+1}\nu_k = A^{\text{iter}}q_k.$$

Additionally a Schur decomposition is applied from time to time, such that the columns of $q_k$ tend to approximate Schur vectors. Without this Schur step the block vector $q_k$ will tend to approximate a random basis of the invariant subspace of maximal modulus.

When the decomposition step is left out to often, i.e., when iter is to large, the block vector $q_k$ may become badly conditioned regarded as basis.

### 3.5.3 The Arnoldi Method

The Arnoldi method is simply the computation of the Arnoldi decomposition

$$\begin{aligned} AQ_m &= Q_m H_m, \\ AQ_k &= Q_{k+1}\underline{H}_k = Q_k H_k + h_{k+1,k}q_{k+1}e_k^T \quad \forall\ k < m, \end{aligned} \tag{3.20}$$

of the Krylov space $\mathcal{K}_m$. In the following $H_k$ will always denote the Hessenberg matrix obtained in the Arnoldi decomposition.

One algorithmic implementation of the Arnoldi method, to be precise, the variant where modified Gram-Schmidt is used to orthogonalise the basis vectors, i.e., MGS-Arnoldi, is given by Algorithm 3.5.

**input** : $A$, $r_0$
**output**: $H_k$, $Q_k$ for all $k \in \mathbb{N}$

**for** $k \in \mathbb{N}$ **do**
  $h_{k,k-1} \leftarrow \|r_{k-1}\|$
  $q_k \leftarrow r_{k-1}/h_{k,k-1}$
  $r_k \leftarrow Aq_k$
  **for** $j \in \underline{k}$ **do**
    $h_{jk} \leftarrow \langle q_j, r_k\rangle$
    $r_k \leftarrow r_k - q_j h_{jk}$
  **end for**
**end for**

Algorithm 3.5: Arnoldi method (MGS variant)

This algorithm is an example of mixing up the steps of computing the moments with the purification of the residual vector. This change corresponds to choosing MGS (Modified Gram-Schmidt) instead of CGS (Classical Gram-Schmidt) as orthogonalisation technique.

As first observation concerning the domain of useful indices we state:

**Theorem 3.29 (breakdown of the Arnoldi method)** *The Arnoldi method is applicable as long as*

$$\dim(\mathcal{K}_k) = k \leq m = \deg \mu_{A,q} \leq \deg \mu_A \leq n.$$

*The subspace spanned by $Q_m$ is an invariant subspace of $A$, and all eigenvalues of $H_m$ are eigenvalues of $A$.*

**Proof.** The first part is trivial. Let the Jordan decompositions of $H_k$, $k \in \underline{m}$ be given as

$$H_k S^{(k)} = S^{(k)} J_\Theta^{(k)} \quad \forall\ k \in \underline{m}.$$

Define the matrices of generalised Ritz vectors

$$Y^{(k)} = Q_m S^{(k)} \quad \forall\ k \in \underline{m}.$$

The algorithm breaks down in step $m$ with $h_{m+1,m} = 0$. In this case

$$AQ_m = Q_m H_m \quad \Rightarrow \quad AY^{(m)} = Y^{(m)} J_\Theta^{(m)}.$$

Hence all eigenvalues of $H_m$ are eigenvalues of $A$. □

Even when $h_{k+1,k}$ is far from zero, parts of the eigensystem

$$J_\Theta^{(k)}, \qquad Y^{(k)} = \left[y_1^{(k)}, \ldots, y_k^{(k)}\right],$$

of $H_k$ might deliver good approximations.

Suppose $k$ fixed. We can measure how well the $j$th right *Ritz pair*

$$\theta_j = \theta_j^{(k)}, \qquad y_j \equiv Q_k s_j \equiv Q_k s_j^{(k)}, \qquad C_k s_j = s_j \theta_j,$$

approximates a right eigenpair without forming $Y = Y^{(k)}$. Multiplying (3.20) by $s_j$ leads to

$$Ay_j - y_j\theta_j = Q_k^H \left(H_k s_j - s_j \theta_j\right) + h_{k+1,k} q_{k+1} s_{kj}.$$

Taking norms we obtain

$$\|Ay_j - y_j\theta_j\| = |h_{k+1,k} s_{kj}|.$$

Keep in mind that

$$\|q_{k+1}\| = 1 \quad \text{and} \quad \|y_j\| = \|Q_k s_j\| = \|s_j\| = 1$$

by construction. We can also think of a component-wise bound, since simple application of component-wise absolute value leads to the similar bound

$$|Ay_j - y_j\theta_j| = |h_{k+1,k} s_{kj}| \cdot |q_{k+1}|\,.$$

It turns out that for outliers the size of the residuals $|h_{k+1,k} s_{kj}|$ very soon becomes negligible.

The Ritz value is the Rayleigh quotient of the Ritz vector,

$$\begin{aligned} y_j^H A y_j - \theta_j &= y_j^H \left(Ay_j - y_j\theta_j\right) \\ &= y_j^H q_{k+1} h_{k+1,k} s_{kj} \\ &= s_j^H Q_k^T q_{k+1} h_{k+1,k} s_{kj} = 0. \end{aligned}$$

We present the results in matrix formulation:

**Theorem 3.30** *The approximation error of the kth Ritz decomposition*

$$Y^{(k)} = Q_k S^{(k)}, \quad J_\Theta^{(k)},$$

*is given by a rank one matrix composed of the residual of the Arnoldi decomposition and the last row of* $S^{(k)}$,

$$AY^{(k)} - Y^{(k)} J_\Theta^{(k)} = q_{k+1} h_{k+1,k} e_k^T S^{(k)}.$$

*The Jordan matrix of* $H_k$ *is the generalised Rayleigh quotient*

$$\left(Y^{(k)}\right)^\dagger A Y^{(k)} = \left(S^{(k)}\right)^{-1} Q_k^H A Q_k S^{(k)} = J_\Theta^{(k)}.$$

As we have already shown in the section on the power method:

**Remark 3.31** The coefficients of the characteristic polynomial of $H_k$ are related to the Krylov bases of subsequent steps via

$$\chi_{H_k}(\theta) = \theta^k - a_1^{(k)}\theta^{k-1} - a_2^{(k)}\theta^{k-2} - \cdots - a_n^{(k)},$$

where

$$a^{(k)} = K_k^\dagger A^k q.$$

We know that application of the pseudo-inverse delivers the solution with minimal (2-norm) residual

$$\|A^k q - K_k a^{(k)}\| = \min. \tag{3.21}$$

We use the correspondence of polynomials in $A$ and vectors in the Krylov space to obtain

$$\begin{aligned} \|A^k q - K_k a^{(k)}\| &= \min_{a\in\mathbb{K}^k} \|A^k q - K_k a\| = \min_{a\in\mathbb{K}^k} \|K_{k+1}\begin{pmatrix} -a \\ 1 \end{pmatrix}\| \\ &= \min_{p_k\in\mathbb{P}_k^k} \|p_k(A)q\| = \|\chi_{H_k}(A)q\|. \end{aligned} \tag{3.22}$$

Equation (3.22) proves:

**Theorem 3.32 (minimisation property of the Arnoldi method)** *The Ritz polynomials computed in the Arnoldi method minimise*

$$\prod_{j=1}^{k} |h_{j+1,j}| = \|q_{k+1}\prod_{j=1}^{k} h_{j+1,j}\| = \|\chi_{H_k}(A)q\| = \min_{p_k\in\mathbb{P}_k^k} \|p_k(A)q\| \tag{3.23}$$

*over the space $\mathbb{P}_k^k$ of all monic polynomials of degree less equal $k$.*

We re-state that we can interprete one step of the Arnoldi decomposition as QR decomposition

$$\begin{aligned} [q_1, AQ_m] = \left[q_1, AK_m R_m^{-1}\right] &= K_{m+1}\cdot\begin{pmatrix} 1 & 0 \\ 0 & R_m^{-1} \end{pmatrix} \\ &= Q_{m+1}R_{m+1}\cdot\begin{pmatrix} 1 & 0 \\ 0 & R_m^{-1} \end{pmatrix} \end{aligned}$$

of the matrix $[q_1, AQ_m]$.

Recall that the Hessenberg matrices $H_k$ and $\underline{H}_k$ of the Arnoldi decomposition are defined by

$$[e_1, \underline{H}_k] = \begin{pmatrix} e_1 & H_k \\ 0 & h_{k+1,k}e_k^T \end{pmatrix} = R_{k+1}\cdot\begin{pmatrix} 1 & 0 \\ 0 & R_k^{-1} \end{pmatrix}. \tag{3.24}$$

These two observations are important in the analysis of the finite precision behaviour.

We know that the Arnoldi method delivers Hessenberg matrices. The next remark clarifies the interplay between Hessenberg matrices and the Arnoldi method.

**Remark 3.33** Every unreduced upper Hessenberg matrix can be inserted into an Arnoldi decomposition,

$$\begin{aligned} H_m I_m &= I_m H_m, \\ H_m I_{m,k} &= I_{m,k+1}\underline{H}_k = I_{m,k}H_k + h_{k+1,k}e_k^T \quad \forall\ k < m. \end{aligned}$$

This is achieved for unreduced lower Hessenberg matrices by choosing a flipped identity. This reveals once again that all unreduced Hessenberg matrices are non-derogatory.

This remark helps to predict the observable behaviour.

The Arnoldi method was originally based on Lanczos ideas. It was developed by and named after Walter Edwin Arnoldi (*1917 – †1996). Arnoldi published his method in 1951 (cf. [Arn51]).

Saad also considered a block variant of the Arnoldi method. In this method the matrix $C_k = H_k$ becomes block Hessenberg (cf. [Saa80]).

### 3.5.4 The Symmetric Lanczos Method

The symmetric Lanczos method is Arnoldi's method applied to selfadjoint matrices. When the Arnoldi method is applied to a selfadjoint matrix $A = A^H$, the Hessenberg matrix obtained must be selfadjoint, i.e., tridiagonal, since

$$H_k = Q_k^H A Q_k = H_k^H \equiv T_k$$

is selfadjoint and tridiagonal.

The symmetric Lanczos method differs from the Arnoldi method in three respects. First, the low storage costs. We only need three basis vectors and three moments per step. Second, the preservation of structure. This implies that both $A$ and $T_k$ have an orthogonal eigenbasis. Third, its connection to orthogonal polynomials.

Like the Arnoldi method, its epigone, the symmetric Lanczos method, can be written as Arnoldi decomposition

$$\begin{aligned} AQ_m &= Q_m T_m, \\ AQ_k &= Q_{k+1}\underline{T}_k = Q_k T_k + \beta_k q_{k+1} e_k^T \quad \forall\ k < m. \end{aligned} \tag{3.25}$$

The matrix $T_k$ is composed of recurrence coefficients,

$$T_k = \operatorname{tridiag}_k(\beta, \alpha, \beta) = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{pmatrix}.$$

We now state an algorithmic implementation of the symmetric Lanczos method with Algorithm 3.6.

**input** : $A$, $r_0$
**output**: $T_k$, $Q_k$ for all $k \in \mathbb{N}$

$q_0 \leftarrow 0$
**for** $k \in \mathbb{N}$ **do**
  $\beta_{k-1} \leftarrow \|r_{k-1}\|$
  $q_k \leftarrow r_{k-1}/\beta_{k-1}$
  $r_k \leftarrow A q_k$
  $\alpha_k \leftarrow \langle q_k, r_k \rangle$
  $r_k \leftarrow r_k - \alpha_k q_k - \beta_{k-1} q_{k-1}$
**end for**

Algorithm 3.6: Lanczos method (symmetric variant)

The matrix $T_k$ obtained in Algorithm 3.6 is an unreduced selfadjoint tridiagonal matrix with real positive off-diagonal elements.

All results obtained for the Arnoldi method hold:

**Theorem 3.34 (breakdown of symmetric Lanczos)** *The symmetric Lanczos method is applicable as long as*

$$\dim(\mathcal{K}_k) = k \leq m = \deg \mu_{A,q} \leq \deg \mu_A \leq n.$$

*The subspace spanned by $Q_m$ is an invariant subspace of $A$, and all eigenvalues of $T_m$ are eigenvalues of $A$.*

The selfadjointness has consequences on the approximation error:

**Theorem 3.35** *The approximation error of the kth Ritz decomposition*

$$Y^{(k)} = Q_m S^{(k)}, \quad \Theta^{(k)},$$

*is given by a rank one matrix composed of residual of the Arnoldi decomposition and the last row of the unitary $S^{(k)}$,*

$$AY^{(k)} - Y^{(k)}\Theta^{(k)} = q_{k+1}\beta_k e_k^T S^{(k)}.$$

*The matrix of Ritz values is the generalised Rayleigh quotient*

$$\left(Y^{(k)}\right)^H AY^{(k)} = \Theta^{(k)}.$$

*Furthermore we can bound the distance to the nearest eigenvalue,*

$$|\lambda - \theta_j| \leq \frac{\|Ay_j - y_j\theta_j\|}{\|y_j\|} = \|\beta_k s_{kj} q_{k+1}\| = |\beta_k s_{kj}| \leq |\beta_k|.$$

The minimisation property holds:

**Theorem 3.36 (minimisation property of symmetric Lanczos)** *The Ritz polynomials computed in the symmetric Lanczos method minimise*

$$\prod_{j=1}^{k} |\beta_j| = \|q_{k+1} \prod_{j=1}^{k} \beta_j\| = \|\chi_{T_k}(A)q)\| = \min_{p_k \in \mathbb{P}_k^k} \|p_k(A)q\|$$

*over the space $\mathbb{P}_k^k$ of all monic polynomials of degree less equal $k$.*

The connection to orthogonal polynomials is a direct consequence of the three-term recurrence and the orthonormality of the basis vectors:

**Theorem 3.37 (orthogonality of the Ritz polynomials)** *The sequence of characteristic polynomials $\chi_k \equiv \chi_{T_k}$ satisfies the three-term recurrence*

$$\beta_{k+1}\chi_{k+1}(\lambda) = (\lambda - \alpha_k)\,\chi_k(\lambda) - \beta_k\chi_{k-1}(\lambda).$$

*The polynomials $\chi_k$ are orthogonal with respect to the inner product*

$$\langle \chi_k, \chi_j \rangle_S = \int \overline{\chi_k}\chi_j \, du. \tag{3.26}$$

*This inner product is a Stieltjes integral with integrating function*

$$u(\lambda) = \sum_i \left|\hat{v}_i^H q\right|^2 H(\lambda - \lambda_i).$$

**Proof.** This is merely a change in notation. We define the Stieltjes integral

$$\int f(\lambda)\, du(\lambda) \equiv q^H f(A)\, q = \sum_i \left|\hat{v}_i^H q\right|^2 f(\lambda_i).$$

This integral has the integrating function

$$u(\lambda) = \sum_i \left|\hat{v}_i^H q\right|^2 H(\lambda - \lambda_i).$$

We can interprete the integral (3.26) as Rayleigh quotient of a matrix function,

$$\int \overline{\chi_k}\chi_l\, du = q^H \overline{\chi_k(A)}\chi_l(A) q = q_k^H q_l \prod_{j=1}^{k} \overline{\beta_j} \prod_{j=1}^{l} \beta_j.$$

The orthonormality of the recurrence vectors simplifies the last line to

$$\int \overline{\chi_k}\chi_l\, du = \delta_{kl} \prod_{j=1}^{k} |\beta_j|^2,$$

which proves the orthogonality of the polynomials $\chi_k$. □

The symmetric (and non-symmetric) Lanczos method was developed by and named after Cornelius Lanczos (Kornél Löwy, Kornél Lánczos, *1893 – †1974). He published two papers (cf. [Lan50, Lan52]) on 'the method of minimised iterations'. These papers formed the basis for a variety of Krylov methods, the most prominent among them the symmetric and non-symmetric Lanczos method as well as some form of BiCG.

Blocked and banded variants of the symmetric Lanczos methods have been considered by Ericsson and Ruhe, by Underwood and Golub, by Cullum and Donath and by Parlett and Kahan (cf. [ER80, Und75]).

### 3.5.5 The Non-Symmetric Lanczos Method

We have defined two *orthogonal* projection methods. The Arnoldi method applies to general matrices, but uses long recurrences. The symmetric Lanczos method uses short recurrences, but applies only to selfadjoint matrices.

Using an *oblique* projection method it is possible to obtain a method that recovers short recurrences *and* applies to general matrices. This method is the *non-symmetric Lanczos method*.

The non-symmetric Lanczos method relies on a Lanczos decomposition and corresponds to a Petrov-Galerkin approach. The Lanczos decomposition is based on two Krylov spaces $\mathcal{K}_k$ and $\hat{\mathcal{K}}_k$, usually

$$\begin{array}{rcl} \mathcal{K}_m & = & \mathcal{K}_m(A, q) \quad = \quad \left\{q, Aq, \ldots, A^{m-1}q\right\}, \\ \hat{\mathcal{K}}_m & = & \hat{\mathcal{K}}_m\left(A^H, \hat{q}\right) \quad = \quad \left\{\hat{q}, A^H\hat{q}, \ldots \left(A^H\right)^{m-1}\hat{q}\right\}. \end{array}$$

The starting vectors are assumed bi-orthonormal, i.e., $\hat{q}^H q = 1$.

We already noted that the matrix $C_m$ resulting from the oblique projection underlying the non-symmetric Lanczos method is tridiagonal, $T_m \equiv C_m$. In contrast to the symmetric Lanczos method $T_m$ is a *general* tridiagonal matrix.

We denote diagonal elements by $\alpha_k$, sub-diagonal elements by $\beta_k$ and super-diagonal elements by $\gamma_k$, i.e.

$$T_m = \mathrm{tridiag}_k(\beta, \alpha, \gamma) = \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{pmatrix}.$$

The resulting Lanczos decomposition

$$\begin{aligned} AQ_m &= Q_m T_m, \\ AQ_k &= Q_{k+1}\underline{T}_k = Q_k T_k + q_{k+1}\beta_k e_k^T \quad \forall\ k < m, \end{aligned} \tag{3.27}$$

need not exist due to breakdown of the underlying LR decomposition of the Hankel matrix. Assume for the moment that no breakdown occurs.

The non-symmetric Lanczos method also computes a second Lanczos decomposition. This Lanczos decomposition is related to $A^H$,

$$\begin{aligned} A^H\hat{Q}_m &= \hat{Q}_m\hat{T}_m, \\ A^H\hat{Q}_k &= \hat{Q}_{k+1}\underline{\hat{T}}_k = \hat{Q}_k\hat{T}_k + \hat{q}_{k+1}\hat{\beta}_k e_k^T \quad \forall\ k < m. \end{aligned} \tag{3.28}$$

For reasons of readability we introduced the matrices $\hat{T}_k = T_k^H$, $k \in \underline{m}$. The elements of $\hat{T}_k$ are denoted with a small hat, and correspond to the elements of $T_k$ by $\hat{\alpha}_k = \overline{\alpha_k}$, $\hat{\beta}_k = \overline{\gamma_k}$ and $\hat{\gamma}_k = \overline{\beta_k}$.

This is also a more general form, where we allow for non-bi-orthonormal bases. In this variant the computed bases $Q_k$, $\hat{Q}_k$ are bi-orthonormal, i.e., such that

$$\hat{Q}_m^H Q_m = I_m$$

holds true.

We are now able to state an algorithmic implementation. The general algorithm for a non-symmetric Lanczos method is given with Algorithm 3.7.

**input** : $A$, $r_0$, $\hat{r}_0$
**output**: $T_k$, $Q_k$, $\hat{Q}_k$ for all $k \in \mathbb{N}$

$q_0 \leftarrow 0, \quad \hat{q}_0 \leftarrow 0$
**for** $k \in \mathbb{N}$ **do**
  $\beta_{k-1}\gamma_{k-1} \leftarrow \langle \hat{r}_{k-1}, r_{k-1}\rangle$
  $q_k \leftarrow r_{k-1}/\beta_{k-1}$
  $\hat{q}_k \leftarrow \hat{r}_{k-1}/\overline{\gamma_{k-1}}$
  $r_k \leftarrow Aq_k$
  $\hat{r}_k \leftarrow A^H\hat{q}_k$
  $\alpha_k \leftarrow \langle \hat{q}_k, r_k\rangle = \langle \hat{r}_k, q_k\rangle$
  $r_k \leftarrow r_k - \alpha_k q_k - \gamma_{k-1}q_{k-1}$
  $\hat{r}_k \leftarrow \hat{r}_k - \overline{\alpha_k}\hat{q}_k - \overline{\beta_{k-1}}\hat{q}_{k-1}$
**end for**

Algorithm 3.7: Lanczos method (non-symmetric variant)

We have some freedom in choosing the off-diagonal elements $\beta_k$ and $\gamma_k$. All we have to ensure is that the relation

$$\beta_k\gamma_k = \hat{r}_k^H r_k$$

holds true. The most common choices, and their advantages and disadvantages, are collected in the following remark:

**Remark 3.38** a) Setting

$$\beta_k = \|r_k\|, \quad \gamma_k = \frac{\hat{r}_k^H r_k}{\beta_k},$$

ensures that the columns of $Q_k$ have unit length. The columns of $\hat{Q}_k$ can vary in magnitude.

b) Setting

$$\beta_k = \sqrt{|\hat{r}_k^H r_k|}, \quad \gamma_k = \frac{\hat{r}_k^H r_k}{\beta_k},$$

ensures that the columns of $Q_k$ multiplied by $|\gamma_0|$ and the columns of $\hat{Q}_k$ multiplied by $|\beta_0|$ have the same length.

c) Setting

$$\beta_k = \|r_k\|, \quad \gamma_k = \|\hat{r}_k\|, \quad \omega_k = \hat{r}_k^H r_k$$

with additional diagonal matrix $\Omega_k$ ensures that the columns of $Q_k$ *and* the columns of $\hat{Q}_k$ have unit length.

This last variant leads to a new algorithm. This algorithm uses the Lanczos decomposition

$$\begin{aligned} AQ_k - Q_k\Omega_k^{-1}T_k &= \beta_k q_{k+1}e_k^T \\ A^H\hat{Q}_k - \hat{Q}_k\left(\overline{\Omega_k}\right)^{-1}T_k^H &= \overline{\gamma_k}\hat{q}_{k+1}e_k^T, \end{aligned}$$

obviously with different tridiagonal $T_k$ given by

$$T_k = \begin{pmatrix} \alpha_1 & \omega_2\gamma_1 & & \\ \omega_2\beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \omega_k\gamma_{k-1} \\ & & \omega_k\beta_{k-1} & \alpha_k \end{pmatrix}.$$

Projection leads to the pencil

$$(A, I) \to (\hat{Q}_k^H A Q_k, \hat{Q}_k^H Q_k) = (\Omega_k\Omega_k^{-1}T_k, \Omega_k) = (T_k, \Omega_k).$$

We have to solve a general eigenvalue problem (a matrix pencil). This variant is due to Day. The implicit scaling can also be used to scale the vectors differently.

The three choices have no influence on the breakdown of the method:

**Theorem 3.39 (breakdown of the Lanczos method)** *The non-symmetric Lanczos method breaks down in case*

$$\hat{r}_{m+1}^H r_{m+1} = 0.$$

*Two cases of breakdown can be distinguished. The first is the case of a* lucky breakdown

$$\hat{r}_{m+1} = 0 \quad \textit{or} \quad r_{m+1} = 0.$$

*In this case we have computed a left or right invariant subspace of $A$. All eigenvalues of $T_m$ are eigenvalues of $A$. Additionally we can compute the corresponding left or right eigenvectors.*

*The second is the case of a* serious breakdown

$$\hat{r}_{m+1}^H r_{m+1} = 0 \quad \textit{and} \quad \hat{r}_{m+1} \neq 0 \neq r_{m+1}.$$

*In this case the algorithm can no longer be continued using short-term recurrences.*

The residual of an un-normalised Ritz pair can be measured with the use of computed quantities:

$$Ay_j - y_j\theta_j = \beta_k q_{k+1} s_{kj}.$$

Since $Q_k$ in general will be non-orthogonal, this gives only a crude estimation of actual convergence. Nevertheless the right-hand side still gives the backward error of the un-normalised Ritz pair.

**Remark 3.40** We have a one-to-one correspondence between Krylov subspaces built with $A^H$ and Krylov subspaces built with $A^T$. By inspection

$$\mathcal{K}_m\left(A^H, \hat{q}\right) = \overline{\mathcal{K}_m\left(A^T, \bar{\hat{q}}\right)}.$$

Under mild circumstances this holds true even numerically.

Spanning the second Krylov space with $A^T$ instead of $A^H$ does not impose restrictions on the class of possible algorithms.

It is possible to extend the non-symmetric Lanczos method to deal with serious breakdowns. This can be done by threshold pivoting in computing the LDMT decomposition of the Hankel matrix.

Methods that try to cope with the occurrence of serious breakdowns are known as *Lanczos with look-ahead*. Look-ahead leads to *block tridiagonal matrices* $T_k$ with varying block-size. A blocked variant of the non-symmetric Lanczos method named ABLE has been considered by Bai, Day and Ye (cf. [BDY99]).

The symmetric Lanczos may be extended by using so-called J-Symmetry. Whenever we have a relation of type

$$AJ = JA^H \quad \text{or} \qquad AJ = JA^T$$

we can construct the second basis without using $A^H$ or $A^T$. Explicit knowledge of the matrix $J$ allows to generalise the symmetric Lanczos method to a method for symplectic $A$, skew-symmetric $A$ and so forth.

## 3.6 Krylov Methods for Solving Linear Systems

Krylov subspace methods seek approximations from a Krylov subspace. Suppose that we already have an approximation to $A^{-1}b$, i.e. $\tilde{x} \approx A^{-1}b$. If such pre-information is lacking, set $\tilde{x} = 0$.

We can think of a transformation of the linear system to the new linear system

$$Ax = r_0 \equiv b - A\tilde{x}.$$

The solution of the system $Ay = b$ is given by $y = \tilde{x} + x$. We choose the starting vector $q = q_1$ in direction of the residual $r_0 = b - A\tilde{x}$.

Suppose we have chosen a Hessenberg decomposition

$$\begin{aligned} AQ_m &= Q_m C_m, \\ AQ_k &= Q_{k+1}\underline{C}_k = Q_k C_k + q_{k+1} c_{k+1,k} e_k^T \quad \forall\ k < m \end{aligned}$$

of the Krylov subspace $\mathcal{K}_m(A, q)$.

The $k$th approximation $x_k \in \mathcal{K}_k$ for $x = A^{-1}r_0$ can be expressed in terms of the new basis,

$$x_k = Q_k z_k, \quad z_k \in \mathbb{K}^k.$$

The task is to find a computable expression for $z_k$.

There are essentially two approaches. The first is related to the matrix $C_k$,

$$\begin{aligned} -r_k = Ax_k - r_0 &= AQ_k z_k - Q_k e_1 \|r_0\| \\ &= Q_k(C_k z_k - e_1\|r_0\|) + q_{k+1} c_{k+1,k} z_{kk}. \end{aligned}$$

We partition the pseudo-inverse of $Q_{k+1}$ as follows,

$$Q_{k+1}^\dagger Q_{k+1} = \begin{pmatrix} \hat{Q}_k^H \\ \hat{q}_{k+1}^H \end{pmatrix} [Q_k, q_{k+1}] = I_{k+1}.$$

We apply $\hat{Q}_k^H$ to the left. This yields

$$-\hat{Q}_k^H r_k = C_k z_k - e_1 \|r_0\|.$$

When we set

$$z_k = C_k^{-1} e_1 \|r_0\|,$$

we annihilate the projected residual. This is a Galerkin approach. To be more precise, we projected the linear system onto a smaller linear system

$$\begin{aligned}(A, r_0) &\rightarrow (AQ_k, Q_k \|r_0\| e_1) \\ &\rightarrow (\hat{Q}_k^H A Q_k, \hat{Q}_k^H Q_k \|r_0\| e_1) = (C_k, \|r_0\| e_1),\end{aligned}$$

using a second matrix $\hat{Q}_k$ (bi)orthogonal to $Q_k$.

The $k$th residual $r_k$ of the transformed system $Ax = r_0$ is the residual of the original system $Ay = b$, since

$$r_k = r_0 - Ax_k = b - A(\tilde{x} + x_k) = b - Ay_k.$$

Here we used the short-hand notation $y_k \equiv \tilde{x} + x_k$.

When the spaces spanned by $Q_k$ and $\hat{Q}_k$ are equal, this is known as a *Bubnov-Galerkin* approach. When the spaces spanned by $Q_k$ and $\hat{Q}_k$ are different, this is known as a *Petrov-Galerkin* approach.

In case of a Bubnov-Galerkin approach usually the same basis is used, i.e., $\hat{Q}_k = Q_k$. Furthermore usually an orthonormal basis is chosen.

**Definition 3.41** Methods using the Bubnov Galerkin approach with orthonormal $Q_k$ are termed *OR* (orthogonal residual) methods for obvious reasons.

Methods using the Petrov-Galerkin approach with bi-orthonormal or dual $\hat{Q}_k$ and $Q_k$ are termed *QOR* (quasi-orthogonal residual) methods.

We remark that when $A$ is HPD, we can interpret the OR orthogonality

$$\langle Ax_k - r_0, y \rangle = 0 \quad \forall\ y \in \mathcal{K}_k$$

as minimisation in the $A$-norm.

The second approach is related to the matrix $\underline{C}_k$. For this reason we re-name the coordinate vector, the approximate solution and the residual $\underline{z}_k$, $\underline{x}_k$, $\underline{r}_k$, respectively,

$$\begin{aligned}-\underline{r}_k = A\underline{x}_k - r_0 &= AQ_k \underline{z}_k - Q_{k+1} e_1 \|r_0\| \\ &= Q_{k+1}(\underline{C}_k \underline{z}_k - e_1 \|r_0\|).\end{aligned}$$

When we set

$$\underline{z}_k = \underline{C}_k^{\dagger} e_1 \|r_0\|,$$

we minimise the residual of the small overdetermined system

$$\underline{C}_k \underline{z}_k - e_1 \|r_0\|.$$

This approach corresponds to the projection of the linear system onto the overdetermined system

$$\begin{aligned}(A, r_0) &\rightarrow (AQ_k, Q_k \|r_0\| e_1) \\ &\rightarrow (\hat{Q}_{k+1}^H A Q_k, \hat{Q}_{k+1}^H Q_k \|r_0\| e_1) = (\underline{C}_k, \|r_0\| e_1).\end{aligned}$$

Afterwards the solution with minimal residual is computed.

This minimisation does not necessarily minimise the true residual, since we only have validity of the following,

$$\| - \underline{r}_k \| = \|A\underline{x}_k - r_0\| \leq \|Q_{k+1}\| \|\underline{C}_k \underline{z}_k - e_1 \|r_0\|\|.$$

**Definition 3.42** Methods using the projection onto an overdetermined system and the minimisation approach as stated above are termed *QMR* (quasi-minimal residual) methods.

When the matrix $Q_k$ is orthonormal, the true residual is minimised. QMR methods using an orthonormal basis $Q_k$ are termed *MR* (minimal residual) methods.

The MR approach computes $z_k$ with

$$\|AQ_k z_k - r_0\| = \min_{z \in \mathbb{K}^k} \|AQ_k z - r_0\|.$$

We remark that this condition is equivalent to the residual being orthogonal to $A\mathcal{K}_k$, i.e., to the QOR approach

$$r_0 - Ax_k \perp A\mathcal{K}_k(A, q).$$

The relations between OR and MR, or more general between QOR and QMR, have been considered in more detail. We state two approaches.

**Remark 3.43** When changing the inner product, MR (QMR) in general can be interpreted as special form of OR (QOR) (cf. [EE99]).

When applying residual smoothing to an OR (QOR) method, we obtain a MR (QMR) method (cf. [Wei94, SW95]).

The OR and QOR approaches rely on the iterated solution of small systems with system matrix $C_k$. The Hessenberg structure of $C_k$ admits the computation of a decomposition along with the computation of the coefficients. We can compute for example the LR decomposition

$$C_k = B_k R_k, \quad c_{k+1,k} = b_{k+1,k} r_{kk}.$$

Note that $B_k$ is bidiagonal since $C_k$ is Hessenberg. We define the matrix of *direction vectors* $P_m = Q_m R_m^{-1}$. The Hessenberg decomposition can be transformed to read

$$\begin{aligned} AP_m &= Q_m B_m, \\ AP_k &= Q_k B_k + q_{k+1} b_{k+1,k} e_k^T \quad \forall\ k < m. \end{aligned}$$

This is a Krylov decomposition with different bases, in contrast to the Krylov decompositions used for the solution of the eigenproblem.

The Krylov decomposition and the relation $Q_k = P_k R_k$ define *two coupled recurrences* in the bases $Q_k$ and $P_k$. This approach supposes knowledge of the recurrence parameters in the triangular matrices $B_k$ and $R_k$.

**Definition 3.44** We define a *split Hessenberg decomposition* to be a sequence of matrix equations that can be expressed as a single Hessenberg decomposition.

Due to the one-to-one correspondence between Krylov and polynomial spaces we can express the approximate solutions $x_k$ also using polynomials and the Krylov matrix,

$$x_k \in \mathcal{K}_k \quad \Leftrightarrow \quad x_k = K_k s_k \quad \Leftrightarrow \quad x_k = p_k(A)q.$$

The residual can also be expressed as polynomial, since it lies in a Krylov space,

$$r_k = r_0 - Ax_k = K_k \|r_0\| e_1 - AK_k s_k = K_{k+1} \begin{pmatrix} \|r_0\| \\ -s_k \end{pmatrix} \in \mathcal{K}_{k+1}.$$

We remark that this offers another possibility to expand the Krylov space. Instead of expanding the basis with $Aq_k$, we can use the $k$th residual $r_k$.

**Definition 3.45** The polynomials $\rho_k$ defined by

$$\frac{r_k}{\|r_0\|} = \rho_k(A)q = (1 - A\zeta_k(A))\, q,$$

are termed *residual polynomials*.

The residual polynomials by derivation fulfil the consistency condition

$$\rho_k(0) = 1, \quad \text{i.e.,} \quad \rho_k \in \mathbb{P}_k^0.$$

The QOR approach may be stated as

$$0 = \hat{Q}_k^H r_k = \hat{Q}_k^H \rho_k(A)q, \quad \rho_k \in \mathbb{P}_k^0,$$

and the QMR approach may be stated as

$$\|\hat{Q}_k^H r_k\| = \|\hat{Q}_k^H \rho_k(A)q\| = \min \|\hat{Q}_k^H \rho(A)q\| \quad \rho \in \mathbb{P}_k^0.$$

In the following we tried to use established standard forms of the algorithms we consider. The presentation of the algorithms was inspired by the books of Saad, Greenbaum, Golub/van Loan and Voß (cf. [Saa92, Saa96, Gre97, GvL96, Voß93]).

### 3.6.1 Richardson Iteration and Polynomial Acceleration

The Richardson iteration for the iterative solution of

$$Ax = r_0 = b - A\tilde{x}$$

is usually stated in its fixed-point form

$$x_{k+1} = (I - A)x_k + r_0.$$

This may be transformed (via $-r_k = Ax_k - r_0$) to read

$$\begin{aligned} AX_k &= X_{k+1}\underline{B}_k + R_0, \\ -R_k &= X_{k+1}\underline{B}_k = X_k B_k - x_{k+1}e_k^T \end{aligned}$$

where

$$B_k = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}.$$

This builds a Krylov subspace, the basis is composed of the residual vectors. It is an unusual representation of the basis expansion, because we have hidden the implicit action of $A$.

Nevertheless we obtain a matrix equation that has enough similarities to a Krylov decomposition to talk of a *generalised Krylov decomposition*. In place of the (general) Hessenberg matrix we obtain the bidiagonal matrix $B_k$.

When Chebychev acceleration (or more general, acceleration using orthogonal polynomials) is used, we obtain a similar relation with a *tridiagonal* matrix $T_k$ in place of the *bidiagonal* matrix $B_k$. This tridiagonal matrix is composed of the recurrence coefficients.

In case of Chebychev acceleration, where we assume the eigenvalues of $I - A$ in the open interval $(-1, 1)$, the recurrence looks like

$$-R_k D_k = X_{k+1}\underline{T}_k = X_k T_k - x_{k+1}e_k^T. \tag{3.29}$$

The tridiagonal matrix $T_k$ is composed of the recurrence coefficients of the three-term recurrence of the Chebychev polynomials. The matrix $T_k$ is given by

$$T_k = \begin{pmatrix} 1 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}.$$

The matrix $D_k$ on the left-hand side of equation (3.29) is a diagonal matrix. The non-zero elements are simply the diagonal elements of $T_k$. We may divide by this diagonal matrix of factors.

The recurrence (3.29) holds true in more generality. Consider a three-term recurrence for orthogonal polynomials with non-constant recurrence coefficients,

$$\begin{aligned} \phi_0 &= 1, \\ \phi_1 &= \gamma_1\lambda + (1-\gamma_1), \\ \phi_{k+1} &= \delta_k(\gamma_{k+1}\lambda + (1-\gamma_{k+1}))\phi_k \\ &+ (1-\delta_k)\phi_{k-1}, \quad \forall\ k \in \mathbb{N}. \end{aligned}$$

Accelerating the Richardson iteration with this orthogonal polynomials we obtain a recurrence formula with the same structure as (3.29).

The tridiagonal matrix $T_k$ now is given by

$$T_k = \begin{pmatrix} 1 & 1-\delta_2 & & \\ -1 & \delta_2 & \ddots & \\ & \ddots & \ddots & 1-\delta_k \\ & & -1 & \delta_k \end{pmatrix}.$$

Additionally, we introduce the shorthand notation $\delta_1 = 1$. $D_k$ is again a diagonal matrix, the diagonal elements are given by the products $d_{jj} = \gamma_j\delta_j$, i.e., by the diagonal of $T_k$ multiplied by some residual damping factors $\gamma_j$.

Again we might transform the tridiagonal by diagonal scaling from the right to a tridiagonal with unit diagonal.

Chebychev acceleration uses $\gamma_k = 1$ for all $k$ and $\delta_k = 2$ for all $k > 1$.

More general forms of acceleration (i.e., using non-orthogonal polynomials) are possible. They lead to a recurrence with a Hessenberg matrix in place of the tridiagonal,

$$-R_kD_k = X_{k+1}\underline{H}_k \quad \Leftrightarrow \quad R_k = -X_{k+1}\underline{H}_kD_k^{-1} \tag{3.30}$$

The question that naturally arises is how to compute the acceleration parameters, i.e., the recurrence coefficients. It is obvious that the columns of $\underline{H}_kD_k^{-1}$ and thus of $\underline{H}_k$ have to sum up to zero, since the true solution has to be reproduced.

The following Krylov methods use acceleration parameters computed during the algorithm. Even though some of them will have a form like (3.30), all are derived using solely Krylov decompositions introduced before.

### 3.6.2 Orthores/Orthomin/Orthodir

Consider the Hessenberg decomposition

$$\begin{aligned} AQ_m &= Q_mC_m, \\ AQ_k &= Q_{k+1}\underline{C}_k = Q_kC_k + c_{k+1,k}q_{k+1}e_k^T \quad \forall\ k < m. \end{aligned} \tag{3.31}$$

In the last section we noted that Hessenberg matrices with zero column sums are useful in the solution process of linear systems. This usefulness can be extended to a certain class of Hessenberg decompositions. The constructive proof is as follows.

The system

$$y^T C_m = \alpha e_m^T$$

has a unique solution $y$, provided $C_m$ is non-singular. Ignoring last column we conclude that

$$y^T \underline{C}_{m-1} = 0.$$

If the $k$th component in the vector $y$ is zero, the matrix $Z$ obtained from $\underline{C}_{m-1}$ by deleting $k$th row must be singular. This matrix is block diagonal,

$$Z = \begin{pmatrix} C_{k-1} & \star \\ 0 & R \end{pmatrix}.$$

The second block $R$ is an upper triangular matrix with diagonal elements $c_{j+1,j}$, $k < j < m$. Since we assume $C_m$ to be an unreduced Hessenberg matrix, $C_{k-1}$ must be singular.

When all $C_k$ are non-singular, we can scale the matrix $C_m$ by $D = \mathrm{diag}(y)$, $C_m^{(0)} \leftarrow DC_mD^{-1}$. Suppose w.l.o.g. that we already have scaled the Hessenberg decomposition, hence

$$e^T \underline{C}_{m-1}^{(0)} = 0, \quad \Leftrightarrow \quad e^T C_k^{(0)} = -c_{k+1,k}^{(0)} e_k^T \quad \forall\ k < m. \tag{3.32}$$

We restate (3.31) in form

$$AQ_k = Q_{k+1}\underline{C}_k^{(0)} = Q_k C_k^{(0)} - q_{k+1}e^T C_k^{(0)} = \left(Q_k - q_{k+1}e^T\right) C_k^{(0)}. \tag{3.33}$$

Furthermore we choose $\alpha$ such that $q_1 = r_0$. Then the OR approach corresponds to

$$z_k = \left(C_k^{(0)}\right)^{-1} e_1.$$

With $x_k = Q_k z_k$ we obtain

$$-r_k = Ax_k - r_0 = c_{k+1,k}^{(0)} q_{k+1} e_k^T z_k = -q_{k+1} e^T C_k^{(0)} \left(C_k^{(0)}\right)^{-1} e_1 = -q_{k+1},$$

i.e., the vectors $q_k$ are just the residuals $r_{k-1}$. For this reason we re-name the scaled basis vectors $r_0, \ldots, r_{m-1}$ instead of $q_1, \ldots, q_m$,

$$\begin{aligned} AR_m &= R_m C_m^{(0)}, \\ AR_k &= R_{k+1}\underline{C}_k^{(0)} = R_k C_k^{(0)} - r_k e^T C_k^{(0)} \\ &= \left(R_k - r_k e^T\right) C_k^{(0)} \quad \forall\ k < m. \end{aligned} \tag{3.34}$$

Note that we introduced the trivial approximation $x_0 = 0$. Next consider equation (3.34) multiplied from the left by $A^{-1}$,

$$\begin{aligned} R_k &= A^{-1} R_{k+1} \underline{C}_k^{(0)} \\ &= [x - x_0, \ldots, x - x_k]\, \underline{C}_k^{(0)} \\ &= \left(xe^T - X_{k+1}\right) \underline{C}_k^{(0)}, \\ R_k &= -X_{k+1} \underline{C}_k^{(0)} \quad \forall\ k < m. \end{aligned} \tag{3.35}$$

Equations (3.34) and (3.35) together form the basis of a method known as Orthores. As we can see, Orthores consists of two long-term recurrences. Orthores was first derived by Young and Jea (cf. [YJ80]).

Equation (3.35) reveals that Orthores can be interpreted as polynomial accelerated Richardson iteration. The acceleration parameters are computed by the algorithm and do not have to be supplied by the user.

The method can be modified realizing that Orthores uses LDMT decomposition of already scaled $C_k$. Considering equations (3.32) gives

$$\begin{pmatrix} 1 & & & 0 \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 1 \end{pmatrix} C_k^{(0)} \equiv D_k M_k^H.$$

where $D_k = -\mathrm{diag}(c_{2,1}^{(0)}, \ldots, c_{k+1,k}^{(0)})$ and $M_k^H$ is upper triangular with unit diagonal. Realize further that

$$\begin{pmatrix} 1 & & & 0 \\ 1 & 1 & & \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \cdots & 1 \end{pmatrix} = L_k^{-1} = \begin{pmatrix} 1 & & & 0 \\ -1 & \ddots & & \\ & \ddots & 1 & \\ 0 & & -1 & 1 \end{pmatrix}^{-1}.$$

Thus we have computed the LDMT decompositions of $C_k^{(0)}$,

$$C_k^{(0)} = L_k D_k M_k^H,$$

where $D_k$ is diagonal and both $L_k$ and $M_k$ are triangular with unit diagonal. We re-write the Hessenberg decomposition,

$$\begin{aligned} AR_m &= R_m L_m D_m M_m^H \\ AR_m M_m^{-H} D_m^{-1} &= R_m L_m. \end{aligned}$$

Defining the *direction vectors* $p_0, \ldots, p_{m-1}$ already mentioned before,

$$P_m = R_m M_m^{-H} \tag{3.36}$$

we obtain the Krylov decomposition

$$\begin{aligned} AP_m D_m^{-1} &= R_m L_m \\ AP_k D_k^{-1} &= R_{k+1} \underline{L}_k = R_k L_k - r_k e_k^T. \end{aligned} \tag{3.37}$$

We restate (3.36) in form

$$R_k = P_k M_k^H \quad \forall\ k \in \underline{m}. \tag{3.38}$$

The update of the approximations $x_k$ is given by

$$P_k D_k^{-1} = -X_{k+1} \underline{L}_k = -X_k L_k + x_k e_k^T \quad \forall\ k < m. \tag{3.39}$$

The coupled recurrences (3.37), (3.38) and (3.39) define the method known as Orthomin. Orthomin was first derived by Vinsome (cf. [Vin76]).

A third method can be obtained by consideration of a different scaling of a Hessenberg decomposition. Suppose w.l.o.g. that we have scaled a Hessenberg decomposition such that

$$\begin{aligned} AP_m &= P_m C_m, \\ AP_k &= P_{k+1} \underline{C}_k = P_k C_k + p_k e_k^T \quad \forall\ k < m. \end{aligned} \tag{3.40}$$

This corresponds to a scaling of the unreduced Hessenberg matrix to have a unit lower diagonal. The naming of the basis changes, because the basis vectors are used as direction vectors, which are usually named $p_0, p_1, \ldots, p_{m-1}$ instead of $q_1, q_2, \ldots, q_m$.

Suppose the solution $x = A^{-1}r_0$ of the linear system $Ax = r_0$ is in the Krylov space $\mathcal{K}_m$. Then it has a basis representation

$$x = x_m = P_m z_m, \quad z_m = (\alpha_0, \ldots, \alpha_{m-1})^T \in \mathbb{K}^m.$$

This shows that in principle we can update the approximate solutions $x_k$ via

$$x_{k+1} = x_k + \alpha_k p_k, \quad P_k D_\alpha = -X_{k+1}\underline{L}_k \quad \forall\ k < m. \tag{3.41}$$

The residuals satisfy

$$r_{k+1} = r_k - \alpha_k A p_k, \quad AP_k D_\alpha = R_{k+1}\underline{L}_k \quad \forall\ k < m. \tag{3.42}$$

The recurrences (3.40), (3.41) and (3.42) define the method known as Orthodir. Orthodir was first derived by Young and Jea (cf. [YJ80]). Another name for this method is GCR (Generalised Conjugate Residuals).

The difference between GCR and Orthomin is the difference between restart and truncation. The restarted variant is called GCR($l$), and the variant where the recurrence (3.40) is truncated is called Orthomin($l$).

We decided to give a general approach to these three methods, with all relations between them. Now we consider the implementational details.

Orthores is normally implemented using a scaled Arnoldi decomposition. The scaling just turns orthonormality into orthogonality. Orthores breaks down when one of the un-scaled Hessenberg matrices $H_k$ is singular.

In this case zero is in the field of values of $A$,

$$H_k z = 0 \quad \Rightarrow \quad z^H H_k z = z^H Q_k^H A Q_k z = y^H A y = 0.$$

Suppose that zero is outside the field of values of $A$. A breakdown in Orthores indicates we have computed the exact solution.

Orthores should remind of orthogonal residuals. Orthores using a scaled Arnoldi decomposition corresponds to the OR approach.

Orthomin uses the matrices $M_k^H$ in equation (3.38) and $D_k^{-1}$ in equations (3.37) and (3.39). Orthodir uses the matrices $C_k$ in equation (3.40) and $D_\alpha$ in equations (3.41) and (3.42).

For these quantities to be computable, Orthomin and Orthodir are normally implemented using the MR approach.

Suppose we are interested in minimising the residual in the 2-norm. Both methods perform in step $k+1$ a line search along search direction $p_k$. Locally minimising corresponds to

$$\|r_{k+1}\| = \|r_k - \alpha_k A p_k\| = \min.$$

The locally optimal $\alpha_k$ is given by

$$\alpha_k = \frac{\langle Ap_k, r_k\rangle}{\langle Ap_k, Ap_k\rangle}.$$

We remark that caution has to be taken in the field of complex values, since the function

$$f(\alpha) = \|r_k - \alpha A p_k\| = \langle r_k, r_k\rangle - \alpha\langle r_k, Ap_k\rangle - \overline{\alpha}\langle Ap_k, r_k\rangle + \alpha\overline{\alpha}\langle Ap_k, Ap_k\rangle$$

is not holomorphic. Using Wirtinger derivatives we obtain the unique solution,

$$\begin{aligned} \frac{d}{d\alpha} f(\alpha) &= -\langle r_k, Ap_k\rangle + \overline{\alpha}\langle Ap_k, Ap_k\rangle, \\ \frac{d}{d\overline{\alpha}} f(\alpha) &= -\langle Ap_k, r_k\rangle + \alpha\langle Ap_k, Ap_k\rangle. \end{aligned}$$

When all $Ap_j$ are orthogonal, local minimisation is sufficient to compute the global minimal residual.

This leads to choosing $M_k^H$ and $\underline{C}_k$ such that we obtain orthogonal $Ap_j$, or equivalently, $A^HA$-orthogonal $p_j$,

$$\langle p_i, p_j\rangle_{A^HA} = \langle p_i, A^HAp_j\rangle = \langle Ap_i, Ap_j\rangle = \delta_{ji}\langle Ap_j, Ap_j\rangle.$$

We denote the elements of $M_{k+1}^H$ by

$$M_{k+1}^H = \begin{pmatrix} 1 & \beta_{01} & \cdots & \beta_{0,k} \\ & 1 & \ddots & \vdots \\ & & \ddots & \beta_{k-1,k} \\ & & & 1 \end{pmatrix}.$$

In other words, we re-write the update formula (3.38) for Orthomin with shifted index and last column equated in the more convenient form

$$p_k = r_k - \sum_{j=0}^{k-1} p_j\beta_{jk}.$$

Setting

$$\beta_{jk} = \frac{\langle Ap_j, Ar_k\rangle}{\langle Ap_j, Ap_j\rangle}$$

leads to the desired orthogonal $Ap_j$ (the $A^HA$-orthogonal $p_j$). Setting the diagonal matrix $D_k^{-1} \equiv D_\alpha$ with $\alpha$ as chosen above, we have obtained the usual form of Algorithm 3.8 of Orthomin.

**input** : $A$, $b$, $\tilde{x}$
**output**: $L_k$, $D_k$, $M_k^H$, $R_k$, $P_k$, $X_k$

$r_0 \leftarrow b - A\tilde{x}, \quad p_0 \leftarrow r_0, \quad x_0 \leftarrow \tilde{x}$
**for** $k \in \mathbb{N}$ **do**
  $\alpha_{k-1} \leftarrow \langle Ap_{k-1}, r_{k-1}\rangle / \langle Ap_{k-1}, Ap_{k-1}\rangle$
  $x_k \leftarrow x_{k-1} + \alpha_{k-1}p_{k-1}$
  $r_k \leftarrow r_{k-1} - \alpha_{k-1}Ap_{k-1}$
  **for** $j < k$ **do**
    $\beta_{jk} \leftarrow \langle Ap_j, Ar_k\rangle / \langle Ap_j, Ap_j\rangle$
  **end for**
  $p_k \leftarrow r_k - \sum_{j=0}^{k-1} p_j\beta_{jk}$
**end for**

Algorithm 3.8: Orthomin

The update for the residuals shows that by induction

$$r_k = r_j - \sum_{i=j}^{k-1} \alpha_i Ap_i = r_0 - \sum_{i=0}^{k-1} \alpha_i Ap_i$$

holds true. Since the direction vectors are $A^HA$-orthogonal, we see that by applying $Ap_j$ from the left,

$$\langle Ap_j, r_k\rangle = \langle Ap_j, r_j - \sum_{i=j}^{k-1} \alpha_i Ap_i\rangle = \langle Ap_j, r_j\rangle - \alpha_j\langle Ap_j, Ap_j\rangle = 0 \quad \forall\ j < k.$$

The last equality follows by definition of $\alpha_j$. Especially we have the identity

$$\alpha_k = \frac{\langle Ap_k, r_k\rangle}{\langle Ap_k, Ap_k\rangle} = \frac{\langle A(r_k - \sum_{j=0}^{k-1} p_j\beta_{jk}), r_k\rangle}{\langle Ap_k, Ap_k\rangle} = \frac{\langle Ar_k, r_k\rangle}{\langle Ap_k, Ap_k\rangle}.$$

This expression can be used as an alternative means of computation of $\alpha_k$. There is also an alternative expression for the $\beta_{kj}$,

$$\begin{aligned}\beta_{kj} &= \frac{\langle Ap_j, Ar_k\rangle}{\langle Ap_j, Ap_j\rangle} \\ &= \frac{1}{\overline{\alpha_j}}\frac{\langle r_j - r_{j+1}, Ar_k\rangle}{\langle Ap_j, Ap_j\rangle} \\ &= \frac{\langle Ap_j, Ap_j\rangle}{\langle r_j, Ar_j\rangle}\frac{\langle r_j - r_{j+1}, Ar_k\rangle}{\langle Ap_j, Ap_j\rangle} \\ &= \frac{\langle r_j - r_{j+1}, Ar_k\rangle}{\langle r_j, Ar_j\rangle}.\end{aligned}$$

We observe that the $k$th residual is orthogonal to $A$ times the old residuals,

$$\langle A^H r_k, r_j\rangle = \langle r_k, Ar_j\rangle = \langle r_k, A(p_j + \sum_{i=0}^{j-1} p_i\beta_{ij})\rangle = 0 \quad \forall\ j < k.$$

From now on we focus on Orthodir. We denote the non-trivial elements of $C_k$ by $\beta_{ij}$,

$$C_k = \begin{pmatrix} \beta_{00} & \beta_{01} & \cdots & \beta_{0k} \\ 1 & \beta_{11} & \cdots & \beta_{1k} \\ & \ddots & \ddots & \vdots \\ & & 1 & \beta_{kk} \end{pmatrix}.$$

We re-write the update formula (3.40) for Orthodir in the same manner we already did for Orthomin,

$$p_k = Ap_{k-1} - \sum_{j=0}^{k-1} p_j\beta_{jk}.$$

The only difference between the update formulae of the direction vectors $p_j$ in Orthomin and Orthodir is the choice of the next vector to be orthogonalised. In case of Orthomin we choose the residual $r_k$, in case of Orthodir we choose $Ap_{k-1}$.

In this case we achieve orthogonal $Ap_j$ ($A^HA$-orthogonal $p_j$) by setting

$$\beta_{jk} = \frac{\langle Ap_j, A^2p_{k-1}\rangle}{\langle Ap_j, Ap_j\rangle}.$$

Orthodir should remind of orthogonal (search) directions. We just obtained the convenient form of Orthodir given by Algorithm 3.9.

All three algorithms more or less *implicitly* use a Krylov decomposition. Two of them, namely Orthomin and Orthores, break down when one of the computed matrices becomes singular.

A natural idea is to *explicitly* use a Krylov decomposition. This is done by the methods introduced in the next section.

### 3.6.3 FOM/GMRES

Consider the case that we explicitly compute the Arnoldi decomposition

$$\begin{aligned}AQ_m &= Q_mH_m, \\ AQ_k &= Q_{k+1}\underline{H}_k = Q_kH_k + h_{k+1,k}q_{k+1}e_k^T \quad \forall\ k < m\end{aligned}$$

**input** : $A$, $b$, $\tilde{x}$
**output**: $C_k^{(0)}$, $L_k$, $D_k$, $R_k$, $P_k$, $X_k$

$r_0 \leftarrow b - A\tilde{x}, \quad p_0 \leftarrow r_0 \quad x_0 \leftarrow \tilde{x}$
**for** $k \in \mathbb{N}$ **do**
  $\alpha_{k-1} \leftarrow \langle Ap_{k-1}, r_{k-1}\rangle / \langle Ap_{k-1}, Ap_{k-1}\rangle$
  $x_k \leftarrow x_{k-1} + \alpha_{k-1} p_{k-1}$
  $r_k \leftarrow r_{k-1} - \alpha_{k-1} Ap_{k-1}$
  **for** $j < k$ **do**
    $\beta_{jk} \leftarrow \langle Ap_j, A^2 p_{k-1}\rangle / \langle Ap_j, Ap_j\rangle$
  **end for**
  $p_k \leftarrow Ap_{k-1} - \sum_{j=0}^{k-1} p_j \beta_{jk}$
**end for**

Algorithm 3.9: Orthodir

and directly use the OR and MR approaches.

We first consider the OR approach. We have to find $x_k = Q_k z_k$ with orthogonal residual,

$$H_k z_k - e_1 \|r_0\| = Q_k^H (Ax_k - r_0) = -Q_k^H r_k = 0.$$

The Hessenberg structure of $H_k$ enables us to compute a Givens QR decomposition

$$H_k = U_k^H R_k, \qquad U_k H_k = R_k$$

of the matrices $H_k$ iteratively. The orthogonal matrix $U_k$ is composed of the Givens rotations,

$$U_k = \prod_{j=1}^{k-1} \begin{pmatrix} I_{j-1} & & 0 \\ & G_j & \\ 0 & & I_{k-j-1} \end{pmatrix}, \qquad G_j = \begin{pmatrix} c_j & s_j \\ s_j & -c_j \end{pmatrix}$$

We only need to compute one Givens rotation per step. Afterwards the system is solved by applying the Givens rotations to the left-hand side and backward substitution,

$$R_k z_k = \|r_0\| U_k e_1.$$

The resulting algorithm is Algorithm 3.10.

**input** : $A$, $r_0$
**output**: $H_k$, $Q_k$, $Z_k$, $X_k$

**for** $k \in \mathbb{N}$ **do**
  $h_{k,k-1} \leftarrow \|r_{k-1}\|$
  $q_k \leftarrow r_{k-1} / h_{k,k-1}$
  $r_k \leftarrow Aq_k$
  **for** $j \in \underline{k}$ **do**
    $h_{jk} \leftarrow \langle q_j, r_k\rangle$
    $r_k \leftarrow r_k - q_j h_{jk}$
  **end for**
  Solve $H_k z_k = \|r_0\| e_1$
  $x_k \leftarrow Q_k z_k$
**end for**

Algorithm 3.10: FOM

This method was developed by Saad and Schultz (cf. [SS86]) and is known as FOM (Full Orthogonalisation Method).

The MR approach is handled similar. We have to find $\underline{x}_k = Q_k \underline{z}_k$ with minimal residual,

$$\|\underline{H}_k \underline{z}_k - e_1 \|b\|\| = \|A\underline{x}_k - b\| = \min$$

The solution of this minimal residual problem is again computed using an orthogonal decomposition

$$\underline{H}_k = U_{k+1}^H \underline{W}_k, \quad \underline{W}_k = U_{k+1} \underline{H}_k$$

of $\underline{H}_k$, apparently with the same Givens rotations,

$$U_{k+1} = \prod_{j=1}^{k} \begin{pmatrix} I_{j-1} & & 0 \\ & G_j & \\ 0 & & I_{k-j} \end{pmatrix}, \quad G_j = \begin{pmatrix} c_j & s_j \\ s_j & -c_j \end{pmatrix}.$$

The decompositions can easily be updated using the old decomposition. Only the last column vector of $H_k$ is influenced by the next Givens rotation.

Algorithm 3.11 stores the elements of the QR decomposed Hessenberg matrices in the upper triangular matrices $\underline{W}_k$, but only the upper part $W_k$ is used for the backward substitution. For simplicity the last column of $\underline{W}_k$ is denoted by $w_k$.

**input** : $A$, $r_0$
**output**: $\underline{H}_k$, $Q_{k+1}$, $G_k$, ...

$\alpha_1 \leftarrow \|r_0\|$
**for** $k \in \mathbb{N}$ **do**
  Compute $AQ_k = Q_{k+1}\underline{H}_k$
  $w_k \leftarrow \underline{H}_k e_k$
  **for** $j < k$ **do**
    $\begin{pmatrix} w_{j,k} \\ w_{j+1,k} \end{pmatrix} \leftarrow \begin{pmatrix} c_j & s_j \\ s_j & -c_j \end{pmatrix} \begin{pmatrix} w_{j,k} \\ w_{j+1,k} \end{pmatrix}$
  **end for**
  $\nu_k \leftarrow \sqrt{w_{kk}^2 + w_{k+1,k}^2}$
  $c_k \leftarrow w_{kk}/\nu_k$, $s_k \leftarrow w_{k+1,k}/\nu_k$
  $w_{kk} \leftarrow \nu_k$, $w_{k+1,k} \leftarrow 0$
  $\alpha_k \leftarrow c_k \alpha_k$, $\alpha_{k+1} \leftarrow s_k \alpha_k$
  **if** $\alpha_{k+1} \approx 0$ **then**
    Solve $W_k \underline{z}_k = (\alpha_1, \ldots, \alpha_k)^T$
    $\underline{x}_k \leftarrow Q_k \underline{z}_k$
  **endif**
**end for**

Algorithm 3.11: GMRES

This method was developed by Saad and Schultz (cf. [SS86]) and is known as GMRES (Generalised Minimal RESidual).

Both methods deliver the exact solution $x_m = x$ in case the Arnoldi recurrence breaks down, because

$$\underline{H}_m^\dagger = \begin{pmatrix} H_m^{-1} & 0 \end{pmatrix} \quad \text{and} \quad Ax_m - r_0 = Q_m \left( z_m - H_m^{-1} e_1 \|r_0\| \right) = 0.$$

In FOM some iterates may not be defined, since the matrix $H_k$ may be singular even though $A$ is regular. This can not occur when zero is outside the field of values of $A$.

We note that in both the OR and MR approach orthogonal decompositions using Givens rotations are used. The relation between these methods becomes more obvious when we look at the intimate relations between FOM and GMRES.

The first $k-1$ Givens matrices $G_j$ in both algorithms are the same. To be more precise,

$$\begin{pmatrix} U_k & 0 \\ 0 & 1 \end{pmatrix} \underline{H}_k = \begin{pmatrix} U_k H_k \\ h_{k+1,k} e_k^T \end{pmatrix} = \begin{pmatrix} R_k \\ h_{k+1,k} e_k^T \end{pmatrix}.$$

The next Givens rotation annihilates the non-zero element in the last row and has influences only on the last column.

We are interested in the residual of the MR solution $\underline{z}_k$. We observe that

$$\begin{aligned} U_{k+1} \left( \underline{H}_k \underline{z}_k - \|r_0\| e_1 \right) &= \underline{W}_k \underline{z}_k - \|r_0\| U_{k+1} e_1 \\ &= -\|r_0\| \left( s_1 \cdot \cdots \cdot s_k \right) e_{k+1}. \end{aligned}$$

The OR solution $z_k$ inserted into the minimal residual equation leads to

$$\begin{aligned} \begin{pmatrix} U_k & 0 \\ 0 & 1 \end{pmatrix} \left( \underline{H}_k z_k - \|r_0\| e_1 \right) &= \begin{pmatrix} R_k \\ h_{k+1,k} e_k^T \end{pmatrix} z_k - \|r_0\| \begin{pmatrix} U_k & 0 \\ 0 & 1 \end{pmatrix} e_1 \\ &= h_{k+1,k} e_k^T z_k e_{k+1}. \end{aligned}$$

The OR solution is given by

$$z_k = R_k^{-1} U_k e_1,$$

and thus

$$e_k^T z_k = e_k^T R_k^{-1} U_k \|r_0\| e_1 = r_{kk}^{-1} \left( s_1 \cdot \cdots \cdot s_{k-1} \right) \|r_0\|.$$

Note that the next Givens matrix $G_k$ annihilates the element $h_{k+1,k}$, i.e.

$$G_k \begin{pmatrix} r_{kk} \\ h_{k+1,k} \end{pmatrix} = \begin{pmatrix} \nu_k \\ 0 \end{pmatrix} \quad \Rightarrow \quad c_k = \frac{r_{kk}}{\nu_k}, \quad s_k = -\frac{h_{k+1,k}}{\nu_k}$$

where $\nu_k = \sqrt{r_{kk}^2 + h_{k+1,k}^2}$ is the 2-norm of the vector consisting solely of the non-zero elements of the last two rows. Thus we can express the residual as

$$\begin{aligned} \|r_0\| h_{k+1,k} e_k^T z_k e_{k+1} &= \|r_0\| \frac{h_{k+1,k}}{r_{kk}} \left( s_1 \cdot \cdots \cdot s_{k-1} \right) e_{k+1} \\ &= -\|r_0\| \frac{1}{c_k} \left( s_1 \cdot \cdots \cdot s_k \right) e_{k+1}. \end{aligned}$$

We know that both $Q_k$ and $U_k$ do not change the length of the residuals. This implies that as long $c_k \neq 0$, the residuals of FOM and GMRES fulfil

$$\|r_k\| = \frac{\|\underline{r}_k\|}{\sqrt{1 - \left( \|\underline{r}_k\| / \|\underline{r}_{k-1}\| \right)^2}}.$$

This result remains valid for methods of type QOR and QMR, when we consider the *quasi-residuals*

$$\zeta_k = \underline{C}_k z_k - \|r_0\| e_1 \quad \text{and } \underline{\zeta}_k = \underline{C}_k \underline{z}_k - \|r_0\| e_1$$

instead of the true residuals.

Mathematically we have equivalence between the methods Orthores and FOM, and between the methods Orthomin, Orthodir and GMRES. On the other hand FOM and GMRES are superior in both storage requirements and concerning numerical stability.

### 3.6.4 Truncated and Restarted Methods

The inherent complexity and work space requirements in the recurrences based on the Arnoldi decomposition are $O(m^2)$. These recurrences, the so-called long-term recurrences restrict the range of possible applications for those methods.

There are essentially two ideas to overcome complexity restrictions. First we can *truncate* the sequence, i.e., we only orthogonalise against $s$ terms. Secondly we can *restart* the method when necessary.

We first consider *restart techniques*. In *explicit restarts* the computed Krylov decomposition is discarded, and the last generated approximation is used as new starting vector.

In *implicit restarts* the computed Krylov decomposition is transformed to a smaller Krylov decomposition. Usually this is done by dividing the eigenvalues into good and bad ones; the bad ones are removed.

According to the number of vectors used in the restart we distinguish *thin* and *thick restart*. If only some vectors are used, we speak of thin restart, else of thick restart.

Methods using explicit thin restart are GCR($m$), FOM($m$), GMRES($m$). These methods just restart with the last approximation as starting vector.

In a thick restart we do not throw away all information, we use a portion of the knowledge obtained thus far. Often *implicit thick restart* is used, for example in ARPACK.

Now we loose some words on truncated methods. We already mentioned the truncated variant of Orthomin, namely Orthomin($m$).

Other methods of this class are IOM and DIOM (direct IOM). These methods are contained in the book of Saad ([Saa96]). IOM and DIOM use truncation of the orthogonalisation in FOM.

Also GMRES can be truncated, leading to Quasi-GMRES and DQGMRES (direct Quasi-GMRES). Also Quasi-GMRES and DQGMRES may be found in the book of Saad ([Saa96]).

New variants of truncated methods throw away only specific vectors, methods of this class are GCRO and GCROT. Here, like in the implicitly restarted methods, the eigenvalues and singular values play a major role.

The ideas of restart and truncation may be used simultaneously. This field is still developing, so we will not discuss restarted and truncated methods any further.

We just mention that both fit into our approach. The Krylov decompositions obtained are still Hessenberg decompositions. The upper triangular part of the Hessenberg matrices obtained changes, to be more precise, the matrices are *block-diagonal*, when restart is used, and *banded*, when truncation is used.

### 3.6.5 CG/CR

In this section we restrict ourselves to Hermitian $A$ and the symmetric Lanczos decomposition

$$\begin{aligned} AQ_m &= Q_m T_m, \\ AQ_k &= Q_{k+1}\underline{T}_k = Q_k T_k + \beta_k q_{k+1} e_k^T \quad \forall\ k < m. \end{aligned}$$

We consider the methods Orthores, Orthomin and Orthodir. They simplify to short recurrences.

We first consider the OR approach. The assembly of the Orthores variant is obvious. The Orthomin and Orthodir variants require further thinking. We only consider Orthomin.

The direction vectors in the Orthomin variant have a special property. Remember that they are computed via

$$P_m = R_m M_m^{-H}.$$

We observe that due to the orthogonality of the residuals, $D_\rho = R_m^H R_m$ is a diagonal matrix.

We apply the matrix of direction vectors to both sides of $A$,

$$P_m^H A P_m = M_m^{-1} R_m^H A R_m M_m^{-H} = M_m^{-1} R_m^H R_m C_m M_m^{-H} = M_m^{-1} D_\rho L_m D_m.$$

The result is a lower triangular matrix. Since $A$ is selfadjoint, this matrix is also upper triangular and thus a diagonal matrix. We can state this relation alternatively as

$$\langle p_i, A p_j \rangle = 0 \quad \forall\ i \neq j.$$

This orthogonality with respect to $A$ is known as $A$-conjugacy.

When $A$ is furthermore HPD (HND), $A^{-1}$ uniquely defines

$$\|x\|_{A^{-1}} \equiv \sqrt{\langle x, x \rangle_{A^{-1}}} \equiv \sqrt{\langle A^{-1} x, x \rangle},$$

a scalar-product and a norm, the $A^{-1}$-norm. The $A^{-1}$-norm of the residual, i.e., the $A$-norm of the error,

$$\|r_{k+1}\|_{A^{-1}} = \|A(x - x_{k+1})\|_{A^{-1}} = \|x - x_{k+1}\|_A$$

can be used in the local minimisation

$$\|r_k - \alpha_k A p_k\| = \min.$$

The optimal $\alpha_k$ in the $A^{-1}$-norm is given by

$$\alpha_k = \frac{\langle A p_k, r_k \rangle_{A^{-1}}}{\langle A p_k, A p_k \rangle_{A^{-1}}} = \frac{\langle p_k, r_k \rangle}{\langle A p_k, p_k \rangle}.$$

Usually the alternative formula for $\alpha_k$ is used,

$$\alpha_k = \frac{\langle A r_k, r_k \rangle_{A^{-1}}}{\langle A p_k, A p_k \rangle_{A^{-1}}} = \frac{\langle r_k, r_k \rangle}{\langle A p_k, p_k \rangle}.$$

The $A$-conjugacy fixes the choice of the $\beta_k \equiv -\beta_{k+1,k}$,

$$\beta_k = \frac{\langle r_{k+1} - r_k, A r_{k+1} \rangle_{A^{-1}}}{\langle r_k, A r_k \rangle_{A^{-1}}} = \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle}.$$

The method we just sketched here is known as the CG method, short, CG. The name CG stands for conjugate gradients. The reason is that CG is usually defined with the help of the auxiliary functional

$$f(x) = \frac{1}{2} x^H A x - x^H r_0.$$

This function is real differentiable and the condition for a stationary point is equivalent for $x$ to be a solution of the system, i.e.

$$f'(x) = Ax - r_0 = 0.$$

The residuals are the gradients of the functional, and the search directions are obtained by conjugating the residuals, thus conjugate gradients.

**input** : $A$, $b$, $x_0$
**output**: $R_k$, $P_k$, $L_k$, $D_k$, $M_k^H$

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$
**for** $k \in \mathbb{N}_0$ **do**
  $\alpha_k \leftarrow \langle r_k, r_k\rangle / \langle p_k, Ap_k\rangle$
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$
  $r_{k+1} \leftarrow r_k - \alpha_k Ap_k$
  $\beta_k \leftarrow \langle r_{k+1}, r_{k+1}\rangle / \langle r_k, r_k\rangle$
  $p_{k+1} \leftarrow r_{k+1} + \beta_k p_k$
**end for**

Algorithm 3.12: CG, Omin variant

The local minimisation in the $A^{-1}$-norm corresponds to the orthogonal approach of OR, i.e., CG is equivalent to Orthores and thus FOM. The striking advantage of using less matrix vector multiplies is superseded by the fact that CG now minimises the error like MR.

The Orthomin variant of CG, or short, the Omin variant, follows as Algorithm 3.12.

The MR approach is handled similarly. Orthomin for this class of matrices is known as CR (conjugate residual method). We only consider the Orthomin variant of CR. The algorithm for CR is given by Algorithm 3.13.

**input** : $A$, $b$, $x_0$
**output**: $R_k$, $P_k$, $L_k$, $D_k$, $M_k^H$

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$
**for** $k \in \mathbb{N}_0$ **do**
  $\alpha_k \leftarrow \langle Ar_k, r_k\rangle / \langle Ap_k, Ap_k\rangle$
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$
  $r_{k+1} \leftarrow r_k - \alpha_k Ap_k$
  $\beta_k \leftarrow \langle Ar_{k+1}, r_{k+1}\rangle / \langle Ar_k, r_k\rangle$
  $p_{k+1} \leftarrow r_{k+1} + \beta_k p_k$
**end for**

Algorithm 3.13: CR, Omin variant

The other methods, i.e., Orthores and Orthodir can also be applied. The resulting variants of CG and CR are not that well-known.

Also a variant using LR decomposition of the tridiagonal matrix of the Lanczos method is used. This method is named D-Lanczos (direct Lanczos) by Saad. It corresponds to a simple scaling in ordinary CG (the Omin variant).

Due to the tridiagonal structure of the matrix in the Lanczos method, the long-term recurrences of the general methods become short-term recurrences. We summarise the properties of the three possible implementations of CG (and also CR).

**Remark 3.46 (Differences in the recurrences)**

**Orthodir,** also known as Odir, is given by one *three-term recurrence* in the *direction vectors* $p_j$ and by two *two-term recurrences* in the *residual vectors* $r_j$ and in the *iterates* $x_j$.

Odir is *immune to breakdowns*, since the three-term recurrence still expands the search space when the iterates stagnate. Odir applies to all selfadjoint matrices.

**Orthomin,** also known as Omin, is given by three *two-term recurrences* in the *direction vectors* $p_j$, in the *residual vectors* $r_j$ and in the *iterates* $x_j$.

Omin *breaks down* when one $\alpha_j$ becomes zero, indicating that zero is in the field of values of $A$. Omin in general applies only to HPD (HND) matrices.

**Orthores,** also known as Ores, is given by two *three-term recurrences* in both the *residual vectors* $r_j$ and in the *iterates* $x_j$.

Ores is *equivalent to* Omin. This can be seen by obtaining Ores from Omin by removing the direction vector recurrence.

The connection between the Omin variant of CG and the symmetric Lanczos method in matrix form

$$AQ_k - Q_k T_k = M_k,$$

is given by

$$\begin{aligned} M_k &= \left(-\alpha_{k-1}^{\mathrm{CG}} \|r_{k-1}\|\right)^{-1} r_k e_k^T \\ &= \frac{\sqrt{\beta_{k-1}^{\mathrm{CG}}}}{-\alpha_{k-1}^{\mathrm{CG}}} q_{k+1} e_k^T \\ &\equiv \beta_k^{\mathrm{Lanczos}} q_{k+1} e_k^T, \\ \alpha_k^{\mathrm{Lanczos}} &= \frac{1}{\alpha_{k-1}^{\mathrm{CG}}} + \frac{\beta_{k-2}^{\mathrm{CG}}}{\alpha_{k-2}^{\mathrm{CG}}}. \end{aligned}$$

The original CG and CR methods (the Omin variants) were developed independently by Hestenes and Stiefel and published in ([HS52]). It is noteworthy that this paper was published roughly the same time than Lanczos both papers appeared.

The connection between CG and the Lanczos method can be found in the books of Householder (cf. [Hou75]) and Saad (cf. [Saa96]).

### 3.6.6 SymmLQ/MinRes

Odir is quite unstable, and due to the (implicit) triangular decomposition, the Omin and Ores variants apply only to $A$ HPD (HND).

A possible improvement is the use of pivoting or the Bunch-Kaufmann decomposition. The latter variant is known as *SymmBK*. We will not discuss pivoting or SymmBK any further.

Another and simpler cure is to proceed like in FOM/GMRES. We can compute the (symmetric) Arnoldi decomposition using the Lanczos method,

$$\begin{aligned} AQ_m &= Q_m T_m, \\ AQ_k &= Q_{k+1}\underline{T}_k = Q_k T_k + \beta_k q_{k+1} e_k^T \quad \forall\ k < m, \end{aligned} \tag{3.43}$$

and use orthogonal decomposition of $T_k$ ($\underline{T}_k$) afterwards to compute the OR and MR solutions.

We first consider the OR approach. The decomposition used is a LQ decomposition of $T_m$. We use a sequence of Givens rotations $G_j$,

$$G_j \equiv \begin{pmatrix} c_j & s_j \\ s_j & -c_j \end{pmatrix},$$

to erase the upper diagonal of $T_m$,

$$T_m = L_m U_m \equiv L_m \prod_{j=1}^{m-1} \begin{pmatrix} I_{j-1} & & \\ & G_j & \\ & & I_{m-j-1} \end{pmatrix}.$$

The lower triangular matrix $L_m$ has only three non-zero diagonals.

The Givens rotations used imply that the LQ decompositions of previous steps are given by

$$T_k = \tilde{L}_k U_k \equiv \tilde{L}_k \prod_{j=1}^{k-1} \begin{pmatrix} I_{j-1} & & \\ & G_j & \\ & & I_{k-j-1} \end{pmatrix}.$$

The next Givens rotation that decomposes $T_{k+1}$ only alters the last column of $\tilde{L}_k$, i.e., the diagonal element $\tilde{l}_{kk}$. All other elements are not altered by the following rotations, we have

$$\tilde{L}_k = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & \ddots & & \\ & \ddots & \ddots & l_{k-1,k-1} & \\ & & l_{k,k-2} & l_{k,k-1} & \tilde{l}_{kk} \end{pmatrix}, \quad l_{kk} = \sqrt{\tilde{l}_{kk}^2 + \beta_k^2}.$$

We define a new orthogonal basis $W_m$, given by

$$W_m = Q_m U_m^H.$$

Obviously the first $k-1$ columns of the bases,

$$\tilde{W}_k = \left[ W_{k-1}, \tilde{w}_k \right] = Q_k U_k^H,$$

are equal to the first $k-1$ columns of $W_m$.

Using this basis, the (symmetric) Arnoldi decomposition becomes

$$\begin{aligned} AW_m &= Q_m L_m \\ A\tilde{W}_k &= Q_k \tilde{L}_k + q_{k+1} \beta_k e_k^T U_k^H. \end{aligned}$$

The equations for the OR solution with unusual notation $\tilde{z}_k$

$$T_k \tilde{z}_k = \|r_0\| e_1,$$

is transformed to the equations

$$\tilde{L}_k z_k \equiv \tilde{L}_k U_k \tilde{z}_k = \|r_0\| e_1.$$

We denote the ultimate solution vector of

$$L_m z_m = \|r_0\| e_1 \quad \text{by} \quad z_m = \begin{pmatrix} \zeta_1 & \cdots & \zeta_m \end{pmatrix}^T.$$

We observe that due to the Givens decomposition

$$z_k = \begin{pmatrix} \zeta_1 & \cdots & \zeta_{k-1} & \tilde{\zeta}_k \end{pmatrix}^T, \quad \text{where} \quad c_k \tilde{\zeta}_k = \zeta_k.$$

The intermediate quantities may not be useful in representing the solutions, since $T_k$ and thus $\tilde{L}_k$ may be singular or badly conditioned. It is better to update the solution in terms of the columns of $W_k$,

$$x_k = W_k z_k = x_{k-1} + \zeta_k w_k.$$

The $k$th CG iterate is given by a small correction to $x_{k-1}$,

$$\begin{aligned} x_k^{\mathrm{CG}} &= x_{k-1}^{\mathrm{SymmLQ}} + \tilde{\zeta}_k \tilde{w}_k \\ &= x_{k-1}^{\mathrm{SymmLQ}} + \zeta_k \tilde{w}_k / c_k. \end{aligned}$$

The CG solution is computed solely in a last step when the desired accuracy has been reached. An implementation of SymmLQ where the computation of the CG solution occurs outside the main loop is given by Algorithm 3.14.

**input** : $A$, $b$, $x_0$
**output**: $T_k$, $L_k$, $G_j$, $Q_k$, ...

$r_0 \leftarrow b - Ax_0$, $q_0 \leftarrow 0$, $\beta_0 \leftarrow \|r_0\|$, $q_1 \leftarrow r_0/\beta_0$
$c_{-1} \leftarrow 1$, $s_{-1} \leftarrow 0$, $c_0 \leftarrow -1$, $s_0 \leftarrow 0$
$\zeta_{-1} \leftarrow 0$, $\zeta_0 \leftarrow -1$

**for** $k \in \mathbb{N}$ **do**
  Expand Arnoldi basis
    $r_k \leftarrow Aq_k$
    $\alpha_k \leftarrow \langle q_k, r_k \rangle$
    $r_k \leftarrow r_k - \alpha_k q_k - \beta_{k-1} q_{k-1}$
    $\beta_k \leftarrow \|r_k\|$
    $q_{k+1} \leftarrow r_k/\beta_k$
  Expand L-factor
    $\tilde{l}_{kk} \leftarrow -c_{k-1}\alpha_k - s_{k-1}c_{k-2}\beta_{k-1}$
    $l_{kk} \leftarrow \sqrt{\tilde{l}_{kk}^2 + \beta_k^2}$
    $l_{k,k-1} \leftarrow s_{k-1}\alpha_k - c_{k-1}c_{k-2}\beta_{k-1}$
    $l_{k,k-2} \leftarrow s_{k-2}\beta_{k-1}$
  Compute Givens rotation
    $c_k \leftarrow \tilde{l}_{kk}/l_{kk}$
    $s_k \leftarrow \beta_k/l_{kk}$
  Expand SymmLQ basis
    $w_k \leftarrow c_k \tilde{w}_k + s_k q_{k+1}$
    $\tilde{w}_{k+1} \leftarrow s_k \tilde{w}_k - c_k q_{k+1}$
  Expand $z_k \leftarrow (z_{k-1}, \zeta_k)$
    $\zeta_k \leftarrow -(l_{k,k-2}\zeta_{k-2} + l_{k,k-1}\zeta_{k-1})/l_{kk}$
    $x_k \leftarrow x_{k-1} + \zeta_k w_k$
**end for**
$x_m^{\mathrm{CG}} \leftarrow x_{m-1} + \zeta_m \tilde{w}_m / c_m$

Algorithm 3.14: SymmLQ

The MR approach, similar to GMRES, uses QR factorisation. The matrix $\underline{T}_k$ is factorised,

$$\underline{T}_k = U_{k+1}^H \underline{L}_k^H.$$

The resulting factors correspond to the factors obtained in SymmLQ. This is an implication of $T_k$ being Hermitian. The minimal residual solution

$$\underline{z}_k = \arg\min_{z \in \mathbb{K}^k} \|\underline{T}_k z - \|r_0\| e_1\| = \arg\min_{z \in \mathbb{K}^k} \|\underline{L}_k^H z - \|r_0\| U_{k+1} e_1\|$$

can be computed by backward substitution,

$$\underline{z}_k = L_k^{-H} \|r_0\| \left(I_{k,k+1} U_{k+1} e_1\right).$$

The left-hand side of this equation can be updated with the Givens rotations involved, only the last elements have to be stored.

We additionally introduce the matrices

$$V_k = Q_k L_k^{-H}.$$

This corresponds to a transformation of the symmetric Arnoldi decomposition

$$\begin{aligned} AQ_m &= Q_m T_m \\ AQ_k &= Q_{k+1}\underline{T}_k = Q_{k+1} U_{k+1}^H \underline{L}_k^H \end{aligned}$$

to a new Krylov decomposition in two bases. When we represent the solution in the basis $V_k$ instead of $Q_k$ we obtain Algorithm 3.15 for MinRes. We additionally use the fact that $\beta_j \in \mathbb{R}$, i.e., that $T_k$ is real symmetric, and thus the elements of $L_k$ are real.

**input** : $A$, $x_0$, $b$
**output**: $\underline{T}_k$, $L_k$, $G_j$, $V_k$

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$
**for** $k \in \mathbb{N}$ **do**
  $q \leftarrow Aq_k$
  $\alpha_k \leftarrow q^H q_k$
  $q \leftarrow q - \alpha_k q_k - \beta_{k-1} q_{k-1}$
  $\beta_k \leftarrow \|q\|$
  $\tilde{l}_{kk} \leftarrow -c_k \alpha_k - c_k c_{k-1} \beta_{k-1}$
  $l_{kk} \leftarrow \sqrt{\tilde{l}_{kk}^2 + \beta_k^2}$
  $l_{k,k-1} \leftarrow s_k \alpha_k - c_k c_{k-1} \beta_{k-1}$
  $l_{k,k-2} \leftarrow s_{k-1} \beta_{k-1}$
  $c_{k+1} \leftarrow \tilde{l}_{kk} / l_{kk}$
  $s_{k+1} \leftarrow \beta_k / l_{kk}$
  $v_k \leftarrow (q_k - l_{k,k-2} v_{k-2} - l_{k,k-1} v_{k-1}) / l_{kk}$
  $x_k \leftarrow x_{k-1} + c_{k+1} \eta_k v_k$
  $\eta_{k+1} \leftarrow s_{k+1} \eta_k$
**end for**

Algorithm 3.15: MinRes

Not all basis vectors $q_j$ and $v_j$ have to be stored, it suffices to store the last three vectors. The same holds true for the inverse of $L_k$ and the right-hand side. Thus MinRes and SymmLQ can be implemented using solely short-term recurrences.

### 3.6.7 Biores/Biomin/Biodir

The non-symmetric Lanczos method computes two Hessenberg decompositions in the bi-orthogonal bases $Q_m$ and $\hat{Q}_m$,

$$\begin{aligned} AQ_m &= Q_m T_m, & A^H \hat{Q}_m &= \hat{Q}_m \hat{T}_m \\ AQ_k &= Q_{k+1}\underline{T}_k, & A^H \hat{Q}_k &= \hat{Q}_{k+1}\underline{\hat{T}}_k \end{aligned}$$

where $\hat{T}_m \equiv T_m^H$ and $\hat{Q}_m^H Q_m = I$.

We can apply Orthores, Orthomin and Orthodir to both decompositions. The resulting three new methods are usually termed Biores, Biomin and Biodir. Some authors refer to them as Lanczos/Orthores, Lanczos/Orthomin and Lanczos/Orthodir.

We first consider Biores and Biomin. Biores and Biomin are based on scaled tridiagonals with column sums equal zero. The scaled decompositions are denoted by

$$AR_k = R_{k+1}\underline{T}_k^{(0)} \quad \text{and} \quad A^H\hat{R}_k = \hat{R}_{k+1}\underline{\hat{T}}_k^{(0)}.$$

As before the scaled basis vectors are the residuals, suppose they are given by

$$R_m = Q_m D_\rho \quad \text{and} \quad \hat{R}_m = \hat{Q}_m D_{\hat{\rho}}.$$

We will not dicuss Biores any further. We just mention that Biores breaks down when one matrix $T_k$ is singular. In contrast to the orthogonal approach this does not necessarily imply that zero is in the field of values of $A$.

Orthomin and Orthodir use a set of direction vectors. The undefined scalars are obtained by minimisation along the direction vectors. Local minimisation is still possible, but the recurrence coefficients needed for the global minimisation turn out to be not computable.

The resort is in the LDMT decompositions of $T_m^{(0)}$ and $\hat{T}_m^{(0)}$, given by

$$T_m^{(0)} = L_1 D_m M_m^H \quad \text{and} \quad \hat{T}_m^{(0)} = L_1 \hat{D}_m \hat{M}_m^H.$$

The LDMT decompositions are used to define the direction vectors $p_j$ and $\hat{p}_j$,

$$R_m = P_m M_m^H \quad \text{and} \quad \hat{R}_m = \hat{P}_m \hat{M}_m^H.$$

We observe that the direction vectors $\hat{p}_j, p_j$ are $A$ bi-conjugate, since

$$\begin{aligned} \hat{P}_m^H A P_m = \hat{M}_m^{-1}\hat{R}_m^H A R_m M_m^{-H} &= \hat{M}_m^{-1} D_{\hat{\rho}} D_\rho L_1 D_m \\ &= \hat{D}_m^H L_1^T D_{\hat{\rho}} D_\rho M_m^{-H}. \end{aligned}$$

The bi-orthogonality and bi-conjugacy relations suffice to derive a unique recurrence. This corresponds to the self-adjoint CG case, where we observed that the direction vectors in the Orthomin variant are $A$-conjugate.

The resulting set of equations is given by

$$\begin{aligned} AP_k D_k^{-1} &= R_{k+1}\underline{L}_k, & R_k &= P_k M_k^H, \\ A\hat{P}_k \hat{D}_k^{-1} &= \hat{R}_{k+1}\underline{L}_k, & \hat{R}_k &= \hat{P}_k \hat{M}_k^H. \end{aligned}$$

We have to fix the unknowns in the matrices $D_m$ and $\hat{D}_m$, $M_m$ and $\hat{M}_m$. Let $D_m$ and $\hat{D}_m$ be given by

$$D_m^{-1} = \begin{pmatrix} \alpha_0 & & \\ & \ddots & \\ & & \alpha_{m-1} \end{pmatrix} \quad \text{and} \quad \hat{D}_m^{-1} = \begin{pmatrix} \hat{\alpha}_0 & & \\ & \ddots & \\ & & \hat{\alpha}_{m-1} \end{pmatrix}.$$

This implies that the residual recurrences are given by

$$\begin{aligned} r_{k+1} &= r_k - \alpha_k A p_k, & \hat{r}_{k+1} &= \hat{r}_k - \hat{\alpha}_k A^H \hat{p}_k. \end{aligned}$$

By the bi-orthogonality

$$0 = \langle \hat{r}_k, r_{k+1}\rangle = \langle \hat{r}_k, r_k - \alpha_k A p_k\rangle, \quad \text{i.e.,} \quad \alpha_k = \frac{\langle \hat{r}_k, r_k\rangle}{\langle \hat{r}_k, A p_k\rangle}.$$

Analogously

$$0 = \langle r_k, \hat{r}_{k+1} \rangle = \langle r_k, \hat{r}_k - \hat{\alpha}_k A^H \hat{p}_k \rangle, \quad \text{i.e.,} \quad \hat{\alpha}_k = \frac{\langle r_k, \hat{r}_k \rangle}{\langle r_k, A^H \hat{p}_k \rangle}.$$

We know that $M_m$ and $\hat{M}_m$ are both bidiagonal with unit diagonal. Assume they are given by

$$M_m^H = \begin{pmatrix} 1 & -\beta_0 & & \\ & 1 & \ddots & \\ & & \ddots & -\beta_{m-2} \\ & & & 1 \end{pmatrix}, \quad \hat{M}_m^H = \begin{pmatrix} 1 & -\hat{\beta}_0 & & \\ & 1 & \ddots & \\ & & \ddots & -\hat{\beta}_{m-2} \\ & & & 1 \end{pmatrix}.$$

This implies that the direction vector recurrences are given by

$$p_{k+1} = r_{k+1} + \beta_k p_k, \qquad \hat{p}_{k+1} = \hat{r}_{k+1} + \hat{\beta}_k \hat{p}_k.$$

We use the $A$ bi-conjugacy of the direction vectors to simplify the representation of $\alpha_k$,

$$\alpha_k = \frac{\langle \hat{r}_k, r_k \rangle}{\langle \hat{r}_k, A p_k \rangle} = \frac{\langle \hat{r}_k, r_k \rangle}{\langle \hat{p}_k - \hat{\beta}_{k-1} \hat{p}_{k-1}, A p_k \rangle} = \frac{\langle \hat{r}_k, r_k \rangle}{\langle \hat{p}_k, A p_k \rangle}.$$

The corresponding simplification of $\hat{\alpha}_k$ proves that $\hat{\alpha}_k = \overline{\alpha}_k$.
The bi-conjugacy is used to obtain a representation of $\beta_k$,

$$0 = \langle \hat{p}_k, A p_{k+1} \rangle = \langle \hat{p}_k, A(r_{k+1} + \beta_k p_k) \rangle, \quad \text{i.e.,} \quad \beta_k = -\frac{\langle \hat{p}_k, A r_{k+1} \rangle}{\langle \hat{p}_k, A p_k \rangle}.$$

This representation is changed using the bi-orthogonality, the relation

$$\hat{r}_{k+1} = \hat{r}_k - \overline{\alpha}_k A^H \hat{p}_k \quad \Leftrightarrow \quad A^H \hat{p}_k = -\frac{1}{\overline{\alpha}_k} (\hat{r}_{k+1} - \hat{r}_k)$$

and the expression for $\alpha_k$ into

$$\beta_k = \frac{1}{\alpha_k} \frac{\langle \hat{r}_{k+1} - \hat{r}_k, r_{k+1} \rangle}{\langle \hat{p}_k, A p_k \rangle} = \frac{\langle \hat{r}_{k+1}, r_{k+1} \rangle}{\langle \hat{r}_k, r_k \rangle}.$$

Similarly we obtain that $\hat{\beta}_k = \overline{\beta}_k$.

Biomin is commonly known as bi-conjugate gradients or BiCG, sometimes as BCG. BiCG was already contained in Lanczos first paper. The form used here was derived by Fletcher. An algorithm for BiCG is given in Algorithm 3.16.
Observe that BiCG solves *two* systems of equations, one with $A$ as system matrix, and one with $A^H$ as system matrix. A lucky breakdown of the *left* recurrence is no longer a lucky one, since we are mostly interested in solving the *right* system of equations.

The BiCG recurrences can be expressed in alternate matrix form,

$$\begin{array}{rclcrcl} r_{k+1} & = & r_k - \alpha_k A p_k & \Leftrightarrow & A P_k D_\alpha & = & R_{k+1} \underline{L}_k, \\ \hat{r}_{k+1} & = & \hat{r}_k - \overline{\alpha}_k A^H \hat{p}_k & \Leftrightarrow & A^H \hat{P}_k D_{\overline{\alpha}} & = & \hat{R}_{k+1} \underline{L}_k, \\ p_{k+1} & = & r_{k+1} + \beta_k p_k & \Leftrightarrow & R_{k+1} & = & P_{k+1} L_\beta, \\ \hat{p}_{k+1} & = & \hat{r}_{k+1} + \overline{\beta}_k \hat{p}_k & \Leftrightarrow & \hat{R}_{k+1} & = & \hat{P}_{k+1} L_{\overline{\beta}}. \end{array}$$

This can be transformed to the two three-term recurrence form of the non-symmetric Lanczos method.

We will not consider Biodir. All three methods use short recurrences like the corresponding variants of CG, i.e., like Ores, Omin and Odir.

**input** : $A$, $x_0$, $b$, $\hat{x}_0$, $\hat{b}$

**output**: $X_k$, $\hat{X}_k$, $R_k$, $\hat{R}_k$, $P_k$, $\hat{P}_k$, $D_\alpha$, $L_\beta$

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$
$\hat{r}_0 \leftarrow \hat{b} - A^H\hat{x}_0$, $\hat{p}_0 \leftarrow \hat{r}_0$

**for** $k \in \mathbb{N}_0$ **do**
  Compute $\alpha_k$:
    $\alpha_k \leftarrow \langle \hat{r}_k, r_k\rangle / \langle \hat{p}_k, Ap_k\rangle$
  Update the solution vectors:
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    $\hat{x}_{k+1} \leftarrow \hat{x}_k + \overline{\alpha}_k \hat{p}_k$
  Update the residual vectors:
    $r_{k+1} \leftarrow r_k - \alpha_k Ap_k$
    $\hat{r}_{k+1} \leftarrow \hat{r}_k - \overline{\alpha}_k A^H\hat{p}_k$
  Compute $\beta_k$:
    $\beta_k \leftarrow \langle \hat{r}_{k+1}, r_{k+1}\rangle / \langle \hat{r}_k, r_k\rangle$
  Update the direction vectors:
    $p_{k+1} \leftarrow r_{k+1} + \beta_k p_k$
    $\hat{p}_{k+1} \leftarrow \hat{r}_{k+1} + \overline{\beta}_k \hat{p}_k$
**end for**

Algorithm 3.16: BiCG, Omin variant

We remark that the algorithm for BiCG might also have been obtained by consideration of preconditioned CG for the system

$$\begin{pmatrix} 0 & A \\ A^H & 0 \end{pmatrix} \begin{pmatrix} \hat{x} \\ x \end{pmatrix} = \begin{pmatrix} b \\ \hat{b} \end{pmatrix}, \qquad P = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}.$$

Note that the preconditioner $P$ is an *indefinite preconditioner*.

### 3.6.8 QOR/QMR

We can use the Lanczos decomposition directly to solve the right system of equations. The QOR and QMR approaches are similar to FOM and GMRES, with the exception of the non-orthogonal bases.

The Galerkin approach QOR is similar to the symmetric variant SymmLQ, i.e., we compute the Lanczos decompositions and perform an LQ decomposition. Again there is no need to store all basis vectors and rotations. This approach is mathematical equivalent to the BiCG variants Biores, Biomin and Biodir. Usually Biomin is preferred.

We consider the approach of computing the solution of the overdetermined system

$$\underline{T}_k \underline{z}_k = \|r_0\| e_1 \tag{3.44}$$

with minimal residual. This minimal residual is in general not the true residual, since we use non-orthogonal bases in the non-symmetric Lanczos method. Nevertheless, a residual is minimised. This residual is usually termed *quasi-residual*.

The relation between residual and quasi-residual is given by

$$A\underline{x}_k - r_0 = Q_{k+1}(\underline{T}_k \underline{z}_k - \|r_0\| e_1).$$

The minimal residual solution to the overdetermined system (3.44) is computed by QR decomposition of $\underline{T}_k$. The QR decomposition is computed with Givens rotations. The algorithm is along the lines of general GMRES and symmetric

MinRes. Similar to MinRes, we can throw away old matrix entries, old basis vectors and old Givens rotations.

The method we sketched is termed *quasi-minimal residual* or QMR. QMR was invented by Freund and Nachtigal. We decided to give only an abstract algorithm for QMR, Algorithm 3.17.

**input** : $A$, $x_0$, $b$

**output**: $\underline{T}_k$, $Q_{k+1}$, $\hat{Q}_k$, $U_k$, $R_k$, $\underline{z}_k$, $\underline{x}_k$

$r_0 \leftarrow b - Ax_0$, $q_1 \leftarrow r_0/\|r_0\|$

**for** $k \in \mathbb{N}$ **do**

Update the Lanczos decomposition:

$AQ_k = Q_{k+1}\underline{T}_k$

Update the minimal residual equation:

$\underline{z}_k = \arg\min_{\underline{z}} \|\underline{T}_k\underline{z} - \|r_0\|e_1\|$

Update the $k$th approximation:

$\underline{x}_k = Q_k\underline{z}_k$

**end for**

Algorithm 3.17: QMR

Freund and Nachtigal also implemented a recurrence based on a splitting of the matrix $\underline{T}_k$. This variant uses two coupled short-term recurrences.

### 3.6.9 Look-Ahead

One main problem with nonsymmetric short-term recurrences is the possibility of serious breakdowns of the underlying recurrence. We note that even though the next coefficients are not defined, the space is still expanded.

We may think of a method that expands the Krylov space for a while, and then uses pivoting in the underlying triangular decomposition of the Hankel matrix. The class of methods we just defined are known as *look-ahead* methods.

It may happen that a pivoting cures the breakdown only when reaching the dimension of $A$. This case will be termed *incurable breakdown*. All other cases are termed *curable breakdown*.

Numerical experiments show that most serious breakdowns are curable. The number of look-ahead steps necessary is in the average at most four.

Different look-ahead techniques have been obtained by the connection of the underlying unsymmetric Lanczos method to Padé tables, rational approximation, block-polynomials and FOP (formal orthogonal polynomials). Despite all effort no ultimate form of look-ahead has developed yet.

We only remark that most look-ahead methods do not destroy the Hessenberg structure of the Krylov decomposition, and thus fit into our framework. The formerly tridiagonal matrices fill up, stairs appear on the upper part.

### 3.6.10 Lanczos-Type Product Methods

BiCG has very irregular convergence properties. The sizes of residuals and iterates vary greatly in magnitude. The need for the Hermitian of $A$ turns out to be a severe restriction since in many cases only matrix-vector products and not the entries of $A$ are accessible.

The basis vectors can be expressed as polynomials in $A$ times the starting vector. These polynomials are the characteristic polynomials of the Hessenberg matrices involved, and thus the polynomials are almost the same for both the left and right recurrence.

Suppose the residual vectors $r_k$, $\hat{r}_k$ and the direction vectors $p_k, \hat{p}_k$ generated by BiCG are given by the polynomial representations

$$\begin{array}{rclcrcl} r_k &=& \rho_k(A)r_0, & & \hat{r}_k &=& \overline{\rho_k}(A^H)\hat{r}_0, \\ p_k &=& \pi_k(A)p_0, & & \hat{p}_k &=& \overline{\pi_k}(A^H)\hat{p}_0. \end{array}$$

We only need the inner products of the residual vectors and the direction vectors. Sonneveld observed that the inner products can be transformed to a form without the need for $A^H$ by squaring the polynomials,

$$\begin{array}{rclcl} \langle \hat{r}_k, r_k \rangle &=& \langle \rho_k(A^H)\hat{r}_0, \rho_k(A)r_0 \rangle &=& \langle \hat{r}_0, \rho_k^2(A)r_0 \rangle, \\ \langle \hat{p}_k, p_k \rangle &=& \langle \pi_k(A^H)\hat{p}_0, \pi_k(A)p_0 \rangle &=& \langle \hat{p}_0, \pi_k^2(A)p_0 \rangle. \end{array}$$

This allows us to express the coefficients of the recurrence with the aid of the left residuals and direction vectors and *one* additional right vector, since $\hat{p}_0 = \hat{r}_0$. We obtain a new method when we can find a recurrence of the *squared* polynomials.

The BiCG polynomials fulfil the recurrences

$$\rho_{k+1} = \rho_k - \alpha_k \lambda \pi_k, \quad \pi_{k+1} = \rho_{k+1} + \beta_k \pi_k.$$

The squared polynomials fulfil the recurrences

$$\begin{array}{rclcl} \rho_{k+1}^2 &=& (\rho_k - \alpha_k \lambda \pi_k)^2 &=& \rho_k^2 - 2\alpha_k \lambda \rho_k \pi_k + \alpha_k^2 \lambda^2 \pi_k^2, \\ \pi_{k+1}^2 &=& (\rho_{k+1} + \beta_k \pi_k)^2 &=& \rho_{k+1}^2 + 2\beta_k \rho_{k+1} \pi_k + \beta_k^2 \pi_k^2. \end{array}$$

We need recurrences for the mixed terms. It is sufficient to use a third recurrence, since

$$\rho_k \pi_k = \rho_k(\rho_k + \beta_{k-1}\pi_{k-1}) = \rho_k^2 + \beta_{k-1}\rho_k \pi_{k-1},$$

He goes on to define the recurrence vectors

$$r_k = \rho_k^2(A)r_0, \quad p_k = \pi_k^2(A)p_0, \quad q_k = \rho_{k+1}(A)\pi_k(A)r_0.$$

The algorithm sketched is named *Conjugate Gradient Squared*, short CGS, because of the squaring of the (Bi)CG polynomials. Using solely three recurrence vectors the algorithm looks as follows,

$$\begin{array}{rcl} r_{k+1} &=& r_k - \alpha_k A(2r_k + 2\beta_{k-1}q_{k-1} - \alpha_k A p_k), \\ q_k &=& r_k + \beta_{k-1}q_{k-1} - \alpha_k A p_k, \\ p_{k+1} &=& r_{k+1} + 2\beta_k q_k + \beta_k^2 p_k. \end{array} \tag{3.45}$$

For reasons of efficiency he sets $u_k = r_k + \beta_{k-1}q_{k-1}$. We state the resulting CGS recurrences together with the alternate matrix form,

$$\begin{array}{rcl} q_k = u_k - \alpha_k A p_k &\Leftrightarrow& Q_k = U_k - AP_k D_\alpha \\ x_{k+1} = x_k + \alpha_k(u_k + q_k) &\Leftrightarrow& X_{k+1}\underline{L}_k = -(U_k + Q_k)D_\alpha \\ r_{k+1} = r_k - \alpha_k A(u_k + q_k) &\Leftrightarrow& R_{k+1}\underline{L}_k = A(U_k + Q_k)D_\alpha \\ u_{k+1} = r_{k+1} + \beta_k q_k &\Leftrightarrow& U_{k+1} = R_{k+1} + Q_{k+1}N_\beta \\ p_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k p_k) &\Leftrightarrow& P_{k+1} = U_{k+1} + Q_{k+1}N_\beta + P_{k+1}N_{\beta^2}. \end{array}$$

The final version of the algorithm for CGS is given by Algorithm 3.18.
These recurrences can not be transformed to a Hessenberg decomposition solely in $R_k$. This becomes obvious when looking at the *expansion tree*

$$\begin{array}{rclcrcl} r_0, u_0, p_0 &\in& \mathcal{K}_1, & & q_0 &\in& \mathcal{K}_2 \\ r_1, u_1, p_1 &\in& \mathcal{K}_3, & & q_1 &\in& \mathcal{K}_4 \\ r_2, u_2, p_2 &\in& \mathcal{K}_5, & & q_2 &\in& \mathcal{K}_6. \end{array}$$

**input** : $A$, $x_0$, $b$, $\hat{r}_0$
**output**: $X_k, R_k, Q_k, P_k, U_k, \underline{L}_k, D_\alpha, N_\beta, N_{\beta^2}$

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$, $u_0 \leftarrow r_0$
**for** $k \in \mathbb{N}_0$ **do**
  $\alpha_k \leftarrow \langle \hat{r}_0, r_k \rangle / \langle \hat{r}_0, Ap_k \rangle$
  $q_k \leftarrow u_k - \alpha_k A p_k$
  $x_{k+1} \leftarrow x_k + \alpha_k (u_k + q_k)$
  $r_{k+1} \leftarrow r_k - \alpha_k A(u_k + q_k)$
  $\beta_k \leftarrow \langle \hat{r}_0, r_{k+1} \rangle / \langle \hat{r}_0, r_k \rangle$
  $u_{k+1} \leftarrow r_{k+1} + \beta_k q_k$
  $p_{k+1} \leftarrow u_{k+1} + \beta_k (q_k + \beta_k p_k)$
**end for**

Algorithm 3.18: CGS

The residuals $r_j$ are only defined in Krylov subspaces with odd dimensions. The recurrences can be expressed as a Hessenberg decomposition in the mixed basis $R_k, Q_k$. We denote the basis by $W_{2k}$, i.e we define the matrix

$$W_{2m} = \Big[ r_0, q_0, \ldots, r_m, q_m \Big].$$

This defines the Hessenberg decomposition

$$\begin{aligned} AW_{2m} &= W_{2m} C_{2m} \\ AW_k &= W_{k+1} \underline{C}_k \quad \forall\ k < 2m \end{aligned}$$

in *implicit form*. The coefficients of the Hessenberg matrix are never *explicitly* given. They may be constructed by the recurrences. It is obvious that they are given by algebraic expressions in the CGS coefficients $\alpha_j, \beta_j$, $j \in \underline{m}$.

The coefficients may be computed by removing $p_k$ from the set of recurrences (3.45). We use induction to obtain

$$\begin{aligned} p_k &= \beta_{k-1}^2 p_{k-1} + r_k + 2\beta_{k-1} q_{k-1} \\ &= \sum_{i=0}^{k} \prod_{j=1}^{i} \beta_{k-j}^2 (r_{k-i} + 2\beta_{k-1-i} q_{k-1-i}). \end{aligned}$$

This representation of $p_k$ is inserted into the remaining two recurrences.

This is an interesting case of a Hessenberg decomposition where the Hessenberg matrix is *structured and not sparse*, i.e., not tridiagonal, banded or blocked.

CGS is a *Lanczos-type product method*, short LTPM. Lanczos-type product methods are based on the non-symmetric Lanczos algorithm and use products of polynomials for the recurrence of the residuals. Another example of a Lanczos-type product method is BiCGSTAB (BiCG Stabilised) by van der Vorst. BiCGSTAB is based on the recurrence

$$r_k = \rho_k(A)\psi_k(A), \qquad \psi_{k+1}(\lambda) = (1 - \omega_k \lambda)\psi_k(\lambda),$$

where $\rho_k$ is the usual polynomial for the residual recurrence of BiCG. The first polynomial $\psi_0$ is assumed constant.

The value of $w_k$ is chosen to minimise the residual

$$\omega_k = \arg\min_{\omega} \|(I - \omega A)(r_k - \alpha_k A p_k)\|.$$

**input** : $A$, $x_0$, $b$, $\hat{r}_0$
**output**: $X_k, R_k, P_k, Q_k, \ldots$

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow r_0$
**for** $k \in \mathbb{N}_0$ **do**
  $\alpha_k \leftarrow \langle \hat{r}_0, r_k \rangle / \langle \hat{r}_0, Ap_k \rangle$
  $q_k \leftarrow r_k - \alpha_k A p_k$
  $\omega_k \leftarrow \langle q_k, Aq_k \rangle / \langle Aq_k, Aq_k \rangle$
  $x_{k+1} \leftarrow x_k + \alpha_k p_k + \omega_k q_k$
  $r_{k+1} \leftarrow q_k - \omega_k A q_k$
  $\beta_k \leftarrow (\alpha_k \langle \hat{r}_0, r_{k+1} \rangle) / (\omega_k \langle \hat{r}_0, r_k \rangle)$
  $p_{k+1} \leftarrow r_{k+1} - \beta_k (p_k - \omega_k A p_k)$
**end for**

Algorithm 3.19: BiCGSTAB

We will not go further into the details. The final algorithm for BiCGSTAB is given by Algorithm 3.19.
Again we consider the expansion tree for the BiCGSTAB recurrences,

$$\begin{array}{llllll} r_0, p_0 & \in & \mathcal{K}_1, & q_0 & \in & \mathcal{K}_2, \\ r_1, p_1 & \in & \mathcal{K}_3, & q_1 & \in & \mathcal{K}_4, \\ r_2, p_2 & \in & \mathcal{K}_5, & q_2 & \in & \mathcal{K}_6. \end{array}$$

We conclude that again we have implicitly defined a Hessenberg decomposition in the basis $W_{2k}$,

$$W_{2m} = \Big[ r_0, q_0, \ldots, r_m, q_m \Big].$$

The coefficients of the Hessenberg matrix may be derived by removing the dependency on $p_k$. This can be done by induction on the equation

$$p_k = r_k + \beta_{k-1}(p_{k-1} - \omega_{k-1} A p_{k-1}).$$

The resulting matrix depends algebraically on the recurrence coefficients.

Other Lanczos-type product methods include BiCGstab($l$), which is based on higher degree polynomials, CGS2, shifted CGS and Transpose-Free QMR, short TFQMR. We remark that TFQMR is not connected to QMR. TFQMR is obtained by consideration of the implicit Hessenberg decomposition we already noted. The idea behind TFQMR is to transform every second vector, which is not a residual for the original system, to a residual in the original system. The coefficients of the recurrence are chosen such that they are minimisers for the quasi-residuals.

LTPMs are distinguished by the second polynomial used in the recurrence of the residual vectors. They are also referred to as smoothing techniques, since they smooth the irregular behaviour of BiCG.

All these methods can be transformed to a Hessenberg decomposition by introducing larger basis matrices with auxiliary vectors. The coefficients of the Hessenberg matrices are defined *implicitly* by the recurrence coefficients.

### 3.6.11 CGNR/CGNE

The methods based on the Arnoldi and non-symmetric Lanczos decompositions apply to general matrices. But these methods have limitations. The long-term recurrences suffer from severe time and memory limitations. The short-term recurrences based on the non-symmetric Lanczos decomposition suffer from breakdown and irregular convergence.

None of these problems occur in case $A$ is Hermitian. The short-term recurrences based on the *symmetric Lanczos decomposition* impose no restrictions on storage and time *and* are immune to breakdown.

We want to obtain *short-term* recurrences *without breakdown* for *general matrices*. We consider the normal equations of type 1 and 2,

$$A^H Ax = A^H r_0 \qquad \text{and} \qquad AA^H y = r_0, \quad \text{where} \quad A^H y = x.$$

In both cases the resulting system has a system matrix that is HPD. The solutions of both systems are the solution of the original system.

We apply CG with small modifications to solve these systems. The resulting two methods are termed CGN methods. CGN stands for CG applied to the normal equations.

The idea of using CG on the normal equations of type 1 was already discussed by Hestenes and Stiefel. The resulting method is termed CGNR. CGNR minimises the residual, since CG minimises the error in the norm

$$\|x_k - A^{-1}r_0\|_{A^H A} = \|Ax_k - r_0\| = \|r_k\|.$$

We consider the Omin variant of CG. CGNR can be simplified. We observe that the residual of the normal equation is given by

$$w_k \equiv A^H b - A^h Ax_k = A^H(b - Ax_k) = A^H r_k.$$

The notation $w_k$ should remind of *wrong* residual.

We define $s_k \equiv Ap_k$. The computation of $\alpha_k$ in the CG method turns into

$$\alpha_k = \frac{\langle w_k, w_k\rangle}{\langle A^H Ap_k, p_k\rangle} = \frac{\langle w_k, w_k\rangle}{\langle Ap_k, Ap_k\rangle} \equiv \frac{\langle w_k, w_k\rangle}{\langle s_k, s_k\rangle}$$

This enables us to express the recurrences in the residuals $r_k$. The resulting algorithm for CGNR is given by Algorithm 3.20.

**input** : $A$, $x_0$, $b$
**output**: $R_k$, $W_k$, $S_k$, ...

$r_0 \leftarrow b - Ax_0$, $w_0 \leftarrow A^H r_0$, $p_0 \leftarrow w_0$

**for** $k \in \mathbb{N}_0$ **do**
  $s_k \leftarrow Ap_k$
  $\alpha_k \leftarrow \langle w_k, w_k\rangle / \langle s_k, s_k\rangle$
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$
  $r_{k+1} \leftarrow r_k - \alpha_k s_k$
  $w_{k+1} \leftarrow A^H r_{k+1}$
  $\beta_k \leftarrow \langle w_{k+1}, w_{k+1}\rangle / \langle w_k, w_k\rangle$
  $p_{k+1} \leftarrow w_{k+1} + \beta_k p_k$
**end for**

Algorithm 3.20: CGNR

When we use the normal equations of type 2, we obtain the method known as CGNE. CGNE was derived by Craig and is also known as Craig's method. CGNE minimises the error, since CG minimises the error of the normal equation approximation in the norm

$$\|y_k - y\|_{AA^H} = \|x_k - x\|.$$

Here we denoted the $k$th approximation to the solution of the normal equation by $y_k$ and defined $x_k \equiv A^H y_k$. Also CGNE can be simplified. The residual of the normal equation is the original residual, since

$$r_k = b - Ax_k = b - AA^H y_k.$$

Denote the direction vectors by $s_k$. Define $p_k \equiv A^H s_k$. Then the computation of $\alpha_k$ is given by

$$\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle AA^H s_k, s_k \rangle} = \frac{\langle r_k, r_k \rangle}{\langle A^H s_k, A^H s_k \rangle} \equiv \frac{\langle r_k, r_k \rangle}{\langle p_k, p_k \rangle}.$$

We drop the update of $y_k$ and use solely the update of the desired solution $x_k$,

$$y_{k+1} = y_k + \alpha_k s_k \quad \Leftrightarrow \quad x_{k+1} = x_k + \alpha_k A^H s_k = x_k + \alpha_k p_k$$

We can get rid of the direction vectors $s_k$, since the update of the direction vectors is given by

$$s_{k+1} = r_{k+1} + \beta_k s_k \quad \Leftrightarrow \quad p_{k+1} = A^H r_{k+1} + \beta_k p_k.$$

Algorithm 3.21 is the resulting implementation of CGNE.

**input** : $A$, $x_0$, $b$
**output**: $X_k$, $R_k$, $P_k$, ...

$r_0 \leftarrow b - Ax_0$, $p_0 \leftarrow A^H r_0$
**for** $k \in \mathbb{N}_0$ **do**
  $\alpha_k \leftarrow \langle r_k, r_k \rangle / \langle p_k, p_k \rangle$
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$
  $r_{k+1} \leftarrow r_k - \alpha_k A p_k$
  $\beta_k \leftarrow \langle r_{k+1}, r_{k+1} \rangle / \langle r_k, r_k \rangle$
  $p_{k+1} \leftarrow A^H r_{k+1} + \beta_k p_k$
**end for**

Algorithm 3.21: CGNE

We remark that despite the fact that both algorithms never explicitly form the normal equations, an implicit squaring of the condition number can be observed in praxis.

## 3.7 Krylov Methods and Preconditioning

Krylov methods are based on a starting vector and a matrix to span the Krylov space. The matrix used to span the Krylov space in most cases is the matrix of some inverted, shifted, or more generally *preconditioned* problem.

Preconditioning is mainly used in context of linear systems of equations. We distinguish between left, right and two-sided preconditioning.

*Left preconditioning* is the multiplication of the system of equations with a matrix $P_l$,

$$P_l A x = P_l b.$$

Left preconditioning tries to achieve $P_l A \approx I$.

*Right preconditioning* substitutes the solution vectors with pre-multiplied vectors $y$,

$$AP_r y = b, \quad P_r y = x.$$

Right preconditioning tries to achieve $AP_r \approx I$.

*Two-sided preconditioning* uses both ideas of preconditioning,

$$P_l A P_r u = b, \quad P_r u = x.$$

Two-sided preconditioning tries to achieve $P_l A P_r \approx I$. This form of preconditioning comes mainly in symmetric fashion, for instance when the preconditioning matrices are the factors of an incomplete Cholesky factorisation.

Preconditioning can be applied to all methods. It is never necessary to form the preconditioned matrix explicitly, mostly we use an approximation to $A$ and have to solve a linear system with this approximation in every step.

Often used preconditioners include incomplete factorisations, like the ILU, ICC and ILQ preconditioners. Another common choice is one or more steps of a basic iterative method, like Jacobi, SOR or SSOR. Polynomials in $A$ are used as preconditioners. These choices usually have a small defect in the solution of the linear system with the preconditioning matrix.

Another Krylov method may be used as preconditioner. This choice leads to methods known as inner-outer iterations. Here the size of the defect is varying from step to step. It is better for the analysis, to express the complete process as *one new single* Krylov method.

Very simple preconditioners in the solution of the eigenproblem are shifts. Together with inversion we obtain the shift and invert approach often used in Krylov methods for the eigenproblem. This may be generalised by applying non-constant shifts. The methods obtained are referred to as rational Krylov methods.

Krylov methods are always used with preconditioning. We will not consider preconditioning in our approach, since this would make the error analysis almost intractable.

# Chapter 4

# A Unified Approach

At first glance the behaviour of Krylov methods in *finite* precision is entirely different from the behaviour in *infinite* precision. This holds true *especially* when we consider short-term recurrences. Surprisingly we still obtain good approximations.

Common to all Krylov methods is that the underlying Krylov decomposition is fulfilled approximately. The other properties, like orthogonality, duality and $A$-conjugacy of vector recurrences, are more or less no longer present. This has severe impacts. Finite precision Krylov eigensolvers often return *multiple instances* of *simple* eigenvalues. The *bounds* indicating that a Ritz value has converged to a certain accuracy *may grow*, as if the Ritz value starts to diverge. Finite precision Krylov linear system solvers still compute approximations with a *small residual*. In many cases the *number of steps* necessary is a moderate *multiple of the dimension* of the matrix.

In infinite precision the derivation of the methods and the convergence analysis was based on (orthogonal) polynomials and the connection to optimisation. In finite precision we decided to focus on a matrix analysis approach based on Krylov decompositions. First we classify the Krylov methods. Then we show how our Krylov decomposition approach can be used for describing the behavior of finite precision Krylov methods. We show why and how orthogonality is lost, and develop recurrences and backward formula describing the propagation of the error terms.

## 4.1 Classification

All Krylov methods introduced in Chapter 3 are explicitly, or at least implicitly, based on Hessenberg decompositions,

$$\begin{array}{lcll} AQ_m & = & Q_m C_m & \\ AQ_k & = & Q_{k+1}\underline{C}_k & = Q_k C_k + q_{k+1} c_{k+1,k} e_k^T \\ & & & \equiv Q_k C_k + M_k \qquad \forall\ k < m. \end{array}$$

The matrix $M_k$ collecting the newest quantities is introduced for reasons of brevity. The matrix $C_m$ is given by the projection $\hat{Q}_m^H A Q_m$, with a second basis $\hat{Q}_m$. The methods can be classified according to several respects.

Depending on the *projection*, i.e. on the bases used, we distinguish between *singular, orthogonal* and *dual* methods. Depending on the *field of application* we distinguish between *eigenproblem solvers* and *linear system solvers*. The linear system solvers are divided further into *QOR and QMR* methods. Depending on the *length of the vector recurrence* used, we distinguish between *short-term recurrences* and *long-term recurrences*.

These distinctions suffice to classify the basic Krylov methods. The resulting classification scheme is given as Table 4.1. The singular projections are left out.

| **eigenproblem solvers** | | |
|---|---|---|
| | orthogonal | dual |
| short recurrence | symmetric Lanczos | dual Lanczos |
| long recurrence | Arnoldi | — |

| **linear system solvers (QOR)** | | |
|---|---|---|
| | orthogonal | dual |
| short recurrence | SymmLQ/CG | BiCG |
| long recurrence | FOM/Orthores | — |

| **linear system solvers (QMR)** | | |
|---|---|---|
| | orthogonal | dual |
| short recurrence | MinRes/CR | QMR |
| long recurrence | GMRES/Orthodir/Orthomin | — |

Table 4.1: Classification of the basic Krylov methods

There are recurrences that use *a single Hessenberg decomposition* for constructing a basis, and methods that use *two Hessenberg decompositions* to construct two bases.

Using block matrices all basic methods can be transformed into one simple matrix recurrence. We assume the step $k$ fixed and leave the subscripts out. After we have transformed the linear system solvers to their eigenproblem counterpart, all methods fulfil

$$\begin{pmatrix} A & 0 \\ 0 & C^H \end{pmatrix} \begin{pmatrix} 0 & \hat{Q} \\ Q^H & 0 \end{pmatrix} - \begin{pmatrix} 0 & \hat{Q} \\ Q^H & 0 \end{pmatrix} \begin{pmatrix} A^H & 0 \\ 0 & C \end{pmatrix} = \begin{pmatrix} 0 & M \\ -\hat{M}^H & 0 \end{pmatrix}.$$

In case of an orthogonal method $\hat{Q} = Q$ and $Q^H Q = I$, in case of a bi-orthogonal method $\hat{Q}^H Q = I$ with a second matrix $\hat{Q}$. In case of a long-term recurrence the matrix $C$ will be Hessenberg, in case of a short-term recurrence $C = T$ will be tridiagonal.

Not contained in this classification scheme are generalisations, like the modified CG methods based on the normal equations, CGNE and CGNR. Also not contained is the important class of Lanczos type product methods (LTPMs).

The Lanczos type product methods are based on the dual Lanczos method. They have in common that they do not use the transpose (Hermitian) of $A$, and use short recurrences. LTPMs are classified according to the polynomial they use to construct the sequence of residual vectors.
The residual polynomial of an LTPM is composed as a product of the ordinary residual polynomial $\rho_k$ and a second polynomial $\phi_k$. A subclass, the quasi-minimal residual Lanczos-type product methods, or QMRLTPMs, minimise and iterate in terms of the *quasi-residual* instead of terms of the *true residual*.

The orthogonal short-term recurrences are the most desirable ones. The methods we stated only apply to *Hermitian* $A$. For many *normal* matrices short-term recurrences are known, for instance for symplectic or skew-symmetric matrices.

Faber and Manteuffel have analysed the conditions for the existence of an orthogonal short-term recurrence in detail. The results may be found in the Faber

| **squared variants** | |
|---|---|
| LTPM | relation of second polynomial to original polynomial |
| CGS | $\phi_k = \rho_k$ |
| CGS2 | $\phi_k = \tilde{\rho}_k$ |
| shifted CGS | $\phi_k = (1-\alpha_k t)\rho_{k-1}$ |

| **stabilised variants** | |
|---|---|
| LTPM | recursion formula for second minimising polynomial |
| BICGSTAB | $\phi_k = (1-\alpha_k t)\phi_{k-1}$ |
| BICG×MR2 | $\phi_k = (\beta_k+\gamma_k t)\phi_{k-1} + (1-\beta_k)\phi_{k-2}$ |
| BICGSTAB2 | $\phi_k = \begin{cases} (1-\alpha_k t)\phi_{k-1}, & k \text{ even} \\ (\beta_k+\gamma_k t)\phi_{k-1} + (1-\beta_k)\phi_{k-2}, & k \text{ odd} \end{cases}$ |
| BiCGstab($\ell$) | $\phi_k = \begin{cases} (1-\alpha_k t)\phi_{k-1}, & k \bmod \ell = 1 \\ (\beta_k+\gamma_k t)\phi_{k-1} + (1-\beta_k)\phi_{k-2}, & k \bmod \ell = 2 \\ \vdots & \vdots \\ \ell\text{–dimensional minimisation}, & k \bmod \ell = \ell \end{cases}$ |

| **QMR variants** | |
|---|---|
| LTPM | minimises the quasi-residual polynomial of |
| TFQMR | CGS |
| QMRCGSTAB | BICGSTAB |

Table 4.2: Classification of the Lanczos-type product methods

and Manteuffel paper (cf. [FM84]) and in the books of Greenbaum and Saad (cf. [Gre97, Saa96]). The results are disillusioning, since only for *very special classes* of matrices such recurrences can exist. Hence, only orthogonal *or* short-term recurrences can be used on general $A$.

It can be shown that *in principle* for every matrix $A$ there exist two starting vectors such that two *dual* short-term recurrences work without any breakdown, but there seems to be no easy way to construct these vectors without knowing the solution one wishes to compute. Nevertheless, some look-ahead technique would have to be used in any case, to ensure the *numerical stability*.

In the analysis of finite precision Krylov methods another classification scheme is valuable. In this case we will distinguish between methods *directly based on Hessenberg decompositions*, methods *based on two recurrences* similar to Orthomin, and *LTPMs*. The errors in this three classes behave similar and can, to some extent, be analysed simultaneously. This will be done in part in the next section.

## 4.2 Finite Precision Krylov Methods

We already know that *infinite precision* Krylov methods are based on Hessenberg decompositions,

$$\begin{aligned} AQ_m &= Q_m C_m \\ AQ_k &= Q_{k+1}\underline{C}_k = Q_k C_k + q_{k+1} c_{k+1,k} e_k^T \quad \forall\, k < m. \end{aligned} \tag{4.1}$$

What about the *finite precision* Krylov methods?

In this section we analyse how the Hessenberg decomposition framework changes due to execution in *finite precision*. In finite precision the order of execution is crucial. To emphasise this, we re-order the Hessenberg decomposition,

$$M_k \equiv q_{k+1} c_{k+1,k} e_k^T = AQ_k - Q_k C_k. \tag{4.2}$$

We refer to equation (4.2) as the *governing equation* of the Krylov method.

**input** : $A$, $r_0$
**output**: $C_k$, $Q_k$ for all $k \in \mathbb{N}$

**for** $k \in \mathbb{N}$ **do**
  Compute the next iterate: $r_k \leftarrow Aq_k$
  Compute the next column of moments:
  **for** $j \in J(k)$ **do**
    $c_{jk} \leftarrow \langle \hat{q}_j, r_k \rangle$
  **end for**
  Compute the purged residual:
  $r_k \leftarrow r_k - \sum_{j \in J(k)} q_j c_{jk}$
  Compute the normalisation constant
  and the new basis vector:
  $q_{k+1} c_{k+1,k} \leftarrow r_k$
**end for**

Algorithm 4.1: Direct Hessenberg decomposition

A broad class of methods uses the recurrence (4.1) directly to compute the basis vectors $Q_k$ and the projected matrix $C_k$. This class compromises of the methods for the eigenproblem and many methods for the solution of linear systems. The Hessenberg decomposition together with a bi-orthonormal basis $\hat{Q}_m$ is the means of computation. We can represent all methods in this class in the abstract framework given by Algorithm 4.1.

The notion $J(k)$ is used to denote a set of indices. In case of a long-term recurrence like the Arnoldi algorithm, $J(k) = \underline{k}$. In case of a short-term recurrence like the Lanczos algorithm, $J(k) = \{k-1, k\}$. In case of restarts and truncation the sets $J(k)$ vary.

Methods implemented in this manner have *two essential* features. First, the *non-zero entries* $(j,k)$ $(j \in J(k))$ of the last computed column of $C_k$ will be *approximately equal* to the corresponding entries in the *exact projection*, i.e.

$$\langle \hat{q}_j, Aq_k \rangle \approx c_{jk} \quad \text{or} \quad e_j^T \left( \hat{Q}_k^H A Q_k - C_k \right) e_k \approx 0.$$

In long-term recurrences the non-zero entries are just the upper triangular part. In short-term recurrences the non-zero entries are given by the main and upper diagonal. Using error analysis, we can prove the forward error bound

$$|c_{jk} - \langle \hat{q}_j, Aq_k \rangle| \leq \gamma_{2n} |\hat{q}_j^H| |A| |q_k| \quad \forall\, j \in J(k).$$

Second, the *vector recurrence*

$$q_{k+1} c_{k+1,k} = Aq_k - \sum_{j \in J(k)} q_j c_{jk} + f_k$$

will be fulfilled approximately. This means that the vector $f_k$, introduced for balancing the local errors, is *small*. The actual size depends on the method. By

application of Lemma 1.7, that is, Lemma 8.4 in Higham ([Hig96], p. 154) we obtain the bound

$$\begin{aligned} |f_k| &\leq \gamma_n |A||q_k| + \gamma_{\mathrm{nnz}(c_k)} |Q_{k+1}||c_k| \\ &\leq \gamma_n |A||q_k| + \gamma_{k+1} |Q_{k+1}||c_k|. \end{aligned}$$

This bound holds regardless of order of evaluation.

These two features imply two other. The recurrence can be interpreted as finite precision two-sided Gram-Schmidt. This implies that the computed vectors $q_{k+1}$ are approximately bi-orthogonal to the set $\{\hat{q}_j\}, j \in J(k)$,

$$\hat{q}_j^H q_{k+1} \approx 0.$$

This will be termed local orthogonality (local duality). This local orthogonality ensures that the *subdiagonal* elements of the Hessenberg matrices are to some extent close to the moments. We will not quantify this closeness because of its problem dependence.

From now on we focus on our matrix approach. We introduce the error matrix

$$F_k = \begin{bmatrix} f_1, \ldots, f_k \end{bmatrix}$$

collecting the local errors and obtain the finite precision variant of the governing equation,

$$M_k = AQ_k - Q_k C_k + F_k \qquad \Leftrightarrow \qquad AQ_k - Q_k C_k = M_k - F_k. \tag{4.3}$$

This proves that the quantities computed in a finite precision Krylov method fulfil a *finite precision analogue* of a Hessenberg decomposition.

The first form of (4.3) shows how $A$, $Q_k$ and $C_k$ and the unknown error term $F_k$ form the *rank-one* matrix $M_k$. The second form shows that when regarded as subspace equation in $Q_k$, the residual is composed of $M_k$ (rank-one, changing) and $F_k$ (small, trailing columns unaltered). This equation is pictorially given by figure 4.1.

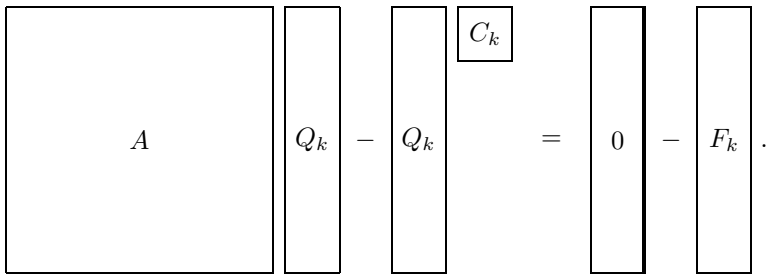


Figure 4.1: The governing equation in finite precision

The error term $F_k$ can be bounded as follows:

**Theorem 4.1** *Let $A \in \mathbb{K}^{n \times n}$. Suppose that a Krylov algorithm based on the direct approach of Algorithm 4.1 has been used and results in the perturbed Hessenberg decomposition*

$$AQ_k - Q_{k+1} \underline{C}_k = -F_k.$$

*Then the error matrix $F_k$ can be bounded by*

$$\begin{aligned} |F_k| &\leq \gamma_n |A||Q_k| + |Q_{k+1}||\underline{C}_k| D_\gamma \\ &\leq \gamma_n |A||Q_k| + \gamma_{k+1} |Q_{k+1}||\underline{C}_k| \\ &\leq \gamma_{\max(n,k+1)} \left( |A||Q_k| + |Q_{k+1}||\underline{C}_k| \right), \end{aligned}$$

*regardless of order of evaluation. Here the diagonal matrix $D_\gamma$ is defined as*

$$D_\gamma = \begin{pmatrix} \gamma_2 & & \\ & \ddots & \\ & & \gamma_{k+1} \end{pmatrix}.$$

*Furthermore, this error analysis is* independent *of a* right *diagonal scaling. This feature can be used to scale the basis vectors to unit length.*

**Proof.** The proof follows by iterated application of Lemma 1.7, i.e. by Lemma 8.4 in Higham. Note that the bounds can be sharpened in case of sparse $A$, since we did not exploit the sparsity of $A$. □

This approach is directly comparable to the error analysis approach when considering GE or other triangular decompositions (cf. [Hig96]). The main difference between the Krylov subspace method error analysis and the GE error analysis is the lack of such notion as a *growth factor* in Krylov subspace methods.

**Remark 4.2** The CGS and MGS variants of Arnoldi, most Lanczos variants and methods based on those are based on Algorithm 4.1, e.g. FOM, GMRES, MinRes, SymmLQ.

Furthermore, Orthores and Orthodir are based on Algorithm 4.1, e.g. Orthores, Orthodir, CG-Ores, CG-Odir, CR-Ores, CR-Odir, Biores, Biodir.

The matrix $F_k$ need not to be small in an absolute sense. This becomes obvious when considering the methods based on Orthores. The *implicit* scaling depends on the *computed* matrices $C_k$. This scaling may magnify the errors.

Methods that do not use an explicit Hessenberg decomposition consist of two or more coupled recurrences for spanning the basis. There are *two types* of methods that do not use the Hessenberg decomposition directly. On *one hand*, we have the methods based on Orthomin, i.e. Orthomin, CG-Omin, CR-Omin and Biomin, the Orthomin variant of BiCG. On the *other hand*, we have the LTPMs, the Lanczos-type product methods. It is easy to show that we can transform coupled finite precision recurrences to exhibit the form (4.3).

We will show that the methods based on Orthomin have an error term that is composed of two terms, namely the errors of the of the residual recurrence times $A$ and the errors of the direction recurrence times a lower triangular matrix times the computed $C_k$.

This follows by consideration of the finite precision recurrences

$$AP_k D_\alpha = R_{k+1}\underline{L}_k + F_k^{(R)}, \qquad R_k = P_k M_k^H + F_k^{(P)}.$$

We insert the second equation into the first one and obtain the perturbed Hessenberg decomposition

$$\begin{aligned} A(R_k - F_k^{(P)}) M_k^{-H} D_\alpha &= R_{k+1}\underline{L}_k + F_k^{(R)} \\ \Rightarrow \quad A(R_k - F_k^{(P)}) &= R_{k+1}\underline{C}_k^{(o)} + F_k^{(R)} L_k^{-1} C_k^{(o)} \\ \Rightarrow \quad AR_k &= R_{k+1}\underline{C}_k^{(o)} - F_k, \end{aligned}$$

where we have defined $F_k$ by

$$-F_k \equiv AF_k^{(P)} + F_k^{(R)} L_k^{-1} C_k^{(o)}.$$

We obtain an equation that has formally the same structure than in case of a direct Hessenberg decomposition. The error matrix is the sum of two *relative* error matrices.

A nice result can be obtained when we scale the residuals, i.e. the basis vectors to have length one. Let the diagonal scaling matrix be given as

$$D_\rho = \mathrm{diag}(\|r_i\|_2).$$

Then the scaled decomposition looks like

$$AQ_k - Q_{k+1}\underline{C}_k = -AF_k^{(P)} D_\rho^{-1} - F_k^{(R)} D_\rho^{-1} D_\rho L_k^{-1} D_\rho^{-1} C_k \tag{4.4}$$

If we can show that the error vectors decay (and possibly grow) like the residual vectors, i.e. if we can show that

$$\|F_k^{(P)} D_\rho^{-1}\| \approx g(k)\epsilon \quad \text{and} \quad \|F_k^{(R)} D_\rho^{-1}\| \approx h(k)\epsilon$$

hold true for some slowly growing functions $g$ and $h$, the error term can be bounded by

$$\|F_k\| \leq g(k)\epsilon\|A\| + h(k)\epsilon\|C_k\| \cdot \|D_\rho L_k^{-1} D_\rho^{-1}\|.$$

If we can show further that $\|C_k\| \approx \|A\|$, we observe that the bound growth mainly depends on the elements of

$$D_\rho L_k^{-1} D_\rho^{-1} = \left(\frac{\|r_j\|}{\|r_i\|}\right)_{ij}.$$

A similar result on the perturbation of the Hessenberg decomposition holds true for the LTPMs. The error matrix $F_k$ in all cases is composed of the relative errors of the single recurrences times the factors used. Our general approach will be based on the finite precision analogue (4.3) of a Hessenberg decomposition.

## 4.3 Outline of Error Analysis

In the last section we have shown that *finite precision Krylov methods* result in *perturbed Hessenberg decompositions*

$$AQ_k - Q_k C_k = M_k - F_k,$$

with $F_k$ small compared to $A$, $C_k$ and $Q_k$. Additionally, $F_k$ may be relative in some sense. This justifies the use perturbed Hessenberg decompositions, or more general, perturbed Krylov decompositions, as basis for an error model. In the remaining part of this chapter we pre-assume only that the governing equation is fulfilled approximately.

A Krylov method consists of a Krylov decomposition and a projection with a second dual basis. Duality follows by induction and is very sensitive to rounding errors. This will be the part we have to examine, that is, we have to investigate *when* and *how* orthonormality, bi-orthonormality or duality is lost. The error analysis of finite precision Krylov methods is intimately connected to the orthogonality properties of the basis vectors. We will mainly consider eigenproblem methods.

We consider two basic approaches to perturbed Krylov decompositions: First we construct *examples* of perturbed Krylov decompositions and categorise the occurring behaviour. Then we look for *connections* to related areas. We can diminish the

resulting gap between observed and predicted behaviour taking the special structure into account.

The *iterative character* of the governing equation is used to construct a propagation formula for the loss of orthogonality. This forms the basis for two backward analysis approaches. Both approaches are based on *splittings* of the matrix $W_k \equiv \hat{Q}_k^H Q_k$, an additive and a multiplicative splitting.

We analyse the linear system, namely the *Sylvester equation*, underlying every Krylov decomposition according to condition and sensitivity. This linear system is a linear system in $Q_k$. We analyse how the Schur and Jordan normal forms of $A$ and $C_k$ can be used to express the Schur and Jordan normal form of the Sylvester equation.

For theoretical purposes of more interest is the deviation in the basis vectors, and not only the size of deviation in inner products. For this reason we consider yet another forward error analysis approach. The Schur and Jordan normal forms of the Sylvester equation indicate the importance of the eigenvalues. We analyse the influences of the errors on the new basis vector $q_{k+1}$ in terms of the eigendecompositions of the system matrix $A$ and the condensed matrix $C_k$. This forward error analysis approach shows that we have a mangling of several methods. The second method converges to a sophisticated *mixture* of desired but yet unconverged and already converged data.

We may ask how the error vector of a certain step affects the recurrence from this step on. This question is addressed in one section. In another section we explore different stopping criteria and their relations to desired values. The main result we whish to prove is that a loss of orthogonality implies prior convergence of the method.

The preceeding analyses are then used to define a class of new Krylov methods that only make sense in a finite precision environment. Here, a *level* of linear independency between basis vectors is defined such that the resulting matrices are *similar* to slightly perturbed projections. In these new methods the level is computed, or more cheaply, guessed. The basis vectors are re-orthogonalised whenever the computed level indicates the necessity to do so.

## 4.4 Where are we?

In this section we are interested in *constructing examples* of perturbed Krylov decompositions. The examples help in understanding the behaviour to be expected in a finite precision run of a Krylov method. To determine behaviour that can occur in a finite precision Krylov method we use the governing equation as simple model. We construct a series of examples of approximately fulfilled governing equations. We start with a very simple example and refine it until it reflects enough of the observed behaviour. We stress that no example comes from an actual finite precision computation.

Every exact governing equation can be perturbed into an approximate. This does not help in understanding what goes wrong. But it serves as *the* building block for the first example.

**Example 4.3 (glued invariant subspaces)** A nice example with remarkable features is constructed using exact governing equations

$$A\tilde{Q}_i - \tilde{Q}_i\tilde{C}_i = 0, \quad i \in \underline{l}.$$

One may think of them as if obtained using a Krylov method of the class investigated. These equations are glued together by introducing small errors. We define

the block matrix $C_m$ by

$$C_m = \begin{pmatrix} \tilde{C}_1 + \Delta_{11} & \Delta_{12} & \Delta_{13} & \cdots & \Delta_{1l} \\ \Delta_{21} & \tilde{C}_2 + \Delta_{22} & \Delta_{23} & \cdots & \Delta_{2l} \\ 0 & \Delta_{32} & \tilde{C}_3 + \Delta_{33} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \Delta_{l-1,l} \\ 0 & \cdots & 0 & \Delta_{l,l-1} & \tilde{C}_l + \Delta_{ll} \end{pmatrix}.$$

The block perturbations $\Delta_{ij}$ have to fit the method characteristics. In case of a symmetric method the perturbations have to be symmetric, i.e.

$$\Delta_{i,i+1} = \Delta_{i+1,i}^H \quad \forall\ i \in \underline{l-1} \qquad \Delta_{ij} = 0 \quad \forall\ |i-j| > 1.$$

They have to be rank matching, i.e. for a block method of block-size one they have to fulfil the constraints

$$\Delta_{i+1,i} = \delta_i e_1 e_{k_i}^T \quad \forall\ i \in \underline{l-1}.$$

With correctly chosen perturbations we can construct

$$\begin{aligned} AQ_m - Q_m C_m &\equiv A\left[\tilde{Q}_1, \ldots, \tilde{Q}_l\right] - \left[\tilde{Q}_1, \ldots, \tilde{Q}_l\right] C_m \\ &= \left[\sum_j \tilde{Q}_j \Delta_{j1}, \ldots, \sum_j \tilde{Q}_j \Delta_{jl}\right] \\ &\equiv F_m, \end{aligned}$$

an equation of type (4.3).

This first example may be enhanced by not only perturbing $C_m$, but also $A$ and $Q_m$. This leads to a model showing essentially the same behaviour.

We could use a single subspace. This shows that we can expect eigenvalue clusters approximating simple and well-conditioned eigenvalues, especially when there are zero blocks in the upper part of $C_m$. At the same time we can expect the columns of $Q_m$ to be linearly dependent, by no means orthogonal.

We remark that the orthogonality pattern in $\hat{Q}_m^H Q_m$ and the small components in the subdiagonal of $C_m$ are closely connected to the occurrence of multiple eigenvalues. Local orthogonality turns out to play a crucial role in the error analysis of short term recurrences. These observations clearly reveal what can be expected from a error analysis using this model of behaviour. The good news is, numerical experiments suggest that finite precision Krylov methods behave like this.

We state an enhancement of the first example for short-term recurrences. In this case we can flip the tridiagonal from both sides, thus reordering the elements.

**Example 4.4 (one flipped subspace)** We restrict attention to short-term methods. We use one single subspace

$$A\tilde{Q} - \tilde{Q}\tilde{C} = 0.$$

We build a matrix $C_m$ like in Example 4.3, where the $\tilde{Q}_i$ are chosen from $\tilde{Q}$ and flipped $\tilde{Q}$ alternating. The matrices $\tilde{C}_i$ change accordingly. In this restricted example we additionally have local orthogonality.

The first two examples suffer from two cross-connected limitations. These are the occurrence of small subdiagonal elements in $C_m$ and the block-structure of the loss of orthogonality in $\hat{Q}_m^H Q_m$.

We can extend this model further by allowing arbitrary similarity transformations in $A$ and $C_m$.

**Example 4.5 (similarity transformations)** We construct a perturbed governing equation with the aid of similarity transformations that change the errors in prescribed ways,

$$\begin{aligned} & AQ_m - Q_m C_m = F_m \\ \Leftrightarrow \quad & U^{-1}\left(UAU^{-1}\right)UQ_m - Q_m W\left(W^{-1}C_m W\right)W^{-1} = F_m \\ \Leftrightarrow \quad & \left(UAU^{-1}\right)UQ_m W - UQ_m W\left(W^{-1}C_m W\right) = UF_m W \\ \Leftrightarrow \quad & \tilde{A}\tilde{Q}_m - \tilde{Q}_m\tilde{C}_m = \tilde{F}_m. \end{aligned}$$

The subspaces need not be from the same class than the Krylov method investigated. This is done by finding the right transformations. The errors in $\tilde{A}$ and $\tilde{C}_m$ have to be chosen to fit the requirements of the methods (structure of $C_m$, i.e. the rank requirements, the symmetry).

This example removes a fault of the previous examples, i.e. the occurrence of small lower diagonal elements which would lead to acceptance of the computed $C_m$ before step $m$ and thus to early termination of the algorithm.

Suppose a similarity transformation has been chosen such that the lower diagonal elements are not small. Then again we observe linearly dependent basis vectors and multiple instances of simple eigenvalues. The convergence of the Ritz values becomes blurred, i.e. smeared over the blocks.

It is an active line of research to find out whether these examples are rich enough in modelling the possible behaviour. If this would turn out to be true, then all computed eigenpairs (eigentriples) would correspond to different slightly perturbed $A$. Formulated in yet another fashion: the behaviour of a finite precision method might correspond to the behaviour of the exact (infinite precision) method when applied to a larger matrix which has only eigentriples that in backward sense are close to eigentriples of $A$.

This question has been answered positively only for the finite precision symmetric Lanczos method by Greenbaum (cf. [Gre89], Theorem 1', page 51). Even in this case the bound proven to be correct seems to be an unrealistic overestimate. In case of long-term recurrences based on the Arnoldi method even part of the usual backward error analysis seems to work correctly. An example of such an analysis may be found in the Ph.d. thesis of Rozložník and the papers cited therein (cf. [Roz97]).

## 4.5 Perturbed Krylov Decompositions

In this section we are concerned with different interpretations and alternate formulations of perturbed Krylov decompositions of type

$$AQ_k - Q_k C_k + F_k = M_k. \tag{4.5}$$

There are various interpretations of equation (4.5). The first interpretations and observations make *no* assumption on the structure of $C_k$. The last observation relies on the Hessenberg structure of the matrix $C_k$.

Most natural is the interpretation of equation (4.5) as subspace relation. The residual is given by $M_k - F_k$. As we have seen in Chapter 1, this residual can be interpreted as backward error bound when $Q_k$ has orthonormal columns. Furthermore, if $A$ and $C_k$ are symmetric, we can use a theorem of Kahan to bound the distance of the eigenvalues of $C_k$ to some eigenvalues of $A$,

$$|\lambda_i - \theta_j| \leq \|M_k - F_k\| \leq |c_{k+1,k}|\|q_{k+1}\| + O(\epsilon) \quad \exists\ i\ \forall\ j.$$

A similar bound involving the smallest singular value of $Q_k$ holds true when $Q_k$ has full column rank. When $Q_k$ is rank deficient, we can think of splitting $Q_k$ into several parts with full rank. Accordingly we would have to split $C_k$ and to use known relations between $C_k$ and the splitting matrices.

We can think of the subspace equation (4.5) as an exact subspace equation for perturbed $A$. We assume the existence of the pseudo-inverse of $Q_k$, i.e. we assume that $Q_k$ has full rank. Then we can formulate the exact subspace relation

$$\left(A - (M_k - F_k)Q_k^\dagger\right) Q_k - Q_k C_k = AQ_k - M_k + F_k - Q_k C_k = 0.$$

This proves that the eigenvalues of $C_k$ are among the eigenvalues of

$$A_k = A - (M_k - F_k)Q_k^\dagger.$$

This point of view brings in structured homotopic perturbation theory. Researchers at CERFACS used homotopic perturbation theory as a tool for the understanding of the Arnoldi method and other Krylov methods (cf. [CC00, CCTP00]).

In all cases the candidate for the invariant subspace is spanned by the columns of $Q_k$. The matrix $Q_k$ is determined linearly by the matrices $A$ and $C_k$ and the right-hand side $M_k - F_k$. This linear equation is known as *Sylvester equation* and can be re-written to expose its linear character,

$$(I_k \otimes A - C_k^T \otimes I_n)\mathrm{vec}(Q_k) = \mathrm{vec}(M_k - F_k).$$

The condition of the Sylvester equation is given by the separation of the matrices $A$ and $C_k$. The *separation of two matrices* $A$ and $C_k$ has been defined in Chapter 1 to be

$$\mathrm{sep}(A, C_k) \equiv \min_X \frac{\|AX - XC_k\|}{\|X\|}.$$

The norm may be arbitrary, of special interest are the cases of the 2-norm and the Frobenius norm. The Frobenius norm can be expressed in terms of singular values, the equality

$$\mathrm{sep}_F(A, C_k) = \sigma_{\min}(I_k \otimes A - C_k^T \otimes I_n)$$

holds true. The exact computation of the separation has a computational amount of $O(n^3k^3)$ and is usually cheaply estimated using $O(n^2k)$ operations (cf. [BDM91]). In the context of large sparse systems and Krylov methods such a computational amount is *prohibitive*.

Krylov decompositions, or more general, subspace equations, are closely connected to block similarity of block matrices and perturbations of these. This follows by rewriting equation (4.5) by adding a trivial block row and column as

$$\begin{pmatrix} C_k & 0 \\ F_k & A \end{pmatrix} \begin{pmatrix} I_k & 0 \\ Q_k & I_n \end{pmatrix} - \begin{pmatrix} I_k & 0 \\ Q_k & I_n \end{pmatrix} \begin{pmatrix} C_k & 0 \\ M_k & A \end{pmatrix} = 0.$$

This is an exact subspace equation in higher dimension. The triangular block matrix can be inverted explicitly,

$$\begin{pmatrix} I_k & 0 \\ Q_k & I_n \end{pmatrix}^{-1} = \begin{pmatrix} I_k & 0 \\ -Q_k & I_n \end{pmatrix}.$$

This is similar to a block Gauß algorithm. This far, we know that the block matrices with $F_k$ and with $M_k$ in the lower left corner are block similar,

$$\begin{pmatrix} C_k & 0 \\ M_k & A \end{pmatrix} = \begin{pmatrix} I_k & 0 \\ -Q_k & I_n \end{pmatrix} \begin{pmatrix} C_k & 0 \\ F_k & A \end{pmatrix} \begin{pmatrix} I_k & 0 \\ Q_k & I_n \end{pmatrix}.$$

The block similarity implies that we have a change in the eigenvectors of the block matrix. For the moment, we restrict ourselves to the infinite precision case. In infinite precision, the block matrix Jordan normal form is given by

$$\begin{pmatrix} C_k & 0 \\ 0 & A \end{pmatrix} \begin{pmatrix} S_k & 0 \\ 0 & V \end{pmatrix} = \begin{pmatrix} S_k & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} J_\Theta & 0 \\ 0 & J_\Lambda \end{pmatrix}.$$

Due to the block similarity the eigenvectors of the block matrix are changed like

$$\begin{pmatrix} I_k & 0 \\ -Q_k & I_n \end{pmatrix} \begin{pmatrix} S_k & 0 \\ 0 & V \end{pmatrix} = \begin{pmatrix} S_k & 0 \\ -Q_k S_k & V \end{pmatrix} = \begin{pmatrix} S_k & 0 \\ -Y_k & V \end{pmatrix}.$$

At this point, the matrix of Ritz vectors naturally enters the scene. The *condition* of the block eigenvector matrix changes, depending on the *convergence* of $Y_k$ to parts of $\hat{V}^H$, the matrix of left eigenvectors of $A$.

In finite precision the matrix $F_k$ is interpreted as a perturbation. This is not rigid, since the deviation alters the entries of $C_k$. But, for a limited number of steps, the influence on $C_k$ may be negligible. In the finite precision case, we use in step $k$ the eigenvector matrix

$$\begin{pmatrix} S_k & 0 \\ -Q_k S_k & V \end{pmatrix} = \begin{pmatrix} S_k & 0 \\ -Y_k & V \end{pmatrix} = \begin{pmatrix} \hat{S}_k^H & 0 \\ \hat{V}^H Q_k & \hat{V}^H \end{pmatrix}^{-1}$$

corresponding to the *unperturbed* block matrix as approximate eigenvector matrix. How is this quantity related to the exact eigenvector matrix?

We assume $X_k$ to be the solution of the *Sylvester equation*

$$AX_k - X_k C_k = F_k.$$

Then the eigendecomposition of the perturbed block matrix is given by

$$\begin{pmatrix} C_k & 0 \\ F_k & A \end{pmatrix} \begin{pmatrix} S_k & 0 \\ -X_k S_k & V \end{pmatrix} = \begin{pmatrix} S_k & 0 \\ -X_k S_k & V \end{pmatrix} \begin{pmatrix} J_\Theta & 0 \\ J_\Lambda & 0 \end{pmatrix}.$$

After transforming the eigenvectors according to the block similarity we obtain the true eigenvector matrix

$$\begin{pmatrix} S_k & 0 \\ -(Q_k + X_k) S_k & V \end{pmatrix} = \begin{pmatrix} \hat{S}_k^H & 0 \\ \hat{V}^H (Q_k + X_k) & \hat{V}^H \end{pmatrix}^{-1}.$$

The error in the eigenvector matrix depends crucially on the magnitude of $X_k$. The magnitude of $X_k$ can be bounded using the separation of $A$ and $C_k$, we have that

$$\|X_k\| \leq \mathrm{sep}(A, C_k) \|F_k\|$$

holds true.

These comments reveal that the error analysis of finite precision Krylov subspace methods is closely related to eigenvector sensitivity of block matrices. We remark that this connection is well-known and may be found in the textbook by Horn and Johnson (cf. [HJ94], Theorem 4.4.22). In the block matrix interpretation the *computed* matrix $C_k$ comes into play like the system matrix $A$, they can not be distinguished any more. This may be considered as an indicator that part of the computed quantities tries actually to converge to itself.

All considerations thus far were independent of any special structure. Thus, in general, the results will be weak, since knowledge, for instance on the Hessenberg structure and the length of the columns of $Q_k$ is not used. This far, only *static*

information has been used. The next result, to some extent, honours the *iterative* character of the methods.

It is possible to construct other Krylov decompositions with the aid of polynomials. These Krylov decompositions may be subject to a larger perturbation, but due to the special structure they provide useful information. The results rely on the fact that polynomials of Hessenberg matrices have a very special structure.

We first derive an expression on the accuracy of the subspace spanned by the columns of $Q_k$ as approximate invariant subspace of *polynomials* of $A$. The polynomial case is treated as linear combination of the monomial case. Our proof is based on the factorisation

$$\begin{aligned} a^{l+1} - b^{l+1} &= a^{l+1} - a^l b + a^l b - \cdots + a b^l - b^{l+1} \\ &= \left( \sum_{i=0}^{l} a^{k-i} b^i \right) (a - b) \end{aligned} \tag{4.6}$$

of the bivariate polynomial $a^{l+1} - b^{l+1}$. This result can be generalised for commuting matrices. Observe that

$$(I \otimes A)\left(C_k^T \otimes I\right) = C_k^T \otimes A = \left(C_k^T \otimes I\right)(I \otimes A)$$

holds true, i.e. the matrices $I \otimes A$ and $C_k^T \otimes I$ commute. The powers of these matrices can be transformed to powers in $A$ and $C_k^T$, respectively, since

$$(A \otimes B)^{l+1} = A^{l+1} \otimes B^{l+1}$$

holds true for *general* square $A$ and $B$. We apply the polynomial factorisation (4.6) to the term

$$\left(I \otimes A^{l+1} - (C_k^T)^{l+1} \otimes I\right) = \left[ \sum_{i=0}^{l} (I \otimes A)^{l-i} \left(C_k^T \otimes I\right)^i \right] \left(I \otimes A - C_k^T \otimes I\right).$$

Next, we apply both sides of this matrix equation to the vector $\mathrm{vec}(Q_k)$ and subsequently use the governing equation

$$\left(I \otimes A - C_k^T \otimes I\right) \mathrm{vec}(Q_k) = \mathrm{vec}(M_k - F_k).$$

We re-write the Sylvester form to an ordinary matrix equation and state the result as lemma.

**Lemma 4.6** *Let $A \in \mathbb{K}^{n \times n}$ and $C_k \in \mathbb{K}^{k \times k}$ be* general *matrices. Suppose that the equation*

$$AQ_k - Q_k C_k = M_k - F_k$$

*holds true for some matrices $Q_k, M_k, F_k \in \mathbb{K}^{n \times k}$. Then for* all *$l \in \mathbb{N}$*

$$\begin{aligned} A^{l+1} Q_k - Q_k C_k^{l+1} &= \sum_{i=0}^{l} A^{l-i} (M_k - F_k) C_k^i \\ &= \sum_{i=0}^{l} A^{l-i} M_k C_k^i - \sum_{i=0}^{l} A^{l-i} F_k C_k^i \end{aligned} \tag{4.7}$$

*holds true.*

Summing monomials we obtain a similar result for polynomials in $A$. We refuse to state this result in full generality, since we are only interested in a *specially structured* case. In what follows, the matrix $C_k$ is an upper unreduced Hessenberg

matrix and $M_k$ is given by a rank-one matrix that has the structure $M_k = r_k e_k^T$ for some $r_k \in \mathbb{K}^n$. Let $I_j$ denote the matrix consisting of the first $j$ columns of $I$. Let $\mathcal{I}_j$ denote the space spanned by the first $j$ standard unit vectors $e_1, \ldots, e_j$,

$$\mathcal{I}_j = \operatorname{span}(I_j) = \operatorname{span}(e_1, \ldots, e_j).$$

Multiplication with an unreduced upper Hessenberg matrix $C_k$ maps any vector $z_j \in \mathcal{I}_j$, $j < k$, to a vector $z_{j+1} \in \mathcal{I}_{j+1}$, i.e. by induction

$$C_k^i \mathcal{I}_j \subset \mathcal{I}_{j+i}.$$

When we take a closer look at the columns of $M_k C_k^i$,

$$M_k C_k^i \mathcal{I}_j = r_k e_k^T \mathcal{I}_{j+i},$$

we observe that the columns $1, \ldots, j$ are zero when $j + i < k$. This proves that the first $k - i - 1$ columns are identical zero. Now, an easy calculation shows that

$$C_k^{k-1} e_1 = \underbrace{C_k \cdots C_k}_{k-1 \text{ times}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \underbrace{C_k \cdots C_k}_{k-2 \text{ times}} \begin{pmatrix} \star \\ c_{21} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \quad \cdots \quad = \begin{pmatrix} \star \\ \star \\ \vdots \\ \star \\ \zeta \end{pmatrix}$$

holds true, where $\zeta$ is given by

$$\zeta = \prod_{p=1}^{k-1} c_{p+1,p}.$$

Thus, we have shown that

$$\begin{aligned} M_k C_k^0 e_1 \quad = \quad M_k C_k^1 e_1 \quad = \quad \ldots \quad = \quad M_k C_k^{k-2} e_1 \quad &= \quad 0, \\ M_k C_k^{k-1} e_1 \quad &= \quad r_k \prod_{p=1}^{k-1} c_{p+1,p} \end{aligned}$$

holds true. More generally, we observe that

$$\begin{aligned} M_k C_k^0 e_l \quad = \quad M_k C_k^1 e_l \quad = \quad \ldots \quad = \quad M_k C_k^{k-l-1} e_l \quad &= \quad 0, \\ M_k C_k^{k-l} e_l \quad &= \quad r_k \prod_{p=l}^{k-1} c_{p+1,p} \end{aligned}$$

holds true.

Putting all this together, we have proven the following result:

**Theorem 4.7** *Let $A \in \mathbb{K}^{n \times n}$. Let $C_k \in \mathcal{H}(k)$. Let $M_k$ be given as the rank-one matrix $M_k \equiv c_{k+1,k} q_{k+1} e_k^T$. Suppose that the equation*

$$AQ_k - Q_k C_k = M_k - F_k$$

*holds true for some matrices $Q_k, F_k \in \mathbb{K}^{n \times k}$. Then for $l \in \underline{k}$*

$$A^{k-l+1} Q_l - Q_k C_k^{k-l+1} I_l = \left( \prod_{p=l}^{k} c_{p+1,p} \right) q_{k+1} e_l^T - \sum_{i=0}^{k-l} A^{k-l-i} F_k C_k^i I_l$$

*holds true.*

*Suppose further that* $\chi \in \mathbb{P}_{k-l+1}$ *is given by*

$$\chi(\lambda) \equiv \alpha_{k-l+1}\lambda^{k-l+1} + \sum_{j=0}^{k-l} a_j \lambda^j.$$

*Then by linearity arguments in the* polynomial *case*

$$\begin{aligned} \chi(A)Q_l - Q_k\chi(C_k)I_l &= \alpha_{k-l+1}\left(\prod_{p=l}^{k} c_{p+1,p}\right) q_{k+1}e_l^T \\ &- \sum_{j=0}^{k-l}\alpha_{j+1}\sum_{i=0}^{j} A^{j-i}F_kC_k^iI_l \end{aligned} \tag{4.8}$$

*holds true. We stress the fact that the constant term* $\alpha_0$ *of the polynomial* $\chi$ *has no influence on the right-hand side of equation (4.8).*

Two cases deserve special attention. The first case is if the polynomial $\chi$ is chosen such that

$$\chi(C_k) = U\begin{pmatrix} \tilde{C}_l & \star \\ 0 & \star \end{pmatrix} U^H \tag{4.9}$$

holds true for some unitary matrix $U \in \mathbb{K}^{k\times k}$. Then this polynomial can be used to construct a smaller Krylov decomposition with a change of the set of basis vectors. We state this case as a corollary:

**Corollary 4.8** *Let* $A$, $C_k$, $M_k$, $Q_k$ *and* $F_k$ *be given as in the preceeding theorem. Suppose that the polynomial* $\chi$ *has been chosen such that equation (4.9) holds. Define* $\tilde{Q}_l \equiv Q_kUI_l$.

*Then*

$$\chi(A)\tilde{Q}_l - \tilde{Q}_l\tilde{C}_l = \tilde{q}_{l+1}\tilde{c}_{l+1,l}e_l^T - \sum_{j=0}^{k-l}\alpha_{j+1}\sum_{i=0}^{j} A^{j-i}\tilde{F}_l\tilde{C}_l^i$$

*holds true. Here we have used the abbreviations*

$$\tilde{q}_{l+1} \equiv q_{k+1}, \qquad \tilde{c}_{l+1,l} \equiv \alpha_{k-l+1}\prod_{p=l}^{k} c_{p+1,p} \quad \text{and} \quad \tilde{F}_l \equiv F_kUI_l.$$

This corollary forms the basis of implicitly restarted methods, like the implicitly restarted Arnoldi method, IRA or IRAM for short.

The other case that deserves special attention is the general case for $l = 1$. The result in this case becomes a useful tool in the convergence analysis of Krylov subspace methods, especially in the solution process of linear systems [TY00]. We also state this case as a corollary:

**Corollary 4.9** *Let* $A$, $C_k$, $M_k$, $Q_kQ$ *and* $F_k$ *be given as in Theorem 4.7. Suppose further that* $l = 1$ *and* $\chi \in \mathbb{P}_k$ *is given by*

$$\chi(\lambda) = \alpha_k\lambda^k + \sum_{j=0}^{k-1}\alpha_j\lambda^j$$

*Then*

$$\chi(A)q_1 - Q_k\chi(C_k)e_1 = \alpha_k\left(\prod_{p=1}^{k} c_{p+1,p}\right) q_{k+1} - \sum_{j=0}^{k-1}\alpha_{j+1}\sum_{i=0}^{j} A^{j-i}F_kC_k^ie_1$$

*holds true. Setting* $\chi \equiv \chi_{C_k}$ *annihilates the second term,*

$$\chi_{C_k}(A)q_1 - \left(\prod_{p=1}^{k} c_{p+1,p}\right) q_{k+1} = -\sum_{j=0}^{k-1} \alpha_{j+1} \sum_{i=0}^{j} A^{j-i} F_k C_k^i e_1. \tag{4.10}$$

*When* $\det(C_k) \neq 0$*, we can construct a polynomial* $\chi_{k-1} \in \mathbb{P}_{k-1}$ *such that* $\chi_{k-1}(C_k) = \|r_0\| C_k^{-1}$*. Defining* $x_k \equiv \|r_0\| Q_k C_k^{-1} e_1$*,*

$$\chi_{k-1}(A)q_1 - x_k = -\sum_{j=0}^{k-1} \alpha_{j+1} \sum_{i=0}^{j} A^{j-i} F_k C_k^i e_1. \tag{4.11}$$

*holds true. In both cases the coefficients* $\alpha_j$ *depend on* $C_k$*.*

The right-hand sides of equations (4.10) and (4.11) are zero in infinite precision. In finite precision they describe the occurring deviation. The error terms are complicated, because of the dependence of the coefficients of the polynomials $\chi_{C_k}$ and $\chi_{k-1}$ on the *computed* matrices $C_k$. For this reason we take another approach in the next sections to understand the local and global deviation occurring in finite precision.

## 4.6 Deviation from Nowhere?

In infinite precision, Krylov methods are based on Hessenberg decompositions. The computation of the decomposition is chosen such that by an *inductive* argument certain relations among *all* basis vectors hold true. We focus on the most important relations among the basis vectors, i.e. the orthogonality and bi-orthogonality relations.

In finite precision, the Hessenberg decomposition has to be replaced by a perturbed variant. In this section we examine what happens to the other relations. It is obvious that when the (bi-)orthogonality is not forced explicitly by the method, it will cease to hold. This was the reason why the methods were often used with full re-(bi-)orthogonalisation in the sixties.

In this section we examine the dependencies between computed quantities and the loss of orthogonality. First, we develop a recurrence formula for the loss of orthogonality. Here, the modus operandi is based on an iterative point of view which is more suitable to the iterative use of the methods. We conclude with two backward results, based on an additive and a multiplicative splitting of the matrix of the loss of orthogonality.

We consider long-term recurrences that fulfil the relation

$$AQ_k = Q_{k+1}\underline{C}_k - F_k, \tag{4.12}$$

and coupled recurrences based on (4.12) and a *second* perturbed Hessenberg decomposition that additionally fulfils the relation

$$A^H \hat{Q}_k = \hat{Q}_{k+1}\underline{\hat{C}}_k - \hat{F}_k. \tag{4.13}$$

These coupled recurrences usually will be short-term recurrences.

For reasons of brevity we introduce the matrix $W_m = \hat{Q}_m^H Q_m$. In infinite precision this matrix is identical to the identity matrix $I_m$. The columns of $W_m$ are denoted by $w_l$, $l \in \underline{m}$, and the elements by $w_{jl}$, $l, j \in \underline{m}$. Leading submatrices of order $k$ are denoted by $W_k$.

We start with the relations between the matrix $W_{k+1} = \hat{Q}_{k+1}^H Q_{k+1}$ and the computed quantities.

**Lemma 4.10 (The Error Relations)** *In case of a long-term recurrence the matrix* $W_{k+1} = \hat{Q}_{k+1}^H Q_{k+1}$ *fulfils the relation*

$$W_{k+1}\underline{C}_k = \hat{Q}_{k+1}^H A Q_k - \hat{Q}_{k+1}^H F_k. \tag{4.14}$$

*In case of coupled (short-term) recurrences the matrix* $W_{k+1} = \hat{Q}_{k+1}^H Q_{k+1}$ *additionally fulfils the relation*

$$\underline{\hat{C}}_k^H W_{k+1} = \hat{Q}_k^H A Q_{k+1} - \hat{F}_k^H Q_{k+1}. \tag{4.15}$$

*This implies that in case of coupled (short-term) recurrences* $W_{k+1} = \hat{Q}_{k+1}^H Q_{k+1}$ *fulfils the fundamental relation*

$$\underline{\hat{C}}_k^H W_{k+1,k} - W_{k,k+1}\underline{C}_k = \hat{Q}_k^H F_k - \hat{F}_k^H Q_k. \tag{4.16}$$

**Proof.** Equation (4.14) is obtained by left multiplication of the perturbed Hessenberg decomposition (4.12) with $\hat{Q}_{k+1}^H$. For latter purposes we strip of the last row to obtain an equation involving quadratic matrices,

$$W_{k,k+1}\underline{C}_k = \hat{Q}_k^H A Q_k - \hat{Q}_k^H F_k. \tag{4.17}$$

Now we consider the case of two coupled (short-term) recurrences. We transform the second perturbed Hessenberg decomposition (4.13) by multiplication with $Q_{k+1}^H$ from the left. Forming the Hermitian of the resulting equation we derive equation (4.15). We strip of the last column to obtain an equation involving quadratic matrices,

$$\underline{\hat{C}}_k^H W_{k+1,k} = \hat{Q}_k^H A Q_k - \hat{F}_k^H Q_k. \tag{4.18}$$

Subtracting equation (4.17) from equation (4.18) we arrive at equation (4.16). This finishes the proof. □

We intend to show that for *any* Hessenberg decomposition the loss of orthogonality or other deviation is a natural consequence of the recurrence. We first restrict ourself to dual methods, i.e. methods where in infinite precision $\hat{Q}_m^H Q_m = I_m$ holds true.

The first approach is a simple forward error analysis. It relates the loss of orthogonality, given by the matrix $W_{k+1} - I_{k+1}$, to the influences. It results in two simple amplification recurrences. The trick is to insert the computed $\underline{C}_k$ on both sides of the equations.

**Lemma 4.11 (The Error Sources)** *In case of a long-term recurrence the relation*

$$\Big(W_{k+1} - I_{k+1}\Big)\underline{C}_k = \Big(\hat{Q}_{k+1}^H A Q_k - \underline{C}_k\Big) - \hat{Q}_{k+1}^H F_k \tag{4.19}$$

*holds true. In case of coupled (short-term) recurrences, additionally the relation*

$$\begin{aligned}\underline{\hat{C}}_k^H \Big(W_{k+1,k} - I_{k+1,k}\Big) - \Big(W_{k,k+1} - I_{k,k+1}\Big)\underline{C}_k = \\ C_k - \hat{C}_k^H + \hat{Q}_k^H F_k - \hat{F}_k^H Q_k\end{aligned} \tag{4.20}$$

*holds true.*

**Proof.** The relation (4.19) follows by subtracting $\underline{C}_k$ on both sides of (4.14). The relation (4.20) follows by subtracting

$$\hat{C}_k^H - C_k = \underline{\hat{C}}_k^H I_{k+1,k} - I_{k,k+1}\underline{C}_k$$

on both sides of (4.16). □

The lemma above stated more informally says that the loss of orthogonality weighted by the *computed* matrix $\underline{C}_k$ is equal to the sum of the projection error and a small error term.

This far, all considerations did not honour the iterative character of the methods. This fault is removed by the transformation of the Hessenberg decomposition to the form of the governing equation,

$$M_k = q_{k+1}c_{k+1,k}e_k^T = AQ_k - Q_kC_k + F_k.$$

When we multiply the Hessenberg decomposition from the left by $\hat{Q}_k^H$, we obtain an iterative *matrix expression* for the loss of orthogonality.

**Theorem 4.12 (The Matrix Error Recurrences)** *A matrix expression of the recurrence of the loss of orthogonality is given by*

$$\begin{aligned}\hat{Q}_k^H M_k &= \hat{Q}_k^H AQ_k - \hat{Q}_k^H Q_k C_k + \hat{Q}_k^H F_k \\ &= \left(\hat{Q}_k^H AQ_k - C_k\right) \\ &- \left(W_k - I_k\right)C_k \\ &+ \hat{Q}_k^H F_k. \end{aligned} \tag{4.21}$$

*In case of methods based on coupled (short-term) recurrences the loss of orthogonality additionally fulfils*

$$\begin{aligned}\hat{Q}_k^H M_k - \hat{M}_k^H Q_k &= \left(\hat{C}_k^H - C_k\right) \\ &+ \hat{C}_k^H\left(W_k - I_k\right) - \left(W_k - I_k\right)C_k \\ &+ \left(\hat{Q}_k^H F_k - \hat{F}_k^H Q_k\right). \end{aligned} \tag{4.22}$$

Next, we multiply equations (4.21) and (4.22) from the right by $e_k$ and right-divide them by $c_{k+1,k}$. This results in a *vector recurrence* for the loss of orthogonality.

**Corollary 4.13 (The Vector Error Recurrences)** *A vector expression of the recurrence of the loss of orthogonality is given by*

$$\begin{aligned}\hat{Q}_k^H q_{k+1} &= \left(\hat{Q}_k^H Aq_k - \hat{Q}_k^H Q_k c_k + \hat{Q}_k^H f_k\right) c_{k+1,k}^{-1} \\ &= \left(\hat{Q}_k^H Aq_k - c_k\right) c_{k+1,k}^{-1} \\ &- \left((W_k - I_k)c_k\right) c_{k+1,k}^{-1} \\ &+ \left(\hat{Q}_k^H f_k\right) c_{k+1,k}^{-1}. \end{aligned} \tag{4.23}$$

*This is a recurrence on the* columns *of the matrix $W_m - I_m$.*

*In case of methods based on coupled (short-term) recurrences the loss of orthogonality additionally fulfils*

$$\begin{aligned}\hat{Q}_k^H q_{k+1} &= \left(\hat{C}_k^H w_k - W_k c_k + \hat{Q}_k^H f_k - \hat{F}_k^H q_k + \hat{M}_k^H q_k\right) c_{k+1,k}^{-1} \\ &= \left(\hat{C}_k^H e_k - c_k\right) c_{k+1,k}^{-1} \\ &+ \left(\hat{C}_k^H (w_k - e_k) - (W_k - I_k)c_k\right) c_{k+1,k}^{-1} \end{aligned}$$

$$+ \quad \left(\hat{Q}_k^H f_k - \hat{F}_k^H q_k\right) c_{k+1,k}^{-1}$$
$$+ \quad \left(\hat{M}_k^H q_k\right) c_{k+1,k}^{-1}. \tag{4.24}$$

*Like before, this is a recurrence on the* columns *of the matrix* $W_m - I_m$. *Analogously we obtain a second recurrence on the* rows *of the matrix* $W_m - I_m$.

In case of a method based on coupled (short-term) recurrences we observe that the term involving $\hat{M}_k$ has *no* influence on the first $k-1$ equations, hence

$$\begin{aligned}
\hat{q}_j^H q_{k+1} &= \left(e_j^T \hat{C}_k^H \hat{Q}_k^H q_k - \hat{q}_j^H Q_k c_k + \hat{q}_j^H f_k - \hat{f}_j^H q_k\right) c_{k+1,k}^{-1} \\
&= \left(\sum_{l=1}^{j+1} \check{c}_{lj} \hat{q}_l^H q_k - \sum_{l=1}^{k} \hat{q}_j^H q_l c_{lj}\right) c_{k+1,k}^{-1} \\
&+ \left(\hat{q}_j^H f_k - \hat{f}_j^H q_k\right) c_{k+1,k}^{-1}
\end{aligned}$$

holds true for all $j < k$. The $k$th equation has the additional term $\check{c}_{k+1,k} \hat{q}_{k+1}^H q_k$. Usually, the method will be such that we have local orthogonality or local duality and this term is negligible.

Representations (4.23) and (4.24) reveal *three sources* for the occurrence of any loss of orthogonality. When we consider methods based on long-term recurrences, the *first ingredient* is the accuracy of the last computed column of $C_k$,

$$\hat{Q}_k^H A q_k - c_k = \left(\hat{Q}_k^H A Q_k - C_k\right) e_k.$$

In a long-term recurrence like Arnoldi, this is just the way of computation of the components of $C_k$. In this case it seems justified that this part is small. In case of coupled (short-term) recurrences, the *first ingredient* measures tridiagonal structure and whether the matrices $\hat{C}_k$ and $C_k$ are adjoint,

$$\hat{C}_k^H e_k - c_k = \left(\hat{C}_k^H - C_k\right) e_k.$$

Usually this part is identical zero, even in finite precision.

The *second ingredient* in case of long-term recurrences is the loss of orthogonality that occurred prior to step $k$ times the last computed column of $C_k$,

$$-\left(\hat{Q}_k^H Q_k - I_k\right) c_k = -\left(W_k - I_k\right) c_k.$$

For coupled (short-term) recurrences it is the amplification of the loss of orthogonality that occurred prior to step $k$ by the matrices $\hat{C}_k^H$ and $C_k$ according to

$$\hat{C}_k^H (w_k - e_k) - (W_k - I_k) c_k = \left(\hat{C}_k^H (W_k - I_k) - (W_k - I_k) C_k\right) e_k.$$

These parts are the ones that may cause severe deviation. We remark that they depend on the *computed matrices* $\hat{C}_k^H$ and $C_k$.

The *third and last ingredient* is a measure of the orthogonality of the actual error on the previously computed basis vectors, $\hat{Q}_k^H f_k$. The actual error in general will be small. The size of this term mostly depends on the length of the basis vectors. All three terms are summed up and amplified by the inverse of residual estimator $c_{k+1,k}^{-1}$. It is well-known that deviation may occur when any $c_{k+1,k}$ becomes small (cf. [GvL96]).

The methods that are not bi-orthogonal can be inserted into this setting by replacing the identity by $X_k$, where $X_k$ is the exact left hand side of the pencil of the projection. Essentially the same results hold true.

**Remark 4.14** The only terms we have influence on are the *accuracy of the vector recurrence* and the *accuracy of the non-zero moments*. Suppose these terms are 'as small as possible' and 'random'. Then a severe deviation can only arise due to the remaining term. This remaining term is given by

$$-\Big(W_k - I_k\Big)C_k \quad \text{or} \quad \hat{C}_k^H\Big(W_k - I_k\Big) - \Big(W_k - I_k\Big)C_k.$$

Thus, severe deterioration is due mainly to the amplification of previous errors. Furthermore, this amplification is intrinsic, i.e. we can not influence the essential behaviour without modifying the matrices $C_k$ and $\hat{C}_k^H$.

So, we can finally give a partial answer to the question raised in the title of this section:

> Yes, a *deviation from* the exact quantities may occur, but we *know* precisely *where* it has its origins.

We might be interested in an explicit computation of the matrix of the loss of orthogonality.

**Lemma 4.15 (Explicit Computation of $W_{k+1} - I_{k+1}$)** *The matrix $W_{k+1} - I_{k+1}$ in any case fulfils the linear system of equations*

$$\begin{aligned}(\underline{C}_k^T \otimes I_{k+1})vec(W_{k+1} - I_{k+1}) &= vec(\hat{Q}_{k+1}^H A Q_k - \underline{C}_k) \\ &- vec(\hat{Q}_{k+1}^H F_k). \end{aligned} \tag{4.25}$$

*Equation (4.25) enables us to compute* explicitly *the matrix $W_{k+1} - I_{k+1}$* whenever *we have access to the matrices $\hat{Q}_{k+1}^H A Q_k - \underline{C}_k$ and $\hat{Q}_{k+1}^H F_k$.*

*In case of coupled (short-term) recurrences the matrix $W_k - I_k$ additionally fulfils the system of equations*

$$\begin{aligned}(I_k \otimes \hat{C}_k - C_k^T \otimes I_k)vec(W_k - I_k) &= vec(C_k - \hat{C}_k^H) \\ &- vec(\hat{M}_k^H Q_k - \hat{Q}_k^H M_k) \\ &- vec(\hat{F}_k^H Q_k - \hat{Q}_k^H F_k) \end{aligned} \tag{4.26}$$

**Proof.** Equations (4.25) and (4.26) are just the Sylvester equational forms of equations (4.19) and (4.22). □

Normally, we have no information at hand on the *lower part* of the matrix $\hat{Q}_{k+1}^H A Q_k - \underline{C}_k$. Usually, we have some extra information about the diagonal elements of $W_{k+1} - I_{k+1}$. For *some* rows of the huge linear systems we can bound the right-hand side. This is used in an *additive splitting* of $W_{k+1} - I_{k+1}$. We emphasise that the additive splitting approach does *not* depend on the *rank* of $W_{k+1}$.

We consider two special cases. First, we consider as an example the Arnoldi method. In this case we know that the matrix $W_{k+1}$ is HPD. By the normalisation we know that the diagonal elements of $W_{k+1} - I_{k+1}$ are close to zero. This fixes the degrees of freedom.

**Theorem 4.16 (The Loss in Arnoldi's Method)** *Let $A \in \mathbb{K}^{n\times n}$. Let the matrices $Q_{k+1}$, $Q_k$ and $\underline{C}_k$ be computed by the Arnoldi method in finite precision. Suppose that the error matrix $F_k$ defined by*

$$AQ_k - Q_{k+1}\underline{C}_k = -F_k$$

*has small columns. Suppose further that the computed moments are accurate, i.e. that*

$$q_j^H A q_k \approx c_{jk} \quad \forall\ j \in \underline{k}.$$

*The columns of $Q_{k+1}$ will approximately have unit length. Let $\mathbf{R}_k$ be defined to be the restriction operator*

$$\mathbf{R}_k : \mathbb{K}^{(k+1)k} \to \mathbb{K}^{(k+1)k/2}$$

*that maps vec($\underline{C}_k$) onto the vector that consists of the elements of $\underline{C}_k$ from the strictly upper triangular part, indexed by columns.*

*Then*

$$\begin{aligned} \mathbf{R}_k((\underline{C}_k^T \otimes I_{k+1})\mathrm{vec}(W_{k+1} - I_{k+1})) &= \mathbf{R}_k(\mathrm{vec}(\hat{Q}_{k+1}^H A Q_k - \underline{C}_k)) \\ &- \mathbf{R}_k(\mathrm{vec}(\hat{Q}_{k+1}^H F_k)), \\ \mathrm{diag}(W_{k+1} - I_{k+1}) &= \text{small}. \end{aligned}$$

*is an* $\mathbb{R}$–linear *system of equations of size $(k+1)k/2$ that describes the loss of orthogonality in Arnoldi's method as the solution vector of a matrix equation which system matrix is composed of the computed moments and the right hand-side consists solely of boundable small quantities.*

**Proof.** We first count the number of unknowns and equations. The numbers are collected in Table 4.3. The number of unknowns is due to the symmetry of $W_{k+1} = W_{k+1}^H$.

| | |
|---|---|
| unknowns in $W_{k+1}$ : | $1 + 2 + \cdots + k + (k+1)$ |
| equations from (4.25) : | $1 + 2 + \cdots + k$ |
| equations from normalisation : | $(k+1)$ |

Table 4.3: The number of unknowns and equations for Arnoldi's method

The arising system is *square* and of size $(k+1)k/2$. When $\mathbb{K} = \mathbb{R}$, the system is linear, since in this case the unknowns in $W_{k+1}$ are connected via $w_{ij} = w_{ji}$, $i, j \in \underline{k+1}$. When $\mathbb{K} = \mathbb{C}$, we can still construct a real linear system of equations by doubling the dimension, but the corresponding complex system of equations is *not* $\mathbb{C}$–linear. This is due to the fact that conjugation is $\mathbb{R}$–linear, but not $\mathbb{C}$–linear and the elements in $W_{k+1}$ are connected via $w_{ij} = \overline{w_{ji}}$, $i, j \in \underline{k+1}$. □

In general, this approach results in formulae for the loss of *local* orthogonality. In this approach we use the fact that the diagonal elements are close to one by the normalisation and the computed moments are accurate. It is applicable to restarted and truncated methods and the symmetric Lanczos method. When $W_{k+1}$ no longer is Hermitian we have coupled recurrences. This again fixes the degrees of freedom and can be applied to ensure local duality in the (non-symmetric) Lanczos method.

As second case, we consider the (non-symmetric) Lanczos method. In case of such coupled *short-term* recurrences we can use a different approach of additive splitting to obtain information on the *global* orthogonality. This approach is based on the special structure of the computed matrices and an additive splitting of the smaller matrix $W_k - I_k = \hat{Q}_k^H Q_k - I_k$.

**Theorem 4.17 (Paige, Bai)** *Suppose that $\hat{C}_k^H = C_k = T_k$ is tridiagonal. Suppose further that $W_k - I_k = L_k + D_k + R_k$, where $L_k$ is used to denote the strictly lower triangular part, $D_k$ the diagonal and $R_k$ the strictly upper triangular part of $W_k - I_k$.*

*Then equation (4.22) can be transformed to the form*

$$\begin{aligned} T_k R_k - R_k T_k &= \mathrm{triu}(\hat{Q}_k^H M_k - \hat{M}_k^H Q_k) \\ &- \mathrm{triu}(T_k L_k - L_k T_k) \\ &- \mathrm{triu}(T_k D_k - D_k T_k) \\ &- \mathrm{triu}(\hat{Q}_k^H F_k - \hat{F}_k^H Q_k). \end{aligned} \tag{4.27}$$

*The right-hand side may be re-written to the more instructive form*

$$\begin{aligned} T_k R_k - R_k T_k &= \hat{Q}_k^H M_k - E_{\text{tot}} \\ &= \hat{Q}_k^H M_k - (E_{\text{loc}} + E_{\text{rec}}), \end{aligned} \tag{4.28}$$

*where the* right total error matrix $E_{\text{tot}}$ *splits into* right local error matrix $E_{\text{loc}}$ *and* right recurrence error matrix $E_{\text{rec}}$. *These error matrices are given by*

$$\begin{aligned} E_{\text{loc}} &= \text{diag}(w_{j+1,j}\gamma_j - w_{j,j-1}\gamma_{j-1}) \\ &+ \text{diag}((w_{j+1,j+1} - w_{jj})\,\gamma_j, 1) \end{aligned} \tag{4.29}$$

$$E_{\text{rec}} = \text{triu}(\hat{Q}_k^H F_k - \hat{F}_k^H Q_k). \tag{4.30}$$

*We remark that we have an implicit connection between the* right local orthogonality *and the* right global orthogonality.

*Similarly we obtain formulae for the lower triangular part, we have that*

$$\begin{aligned} T_k L_k - L_k T_k &= \hat{M}_k^H Q_k - \hat{E}_{\text{tot}} \\ &= \hat{M}_k^H Q_k - (\hat{E}_{\text{loc}} + \hat{E}_{\text{rec}}), \end{aligned} \tag{4.31}$$

*where the* left total error $\hat{E}_{\text{tot}}$ *splits into* left local error $\hat{E}_{\text{loc}}$ *and* left recurrence error $\hat{E}_{\text{rec}}$. *These error matrices are given by explicit expressions similar to the previously defined error matrices.*

**Proof.** In the special case $\hat{C}_k^H = C_k = T_k$ equation (4.22) takes the form

$$T_k(W_k - I_k) - (W_k - I_k)T_k = \hat{Q}_k^H M_k - \hat{M}_k^H Q_k - \hat{Q}_k^H F_k + \hat{F}_k^H Q_k.$$

When we insert the splitting $W_k - I_k = L_k + D_k + R_k$ and consider only the upper triangular part, equation (4.27) follows, since

$$\text{triu}(T_k R_k) = T_k R_k \qquad \text{and} \qquad \text{triu}(R_k T_k) = R_k T_k.$$

A short computation results in the explicit representation of the error matrices. We stress that, for sake of simplicity, in the theorem we have used the additional quantity $\gamma_0 = 0$. □

The theorem proves that the *global* loss of orthogonality depends on the non-orthogonality of the error vectors and the *local* loss of orthogonality, measured by some sort of first discrete derivative of $\gamma_j w_{jj}$ (and $\beta_j w_{jj}$).

The next theorem uses knowledge on the eigenvectors of $T_k$ to gain information on the way *how* orthogonality is lost and in *which* direction the loss occurs, tailored for coupled short-term recurrences.

**Theorem 4.18 (Paige, Bai)** *Let all notations be as in the last theorem. We define the error quantities*

$$\epsilon_{ij} \equiv \hat{s}_i^H E_{\text{tot}} s_j, \qquad \hat{\epsilon}_{ij} \equiv \hat{s}_i^H \hat{E}_{\text{tot}} s_j \tag{4.32}$$

*for some left and right eigenvectors $\hat{s}_i^H$ and $s_j$. In most cases these error quantities will be small.*

*Then loss of orthogonality goes hand in hand with convergence, i.e.*

$$\hat{y}_j^H q_{k+1}\beta_k s_{kj} = \epsilon_{jj}, \qquad \hat{\epsilon}_{jj} = \gamma_k \hat{s}_{kj} \hat{q}_{k+1}^H y_j. \tag{4.33}$$

*Transforming the relations further to*

$$\hat{y}_j^H q_{k+1} = \frac{\epsilon_{jj}}{\beta_k s_{kj}}, \qquad \frac{\hat{\epsilon}_{jj}}{\gamma_k \hat{s}_{kj}} = \hat{q}_{k+1}^H y_j, \tag{4.34}$$

*shows that the loss of orthogonality that occurs when a* left/right *Ritz pair is converging, is mainly in direction of the* right/left *Ritz vector.*

**Proof.** We multiply equation (4.28) from the left by a left *eigenvector* $\hat{s}_i^H$ and from the right by a right *eigenvector* $s_j$. This removes the dependence on the matrix $T_k$ and we obtain the equation

$$(\theta_i - \theta_j)\, \hat{s}_i^H R_k s_j = \hat{y}_i^H q_{k+1} \beta_k s_{kj} - \hat{s}_i^H E_{\mathrm{tot}} s_j.$$

We set $i = j$. Then the left-hand side is zero. By definition of $\epsilon_{jj}$ the remaining equality is the first equation in (4.33).

Similarly, for the other recurrence, by applying eigenvectors of $T_k$ to equation (4.31) we obtain the set of equations

$$(\theta_i - \theta_j)\, \hat{s}_i^H L_k s_j = \gamma_k \hat{s}_{ki} \hat{q}_{k+1}^H y_j - \hat{s}_i^H \hat{E}_{\mathrm{tot}} s_j.$$

We again set $i = j$ and obtain the second equation in (4.33).

The only remarkable thing about the *reordering* of the equations (4.33) to obtain the equations (4.34) is the fact that the denominators can become zero *only* in case of an exact breakdown. □

Theorem 4.18 is one of the main results of Paige's analysis. The setting of the theorem can be extended to take also care of the *principal* vectors. But in finite precision, particularly with regard to the impacts of the theorem just proven, the occurrence of multiple eigenvalues is unlikely. Thus we restrict ourselves to the diagonalisable case.

The error quantities can also be used to express the accuracy of the (unnormalised) Rayleigh quotients. An easy calculation shows that

$$\begin{aligned} \hat{y}_j^H q_{k+1} \beta_k s_{kj} &= \hat{y}_j^H M_k s_j \\ &= \hat{y}_j^H \left( A Q_k - Q_k T_k + F_k \right) s_j \\ &= \hat{y}_j^H \left( A y_j - y_j \theta_j \right) + \hat{y}_j^H F_k s_j \\ &= \hat{y}_j^H A y_j - \hat{y}_j^H y_j \theta_j \;+\; \hat{y}_j^H F_k s_j \end{aligned}$$

holds true. A similar formula holds true for the second recurrence.

**Theorem 4.19 (Paige)** *Let all notations be as above. Despite any loss of orthogonality the* absolute error *of the unnormalised Rayleigh quotients*

$$\hat{y}_j^H A y_j - \hat{y}_j^H y_j \theta_j = \epsilon_{jj} - \hat{y}_j^H F_k s_j$$

*remains small. Because of the* equality *in this equations, this proves that the* relative error *of the Rayleigh quotients*

$$\frac{\hat{y}_j^H A y_j}{\hat{y}_j^H y_j} - \theta_j = \frac{\epsilon_{jj} - \hat{y}_j^H F_k s_j}{\hat{y}_j^H y_j}$$

*will be very large, when the inner product of left and right Ritz vector is very small, and vice versa.*

An expression shedding light on the behaviour of these inner products is the formula that can be obtained by forming $\hat{s}_i^H (4.28) s_j$:

**Theorem 4.20 (Paige)** *Let all notations be as before. The inner product of a left and right Ritz vector weighted by the distance between the corresponding Ritz values is given by*

$$(\theta_i - \theta_j)\, \hat{y}_i^H y_j = \epsilon_{ii} \frac{s_{kj}}{s_{ki}} - \epsilon_{jj} \frac{s_{ki}}{s_{kj}} - \hat{s}_i^H (\hat{Q}_k^H F_k - \hat{F}_k^H Q_k) s_j \tag{4.35}$$

**Proof.** The multiplication of equation (4.28) with eigenvectors from both sides results in

$$\begin{aligned}(\theta_i - \theta_j)\,\hat{y}_i^H y_j &= \hat{y}_i^H q_{k+1}\beta_k s_{kj} - \hat{y}_j^H q_{k+1}\beta_k s_{ki} \\ &- \hat{s}_i^H(\hat{Q}_k^H F_k - \hat{F}_k^H Q_k)s_j.\end{aligned}$$

By Theorem 4.18, equation (4.33), equation (4.35) follows. □

We observe that orthogonality can get lost when several Ritz values form a cluster. In this case the loss of orthogonality will take part in the space spanned by the corresponding Ritz vectors. This suggests that at least one inner product will be large. When the Ritz values are well separated, loss of orthogonality can still occur when $s_{kj}$ and $s_{ki}$ are very different in magnitude. In this case one Ritz pair has a small backward residual estimator and the other one has not converged yet.

We focus on the loss of orthogonality. It turns out to be advantageous to multiply the matrix $\hat{Q}_k^H Q_k$ from the left and the right with the left and right eigenvectors. The deviation of the resulting matrix $\hat{Y}_k^H Y_k$ from the identity matrix splits into

$$\hat{Y}_k^H Y_k - I_k = \hat{S}_k^H(L_k + D_k + R_k)S_k.$$

We define the matrices

$$X_k = \hat{S}_k^H R_k S_k \quad \text{and} \quad \hat{X}_k = \hat{S}_k^H L_k S_k.$$

Neglecting the middle term involving the diagonal $D_k$, the departure from the identity can be measured by the expressions $X_k$ and $\hat{X}_k$. By derivation it is obvious that the analysis for $\hat{X}_k$ will be similar to the analysis for $X_k$. Now, the question is, how can we measure the size or determine the behaviour of $X_k$?

Paige solved this problem with the use of the eigenvector – eigenvalue relations of Chapter 2. He used the results for Hermitian matrices stated in the paper by Thompson and McEnteggert (cf. [TM68]). A simplification for symmetric tridiagonals similar to our simplification of the Hessenberg case is included in Paige's thesis (cf. [Pai71])). Paige's analysis is contained in his 1980 paper (cf. [Pai80]).

The equations in (4.34) can be transformed to matrix form, the first equation is transformed to

$$\hat{Y}_k^H q_{k+1} = \hat{S}_k^H \hat{Q}_k^H q_{k+1} = \begin{pmatrix} \dfrac{\epsilon_{11}}{\beta_k s_{k1}} \\ \vdots \\ \dfrac{\epsilon_{kk}}{\beta_k s_{kk}} \end{pmatrix},$$

or equivalently, to

$$\hat{Q}_k^H q_{k+1} = \frac{1}{\beta_k} S_k \Big( \operatorname{diag}(s_{k1}, \ldots, s_{kk}) \Big)^{-1} \begin{pmatrix} \epsilon_{11} \\ \vdots \\ \epsilon_{kk} \end{pmatrix}.$$

In the following we use upper indices $(l)$ to denote the size of the matrix $T_l$ whose eigenvectors $s_j$ and error quantities $\epsilon_{jj}$ we are interested in. The preceeding repre-

sentation is used to express the columns of $R_k \equiv [r_1^{(k)}, \ldots, r_k^{(k)}]$,

$$
\begin{aligned}
r_l^{(k)} &= \begin{pmatrix} \\ \hat{Q}_{l-1}^H q_l \\ \\ \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} S_{l-1} \begin{pmatrix} \epsilon_{11}^{(l-1)} \Big/ \beta_{l-1} s_{l-1,1}^{(l-1)} \\ \vdots \\ \epsilon_{l-1,l-1}^{(l-1)} \Big/ \beta_{l-1} s_{l-1,l-1}^{(l-1)} \end{pmatrix} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} S_{l-1} & 0 \\ & \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \epsilon_{11}^{(l-1)} \Big/ \beta_{l-1} s_{l-1,1}^{(l-1)} \\ \vdots \\ \epsilon_{l-1,l-1}^{(l-1)} \Big/ \beta_{l-1} s_{l-1,l-1}^{(l-1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.
\end{aligned}
$$

To simplify notation and to grasp the correct dimensions we define some auxiliary matrix notations. We define abbreviations for the augmented matrices

$$
\begin{aligned}
S_{l-1}^{(k)} &\equiv \begin{pmatrix} S_{l-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{K}^{k \times k}, \\
D_{l-1}^{(k)} &\equiv \beta_{l-1} \begin{pmatrix} \mathrm{diag}(e_{l-1}^T S_{l-1}) & 0 \\ 0 & I \end{pmatrix} \in \mathbb{K}^{k \times k},
\end{aligned}
$$

and

$$
g_{l-1}^{(k)} \equiv \begin{pmatrix} \epsilon_{11}^{(l-1)} & \cdots & \epsilon_{l-1,l-1}^{(l-1)} & 0 & \cdots & 0 \end{pmatrix}^T \in \mathbb{K}^k.
$$

With these abbreviations the relation may be stated as

$$
r_l^{(k)} = S_{l-1}^{(k)} \left( D_{l-1}^{(k)} \right)^{-1} g_{l-1}^{(k)}.
$$

We note that the factor $\left( D_{l-1}^{(k)} \right)^{-1}$ can either be interpreted as an amplification factor for the small errors $g_{l-1}^{(k)}$, or as scaling of the eigenvectors $S_k$ to have a last component equal to $\beta_{l-1}^{-1}$. The matrix of eigenvectors scaled to have last element equal to $\beta_{l-1}^{-1}$ will be denoted by

$$
\tilde{S}_{l-1}^{(k)} \equiv S_{l-1}^{(k)} \left( D_{l-1}^{(k)} \right)^{-1},
$$

and the amplified errors will be denoted by

$$
\tilde{g}_{l-1}^{(k)} \equiv \left( D_{l-1}^{(k)} \right)^{-1} g_{l-1}^{(k)}.
$$

We use the Kronecker product formulation of the unknown quantities $X_k$,

$$
\begin{aligned}
X_k &= \hat{S}_k^H R_k S_k \qquad \Rightarrow \\
\mathrm{vec}(X_k) &= (S_k^T \otimes \hat{S}_k^H) \mathrm{vec}(R_k) = (S_k \otimes \check{S}_k)^T \mathrm{vec}(R_k).
\end{aligned}
$$

We define the shorthand notations

$$G_k = \left[g_0^{(k)}, \ldots, g_{k-1}^{(k)}\right] \qquad \tilde{G}_k = \left[\tilde{g}_0^{(k)}, \ldots, \tilde{g}_{k-1}^{(k)}\right].$$

This gives a first expression of the quantities in $X_k$:

**Theorem 4.21** *Let all notations be as before. The* amplification *of the* local *errors, that is the local loss of orthogonality and the recurrence errors is given by*

$$\begin{aligned}
\mathrm{vec}(X_k) &= (S_k \otimes \check{S}_k)^T \begin{pmatrix} S_0^{(k)} & & & \\ & S_1^{(k)} & & \\ & & \ddots & \\ & & & S_{k-1}^{(k)} \end{pmatrix} \mathrm{vec}(\tilde{G}_k) \\
&= (S_k \otimes \check{S}_k)^T \begin{pmatrix} \tilde{S}_0^{(k)} & & & \\ & \tilde{S}_1^{(k)} & & \\ & & \ddots & \\ & & & \tilde{S}_{k-1}^{(k)} \end{pmatrix} \mathrm{vec}(G_k).
\end{aligned}$$

*This expression is solely in terms of left and right* eigenvectors *of the* condensed *matrices $T_l$ of steps one to $k$. The amplification is large in magnitude, when a small residual bound $\beta_l s_{lj}^{(l)}$ has occurred previously.*

One weakness of the theorem above is the dependence on *all* eigenvector components of *all* steps. We are mainly interested in eigen*values*, not in eigen*vectors*. Thus we seek alternate formulations that do not rely on *all* eigenvector components. As first step, note that the second matrix – matrix product in the theorem can be reformulated to take the form

$$(S_k \otimes I)^T \begin{pmatrix} \check{S}_k^T \tilde{S}_0^{(k)} & & & \\ & \check{S}_k^T \tilde{S}_1^{(k)} & & \\ & & \ddots & \\ & & & \check{S}_k^T \tilde{S}_{k-1}^{(k)} \end{pmatrix}.$$

The diagonal of the second matrix consists of block matrices formed of prolonged right eigenvector matrices $S_l^{(k)}$ and left eigenvector matrices $\check{S}_k^T$. By Theorem 2.25 in Chapter 2 it follows that

$$\begin{aligned}
J_k \check{S}_k^T S_l^{(k)} - \check{S}_k^T S_l^{(k)} J_l^{(k)} &= \beta_l \left(\check{S}_k^T e_{l+1}\right) \begin{pmatrix} S_l^T e_l \\ 0 \end{pmatrix}^T \\
&= \beta_l \left(\check{S}_k^T e_{l+1}\right) \left(e_l^T S_l^{(k)}\right) \qquad (4.36)
\end{aligned}$$

holds true. Here the matrix $J_l^{(k)}$ is defined to be the prolonged Jordan matrix

$$J_l^{(k)} \equiv \begin{pmatrix} J_l & 0 \\ 0 & I \end{pmatrix}.$$

The prolongation is such that the resulting matrix is invertible when $T_l$ is invertible. The relation (4.36) can be stated equivalently as

$$\left(I \otimes J_k - J_l^{(k)} \otimes I\right) \mathrm{vec}(\check{S}_k^T S_l^{(k)}) = \mathrm{vec}\left(\beta_l \left(\check{S}_k^T e_{l+1}\right) \left(e_l^T S_l^{(k)}\right)\right).$$

Let

$$J^{k,l} = \left(I \otimes J_k - J_l^{(k)} \otimes I\right)^{-1}$$

denote the inverse of the huge matrix of distances of Ritz values. From the computations thus far it follows that

$$\begin{aligned}
\mathrm{vec}(\check{S}_k^T \tilde{S}_l^{(k)}) &= \left(D_l^{(k)} \otimes I\right)^{-1} \mathrm{vec}(\check{S}_k^T S_l^{(k)}) \\
&= \left(D_l^{(k)} \otimes I\right)^{-1} J^{k,l}\ \mathrm{vec}\left(\beta_l \left(\check{S}_k^T e_{l+1}\right)\left(e_l^T S_l^{(k)}\right)\right) \\
&= \tilde{J}^{k,l}\ \left(D_l^{(k)} \otimes I\right)^{-1} \mathrm{vec}\left(\beta_l \left(\check{S}_k^T e_{l+1}\right)\left(e_l^T S_l^{(k)}\right)\right) \\
&= \tilde{J}^{k,l}\ \mathrm{vec}\left(\left(\check{S}_k^T e_{l+1}\right)\beta_l \left(e_l^T S_l^{(k)}\right)\left(D_l^{(k)}\right)^{-1}\right) \\
&= \tilde{J}^{k,l}\ \mathrm{vec}\left(\left(\check{S}_k^T e_{l+1}\right)\begin{pmatrix} e \\ 0 \end{pmatrix}^T\right) \\
&\equiv \mathrm{vec}\left(\check{S}_{\Theta^{(l)}}^{(k)}\right)
\end{aligned}$$

holds true. The matrix $\tilde{J}^{k,l}$ is defined by the constraint

$$\left(\tilde{J}^{k,l}\right)^{-1}\left(D_l^{(k)} \otimes I\right) = \left(D_l^{(k)} \otimes I\right)\left(J^{k,l}\right)^{-1}$$

and is given by

$$\tilde{J}^{k,l} \equiv \left(I \otimes J_k - (D_l^{(k)})^{-1} J_l^{(k)} D_l^{(k)} \otimes I\right)^{-1}.$$

Everything simplifies, when we only have degenerate Jordan blocks. In this case the diagonal scaling does not alter the matrix $J^{k,l}$, i.e. we have that in this case $\tilde{J}^{k,l} = J^{k,l}$ holds true. Furthermore, in the diagonalisable case the matrix $\check{S}_{\Theta^{(l)}}^{(k)}$ *implicitly* defined in the last line takes the *explicit* form

$$\check{S}_{\Theta^{(l)}}^{(k)} = \Big[\underbrace{(\Theta_k - \theta_1^{(l)})^{-1}\check{S}_k^T e_{l+1}, \ldots, (\Theta_k - \theta_l^{(l)})^{-1}\check{S}_k^T e_{l+1}}_{l \text{ columns}}, \quad \underbrace{0, \ldots, 0}_{k-l \text{ columns}}\Big].$$

Now we are able to state the correlation between the quantities describing local errors and the quantities governing the behaviour of the Ritz vectors in another form:

**Theorem 4.22** *Let all notations be as previously defined. The amplification of the local errors can be expressed alternatively as*

$$\begin{aligned}
vec(X_k) &= (S_k \otimes I)^T \begin{pmatrix} \check{S}_{\Theta^{(0)}}^{(k)} & & & \\ & \check{S}_{\Theta^{(1)}}^{(k)} & & \\ & & \ddots & \\ & & & \check{S}_{\Theta^{(k-1)}}^{(k)} \end{pmatrix} vec(G_k) \\
&= \begin{pmatrix} s_{11}^{(k)}\check{S}_{\Theta^{(0)}}^{(k)} & s_{21}^{(k)}\check{S}_{\Theta^{(1)}}^{(k)} & \cdots & s_{k1}^{(k)}\check{S}_{\Theta^{(k-1)}}^{(k)} \\ s_{12}^{(k)}\check{S}_{\Theta^{(0)}}^{(k)} & s_{22}^{(k)}\check{S}_{\Theta^{(1)}}^{(k)} & \cdots & s_{k2}^{(k)}\check{S}_{\Theta^{(k-1)}}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1k}^{(k)}\check{S}_{\Theta^{(0)}}^{(k)} & s_{2k}^{(k)}\check{S}_{\Theta^{(1)}}^{(k)} & \cdots & s_{kk}^{(k)}\check{S}_{\Theta^{(k-1)}}^{(k)} \end{pmatrix} vec(G_k).
\end{aligned}$$

*In this expression, the entries of the amplifying matrix are composed of* products *of elements of the* eigenvector *matrices* solely *of step k and, at least in the diagonalisable case, the* inverses *of* all distances *of Ritz values prior step k to those of step k.*

We made the observation that *all* elements of the huge matrix in the last theorem are composed of products of left and right eigenvector components. In the diagonalisable case, a way of writing the matrix elements is the notation

$$s_{l+1,j}^{(k)} \check{S}_{\Theta^{(l)}}^{(k)} = \left( \frac{s_{l+1,j}^{(k)} \check{s}_{l+1,i}^{(k)}}{\theta_j^{(k)} - \theta_p^{(l)}} \right)_{\{p,i \in \underline{k}\}}.$$

Here, the index $p$ denotes the *column* (some elements are zero), $l$ the *block–column*, $i$ the *row* and $j$ the *block–row*. The entries of the left hand side are related to those of the right hand side via

$$\check{s}_i^T R_k s_j = \sum_{l=1}^{k-1} s_{l+1,j}^{(k)} \check{s}_{l+1,i}^{(k)} \sum_{p=1}^{l} \frac{\epsilon_{pp}^{(l)}}{\theta_j^{(k)} - \theta_p^{(l)}}.$$

When $j = i$, the elements can be transformed to a representation composed solely of eigenvalues $\nu_p^{(l+1)}$ of tridiagonals that are principal submatrices of $T_k$, since by application of our eigenvalue – eigenvector relations of Chapter 2, Theorem 2.19, page 69,

$$\begin{aligned} s_{l+1,j}^{(k)} \check{s}_{l+1,j}^{(k)} &= \left[ \frac{\chi_{T_{1:l}} \chi_{T_{l+2:k}}}{\chi'_{T_{1:k}}} \right] \left( \theta_j^{(k)} \right) \\ &= \frac{\prod_\ell \left( \theta_j^{(k)} - \nu_\ell^{(l+1)} \right)}{\prod_{\ell \neq j} \left( \theta_j^{(k)} - \theta_\ell^{(k)} \right)} \\ \left\{ \nu_\ell^{(l+1)} \right\}_{\ell \in \underline{k-1}} &\equiv \left\{ \theta_j^{(l+2:k)} \right\}_{j \in \underline{k-l-1}} \cup \left\{ \theta_j^{(l)} \right\}_{j \in \underline{l}} \end{aligned}$$

holds true. Hence,

$$\begin{aligned} \frac{s_{l+1,j}^{(k)} \check{s}_{l+1,j}^{(k)}}{\theta_j^{(k)} - \theta_p^{(l)}} &= \frac{\prod_\ell \left( \theta_j^{(k)} - \nu_\ell^{(l+1)} \right)}{\left( \theta_j^{(k)} - \theta_p^{(l)} \right) \prod_{\ell \neq j} \left( \theta_j^{(k)} - \theta_\ell^{(k)} \right)} \\ &= \frac{\prod_{\ell \neq p_j} \left( \theta_j^{(k)} - \nu_\ell^{(l+1)} \right)}{\left( \theta_j^{(k)} - \theta_{p_j}^{(k)} \right) \prod_{\ell \neq j, p_j} \left( \theta_j^{(k)} - \theta_\ell^{(k)} \right)}. \end{aligned}$$

In element-wise notation this takes the form

$$\begin{aligned} \check{s}_j^T R_k s_j &= \sum_{l=1}^{k-1} s_{l+1,j}^{(k)} \check{s}_{l+1,j}^{(k)} \sum_{p=1}^{l} \frac{\epsilon_{pp}^{(l)}}{\theta_j^{(k)} - \theta_p^{(l)}} \\ &= \sum_{l=1}^{k-1} \sum_{p=1}^{l} \frac{\epsilon_{pp}^{(l)}}{\theta_j^{(k)} - \theta_{p_j}^{(k)}} \frac{\prod_{\ell \neq p_j} \left( \theta_j^{(k)} - \nu_\ell^{(l+1)} \right)}{\prod_{\ell \neq j, p_j} \left( \theta_j^{(k)} - \theta_\ell^{(k)} \right)}. \end{aligned}$$

From these expressions we can conclude that orthogonality is lost, when one of three equivalent conditions emerge. There is some small residual bound, a Ritz value of step $l$, $l < k$ is close to an Ritz value of step $k$, and, the third condition, there do exist two close Ritz values. We remark that it is possible to expand the analysis to some extent to deal with the case of *clusters* of Ritz values.

We switch back to the general case and consider the third error analysis approach. It relies on a backward error analysis and is based on a multiplicative splitting of the matrix $W_k = \hat{Q}_k^H Q_k$. This approach enables us to derive bounds,

but only applies *as long as the basis vectors remain linear independent.* The bounds are based on computed quantities and relate the computed pencil to a perturbation of an exact pencil. This exact pencil in general looses the simple Hessenberg structure. This approach is based on Simon's thesis, Theorem 2.3, page 40, Theorem 2.4, page 42 and Day's thesis, Theorem 7, page 64 (cf. [Sim82, Day93]).

We first give a version of the main theorem using an LU-type decomposition:

**Theorem 4.23 (Simon, Day)** *Let $A \in \mathbb{K}^{n\times n}$, $C_k \in \mathbb{K}^{k\times k}$ and $M_k \in \mathbb{K}^{n\times k}$, where $M_k = c_{k+1,k} q_{k+1} e_k^T$, be given. Suppose that*

$$AQ_k - Q_k C_k = M_k - F_k$$

*holds true for some $Q_k, F_k \in \mathbb{K}^{n\times k}$. Suppose that $W_{k+1} = \hat{Q}_{k+1}^H Q_{k+1}$ (for some matrix $\hat{Q}_{k+1} \in \mathbb{K}^{n\times(k+1)}$) can be decomposed without pivoting, $W_{k+1} = \hat{R}_{k+1}^H R_{k+1}$. Thus, obviously $W_k = \hat{R}_k^H R_k$. Denote by $C_k^{\mathrm{sim}}$ the matrix that is* triangular similar *to the computed $C_k$,*

$$C_k^{\mathrm{sim}} \equiv R_k C_k R_k^{-1}.$$

*Observe that the triangular similarity assures that the matrix $C_k^{\mathrm{sim}}$ is Hessenberg. Define the* exact oblique projection

$$(C_k^{\mathrm{exact}}, I_k) \equiv (\hat{P}_k^H A P_k, \hat{P}_k^H P_k) \equiv (\hat{R}_k^{-H} \hat{Q}_k^H A Q_k R_k^{-1}, \hat{R}_k^{-H} W_k R_k^{-1})$$

*of $A$. Then $C_k^{\mathrm{sim}}$ may be interpreted as a* perturbation of $C_k^{\mathrm{exact}}$,

$$\begin{aligned} C_k^{\mathrm{exact}} - C_k^{\mathrm{sim}} &= c_{k+1,k} \hat{R}_k^{-H} \hat{Q}_k^H q_{k+1} e_k^T R_k^{-1} - \hat{R}_k^{-H} \hat{Q}_k^H F_k R_k^{-1} \\ &= \frac{c_{k+1,k}}{r_{kk}} \begin{pmatrix} r_{1,k+1} \\ \vdots \\ r_{k,k+1} \end{pmatrix} e_k^T - \hat{P}_k^H F_k R_k^{-1}. \end{aligned} \tag{4.37}$$

*The perturbation can be bounded normwise by*

$$\|C_k^{\mathrm{exact}} - C_k^{\mathrm{sim}}\|_2 \le \|\hat{R}_k^{-1}\|_2 \|R_k^{-1}\|_2 (\|c_{k+1,k} \hat{Q}_k^H q_{k+1}\|_2 - \|\hat{Q}_k^H F_k\|_2).$$

**Proof.** The implicitly in the theorem defined projection matrices

$$P_k = Q_k R_k^{-1}, \qquad \hat{P}_k = \hat{Q}_k \hat{R}_k^{-1}$$

are bi-orthogonal, since $\hat{P}_k^H P_k = \hat{R}_k^{-H} \hat{Q}_k^H Q_k R_k^{-1} = I_k$. Thus, the oblique projection of $(A, I)$ onto the pencil

$$(A, I) \to (\hat{P}_k^H A P_k, \hat{P}_k^H P_k) \equiv (C_k^{\mathrm{exact}}, I_k)$$

is *exact*. As a next step, we transform the pre-multiplied equation

$$\hat{Q}_k^H A Q_k - \hat{Q}_k^H Q_k C_k = \hat{Q}_k^H M_k - \hat{Q}_k^H F_k, \tag{4.38}$$

by inserting the LU decomposition of $W_k = \hat{Q}_k^H Q_k = \hat{R}_k^H R_k$. When we multiply equation (4.38) by the inverses of the triangular factors form the left and the right, we obtain

$$\begin{aligned} \hat{P}_k^H A P_k - R_k C_k R_k^{-1} &= \hat{P}_k^H M_k R_k^{-1} - \hat{P}_k^H F_k R_k^{-1} \\ &= c_{k+1,k} \hat{P}_k^H P_{k+1} R_{k+1} e_{k+1} r_{kk}^{-1} - \hat{P}_k^H F_k R_k^{-1}. \end{aligned}$$

Equation (4.37) follows, since the matrices $\hat{P}_{k+1}$ and $P_{k+1}$, defined in analogue to $\hat{P}_k$ and $P_k$, are bi-orthogonal. The bound follows upon application of norms. □

This theorem may be interpreted as finding the closest Hessenberg decomposition to the computed one, such that the eigenvalues are good approximations, even if this implies that the eigenvectors are bad. It is an indicator that this will be the case whenever the loss of orthogonality has reached some limit.

The ingredients of the bound are of particular interest. The part depending on the norms of the inverses of the triangular factors, $\|\hat{R}_k^{-1}\|_2\|R_k^{-1}\|_2$, is comparable to the growth factor in the LU decomposition. When the matrix $\hat{Q}_k^H Q_k$ is Hermitian, we can bound this part by a constant. The remaining part measures the new loss of orthogonality corresponding to the last step times the next off-diagonal element and the accuracy of the recurrence. This implies that when the loss is moderate, the computed eigenvalues are as accurate as one can hope for.

Some comments on the behaviour of the *eigenvectors* are in order. The triangular similarity transformation maps the computed eigenvectors onto the improved eigenvectors

$$s_j^{\text{sim}} \equiv R_k s_j.$$

The loss of orthogonality $W_k$ as well as the size of the elements of the triangular factors is usually *gradual*. It is gradual in the sense of having small elements close to the main diagonal and large elements further apart. The eigenvectors corresponding to *well-converged* Ritz values do not change drastically, since the sizes of the elements are also graded. The eigenvectors corresponding to *unconverged* Ritz values are changed. The main change is that the first component is damped, as if the finite precision run corresponds to an exact one with a different starting vector with a small component in direction of the corresponding eigenvector. This leads to a *delay of the convergence* of the unconverged Ritz values. This, in turn, matches the behaviour typically observed in finite precision when using a method based on *short-term recurrences*.

Roughly speaking, we might say that the triangular similarity transformation 'twists' the *tridiagonal* matrix of a short-term recurrence like the Lanczos process to a *Hessenberg* matrix. The finite precision run produces tridiagonal matrices. Adding a grain of salt, we can think of the twist moving along the anti-diagonal to the other side producing copies of the converging Ritz values. This is not predicted by the theorem, since at that point the linear dependence among the basis vectors is lost completely.

We might use *any* matrix $\hat{Q}_{k+1}$ and *any* triangular decomposition for this approach. As example we might use the LDMT decomposition. The assumption that the decomposition exists is always the drawback of the multiplicative splitting approach.

**Corollary 4.24 (Day)** *Let $A \in \mathbb{K}^{n\times n}$, $T_k \in \mathbb{K}^{k\times k}$, $\Omega_k \in \mathbb{K}^{k\times k}$ and $M_k \in \mathbb{K}^{n\times k}$, where $M_k = \beta_k q_{k+1} e_k^T$, be given. Suppose that*

$$AQ_k - Q_k\Omega_k^{-1}T_k = M_k - F_k$$

*holds true for some $Q_k, F_k \in \mathbb{K}^{n\times k}$. Assume that $W_{k+1} = \hat{Q}_{k+1}^H Q_{k+1}$ (for some matrix $\hat{Q}_{k+1} \in \mathbb{K}^{n\times(k+1)}$) can be decomposed without pivoting, $W_{k+1} = \hat{L}_{k+1}\tilde{\Omega}_{k+1}R_{k+1}$. Thus, obviously $W_k = \hat{L}_k\tilde{\Omega}_k R_k$. Denote by $(C_k^{\text{sim}}, \tilde{\Omega}_k)$ the pencil that is* triangular similar *to the computed pencil $(T_k, \Omega_k)$,*

$$(C_k^{\text{sim}}, \tilde{\Omega}_k) \equiv (\tilde{\Omega}_k R_k \Omega_k^{-1} T_k R_k^{-1}, \tilde{\Omega}_k).$$

*Observe that the triangular similarity assures that the matrix $C_k^{\text{sim}}$ is Hessenberg. Define the* exact oblique projection

$$(C_k^{\text{exact}}, \tilde{\Omega}_k) \equiv (\hat{P}_k^H A P_k, \hat{P}_k^H P_k) \equiv (\hat{L}_k^{-1}\hat{Q}_k^H A Q_k R_k^{-1}, \hat{L}_k^{-1} W_k R_k^{-1})$$

*of the* pencil $(A, I)$. *Then the* pencil $(C_k^{\rm sim}, \tilde{\Omega}_k)$ *may be interpreted as a* perturbation *of the* pencil $(C_k^{\rm exact}, \tilde{\Omega}_k)$,

$$\begin{aligned} C_k^{\rm exact} - C_k^{\rm sim} &= \beta_k \hat{L}_k^{-1} \hat{Q}_k^H q_{k+1} e_k^T R_k^{-1} - \hat{L}_k^{-1} \hat{Q}_k^H F_k R_k^{-1} \\ &= \frac{\beta_k}{r_{kk}} \tilde{\Omega}_k \begin{pmatrix} r_{1,k+1} \\ \vdots \\ r_{k,k+1} \end{pmatrix} e_k^T - \hat{P}_k^H F_k R_k^{-1}. \end{aligned} \tag{4.39}$$

*The perturbation can be bounded normwise by*

$$\|C_k^{\rm exact} - C_k^{\rm sim}\|_2 \leq \|\hat{L}_k^{-1}\|_2 \|R_k^{-1}\|_2 (\|\beta_k \hat{Q}_k^H q_{k+1}\|_2 - \|\hat{Q}_k^H F_k\|_2).$$

**Proof.** That the pencil $(C_k^{\rm sim}, \tilde{\Omega}_k)$ is similar to the pencil $(T_k, \Omega_k)$ follows by a straight calculation,

$$\begin{aligned} (T_k, \Omega_k) &\Leftrightarrow (\Omega_k^{-1} T_k, I_k) \\ &\Leftrightarrow (R_k \Omega_k^{-1} T_k R_k^{-1}, I_k) \\ &\Leftrightarrow (\tilde{\Omega}_k R_k \Omega_k^{-1} T_k R_k^{-1}, \tilde{\Omega}_k). \end{aligned}$$

The proof for the remaining part of the corollary is along the lines of the proof for Theorem 4.23. The only difference is the occurrence of the diagonal scaling matrix $\tilde{\Omega}_k$. □

The corollary is tailored to work for Day's variant of the Lanczos method that projects the pencil $(A, I)$ to the pencil $(T_k, \Omega_k)$. This corollary is the key result of Day's thesis and serves as basis for Day's semiduality algorithm.

Both, the theorem and the corollary, rest upon strong assumptions in the sense that they require $W_k$ to be decomposable *without pivoting*, i.e. such that all leading principal minors are non-zero. This can be weakened by applying GEPP or even GECP. We state the result based on GECP as a second corollary:

**Corollary 4.25** *Let $A$, $C_k$, $M_k$, $Q_k$ and $F_k$ be given as in Theorem 4.23. Suppose that $W_k = \hat{Q}_k^H Q_k$ (for some matrix $\hat{Q}_k \in \mathbb{K}^{n\times k}$) can be decomposed with GECP, $W_k = O_1 \hat{L}_k R_k O_2$. Denote by $C_k^{\rm sim}$ the matrix that is* similar *to the computed $C_k$,*

$$C_k^{\rm sim} \equiv R_k O_2 C_k O_2 R_k^{-1}.$$

*Observe that when $O_2 \neq I$ the matrix $C_k^{\rm sim}$ will, in general, no longer be a Hessenberg matrix. Define the* exact oblique projection

$$(C_k^{\rm exact}, I_k) \equiv (\hat{P}_k^H A P_k, \hat{P}_k^H P_k) \equiv (\hat{L}_k^{-1} O_1 \hat{Q}_k^H A Q_k O_2 R_k^{-1}, \hat{L}_k^{-1} O_1 W_k O_2 R_k^{-1})$$

*of $A$. Then $C_k^{\rm sim}$ may be interpreted as a* perturbation of $C_k^{\rm exact}$,

$$C_k^{\rm exact} - C_k^{\rm sim} = c_{k+1,k} \hat{L}_k^{-1} O_1 \hat{Q}_k^H q_{k+1} e_k^T O_2 R_k^{-1} - \hat{L}_k^{-1} O_1 \hat{Q}_k^H F_k O_2 R_k^{-1}$$

*The perturbation can be bounded normwise by*

$$\|C_k^{\rm exact} - C_k^{\rm sim}\|_2 \leq \|\hat{L}_k^{-1}\|_2 \|R_k^{-1}\|_2 (\|c_{k+1,k} \hat{Q}_k^H q_{k+1}\|_2 - \|\hat{Q}_k^H F_k\|_2).$$

**Proof.** The proof is along the lines of the proof of Theorem 4.23. The formulation expressing the first perturbation term in elements of the triangular factor $R_{k+1}$ in general will not be possible, since the pivoting strategy might destroy the structure between two successive matrices $\hat{P}_k^H \equiv \hat{L}_k^{-1} O_1 \hat{Q}_k^H$ and $\hat{P}_{k+1}^H$. □

To be applicable, the result of Theorem 4.23 has to be tailored to fit the characteristics of the method to be used. These results have to be used with caution, since there is a natural trade-off between a *strong* theorem and its *domain of applicability*. The stronger the bound becomes, the more knowledge on the loss of orthogonality has to be inserted to be of practical use. In case of an orthogonal basis, i.e. in case of the Arnoldi method, including as a special case the symmetric Lanczos method, $W_k = Q_k^H Q_k$ is HPD. Instead of the LU decomposition the Cholesky decomposition should be preferred. In this case the norm of $R_k^{-1}$ in the bounds is less equal 2, which simplifies the analysis substantially.

This type of backward error analysis requires some justification. First of all, they do not result in backward error formula, since they express the occurring loss of orthogonality and the accuracy of the returned approximations in terms of *computed* quantities. That is, simply using the results obtained, we have no clue why some of the methods converge beyond the point where the assumptions of the theorems are violated. Then, we can not predict by theory if they are good in describing the actual behaviour. Even if numerical experiments suggest so, it would be nice to actually prove that a backward error analysis can not apply and that the results of this section are in some sense optimal.

For this reason, we first try to apply the standard backward error analysis to the Sylvester equation. This is done in the next section and should clarify the role that a standard error analysis can take. Then, in the proximate section, the recurrence of the basis vectors is analysed in more detail. This analysis results in statements that reveal the restrictions of all possible error analysis approaches.

## 4.7 The Sylvester Equation

In this section we interprete the governing equation

$$AQ_k - Q_k C_k = M_k - F_k \tag{4.40}$$

as a linear equation in $Q_k$. This linear equation, i.e. the *Sylvester equation*, is an ordinary matrix equation in higher dimension $nk$. We re-write the governing equation using the Kronecker product and the vec operator,

$$(I_k \otimes A - C_k^T \otimes I_n)\mathrm{vec}(Q_k) = \mathrm{vec}(M_k - F_k).$$

The system matrix is known as the *Kronecker sum* of the matrices $A$ and $-C_k^T$.

We want to apply stability analysis to the huge matrix of the Sylvester equation. We are interested in the Schur and Jordan normal forms. The Schur form can easily be obtained. Let the Schur forms of $A$ and $C_k^T$ be given as

$$AU = UR \qquad \text{and} \qquad C_k^T W_k = W_k R_k.$$

Then the matrix of the Sylvester equation has Schur form

$$\begin{aligned}(I_k \otimes A - C^T \otimes I_n)(W_k \otimes U) &= W_k \otimes AU - C_k^T W_k \otimes U \\ &= W_k \otimes UR - W_k R_k \otimes U \\ &= (W_k \otimes U)(I_k \otimes R - R_k \otimes I_n).\end{aligned}$$

The matrix $I_k \otimes R - R_k \otimes I_n$ is upper triangular by inspection. The matrix $W_k \otimes U$ is unitary, since

$$\begin{aligned}(W_k \otimes U)(W_k \otimes U)^H &= (W_k \otimes U)(W_k^H \otimes U^H) \\ &= W_k W_k^H \otimes U U^H \\ &= I_k \otimes I_n = I_{nk}.\end{aligned}$$

Thus we have computed the Schur decomposition. This already shows that the eigenvalues are given by the numbers $\lambda_i - \theta_j$, $i \in \underline{n}, j \in \underline{k}$.

The Schur vectors are the columns of the matrix $W_k \otimes U$,

$$W_k \otimes U = \big[w_1 \otimes u_1, w_1 \otimes u_2, \ldots, w_k \otimes u_n\big].$$

We observe that the right Schur vectors of the large matrix compromise of the *right* Schur vectors of $A$ and some sort of *left* Schur vectors of $C_k$.

Now we focus on the Jordan normal form of the matrix of the Sylvester equation. Suppose that the Jordan normal forms of $A$ and $C_k^T$ are given by

$$AV = VJ_\Lambda \qquad \text{and} \qquad C_k^T Z_k = Z_k J_\Theta.$$

We use the Jordan normal form of $C_k^T$, the transpose of $C_k$. The Jordan normal form of the transpose can be obtained from the Jordan normal form of the matrix $C_k$,

$$\hat{S}_k^H C_k = J_\Theta \hat{S}_k^H \quad \Leftrightarrow \quad C_k^T \check{S}_k = \check{S}_k J_\Theta^T.$$

Every Jordan block is similar to its transpose via

$$\begin{pmatrix} \theta & & & \\ 1 & \theta & & \\ & \ddots & \ddots & \\ & & 1 & \theta \end{pmatrix} = J \begin{pmatrix} \theta & 1 & & \\ & \theta & \ddots & \\ & & \ddots & 1 \\ & & & \theta \end{pmatrix} J^{-1},$$

where $J$ is the flip-matrix

$$J = J^{-1} = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix}.$$

The matrix $Z_k$ of the eigendecomposition of $C_k^T$ is obtained from the matrix $\check{S}_k$ by a simple re-ordering of the column vectors. We use this strange re-ordering because it simplifies notation.

We go on to proceed like in case of the Schur form,

$$\begin{aligned} (I_k \otimes A - C_k^T \otimes I_n)(Z \otimes V) &= Z_k \otimes AV - C_k^T Z_k \otimes V \\ &= Z_k \otimes VJ_\Lambda - Z_k J_\Theta \otimes V \\ &= (Z_k \otimes V)(I_k \otimes J_\Lambda - J_\Theta \otimes I_n). \end{aligned}$$

The second huge matrix in the product of the right-hand side,

$$I_k \otimes J_\Lambda - J_\Theta \otimes I_n, \tag{4.41}$$

in general is *no* Jordan matrix. Nevertheless, the eigenvectors of the Kronecker sum must to be constructible using the columns of $\tilde{V} \equiv Z_k \otimes V$, i.e. the vectors

$$\tilde{v}_{ij} \equiv z_j \otimes v_i = \operatorname{vec}\left(v_i z_j^T\right)$$

as trial vectors. We distinguish the four cases that a simple (non-simple) block meets a simple (non-simple) block and state the results as a series of three examples, one lemma and two conjectures.

**Example 4.26 (simple block meets simple block)** Let $A \in \mathbb{K}^{n\times n}$. Let $C_k^T \in \mathbb{K}^{k\times k}$. Let $\lambda_i$ and $v_i$ be an eigenpair corresponding to a one by one Jordan block

of $A$. Let $\theta_j$ and $z_j$ be an eigenpair corresponding to a one by one Jordan block of $C_k^T$. Then the *trial vector* $\tilde{v}_{ij}$ and $\lambda_{ij}$ defined by

$$\tilde{v}_{ij} \equiv z_j \otimes v_i = \operatorname{vec}\left(v_i z_j^T\right), \qquad \lambda_{ij} \equiv \lambda_i - \theta_j,$$

is an *eigenpair* of the Kronecker sum of $A$ and $-C_k^T$, since

$$\begin{aligned}\left(I_k \otimes A - C_k^T \otimes I_n\right) \operatorname{vec}\left(v_i z_j^T\right) &= \operatorname{vec}\left(A v_i z_j^T - v_i z_j^T C_k\right) \\ &= \left(\lambda_i - \theta_j\right) \operatorname{vec}\left(v_i z_j^T\right).\end{aligned}$$

This first example is covered by the following two examples. The first example is important when $A$ and $C_k$ are both diagonalisable, which is the generic case when considering matrices that are subject to small perturbations.

**Example 4.27 (non-simple block meets simple block)** Let $A$ and $C_k^T$ be given as in the preceeding example. Let $J_i$ be an $\ell \times \ell$ Jordan block of $A$,

$$J_i = J_{\lambda_i} = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}.$$

Suppose the corresponding subspace is spanned by

$$V_i \equiv \left[v_i, \ldots v_{i+\ell-1}\right], \qquad AV_i = V_i J_i.$$

Let $\theta_j$ and $z_j$ be an eigenpair corresponding to a one by one Jordan block of $C_k^T$. Then the *block trial vector*

$$z_j \otimes V_i \equiv \left[z_j \otimes v_i, \ldots, z_j \otimes v_{i+\ell-1}\right],$$

compromising of the trial vectors

$$z_j \otimes v_{i+p-1} = \operatorname{vec}\left(v_{i+p-1} z_j^T\right), \quad p \in \underline{\ell}$$

as columns spans an invariant subspace of the Kronecker sum of $A$ and $-C_k^T$ to the eigenvalue $\lambda_{ij} \equiv \lambda_i - \theta_j$, since

$$\begin{aligned}\left(I_k \otimes A - C_k^T \otimes I_n\right) z_j \otimes V_i &= z_j \otimes AV_i - C_k^T z_j \otimes V_i \\ &= z_j \otimes V_i J_i - z_j \theta_j \otimes V_i \\ &= (z_j \otimes V_i)(J_i - \theta_j I_\ell).\end{aligned}$$

The columns of the block vector, i.e. the trial vectors, are the eigenvector of the Jordan block and the corresponding principal vectors, respectively.

For the next example it is useful to observe that the negative of a Jordan block has the Jordan decomposition

$$-J_\lambda = S J_{-\lambda} S$$

where $S$ is the sign matrix with diagonal elements alternating between plus and minus ones that already appeared in Chapter 2.

**Example 4.28 (simple block meets non-simple block)** Let $A$ and $C_k^T$ be given as in the preceeding examples. Let $\lambda_i$ and $v_i$ be an eigenpair corresponding to a one by one Jordan block of $A$. Let $J_j$ be an $\ell \times \ell$ Jordan block of $C_k^T$,

$$J_j = J_{\theta_j} = \begin{pmatrix} \theta_j & 1 & & \\ & \theta_j & \ddots & \\ & & \ddots & 1 \\ & & & \theta_j \end{pmatrix}.$$

Then the sign-changed block trial vector

$$(Z_j \otimes v_i)S = Z_j S \otimes v_i \equiv \left[z_j \otimes v_i, -z_{j+1} \otimes v_i, \ldots, \pm z_{j+\ell-1} \otimes v_i\right],$$

compromising of the sign-changed trial vectors

$$(-1)^{p-1} z_{j+p-1} \otimes v_i = (-1)^{p-1} \mathrm{vec}\left(v_i z_{j+p-1}^T\right), \quad p \in \underline{\ell}$$

spans an invariant subspace of the Kronecker sum of $A$ and $-C_k^T$ to the eigenvalue $\lambda_{ij} \equiv \lambda_i - \theta_j$, since

$$\begin{aligned} \left(I_k \otimes A - C_k^T \otimes I_n\right) Z_j S \otimes v_i &= Z_j S \otimes A v_i - C_k^T Z_j S \otimes v_i \\ &= Z_j S \otimes v_i \lambda_i - Z_j J_j S \otimes v_i \\ &= (Z_j S \otimes v_i)(\lambda_i I_\ell + J_{-\theta_j}). \end{aligned}$$

The columns of the block vector, i.e. the sign-changed trial vectors, are the eigenvector of the Jordan block and the corresponding principal vectors, respectively.

The next example is a little bit more involved and covers the previous three. We split our approach into three smaller steps. In a first step, we construct in a first lemma a set of *eigenvectors*. Then, in the second step, we go on to construct in a second lemma a set of *principal vectors* that correspond to the eigenvectors already constructed. In the last step, we count the vectors constructed to make sure we have found the entire Jordan decomposition.

For ease of understanding we numerate the vectors simply by single indices. This corresponds to $A$ and $C_k^T$ having only one single Jordan block. The sizes of the blocks are denoted as before by $\ell_A$ and $\ell_C$, the subscript denoting the matrix the index originates from. Since the trial vectors form a basis of the space, the eigenvectors and the principal vectors have be representable by linear combinations of them.

The eigenvectors in this representation can be found almost immediately:

**Lemma 4.29** *A set of* $\min(\ell_A, \ell_C)$ *unnormalised eigenvectors is given by*

$$\sum_{p=1}^{l} z_{l-p+1} \otimes v_p, \quad l \in \{1, \ldots, \min(\ell_A, \ell_C)\}.$$

**Proof.** These vectors are eigenvectors. This follows by straight calculation:

$$\begin{aligned} \left(I_k \otimes A - C_k^T \otimes I_n\right) \sum_{p=1}^{l} z_{l-p+1} \otimes v_p &= \sum_{p=1}^{l} z_{l-p+1} \otimes A v_p \\ &- \sum_{p=1}^{l} C_k^T z_{l-p+1} \otimes v_p \end{aligned}$$

$$
\begin{aligned}
&= (\lambda_i - \theta_j) \sum_{p=1}^{l} z_{l-p+1} \otimes v_p \\
&+ \sum_{p=1}^{l} z_{l-p+1} \otimes v_{p-1} \\
&- \sum_{p=1}^{l} z_{l-p} \otimes v_p \\
&= (\lambda_i - \theta_j) \sum_{p=1}^{l} z_{l-p+1} \otimes v_p.
\end{aligned}
$$

In the above, indices which are zero refer to the zero vector. By construction the number of eigenvectors of this type is $\min(\ell_A, \ell_C)$. $\square$

We aim at constructing principal vectors of highest degree. Our ansatz is based on linear combinations of trial vectors. To derive the explicit formula for the principal vectors, we use a slightly different point of view, which is best introduced using a small picture. We think of the trial vectors as the *nodes of a grid*, as is shown in figure 4.2.

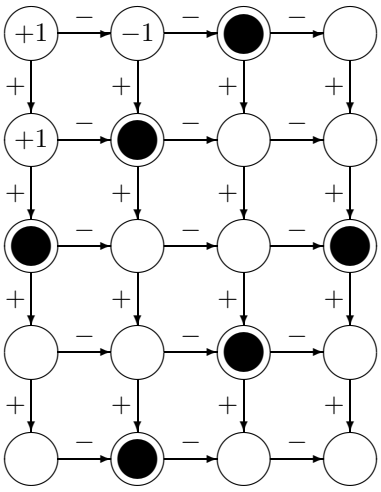


Figure 4.2: The two-dimensional generalised eigenvector grid

Figure 4.2 is a small example, the Jordan block of $A$ is of size five and the Jordan block of $C_k$ is of size four. Every grid point corresponds to one of the trial vectors, sorted by degree. The grid point in the *upper left corner* for instance corresponds to the trial vector composed of the principal vectors of highest degree, i.e. of degree five and four, respectively. The plus and minus signs indicate how a single element is transported through the grid.

The *anti-diagonals* of the grid deserve attention, since the constructed *eigenvectors* are composed merely of trial vectors corresponding to diagonal elements. We observe that when we form any linear combination of trial vectors whose nodes are along one anti-diagonal, application of the Kronecker sum maps the anti-diagonal to the next anti-diagonal, the new coefficient vector is the discrete derivative of the old coefficient vector.

We think of the grid prolonged to all sides and define the value for entries outside the grid range to be zero. We interprete the coefficient vectors along the anti-diagonals as arithmetic functions $f : \mathbb{N} \mapsto \mathbb{N}, \quad f : n \to f(n)$. By inspection we see that the next anti-diagonal is computed as $\Delta f\,(n) = f\,(n+1) - f\,(n)$, i.e. we compute the discrete derivative of the function $f$. Our requirement that the last values in the range $\underline{k}$ are equal reads as

$$\begin{aligned} \Delta^l f\,(i) &= \Delta\left(\Delta^{l-1} f\right)(i) \\ &= \sum_{j=0}^{l} (-1)^{l-j} \binom{l}{j} f\,(i+j) \\ &= \text{const } \forall\ i \in \underline{k}. \end{aligned}$$

When we look at a single nonzero element somewhere in the grid, we see that it is propagated in terms of binomial coefficients. We define the binomial coefficients as usual to be

$$\binom{r}{k} = \begin{cases} r(r-1)\ldots(r-k+1)\Big/k! & k \geq 0 \\ 0 & k < 0 \end{cases} \qquad \forall\ r \in \mathbb{C},\ k \in \mathbb{Z}.$$

As a natural consequence of the observation of the binomial coefficients we try to find a starting vector composed of binomial coefficients. This approach seems to work, when the $k$ non-zero elements of the starting vector along one anti-diagonal are given by the Hadamard (Schur) product

$$f = f_{\text{left}} \circ f_{\text{right}}$$

of the two length $k$ vectors

$$f_{\text{left}} = \left(\binom{n-k}{0}, \ldots, \binom{n-k+i-1}{i-1}, \ldots, \binom{n-1}{k-1}\right)^T$$

and

$$f_{\text{right}} = \left(\binom{m-1}{k-1}, \ldots, \binom{m-i}{k-i}, \ldots, \binom{m-k}{0}\right)^T.$$

The $i$th non-zero component, $i \in \underline{k}$ is given by the product of two binomial coefficients

$$f(i) = \binom{n-k+i-1}{i-1} \cdot \binom{m-i}{k-i}.$$

This arithmetic function is defined for *all values*, because outside the range one of the vectors, the left or right, is zero *by definition* of the binomial coefficients. To use the discrete derivative we pad the vector with $n-k$ zeros on the left and $m-k$ zeros on the right, such that the length of the vector now is $m+n-k$.

**Conjecture 4.30 (non-simple block meets non-simple block)** The principal vectors of degree $s$ are given by the discrete derivatives

$$\begin{aligned} \Delta^s f\,(i) &= \sum_{j=0}^{s} (-1)^{s-j} \binom{s}{j} f\,(i+j) \\ &= \sum_{j=0}^{s} (-1)^{s-j} \binom{s}{j} \binom{n-k+i+j-1}{i+j-1} \binom{m-i-j}{k-i-j}, \end{aligned}$$

where $s$ runs from zero up to $n+m-(2k-1) = n+m-2k+1$. The last vector constructed this manner has constant coefficients along the anti-diagonal in the range of interest, i.e. is one of the eigenvectors previously constructed.

The proof might be based on the $(n+m-2k+2)$th derivative. When the resulting coefficients in the range of interest are zero, our conjecture has been proven to be correct. In this case we have exactly $\min(\ell_A, \ell_C)$ Jordan blocks of sizes

$$(\ell_A + \ell_C) - 2k + 1, \text{ where } 1 \leq k \leq \min(\ell_A, \ell_C).$$

Thus, we would have constructed a *complete Jordan decomposition* of the Kronecker sum, since

$$\begin{aligned} \ell_A \cdot \ell_C &= \sum_{k=1}^{\min(\ell_A,\ell_C)} (\ell_A + \ell_C) - 2k + 1 \\ &= \min(\ell_A, \ell_C) \cdot (\ell_A + \ell_C) - \min(\ell_A, \ell_C)^2 \end{aligned}$$

holds true.

**Conjecture 4.31 (Kronecker sum: Jordan normal form)** A Jordan normal form of the Kronecker sum of $A \in \mathbb{K}^{n\times n}$ and $-C_k^T \in \mathbb{K}^{k\times k}$,

$$I_k \otimes A - C_k^T \otimes I_n,$$

can be constructed from the Jordan normal forms of $A$ and $C_k^T$,

$$AV = VJ_\Lambda \qquad \text{and} \qquad C_k^T Z = ZJ_\Theta.$$

The right eigenvector matrix is given implicitly by the vectors constructed in examples (4.26)–(4.28), Lemma 4.29 and Conjecture 4.30. In complete analogy, also a *left* eigenvector matrix can be constructed by interchanging the roles of $A$ and $C_k$.

**Proof.** This theorem is unproven only because of the lacking proof for Conjecture 4.30. □

We have proven that the eigenvalues of the Kronecker sum of $A$ and $C_k^T$ are given by $\lambda_i - \theta_j$. When $A$ and $C_k$ are both normal, also the Kronecker sum is normal and the condition number is given by

$$\kappa = \frac{\max_{i,j} |\lambda_i - \theta_j|}{\min_{i,j} |\lambda_i - \theta_j|}.$$

The matrix $C_k$ is used as an approximation to the matrix $A$, and so in case of convergence the condition number of the Kronecker sum tends to infinity. Even when $A$ and $C_k^T$ are non-normal, the Kronecker sum will become singular whenever there are two indices $i_0 \in \underline{n}, j_0 \in \underline{m}$ such that $\lambda_{i_0} - \theta_{j_0} = 0$.

Higham considered backward error analysis of the Sylvester equation (cf. [Hig93]). He gives bounds for the backward error of an approximate solution of the Sylvester equation. He also gives a new condition number that supersedes the obvious choice of a condition number based on the interpretation as a linear system. The interesting thing about the analysis is the dependence of the backward error and the condition number on the (approximate) solution $Q_k$. The backward error of the Sylvester equation $AQ - QC = M$, where $A \in \mathbb{K}^{n\times n}$, $C \in \mathbb{K}^{m\times m}$, $Q, M \in \mathbb{K}^{n\times m}$ and $m < n$, is defined as

$$\begin{aligned} \eta(\tilde{Q}) \equiv \min\{\epsilon : &(A + \Delta A)\tilde{Q} - \tilde{Q}(C + \Delta C) = M + \Delta M, \\ &\|\Delta A\|_F \leq \epsilon\alpha, \|\Delta C\|_F \leq \epsilon\gamma, \|\Delta M\|_F \leq \epsilon\mu\} \end{aligned}$$

The occurring error matrices again fulfil a Sylvester equation, namely

$$(\Delta A\tilde{Q} - \tilde{Q}\Delta C) - \Delta M = R \equiv M - (A\tilde{Q} - \tilde{Q}C).$$

This equation can be re-written using the vec operator,

$$\left( \alpha(\tilde{Q}^T \otimes I_n) \quad -\gamma(I_m \otimes \tilde{Q}) \quad \mu I_{nm} \right) \begin{pmatrix} \frac{\mathrm{vec}(\Delta A)}{\alpha} \\ \frac{\mathrm{vec}(\Delta C)}{\gamma} \\ \frac{\mathrm{vec}(\Delta M)}{\mu} \end{pmatrix} = \mathrm{vec}(R)$$

This is a linear equation $Bv = r$. The backward error can be bounded using the minimal residual solution $B^\dagger r$ of this equation, that is by

$$\frac{1}{\sqrt{3}} \|B^\dagger r\|_2 \leq \eta_2(\tilde{Q}) \leq \|B^\dagger r\|_2 \leq \|B^\dagger\|_2 \|r\|_2.$$

Neglecting the influence of the right-hand side, i.e. the cross-dependencies between the matrix $B$ and the residual of the approximate solution, we observe that the bound depends mainly on the smallest singular value of $B$ times the Frobenius norm of the residual. The smallest singular value of $B$ can be expressed using the smallest singular value $\sigma_m \equiv \sigma_m(\tilde{Q})$ of the *approximate solution* $\tilde{Q}$,

$$\|B^\dagger\|_2 = (\alpha^2 \sigma_m^2 + \mu^2)^{-1/2}.$$

For the direct proof we refer to Higham's paper (cf. [Hig93], equation (3.7)). In our case we set $\mu$ to zero, i.e. we are not concerned with any perturbations of the right-hand side. This implies that an *upper bound* for the backward error is given by

$$\eta_2(\tilde{Q}) \leq (\alpha \sigma_{\min}(\tilde{Q}))^{-1} \|R\|_F.$$

The minimal perturbation to achieve equality is small, as long as the matrix $\tilde{Q}$ has approximately orthonormal columns. The derivation of this bound clearly reveals that this bound may *grossly overestimate* the size of the minimal perturbation necessary to achieve equality.

Apart from the results based on constructing polynomial subspace equations, the stability analysis in this section did not honour the special *structure of the right-hand side* and the *iterative character* of the methods. In the next section we will derive some results that make extensive use of the structure.

## 4.8 The Computed Basis Vectors

In this section we are interested in the *local* and *global* deviation of the *computed* basis vectors $q_j$. We state a variety of results expressing the deviation in terms of different *computed* quantities.

We again use the governing equation

$$AQ_k - Q_k C_k = M_k - F_k \tag{4.42}$$

as the starting point of our examination. The convergence analysis in infinite precision is based on the eigendecomposition, thus we proceed by diagonalising the governing equation.

Multiplying (4.42) by $\hat{v}_i^H$ from the left and by $s_j$ from the right we obtain the set

$$(\lambda_i - \theta_j)\, \hat{v}_i^H Q_k s_j = \hat{v}_i^H q_{k+1} c_{k+1,k} s_{kj} - \hat{v}_i^H F_k s_j \quad \forall\, i \in \underline{n}, j \in \underline{k} \tag{4.43}$$

of equations. Here we have assumed that both $A$ and $C_k$ are diagonalisable.

Reordering the set of equations (4.43), we obtain the set

$$\begin{aligned} \hat{v}_i^H q_{k+1} &= \frac{(\lambda_i - \theta_j)\,\hat{v}_i^H Q_k s_j + \hat{v}_i^H F_k s_j}{c_{k+1,k} s_{kj}} \quad \forall\, i \in \underline{n}, j \in \underline{k} \\ &= \hat{v}_i^H Q_k \left[\frac{\lambda_i - \theta_j}{c_{k+1,k} s_{kj}}\right] s_j + \hat{v}_i^H F_k \left[\frac{1}{c_{k+1,k} s_{kj}}\right] s_j. \end{aligned} \tag{4.44}$$

The vector $y_j = Q_k s_j$ is the computed $j$th Ritz vector. This result relates a deviation to convergence of a Ritz pair. The result applies also to *non-diagonalisable* $A$ and $C_k$, since for *every* eigenvalue at least *one* eigenvector must exist.

**Theorem 4.32** *Let $A \in \mathbb{K}^{n\times n}$. Let $\hat{v}_i^H$ be a* left *eigenvector of $A$ to eigenvalue $\lambda_i$. Let $q = q_1$ be the starting vector of a Krylov method that computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*Let $s_j$ be a* right *eigenvector of $C_k$ to eigenvalue $\theta_j$, and $y_j = Q_k s_j$ the corresponding Ritz vector.*

*When $(\lambda_i - \theta_j)\,\hat{v}_i^H y_j$ becomes* as small *as $\hat{v}_i^H F_k s_j$, the basis vector $q_{k+1}$ starts to deviate in direction $\hat{v}_i^H$,*

$$\hat{v}_i^H q_{k+1} = \frac{(\lambda_i - \theta_j)\,\hat{v}_i^H y_j + \hat{v}_i^H F_k s_j}{c_{k+1,k} s_{kj}}.$$

*The size of deviation is proportional to the inverse of the last component of $s_j$.*

In other words, a deviation can only occur when the method is converging. Furthermore, the denominator $c_{k+1,k} s_{kj}$ is frequently used as indicator for convergence in eigenproblem solvers. When the rounding errors are random, we almost surely have a deviation in case of *observable* convergence.

When $C_k$ is diagonalisable, we can say more about size and shape of the deviation. The matrix of eigenvectors $S \equiv S_k$ of course is invertible. We use this fact to express the $l$th unit vector in terms of eigenvector entries:

$$I = SS^{-1} = S\check{S}^T \quad \Rightarrow \quad e_l = S\check{S}^T e_l \equiv \sum_{j=1}^{k} \check{s}_{lj} s_j.$$

Here again we used the shorthand notation $\check{s} = \overline{\hat{s}}$.

We are interested in the contribution of the error between steps $l$ and $k+1$. For this reason we introduce the representation

$$\sum_{j=1}^{k} \left(\frac{c_{k+1,k} s_{kj} \check{s}_{lj}}{\lambda_i - \theta_j}\right) \left(\frac{\lambda_i - \theta_j}{c_{k+1,k} s_{kj}}\right) s_j = e_l$$

of the $l$th unit vector. We sum up equations (4.44) to pick up the single equation

$$\left[\sum_{j=1}^{k} \frac{c_{k+1,k} s_{kj} \check{s}_{lj}}{\lambda_i - \theta_j}\right] \hat{v}_i^H q_{k+1} = \hat{v}_i^H q_l + \hat{v}_i^H F_k \left[\sum_{j=1}^{k} \left(\frac{\check{s}_{lj}}{\lambda_i - \theta_j}\right) s_j\right]. \tag{4.45}$$

For $l = 1$ we obtain a formula that reveals how the errors affect the recurrence from the beginning.

**Theorem 4.33** *Let $A \in \mathbb{K}^{n\times n}$. Let $\hat{v}_i^H$ be a* left *eigenvector of $A$ to eigenvalue $\lambda_i$. Let $q = q_1$ be the starting vector of a Krylov method that computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*Assume $C_k$, $k \in \underline{m}$ diagonalisable. Assume further that the right eigenvectors $s_j$, $j \in \underline{k}$ of $C_k$ are normalised such that $\check{s}_j^T s_j = \hat{s}_j^H s_j = 1$, where $\hat{s}_j^H$, $j \in \underline{k}$ denotes the left eigenvector to eigenvalue $\theta_j$.*

*Then the part of the finite precision recurrence of the basis vectors in direction of $\hat{v}_i^H$ is given by*

$$\left[\sum_{j=1}^{k} \frac{c_{k+1,k} s_{kj} \check{s}_{1j}}{\lambda_i - \theta_j}\right] \hat{v}_i^H q_{k+1} = \hat{v}_i^H q + \hat{v}_i^H F_k \left[\sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{\lambda_i - \theta_j}\right) s_j\right].$$

Theorem 4.33 sheds some more light on the deviation. It can be interpreted as follows: A deviation in direction of the $i$th eigenvector $\hat{v}_i^H$ of $A$ can only occur when an eigenvalue $\theta_j$ approaches the corresponding eigenvalue $\lambda_i$. The size of the deviation depends on the *size* of the *first component* of the *left* eigenvector $\hat{s}_j$ of $C_k$ and the *shape and size* of the *right* eigenvector $s_j$. When a Ritz value is converging, the eigenvector component is damped when no errors are present. In the presence of errors the picture changes. When we assume the Ritz value to be simple, we observe the rate of deviation we already mentioned in Theorem 4.32. This rate may be a crude overestimate in case of several close Ritz values.

We are usually more interested in Ritz *values*, than in Ritz *vectors* and thus in eigenvectors of $C_k$. It turns out that it is possible to express part of the left-hand side in terms of Ritz values and sub-diagonal elements instead solely in terms of eigenvectors.

To see this, observe that the quantity $c_{k+1,k} s_{kj} \check{s}_{lj}$ can be expressed in terms of Ritz values and sub-diagonal elements. To be more precise, we have proven in Chapter 2 that in case of diagonalisable Hessenberg matrices the relations

$$\begin{aligned} s_{kj}\check{s}_{lj} &= \frac{\prod_{p=l}^{k-1} c_{p+1,p} \prod \left(\theta_i^{(l-1)} - \theta_j^{(k)}\right)}{\prod_{i\neq j} \left(\theta_i^{(k)} - \theta_j^{(k)}\right)} \\ &= \frac{\left(\prod_{p=l}^{k-1} c_{p+1,p}\right) \chi_{C_{l-1}}(\theta_j)}{\chi'_{C_k}(\theta_j)} \quad \forall\, l \in \underline{k} \end{aligned}$$

hold true. Thus, in the most interesting case $l = 1$,

$$s_{kj}\check{s}_{1j} = \frac{\prod_{p=1}^{k-1} c_{p+1,p}}{\chi'_{C_k}(\theta_j)}, \quad \text{i.e.} \quad c_{k+1,k} s_{kj} \check{s}_{1j} = \frac{\prod_{p=1}^{k} c_{p+1,p}}{\chi'_{C_k}(\theta_j)}$$

holds true. This proves the following theorem:

**Theorem 4.34** *Let $A \in \mathbb{K}^{n\times n}$. Let $\hat{v}_i^H$ be a* left *eigenvector of $A$ to eigenvalue $\lambda_i$. Let $q = q_1$ be the starting vector of a Krylov method that computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*Assume $C_k$, $k \in \underline{m}$ diagonalisable. Assume further that the right eigenvectors $s_j$, $j \in \underline{k}$ of $C_k$ are normalised such that $\check{s}_j^T s_j = \hat{s}_j^H s_j = 1$, where $\hat{s}_j^H$, $j \in \underline{k}$ denotes the left eigenvector to eigenvalue $\theta_j$.*

*Then the part of the finite precision recurrence of the basis vectors in direction of $\hat{v}_i^H$ is given by*

$$\left[\sum_{j=1}^{k} \frac{\prod_{p=1}^{k} c_{p+1,p}}{\chi'_{C_k}(\theta_j)(\lambda_i - \theta_j)}\right] \hat{v}_i^H q_{k+1} = \hat{v}_i^H q_1 + \hat{v}_i^H F_k \left[\sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{\lambda_i - \theta_j}\right) s_j\right]$$

*We remark that close to every matrix with a non-simple Jordan block there is a diagonalisable matrix with ill-conditioned eigenvectors.*

The quantity $\chi'_{C_k}(\theta_j)(\lambda_i - \theta_j)$ is simply the first nonzero term in the Taylor expansion around $\theta_j$ of $\chi_{C_k}$ evaluated at $\lambda_i$, i.e.

$$\begin{aligned} \chi_{C_k}(\lambda_i) &= \chi_{C_k}(\theta_j) \\ &+ \chi'_{C_k}(\theta_j)(\lambda_i - \theta_j) \\ &+ \frac{\chi''_{C_k}(\phi_{ij})}{2}(\lambda_i - \theta_j)^2 \end{aligned}$$

where $\phi_{ij}$ lies somewhere between $\lambda_i$ and $\theta_j$. Of course, since the $\chi$ are polynomials, we can determine the complete Taylor expansion and give another representation, since many terms cancel.

The explicit expression of the expansion *after* cancellation took place is easily obtained using a slightly different point of view. It is simple calculation to show that when $\lambda_i \neq \theta_\ell$ for all $\ell$,

$$(\lambda_i - \theta_j) = \frac{\chi_{C_k}(\lambda_i)}{\prod_{\ell \neq j}(\lambda_i - \theta_\ell)}$$

holds true. Thus, using the theory of polynomial interpolation, we observe that we have obtained nothing but an *interpolating polynomial* in *Lagrange form* that interpolates a *constant function*,

$$\begin{aligned} \sum_{j=1}^{k} \frac{1}{\chi'_{C_k}(\theta_j)(\lambda_i - \theta_j)} &= \frac{1}{\chi_{C_k}(\lambda_i)} \sum_{j=1}^{k} \frac{\prod_{\ell \neq j}(\lambda_i - \theta_\ell)}{\prod_{\ell \neq j}(\theta_j - \theta_\ell)} \\ &= \frac{1}{\chi_{C_k}(\lambda_i)}. \end{aligned}$$

Our recurrence is then transformed into the equation

$$\hat{v}_i^H q_{k+1} = \frac{\chi_{C_k}(\lambda_i)}{\prod_{p=1}^{k} c_{p+1,p}} \left(\hat{v}_i^H q_1 + \hat{v}_i^H F_k \left[\sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{\lambda_i - \theta_j}\right) s_j\right]\right), \tag{4.46}$$

a formula revealing explicitly the forward error in the finite precision counterpart to the evaluation in infinite precision given in formula (2.11).

This formula clearly reveals that we will almost never compute a zero vector when using a Krylov method in finite precision. A second message is that unless the Arnoldi vectors are stored and orthogonalised explicitly, the orthogonality will become lost *whenever* the method converges, and from formula (4.46) we see in which direction the deviation occurs.

We can collect all eigenparts to rebuild the vectors $q_{k+1}$. This results in the following theorem:

**Theorem 4.35** *Let $A \in \mathbb{K}^{n \times n}$. Let $q = q_1$ be the starting vector of a Krylov method that computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*Assume $A$ and $C_k$, $k \in \underline{m}$ diagonalisable. Assume further that the right eigenvectors $v_i$, $i \in \underline{n}$ of $A$ and $s_j$, $j \in \underline{k}$ of $C_k$ are normalised such that $\hat{v}_i^H v_i = 1$ and $\check{s}_j^T s_j = 1$, where $\hat{v}_i^H$, $i \in \underline{n}$ and $\check{s}_j^T$, $j \in \underline{k}$ is used to denote the left eigenvectors.*

*Then the $(k+1)$th basis vector has the representation*

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}} \left( q + \left[ \sum_{i=1}^{n} v_i \hat{v}_i^H F_k \left[ \sum_{j=1}^{k} \left( \frac{\check{s}_{1j}}{\lambda_i - \theta_j} \right) s_j \right] \right] \right). \tag{4.47}$$

*This implies that the $(k+1)$st basis vector $q_{k+1}$ is the* exact *$(k+1)$st basis vector for a* perturbed *starting vector and a* different *Krylov method. Furthermore, this theoretical Krylov method in general uses a different way to compute the matrices $C_k$.*

Theorem 4.35 implies that as long as no Ritz pair has converged and the matrices $C_k$ are computed in a stable manner, the computed basis vectors will be close to the exact ones.

From now on we focus on the error term on the right-hand side of equation (4.46). We want to derive a different representation of the complicated looking error term, given by the vector

$$F_k \left[ \sum_{j=1}^{k} \left( \frac{\check{s}_{1j}}{\lambda_i - \theta_j} \right) s_j \right] \equiv F_k w_k.$$

First we consider the $l$th component $w_{lk}$ of the composed vector $w_k$, where $l \in \underline{k}$ is arbitrary. This component is given by

$$w_{lk} = e_l^T w_k = e_l^T \left[ \sum_{j=1}^{k} \left( \frac{\check{s}_{1j}}{\lambda_i - \theta_j} \right) s_j \right] = \sum_{j=1}^{k} \left( \frac{\check{s}_{1j} s_{lj}}{\lambda_i - \theta_j} \right).$$

Again we make use of our eigenvector – eigenvalue relations we have obtained in Chapter 2. We know that

$$\check{s}_{1j} s_{lj} = \frac{\left( \prod_{p=1}^{l-1} c_{p+1,p} \right) \chi_{C_{l+1:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)}$$

holds true. The second tool again is polynomial interpolation. We rewrite another interpolation polynomial slightly (note that $l < k$):

$$\begin{aligned} \sum_{j=1}^{k} \frac{\chi_{C_{l+1:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)(\lambda_i - \theta_j)} &= \frac{1}{\chi_{C_k}(\lambda_i)} \sum_{j=1}^{k} \frac{\prod_{\ell \neq j} (\lambda_i - \theta_\ell)}{\prod_{\ell \neq j} (\theta_j - \theta_\ell)} \chi_{C_{l+1:k}}(\theta_j) \\ &= \frac{\chi_{C_{l+1:k}}(\lambda_i)}{\chi_{C_k}(\lambda_i)}. \end{aligned}$$

This proves that the above mentioned $l$th component of $w_k$ is given by

$$w_{lk} = \sum_{j=1}^{k} \left( \frac{\check{s}_{1j} s_{lj}}{\lambda_i - \theta_j} \right) = \frac{\left( \prod_{p=1}^{l-1} c_{p+1,p} \right) \chi_{C_{l+1:k}}(\lambda_i)}{\chi_{C_k}(\lambda_i)}.$$

We express the final result for the recurrence as theorem:

**Theorem 4.36** *Let $A \in \mathbb{K}^{n\times n}$. Let $\hat{v}_i^H$ be a* left *eigenvector of $A$ to eigenvalue $\lambda_i$. Let $q = q_1$ be the starting vector of a Krylov method that computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*Then the part of the finite precision recurrence of the basis vectors in direction of $\hat{v}_i^H$ is given by*

$$\hat{v}_i^H q_{k+1} = \frac{\chi_{C_k}(\lambda_i)}{\prod_{p=1}^{k} c_{p+1,p}}\ \hat{v}_i^H q_1 + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(\lambda_i)}{\prod_{p=l}^{k} c_{p+1,p}}\ \hat{v}_i^H f_l \right]. \tag{4.48}$$

**Proof.** The derivation of the result covers only the cases where $C_k$, $k \in \underline{m}$ is diagonalisable. This restriction is not necessary, which will follow by the proof of the next theorem. □

We noted that in infinite precision the basis vectors fulfil a recurrence that involves a polynomial of the matrix $A$. By summing the eigenparts we conclude that the finite precision methods still fulfils a perturbed equation of that type.

As we already mentioned, this result applies to *general* $A$ and *general* unreduced Hessenberg $C_k$:

**Theorem 4.37** *Let $A \in \mathbb{K}^{n\times n}$. Let $q = q_1$ be the starting vector of a Krylov method that computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*The vectors $q_j$ constructed to serve as a basis obey the relation*

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}}\ q + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(A)}{\prod_{p=l}^{k} c_{p+1,p}}\ f_l \right].$$

*This is similar to the relation*

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}}\ q$$

*that holds true in infinite precision. We stress the fact that the polynomials $\chi_{C_k}$* depend on the vectors constructed *and thus* differ *from the exact ones whenever $F_k \neq 0$. This also can be expected when* several methods *are used that compute* different *$F_k$.*

**Proof.** The proof relies on induction and on the structure of the Kronecker sum of $A$ and $C_k$. We know that in case of a simple (non-block) Krylov method $C_k$ is unreduced upper Hessenberg. Thus, the Sylvester equation

$$\left(I_k \otimes A - C_k^T \otimes I_n\right) \mathrm{vec}\,(Q_k) = \mathrm{vec}\,(M_k - F_k)$$

corresponds to the huge linear system

$$\begin{pmatrix} (A - c_{11}I) & -c_{21}I & 0 & \cdots & 0 \\ -c_{12}I & (A - c_{22}I) & -c_{32}I & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ -c_{1,k-1}I & \cdots & -c_{k-2,k-1}I & (A - c_{k-1,k-1}I) & -c_{k,k-1}I \\ -c_{1k}I & -c_{2k}I & \cdots & -c_{k-1,k}I & (A - c_{kk}I) \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{k-1} \\ q_k \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ q_{k+1} \end{pmatrix} c_{k+1,k} - \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{k-1} \\ f_k \end{pmatrix}.$$

For $q_1$ there is nothing to prove. Using the first row, for $q_2$ the result obviously holds true,

$$q_2 = \frac{(A - c_{11}I)}{c_{21}}\ q_1 + \frac{I}{c_{21}}\ f_1 = \frac{\chi_{C_1}(A)}{c_{21}}\ q_1 + \frac{\chi_{C_{2:1}}(A)}{c_{21}}\ f_1.$$

Next, assume that

$$q_j = \frac{\chi_{C_{j-1}}(A)}{\prod_{p=1}^{j-1} c_{p+1,p}}\ q_1 + \sum_{l=1}^{j-1} \left[ \frac{\chi_{C_{l+1:j-1}}(A)}{\prod_{p=l}^{j-1} c_{p+1,p}}\ f_l \right]$$

holds true *for all* $j \leq k$. Then using the last row and our induction hypothesis, indicated by $(\star)$,

$$\begin{aligned}
c_{k+1,k} q_{k+1} &= \sum_{j=1}^{k-1} -c_{jk} q_j + (A - c_{kk}I)\, q_k + f_k \\
&\overset{(\star)}{=} \sum_{j=1}^{k-1} -c_{jk} \left[ \frac{\chi_{C_{j-1}}(A)}{\prod_{p=1}^{j-1} c_{p+1,p}}\ q_1 + \sum_{l=1}^{j-1} \left[ \frac{\chi_{C_{l+1:j-1}}(A)}{\prod_{p=l}^{j-1} c_{p+1,p}}\ f_l \right] \right] \\
&+ (A - c_{kk}I) \left[ \frac{\chi_{C_{k-1}}(A)}{\prod_{p=1}^{k-1} c_{p+1,p}}\ q_1 + \sum_{l=1}^{k-1} \left[ \frac{\chi_{C_{l+1:k-1}}(A)}{\prod_{p=l}^{k-1} c_{p+1,p}}\ f_l \right] \right] + f_k \\
&= \left[ \sum_{j=1}^{k} -c_{jk} \frac{\chi_{C_{j-1}}(A)}{\prod_{p=1}^{j-1} c_{p+1,p}} + (A - c_{kk}I)\, \frac{\chi_{C_{k-1}}(A)}{\prod_{p=1}^{k-1} c_{p+1,p}} \right] q_1 \\
&+ \sum_{l=2}^{k-1} \left[ \sum_{j=l}^{k} -c_{jk} \frac{\chi_{C_{l+1:j-1}}(A)}{\prod_{p=l}^{j-1} c_{p+1,p}} + (A - c_{kk}I)\, \frac{\chi_{C_{l+1:k-1}}(A)}{\prod_{p=l}^{k-1} c_{p+1,p}} \right] f_l \\
&+ f_k \quad \left( = \frac{\chi_{C_{k:k-1}}(A) f_k}{\prod_{p=k}^{k-1} c_{p+1,p}} \right) \\
&= \frac{\chi_{C_k}(A) q_1}{\prod_{p=1}^{k-1} c_{p+1,p}} + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(A) f_l}{\prod_{p=l}^{k-1} c_{p+1,p}} \right].
\end{aligned}$$

The last line follows by block Laplace expansion of the block determinant by last row for the matrix and all trailing principal submatrices. This block determinant is well-defined, since all block entries of the Kronecker sum of $A$ and $-C_k^T$ commute and the determinant is uniquely defined for commutative rings. We stress the fact that we have used expansion by *cofactors* and not by *minors* which are related by

$$\frac{\chi_{C_{j-1}}(A)}{\prod_{p=1}^{j-1} c_{p+1,p}}\ q_1 = (-1)^{k-j}\ \frac{\chi_{C_{j-1}}(A)}{\prod_{p=1}^{j-1} -c_{p+1,p}}\ q_1.$$

This finishes the proof. □

The finite precision version can be seen as a the additive mixture of several runs of the method simultaneously, where the runs 2 to $k$ take part with a starting vector

that has a small norm. In other words, a finite precision Krylov methods behaves as if it were invoked several times (once every new step) with a new starting vector, the result obtained is the sum of all runs implicitly done in the method. We remark that the overlay of these methods is used in the normalisation step, i.e. the single methods use different normalisations.

We note that Theorem 4.37 together with Corollary 4.9 implies a nice identity we just want to state as a corollary:

**Corollary 4.38** *Let $A \in \mathbb{K}^{n\times n}$. Let $C_k \in \mathcal{H}_k$. Let $F_k \in \mathbb{K}^{n\times k}$ be such that for some $Q_k \in \mathbb{K}^{n\times k}$ and $M_k \equiv r_k e_k^T$, $r_k \in \mathbb{K}^n$*

$$AQ_k - Q_k C_k = M_k - F_k$$

*holds true. Let the coefficients of the characteristic polynomial $\chi_{C_k}$ of $C_k$ be given by*

$$\chi_{C_k}(\lambda) = \det(\lambda I_k - C_k) = \sum_{j=0}^{k} \alpha_j \lambda^j.$$

*Then the identity*

$$\frac{\sum_{j=0}^{k-1} \alpha_{j+1} \sum_{i=0}^{j} A^{j-i} F_k C_k^i}{\prod_{p=1}^{k} c_{p+1,p}}\ e_1 = \sum_{l=1}^{k} \frac{\chi_{C_{l+1:k}}(A)}{\prod_{p=l}^{k} c_{p+1,p}}\ f_l$$

*holds true. The* left*-hand side reveals the influences of the* coefficients of the characteristic polynomial*, whereas the* right*-hand side reveals the influences of the* columns of the error matrix $F_k$.

The error vectors $f_j$ have severe impacts on the behaviour of the methods. The preceeding considerations where in some sense related to a forward approach. From the theorems in this section it is quite obvious that only in very limited cases the methods can be predicted beyond the step where the *first Ritz pair* has converged. Using the computed quantities, the errors can be described quite accurate. This backward approach is considered in the next section in more detail.

## 4.9 Impacts of the $l$th Error

Let us once again consider the set of equations (4.44),

$$\hat{v}_i^H q_{k+1} = \hat{v}_i^H Q_k \left[\frac{\lambda_i - \theta_j}{c_{k+1,k} s_{kj}}\right] s_j + \hat{v}_i^H F_k \left[\frac{1}{c_{k+1,k} s_{kj}}\right] s_j.$$

Similarly to the formulae showing how the errors are amplified in the methods, a formula expressing the error of the $l$th step in terms of quantities computed later on can be obtained, because we can deduce from

$$I = SS^{-1} = S\check{S}^T \quad \Rightarrow \quad e_l = S\check{S}^T e_l \equiv \sum_{j=1}^{k} \check{s}_{lj} s_j$$

that the $l$th standard unit vector can alternatively be represented by

$$\sum_{j=1}^{k} \frac{c_{k+1,k} s_{kj} \check{s}_{lj}}{c_{k+1,k} s_{kj}} s_j = e_l.$$

Carefully choosing the appropriate linear combination of equations from (4.44), this results in

**Theorem 4.39** *Let $A \in \mathbb{K}^{n\times n}$. Let $\hat{v}_i^H$ be a* left *eigenvector of $A$ to eigenvalue $\lambda_i$. Assume that a finite precision Krylov method computed the perturbed Hessenberg decomposition*

$$AQ_m - Q_m C_m = q_{m+1} c_{m+1,m} e_m^T - F_m.$$

*Suppose that $C_k$, $k \in \underline{m}$ is diagonalisable. Assume further that the right eigenvectors $s_j$, $j \in \underline{k}$ of $C_k$ are normalised such that $\check{s}_j^T s_j = \hat{s}_j^H s_j = 1$, where $\hat{s}_j^H$, $j \in \underline{k}$ denotes the left eigenvector to eigenvalue $\theta_j$.*

*Then the part of the error vector of step $l < k$ in direction of $\hat{v}_i^H$ can be recovered from the exact quantities of $A$ and the in step $k$ computed approximations,*

$$\hat{v}_i^H f_l = \left[\sum_{j=1}^{k} (\theta_j - \lambda_i)\, \check{s}_{lj} \hat{v}_i^H y_j\right] = \hat{v}_i^H Q_k \left[\sum_{j=1}^{k} (\theta_j - \lambda_i)\, \check{s}_{lj} s_j\right] \tag{4.49}$$

*The error vectors $f_l$, $l < k$ can, even in the* non-diagonalisable *case, be recovered by forming the linear combination*

$$f_l = \sum_{j=1}^{k} \check{s}_{lj} r_j = R_k(\check{S}_k^T e_l) \tag{4.50}$$

*of the residuals $r_j$ of step $k$. Here we have defined the matrix $R_k$ of residual vectors $r_j$ to be $R_k \equiv Y_k J_\Theta - AY_k$.*

**Proof.** The expression resulting from forming the appropriate linear combination is given by

$$\begin{aligned}\left[\sum_{j=1}^{k} c_{k+1,k} s_{kj} \check{s}_{lj}\right] \hat{v}_i^H q_{k+1} &= \left[\sum_{j=1}^{k} (\lambda_i - \theta_j)\, \check{s}_{lj} \hat{v}_i^H y_j\right] + \hat{v}_i^H f_l \qquad (4.51)\\ &= \hat{v}_i^H Q_k \left[\sum_{j=1}^{k} (\lambda_i - \theta_j)\, \check{s}_{lj} s_j\right] + \hat{v}_i^H f_l.\end{aligned}$$

The bracket on the left-hand side turns out to be zero, since by definition of $\check{S}_k^T = S_k^{-1}$ obviously $S_k \check{S}_k^T = I_k$. Thus the inner product of the vectors $e_k^T S_k$ and $\check{S}_k^T e_l$, i.e. the sum $\sum_{j=1}^{k} s_{kj}\check{s}_{lj}$, is zero whenever $l < k$.

For the second part, observe that the multiplication of the governing equation with $S_k$ results in

$$\begin{aligned}-F_k S_k &= AY_k - Y_k J_\Theta - M_k S_k\\ -F_k S_k(\check{S}_k^T e_l) &= (AY_k - Y_k J_\Theta - M_k S_k)(\check{S}_k^T e_l)\\ -f_l &= (AY_k - Y_k J_\Theta)(\check{S}_k^T e_l)\end{aligned}$$

This finishes the proof. □

We have products of left and right eigenvector components on both sides of equation (4.51). Again we can express these parts of the equation with the products of eigenvector components replaced by fractions composed of off-diagonal elements and Ritz values. First, we focus on the left-hand side, even though we already know that the sum is zero. By previous considerations the summands inside the brackets can be expressed like

$$c_{k+1,k} s_{kj} \check{s}_{lj} = \frac{\left(\prod_{p=l}^{k} c_{p+1,p}\right) \chi_{C_{l-1}}(\theta_j)}{\chi'_{C_k}(\theta_j)} \quad \forall\, l \in \underline{k}. \tag{4.52}$$

This gives rise to the following corollary:

**Corollary 4.40** *Let all notations be defined as before. As a direct consequence of the bi-orthogonality of the eigenvectors, for all $l < k$*

$$\left(\prod_{p=l}^{k} c_{p+1,p}\right) \sum_{j=1}^{k} \frac{\chi_{C_{l-1}}(\theta_j)}{\chi'_{C_k}(\theta_j)} = 0 \tag{4.53}$$

*holds true.*

The eigenvalues of $A$ played no role in determining the size of the term arising in the right-hand side of equation (4.52). This enables us to consider a more geometric approach towards a better understanding of this term. Following Wilkinson ([Wil63], chapter 2, section 7, pages 38–41), we can interpret the right-hand side as the first order perturbation term of the simple zero $\theta_j$ when we perturb the polynomial $\chi_{C_k}$ by $\epsilon$ times the polynomial $\chi_{C_{l-1}}$ times a constant factor. In other words, when we examine the zeros $\tilde{\theta}_j$ of

$$\tilde{\chi}_{C_k} \equiv \chi_{C_k} + \epsilon \left(\prod_{p=l}^{k} c_{p+1,p}\right) \chi_{C_{l-1}}, \tag{4.54}$$

we observe that for sufficiently small $\epsilon$ they behave according to

$$\tilde{\theta}_j = \theta_j + \epsilon \frac{\left(\prod_{p=l}^{k} c_{p+1,p}\right) \chi_{C_{l-1}}(\theta_j)}{\chi'_{C_k}(\theta_j)} + O(\epsilon^2).$$

So, the sum on the left-hand side of equation (4.53) is the sum of the *condition numbers* of the single zeros $\theta_j$ when the characteristic polynomial is subject to a *structured perturbation* as indicated by equation (4.54). Equation (4.53) proves that the single perturbations are *by no means unstructured*, since they have to sum up to zero.

From now on, we focus on the right-hand side of equation (4.49). The vectors $(\theta_j - \lambda_i)\, \check{s}_{lj} s_j$ consist of products having the form

$$(\theta_j - \lambda_i)\, \check{s}_{lj} s_{\ell j}, \qquad \ell \in \underline{k}. \tag{4.55}$$

These expressions can, in general, no longer be transformed using the eigenvector – eigenvalue relations, since in case of Hessenberg matrices the occurring polynomials are only easily computable whenever $l \leq \ell$. There are two special cases we want to focus on. In the first case we stick to the case of a Hessenberg matrix, which seems to fix $l \equiv 1$, i.e. we are analysing the impacts of the *first* error vector on the recurrence. In this case the terms (4.55) can be transformed into

$$(\theta_j - \lambda_i)\, \check{s}_{1j} s_{\ell j} = (\theta_j - \lambda_i) \frac{\left(\prod_{p=1}^{\ell-1} c_{p+1,p}\right) \chi_{C_{\ell+1:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)} \qquad \forall\, \ell \in \underline{k}.$$

Again, we may interpret this term as the condition number for the sensitivity of the characteristic polynomial $\chi_{C_k}$ when subject to a structured perturbation. This time the perturbations are weighted by the distances of the Ritz values to an eigenvalue. But, more interesting is the interpretation of the sum as a polynomial interpolation. When $\ell > 2$, we can re-write the sum with constant factors left out according to

$$\sum_{j=1}^{k} (\theta_j - \lambda_i) \frac{\chi_{C_{\ell+1:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)} \quad = \quad \sum_{j=1}^{k} \frac{\chi_{C_{\ell+1:k}}(\theta_j)\,(\lambda_i - \theta_j)\,(\theta_j - \lambda_i)}{\chi'_{C_k}(\theta_j)\,(\lambda_i - \theta_j)}$$

$$\begin{aligned} &\equiv \sum_{j=1}^{k} \frac{\psi_{k-\ell+2}(\theta_j)}{\chi'_{C_k}(\theta_j)\,(\lambda_i-\theta_j)} \\ &= \frac{1}{\chi_{C_k}(\lambda_i)} \sum_{j=1}^{k} \frac{\prod_{s\neq j}(\lambda_i-\theta_s)}{\prod_{s\neq j}(\theta_j-\theta_s)}\ \psi_{k-\ell+2}(\theta_j) \\ &= \frac{\psi_{k-\ell+2}(\lambda_i)}{\chi_{C_k}(\lambda_i)} \ = \ 0. \end{aligned}$$

In the above, we have used the abbreviation

$$\psi_{k-\ell+2}(\theta) \equiv -\chi_{C_{\ell+1:k}}(\theta)(\theta-\lambda_i)^2.$$

The characteristic polynomial $\chi_{C_{\ell+1:k}}$ has degree $k-\ell$ by definition. Thus, the polynomial $\psi_{k-\ell+2}$ has degree $k-\ell+2$. Interpolation is *exact*, whenever the number of nodes exceeds the degree of the polynomial interpolated. Hence the last line follows, since the assumption $\ell > 2$ implies $k > k-\ell+2$. This proves that most of the entries of the vector are zero.

The first two entries can still be re-written as an interpolation of the polynomial $\psi_{k-\ell+2}$, but this time interpolation is *not exact*, and thus these entries in general will not vanish. Nevertheless, this interpolational point of view might help in understanding the occurring phenomena, especially the sizes of the eigenvector entries. Observe furthermore that the proof did not depend on the fact that the polynomials $\chi_{C_{\ell+1:k}}$ are characteristic polynomials of submatrices. This observation is the key result to extend the preceeding considerations. Thus far, they can be used to shed some light on the relations between the *first* error vector $f_1$ and subsequently computed iterates:

**Lemma 4.41** *Let all notations be defined as stated in the last theorem. Then the part of the* first error vector *in direction* $\hat{v}_i^H$ *can be expressed with the aid of the* first two entries *of the eigenvectors of step* $k > 1$ *and the* first two *computed basis vectors,*

$$\begin{aligned} \hat{v}_i^H f_1 &= \sum_{j=1}^{k} (\theta_j-\lambda_i)\ \check{s}_{1j} \begin{pmatrix} s_{1j} \\ s_{2j} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \hat{v}_i^H q_2 \end{pmatrix} \\ &= \sum_{j=1}^{k} (\theta_j-\lambda_i) \begin{pmatrix} \dfrac{\chi_{C_{2:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)} \\ c_{21}\dfrac{\chi_{C_{3:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \hat{v}_i^H q_2 \end{pmatrix} \\ &= \frac{1}{\chi_{C_k}(\lambda_i)} \begin{pmatrix} \mathcal{L}\left[\zeta_1\right](\lambda_i) \\ \mathcal{L}\left[\zeta_2\right](\lambda_i) \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \hat{v}_i^H q_2 \end{pmatrix}. \end{aligned}$$

*Here, the polynomials* $\zeta_1$ *and* $\zeta_2$ *are given by*

$$\begin{aligned} \zeta_1(\theta) &\equiv -\chi_{C_{2:k}}(\theta)(\theta-\lambda_i)^2 \quad \textit{and} \\ \zeta_2(\theta) &\equiv -c_{21}\chi_{C_{3:k}}(\theta)(\theta-\lambda_i)^2. \end{aligned}$$

*Furthermore, we used the short-hand notation* $\mathcal{L}\left[p\right](\lambda)$ *to denote the value of the (Lagrange) interpolation polynomial that interpolates the function* $p$ *at the nodes* $\theta_j$, $j \in \underline{k}$, *evaluated at* $\lambda$,

$$\mathcal{L}\left[p\right](\lambda) \equiv \sum_{j=1}^{k} \frac{\prod_{i\neq j}(\lambda-\theta_i)}{\prod_{i\neq j}(\theta_j-\theta_i)}\, p(\theta_j).$$

*This shows that the first error vector corresponds to an error in an interpolation process. The error vector is* damped *when the characteristic polynomial $\chi_{C_k}$ has a small value at $\lambda_i$ and the interpolation processes are more accurate.*

In case of $l = 1$ the interpolation terms can be handled explicitely. It is a well-known fact that the Newton form of the interpolation polynomial can be used to express the interpolation error *explicitly*. We are interested in the interpolation of a polynomial $p$ at knots $\theta_j$, $j \in \underline{k}$ and wish to evaluate the resulting polynomial at some value $\lambda$. Then one can show that the interpolation error is given by the next summand in the interpolation at knot $\lambda$, i.e. by

$$(p - \mathcal{L}[p])(\lambda) = \omega(\lambda)\, p[\theta_1, \ldots, \theta_k, \lambda].$$

Here, the notation $\omega(\lambda)$ is used to denote the knot-polynomial, which simplifies in our case to the characteristic polynomial of $C_k$,

$$\omega(\lambda) \equiv \prod_{j=1}^{k} (\lambda - \theta_j) = \chi_{C_k}(\lambda).$$

The notation $p[\theta_1, \ldots, \theta_k, \lambda]$ is used to denote the divided difference of $p$ at knots $\theta_j$, $j \in \underline{k}$ and $\lambda$. Since both polynomials $\zeta_1$, $\zeta_2$ are zero when evaluated at $\lambda_i$, we know that the interpolation error is equal to the negated evaluated interpolation polynomial at $\lambda_i$, i.e.

$$\begin{aligned} -\mathcal{L}[\zeta_1](\lambda_i) &= \chi_{C_k}(\lambda_i)\, \zeta_1[\theta_1, \ldots, \theta_k, \lambda_i], \\ -\mathcal{L}[\zeta_2](\lambda_i) &= \chi_{C_k}(\lambda_i)\, \zeta_2[\theta_1, \ldots, \theta_k, \lambda_i]. \end{aligned}$$

To obtain a more convenient expression for the divided difference, we generalise the well-known formula for the error in the real interpolation of sufficiently smooth functions to the complex case. For this generalisation it is necessary to observe that the polynomials $\zeta_1$ and $\zeta_2$ have degrees $k + 1$ and $k$, respectively.

**Lemma 4.42** *Let $\theta_j \in \mathbb{C}$, $j \in \underline{k}$ be $k$ point in the complex plane. Let an additional distinct point $\lambda \in \mathbb{C}$ and a polynomial $p \in \mathbb{P}_k$ be given. Then the interpolation error can be expressed as follows:*

$$(p - \mathcal{L}[p])(\lambda) = \omega(\lambda) \frac{p^{(k)}(\xi)}{k!}.$$

*Here, $\xi$ is some value contained in the convex hull of the points $\theta_j$, $j \in \underline{k}$ and $\lambda$. This implies that the divided difference $p[\theta_1, \ldots, \theta_k, \lambda]$ can be expressed as*

$$p[\theta_1, \ldots, \theta_k, \lambda] = \frac{p^{(k)}(\xi)}{k!},$$

*obviously for the same value $\xi$.*

**Proof.** The proof is along the lines of the proof for the real case. We consider the auxiliary function

$$f(\lambda) \equiv (p - \mathcal{L}[p])(\lambda) - \alpha\omega(\lambda).$$

Here, we choose the constant $\alpha$ such that $\lambda$ is a root of $f$. We observe that the function $f$ is a polynomial of degree less equal $k + 1$ by construction. Furthermore, this polynomial has $k$ root at the knots $\theta_j$, $j \in \underline{k}$ and one at $\lambda$. By Lucas' theorem we know that the roots of the derivative of a polynomial are contained in the convex hull of the roots of the polynomial, and thus we have a nested sequence of convex hulls containing all roots of all derivative. Next, we form the $k$th derivative. When

this derivative is zero, we can choose any point $\xi$ in the complex plane, otherwise we choose $\xi$ to be the unique root. In any case, we obtain that

$$0 \equiv f^{(k)}(\xi) = p^{(k)}(\xi) - 0 - \alpha k!$$

holds true. This determines the constant $\alpha$ and shows that the interpolation error is given by

$$(p - \mathcal{L}[p])(\lambda) = \omega(\lambda)\frac{p^{(k)}(\xi)}{k!}.$$

This proves the first proposition of the lemma. Together with the Newton-type interpolation error formula, this proves the second proposition. □

We will not consider this approach in full detail. We only state the reformulation of Lemma 4.41 arising from the last lemma:

**Lemma 4.43** *Let all notation be as in Lemma 4.41. Denote the $k-1$ roots of the characteristic polynomial of $C_{2:k}$ by $\theta_j^{(2:k)}$, $j \in \underline{k-1}$. Then*

$$\begin{aligned}
\hat{v}_i^H f_1 &= -\begin{pmatrix} \zeta_1[\theta_1, \ldots, \theta_k, \lambda_i] \\ \zeta_2[\theta_1, \ldots, \theta_k, \lambda_i] \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \hat{v}_i^H q_2 \end{pmatrix} \\
&= \frac{-1}{k!}\begin{pmatrix} \zeta_1^{(k)}(\xi) \\ \zeta_2^{(k)} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \hat{v}_i^H q_2 \end{pmatrix} \\
&= \begin{pmatrix} (k+1)\xi - (2\lambda_i + \sum_{j=1}^{k-1}\theta_j^{(2:k)}) \\ c_{21} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \hat{v}_i^H q_2 \end{pmatrix} \qquad (4.56)
\end{aligned}$$

*holds true. The value $\xi$ is contained in the convex hull of the eigenvalues of $C_k$ and $\lambda_i$. To be more precise, $\xi$ is uniquely determined to be the* arithmetic mean *of the eigenvalues of $C_k$ and $\lambda_i$,*

$$\xi \equiv \frac{\lambda_i + c_{11} + \sum_{j=1}^{k-1}\theta_j^{(2:k)}}{k+1} = \frac{\lambda_i + \operatorname{trace}(C_k)}{k+1}. \qquad (4.57)$$

**Proof.** The first two lines are immediate consequences of Lemma 4.42. For the third line, observe that the polynomial $\zeta_1$ has degree $k+1$ and has $k-1$ roots at the eigenvalues $\theta_j^{(2:k)}$, $j \in \underline{k-1}$ of $C_{2:k}$ and a double root at $\lambda_i$. The leading coefficient is minus one. Thus, the explicit expression of the $k$th derivative is given by

$$\begin{aligned}
\zeta_1^{(k)}(\xi) &= -(k+1)!\,\xi + k!\left(2\lambda_i + \sum_{j=1}^{k-1}\theta_j^{(2:k)}\right) \\
&= -k!\left(2(\lambda_i - \xi) + \sum_{j=1}^{k-1}\left(\theta_j^{(2:k)} - \xi\right)\right).
\end{aligned}$$

Similarly, the polynomial $\zeta_2$ has degree $k$ and leading coefficient $-c_{21}$. Thus, the the $k$th derivative of $\zeta_2$ is constant and given by

$$\zeta_2^{(k)} \equiv -k!c_{21}.$$

To derive the explicit representation of $\xi$ we observe that simple algebraic transformations on the governing equation ensure the validity of the relation

$$c_{21}\hat{v}_i^H q_2 = (\lambda_i - c_{11})\hat{v}_i^H q_1 + \hat{v}_i^H f_1. \qquad (4.58)$$

Comparing the coefficients of the scalar products in (4.56) and (4.58), we obtain the relation

$$c_{11} - \lambda_i = (k+1)\xi - (2\lambda_i + \sum_{j=1}^{k-1} \theta_j^{(2:k)}).$$

This proves the first part of equation (4.57). The second part follows by the invariance of the trace of a matrix under similarity transformations. Assume that the eigendecomposition of the trailing principal submatrix $C_{2:k}$ is given by

$$S_{2:k}^{-1} C_{2:k} S_{2:k} = J_{2:k}.$$

The matrix $C_k$ is block-similar to the matrix

$$\begin{pmatrix} 1 & \\ & S_{2:k} \end{pmatrix}^{-1} C_k \begin{pmatrix} 1 & \\ & S_{2:k} \end{pmatrix} = \begin{pmatrix} c_{11} & \star \\ \star & J_{2:k} \end{pmatrix}.$$

The latter has trace $c_{11} + \sum_{j=1}^{k-1} \theta_j^{(2:k)}$. This shows that this sum is indeed the trace of $C_k$, which finishes the proof. □

Instead, we focus on the natural extension of Lemma 4.41 to all error vectors *and* all Krylov methods, obviously with different polynomials. The polynomials are given by the polynomials $p_{\ell l}$ defined in the derivation of the eigenvalue – eigenvector relations, i.e. by

$$p_{\ell l} = \det L_{l\ell}, \quad \text{where} \quad L = \theta I - C_k.$$

The case $l \leq \ell$ can be handled by previous considerations. The case $l > \ell$ can be handled by expansion of the determinant using Leibniz' formula,

$$\det(A) = \sum_{\sigma \in S_n} (-1)^{\mathrm{sgn}(\sigma)} \prod_{i=1}^{n} a_{i\sigma(i)}.$$

The matrix $L_{l\ell}$ looks in case $l > \ell$ as follows:

$$L_{l\ell} \equiv \begin{pmatrix} B_{\ell-1} & \star & \star & \star \\ \star & \star & -c_{\ell l} & \star \\ 0 & B_{l-\ell-1} & \star & \star \\ 0 & 0 & \star & B_{k-l} \end{pmatrix} \in \mathbb{K}^{(k-1)\times(k-1)},$$

where $B_z$, for some integer $z$, denotes a matrix that has symbolically the form

$$B_z \equiv \begin{pmatrix} \theta+\star & \star & \star & \star \\ \star & \theta+\star & \star & \star \\ 0 & \ddots & \ddots & \star \\ 0 & 0 & \star & \theta+\star \end{pmatrix} \in \mathbb{K}^{z\times z}.$$

The symbol $\star$ is used to denote the negative of some block, row, column or entry of the original matrix $C_k$. Such a part is *independent* of the variable $\theta$. This implies that whenever $c_{\ell l}$ is non-zero, the polynomial $p_{\ell l}$, given by

$$p_{\ell l} = \det L_{l\ell} \equiv c_{\ell l}\, \theta^{k-2} + \sum_{j=0}^{k-3} \alpha_j \theta^j, \tag{4.59}$$

has *exactly* degree $k-2$. Thus, in case of $l > \ell$, the sum in the parentheses in the second line of equation (4.51),

$$\sum_{j=1}^{k} (\theta_j - \lambda_i) \check{s}_{lj} s_{\ell j},$$

will, in the generic case, be non-zero. Whenever the degree of the polynomial $p_{\ell l}$ is *strictly* less than $k-2$, the sum *will* be zero. This implies that in case of a generic *Hessenberg* matrix we also obtain again a relation that builds upon a *Hessenberg* matrix. A refined analysis actually proves that in the special case of a *truncated* recurrence we also obtain again a *truncated* recurrence. The last observation in turn implies that in case of a *short-term* recurrence again a *short-term* recurrence evolves. We state the corresponding results for the case of $C_k = H_k$ being a Hessenberg matrix, e.g. for the Arnoldi method, and for the case of $C_k = T_k$ being a tridiagonal matrix, e.g. for the Lanczos method.

**Theorem 4.44** *Let $A \in \mathbb{K}^{n\times n}$. Suppose the relation*

$$F_k = Q_k C_k - AQ_k + M_k$$

*holds true for some $F_k, Q_k \in \mathbb{K}^{n\times k}$, $C_k \in \mathcal{H}(k)$ and $M_k \in \mathbb{K}^{n\times k}$, where $M_k \equiv r_k e_k^T$. Furthermore, let $\hat{v}_i^H$ be a left eigenvector of $A$ and $\lambda_i$ be the corresponding eigenvalue. Suppose that $C_k$ is diagonalisable.*

*Then with the notation of Lemma 4.41*

$$\begin{aligned}
\hat{v}_i^H f_l &= \sum_{j=1}^{k} (\theta_j - \lambda_i)\, \check{s}_{lj} \begin{pmatrix} s_{1j} \\ \vdots \\ s_{l+1,j} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \vdots \\ \hat{v}_i^H q_{l+1} \end{pmatrix} \\
&= \frac{1}{\chi_{C_k}(\lambda_i)} \begin{pmatrix} \mathcal{L}\left[\zeta_{1l}\right](\lambda_i) \\ \vdots \\ \mathcal{L}\left[\zeta_{l+1,l}\right](\lambda_i) \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \vdots \\ \hat{v}_i^H q_{l+1} \end{pmatrix} \\
&= \frac{-1}{k!} \begin{pmatrix} \zeta_{1l}^{(k)} \\ \vdots \\ \zeta_{l-1,l}^{(k)} \\ \zeta_{ll}^{(k)}(\xi) \\ \zeta_{l+1,l}^{(k)} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \vdots \\ \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix} = \begin{pmatrix} c_{1l} \\ \vdots \\ c_{l-1,l} \\ (k+1)\xi - \upsilon \\ c_{l+1,l} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \vdots \\ \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix}
\end{aligned}$$

*holds true. Here, the polynomials $\zeta_{\ell l}$, $\ell \in \underline{l+1}$ are defined by*

$$\zeta_{\ell l}(\theta) \equiv -p_{\ell l}(\theta)(\theta - \lambda_i)^2 \quad \text{where} \quad p_{\ell l} = \det L_{l\ell}, \quad L = \theta I - C_k.$$

*The constant $\upsilon$ is given by*

$$\upsilon \equiv 2\lambda_i + \sum_{j=1}^{l-1} \theta_j^{(1:l-1)} + \sum_{j=1}^{k-l} \theta_j^{(l+1:k)}$$

*and the value of $\xi$ again is equal to the* arithmetic mean *of $\lambda_i$ and the eigenvalues of $C_k$,*

$$\xi = \frac{\lambda_i + \sum_{j=1}^{l-1} \theta_j^{(1:l-1)} + c_{ll} + \sum_{j=1}^{k-l} \theta_j^{(l+1:k)}}{k+1} = \frac{\lambda_i + \operatorname{trace}(C_k)}{k+1}.$$

**Proof.** First, we investigate the $k$th derivative of polynomials $\zeta_{\ell l}$ where $\ell < l$. From equation (4.59) we can conclude that the polynomial $\zeta_{\ell l}$ has (maximal) degree $k$ and that the leading term is given by $c_{\ell l}$. The case $\ell = l+1$ is treated analogously, using the eigenvector – eigenvalue relation. In the remaining case $\ell = l$, the polynomial $\zeta_{ll}$ takes the form

$$\begin{aligned}
\zeta_{ll}(\theta) &= -\chi_{C_{1:l-1}}(\theta)\chi_{C_{l+1:k}}(\theta)(\theta - \lambda_i)^2 \\
&= -\theta^{k+1} + \left(2\lambda_i + \sum_{j=1}^{l-1} \theta_j^{(1:l-1)} + \sum_{j=1}^{k-1} \theta_j^{(l+1:k)}\right)\theta^k + \dots
\end{aligned}$$

This fixes the $k$th derivative. The explicit expression for $\xi$ follows, since again by simple algebraic transformations the governing equation ensures that

$$\hat{v}_i^H f_l \;=\; \begin{pmatrix} c_{1l} \\ \vdots \\ c_{l-1,l} \\ c_{ll} - \lambda_i \\ c_{l+1,l} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_1 \\ \vdots \\ \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix}$$

holds true. By comparison of the coefficients, $\xi$ has to be the arithmetic mean of $\lambda_i$ and the eigenvalues of $C_k$. $\square$

The theorem simplifies when we are using a short-term recurrence, for example when we analyse the symmetric or unsymmetric Lanczos method:

**Theorem 4.45** *Let matrices and notation be defined as in Theorem 4.44 and Lemma 4.41. Suppose that the matrix $C_k \equiv T_k$ is tridiagonal, diagonalisable and given by*

$$T_k \;=\; \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{pmatrix} \;\in\; \mathbb{K}^{k\times k}.$$

*Then*

$$\begin{aligned} \hat{v}_i^H f_l &= \sum_{j=1}^{k} (\theta_j - \lambda_i)\, \check{s}_{lj} \begin{pmatrix} s_{l-1,j} \\ s_{lj} \\ s_{l+1,j} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix} \\ &= \frac{1}{\chi_{C_k}(\lambda_i)} \begin{pmatrix} \mathcal{L}\left[\zeta_{l-1}\right](\lambda_i) \\ \mathcal{L}\left[\zeta_l\right](\lambda_i) \\ \mathcal{L}\left[\zeta_{l+1}\right](\lambda_i) \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix} \\ &= \frac{-1}{k!} \begin{pmatrix} \zeta_{l-1}^{(k)} \\ \zeta_l^{(k)}(\xi) \\ \zeta_{l+1}^{(k)} \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix} \;=\; \begin{pmatrix} \gamma_{l-1} \\ (k+1)\xi - \upsilon \\ \beta_l \end{pmatrix}^T \begin{pmatrix} \hat{v}_i^H q_{l-1} \\ \hat{v}_i^H q_l \\ \hat{v}_i^H q_{l+1} \end{pmatrix} \end{aligned}$$

*holds true. Here, the polynomials $\zeta_{l-1}$, $\zeta_l$ and $\zeta_{l+1}$ are given explicitely by the expressions*

$$\begin{aligned} \zeta_{l-1}(\theta) &\equiv -p_{l-1,l}(\theta)(\theta-\lambda_i)^2 \\ &= -\gamma_{l-1}\chi_{T_{1:l-2}}(\theta)\chi_{T_{l+1:k}}(\theta)(\theta-\lambda_i)^2, \\ \zeta_l(\theta) &\equiv -p_{ll}(\theta)(\theta-\lambda_i)^2 \\ &= -\chi_{T_{1:l-1}}(\theta)\chi_{T_{l+1:k}}(\theta)(\theta-\lambda_i)^2, \\ \zeta_{l+1}(\theta) &\equiv -p_{l+1,l}(\theta)(\theta-\lambda_i)^2 \\ &= -\beta_l\chi_{T_{1:l-1}}(\theta)\chi_{T_{l+2:k}}(\theta)(\theta-\lambda_i)^2. \end{aligned}$$

*The value of the auxiliary constant $\upsilon$ is*

$$\upsilon = 2\lambda_i + \sum_{j=1}^{l-1} \theta_j^{(1:l-1)} + \sum_{j=1}^{k-l} \theta_j^{(l+1:k)}$$

*and $\xi$ is given explicitely by*

$$\xi \equiv \frac{\lambda_i + \sum_{j=1}^{l-1} \theta_j^{(1:l-1)} + \alpha_l + \sum_{j=1}^{k-l} \theta_j^{(l+1:k)}}{k+1} = \frac{\lambda_i + \operatorname{trace}(T_k)}{k+1}.$$

**Proof.** The proof is completely analogous to the proof for Theorem 4.44. □

The last two theorems clearly reveal the domain of influence of the local error vectors. Depending on the type of method, i.e. the length of the underlying recurrence, this domain is restricted to the same class. The relations confirm that there is a strong connection between Krylov methods and interpolation, and that the local errors are closely linked to the errors in a Lagrange-type interpolation. The results are also, not to say, *especially*, of interest in case of *infinite precision* arithmetic. In this case the right-hand side is zero, and the interpolation error vector must be *orthogonal* to the first components of the final right eigenvector to eigenvalue $\lambda_i$.

We briefly mention the idea to write equation (4.44) in the alternate form

$$\hat{v}_i^H q_{k+1} = \frac{1}{c_{k+1,k} s_{kj}} \left[ (\lambda_i - \theta_j) \begin{pmatrix} \hat{v}_i^H q_1 \\ \vdots \\ \hat{v}_i^H q_k \end{pmatrix} + \begin{pmatrix} \hat{v}_i^H f_1 \\ \vdots \\ \hat{v}_i^H f_k \end{pmatrix} \right]^T \begin{bmatrix} s_{1j} \\ \vdots \\ s_{kj} \end{bmatrix}. \tag{4.60}$$

With the results of this and the preceeding section in mind, formula (4.60) explains very nicely the influences of the single quantities on the part of the next basis vector $q_{k+1}$ in direction of the left eigenvector $\hat{v}_i^H$ of the original $A$. From the error analysis point of view we may regard the parts of the local error vectors in direction of $\hat{v}_i^H$ as constant and obtain a model that fits nicely the observed behaviour.

This way of representing the relations is of interest in the so-called *inexact Krylov methods*, in which the errors are allowed to grow, depending on the decay of the residuals of the quantities to be computed. Usually inexact Krylov methods are used in context of the solution of linear systems. Similar conclusions also hold in the context of eigenvalue computations. This is based on two observations. First, assume that the method in *infinite precision* arithmetic succeeds to compute the *final* Hessenberg decomposition

$$AQ_m = Q_m C_m.$$

Now, observe that in this case the vector

$$\hat{v}_i^H Q_m, \qquad \lambda_i \hat{v}_i^H Q_m = \hat{v}_i^H A Q_m = \hat{v}_i^H Q_m C_m$$

is a *left eigenvector* of $C_m$. The next observation is that this vector is computed componentwise starting with the first component. In finite precision, equation (4.60) reveals the influences of the error vectors $f_l$ on the next component of the computed *left eigenvector*. We assume that the Ritz values are more accurate than the Ritz vectors, which is usually the case. Furthermore, we assume that we can neglect the errors in previously computed components. Then, the deviation from the *exact* counterpart can be described componentwise by

$$\left| \hat{v}_i^H q_{k+1}^{\text{exact}} - \hat{v}_i^H q_{k+1}^{\text{computed}} \right| \lessapprox \frac{1}{|c_{k+1,k} s_{kj}|} \sum_{l=1}^{k} \|\hat{v}_i^H f_l\| |s_{lj}|.$$

When the eigenvector elements decay, the error vectors may be allowed to grow without spoiling the accuracy of $\hat{v}_i^H Q_{k+1}$. Caution has to be used in the non-normal case, since we basically ignored the effects of the angles between left and right eigenvectors, i.e. the eigenvalue conditions.

## 4.10 Measures of Convergence and Deviation

In this section we exploit the results of the preceeding sections to understand the intimate relations between a *convergence* and a *deviation*, which not necessarily

implies a *deterioration* of the process. The results presented are, to some extent, rather rules of thumb than strict mathematical theory. Despite lack of rigour, most observations could be treated and quantified by specialised perturbation theory. We first consider the methods for the eigenproblem. We are interested in the accuracy of Ritz values, Ritz pairs or even Ritz triplets. We aim at comparing the exact (a priori unknown) quantities with their (a posteriori known) approximations. Then we switch to the (Q)OR methods and briefly consider (Q)MR methods.

All insights are based on the general and very simple idea to look for quantities that are supposed to become small in infinite precision. Then, in a subsequent step, the appropriate linear combination of equations from the set of equations (4.44) is formed. There are always (at least) two ways to construe the resulting formulae. The first way is similar to the approach used in Section 4.8 and exhibits the impacts of a *convergence* onto the *deviation* of the recurrence of the basis vectors. The second way is of interest in the context of attainable accuracy. Here, the quantities are ordered differently. The component measuring the convergence is written as an overlap of a term involving the next basis vector, which is usually precisely the term used as estimator for convergence, and an (amplified) error term.

The set of equations (4.44) applies directly in case we are interested in the eigenvalues of the matrix $A$. In the beginning, when no deviation occurred, it makes perfect sense to measure the terms $(\lambda_i - \theta_j)\hat{v}_i^H y_j$. In infinite precision these terms will converge to zero. We depict the case that the Ritz pair $(\theta_j, y_j)$ is chosen as the closest one in backward sense to the (left) eigen-pair among all possible Ritz pairs. Then the deviation of the *computed* basis vector $q_{k+1}$ from the *exact* one is described by

$$\hat{v}_i^H q_{k+1} = \frac{(\lambda_i - \theta_j)\hat{v}_i^H y_j + \hat{v}_i^H F_k s_j}{c_{k+1,k} s_{kj}}. \tag{4.61}$$

Suppose that no substantial deviation occurred. Then at least one of the terms $(\lambda_i - \theta_j)\hat{v}_i^H y_j$ will converge. The behaviour of the recurrence is dominated by the error terms $\hat{v}_i^H F_k s_j$, when

$$|(\lambda_i - \theta_j)\hat{v}_i^H y_j| \approx |\hat{v}_i^H F_k s_j|.$$

This will usually spoil the convergence. The best we can hope for, i.e. the *least attainable accuracy* is given by the size of $|\hat{v}_i^H F_k s_j|$.

To give a grasp of the intimate connections between convergence and deviation we give two pictures obtained by a finite precision run of the symmetric Lanczos method. The matrix $A$ is a Poisson matrix of order 100, i.e. the discrete Laplace operator on the unit square. The starting vector was created by computing a random vector and modifying the lower bits such that resulting vector upon multiplication by 100 had only small integer components. The finite precision run was on an IBM RS 6000 machine using Matlab 5.3 with IEEE double precision. Machine precision was $2^{-53} \approx 1.1102 \cdot 10^{-16}$. A second run using Maple V 5.1 was done in exact arithmetic with the same matrix and the same starting vector. The resulting quantities, i.e. the tridiagonal matrix $T$ and the basis $Q$ were transferred to Matlab by rounding them, such that the *relative* error in the entries is at the level of the machine precision.

The first picture, given by figure 4.3, plots on the right-hand side the upper triangular part of the matrix $Q_k^H Q_k - I_k$ in semilogarithmical scale. This plot clearly depicts the occurring loss of orthogonality. The left-hand side plots for the same run the magnitudes of all residual estimators of all steps in semilogarithmical scale. In step $k$ we have $k$ residual estimators, thus, also this matrix is triangular. The residual estimators are plotted sorted according to the *values* of the corresponding real eigenvalues.

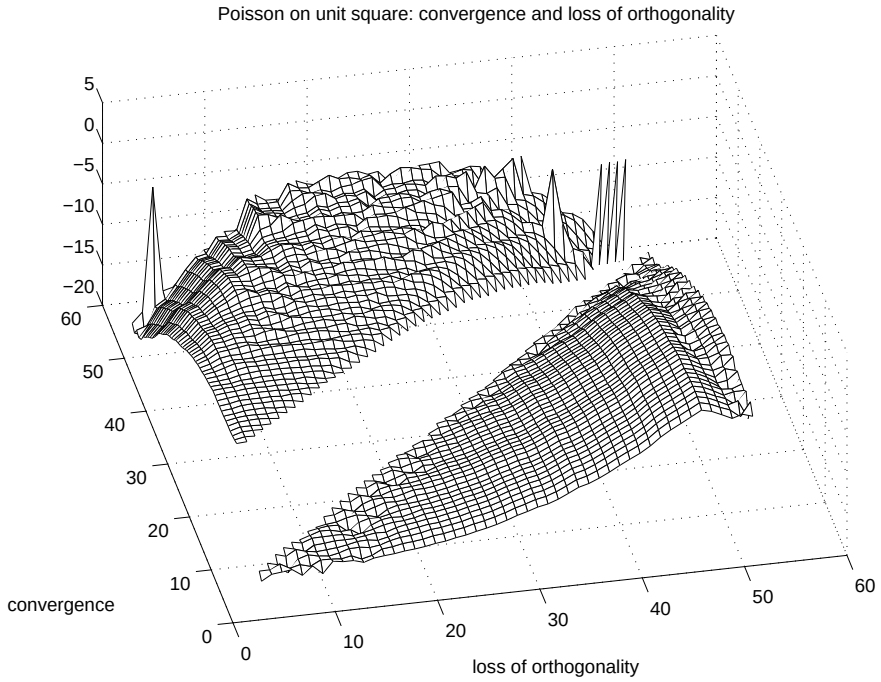


Figure 4.3: Symmetric Lanczos, loss versus convergence (I)

In the second picture, given by figure 4.4, the left-hand side plots the difference between the in finite precision computed (Matlab) and exact (Maple) quantities. The right-hand side of the plot is chosen like in the preceeding example. We observe that the computed residual estimators start to differ from the exact ones *a while after* the loss of orthogonality started. To be more precise, the deviation in the residual estimators started when the first cluster of Ritz values appears. This, apparently, is not obvious from the picture, but follows upon application of Paige's analysis.

The pictures suggest that the deviation in the matrix $C_k$ is a second order effect and the loss might be monitored by knowledge of the infinite precision behaviour. We give an outline of some ideas in that direction.

In exact arithmetic, the Ritz vector will converge to (a multiple of) the right eigenvector $v_i$ of $A$. To be more precise, when the Krylov methods runs to completion and the vector $\hat{v}_i^H Q_m$ is a left eigenvector of $C_m$,

$$\hat{v}_i^H A Q_m \;=\; \lambda_i \hat{v}_i^H Q_m \;=\; \hat{v}_i^H Q_m C_m.$$

This implies that in exact arithmetic

$$\frac{\left(\lambda_i - \theta_j^{(k)}\right) \hat{v}_i^H Q_k s_j}{c_{k+1,k} s_{kj}^{(k)}} = \frac{\left(\theta_j^{(m)} - \theta_j^{(k)}\right) \left(\hat{s}_j^{(m)}\right)^H \begin{pmatrix} s_j^{(k)} \\ 0 \end{pmatrix}}{c_{k+1,k} s_{kj}^{(k)}} = \check{s}_{k+1,j}^{(m)}$$

holds true. Neglecting the influences of the errors in prior steps, a deviation occurs when

$$\check{s}_{k+1,j}^{(m)} \approx \frac{\hat{v}_i^H F_k s_j}{c_{k+1,k} s_{kj}^{(k)}}.$$

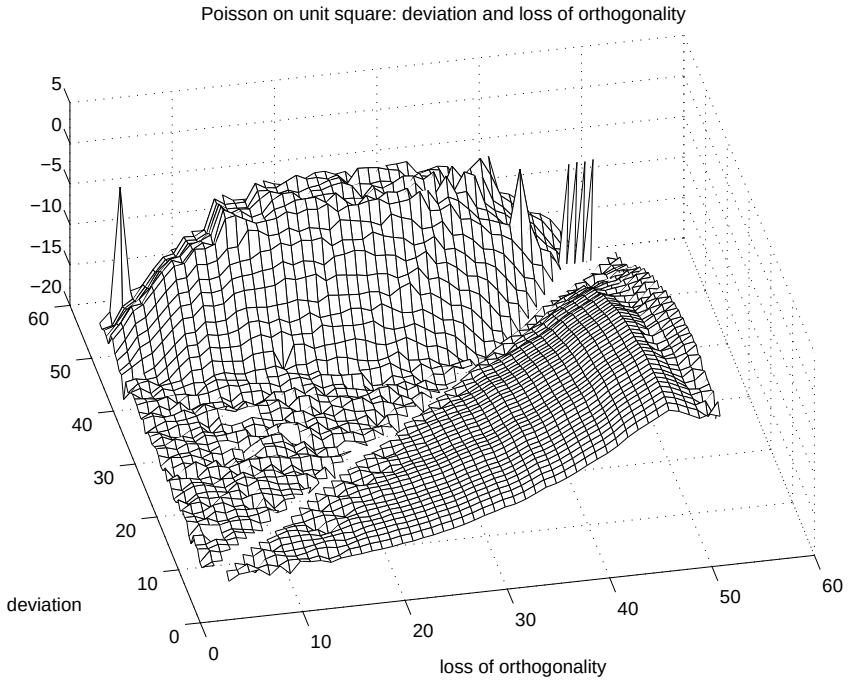


Figure 4.4: Symmetric Lanczos, exact versus floating point

This is of interest in a theoretical forward error approach, when we know the exact left and right eigenvectors of $C_k$ for all steps $k$. In this form, it does not tell anything about the attainable accuracy of Ritz values. Results in that direction are based on another observation. We can choose $\hat{v}_i$ and $v_i$ such that the sequence of Ritz vectors $y_j$ converges to $v_i$ and furthermore

$$\frac{\|s_j\|}{|\hat{v}_i^H y_j|} \to \frac{\|v_i\|}{|\hat{v}_i^H v_i|}$$

holds true. This is due to the fact that in case of successful termination of the infinite precision Krylov method the computed Ritz vectors *are* eigenvectors. Suppose that in step $k$ the index $j = j(i)$ is chosen to ensure that the sequence

$$\theta_j^{(k)} \to \theta_j^{(m)} = \lambda_i$$

converges to $\lambda_i$. Next, suppose that the first $k$ components of the *final* left eigenvector $\hat{s}_j^{(m)}$ of $C_m$ are close to the $k$th left eigenvector $s_j^{(k)}$ of $C_k$. Putting in another way, the Ritz *value* $\theta_j$ can come as close to an eigenvalue $\lambda_i$, as suggested by

$$|\lambda_i - \theta_j| \gtrapprox \frac{\left|\hat{v}_i^H F_k s_j\right|}{\left|\hat{v}_i^H y_j\right|} \to \frac{\|\hat{v}_i^H\|\|F_m\|\|v_i\|}{|\hat{v}_i^H v_i|} = \kappa(\lambda_i)\|F_m\|.$$

This is essentially the backward error when the matrix $A$ is perturbed normwise by $\|F_m\|$. Since we have control over the size of $F_m$, this is the best result we could hope to achieve. When we look at the case of an early termination, i.e. the estimated residual implies a backward error small enough to consider the Ritz pair as close approximation to an eigenpair, the norm $\|F_m\|$ my be replaced for any consistent norm by $\|F_k\|$. To have a rather crude estimate of the attainable accuracy, we can

use the condition of the computed Ritz value,

$$|\lambda_i - \theta_j| \gtrapprox \kappa(\theta_j)\|F_k\|,$$

or as a better estimate,

$$|\lambda_i - \theta_j| \gtrapprox \frac{\|\hat{s}_j^H\| \left(\sum_{l=1}^{k} \|f_l\||s_{lj}|\right)}{|\hat{s}_j^H s_j|} \geq \frac{\|\hat{s}_j^H\|\|F_k s_j\|}{|\hat{s}_j^H s_j|}.$$

These estimates do only work as long as the basis vectors have not deviated too much from the exact ones. This is the case when no significant convergence has taken place. Therefore, the estimation works at least for one Ritz value, to be precise, for the fastest converging one. When the eigenvalue $\lambda$ is well separated from the other eigenvalues and has a small condition number, the convergence of the whole process to other eigenvalues is not spoiled, only delayed, and similar conclusions can be drawn for the second Ritz value.

The *bounds* obtained stem from eigenvector components. In most cases we have no easily accessible information on the *real* accuracy of the eigenvalues at hand. We observe that a *measurable* convergence of Ritz values is closely connected to the *convergence* of the corresponding right Ritz vectors. Perturbation theory can be used to gain more information whenever information on gaps in the spectrum and the conditioning of eigenspaces is at hand, for instance when the matrices are symmetric and we apply Temple-Kato bounds.

For simple Krylov subspace methods, the *natural measure* of convergence of a Ritz pair is given by the distance between the eigenvalue and the Ritz value times the scalar product of *left eigenvector* and *right Ritz vector*. Because of its close relation with the backward error, a natural *estimator* of convergence of a Ritz pair is given by the size of the estimated residual, i.e. the last component of the eigenvector corresponding to the Ritz pair we investigate.

To treat a more general case, instead of diagonalising the governing equation (4.3) to obtain the set of equations (4.44), we block-diagonalise equation (4.3) to obtain the set

$$\hat{v}_i^H q_{k+1} = \left(J_{\lambda_i}\hat{v}_i^H Q_k S_j - \hat{v}_i^H Q_k S_j J_{\theta_j} + \hat{v}_i^H F_k S_j\right)\left(e_k^T S_j\right)^{-1}(c_{k+1,k})^{-1} \qquad (4.62)$$

of equations. We replace the absolute values of the quantities occurring in the simple Krylov subspace method case by the *singular values* of the blocks occurring in the block Krylov subspace methods. Again we can relate convergence to a deviation. A more thorough treatment of block Krylov subspaces will be part of future research.

Similar to the approach used in Section 4.8, we alternatively can describe the deviation in terms of the left eigenvector components of successive solution vectors $z_k$, $\underline{z}_k$. We first consider the methods that are based on the (Q)OR approach. We assume for the moment that the matrix $C_k$ is diagonalisable. The solution vector $z_k$ can be described by the inverse of $C_k$ applied to a multiple of the first unit vector $e_1$ of appropriate length,

$$\frac{z_k}{\|r_0\|} = C_k^{-1}e_1 = S_k\Theta_k^{-1}\check{S}_k^T e_1 = \sum_{j=1}^{k}\frac{\check{s}_{1j}}{\theta_j}s_j.$$

Using this representation, we obtain a new expression for the estimated residual, since the (pre-multiplied) last component of this vector is given by

$$\begin{aligned}\frac{c_{k+1,k}z_{kk}}{\|r_0\|} &= c_{k+1,k}e_k^T C_k^{-1}e_1 = c_{k+1,k}e_k^T S_k\Theta_k^{-1}\check{S}_k^T e_1 \\ &= c_{k+1,k}\sum_{j=1}^{k}\frac{\check{s}_{1j}s_{kj}}{\theta_j} = \sum_{j=1}^{k}\frac{\prod_{l=1}^{k}c_{l+1,l}}{\theta_j\chi'_{C_k}(\theta_j)}.\end{aligned}$$

We proceed by forming linear combinations of the equations from the set (4.44), for convenience stated once again explicitely:

$$\hat{v}_i^H q_{k+1} = \hat{v}_i^H Q_k \left[\frac{\lambda_i - \theta_j}{c_{k+1,k} s_{kj}}\right] s_j + \hat{v}_i^H F_k \left[\frac{1}{c_{k+1,k} s_{kj}}\right] s_j.$$

In context of the (Q)OR methods, we use the identity

$$\frac{z_k}{\|r_0\|} = \sum_{j=1}^{k} \frac{\check{s}_{1j}}{\theta_j} s_j = \sum_{j=1}^{k} \left(\frac{c_{k+1,k}\check{s}_{1j} s_{kj}}{(\lambda_i - \theta_j)\,\theta_j}\right)\left(\frac{\lambda_i - \theta_j}{c_{k+1,k} s_{kj}}\right) s_j$$

to choose the coefficients of the linear combination. This results in the following theorem:

**Theorem 4.46** *Let $A \in \mathbb{K}^{n\times n}$. We are interested in the solution $x = A^{-1}b$ of the linear system $Ax = b$. Suppose a finite precision Krylov method with starting vector $q = q_1 \equiv b/\|r_0\|$ resulted in the perturbed Krylov decomposition*

$$AQ_k - Q_k C_k = M_k - F_k,$$

*where $Q_k$, $F_k \in \mathbb{K}^{n\times k}$, $C_k \in \mathcal{H}(k)$ and $M_k$ is given by $M_k \equiv c_{k+1,k} q_{k+1} e_k^T \in \mathbb{K}^{n\times k}$. Suppose further that $A$ and $C_k$ are diagonalisable. Let $\hat{v}_i^H$ be a left eigenvector of $A$ to eigenvalue $\lambda_i$. Let $\check{s}_j^T$, $s_j$ and $\theta_j$ denote the $j$th left eigenvector, right eigenvector and eigenvalue of $C_k$, respectively. We assume that an (Q)OR solution exists, i.e. that $C_k$ is non-singular. Let $z_k = C_k^{-1} e_1 \|r_0\|$ and $x_k = Q_k z_k$.*

*Then the recurrence of the basis vectors can be described as a mixture of eigencomponents of the solution in step $k$ and amplified errors,*

$$\left[\sum_{j=1}^{k} \frac{c_{k+1,k}\check{s}_{1j} s_{kj}}{(\lambda_i - \theta_j)\,\theta_j}\right] \hat{v}_i^H q_{k+1} = \frac{\hat{v}_i^H x_k}{\|r_0\|} + \hat{v}_i^H F_k \left[\sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{(\lambda_i - \theta_j)\,\theta_j}\right) s_j\right]$$

*We observe that small eigenparts in the solution $x \equiv A^{-1}b$ will not be computed accurately whenever they are less in magnitude than the error terms on the right-hand side.*

The representation of the perturbed process can be better understood in terms of interpolating functions. By the transformation of the first term using the eigenvector – eigenvalue relations, we arrive at

$$\sum_{j=1}^{k} \frac{c_{k+1,k}\check{s}_{1j} s_{kj}}{(\lambda_i - \theta_j)\theta_j} = \sum_{j=1}^{k} \frac{\prod_{l=1}^{k} c_{l+1,l}}{\chi'_{C_k}(\theta_j)(\lambda_i - \theta_j)\,\theta_j}.$$

This is nothing but interpolation of the function $x^{-1}$ at knots $\theta_j$, $j \in \underline{k}$. As in the last section, when all eigenvalues are real it is possible to express the interpolation error in terms of *derivatives*. We proceed similarly for the components $w_{lk}$ of the $k$th error amplification vector,

$$w_k = \sum_{j=1}^{k} \frac{\check{s}_{1j}}{(\lambda_i - \theta_j)\,\theta_j} s_j.$$

The eigenvector – eigenvalue relations assure that

$$w_{lk} = \sum_{j=1}^{k} \frac{\check{s}_{1j} s_{lj}}{(\lambda_i - \theta_j)\,\theta_j} = \sum_{j=1}^{k} \frac{\prod c_{p+1,p} \chi_{C_{l+1:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)(\lambda_i - \theta_j)\,\theta_j}$$

holds true. Again we obtain interpolating functions for the inverse, this time weighted by *trailing characteristic polynomials*.

To better understand the impacts of a convergence towards the true solution, it is more convenient to express the *error* in terms of the eigenbasis suggested by the computed $C_k$. The error in direction $\hat{v}_i^H$ is given by

$$\begin{aligned}\hat{v}_i^H(x - x_k) &= \hat{v}_i^H A^{-1} b - \|r_0\| \hat{v}_i^H Q_k z_k \\ &= \|r_0\| \hat{v}_i^H Q_k \left(\lambda_i^{-1} I_k - C_k^{-1}\right) e_1\end{aligned}$$

At this point we insert the representation in terms of the eigenvectors $s_j$,

$$\begin{aligned}\left(\lambda_i^{-1} I_k - C_k^{-1}\right) e_1 &= \left(S \lambda_i^{-1} \check{S}^T - S \Theta_k^{-1} \check{S}^T\right) e_1 \\ &= \sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{\lambda_i} - \frac{\check{s}_{1j}}{\theta_j}\right) s_j.\end{aligned}$$

This proves that the error, split into eigencomponents, can be written as

$$\begin{aligned}\frac{\hat{v}_i^H(x - x_k)}{\|r_0\|} &= \hat{v}_i^H Q_k \left(\sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{\lambda_i} - \frac{\check{s}_{1j}}{\theta_j}\right) s_j\right) \\ &= \hat{v}_i^H Q_k \left(\sum_{j=1}^{k} \left(\frac{c_{k+1,k} \check{s}_{1j} s_{kj}}{(0 - \theta_j)\lambda_i}\right) \left(\frac{\lambda_i - \theta_j}{c_{k+1,k} s_{kj}}\right) s_j\right).\end{aligned}$$

We use the (rather strange) notation $0 - \theta_j$ to simplify the understanding of subsequent steps. The last line holds true, since by algebraic manipulation

$$\frac{1}{\lambda_i} - \frac{1}{\theta_j} = \frac{\lambda_i - \theta_j}{(0 - \theta_j)\lambda_i}$$

holds true. We insert this expression into the set of equations (4.44) to obtain the following theorem:

**Theorem 4.47** *Let $A \in \mathbb{K}^{n \times n}$. We are interested in the solution $x = A^{-1}b$ of the linear system $Ax = b$. Suppose a finite precision Krylov method with starting vector $q = q_1 \equiv b/\|r_0\|$ resulted in the perturbed Krylov decomposition*

$$AQ_k - Q_k C_k = M_k - F_k,$$

*where $Q_k$, $F_k \in \mathbb{K}^{n \times k}$, $C_k \in \mathcal{H}(k)$ and $M_k$ is given by $M_k \equiv c_{k+1,k} q_{k+1} e_k^T \in \mathbb{K}^{n \times k}$. Suppose further that $A$ and $C_k$ are diagonalisable. Let $\hat{v}_i^H$ be a left eigenvector of $A$ to eigenvalue $\lambda_i$. Let $\check{s}_j^T$, $s_j$ and $\theta_j$ denote the $j$th left eigenvector, right eigenvector and eigenvalue of $C_k$, respectively. We assume that an (Q)OR solution exists, i.e. that $C_k$ is non-singular. Let $z_k = C_k^{-1} e_1 \|r_0\|$ and $x_k = Q_k z_k$.*

*Then the recurrence of the basis vectors in direction $\hat{v}_i^H$ can be split into a term involving the kth* error $x - x_k$ *and an error term,*

$$\left[\sum_{j=1}^{k} \frac{c_{k+1,k} \check{s}_{1j} s_{kj}}{0 - \theta_j}\right] \frac{\hat{v}_i^H q_{k+1}}{\lambda_i} = \frac{\hat{v}_i^H(x - x_k)}{\|r_0\|} + \frac{\hat{v}_i^H F_k}{\lambda_i} \left[\sum_{j=1}^{k} \left(\frac{\check{s}_{1j}}{0 - \theta_j}\right) s_j\right].$$

*This can be re-written to expose its* dependency on determinants *of trailing submatrices of $C_k$,*

$$\frac{\hat{v}_i^H q_{k+1}}{\lambda_i} = \frac{\chi_{C_k}(0)}{\prod_{p=1}^{k} c_{p+1,p}} \left(\frac{\hat{v}_i^H(x - x_k)}{\|r_0\|}\right) + \sum_{l=1}^{k} \left[\frac{\chi_{C_{l+1:k}}(0)}{\prod_{p=l}^{k} c_{p+1,p}} \left(\frac{\hat{v}_i^H f_l}{\lambda_i}\right)\right].$$

**Proof.** The first equation is the linear combination suggested by the preceeding argumentations. The re-writing is based on the eigenvalue – eigenvector relations. The first term in brackets can be re-written using

$$\sum_{j=1}^{k} \frac{c_{k+1,k}\check{s}_{1j}s_{kj}}{0-\theta_j} = \sum_{j=1}^{k} \frac{\prod_{p=1}^{k} c_{p+1,p}}{\chi'_{C_k}(\theta_j)(0-\theta_j)} = \frac{\prod_{p=1}^{k} c_{p+1,p}}{\chi_{C_k}(0)}.$$

Similarly, we proceed with the error vector components:

$$\sum_{j=1}^{k} \frac{\check{s}_{1j}s_{lj}}{0-\theta_j} = \sum_{j=1}^{k} \frac{\left(\prod_{p=1}^{l-1} c_{p+1,p}\right)\chi_{C_{l+1:k}}(\theta_j)}{\chi'_{C_k}(\theta_j)(0-\theta_j)} = \frac{\left(\prod_{p=1}^{l-1} c_{p+1,p}\right)\chi_{C_{l+1:k}}(0)}{\chi_{C_k}(0)}.$$

This finishes the proof. □

This result comes in a similar flavour as the results in Section 4.8. There, the recurrence was re-written to expose its additive character as overlay of several (exact) methods with wrong normalisation, here, the recurrence is re-written in terms of the actual and estimated error with identical conclusions. We observe that we have an overlap of several methods. Before we go into the details, we sum up the equations to arrive at a vector recurrence:

**Corollary 4.48** *Let $A \in \mathbb{K}^{n\times n}$. We are interested in the solution $x = A^{-1}b$ of the linear system $Ax = b$. Suppose a finite precision Krylov method with starting vector $q = q_1 \equiv b/\|r_0\|$ resulted in the perturbed Krylov decomposition*

$$AQ_k - Q_kC_k = M_k - F_k,$$

*where $Q_k$, $F_k \in \mathbb{K}^{n\times k}$, $C_k \in \mathcal{H}(k)$ and $M_k$ is given by $M_k \equiv c_{k+1,k}q_{k+1}e_k^T \in \mathbb{K}^{n\times k}$. We assume that an (Q)OR solution exists, i.e. that $C_k$ is non-singular. Let $z_k = C_k^{-1}e_1\|r_0\|$, $x_k = Q_kz_k$ and $r_k = b - Ax_k$.*

*Then the recurrence of the basis vectors (pre-multiplied with $A^{-1}$) can be written as*

$$A^{-1}q_{k+1} = \frac{\chi_{C_k}(0)}{\prod_{p=1}^{k} c_{p+1,p}}\left(\frac{x-x_k}{\|r_0\|}\right) + \sum_{l=1}^{k}\left[\frac{\chi_{C_{l+1:k}}(0)}{\prod_{p=l}^{k} c_{p+1,p}}A^{-1}f_l\right].$$

*If we are merely interested in the basis vectors, we obtain a recurrence involving the true residuals $r_k$,*

$$q_{k+1} = \frac{\chi_{C_k}(0)}{\prod_{p=1}^{k} c_{p+1,p}}\left(\frac{r_k}{\|r_0\|}\right) + \sum_{l=1}^{k}\left[\frac{\chi_{C_{l+1:k}}(0)}{\prod_{p=l}^{k} c_{p+1,p}}f_l\right].$$

*This representation can be used to reveal* when *and* why *the error terms dominate the recurrence, based on the size of the true residual.*

**Proof.** In the diagonalisable case, the proof follows by summing the results in the last theorem. The proof for the general case follows by the proof for the next theorem. □

Some comments are in order. The recurrence behaves as if in *every* step a new a Krylov method is invoked and the basis vectors are normalised by considering only the *additive overlay* of the basis vectors of the single methods. In the beginning, the first run, started with the vector $q \equiv q_1$ dominates the recurrence. The other runs are started with amplified error vectors, $f_l/c_l$. Clearly, we have a domination of the recurrence by errors, whenever $c_l$ is considerably small. The other case where

the errors gain influence is when the method is (partially) converging, that is when (parts of) the approximate solution vectors tend to the true solution. In most methods, these two cases coincide.

We re-consider the result from a slightly different point of view. When the terms are sorted differently, we obtain a representation of the computed errors and residuals,

**Theorem 4.49** *Let $A \in \mathbb{K}^{n\times n}$. We are interested in the solution $x = A^{-1}b$ of the linear system $Ax = b$. Suppose a finite precision Krylov method with starting vector $q = q_1 \equiv b/\|r_0\|$ resulted in the perturbed Krylov decomposition*

$$AQ_k - Q_k C_k = M_k - F_k,$$

*where $Q_k$, $F_k \in \mathbb{K}^{n\times k}$, $C_k \in \mathcal{H}(k)$ and $M_k$ is given by $M_k \equiv c_{k+1,k} q_{k+1} e_k^T \in \mathbb{K}^{n\times k}$. We assume that an (Q)OR solution exists, i.e. that $C_k$ is non-singular. Let $z_k = C_k^{-1} e_1 \|r_0\|$, $x_k = Q_k z_k$ and $r_k = b - Ax_k$.*

*Then the kth* true error *$x - x_k$ is composed of two terms, namely*

$$\frac{x - x_k}{\|r_0\|} = \frac{\prod_{p=1}^{k} c_{p+1,p}}{\chi_{C_k}(0)} A^{-1} q_{k+1} - \sum_{l=1}^{k} \left[ \frac{\left(\prod_{p=1}^{l-1} c_{p+1,p}\right) \chi_{C_{l+1:k}}(0)}{\chi_{C_k}(0)} A^{-1} f_l \right].$$

*This implies that the* true residual *$r_k$ can be expressed by*

$$\frac{r_k}{\|r_0\|} = \frac{\prod_{p=1}^{k} c_{p+1,p}}{\chi_{C_k}(0)} q_{k+1} - \sum_{l=1}^{k} \left[ \frac{\left(\prod_{p=1}^{l-1} c_{p+1,p}\right) \chi_{C_{l+1:k}}(0)}{\chi_{C_k}(0)} f_l \right].$$

*Clearly, we can ask* when *and* why *the computed quantities are dominated by errors and the recurrence might be stopped since we already have reached the ultimately attainable accuracy. To complete this task, we have to compare the magnitudes of the vectors*

$$\left(\prod_{l}^{k} c_{p+1,p}\right) q_{k+1} \qquad \textit{and} \qquad \det C_{l+1:k}\, f_l \quad l \in \underline{k}.$$

*This theorem is useful mostly in context of* backward error analysis*, since it is based on the* computed *basis vectors.*

**Proof.** We already have claimed that the results hold true also in the general, non-diagonalisable case. This follows by a direct proof. We multiply the perturbed Krylov decomposition by the scaled solution vector to obtain

$$\frac{Ax_k - b}{\|r_0\|} = c_{k+1,k} q_{k+1} e_k^T C_k^{-1} e_1 - \sum_{l=1}^{k} e_l^T C_k^{-1} e_1 f_l. \tag{4.63}$$

The definition of the classical adjoint ensures that the entries of interest of the inverse of the Hessenberg matrix $C_k$ are given by

$$\begin{aligned} -e_l^T C_k^{-1} e_1 = e_l^T (-C_k)^{-1} e_1 &= \frac{e_l^T \operatorname{adj}(-C_k) e_1}{\det(-C_k)} \\ &= \frac{\left(\prod_{p=1}^{l-1} c_{p+1,p}\right) \chi_{C_{l+1:k}}(0)}{\chi_{C_k}(0)}. \end{aligned}$$

Inserted into equation (4.63), this is precisely the representation given in the theorem. □

When the matrices $C_k$ become singular, the size of the residuals grows. This seems to be no problem, since both the part depending on the next basis vector, as well as the error vectors are amplified. But, the size of the error vectors in many methods depends *heavily* on the size of the residuals, e.g. Orthores. So, indeed, nearly singular $C_k$ are almost surely bad news.

This direct proof is also possible for (Q)MR solutions, but this time we have no such nice representation of the elements of the pseudo-inverse:

**Lemma 4.50** *Let $A \in \mathbb{K}^{n\times n}$. We are interested in the solution $x = A^{-1}b$ of the linear system $Ax = b$. Suppose a finite precision Krylov method with starting vector $q = q_1 \equiv b/\|r_0\|$ resulted in the perturbed Krylov decomposition*

$$AQ_k - Q_k C_k = M_k - F_k,$$

*where $Q_k$, $F_k \in \mathbb{K}^{n\times k}$, $C_k \in \mathcal{H}(k)$ and $M_k$ is given by $M_k \equiv c_{k+1,k} q_{k+1} e_k^T \in \mathbb{K}^{n\times k}$. Let $\underline{z}_k = \underline{C}_k^\dagger e_1 \|r_0\|$, $\underline{x}_k = Q_k \underline{z}_k$ and $\underline{r}_k = b - A\underline{x}_k$.*

*Then the true* error *$x - \underline{x}_k$ can be expressed as composed of the terms*

$$\frac{\underline{x}_k - x}{\|r_0\|} = c_{k+1,k} A^{-1} q_{k+1} e_k^T \underline{C}_k^\dagger e_1 - \sum_{l=1}^{k} e_l^T \underline{C}_k^\dagger e_1 A^{-1} f_l.$$

*Similarly, the true* residual *can be expressed as*

$$\frac{-\underline{r}_k}{\|r_0\|} = c_{k+1,k} q_{k+1} e_k^T \underline{C}_k^\dagger e_1 - \sum_{l=1}^{k} e_l^T \underline{C}_k^\dagger e_1 f_l.$$

To summarise the results obtained in this section, there is a certain measure of *convergence* for every method and a measure of the deviation, like the loss of orthogonality or bi-orthogonality. Moreover, when the right way of measurement has been chosen, the convergence and the deviation are reciprocals of each other.

## 4.11 Re-Orthogonalisation Techniques

The previous sections have shown that in general we can not hope to compute bases that reflect the infinite precision behaviour. The most obvious remedy is to use the methods with full re-orthogonalisation or some other form of re-inforcing the lost infinite precision structure. This was proposed soon after the methods itself had been published in the early years by Lanczos and Wilkinson. This has one major drawback, they are no longer competitive with the direct methods developed roughly the same time.

It is less well-known that this approach does *not reconstitute* the infinite precision behaviour. Consider as an example a matrix $A$ with (geometrically) multiple eigenvalues. The re-orthogonalisation, for instance in the method of Arnoldi, can not prevent any deviation orthogonal to the converging Ritz vector. When we make the natural assumption that errors occur in direction of the remaining eigenspace, we observe that the method computes a small cluster of Ritz values located nearby the multiple eigenvalue. At the same time, the existence of non-zero error vectors will cause the Ritz values to move in small circles around the exact eigenvalues. Because of the representation of the eigenvector components as polynomials in Ritz values, this implies a large error in the *relative accuracy* of small eigenvector components. We feel that this point is worth mentioning, since many authors use the methods with (multiple) re-orthogonalisation as a model for the (unknown) infinite precision behaviour.

Nevertheless, the re-orthogonalisation can ensure *backward stability*. The preceeding argumentations actually explain *when* the original matrix has to be perturbed, and if so, *how* the perturbation affects the *number* of Ritz values in a cluster and *how to measure* the accuracy of the computed Ritz values depending on the step.

The analysis of how orthogonality is lost enables another approach. Based on the error analysis of the last sections, Scott, Grcar, Simon and Bai developed methods subsumed as *semiorthogonalisation techniques*. At the time of writing the techniques apply only to Lanczos methods and are known as LanPR, LanSO, LanPRO and ABLE. In infinite precision the methods reduce to classical Krylov methods. In finite precision they enforce a level of linear independency sufficient to achieve backward stability. The methods are built upon the error analysis of the preceeding sections. Matrix perturbation theory is used to determine computable conditions how long the computed matrices are close to orthogonal or oblique projections. Whenever such a condition is violated, the basis vectors are (partially) re-orthogonalised.

The first method, Lanczos with selective orthogonalisation, LanSO for short, was developed by Scott and Parlett. This method applies to the symmetric Lanczos algorithm and is based on Paige's observation that the loss of orthogonality is entirely in direction of the converging Ritz vector. Based on heuristic arguments, the next basis vector is orthogonalised against all Ritz vectors that have converged in the sense of $|\beta_k s_{kj}| \leq \sqrt{\epsilon}$, where $\epsilon$ denotes the machine precision.

The next method was developed by Grcar, also for the symmetric Lanczos method. His method is known as Lanczos with periodic reorthogonalisation, LanPR for short. He gives a sophisticated scheme that models the loss of orthogonality by keeping track of a vector of length $n$. When one of the components of this vector exceeds the square root of the machine precision, a complete re-orthogonalisation against all previously constructed basis vectors is invoked.

Cheaper and more elegant is the method developed by Simon. Simon was the first who observed that the computed matrix $T_k$ is close to an exact projection of the matrix $A$ as long as the level of linear dependency, defined by

$$\|\hat{Q}_k^H Q_k - I_k\| \equiv \text{ level of linear dependency}$$

does not exceed the square root of the machine precision. His ideas where generalised by Day to the non-symmetric Lanczos method. The level of linear dependency can be measured by Gerschgorins circle theorem or M-Matrix, to be more precise, using H-Matrix theory on the matrix $W_k = \hat{Q}^H Q_k$ of the loss of orthogonality. These mathematical tools of trade are used to derive conditions how long $W_k$ admits Cholesky, LDLT, LDMT decomposition. Simon was also the one who recognised that LanPR and LanSO were just instances of such methods.

These approaches are 'as is' very cumbersome to apply to methods for the solution of linear systems, especially to the LTPMs. Thus far, nobody considered the application to truncated methods relying on Hessenberg structure, mostly methods based on the Arnoldi recurrence. Block methods have only been considered by Bai et.al. in the package ABLE.

The last idea that does not seem to fit into the scheme 're-orthogonalisation techniques' is to use the methods 'as is', i.e. with no re-orthogonalisation at all. Only the stopping criteria and the picking of information from the computed matrices is adapted to the finite precision behaviour. Methods of this class have been developed and implemented by Cullum and Willoughby, but the algorithms are restricted to the Lanczos variants. These methods are based on picking eigenvalues and labelling them as 'true' or 'spurious' which can be considered as a form of semi-orthogonalisation by hand. It is remarkable that their approach of identifying

spurious eigenvalues is based on a comparison of the eigenvalues of $C_k$ and those of $C_{2:k}$. An eigenvalue is labelled spurious, when it is an eigenvalue of *both* matrices. By the preceeding error analysis, this is always the case, when the deviation is such that the amplified errors result in Ritz values. What might be considered a major drawback is that eigenvalues, whose corresponding eigencomponents in the starting vector are below a certain threshold, are also labelled spurious. This is, to some extent, justified, since the resulting Ritz vector consists mainly of amplified errors.

# Chapter 5

# Krylov Subspace Methods in Finite Precision

In our derivation of the well-known Krylov methods as well in the proofs concerned with statements on their behaviour the implicitly defined relations, like, e.g. the orthogonality of the basis vectors, shows up to be very important. Chapter 4 reveals that these relations usually will not hold in a finite precision environment. The results of Chapter 4 also imply that the methods will cease to converge at least when the error matrix $F_k$ has grown so large that the backward error of the approximate solution is at about the same size.

Chapter 4 provides hints how to *measure stability* and how to *ensure accuracy*. In the following we will use this knowledge to broaden and to unify the understanding of the Krylov methods introduced in Chapter 3. We mainly give hints towards existing literature on the subject and summarize the main results. We give some numerical experiments to show up characteristic behaviour. What is typical in numerical experiments is that a Krylov method deteriorates *precisely* when the backward error becomes small.

## 5.1 Krylov Methods for the Eigenproblem

We distinguish between short-term and long-term methods and divide them further into singular, orthogonal and bi-orthogonal methods. In *long-term* methods (Arnoldi based) the orthogonality is typically enforced by explicit calculation. A cure for the loss of orthogonality is, due to the accessibility of all computed basis vectors, complete re-orthogonalisation. In *short-term* methods (Lanczos based) typically several copies of Ritz values are computed that correspond to single eigenvalues of $A$. Every time a Ritz pair converges, we observe a deterioration of the process. This process essentially spoils the ultimately attainable accuracy and slows down the convergence of the unconverged eigenvalues. Yet, due to the smaller amount of storage and operations these methods are still competitive with the *long-term* methods. The *dual* short-term methods suffer from near-breakdowns that cause the error matrix $F_k$ to grow unboundedly and thus may prevent accurate solutions. We stress the fact that up to now *no a-priori choice* for the off-diagonal elements of the unsymmetric $T_k$ is known that delivers optimal results.

This is the trade-off between the three main approaches, i.e., the *generally* applicable Arnoldi method using an *orthogonal* projection and *long-term* recurrences, the (with some restrictions) *generally* applicable dual Lanczos method using *oblique* projections and *short-term* recurrences, and the intersection of both, the symmetric or Hermitian Lanczos method that applies *only* to symmetric or Hermitian $A$. This

trade-off is best explained with the aid of a little picture:

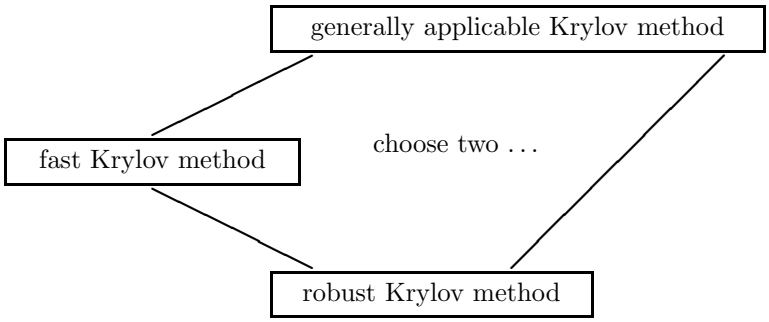


Figure 5.1: Trade-off between *fast*, *robust* and *general applicable*

In the sequel, we will often use one of the main results of the last section, namely that the basis vectors are constructed by

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}}\, q + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(A)}{\prod_{p=l}^{k} c_{p+1,p}}\, f_l \right],$$

where $\chi_{C_k}$ denotes the characteristic polynomial of the Hessenberg matrix computed by the method in use.

### 5.1.1 The Power Method

The power method might be considered one of the 'simplest' Krylov methods. Thus, we might suspect that the error analysis is also very simple. Over more, the finite precision power method paves way for the understanding of the more complicated looking results of the other Krylov methods. The first analysis of the finite precision power method is included in the the textbook by Wilkinson (cf. [Wil65]).

The finite precision power method can be interpreted locally as if we apply the *exact* power method to a *perturbed starting vector*. We know that the characteristic polynomials of the Hessenberg matrices constructed in the power method are monomials:

$$\begin{aligned} q_{k+1} &= \frac{A^k}{\prod_{p=1}^{k} c_{p+1,p}}\, q + \sum_{l=1}^{k} \left[ \frac{A^{k-l}}{\prod_{p=l}^{k} c_{p+1,p}}\, f_l \right] \\ &= \frac{A^k}{\prod_{p=1}^{k} c_{p+1,p}} \left( q + \sum_{l=1}^{k} \left[ \left( \prod_{p=1}^{l-1} c_{p+1,p} \right) A^{-l} f_l \right] \right). \end{aligned}$$

Some of the components in the 'new' starting vector in direction of smaller eigenvalues are amplified such that the recurrence is slowed down. In any case, the constructed basis vectors $q_j$ can not easily be related to a *single* perturbed matrix $A$ and a *single* perturbed starting vector. This implies that the basis vectors $q_j$ are not computed in a backward stable manner in the finite precision power method.

Nevertheless, in case of a single, well separated eigenvalue of maximal modulus, the recurrence is more likely to converge to the unique eigenvector in finite precision. In case of a multiple or clustered eigenvalue of maximal modulus the recurrence will also work, problems occur when there are distinct eigenvalues having approximately the same (maximal) modulus. In infinite precision in this case we usually look for a *polynomial of small degree* in $A$ instead of the *linear polynomial* $A - \theta_j I$ that

maps the constructed vector to the zero vector, see the example of Wilkinson for complex conjugate eigenvalues of real matrices (cf. [Wil65], chapter 9, section 12). Depending on the degree of the polynomial we are looking for and the condition of the eigenvalues and the normality of $A$ the estimated polynomials may vary greatly in the finite precision case.

The power method frequently is used with *deflation* to compute several eigenvalues with descending moduli. It is well-known that in *infinite* precision deflation by erasing the component in direction of the *exact* eigenvector is sufficient, whereas in *finite* precision we have to erase the corresponding component several times because the error vectors bring in a non-zero component and the method will otherwise result in computing the *dominant* eigenvalue, i.e., the eigenvalue of maximal modulus *twice*. When the dominant eigenvalue is *defective*, convergence even in *infinite* precision is slow.

In the context of large sparse matrices mostly *Wielandt deflation* is used. Wielandt deflation changes the original matrix $A$ by a rank-one update,

$$A \leftarrow A - \lambda v y^H, \qquad \text{where} \quad y^H v = 1.$$

The new matrix is only *implicitly* defined and not computed explicitly. The action of the transformed matrix on a vector $q$ can be simulated by computing

$$(A - \lambda v y^H)q = Aq - \lambda y^H q v.$$

The special case $y = \hat{v}$ is known as *Hotelling deflation*. This choice of $y$ makes sense in the normal case, since then $y = \hat{v} = v$ is available for free. Hotelling deflation might not be a good choice. The textbook of Saad contains a section on how to choose the optimal $y$. Wielandt deflation has a natural extension to the case that several eigenvalues have to be deflated. This extension is known as *Schur-Wielandt deflation* because it is based on the (partial) Schur form $AY_k = Y_k R_k$ of $A$. Schur-Wielandt deflation changes the matrix by a rank-$k$ update with $k$ (approximate) Schur vectors $y_j$,

$$A \leftarrow A - Y_k R_k Y_k^H, \qquad \text{where} \quad Y_k^H Y_k = I_k.$$

All these forms of deflation can be incorporated in the error vectors and have the main effect to cause the error matrix $F_k$ to grow.

The general message for the finite precision power method and its epigone, the finite precision inverse iteration with shifts is that the *small parts* in *dominant eigenspaces* are *amplified*. The dominant eigenspaces are what we are looking for. Depending on the point of view, this may be beneficial, i.e., when we have no part of the dominant eigenspace in the starting vector, or not, i.e., when we aim for the eigenvalue of second largest modulus and deflation has to be done (almost) every step.

### 5.1.2 Subspace Iteration

The subspace iteration is just the block variant of the power method, thus nearly the same observations apply. In contrast to the simple variant, deflation is not useful. Instead techniques known as *locking* and *purging* have to be used. Locking just freezes the Schur vectors in the partial Schur form that are considered as being converged. Subspace iteration is known to be very robust. The price one has to pay for this robustness is that subspace iteration is slower than the other Krylov subspaces.

Available packages are LOPSI by Jennings and Stewart dating to 1981 (cf. [SJ81]), EB12 by Duff and Scott dating to 1993 (cf. [DS93, DS95]) and SRRIT

by Bai and Stewart dating to 1997 (cf. [BS97]). All these packages are coded in FORTRAN 77 and have been published in ACM TOMS.

The major effort in subspace iteration is in selecting the appropriate size of the subspace and in a good decision rule how often to re-orthogonalise the vectors. Moreover, a cheap *and* effective *locking* and *purging* strategy has to be used. The codes mentioned above can be viewed as *the* working horses when a stable algorithm for the computation of a few selected eigenvalues and eigenvectors for (non-symmetric) matrices is needed.

Finite precision subspace iteration is considered a stable, but slow method to approximate eigenvalues. Similar to the power method, the finite precision errors are beneficial in 'enriching' the space such that usually *all* eigenvalues of interest can be found. We stress that the convergence rate and the computed eigenvalue approximations only weakly depend on the vectors spanning the actual starting subspace, provided that errors in all directions occur. This seems to be the generic case when we neglect diagonal and block-diagonal matrices $A$.

### 5.1.3 The Arnoldi Method

> "Now it is essential that the $c_i$ should remain strictly orthogonal to working accuracy or there will be no guarantee that $c_{n+1}$ is null to working accuracy. *The difficulty which arises should not be attributed to the cumulative effect of rounding errors.*"
>
> James Hardy Wilkinson,
> The Algebraic Eigenvalue Problem (1965).

We have shown that the Arnoldi method is closely connected to one of the *best-understood* and *very* stable algorithms, namely, the *QR decomposition*. More precise, we have shown in Chapter 3 that the (abstract) Arnoldi method can be obtained from the QR decompositions of the *Krylov matrices* of subsequent steps,

$$[q, AQ_k] = Q_{k+1}R_{k+1}\begin{pmatrix} 1 & 0 \\ 0 & R_k^{-1} \end{pmatrix} = Q_{k+1}\begin{pmatrix} 1 & \\ 0 & \underline{H}_k \end{pmatrix}.$$

This is an *iterated* QR decomposition of the *augmented matrix* $[q, AQ_k]$. This is also the *way of computation* in finite precision. Hence, the matrix is not fully defined in the early stages, but is *computed column by column in the algorithm*. Thus, it is an obvious idea to divide the errors into *first order* and *second order effects*. The error analysis and a monitoring of errors is divided accordingly. This distinction measures in the first stage how much *the basis deviates* and in the second stage whether *the QR decomposition fails* to produce an orthogonal basis $Q_k$. Both, the first as well as the second stage may be empty.

The first stage has to incorporate the expansion of the basis. This basis expansion in step $k$ is given by the perturbed relation

$$X_k + \Delta X_k = AQ_k.$$

The error matrix $\Delta X_k$ is the result of a matrix product and can thus be bounded by

$$\|\Delta X_k\| \leq \gamma_n \|A\| \|Q_k\|, \qquad |\Delta X_k| \leq \gamma_n |A||Q_k|.$$

It remains to understand the *loss of orthogonality*. This loss is due to the QR decomposition.

Despite the fact that in *infinite precision* 'the' Arnoldi method is uniquely defined (up to some signs), there *does* exist a variety of different *implementations* of the Arnoldi method. The behaviour of these implementations differs *extremely*

when executed in finite precision. The menagerie of orthogonalisation schemes available causes a whole bunch of implementations. Among those orthogonalisation schemes we find schemes based on *Householder reflectors*, *Givens rotations*, the generalised *Givens-Kahan rotations*, *CGS* (classical Gram-Schmidt), *MGS* (modified Gram-Schmidt), *IGS*($\ell$) (iterated Gram-Schmidt, $\ell$ times) and as special case of the latter, *DOGS* (double orthogonalisation Gram-Schmidt, i.e., IGS(2)). Iterated Gram-Schmidt can be further divided into iterated *classical* and iterated *modified* Gram-Schmidt. Iterated Gram-Schmidt algorithms, especially the classical variant, are useful when orthogonalisation schemes are implemented on parallel architectures and are currently an area of active research.

This implies that the *main ingredient* of the error analysis of a *finite precision implementation* of the Arnoldi method is the error analysis of the *orthogonalisation scheme* used. The information needed about the Householder, Givens and MGS QR decomposition is easily accessible in modern textbooks on numerical linear algebra (cf. [Hig96]), compare the results presented in Chapter 1. The results of the error analysis of Householder QR are summarised in Lemma 1.13, Lemma 1.15 does the same for Givens QR, and some results on CGS and MGS are collected in Lemma 1.16 and Lemma 1.17.

We need to define error matrices $\Delta Y_k^e$ and $\Delta Y_k$ to measure the accuracy of the QR decomposition used. We use an *e*xactly orthogonal $Q_{k+1}^e$ to define $\Delta Y_k^e$ by

$$Y_k = [q, X_k], \qquad Y_k + \Delta Y_k^e = Q_{k+1}^e R_{k+1}.$$

The computed Q factor $Q_{k+1}$ is used to define $\Delta Y_k$ by

$$Y_k = [q, X_k], \qquad Y_k + \Delta Y_k = Q_{k+1} R_{k+1}.$$

In the Householder variant, the error matrix $\Delta Y_k^e$ is bounded by

$$\|\Delta Y_k^e\|_F \leq (k+1)\gamma_{cn}\|Y_k\|_F, \quad |\Delta Y_k^e| \leq n(k+1)\gamma_{cn}G|Y_k|, \quad \|G\|_F = 1.$$

In the Givens variant, the error matrix $\Delta Y_k^e$ is bounded by

$$\|\Delta Y_k^e\|_F \leq \gamma_{c(n+k+1)}\|Y_k\|_F, \quad |\Delta Y_k^e| \leq n\gamma_{c(n+k+1)}G|Y_k|, \quad \|G\|_F = 1.$$

In the MGS variant, the error matrices $\Delta Y_k^e$ and $\Delta Y_k$ and the loss of orthogonality are bounded by

$$\begin{aligned} \|\Delta Y_k\|_F &\leq c_1\epsilon\|Y_k\|_F, \\ \|Q_k^H Q_k - I\|_2 &\leq c_2\epsilon\kappa_2(Y_k) + O((\epsilon\kappa_2(Y_k))^2) \\ |\Delta Y_k^e| &\leq c_3\epsilon G|Y_k|, \quad \|G\|_F = 1. \end{aligned}$$

In the CGS variant, we can only bound the error matrix $\Delta Y_k$,

$$\|\Delta Y_k\|_F \leq c_4\epsilon\|Y_k\|_F.$$

Here, a term $c$, $c_i$ denotes a small constant independent of $n$. For MGS, explicit constants can be found in the paper by Giraud and Langou (cf. [GL02b]). We can put these pieces together to obtain the perturbed Krylov decomposition

$$AQ_k - Q_{k+1}\underline{H}_k = -F_k.$$

The error matrix $F_k$ can be bounded by adding the bounds for the error matrices $\Delta X_k$, $\Delta Y_k$:

$$\|F_k\| \leq \|\Delta X_k\| + \|\Delta Y_k\|.$$

We stress that in some implementations (especially in the Householder variant), the basis vectors are only implicitly defined and can be 'better' than it would be possible in case the exact columns are known and stored in finite precision.

The MGS variant of Arnoldi has been analysed by Rozložník in his thesis (cf. [Roz97]). It is well known, compare Lemma 1.17, that the computed $Q$ factor will loose orthogonality depending on the condition of the matrix to be factorised. Rozložník showed that the condition of the augmented matrix $[q, AQ_k]$ depends on the condition of $A$ and on the norm of the residual for the minimal solution $\min_z \|q - AQ_k z\|_2$. To be more precise, he showed that as long as this minimum is not too small and the machine precision is small enough,

$$\kappa([q, AQ_k]) \leq \frac{\sqrt{21}\kappa(A)}{\min_z \|q - AQ_k z\|_2}$$

holds true (cf. [Roz97], equation (5.5), page 38). This, in turn, can be used to bound the loss of orthogonality. When we are only interested in the eigenvalues, we need no error analysis of the Q factor, since the analysis shows that the computed R factor, i.e., the matrix $\underline{H}_k$ and thus $H_k$, is close to the R factor constructed by Householder QR.

The *iterated* CGS or MGS variants recently have caught attention by Czech and French researchers. *Two times* iterated Gram-Schmidt is known to be stable when we only have *two* vectors we want to orthogonalise against each other. This result is known as Kahan's "twice is enough" algorithm (cf. [Par98], page 115/116) and is based on the observation that the loss of accuracy is due to a *cancellation* which can only occur *once*. Giraud, Langou and Rozložník proved that this "twice is enough" algorithm is also stable in the case of *several* vectors (cf. [GLR02b]). Based on these results, interesting relations for the singular values of the computed $Q_k$ have been obtained (cf. [GL02b, GL02a, GLR02a]).

A very interesting relation of the singular values of $F_n$ defined by

$$F_n \equiv Q_n - Q_n^e,$$

and the singular values of the matrix $A$ whose columns are to be orthogonalised by MGS was presented by Langou at the Milovy Conference (cf. [Lan02]). When we label the singular values of $A$ in decreasing order with negative indices, i.e., $\sigma_{-1} \leq \cdots \leq \sigma_{-n}$, the relation takes the simple form

$$\sigma_i(F_n)\sigma_{-i}(A) \lessapprox \frac{2c\mathbf{u}}{1 - c\mathbf{u}\kappa(A)}\|A\|_2 \qquad \forall\ i \in \underline{n}.$$

Here, $c = 18.53n^{3/2}$ is a moderate constant and $\mathbf{u}$ denotes the unit roundoff. This shows that the singular values of $Q_n$ deviate *one after the other* from the singular values of the exact unitary $Q_n^e$ *without even interfering* each other.

In the CGS QR decomposition little seems to be known. CGS seems to be too unstable to be incorporated into a finite precision Krylov method based on Arnoldi. Nevertheless, figure 5.2 and figure 5.3 depict that the loss of orthogonality is by no means random, it occurs shortly after the first Ritz value has converged to sufficient accuracy in *any* variant.

We have grouped the orthogonalisation schemes into *three groups*. Those, where we *know* that the orthogonality is not lost (up to machine precision). Those, where we know that a loss occurs, but at the same time we can describe, *how* and in *which direction*. The last group, at the time of writing only the CGS Arnoldi variant, consists of methods where we know that orthogonality is lost, and we do not know *when* and *how*.

The results on the QR decomposition stated above clearly reveal that the Householder and Givens variants of the Arnoldi method will result in approximately orthogonal (unitary) bases $Q_k$. Thus, they are in the first group. This information suffices to show that the eigenvalues of $H_k$ tend to approximate the eigenvalues of $A$ and that the estimated (eigen)residuals are close to the exact (eigen)residuals. What is less known, is that despite the methods deliver accurate results, the *computed space* may be very far away from the exact Krylov subspace defined by the starting vector (or any other starting vector). We will come back to this point later on.

In case of (iterated) MGS and iterated CGS, we have cheap estimators at hand to conclude how long the algorithm will produce accurate results. We already mentioned that orthogonality is lost in (classical) Gram-Schmidt when severe cancellation occurs. In general, there is no easy way to see when this will be the case. When Gram-Schmidt is applied to the augmented matrix $[q, AQ_k]$, we can relate the loss of orthogonality to the convergence of Ritz values. The loss will be in direction of the *left eigenvector*. This knowledge may be used to invent a re-orthogonalisation scheme to be incorporated into finite precision CGS Arnoldi.

We now switch to some explanatory numerical experiments. We have chosen small sized toy problems that reflect enough of the numerical behaviour *and* allow the *explicit* computation of the interesting quantities with backward stable algorithms for dense systems. Essentially the same phenomena show up in case of *large sparse matrices*, but then we can not compute the quantities we want to focus on.

Figure 5.2 and figure 5.3 show the results of a run of finite precision CGS, finite precision MGS and finite precision DOGS Arnoldi. DOGS is an abbreviation for *d*ouble *o*rthogonalisation *G*ram-*S*chmidt. The plots contain three different kinds of curves plotted in a semilogarithmical scale on the $y$ axis. For better readability, three dotted straight lines are plotted additionally at the levels *one*, $\sqrt{\epsilon}$ and $\epsilon$.

In both plots, the matrix $A$ has been constructed from a diagonal matrix $D$ of size 100, with diagonal elements equidistant between zero and one. The normal matrix $A$ used in the first plot was constructed by randomly choosing an unitary similarity transformation $U$ and setting $A = U^H DU$. The non-normal matrix $A$ used in the second plot was constructed by randomly choosing a similarity transformation $X$ and setting $A = X^{-1}DX$. All transformations where computed in Matlab 6.3 with ANSI/IEEE floating point arithmetic, thus the eigenvalues of the resulting matrices are only *close* to the elements of $D$. The computed (non-unitary) transformation $X$ had the approximate condition number $\kappa(X) \approx 1.1244 \cdot 10^3$.

The plots show the convergence to the largest eigenvalue $\lambda_{\max} = 1$. This convergence is measured by the distance to the *two* Ritz values $\theta_{\max}$ and $\theta_{\max -1}$ closest to $\lambda_{\max}$, in the legends of the figures denoted by *real convergence*. These two curves are plotted with *dashed* lines (– –). *This* is the information we are *really* interested in. The information we have *at hand* is the size of the last entry of the eigenvector $s_{\max}$ times the last component $h_{k+1,k}$ of the expanded Hessenberg matrix $\underline{H}_k$. In infinite precision, as well as in the early stages of the finite precision algorithm, this is a bound on the *backward error* of the Ritz pair, i.e., a scaled residual. In the legends, this (hopefully) approximate bound is called the *estimated residual*. The *three curves* resulting from the *three Arnoldi variants* used are scaled by $\||A|\|_2$ and plotted with *solid* lines (—).

In finite precision they describe, together with the unknown errors, the part of the basis vectors in direction of the left eigenvector $\hat{v}_{\max}$ of $A$ to eigenvalue $\lambda_{\max}$,

$$\hat{v}_{\max}^H q_{k+1} = \frac{(\lambda_{\max} - \theta_{\max})\, \hat{v}_{\max}^H y_{\max} + \hat{v}_{\max}^H F_k s_{\max}}{h_{k+1,k} s_{k\,\max}}. \tag{5.1}$$

In the legends the part of the basis vector $q_{k+1}$ in direction of $\hat{v}_{\max}$ is called the *actual eigenpart*. The *three curves* resulting from the *three Arnoldi variants* used

are scaled by $\|\|A\|\|_2$ plotted with *solid dotted* lines (-·-). The starting vector was the same randomly chosen vector for all three runs.

All curves in both figures behave very similar in the *early stages* of *all three* Arnoldi variants used. This is natural, since the error vectors are small in all variants and where not yet amplified by the convergence of a Ritz pair.

By previous considerations, there are circumstances when the parts $\hat{v}_{\max}^H q_{k+1}$ are behaving similar to the (left) eigenvectors of the (exact) final $H_m$ (cf. Chapter 1, Theorem 1.22, page 26, Chapter 2, equation (2.11), page 66, Theorem 2.22, page 70). Furthermore, parts of the eigenvectors of leading submatrices are good approximations of the final eigenvectors, as is obvious by the description of the eigenvectors using polynomial recurrences (cf. Chapter 2, Theorem 2.20, page 70). When $A$ is normal, the left and right eigenvectors coincide, thus we may expect these parts of the basis vectors to decrease with the components of $s_{\max}$. When $A$ is non-normal, the dependencies are not that obvious. These results are used in the sequel to *interpret* the plots and to *highlight* special phenomena.

In the first figure, i.e., figure 5.2, we are in the case of Arnoldi applied to a normal matrix $A$. The convergence of the first Ritz pair takes place in steps 1 to 51. Until step 51, the quantities computed behave almost as if they where computed in infinite precision. The largest Ritz value $\theta_{\max}$ converges to the largest eigenvalue $\lambda_{\max}$ with approximately geometric convergence rate. The estimate $h_{k+1,k}s_{k\max}$ decreases with a rate in size approximately equal to the *square root* of the convergence of $\theta_{\max}$ to $\lambda_{\max}$. This can be interpreted using a Temple-Kato type bound, since the largest eigenvalue is well-separated from the second largest eigenvalue. The size of the part of the next basis vector $q_{k+1}$ in direction of the (left) eigenvector $\hat{v}_{\max} \equiv v_{\max}$ decreases at a similar rate. When we assume that the lower diagonal elements $h_{j+1,j}$ do not become small and the resulting Hessenberg matrix is normal, this can be predicted by perturbation theory. Suppose for simplicity that the conditions of Theorem 1.22 are fulfilled. Then, the leading components of the *final* $\hat{s}_{\max}^{(m)} = \hat{v}_{\max}^H Q_m = s_{\max}$ are given (Chapter 2, Theorem 2.20, page 70) by the polynomial recurrences

$$\frac{\check{s}_{\max}^{(m)}(k)}{\check{s}_{\max}^{(m)}(1)} = \frac{\chi_{H_{k-1}}(\theta_{\max}^{(m)})}{\prod_{l=1}^{k-1} h_{l+1,l}} \quad \forall\, k \in \underline{m},$$

whereas the last components of the intermediate $s_{\max} = s_{\max}^{(k)}$ are given by

$$\frac{\check{s}_{\max}^{(k)}(k)}{\check{s}_{\max}^{(k)}(1)} = \frac{\chi_{H_{k-1}}(\theta_{\max}^{(k)})}{\prod_{l=1}^{k-1} h_{l+1,l}}. \quad \forall\, k \in \underline{m}.$$

For indices, where the leading characteristic polynomial is *insensitive* to small changes in $\theta_{\max}$, the components do not vary greatly. This is especially the case for the *leading* components since then the degree of the polynomial is *small* and there is only *one* root close to the final $\theta_{\max}$. Another difference is the scaling by the first component. In infinite precision, when we consider only one maximal well-separated eigenvalue and $\hat{v}_{\max}^H q_1$ is large enough, the trailing components will be small. Thus, we may expect the scaling to be *negligible*.

The picture changes when we move on in the algorithm. Next, we discuss the behaviour of the three Arnoldi variations under consideration in steps 51 to 74. The Ritz value $\theta_{\max}$ does not move any more, which follows readably from equation (5.1). This is the first difference from the infinite precision Arnoldi method. The other, more important difference is that in the CGS and MGS variants the part of the basis vectors in direction of the (left) eigenvector starts to *increase* with a rate inverse proportional to the rate of $h_{k+1,k}s_{k\max}$. Only the DOGS variant manages to reflect the expected infinite precision behaviour. This describes in detail the

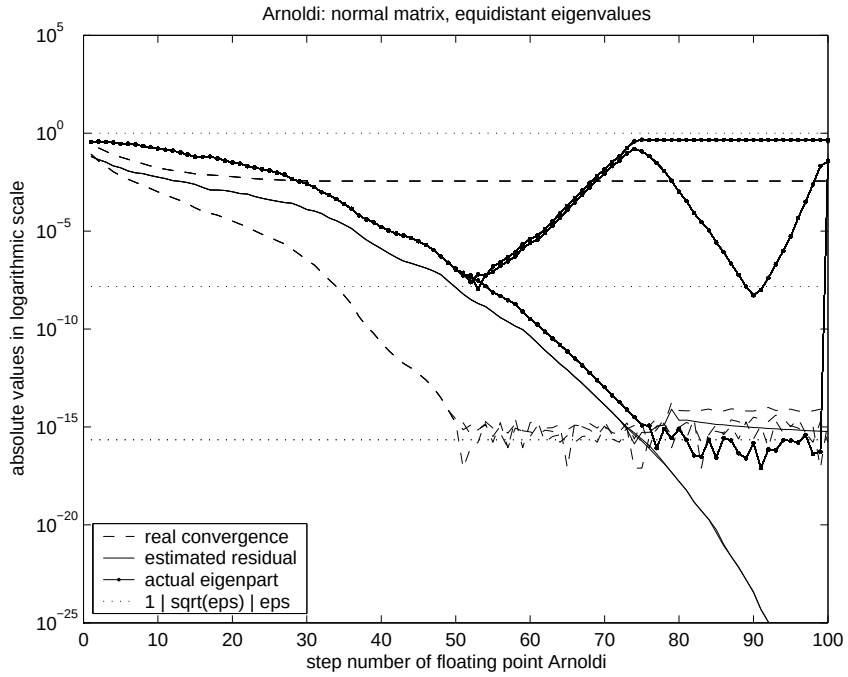


Figure 5.2: Arnoldi, normal matrix, equidistant eigenvalues

occurring *loss of orthogonality*. Nevertheless, the rate of the estimate $h_{k+1,k}s_{k\,\max}$ behaves as if the basis vectors where unperturbed. This is linked to the *normality* of the matrix $A$. The perturbation mostly in direction of *the* eigenvector $v_{\max} = \hat{v}_{\max}$ has only the effect of moving the other Ritz values slightly (this is not visible in the picture).

In steps 74 to 100, the picture changes again. In the CGS variant the estimate $h_{k+1,k}s_{k\,\max}$ moves up in magnitude for a few steps and remains at a level of magnitude approximately equal to $10^{-15}$. The true distance between $\theta_{\max}$ and $\lambda_{\max}$ is even slightly larger, almost $10^{-14}$. The size of the next basis vector $q_{k+1}$ in direction of the eigenvector $\hat{v}_{\max}$ is all the time at the same high *constant level*, only slightly lower than one. This is in contrast to the MGS and the DOGS variants of Arnoldi. Here, the estimates $h_{k+1,k}s_{k\,\max}$ continue to decrease comparable to each other (and to the infinite precision counterparts). The minimal distance of the largest Ritz value $\theta_{\max}$ to the largest eigenvalue $\lambda_{\max}$ moves around the machine precision. This comes as no surprise for DOGS Arnoldi, since the DOGS Arnoldi variant is stable enough to ensure that orthogonality is maintained up to the order of the machine precision. When the Ritz vector is close to the right eigenvector, the same accuracy is attained by keeping the loss of orthogonality restricted, such that $\hat{v}_{\max}^H q_{k+1} h_{k+1,k} s_{k\,\max} \approx \hat{v}_{\max}^H F_k s_{\max}$ holds true. This is true, because the restriction, together with equation (5.1), implies that

$$|\lambda_{\max} - \theta_{\max}||\hat{v}_{\max}^H y_{\max}| \approx 2|\hat{v}_{\max}^H F_k s_{\max}| = O(\epsilon \|A\|_2)$$

holds true. The picture reveals that the MGS Arnoldi variant assures the validity of the restriction for the first converging Ritz pair.

Now, we switch to the second figure, i.e., figure 5.3. Here, we investigate what happens when we apply the three variants CGS, MGS and DOGS of floating point Arnoldi to a *non-normal* matrix. The line styles are all chosen as in the preceeding

figure. That is, we plot the quantities we labelled *real convergence* with *dashed* lines (– –), the quantities we labelled *estimated residuals* with *solid* lines (—) and the quantities we labelled the *actual eigenpart* with *solid dotted* lines (–·–).

In the case of the normal matrix $A$ we successfully identified *three* stages of the finite precision behaviour. In figure 5.3, the first stage is easily identified to take place in steps 1 to 50. At that point, the CGS variant completely looses track. The MGS and DOGS variants do *much* better.

The second and third stage can not be identified with the aid of the picture. The part of the basis vectors in direction of the *left* eigenvector does not decrease, thus we can not see, that the MGS variant alters this part significantly. We observe that the CGS variant clearly perturbs this part, since we obtain an almost *constant solid-dotted* line between steps 50 and 100. Also the in the CGS variant computed *Ritz values* $\theta_{\max}$ are only correct to the order of the square root $\sqrt{\epsilon}$ of the machine precision, resulting in the almost *constant dashed* line. The estimates $h_{k+1,k}s_{k\,\max}$ from the CGS variant are smaller than the distance between $\theta_{\max}$ and $\lambda_{\max}$ and are still decreasing.

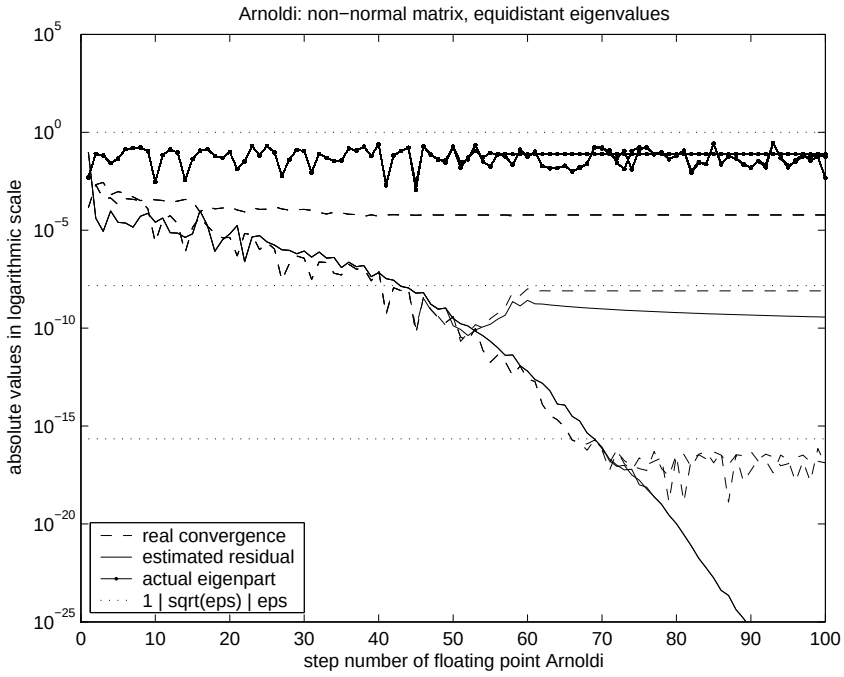


Figure 5.3: Arnoldi, non-normal matrix, equidistant eigenvalues

We give some rather philosophical comments on the behaviour finite precision Arnoldi. We have to distinguish between normal and non-normal matrices, or to be more precise, between normal and non-normal eigenvalues. This will become obvious in a moment. Assume that the *left* eigenvectors $\hat{v}_i$ are normalised, such that $\|\hat{v}_i\|_2 = 1$. The relations

$$\hat{v}_i^H q_{k+1} = \frac{(\lambda_i - \theta_j)\,\hat{v}_i^H y_j + \hat{v}_i^H F_k s_j}{h_{k+1,k} s_{kj}}$$

show that a deviation from the theoretical quantities occurs every time a Ritz pair is converging. Thus, convergence can not exceed a certain limit, since with the usual normalisation of the basis vectors $\|q_j\|_2 \approx 1$,

$$1 \approx \|q_{k+1}\| > |\hat{v}_i^H q_{k+1}|$$

surely is an upper bound. What happens *after* this bound has been 'reached', depends on the normality of the eigenpair, since in the normal case the *left* eigenvector is also the *right* eigenvector. Then, at least as late as $|\hat{v}_i^H q_{k+1}| \approx 1$, i.e., when $q_{k+1} \approx c\hat{v}_i = cv_i$, $|c| = 1$, a second Ritz value approximating $\lambda_i$ *has* to emerge.

This can only happen, when the orthogonalisation scheme is *not strong enough* to ensure sufficiently orthogonal basis vectors. But the good news in this case is, that still we only will compute a second Ritz pair approximating the same eigenpair. If the orthogonalisation scheme *is* strong enough to prevent a repeated copy, which is the case for all orthogonalisation schemes we (numerically) investigated, the computed Ritz pair looses already gained accuracy. In the non-normal case, the divergence is still bounded from above,

$$1 \approx \|q_{k+1}\| > |\hat{v}_i^H q_{k+1}|,$$

but we no longer have the nice interpretation we had in the normal case.

We now switch to the accuracy of the computed basis. The computed basis deviates whenever there is a cluster. As an example, we consider a multiple derogatory eigenvalue. The starting vector defines only *one* eigenvector that should be computed. The finite precision recurrence for the basis vectors,

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}}\, q + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(A)}{\prod_{p=l}^{k} c_{p+1,p}}\, f_l \right],$$

shows that *any* errors in direction of the eigenspace of the derogatory eigenvalue *orthogonal* to the eigenvector defined by the starting vector will eventually blow up and naturally are *not* removed by the orthogonalisation. That these error components *will* blow up follows from the identity

$$\hat{v}_i^H q_{k+1} = \frac{(\lambda_i - \theta_j)\, \hat{v}_i^H y_j + \hat{v}_i^H F_k s_j}{h_{k+1,k} s_{kj}}.$$

It is easy to extend this observation to *clusters* of eigenvalues, since for clusters, $(\lambda_i - \theta_j)\hat{v}_i^H y_j$ will be small for *all* eigenvalues $\lambda_i$ in the cluster, when *one* Ritz value approximates *one* eigenvalue in the cluster.

This proves that *even if the computed matrices* $Q_k$ are *orthogonal to machine precision*, different methods will often result in very different $Q_k$, since the error vectors $f_j$ of two implementations are usually not related. The same holds true for *one* implementation applied first to a given starting vector $q$ and *any* slightly perturbed vector $\tilde{q} = q + \Delta q$.

This shows that all variants of finite precision Arnoldi are not forward stable in computing the Krylov space and corresponding basis. We may ask for classes of matrices such that finite precision Arnoldi stably computes the Krylov space and basis. This class corresponds to the class of matrices, such that convergence of all Ritz values is delayed until the last step. Accordingly, the basis and space is accurate as long as no convergence occurred.

The Householder Arnoldi variant does not lie, i.e., the bounds on the accuracy of the computed Ritz values (the estimated residuals) returned are close to the true residuals as long as they are slightly larger than the backward error. The convergence to single eigenvalues is comparable to the convergence of the infinite

precision Arnoldi method. The same remarks apply to the Givens variant. With respect to these remarks, we might consider the finite precision Householder and Givens implementations for the computation of approximate eigen*values* 'backward stable'.

### 5.1.4 The Symmetric Lanczos Method

> "[..]. However in computations comparing the accuracy of the computed eigenvalues with the loss of orthogonality and the bounds on this, it was found in many cases that several eigenvalues converged to great accuracy despite complete loss of orthogonality. Startling examples of this occurred when the number of steps far exceeded the dimension of the matrix, as in such cases it often happens that repeated eigenvalues of the tri-diagonal matrix correspond accurately with single eigenvalues of the original matrix."
>
> Christopher Conway Paige,
> The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices (1971, page 87).

The history of the error analysis of the symmetric Lanczos algorithm is closely connected to a complete *series* of theses starting with Paige's thesis in 1971 (cf. [Pai71]), followed by the theses of Scott (cf. [Sco78], 1978), Grcar (cf. [Grc81], 1981) and Simon (cf. [Sim82], 1982). Until 1971, it was believed that maintaining **global orthogonality is crucial**. Unfortunately, global orthogonality and even linear independency of the computed basis vectors is lost shortly after the first Ritz pair has converged. As remedy, the Lanczos method was implemented with full re-orthogonalisation and *all* computed vectors had to be stored. The resulting algorithm is termed *LanFO* for short.

Inspired by observations made in the late sixties and the analysis of LanFO (cf. [Pai69, Pai70]), Paige was the first to give a detailed error analysis explaining that the computed eigenpairs are close to exact eigenpairs in case only *local* orthogonality is maintained, i.e., he showed that **local orthogonality is sufficient**. To be more precise, he showed that to every cluster of Ritz values there corresponds (at least) one close eigenvalue, and that the estimated residuals are close to the true residuals. Paige's results can be summarised as follows:

a) A classification and criticism of the different implementations (including the classical and modified Gram-Schmidt variants) of the finite precision symmetric Lanczos method (cf. [Pai71, Pai72]). He pointed out the importance to compute a *symmetric* tridiagonal.

b) The proof that the error matrix $F_k$ in $AQ_k = Q_{k+1}\underline{T}_k - F_k$ is small. Paige gave an explicit normwise bound and introduced the additive splitting that forms the core of Theorem 4.17 (cf. [Pai71, Pai76]).

c) Paige used the splitting and the relations given in Theorem 4.18, Theorem 4.21 and Theorem 4.22 to show that the Ritz values converge to eigenvalues even if they appear in multiple copies and the Ritz vectors deviate substantially from unit length (cf. [Pai71, Pai80]).

Moreover, he showed that the Ritz values are contained in a slightly enlarged version of the field of values of $A$, which is important for the convergence of CG. These results are *not* included in Chapter 4, since they only apply to the symmetric Lanczos method. The results in the article (cf. [Pai80]) are slightly stronger than the results included in his thesis. We mention that a couple of results, some of them on the deviation of the *norm of Ritz vectors* that correspond to clusters, are only published in his thesis.

Another approach was successfully used by Cullum and Willoughby (cf. [CW85a, CW85b]). They tried to distinguish 'good' Ritz values from 'spurious' Ritz values. Their approach is based on the connection of the Lanczos algorithm to the CG method by Hestenes and Stiefel. They proved that Ritz values that are also Ritz values of the tridiagonal matrix consisting of the second to last row and column are due to the finite precision. If they are close to another converged Ritz value they are accepted as *duplicate*, otherwise they are marked as *spurious*. This can be compared with the role of the associated polynomials in the example of the Bessel labyrinth given in Chapter 1. Paige's analysis proves that no spurious Ritz values exist.

Cullum and Willoughby where the first that attempted an interpretation of the so-called Lanczos phenomenon (cf. [CW80]). The Lanczos phenomenon consists in the following: Every eigenvalue of $A$ tends to occur (under some circumstances) as a Ritz value of $T_k$ for 'large' $k$, $k \approx cn$, $3 \leq c \leq 5$. The interpretation is again based on the connection of the Lanczos algorithm to the CG method.

Among other results, Paige showed that loss of orthogonality is in direction of converging Ritz vectors. This knowledge was used by Parlett and Scott to derive a variant of the Lanczos algorithm that re-orthogonalises against converging Ritz vectors whenever necessary. This variant is known as *Lanczos with selective re-orthogonalisation* or *LanSO* for short.

Grcar analysed in more detail the way orthogonality is lost and was the first to show that loss of orthogonality occurs in groups of vectors of the Arnoldi basis. He derived a strategy using one vector of length $n$ to store estimators on the deviation of the recurrence, and to re-orthogonalise when one of these has grown to a prescribed size. This variant of the Lanczos algorithm is known as *Lanczos with periodical re-orthogonalisation*, or *LanPR* for short. He also was one of the first to give hints to examples of matrices where the Lanczos algorithm 'as is', i.e., without any re-orthogonalisation, works fine.

Simon analysed the underlying three-term recurrence. He formed the Green function and derived a recurrence for the errors $W_k = Q_k^H Q_k$. This error recurrence is basically a generalised Casorati determinant. The error recurrence was first discovered by Takashi and Natori and later independently rediscovered by Simon (cf. [Sim82, Sim84]). When looking more closely, this recurrence is already contained in Paige's thesis. Simon used Gersgorin disks and Cholesky decomposition of the error matrix $W_k$ to derive a variant of Lanczos he named *Lanczos with partial re-orthogonalisation*, or *LanPRO* for short. This multiplicative (Cholesky) decomposition is the core of Theorem 4.23. Simon was the first to realize that LanSO and LanPRO are variants of the larger class of Lanczos algorithms where *semi-orthogonality* of the basis vectors is maintained. The basis vectors are labelled *semi-orthogonal* if the condition $\|Q_k^H q_{k+1}\|_2 \leq \sqrt{\epsilon}$ is fulfilled.

Greenbaum showed that every finite precision recurrence may be continued to lead to a value $\beta_{n+m} = 0$ after at least $n + m$ steps

$$AQ_{n+m} = Q_{n+m}T_{n+m} - F_{n+m}, \qquad F_{n+m} = [F_k, F_k^a]$$

with some additional perturbations $f_l^a$ (cf. [Gre89], see also [GS92]). She concluded from the results obtained by Paige that the size of the additional error vectors $f_l^a$ can be bounded by

$$O\left((n+m)^3 k^3 \|A\| \sqrt{\epsilon}\right) \qquad \text{or} \qquad O\left((n+m)^3 \sqrt{c_{\max}^3 k} \|A\| \sqrt[4]{\epsilon}\right)$$

where $c_{\max}$ is the maximal size of a cluster of Ritz values that has not converged yet (cf. [Gre89], Theorem 1', page 51). This, of course, is a very weak bound.

A result of Paige implies that the eigenvalues of $T_{n+m}$ are bounded by these expressions. We remarked that every Hessenberg matrix (symmetric tridiagonal

matrix) is re-constructed by application of the infinite precision Arnoldi method (symmetric Lanczos method). Thus, Greenbaum concluded that the *finite* precision Lanczos algorithm behaves like an *infinite* precision algorithm applied to a *larger* matrix $\tilde{A} \in \mathbb{K}^{(n+m)\times(n+m)}$, whose eigenvalues lie in clusters around the eigenvalues of the original $A \in \mathbb{K}^{n\times n}$. The numerical experiments carried out in the paper with Strakoš (cf. [GS92]) suggest that the crude bounds on the diameter of these intervals may be replaced by $O(\|A\|\epsilon)$. Unfortunately, no rigorous proof has been found yet to support this assumption.

In contrast to the general case in the Arnoldi method, the *left* eigenvector $\hat{v}$ of $A$ is always equal to the right eigenvector $v$, because the symmetric Lanczos method is restricted to symmetric (or Hermitian) matrices $A$. A *major* distinction from the finite precision Arnoldi method is that the loss of orthogonality can happen 'freely' in the finite precision symmetric Lanczos method, since we *never* compute the inner products of basis vectors whose indices are far apart from each other. Thus, the loss of orthogonality will be recovered when the quantity $v^H q_{k+1}$ approaches the 'sky', since once $v^H q_{k+1} = O(1)$, a new Ritz value has to emerge and so forth. This will become clear in the following examples.

Again, we give two plots resulting from a toy-sized problem. The problem has been carefully chosen, such that the convergence of the largest Ritz value is *very* fast and thus also the *loss of orthogonality* occurs almost *immediate*. The matrix $A$ has been chosen such that $A \in \mathbb{R}^{100\times 100}$, with elements randomly chosen (except for the symmetry constraint) such that all $\{a_{ij}\}_{i,j=1}^{100}$ fulfil $a_{ij} \in [0,1]$. This ensures that $A$ is non-negative, so we can apply *Perron-Frobenius theory* to conclude that the spectral radius is equal to the largest eigenvalue, the so-called *Perron root*.

With this special choice, the matrix $A$ may be considered as a perturbation of the rank-one matrix

$$E = \frac{1}{2}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}, \qquad A = E + \Delta E.$$

We sketch the general case where $A \in \mathbb{K}^{n\times n}$ is chosen to fit the mentioned constraints. The matrix $E$ has only one non-zero eigenvalue $n/2$ to the eigenvector $e$, where $e$ is the vector of all ones. In our case, the non-zero eigenvalue of $E$ is $50 = 100/2$.

The coefficients of the symmetric matrix $2\Delta E$ are *independent* and *identically distributed* (iid) random variables in the range $[-1,1]$. The normalised $(\lambda/\sqrt{n})$ eigenvalues have limiting $(n \to \infty)$ semicircle distribution given by

$$P(x) = \begin{cases} 0, & x \notin [-1,1], \\ \dfrac{2}{\pi}\sqrt{1-x^2}, & x \in [-1,1]. \end{cases}$$

This result is known as *Wigner's Semicircle Law*. Thus, we may expect the eigenvalues of $\Delta E$ to be approximately semicircle distributed in the range $[-\sqrt{n}/2, \sqrt{n}/2] = [-5,5]$.

By taking a closer look at Weyl's Theorem and by preceeding arguments we may expect the largest eigenvalue of $A$ somewhere near 50, the second largest eigenvalue near 5 and the remaining part of the eigenvalues in the interval $[-5,5]$. The eigenvector by Perron-Frobenius theory has only non-negative entries and perturbation theory ensures that it will be close to $e$. In our example, the Perron root was given by 50.5877, the second largest eigenvalue was given by 4.1676 and the angle between the computed Perron vector and $e$ was 0.0379.

We chose a non-negative random starting vector. By Kaniel-Paige-Saad theory we expect a very fast convergence towards the largest eigenvalue. The convergence

history is shown in the first figure, figure 5.4. The line styles have been chosen as in the examples for finite precision Arnoldi, that is, we plot the quantities $|\lambda_{\max} - \theta_{\max}|/\|\|A\|\|_2$ we labelled *real convergence* with *dashed* lines (– –), the quantities $|\beta_k s_{k\,\max}|$ we labelled *estimated residuals* with *solid* lines (—) and the quantities $v_{\max}^H q_{k+1}/\|\|A\|\|_2$ we labelled the *actual eigenpart* with *solid dotted* lines (·–·).

We observe a fast and almost linear convergence in the semilogarithmic plot in steps 1 to 5. In step 5 the Ritz value $\theta_{\max}$ has converged to the largest eigenvalue $\lambda\,\max$ with a relative accuracy up to machine precision. In this first range, the actual eigenpart and the estimated residual are in theory and in the numerical experiment close to each other with magnitude more or less given by the square root of the distance between $\theta_{\max}$ and $\lambda_{\max}$. The last component $s_{k\,\max}$ of the actual $s_{\max}$ is given by (Chapter 2, Theorem 2.20, page 70)

$$\frac{s_{\max}^{(k)}(k)}{s_{\max}^{(k)}(1)} = \frac{\chi_{T_{k-1}}(\theta_{\max}^{(k)})}{\prod_{l=1}^{k-1}\beta_l}.$$

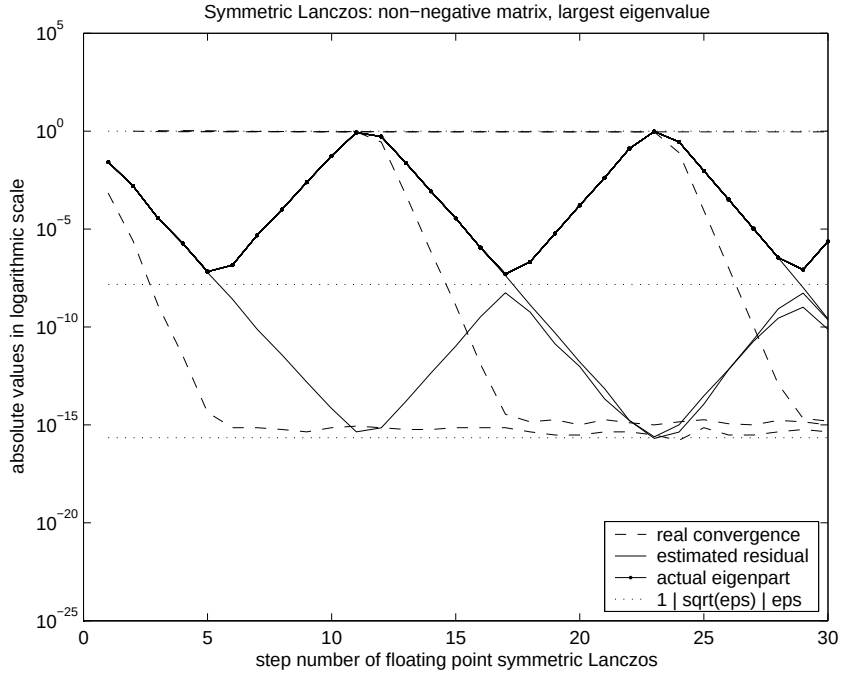


Figure 5.4: Symmetric Lanczos, non-negative matrix, $\lambda_{\max}$

With our choice of starting vector, the scaling by $s_{\max}^{(k)}(1)$ will not vary greatly with $k$. The evaluation at value $\theta_{\max}^{(k)}$ of the characteristic polynomial of $T_{k-1}$ is insensitive to perturbations of roots (i.e., Ritz values) that are far away from $\theta_{\max}^{(k)}$. The result, i.e., that $s_{k\,\max}$ becomes only sensitive when errors cause a new Ritz value close to $\theta_{\max}$, also follows from the eigenvector – eigenvalue relations of Chapter 2, since for symmetric matrices $T_k$

$$\left(s_{k\,\max}^{(k)}\right)^2 = \frac{\chi_{T_{k-1}}(\theta_{\max}^{(k)})}{\chi'_{T_k}(\theta_{\max}^{(k)})} = \frac{\prod_j \left(\theta_{\max}^{(k)} - \theta_j^{(k-1)}\right)}{\prod_{j\neq\max}\left(\theta_{\max}^{(k)} - \theta_j^{(k)}\right)} \tag{5.2}$$

holds true. In other words, as long as there is no other Ritz value close to $\theta_{\max}^{(k)}$, the estimated residual curve decreases as the infinite precision counterpart would have done.

This, together with a bound on the errors $v_{\max}^H F_k s_{\max}$, implies that the curve corresponding to the part of $q_{k+1}$ in direction of the eigenvector $v_{\max}$ will start to *increase* as long as it has not reached substantial magnitude. This can be observed in between steps 5 and 11. Remember that all the curves are *only defined for integer values*. Keeping this in mind, we observe that we can *'erase the wrong edges'* between steps 5 and 6 in the $v_{\max}^H q_{k+1}$ curve and in the (relative) distance between $\theta_{\max}$ and $\lambda_{\max}$. The increase in direction of the eigenvector causes the *loss of orthogonality* shown in figure 5.5.

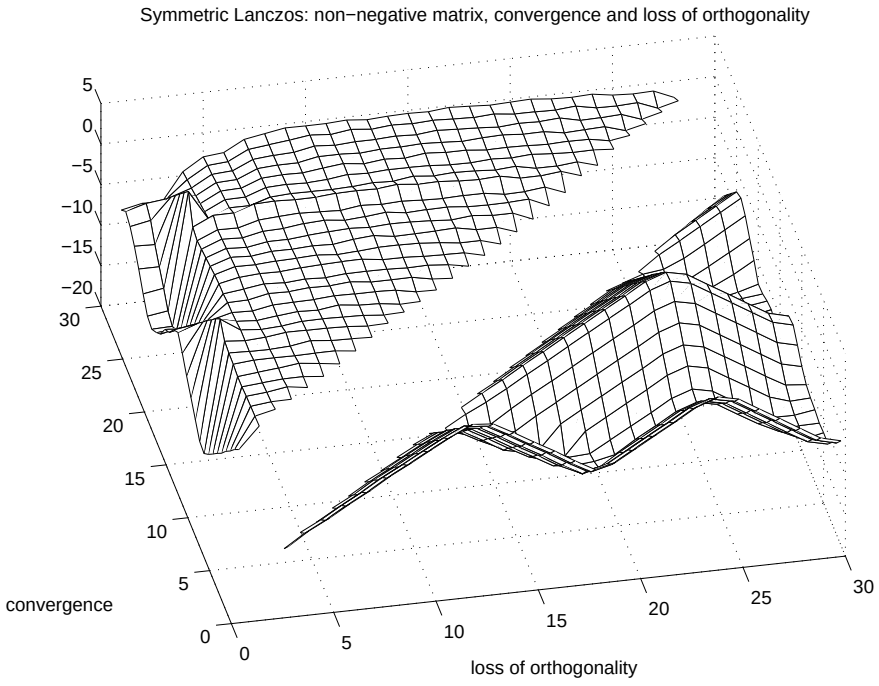


Figure 5.5: Symmetric Lanczos, loss versus convergence (II)

As already mentioned, a new Ritz value *has* to appear that converges to the *same* eigenvalue $\lambda_{\max}$. This happens in step 11. Remember that the rate of convergence of the first converging Ritz value was almost linear and took about 5.5 steps. Then it is obvious that the next Ritz value (as long as no other Ritz values interact with the convergence) appears at step 11, the next one at step 22 and so forth. This is nicely to see in the picture. We shortly go on to explain some important aspects of the behaviour observable in the plot. Every Ritz value that has converged to attainable accuracy can not move very far away once the estimated residual is small. This is known as Paige's Persistence Theorem and is the result of the backward errors of tridiagonal eigensystems introduced in Chapter 2 and the perturbation theory of Chapter 1. The convergence of the Ritz values follows again approximately Kaniel-Paige-Saad Theory. This can be seen by introducing a 'magic perturbation'. This perturbation is defined by first multiplying the perturbed Krylov decomposition

from the right by the matrix factor $(I_k - s_{\max} s_{\max}^H)$, resulting in

$$A\tilde{Q}_k - \tilde{Q}_k \tilde{T}_k = M_k - \tilde{F}_k,$$

where $\tilde{Q}_k = Q_k(I_k - s_{\max} s_{\max}^H)$, $\tilde{T}_k = T_k - s_{\max}\theta_{\max} s_{\max}^H$ and the new error matrix is defined by $\tilde{F}_k = F_k(I_k - s_{\max} s_{\max}^H) + M_k s_{\max} s_{\max}^H$. We note that the error matrix has maximal growth given by $|\beta_k s_{k\,\max}|\|q_{k+1}\| \approx |\beta_k s_{k\,\max}|$ and that the matrix $\tilde{T}_k$ has the same eigenvalues and eigenvectors as $T_k$, with the exception of $\theta_{\max}$, which has been replaced by zero.

As a last step, we transform the (non-tridiagonal) $\tilde{T}_k$ to tridiagonal form by a similarity transformation $U$ and set $Q_k^{\text{new}} = Q_k U$. We seek $U$ such that local orthogonality is not spoiled. This can be achieved using Householder transformations backwards on $\tilde{T}_k$,

$$U T_k^{\text{new}} U^H = \tilde{T}_k.$$

Let $s_{\max}^{(1:k-1)}$ denote the reduced vector consisting of the first $k-1$ elements of $s_{\max}$. Then, the first $k-1$ components of the last column of Hermitian $\tilde{T}_k$ are given by

$$w_{k-1} = \overline{s}_{k\,\max} s_{\max}^r + \beta_{k-1} e_{k-1}.$$

This shows that the last column of $U$, given by the last column of the first Householder matrix $H$ defined by $H w_{k-1} = \|w_{k-1}\|_2 e_{k-1}$, is very close to the last unit vector $e_{k-1}$. Furthermore, $\beta_{k-1} - \|w_{k-1}\|_2$ is small. The size of the error matrix is not altered by multiplication with a unitary matrix. The recurrence continues without touching and altering the other quantities we have already transformed, especially, $M_k U = M_k$. We only have increased the bound on the error matrix and removed the largest Ritz value. Putting things together, this implies that the newest Ritz value will converge approximately like its predecessors with about the same rate of convergence.

Once we have several Ritz values approximating one eigenvalue, the corresponding estimated residual curves $\beta_k s_{kj}$ will all move in a similar fashion, since they only differ in the point of evaluation of the polynomials given in equation (5.2). That is, when a new Ritz value is converging and then as long as no new Ritz value appears, the curves are decreasing, and when a new Ritz value comes close, the curves are all increasing. Since $\beta_k s_{kj}$ is used as estimate for the residual, we may (wrongly) think that we are far away from desired accuracy when we are in the latter case.

Next, we consider a more complicated case. The first case resulted in such beautiful pictures, because no other Ritz value converged substantially in the steps 1 to 30. We focus on the interactions between converging Ritz values. To do so without discarding previously obtained information, we simply plot the curves for the convergence of the *second largest eigenvalue* of the *same matrix A*. To better see the interactions with the convergence of the largest eigenvalue, we also plot the curves corresponding to this case in a dotted line style. The convergence of the second largest eigenvalue takes more steps, we chose to show steps 1 to 80. The resulting plot is given by figure 5.6.

We summarise the main changes. The curves are only irritated every time the deviation in the basis vectors is largest, i.e., every time a new Ritz value starts to converge to the largest eigenvalue. This effect is due to the normalisation of the basis vectors.

Let us assume that the basis vectors are expanded in the eigenbasis, $q_j = \alpha_{ji} v_i$, where $\alpha_{ji} = v_i^H q_j$. Then by Theorem 4.32, page 172, the loss in direction of $v_{\max}$ adds to the (un-normalised) vector $r_k = \beta_k q_{k+1}$ expanding the Krylov space,

$$v_{\max}^H r_k = \frac{(\lambda_{\max} - \theta_{\max}) v_{\max}^H y_{\max}}{s_{k\,\max}} + \frac{v_{\max}^H F_k s_{\max}}{s_{k\,\max}}.$$

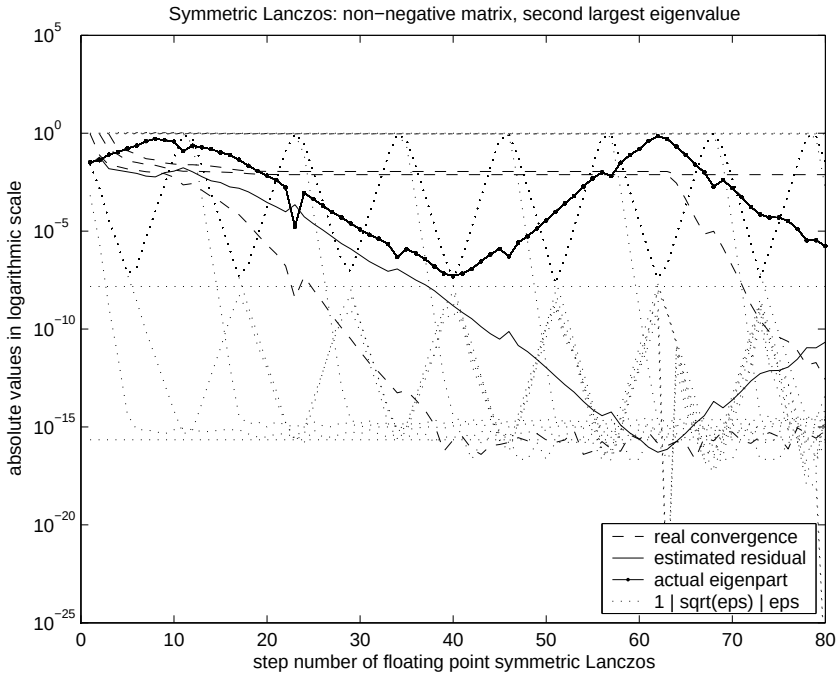


Figure 5.6: Symmetric Lanczos, non-negative matrix, $\lambda_{\max -1}$

It appeals to be more natural to consider only the first summand. Since the normalisation can not distinguish which part of the vector $r_k$ comes from errors and which is locally correct, the norm of $r_k$ measures mainly the *dominating errors* in direction of $v_{\max}$ and results in $\alpha_{j,k+1}$ having too small magnitude for *all* $j \neq \max$. This is corrected once the convergence of this newly arisen Ritz value takes place.

Depending on the distance of the Ritz value to the converging one, i.e., on the gap between two neighbouring Ritz values, the residual estimator curves may move up slightly. This follows by the normalisation of the eigenvectors and also by the representation of the last components according to the eigenvector – eigenvalue relations of Chapter 2 as stated above, equation (5.2),

$$\left(s^{(k)}_{k\,\max -1}\right)^2 = \frac{\chi_{T_{k-1}}(\theta^{(k)}_{\max -1})}{\chi'_{T_k}(\theta^{(k)}_{\max -1})} = \frac{\prod_j \left(\theta^{(k)}_{\max -1} - \theta^{(k-1)}_j\right)}{\prod_{j\neq \max -1}\left(\theta^{(k)}_{\max -1} - \theta^{(k)}_j\right)}.$$

Here, with some abuse of notation, $\theta_{\max -1}$ is *not necessarily* the second largest Ritz value, but the first Ritz value that converges to the second largest eigenvalue of $A$, denoted $\lambda_{\max -1}$. We label the Ritz values approximating the largest eigenvalue by $\theta_{\max,i}$. When a *new* Ritz value $\theta_{\max,i}$ close to $\lambda_{\max}$ occurs, the term

$$\frac{(\theta^{(k)}_{\max -1} - \theta^{(k-1)}_{\max -1})}{(\theta^{(k)}_{\max -1} - \theta^{(k)}_{\max,i})}$$

dominates the representation and causes a jump in the quantity $s_{k,\max -1}$ of height approximately given by $\sqrt{\text{dist}/\text{sep}}$, where dist denotes the distance between two successive Ritz values $\theta^{(k-1)}_{\max -1}$ and $\theta^{(k)}_{\max -1}$ that approximate the second largest eigenvalue $\lambda_{\max -1}$, and sep denotes the (smallest) spectral separation between the

approximations $\theta^{(k)}_{\max,i}$ and $\theta^{(k)}_{\max-1}$, approximating the largest eigenvalues $\lambda_{\max}$ and $\lambda_{\max-1}$.

This proves that the distraction of the curve $s_{kj}$ depends on the square root of the speed of convergence (dist) divided by the separation (sep) to the other, faster converging, Ritz value. The curve $\beta_k s_{kj}$ also depends on the norms $\beta_k$ of the residual vectors. The computed $\beta_k$ is (by previously stated arguments) dominated by errors in direction of the eigenvector $v_{\max}$ corresponding to the largest eigenvalue. Thus, this $\beta_k$ will, in this setting, be a large overestimate over the 'better' $\beta_k$ taking into account only the components that measure convergence.

In the above, we neglected the variation of all other Ritz values. This is in order, since most Ritz values in the example are *slowly converging*. Only the *largest* Ritz value already has converged, but this Ritz value can not vary greatly as is implied by *Paige's Persistence Theorem*.

All the above results apply to infinite precision and clusters. Essential the same behaviour can be observed. Here, we can interpret the clustered eigenvalues as perturbed single eigenvalues of $\tilde{A} = A+\Delta A$ and define the perturbation term $\Delta A Q_m$ as error matrix. This interpretation nicely explains the so-called misconvergence of the symmetric Lanczos method. The phenomenon entitled 'misconvergence' is based on the observation that a Ritz value approximating a cluster moves comparably fast to a certain value and then, after stagnating a while, starts to move on to approximate a single eigenvalue of the cluster. The period of stagnation corresponds exactly to the number of steps it takes for the part of the basis vectors in direction of the eigenvectors to be amplified sufficiently.

This can be memorised as the following rule of thumb:

> The symmetric Lanczos method is *short-sighted* with respect to multiplicity or clustering of eigenvalues.

In the first steps, the symmetric Lanczos algorithm 'sees' only one single eigenvalue, realises soon that there are at least two, then begins to 'see' the whole spectrum.

This holds true for the infinite precision symmetric Lanczos method. In case of execution in an finite precision environment, we can say moreover (even more loosely speaking):

> The *finite precision implementations* of the symmetric Lanczos algorithm *tend to forget* about the locations of eigenvalues they already have determined as accurate as possible before.

The intimate connection between the *convergence rate* and the *loss of orthogonality* is the reason for the frequently observed behaviour that Ritz values to the same eigenvalue occur almost periodical, i.e., almost every $2\ell$ steps, where $\ell$ is the number of steps necessary to converge once to attainable accuracy.

### 5.1.5 The Non-Symmetric Lanczos Method

> "If the Lanczos process is carried out in practice [..], then the strict bi-orthogonality of the two sequences $c_i$ and $c_i^*$ is usually soon lost. Whenever considerable cancellation takes place on forming a vector $b_i$ or $b_i^*$, a catastrophic deterioration in the orthogonality ensues immediately. To avoid this it is essential that re-orthogonalisation is performed at each stage."
>
> James Hardy Wilkinson,
> The Algebraic Eigenvalue Problem (1965).

The different non-symmetric Lanczos algorithms are very close to their symmetric counterpart. This holds also true for part of the error analysis. The names that are connected with the error analysis of the non-symmetric Lanczos algorithms are Bai and Day. Bai extended some of the first results of Paige to the non-symmetric case (cf. [BDM91, Bai94, BDY99]). These extensions are all based on the additive splitting approach of the matrix $\hat{Q}_k^H Q_k$ into diagonal, strictly upper and strictly lower part (Chapter 4, Theorem 4.17, page 153, Theorem 4.18, page 154, equation (4.33), page 154). Day extended Simon's results, namely the multiplicative splitting, to the non-symmetric Lanczos variant baring his name. The development of Day's Lanczos variant and the error analysis can be found in his thesis (cf. [Day93]), or in shorter form in an article (cf. [Day97]). For sake of completeness, we note that Cullum and Willoughby used their identification and rejection scheme also in the *non-symmetric case*.

All in one, the most important parts of the error analysis of the *non-symmetric* Lanczos process can be handled similar to the error analysis of the *symmetric* Lanczos process, the major change is that the condition of the eigenvalues, and more serious, the condition of the Ritz values comes into play. By now, no final variant has been elected that has been proven to be the 'best' variant, for instance with respect to minimising the condition number of the Ritz values computed.

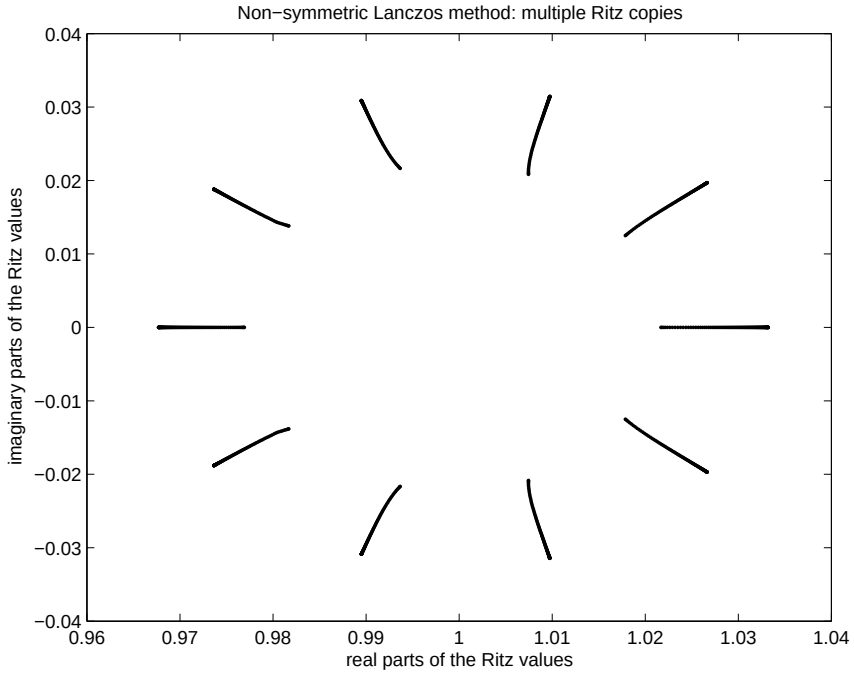


Figure 5.7: Non-symmetric Lanczos, Jordan-block of size 10

Important for the understanding of the success of the non-symmetric Lanczos algorithm is that the convergence of the right Ritz vector leads to a loss in the left basis vectors that will be recovered because of the normalisation, and vice versa with the roles of *right* and *left* switched. When the wrong normalisation has been chosen, the errors will become very large and prevent a high accuracy. This need not be the case, as can be seen in the plot of figure 5.7. The plot shows

the eigenvalues of a 1000 by 1000 tridiagonal matrix obtained by application of a finite precision non-symmetric Lanczos algorithm to a Jordan block of size 10 with single eigenvalue one. Some more aspects of the floating point behaviour of the non-symmetric Lanczos algorithm (together with Arnoldi and the symmetric Lanczos algorithm) have already been discussed in part in an article by Zemke (cf. [Zem01]) in the Dagstuhl proceedings [ARRY01].

Nearly all comments made on the runs of the finite precision symmetric Lanczos method apply also to the non-symmetric variants. We only have to ensure that the quantities $\beta_k s_{kj}$, that gave estimates for residual bounds in the symmetric variant, are still bounds for enlarged tridiagonal matrices. In the backward sense this remains true as an easy calculation shows. Thus, a non-symmetric version of Paige's Persistence Theorem holds true. This is apparent from the theorem of Kahan, Parlett and Jiang (see Chapter 1, Lemma 1.35, page 38, Lemma 1.36, page 38). The major distinction from the symmetric Lanczos method is that the error vectors may grow unbounded (with respect to *a priori* bounds), or in case of Day's variant, the computed inner products $\omega_k$ may become arbitrarily small.

We just state as a rule of thumb that all computed $T_k$ are small perturbations of exact oblique projections (in general *not* resulting in a tridiagonal matrix) as long as the condition of the eigentriplets is not too large and the errors have not amplified yet. We remark that there is plenty room for improvement in the results that can be found in the literature on the subject. These improvements might be based on the generalised eigenvalue – eigenvector relations we have presented in this thesis.

## 5.2 Krylov Methods for Solving Linear Systems

Krylov methods for linear systems tend to deviate like their eigenproblem counterparts. Residuals that span the orthonormal basis of the Krylov subspace loose their orthogonality. Direction vectors loose their $A$-conjugacy. In general, the bases that correspond to short-term methods based on Lanczos variants will loose linear independency and the methods based on long-term methods based on Arnoldi will deviate depending on the orthogonalisation scheme. This is quite obvious from the connection to the appropriate Krylov eigensolvers.

Long-term methods will produce residuals of the same magnitude depending on the size of the error matrix, i.e., on the *quality* of the *orthogonalisation scheme* and on the *magnitude* of arising *intermediate quantities*. Short-term methods will often still converge, but the finite termination property is lost. This means that we can use them only as an *iterative* solver. We have to ask for the rate of convergence, i.e., when will the method reach a prescribed level of accuracy, and for the ultimately attainable accuracy. We have to investigate how close the cheap estimates returned in finite precision are to bounds. If they deviate significantly, we have to find another way to measure the accuracy of the residual *cheaply*, that is, without explicitly computing it *directly*.

### 5.2.1 Richardson Iteration and Polynomial Acceleration

The finite precision Richardson iteration is stable as long as the spectral radius of the iteration matrix $I - A$ is bounded by a constant somewhat less than one. The method introduces no new errors, since all quantities necessary to use Richardson iteration are given *a priori*. The errors due to execution in finite precision can be interpreted as perturbations $\tilde{A} = A + \Delta A$ of the system matrix $A$. The method will converge as long as *all* perturbed methods fulfil $\rho(\tilde{A} - I) < 1$. This robustness has to be paid for by an extremely slow rate of convergence.

The same comments apply to the more complicated polynomial accelerations like Chebychev acceleration. Here, we have to be sure that the parameters we chose *a priori* are such that the method converges for *all* slightly perturbed $\tilde{A} = A + \Delta A$. This can be accomplished, at least in theory, by choosing the parameters such that the polynomials dampen all values in an $\varepsilon$-*pseudospectrum* of $A$, where $\varepsilon$ will in most cases be a small multiple of the machine precision, the constant depending on the number of steps we want to make and the degree of the polynomial we use.

Since in these 'parameter-dependent' polynomial acceleration methods *all* knowledge has to be given *a priori* and is not altered during the algorithms, the methods are stable when the parameters have been chosen carefully, but at the same time slow. The good news is that the rate of convergence can be predicted quite sharply. We will not consider a detailed error analysis of such methods.

### 5.2.2 Orthores/Orthomin/Orthodir

In this section we consider the general forms of the three algorithmic families *Orthores*, *Orthomin* and *Orthodir*. The derivation of these three methods in Chapter 3 already shows many aspects of the numerical behaviour to be expected.

The class *Orthores* of Krylov subspace methods for the approximation of the solution of a linear system had been obtained by a re-scaling of the basis vectors, such that the basis vectors are not parallel to, but *are* the residual vectors. This scaling is equivalent to a diagonal similarity scaling of the Hessenberg matrices of any Krylov decomposition to achieve zero column sums in the resulting new Hessenberg matrix $C_m^{(0)}$. The diagonal of the scaling is the vector $y$ obtained as the solution of a system with the original Hessenberg matrix,

$$y^T C_m = \alpha e_m^T.$$

Thus, whenever the solution of this system has very small components, the error vectors will have large components. This prevents the method from converging to a high relative accuracy, since the approximate solution vectors $x_k$ are obtained by a linear combination of the residual vectors, $R_k = -X_{k+1}\underline{C}_k^{(0)}$. The vector $y$ is the (scaled) last row of the inverse of the Hessenberg matrix (when defined), since $y^T = \alpha e_m^T C^{-1}$. With knowledge on the relations between the elements of the inverse of $C_m$ and exploiting the proof for Theorem 4.49, i.e., equation (4.63) we could improve this analysis.

The usual choice of orthogonal residuals in case of a general matrix, i.e., the Orthores *method*, is known to be less stable than GMRES and has larger operation count per step. Orthores is mostly preferred to GMRES *only* when one wants to apply some form of truncation strategy, which is easier for Orthores. From a computational point of view, the difference between (standard) Orthores and GMRES is the difference between ortho*gonalisation* and ortho*normalisation* of the basis vectors.

Orthomin is based on a splitting approach of the Hessenberg matrix. We have shown in Chapter 4 (equation (4.4), page 139) that this (after scaling the columns of the basis to have unit length) results in a special structured error matrix,

$$-F_k = AF_k^{(P)} D_\rho^{-1} + F_k^{(R)} D_\rho^{-1} D_\rho L_k^{-1} D_\rho^{-1} C_k.$$

We define the short-hand notations

$$L_k^{(\mathrm{res})} \equiv D_\rho L_k^{-1} D_\rho^{-1}, \qquad E_k^{(P)} \equiv F_k^{(P)} D_\rho^{-1} \quad \text{and} \quad E_k^{(R)} \equiv F_k^{(R)} D_\rho^{-1}.$$

In the context of eigenvalues, we achieve a *relative accuracy* as long as $L_k^{(\mathrm{res})}$ is

small, since the convergence and deviation relation takes the form

$$\begin{aligned}\hat{v}_i q_{k+1} &= \frac{(\lambda_i-\theta_j)\hat{v}_i^H y_j + \hat{v}_i^H F_k s_j}{c_{k+1,k}s_{kj}} \\ &= \frac{(\lambda_i-\theta_j)\hat{v}_i^H y_j + \lambda_i \hat{v}_i^H E_k^{(P)} s_j + \theta_j \hat{v}_i^H E_k^{(R)} L_k^{(\mathrm{res})} s_j}{c_{k+1,k}s_{kj}}.\end{aligned}$$

The error terms are multiplied by either an eigenvalue or a Ritz value. If we consider a Ritz value that converges to the eigenvalue, the terms in the numerator will have the same order. This is better than the eigenvalue bounds in the approaches that are *not* based on a splitting of the Hessenberg (tridiagonal) matrix. We remark that the result implied by this observation, i.e., that computations of eigenvalues (and eigenvectors) based on the already *factored matrix* gives better results than computations based *directly* on the matrix, is mainly the result underlying Dhillon's *relatively robust representations*.

Thus, we may expect *Orthomin* to be more stable than *Orthores*, even if the *implicit* scaling in Orthomin is the same than in Orthores. Nevertheless, in the general long-term recurrence setting, the method is unstable. This is due to the fact that the scaled error matrices

$$E_k^{(P)} = F_k^{(P)} D_\rho^{-1} \quad \text{and} \quad E_k^{(R)} = F_k^{(R)} D_\rho^{-1}$$

may (and usually will) grow.

Orthodir uses the Hessenberg decomposition for the computation of the *direction vectors*. Again, the *scaling* used to define Orthodir will introduce vectors of different length. Furthermore, Orthomin and Orthodir with the usual choice of Hessenberg decomposition are closely related. They differ only in the way the basis is expanded. *Orthomin* uses the next *residual vector*, *Orthodir* uses the vector $Ap_k$, where $p_k$ is last *direction vector*. These vectors are $A$-orthogonalised against the previously computed vectors. The next vector in Orthomin uses one multiplication with $A$ more than Orthomin does. This is prohibitive when $A$ has a comparable large condition number.

Summarised for all three methods, we observe that all three are more or less *unstable*. This is because the update formula used for the residual and/or direction vectors uses *linear combinations* of previously computed vectors of in general *greatly varying length*. At least the *residual* vectors should converge to zero. When we expect them to have a small norm for larger indices $k$, the recurrences are dominated by *cancellation*, i.e., by *error terms*. The winner among Orthores, Orthomin and Orthodir is Orthomin. The methods are usually stopped when we have reached a sufficient level of accuracy. If only a relatively large-sized residual is wanted, for instance in restarted methods, all three types of methods could be applied, preferably Orthomin.

### 5.2.3 FOM/GMRES

FOM and GMRES by Saad and Schultz (cf. [SS86]) had been introduced to overcome the inherent instabilities in Orthores/Orthomin/Orthodir and to generalise the pair SymmLQ/MinRes. No scaling of the Hessenberg decomposition is needed. FOM and GMRES are directly based on the Arnoldi recurrence 'as is'. Thus, we can plug in the results for the finite precision Arnoldi recurrence, and add facts about the solution of small Hessenberg linear systems of equations and small Hessenberg least squares problems.

First we focus on some aspects of FOM. Even though the matrix $A$ may be perfectly regular, the matrix $H_k$ may be singular, even in infinite precision. In

infinite precision, this can only be the case, when zero is in the field of values of $A$. When the field of values of $A$ is far from zero, the in infinite precision computed Hessenberg matrices $H_k$ will be non-singular. The remarkable fact about the FOM approach is that the solution of the small system is usually done in a backward stable manner, mostly with Givens rotations. The backward stability takes the form

$$r_k^{(H)} = \|r_0\|e_1 - H_k z_k, \quad \|r_k^{(H)}\|_2 = O(\|H_k\|_2 \epsilon).$$

This ensures that *all* solutions that are well-defined by Hessenberg systems with a small condition number are computed quite accurately. This, in turn, implies that the *true* residual of $x_k \equiv Q_k z_k$ is given by

$$\begin{aligned} r_k^{(A)} &\equiv r_0 - Ax_k = Q_k\|r_0\|e_1 - AQ_k z_k \\ &= Q_k(H_k z_k + r_k^{(H)}) - AQ_k z_k \\ &= Q_k r_k^{(H)} - M_k z_k + F_k z_k. \end{aligned}$$

When the orthogonalisation scheme is sufficient to ensure that the norm of $F_k$ is small, let us assume the norm has order $O(\|A\|\epsilon)$, as is the case for all variants we have considered so far, we observe that the deviation of the true residual from the estimated residual is small,

$$\|r_k^{(A)} - (-h_{k+1,k})q_{k+1}z_{kk}\|_2 \;=\; O((\|Q_k\|_2\|H_k\|_2 + \|A\|_2)\epsilon).$$

When the orthogonalisation is also sufficiently strong to ensure numerical orthogonal basis vectors, the $k$th approximation will be close to a $k$th approximation with roughly the same residual. This will be the case for the Householder Arnoldi variant.

In the minimal residual case, we have to compute the solution of the small least squares system

$$\|\tilde{H}_k z_k - e_1\| = \min.$$

This is done in a iterative manner, and is stable as long as the condition of the computed $\underline{H}_k$ is not too large. Rozložník worked on GMRES in his thesis (cf. [Roz97]). He proved that the *Householder GMRES* variant is *backward stable* (cf. [Roz97], Theorem 4.6, page 34). Rozložník also considered MGS GMRES and proved that the orthogonality is lost inverse proportional to the norm $\min_y \|q_1 - AQ_k y\|$ of the best approximation to $q_1$ in the space spanned by $AQ_k$. He then proved that the norm of the true residual $r_k = b - Ax_k$ and the norm of the Hessenberg residual have the same order of magnitude as long as this term reaches a level proportional to $\kappa(A)\epsilon$ (cf. [Roz97], Theorem 5.5, page 40, Theorem 5.6, page 43).

### 5.2.4 Truncated and Restarted Methods

When we consider truncated and restarted methods, the underlying Hessenberg decompositions have zeros in the upper part. This removes the constraints on the inner products of the basis and the constraint vectors. When executed in finite precision, every component in direction of these constraints is not removed, regardless how good the (orthogonalisation) scheme works. This will often result in an amplification of parts in the basis vectors that already had been removed form the recurrence.

As an example, we might consider the finite precision Arnoldi method 'as is' applied to a *symmetric* matrix $A \in \mathbb{K}^{n\times n}$. Then, the method will compute a *Hessenberg* matrix, where the upper part is not zero as it would be in infinite precision. When we assume that the Householder variant has been used, we obtain at least in the $n$th step an approximation that is very close in backward sense

to the exact solution. When we now truncate the recurrence to obtain a three-term recurrence, we obtain the finite precision Lanczos recurrence. This introduces multiple Ritz values and we will re-do parts of the solution process again and again. Mostly, the method will converge in a number of steps that is a *multiple* of the dimension of the problem.

We refuse to consider truncation and restart in full generality, since no general way of choosing the *optimal* parameters has evolved in the *infinite precision* context. Thus, we may suspect that it will take a while until the development of these method classes has settled enough to make a finite precision error analysis useful.

### 5.2.5 CG/CR

The three variant of CG and CR have different finite precision behaviour. We consider mainly the Orthomin variant. The finite precision Orthomin variant of CG is closely related to finite precision symmetric Lanczos, like in infinite precision. The relation is not bijective. For an SPD matrix $A$, the way from the finite precision Lanczos method to a perturbed Orthomin CG variant is contained in an article by Cullum and Willoughby (cf. [CW80]) and, using heavily Paige's results, in the book by Cullum and Willoughby (cf. [CW85a], pages 101–118). The correspondence is based on constructing residual vectors by a scaling of the basis and direction vectors by an LDLT decomposition of $T_k$. In a second step, Cullum and Willoughby prove that this results in a perturbed CG method with approximately locally $A$ conjugate direction vectors. Cullum and Willoughby use this correspondence to show that eventually *all* eigenvalues of $A$ become approximated by (certain) Ritz values, i.e., eigenvalue of $T_k$ for large values of $k$.

The other way, from finite precision CG to a perturbed Lanczos method is given in the section on Orthomin. The CG method may also be considered for *indefinite* symmetric matrices, but the underlying LDLT decomposition behaves frequently forward unstable. This results often in large multipliers which raises the ultimately attainable accuracy to a higher level, often such that the method can not converge any further. The *relative* (i.e., multiplied with $A$ and the implicitly computed $T_k$) error matrices of such a coupled two-term recurrence suggest better numerical properties than the three-term recurrence forms Orthores and Orthodir of CG. This was analysed by Gutknecht and Strakož in a technical report (cf. [GS97], see also [GS00]). In this report and paper, the authors prove that coupled two-term recurrences in general have better numerical properties than three-term variants.

The minimising property of the CG method, i.e., viewing CG as an optimisation procedure shows that locally the method is still optimising. It can be shown that CG with restart after $n$ steps will converge. The Cullum and Willoughby approach shows that when the Orthomin variant of CG converges, the eigenvalues of $A$ have to be found. The error analysis for CR is similar. CR is seldom used, mostly MinRes is preferred.

### 5.2.6 SymmLQ/MinRes

When it comes to the solution of *indefinite* symmetric linear systems of equations, SymmLQ for the OR solution and MinRes for the MR solution by Paige and Saunders are more suitable. These methods can be seen as special forms of FOM and GMRES. SymmLQ uses the tridiagonal $T_k$ from the finite precision Lanczos algorithm and performs an LQ decomposition. This decomposition is backward stable and gets along with only small additional storage. MinRes is based on the LQ decomposition of the enlarged tridiagonal matrices $\underline{T}_k$. Also this part is backward

stable. Thus, the accuracy in both methods depend basically on the size of the error term $F_k$ of finite precision Lanczos.

Sleijpen, van der Vorst and Modersitzki have analysed the differences and the ultimately attainable accuracy in SymmLQ, MinRes and a specially adopted GMRES variant using the (non-orthonormal) basis constructed by the finite precision Lanczos algorithm (cf. [SvdVM97, SvdVM01]). The only *computational* difference between MinRes and the adopted GMRES is the order of evaluation of the computation of the approximation $x_k$:

$$\begin{aligned} x_k &= Q_k(L_k^{-H} y_k), && \text{(adopted GMRES)}, \\ x_k &= (Q_k L_k^{-H}) y_k, && \text{(MinRes)}. \end{aligned}$$

Here, $y_k = I_{k,k+1}\|r_0\|U_{k+1}e_1$ denotes the transformed right-hand side of the least squares problem

$$\|\underline{T}_k z_k = \|r_0\|e_1\| = \min, \qquad \underline{T}_k = U_{k+1}^H \underline{L}_k^H.$$

Concerning storage, the difference between MinRes and the adopted GMRES is that MinRes uses short recurrences ($V_k = Q_k L_k^{-H}$) and the adopted GMRES uses *all* previously computed vectors.

The authors pre-assume that the error $F_k$ in the perturbed Hessenberg decomposition is bounded, i.e., they use the results of Paige. They give relative bounds on the accuracy of the computed approximations. These bounds are of order $\kappa_2(A)\epsilon$ for both the adopted GMRES and SymmLQ, which is to be expected. The bound for MinRes is of order $\kappa_2^2(A)\epsilon$. Numerical experiments support the claim that adopted GMRES and SymmLQ really are *superior* to MinRes with respect to the *ultimately attainable accuracy*, i.e., the magnitude where the *true* and *estimated* residual deviate from each other.

### 5.2.7 Biores/Biomin/Biodir

The method of choice amongst Biores, Biomin and Biodir is often *Biomin*, to most people known as 'the' BiCG method. The finite precision Biomin variant of BiCG has been abandoned by many researchers, because of the irregular convergence properties. No minimisation like in the SPD case is behind BiCG, thus even in infinite precision the method computes residuals, direction vectors and approximations of different magnitudes. This increases the size of the error matrix $F_k$.

A bound on the error term $F_k$ is given in the error analysis of finite precision BiCG (Biomin) algorithm by Tong and Ye (cf. [TY95, TY00]), where Tong and Ye re-scale and glue together the finite precision Biomin recurrence to obtain the usual form of a perturbed Hessenberg decomposition. They show that the columns of the error matrix $F_k$ in the *unscaled* form are bounded (componentwise) by

$$\frac{|f_i|}{\mathbf{u}} \leq ((n+6)|A| + \frac{1}{|\alpha_i|} + \frac{|\beta_i|}{|\alpha_{i-1}|})|r_i| + (2n+7)|A||p_i| + O(\mathbf{u}).$$

In the sequel, Tong and Ye go on to derive general applicable results, by no means restricted to the BiCG/Biomin case. Some of the basic results contained in their paper are stated in this thesis in a more general setting (Chapter 4, Corollary 4.9, page 147).

### 5.2.8 QOR/QMR

QOR as it would be the counterpart of FOM is seldom used. Mostly one of the (mathematical) equivalent Biores, Biomin and Biodir variants of BiCG are used.

Finite precision QMR shows better numerical properties than the BiCG implementations. This is due to the fact that we have at least a *quasi-minimisation* in contrast to BiCG variants applied to general (non-symmetric) matrices. The multiplication by the non-orthogonal basis makes is quite hard to give explicit results on the behaviour of finite precision QMR.

### 5.2.9 Look-Ahead

We shortly mention some impacts of the execution of the non-symmetric Lanczos method in finite precision with *look-ahead*. Look-ahead is usually applied when the inner products, or, more general, the smallest singular values of the actual blocks, are to close to zero. This, of course, results in a more stable variant of the non-symmetric Lanczos method. A small inner product (singular value) results in large components in the error vectors which diminishes the chances for a small ultimately attainable accuracy in the eigenvalue case, as well as in the linear system case.

In the look-ahead context, the 'final' strategy has not been found, like in the case of truncated and restarted methods based on Arnoldi. But, the state of the art in look-ahead is far more superior in look-ahead methods than in truncation and restart. Thus, we are not that far from a working running error analysis. Due to the indeterminacies underlying these methods, i.e., the choice of the *starting vectors*, the choice of *normalisation* and the choice of *optimal look-ahead parameters*, we may suspect that we will not succeed in constructing an *a priori* error analysis.

### 5.2.10 Lanczos-Type Product Methods

Thus far, no error analysis of LTPMs has evolved that gives substantial results. The error analysis will, in general, be involved. We might think of re-writing the recurrence equations in the form of a single perturbed Hessenberg decomposition. The entries in the upper triangular part of the Hessenberg matrix will mostly be *non-zero* and are algebraically defined by the recurrence coefficients. Similar terms will occur in the assembling of the error matrix $F_k$. Just putting these relations together already is hard work. Then, in a second step, relations between the eigenvectors and eigenvalues of these matrices and submatrices and the similar error matrices have to be devised. This might be even harder, if not intractable.

Nevertheless, the methods will work fine until the first Ritz pair of the assembled Hessenberg matrix has converged to a certain accuracy, determined by the size of the assembled error matrix in direction $\hat{v}_i$ and $s_j$. This point of deviation may be easier to predict in context of LTPMs.

### 5.2.11 CGNE/CGNR

When it comes to the application of CGN methods to non-symmetric systems of equations, we observe a *squaring of the condition number* in practice. When $A$ well-conditioned, these methods can be applied quite safely. The deviation will be like in the CG case, especially, the Orthomin variants of CGNE and CGNR will be more stable and will have ultimately attainable accuracy of smaller magnitude.

## 5.3 Krylov Methods and Preconditioning

We just give a few comments on how one might proceed in order to obtain a form of error analysis for *preconditioned* Krylov subspace methods.

First, consider that the preconditioner in use is *static*, i.e., is such that it does not change from step to step. In this case, we can incorporate the error of the

preconditioning scheme into the error matrix $F_k$. Without major changes, all error analysis results stated in this thesis apply. The *speed-up in the rate of convergence* is payed by (in general) a larger *ultimately attainable accuracy*.

In inexact Krylov methods we wish to use in the first steps a very accurate preconditioning, then we wish to relax as soon and as strong as possible. Here, application of general results on the successful completion of the methods in a finite precision environment may be fruitful in determining the relaxation parameters. For GMRES, for example, the results on the MGS variant by Rozložník suggest that the relaxation parameter can be chosen inverse proportional to the actual size of the residual.

When we consider inner-outer iterations, where both iterations are based on Krylov subspaces, we could (at least theoretically) consider them as a new Krylov method and design a new (more appropriate) form of error analysis. This area of methods is closely connected with the LTPMs, the Lanczos-type product methods, since some of them perform a mixture of GMRES and the non-symmetric Lanczos method. This point of view might be an alternative approach to a successful error analysis of LTPMs.

# Chapter 6

# Krylov Methods: Miscellanea

This chapter is devoted to some of the more philosophical aspects that arise when one is concerned with numerical analysis and especially with Krylov methods. First of all, the previous investigations reveal that Krylov methods and terms like backward and forward stability do not go hand in hand when we try to apply the usual definitions of these terms. The first part of the following comments is devoted to a brief treatment of how terms might be defined alternatively to assure a form of stability.

As the title of the thesis already suggests: many aspects discussed are heavily influenced by the execution of the algorithms in a *finite precision* environment. We consider the implementation of the methods in alternate arithmetics and apply the results of chapters three and four. Especially, we consider shortly the execution of the methods in *symbolic computations*, in a *multiple precision environment* and in a *variable precision environment*, the latter closely related to the hot topic of the so-called *inexact Krylov methods*.

A special kind of arithmetic is interval arithmetic. Interval arithmetic can be used to free the user from the necessity of supplying additional information and to built a black-box solver that indicates whether the algorithm can be used to achieve the goal indicated. The resulting algorithms are usually termed self-validating algorithms. We give the sketches of two self-validating algorithms for linear systems based on short-term methods.

## 6.1 Krylov Methods and Stability

The results of chapter four make it obvious that the terms backward error and stability are not applicable 'as is', i.e., new meanings of these terms have to be coined. Furthermore, Krylov methods compute a variety of quantities and several relations are fulfiled in infinite precision. We have to define precisely, *which computed quantities* we want to relate to *which unperturbed quantities*, and *how* we want to measure the occurring deviation. We just give a rough sketch of the quantities we are usually interested in, and those we are not interested in.

Krylov methods compute (at least)

- a basis of the Krylov space,
- a representation of $A$ on the space spanned by that basis,
- approximations to
  - eigenvalues,
  - invariant subspaces and
  - solutions to linear systems.

The relations used to define the methods hold approximately in a finite precision environment. This is mainly the Hessenberg decomposition, in case of split Hessenberg decompositions the residual and the direction vector recurrences.

In the beginning, Krylov methods were used to compute normal forms, i.e., tridiagonal and Hessenberg matrices, of the general matrices $A$ that are inserted into the methods. This computation can only be stable in general for long-term methods, since any convergence of a Ritz pair in a short-term method will cause a huge deviation from the exact tridiagonal matrix.

Then, Krylov methods were considered as a good choice for the computation of single eigenpairs and approximate solution to linear systems of equations. These computations are more stable, and we might look for methods that are backward stable in this sense. But, this does by *no means imply that the methods compute appropriate approximations*. The Ritz vectors that are good approximations in the backward sense returned by *one* run of *one* Krylov method of a symmetric matrix will not neccessarily be orthogonal, especially, when the corresponding eigenvalues are close to each other. Usually, they will be as worse as possible, i.e., the variation in the eigenvectors will be inverse proportional to the spectral separation of the eigenvalues.

When it comes to the solution of linear systems, we may ask, whether the approximate solutions returned in *finite precision* can be small perturbations of approximate solutions returned by an *infinite precision* Krylov method. For short-term methods this is seldom the case, in the long-term setting this can happen, for instance when we consider the Householder variant of GMRES. Nevertheless, we can ask for conditions, such that a finite precision Krylov method based on short-term recurrences returns eventually a backward stable solution. By *eventually* we think of a larger number of steps than *any* infinite precision short-term method would need.

### 6.1.1 *Can* Krylov Methods be Backward Stable?

All major aspects of the error analysis of problems related to *dense* matrices are more or less satisfactorily solved, when numerical analysts can show that the method is *backward stable*, i.e., the solution computed in the *final* step is the solution of a near-by problem. The problem of identifying 'useful' solutions is transferred to condition estimation. Under the given circumstances this is possible, since the algorithms are direct methods and the solution process directly is influenced by the conditioning of the problem. In context of Krylov methods, this, for two main reasons, no longer makes sense:

- Despite the fact that infinite precision Krylov methods are finite termination methods, they are almost never carried out this far. Krylov methods are carried out until a user-specified threshold has been reached, usually selected proportional to the *backward error*.
- The information *whether* the finite precision Krylov method reaches the desired level of accuracy is not that important. More important is the information, *when* this will be the case. In general, this information can only be retrieved when considering more general methods (Ritz-Galerkin type methods) instead of the investigated Krylov method.

Formulated in another fashion, Krylov methods proceed in two steps. The first step, the computation of a basis frequently is unstable. The second step, the computation of the desired approximations may still be stable.

### 6.1.2 *All* Krylov Methods are Forward Unstable

As mentioned, a Krylov methods first computes a basis and then computes the quantities related to the projection. When the first step is forward unstable, the method will be. The computation of the basis vectors is forward unstable, this follows readily from Chapter 4, Section 4.8. The computation of the basis vectors can only be forward stable as long as no Ritz pair has converged, compare Theorem 4.32:

$$\hat{v}_i^H q_{k+1} = \frac{(\lambda_i - \theta_j)\,\hat{v}_i^H y_j + \hat{v}_i^H F_k s_j}{c_{k+1,k} s_{kj}}.$$

It is not sufficient to predict the behaviour by bounding the error terms from above and below, because the resulting interval will blow up when the convergence estimator $c_{k+1,k}s_{kj}$ becomes small.

We can not be sure if there does not exist a Krylov method such that the method converges, but the convergence estimators remain relatively large. We consider Theorem 4.37, page 176 in a slightly rewritten form,

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}}\; q + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(A)}{\prod_{p=l}^{k-1} c_{p+1,p}}\; \frac{f_l}{c_{k+1,k}} \right].$$

It is obvious that the method becomes forward unstable when one $c_{k+1,k}$ becomes small, or one $f_l$ becomes large. In general, the method becomes unstable, when the error terms are amplified sufficiently by the polynomials of trailing submatrices.

### 6.1.3 Most Krylov Methods are *not* Backward Stable

Short-term methods do not compute inner products related to basis vectors (residual vectors, direction vectors) that have indices that differ greatly to each other. This leaves plenty of freedom in the variation of the vectors, such that orthogonality and $A$-conjugacy is lost *completely*. The only chance to cure the occurring loss of orthogonality lies in the computation of a new Ritz value that converges to the same eigenvalue. Thus these methods, even if they might compute backward stable *eigenpairs*, can, in the generic case, never compute a backward stable *basis* of the Krylov space and a corresponding *projection*.

Essentially the same holds true in methods that fail to ensure orthogonality of the basis vectors or other desired property. Here, some methods even fail to produce backward stable eigenpairs, like counterexamples obtained by application of the CGS variant of the Arnoldi method show.

### 6.1.4 *Which* Krylov Methods *are* Stable

There are some Krylov methods that *are* backward stable in the usual sense. These methods will, by nature, be long-term methods. The best example is the Householder variant of GMRES. When carried out $n$ steps, the solution computed is a backward stable solution under mild circumstances. In the Householder variant, the computed basis vectors share the property of orthogonality (up to machine precision) with the exact basis vectors. This correspondence is lost in the MGS variant of GMRES. But, even in this case, the solution computed seems to be backward stable. The same seems to hold true for GMRES variants based on *iterated* Gram-Schmidt.

### 6.1.5 *Where* Krylov Methods are Stable

Various authors have pointed out the existence of matrices (and starting vectors) that result in a stable computation of the Krylov subspace. By preceeding analysis

these matrices coincide with the matrices (and starting vectors) where the convergence is delayed to the final step. More general, some Krylov methods are stable when applied to matrices from some class, when a *generic starting vector* has been chosen. A generic vector is one that has non-negligible portion in direction of *all* eigenvectors and principal vectors. The matrices that can be safely used in finite precision Krylov methods, are those, whose eigenvalues do *not converge rapidly*, and the convergence estimators drop *simultaneously* below a certain threshold after some steps. To the authors knowledge, the existence of such matrices has been discussed for the first time in a paper by Scott and in Grcar's thesis (cf. [Sco79, Grc81]).

Scott explicitly constructed matrices for the symmetric Lanczos algorithm, such that convergence is delayed to the last step (cf. [Sco79]). The eigenvalues where distributed at Chebyshev points, i.e., he used matrices with eigenvalues given by the zeros of Chebyshev polynomials $\mathcal{C}_n$,

$$\mathcal{C}_n(x) = \begin{cases} \cos(n\mathrm{acos}(x)) & x \in [-1,1], \\ \cosh(n\mathrm{acosh}(x)) & x \notin [-1,1]. \end{cases}$$

A typical and prominent example is the matrix

$$T = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}$$

resulting from discretisation of the second derivative on the real axis. The class of matrices can be slightly generalised, one can use matrices with eigenvalues clustering quadratically at both ends of the spectrum, i.e., matrices with eigenvalues that are the result of a projection to the real axis of a perturbed equidistant distribution on the half circle.

For the non-symmetric orthogonal variant (e.g. Arnoldi) *normal* matrices of this class are those that have eigenvalues on a circle or an ellipse in the complex plane. When the eigenvalues are distributed equidistantly on the border of the unit circle, then the convergence is delayed to the last step.

When the matrix $A$ is given, one may ask if there exists a *starting* vector such that the convergence is delayed until the last step. Scott gave a partial answer for the case of $A = A^H$, i.e., symmetric Lanczos. In this case one can compute *explicitly* the starting vector corresponding to a selected distribution of Ritz values in the $n-1$th step. He concluded from some examples that the starting vectors with the property of delaying the convergence to the last step have very small components (absolute size $O(10^{-30})$) in some eigendirections. In finite precision execution it will be very unlikely that these components are *not* replaced by errors of order $O(10^{-16})$ when the first error vector has to be introduced.

### 6.1.6 Alternate Notions of Stability

The considerations thus far make it obvious that the usual forward, backward and mixed error analysis is not applicable to (short-term) finite precision Krylov subspace methods, at least 'as is'. Greenbaum has given a new notion of backward stability that better fits the framework of short-term Krylov subspace methods. Short-term Krylov methods frequently return accurate approximations to eigenpairs, but get in trouble when it comes to multiplicity and an appropriate choice of basis of the invariant subspaces of clusters. Thus we may think of the (finite dimensional) operator $A$ replaced by some other (finite dimensional) operator $\tilde{A}$ with not necessarily the same dimension, but with similar spectral properties. In the normal

case this is sufficient to understand the behaviour of finite precision Lanczos and CG methods, like was proven by Greenbaum (cf. [Gre89]) and demonstrated three years later by Greenbaum and Strakoš (cf. [GS92]). A corresponding interpretation as backward stability with enlarged 'patchwork' operator $\tilde{A}$ has not yet evolved for general non-normal $A$.

## 6.2 Krylov Methods and Other Arithmetics

This far, we only have considered Krylov subspace methods executed with the aid of *IEEE floating point arithmetic*. Iterative methods are (and necessarily will be) used with a growing amount. So we may ask, if the trouble lies not in the *methods* itself, but in the chosen computer arithmetic. We try to give a partial answer in considering some of the usual alternatives, i.e., exact and symbolic computation, multiple precision, variable precision, stochastic and interval arithmetic. The starting point is the choice that guarantees that *no errors at all* can occur, e.g. exact and symbolic computations.

### 6.2.1 Exact/Symbolic Arithmetic

The computation of the Hessenberg decomposition relies purely on *algebraic operations*. When we postpone the normalisation of the basis vectors to a second stage, or use a rational scaling, the computation of a (obviously differently scaled) Hessenberg decomposition relies only on *rational operations*. Thus, we may use integer arithmetic to compute an exact Hessenberg decomposition when the matrix and the starting vector are integers. This can be used to compute the minimal polynomial of the matrix $A$ with respect to the starting vectors. This was the basic idea behind Krylov's paper. Variants of this approach are still used in *Computer Algebra Systems*, short CAS. The Hessenberg decomposition may be used in further steps to compute the Frobenius normal form, to determine the rank of matrices and to determine the solution to linear systems.

These approaches naturally extend to fields other than $\mathbb{K} = \mathbb{Q}$, i.e., to finite fields like Galois fields. In this context, the algorithm of Wiedemann is frequently used to solve large sparse matrix equations. Given a matrix $A \in \mathbb{K}^{n\times n}$, this method computes a sequence of field numbers

$$a_i = u^T A^i q, \quad i \in \underline{k}$$

with randomly chosen $u \in \mathbb{K}^n$ (and $q \in \mathbb{K}^n$) and uses the *minimal linear generator* of this sequence to almost surely compute the minimal polynomial of $A$. This knowledge is used in a second stage to solve linear systems of equations.

Wiedemann's algorithm is, like every other Krylov method, only based on the *action* of $A$ on vectors. In contrast to the numerical analysis notion *matrix-free* methods, the computer algebra community sticks more closely to the term *black-box* methods. Wiedemann's algorithm has been generalised by Coppersmith to block form. Wiedemann's method is incorporated into several software products, we mention as example the method `linalg::wiedemann` in the CAS MuPAD (cf. [Fuc96]) and the implementation of Wiedemann's algorithm in LinBox (cf. [DGG$^+$]).

When the field of interest is $\mathbb{R}$ or $\mathbb{C}$, the execution of Krylov subspace methods using symbolic or exact computations makes the methods to slowly to be competitive for most applications.

### 6.2.2 Multiple Precision Arithmetic

There are several ideas how to use a given higher precision than double precision (so-called extended precision) or to implement quadruple or higher precision with

IEEE double precision (cf. [Knu98], Chapter 4). Regardless of implementational details, all error analysis applies to higher precision, since the error analysis was only based on the IEEE arithmetic rounding error model and on the machine precision. Nevertheless, the number of steps necessary to obtain a small residual might be reduced drastically, especially in case of outliers of eigenvalues, for example in the solution of elliptic PDE.

The analysis of preceeding chapters gives crude estimations on *how* accurate the machine precision has to be when we *prescribe* an eigenvalue distribution (or only the condition number) and a level of accuracy we want to reach. When we have a lower bound $\rho^k$ on the rate of convergence of the fastest converging Ritz pair,

$$\left(\lambda_i - \theta_j^{(k)}\right) \hat{v}_i^H y_j^{(k)} \geq \rho^k,$$

we can assure that the infinite precision properties carry over to the multiple precision properties in the first $m$ steps when the machine precision is sufficiently small,

$$\rho^m \gg \epsilon \quad \Rightarrow \quad \hat{v}_i^H q_{k+1} \approx \frac{\left(\lambda_i - \theta_j^{(k)}\right) \hat{v}_i^H y_j^{(k)}}{c_{k+1,k} s_{kj}^{(k)}} \quad \forall\ k \in \underline{m}.$$

When the (multiple) machine precision only is *small*, the convergence scenario still may *change drastically*. This is due to the underlying non-linearity of the relations between *convergence* and *deviation*. So, it may be advantageous to figure out the precision that *increases* the speed of convergence without *spoiling* the execution time due to the implementational aspects to much. This has been experimentally investigated by Facius in his thesis dating to 2000 (cf. [Fac00]). Facius uses the notion *variable precision* in favour to *multiple precision* to make clear that the methods can be executed in *arbitrary*, but *fixed* precision.

### 6.2.3 Variable Precision Arithmetic

Another idea is to *change the precision in every step*. This is an idea that stems from the field of *inexact Krylov methods*. In inexact methods, accurate matrix vector multiplies are (very) costly. So, to decrease the overall cost of the algorithms, variable precision is used to compute the matrix vector multiply only as accurate as necessary.

Usually, this has nothing to do with the *machine* precision, but comes from the characteristics of the problem. An example of such costly matrix vector multiplies comes from *Quantum Chromodynamics* (QCD), where one has to multiply vectors with the *sign* of a matrix. Nevertheless, this makes sense when the given matrix is not accurately representable with the aid of the floating point numbers of the machine precision. Then, the precision in the beginning has to be high (e.g. quadruple precision), but this constraint may be *relaxed* once the method has *started to converge*.

### 6.2.4 Stochastic Arithmetic

Stochastic arithmetic utilises the idea that errors due to finite precision computations are 'more or less' random. An operation using stochastic arithmetic is based on switching the rounding mode of the machine and returning the distribution that models the behaviour. When applied to Krylov subspaces methods, this reveals the sensitivity of the method. This is particularly useful in linear system solvers, since this kind of arithmetic reveals when the ultimately attainable accuracy has been reached. A forward stochastic approach known as the CESTAC method has

been developed in 1974 by La Porte and Vignes (cf. [LV74]). A backward stochastic approach known as PRECISE has been implemented by Chaitin-Chatelin and co-workers (cf. [CCF96]).

### 6.2.5 Interval Arithmetic

Interval arithmetic also is concerned with control over errors, but is by no means restricted to floating point errors. Interval arithmetic is based on some ideas by Kahan, was introduced in Moore's book and was propagated by Kulisch and Miranker (cf. [Moo66, KM81]). Implementations of interval arithmetic include the (X)SC packages developed by researchers at Karlsruhe and Wuppertal (cf. [XSC]) and PROFIL/BIAS by Knüppel (cf. [Knü94]). Matlab add-ons include `b4m` (BIAS for Matlab) by Zemke (cf. [Zem99]) and Intlab by Rump (cf. [Rum98]). The prime source for interval arithmetic software and related areas is the Internet address (cf. [Int]) of the reliable computing mailing list (cf. [Rel]).

In interval arithmetic, instead of working with real numbers $a \in \mathbb{R}$, we work with (real) intervals $[a]$. A (real) interval $[a] \subset \mathbb{R}$ is defined by

$$[a] \;\equiv\; [\underline{a}, \overline{a}], \qquad \underline{a}, \overline{a} \in \mathbb{R}, \qquad \underline{a} \;\leq\; \overline{a}.$$

When $\underline{a} = \overline{a}$, such that $[a]$ contains just one single value $\underline{a}$, we talk of a *degenerate interval*. The real numbers are *identified* with degenerate intervals. Thus, the space of real numbers $\mathbb{R}$ is embedded in the space $\mathbb{IR}$ of all intervals,

$$\mathbb{R} \subset \mathbb{IR} \equiv \big\{[a],\, [a] \text{ is an interval}\big\}.$$

Intervals are used to *bound* errors. Thus, as a natural extension of the functions $\circ \in \{+, -, *, /\}$ defined on elements in $\mathbb{R}$, the corresponding interval operations are defined by

$$[a] \circ [b] \;=\; \left[\min\left\{\underline{a} \circ \underline{b}, \underline{a} \circ \overline{b}, \overline{a} \circ \underline{b}, \overline{a} \circ \overline{b}\right\}, \max\left\{\underline{a} \circ \underline{b}, \underline{a} \circ \overline{b}, \overline{a} \circ \underline{b}, \overline{a} \circ \overline{b}\right\}\right].$$

There are some remarkable facts about this way of computing with intervals. The most prominent property is the *inclusion property*. This property states that when $a \in [a]$ and $b \in [b]$, then also $a \circ b \in [a] \circ [b]$. The bad news is that the algebraic properties are lost to some extent. In general only

$$0 \in [x] - [x] = [-\mathrm{diam}([x]), \mathrm{diam}([x])] \quad \text{and} \quad 1 \in [x]/[x].$$

This is known as the *dependency problem*, since interval arithmetic treats the *same* interval as if it were *two different* intervals. The quantity $\mathrm{diam}([x])$ is the *diameter* of the interval $[x]$.

The distributivity law has to be weakened to the so-called *sub-distributivity*:

$$[a]\,([b] + [c]) \subset [a][b] + [a][c].$$

The algebraic properties are lost *as soon* as we are working with non-degenerate intervals.

There are *two* main approaches to represent intervals. One is the so-called infimum-supremum form

$$[a] = [\underline{a}, \overline{a}] = \{x \in \mathbb{R},\, \underline{a} \leq x \leq \overline{a}\},$$

the other the so-called midpoint-radius form

$$\langle a \rangle = \langle m, r \rangle = \{x \in \mathbb{R},\, |x - m| \leq r\}$$

of intervals. The radius $r$ necessarily is a non-negative number. When they are used to represent real line segments, they are equivalent representations. The midpoint-radius form of the interval $[a] = [\underline{a}, \overline{a}]$ is given by

$$\langle a \rangle = \langle \mathrm{mid}([a]), \mathrm{rad}([a]) \rangle,$$

where

$$\mathrm{mid}([a]) \equiv \frac{\overline{a} + \underline{a}}{2}, \qquad \mathrm{rad}([a]) \equiv \frac{\mathrm{diam}([a])}{2} = \frac{\overline{a} - \underline{a}}{2}$$

denotes the *midpoint* and the *radius* of the interval $[a]$, respectively.

The representation of subsets of $\mathbb{C}$ changes depending on the use of the former or the latter. The former, i.e., the infimum-supremum form is based on a partial ordering in $\mathbb{C}$ induced by the isomorphism to $\mathbb{R}^2$. This representation results in *boxes* in the complex plane. The midpoint-radius form carries over without changes, the result being *circles* in the complex plane. In either case, the resulting set of subsets ('intervals') of $\mathbb{C}$ is denoted by $\mathbb{IC}$.

This definitions are extended similarly to vectors $v \in \mathbb{K}^n$ and matrices $A \in \mathbb{K}^{n \times m}$. An interval vector in infimum-supremum form for instance, is defined by

$$[v] \equiv [\underline{v}, \overline{v}] = \{x \in \mathbb{K}^n, \underline{v} \leq x \leq \overline{v}\}, \qquad \underline{v}, \overline{v} \in \mathbb{K}^n.$$

The less equal sign is interpreted componentwise. This results in the interval vectors being boxes in $\mathbb{K}^n$. The set of all interval vectors is denoted by $\mathbb{IK}^n$. An interval matrix in midpoint-radius form looks like

$$\begin{aligned} \langle A \rangle &\equiv \langle M, R \rangle, \qquad M \in \mathbb{K}^{n \times m}, R \in \mathbb{R}^{n \times m}_{\geq 0}, \\ &= \{X \in \mathbb{K}^{n \times m}, |X - M| \leq R\}. \end{aligned}$$

Absolute value and less equal sign again have to be interpreted componentwise.

In higher dimensions a second serious problem of naïve interval arithmetic occurs, the so-called *wrapping problem*. It occurs even when we compute the *narrowest* (infimum-supremum) interval inclusion of $A \cdot [v]$, where $A \in \mathbb{K}^{n \times n}$ and $[v] \in \mathbb{IK}^n$. Suppose that $A$ is unitary, such that the SVD of $A$, $A = U \Sigma V^H = U$ is very simple. Then the multiplication of $[v]$ with $U$ more or less *rotates* the *box* $[v]$. When we apply powers of $A$ to an interval vector by forming $A^k \cdot [v] = A \cdot (A^{k-1} \cdot [v])$, the problem becomes obvious and more serious, see the illustration drawn in figure 6.1.

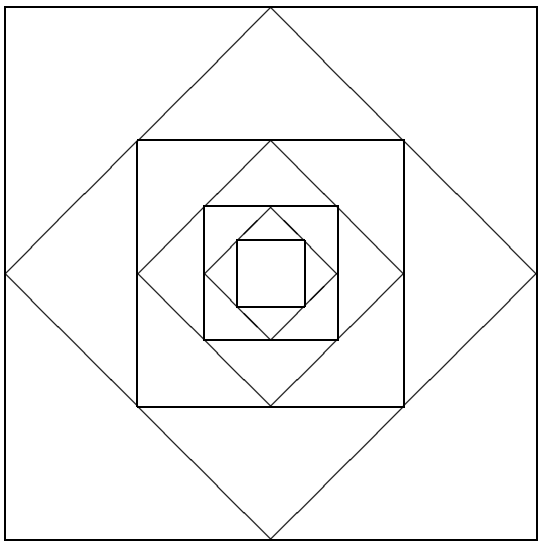

Figure 6.1: The wrapping effect in $\mathbb{R}^2$

Even though both representations (denoted by *inf-sup* and *mid-rad*) can be used to compute enclosures of solutions, there are two major differences, concerning the *implementational details* and the *achievable accuracy*. A standard implementation of a mid-rad interval arithmetic, as example consider the implementation of a mid-rad arithmetic in the Matlab add-on Intlab (cf. [Rum98]) by Rump, overestimates the sharp bounds returned by an inf-sup interval arithmetic by a factor of at most 1.5. More precise, overestimation occurs only in case of multiplication (and division, since division is based on multiplication). The overestimation is not too large when we are interested in the solution of *point* problems, i.e., problems defined with *degenerate* interval data.

We have been a little bit to sloppy in the considerations thus far to talk of the implementational aspects. The by far most important aspect is that the endpoints can not be elements of $\mathbb{R}$ ($\mathbb{R}^n$, $\mathbb{R}^{n\times m}$). The domain of elements is naturally restricted to elements from $\mathbb{F}$ ($\mathbb{F}^n$, $\mathbb{F}^{n\times m}$). So, in the computation of the endpoints (midpoints, radii), we are restricted to floating point numbers. In order to fulfil the inclusion property, we have to ensure that the endpoints are *rounded correctly*, i.e., we have to achieve outward rounding. Switching rounding is part of ANSI/IEEE 754, but the implementation still is a somewhat non-trivial task, because there is no direct support from high level languages and compilers. The rounding mode switches have frequently to be coded in assembler language.

One of the bottlenecks in (self-validating) algorithms based on interval arithmetic lies in the (interval) matrix times (point) matrix multiplication. We give the aspects of three different implementations of matrix times matrix when we want to compute an enclosure of the product $[C] \leftarrow R \cdot [A]$ where $R$ is a *point matrix*, i.e., composed of *degenerate* intervals. The example and the pseudo-code for the three implementations originates from slides of a talk given by Rump.

The implementations are given in a Matlab-style pseudo-code. The routines `SetRoundDown` and `SetRoundUp` switch the rounding mode to round towards *minus infinity* (down) and towards *plus infinity* (up), respectively. The *naïve implementation* (inner loop: single operations) is given as Algorithm 6.1. The more sophisticated approach by Knüppel that is used in PROFIL/BIAS moves the rounding

```
input : [A]=[Ainf,Asup],R
output: [C]=[Cinf,Csup]

for i=1:n
   for j=1:n
      Cinf(i,j)=0;
      Csup(i,j)=0;
      for k=1:n
         if R(i,k)>=0
            SetRoundDown
            Cinf(i,j)=Cinf(i,j)+R(i,k)*Ainf(k,j);
            SetRoundUp
            Csup(i,j)=Csup(i,j)+R(i,k)*Asup(k,j);
         else
            SetRoundDown
            Cinf(i,j)=Cinf(i,j)+R(i,k)*Asup(k,j);
            SetRoundUp
            Csup(i,j)=Csup(i,j)+R(i,k)*Ainf(k,j);
         end
      end
   end
end
```

Algorithm 6.1: Naïve implementation of $[C] \leftarrow R \cdot [A]$

mode switches out of the inner loop (new inner loop: scalar products). This enables the use of BLAS level 1. The resulting algorithm is given as Algorithm 6.2. This algorithm is substantially *faster* than the first, naïve implementation.

The PROFIL/BIAS library (cf. [Knü94]) is known to be very fast when compared to other available interval arithmetic software libraries. The key to fastness is to use rounding mode switches as seldom as possible and to use higher level (BLAS) routines as often as possible. The shifting of rounding mode switches makes it possible to use optimisation switches (i.e., `-O2`, or even `-O3` when the compiler is part of the GCC, the Gnu Compiler Collection, cf. [GNU]) in generating the code. The BLAS, the building blocks of numerical linear algebra are based on *pure floating point* computations without *any rounding mode switches*. Thus, an occurrence of a rounding mode switch inside a loop makes the use of BLAS routines impossible.

The next step would be to move the rounding mode switches out of the next loop, this would result in a rank-one update formula. We will not consider this approach. Instead we give the algorithm for the computation of an enclosure of the matrix equation $\langle C \rangle \leftarrow R \cdot \langle A \rangle$ in mid-rad interval arithmetic. The algorithm is given as Algorithm 6.3.

As this example shows, the mid-rad interval arithmetic approach enables the use of higher levels, i.e., of BLAS level 3 instead of level 1 or level 2. The occurrence of rounding mode switches is diminished. Far more interesting is that *any* level 3 BLAS routines can be used to implement this style of mid-rad interval arithmetic. The key property is that the error analysis is only based on the *number and type* of the atomic operations that are involved, but not on the *order* in that they appear. Thus, we might use any vendor-supplied BLAS. The details how to implement such a fast and portable interval arithmetic can be found in an article by Rump in BIT (cf. [Rum99]).

```
input : [A]=[Ainf,Asup],R
output: [C]=[Cinf,Csup]

for i=1:n
   for k=1:n
      Cinf(i,:)=0;
      Csup(i,:)=0;
      if R(i,k)>=0
         SetRoundDown
         Cinf(i,:)=Cinf(i,:)+R(i,k)*Ainf(k,:);
         SetRoundUp
         Csup(i,:)=Csup(i,:)+R(i,k)*Asup(k,:);
      else
         SetRoundDown
         Cinf(i,:)=Cinf(i,:)+R(i,k)*Asup(k,:);
         SetRoundUp
         Csup(i,:)=Csup(i,:)+R(i,k)*Ainf(k,:);
      end
   end
end
```

Algorithm 6.2: PROFIL/BIAS-style implementation of $[C] \leftarrow R \cdot [A]$

```
input : ⟨A⟩=⟨Amid,Arad⟩,R
output: ⟨C⟩=⟨Cmid,Crad⟩

SetRoundDown
Cmid_inf=R*Amid;
SetRoundUp
Cmid_sup=R*Amid;
Cmid_rad=0.5*(Cmid_sup-Cmid_inf);
Cmid=Cmid_inf+Cmid_rad;
Crad=(Cmid-Cmid_inf)+|R|*Arad;
```

Algorithm 6.3: Intlab-style implementation of $\langle C \rangle \leftarrow R \cdot \langle A \rangle$

### Numerical Result Verification

Interval arithmetic has several applications. The most important seem to be *global optimisation* and the *verification* of results that have been computed by usual floating point algorithms. We only consider the latter. Ideally, the verification phase should form a second stage of an existing floating point 'solution' process and should at least be comparable with respect to time and storage amount. It turns out to be a good idea to use *every* piece of information gained in stage one, the floating point solution. In case of *dense* linear systems these restrictions can be met, as is well known.

Now, we consider some of the aspects of the verification phase. The verification phase is usually based on some form of fixed point iteration. The idea is to consider in place of the original system $Ax = b$ a preconditioned system. The preconditioning is based on the best (floating point) knowledge at hand, i.e., a preconditioner $R \approx A^{-1}$ (a floating point inverse) and an approximate solution $\tilde{x}$ (a floating point solution). The new system takes the form

$$RAy = R(b - A\tilde{x}), \qquad y = x - \tilde{x}.$$

This equation is re-written in a fixed point form (comparable to the key idea behind

Richardson iteration or many Newton-type iterations),

$$y = R\underbrace{(b - A\tilde{x})}_{\approx 0} + \underbrace{(I - RA)}_{\approx 0}\, y.$$

Now, using for instance inf-sup interval arithmetic, we set $[A] = [A, A]$ and chose an initial guess $[y]$ and compute an enclosure $[w]$ of

$$R\,(b - [A]\tilde{x}) + (I - R \cdot [A])\,[y].$$

Brouwer's fixed point theorem ensures that when $[w] \subset [y]$, a fixed point $y \in [y]$ exists such that

$$R\,(b - A\tilde{x}) + (I - RA)\,y = y.$$

Furthermore, when $[w]$ is included in the *interior* of $[y]$, we can not have a *ray* of fixed points. Thus, the condition that $[w]$ is in the interior of $[y]$,

$$R\,(b - [A]\tilde{x}) + (I - R \cdot [A])\,[y] \overset{\circ}{\subset} [y], \tag{6.1}$$

ensures that $A$ and $R$ must be regular, and thus we have computed an enclosure of the true solution $x$, $x \in \tilde{x} + [y]$. When the computed interval is not included in the interior of $[y]$ we set $[y] \equiv [w]$ and start over. To work properly, so-called *epsilon-inflation* has to be used to ensure that in case of convergence the inclusion occurs.

The self-validating algorithm we have just sketched can be modified to fit any special purposes. One frequent reformulation utilises the perception that the approximate inverse $R$ need not been given explicitly, but can be represented by any routine that computes the inverse. This gives rise to variants based on the LU, Cholesky and other decompositions.

The operation count and storage requirement for these approaches lies between $O(n)$, $O(n^2)$ in special cases (SPD matrices, banded matrices) and $O(n^3)$. Operation counts of oder $O(n^3)$ (and mostly also $O(n^2)$) are prohibitive for large, sparse matrices. Current approaches to self-validating algorithms are based on decompositions and no longer work when the decomposition no longer can be computed.

We consider one special generalisation of the class of self-validating algorithms. The approach to this generalisation is based on forcing special structure in the correction interval $[y]$. The correction interval $[y]$ is chosen to be symmetric to the origin and has all entries of one magnitude, $[y] = \mu[-e, e]$. We consider point problems, i.e., $[A] = A$. With this special choice of correction intervals, the inclusion condition (6.1), when considered in original form without the refinement that the interval vector $[w]$ should be contained in the *interior* of $[y]$, is equivalent to

$$|R\,(b - A\tilde{x})\,| + |I - RA|\,e\mu \leq e\mu.$$

We can try to compute the preconditioner $R$ such that $|I - RA|\,e < e$ holds true and set

$$\mu \equiv \max_i \frac{|R\,(b - A\tilde{x})\,|_i}{(e - |I - RA|\,e)_i}. \tag{6.2}$$

Then, the true solution $x$ is included in the interval $\tilde{x} + \mu[-e, e]$.

### A Parameter-Dependent Verification Method

Over the years, a group of researchers around Golub, including Dahlquist, Fischer, Meurant and Strakoš have utilised the Lanczos algorithm and its connection to Gauss quadrature to develop algorithms that measure the error of the approximate solution (cf. [DEG72, DGN79, FG93, GS94a, GS94b, GM94a, GM94b, GM97]).

Frommer and Weinberg have used this idea to implement a self-validating algorithm (cf. [FW98]). The method is not restricted to SPD $A$. When $A$ is non-SPD, an augmented system is used.

In their original paper, Golub and Dahlquist used the representation of the symmetric Lanczos algorithm as Gaussian quadrature to derive error bounds (using Gaussian quadrature with prescribed nodes, i.e., Gauss-Jordan and Gauss-Legendre quadrature) and error estimators (simply Gaussian quadrature) by invoking Lanczos on the residual. The error can be estimated quite satisfactorily, and if the smallest singular value $\sigma_{\min}$ is known (or a lower bound), the method can be implemented using interval arithmetic to deliver bounds that are better than the simple bound

$$\|x - \tilde{x}\| = \|A^{-1}r\| \leq \|A^{-1}\|\|r\| = \frac{\|r\|}{|\sigma_{\min}|}.$$

Due to the forward instability of the Lanczos algorithm and interval arithmetic related problems (namely the wrapping effect) the bounds returned can not gain very much. They are frequently better by a factor of size approximately 10. A field of application lies in Tikhonov regularisation, since the smallest singular value of the regularised problem is well-known. This verification step based on the knowledge of the smallest singular value is implemented in Facius' software package (cf. [Fac00]).

**A Parameter-Free Verification Method**

In his 1998 thesis (cf. [Sin98]) Sinn gave an example of how to use the quantities computed in the floating point run of a Krylov subspace method in a self-validating algorithm. His approach was based on an approximate inverse implicitly defined by the recurrence polynomial. The approach was stated only for symmetric short-term recurrences based on Orthores, i.e., for CR-Ores. We give a sketch of the algorithmic details along with some generalisations. The method can be applied also with the aid of *unsymmetric* short-term recurrences, i.e., with Biores.

Sinn used the fact that in infinite precision the residuals converge to zero, $r_k = \rho_k(A)r_0 \to 0$. Under mild circumstances that mainly pose some additional constraints on the starting vector of the recurrence (the first residual) the sequence of polynomials also converges to zero, $\rho_k(A) \to 0$. The residual polynomials are polynomials with constant term equal to one, thus we can write them in the form

$$\rho_k(A) = I - \iota_{k-1}(A)A \to 0, \qquad \text{which implies} \qquad \iota_{k-1}(A) \to A^{-1}.$$

They key idea behind Sinn's approach is that in case the residual polynomials $\rho_{k+1}$ satisfy a *three-term recurrence*,

$$\beta_k\rho_{k+1}(A) = (A - \alpha_k I)\,\rho_k(A) - \gamma_{k-1}\rho_{k-1}(A)$$

also the polynomials $\iota_k$ satisfy some sort of three-term recurrence, this time an inhomogeneous three-term recurrence with *additional terms* given by the identity matrix $I$ and multiples of $A^{-1}$,

$$\beta_k\iota_k(A) = (A - \alpha_k I)\,\iota_{k-1}(A) - \gamma_{k-1}\iota_{k-2}(A) - I + (\alpha_k + \beta_k + \gamma_{k-1})\,A^{-1}.$$

This follows by replacing every $\rho_j(A)$ by $I - \iota_{j-1}(A)A$, sorting the resulting terms and multiplication by $A^{-1}$ from the right.

When the three-term is of Orthores type, the recurrence coefficients sum to zero, i.e., they satisfy

$$\alpha_k + \beta_k + \gamma_{k-1} = 0.$$

This removes the term involving the unknown $A^{-1}$ and makes the three-term recurrence computable. We are left with the 'simpler' recurrence

$$\beta_k\iota_k(A) = (A - \alpha_k I)\,\iota_{k-1}(A) - \gamma_{k-1}\iota_{k-2}(A) - I$$

for the approximate inverse $R \equiv \iota_m(A)$ for some (heuristically) chosen $m$. These approximate inverses can not be computed *explicitely* since in general they will be full matrices and we can usually not afford to store all $n^2$ entries.

The key is that when we use verification routines based on equation (6.2) or similar, we only need the *rows* $e_i^T R$ of the approximate inverse. For the *numerator* of the fraction in equation (6.2) this is obvious, the denominator can be bounded using

$$(|I - RA|e)_i = e_i^T |I - RA|e = |e_i^T - (e_i^T R)A|e.$$

The remaining task is to compute the rows of the approximate inverse $R \equiv \iota_m(A)$ for some chosen $m$. When $A$ is symmetric, the $j$th *row* $e_j^T R_k \equiv e_j^T \iota_k(A)$ is equal to the transposed $j$th *column* $R_k e_j \equiv \iota_k(A)e_j$, since polynomials in symmetric (Hermitian) $A$ are again symmetric (Hermitian). Thus, the recurrence for the columns of $R$ (the transposed rows) is given by

$$\beta_k(\iota_k(A)e_j) = (A - \alpha_k I)\,(\iota_{k-1}(A)e_j) - \gamma_{k-1}(\iota_{k-2}(A)e_j) - e_j.$$

Sinn only considered symmetric $A$ and gave a fairly good way of choosing $m \leq k$ with the knowledge available after the residual $r_k$ is small enough. Since $A$ is symmetric, short-term recurrences are the natural choice amongst the available Krylov subspace methods to compute approximate solutions.

When $A$ is *not* symmetric, we still can use a three-term recurrence, i.e., Biores. The problem is that the rows of the inverse are no longer the transposed columns. The solution to this problem lies in the evaluation of the recurrence with the transpose of $A$. This is based on the observation

$$\begin{array}{rccccc} & \rho_k(A) & = & 1 - \iota_{k-1}(A)A & \to & 0 \\ \Leftrightarrow & \rho_k(A^T) & = & 1 - \iota_{k-1}(A^T)A^T & \to & 0. \end{array}$$

The resulting evaluation of the *columns* of the approximate inverse $R^T \equiv \iota_m(A^T)$ of $A^T$ for some $m$ is given by

$$\beta_k(\iota_k(A^T)e_j) = \left(A^T - \alpha_k I\right)(\iota_{k-1}(A^T)e_j) - \gamma_{k-1}(\iota_{k-2}(A^T)e_j) - e_j.$$

Numerical experiments show that even though the unsymmetric variant is less well justified, the method behaves comparable to the symmetric variant.

In general, the algorithm proceeds as follows. First, compute recurrence coefficients and an approximate solution $x_k$ with a small residual (for example $10^{-10}$) using any floating point short-term Orthores variant. Then, verify the residual using interval arithmetic. Chose the appropriate $m$ for the trial approximate inverse using some heuristic such that the residuals $e_j^T - e_j^T RA$ will be moderately small, for instance less than 0.9 in maximum norm. Usually, this implies $m \ll k$. Compute the rows of the approximate inverse one after the other using the same storage space. The overall cost will be $O(mnp)$, where $p \geq n$ denotes the numbers of non-zeros of sparse $A$, i.e., when $m$ is moderate, the cost is dominated by $O(n^2)$ which makes the method costly compared to the floating point cost of computing an approximate solution.

The methods can, in principle, be extended to long-term recurrences. This seems worthless, since the resulting operation count will be even higher and such comparable to self-validating methods based on *direct approaches*. Short-term methods can be transformed to the Orthores form by a scaling under some circumstances. A possible diagonal scaling matrix $D = \mathrm{diag}(y)$ can be computed as the (unique) solution of the system $y^T T_m = e_m$, where $T_m$ is the tridiagonal matrix from the underlying Hessenberg decomposition. When the elements in $y$ are sufficiently fast away from zero, the recurrence coefficients necessary for the computation of the approximate inverse can be computed using $y$ and $T_m$ with negligible extra cost.

Extensions that seem a little more promising are considered in part in Sinn's thesis. The main idea is to use approximate inverses defined implicitly as polynomials in $A$. We state three alternate approaches from Sinn's thesis.

The first approach is based on the Ritz values $\theta_j$. We can explicitly represent the polynomial $\iota_{k-1}(z)$ in terms of Ritz values, since we know by definition that

$$\iota_{k-1}(z) = \frac{1-\rho_k(z)}{z}.$$

Again by definition, the polynomial $\iota_{k-1}(z)$ has maximal degree $k-1$ and we know that the evaluation at the $k$ Ritz values $\theta_j$, $j \in \underline{k}$ gives

$$\iota_{k-1}(\theta_j) = \frac{1-\rho_k(\theta_j)}{\theta_j} = \frac{1}{\theta_j} \equiv \zeta_j.$$

These data pairs define a *unique* interpolation polynomial of maximal degree $k-1$,

$$\mathcal{L}\left[1/z\right](\zeta) = \sum_{j=1}^{k} \frac{\prod_{i\neq j}(\zeta-\theta_i)}{\prod_{i\neq j}(\theta_j-\theta_i)}\zeta_j.$$

This polynomial thus *must be* the polynomial $\iota_{k-1}$ itself. The approximate inverse can now be computed using divided differences (the Newton form of the interpolation polynomial) in a stable manner. This might be based on *Leja points* or similar. This corresponds to a different evaluation of the polynomials $\iota_{k-1}$ which might be more stable.

The second approach is based on the computation of the desired rows of the approximate inverse by the approximate solution of the $n$ systems $A^T r_j = e_j$ to sufficient accuracy (residual norm less than 0.9) using any Krylov method available. This is an obvious choice but slightly more costly than the three-term recurrence approach for $\iota_{k-1}$, since we throw away the recurrence coefficients computed in previous runs.

The third approach is based on Chebyshev acceleration. The three-term recurrence for the residual polynomials using Chebyshev points on a prescribed region in the complex plane (here an interval on the real axis) is transformed to an inhomogeneous three-term recurrence for the approximate inverse polynomial like before. The choice of the region might be based again on previously computed Ritz values. More general, *any polynomial acceleration* method can be used and transformed to a recurrence for approximate inverses.

All three approaches might be optimised with respect to storage and computing time, since we have to solve (to comparable low accuracy) many systems of equations. This brings in, naturally, *block Krylov subspace methods*.

# Chapter 7

# Conclusion and Outview

In this thesis we have tried to gather existing knowledge related to the error analysis of (finite precision) Krylov subspace methods. In the presentation, we have tried to treat the methods such that the common characteristics become obvious. We have mainly focussed on an approach that rests upon eigenvalues and eigenvectors of the matrix $A$ and the condensed Krylov subspace representation $C_k$.

The approach is partially along the lines of Paige's analysis of the symmetric Lanczos method. Paige considered only the eigenvalues and eigenvectors of the computed condensed matrices, which, of course, is justified because of the removal of the matrix $A$ from the error relations. One of the main points in Paige's analysis lies in the relation

$$\left(\hat{y}_j^{(k)}\right)^H q_{k+1} = \frac{\left(\hat{y}_j^{(k)}\right)^H A y_j^{(k)} - \left(\hat{y}_j^{(k)}\right)^H y_j^{(k)} \theta_j^{(k)} + \left(\hat{y}_j^{(k)}\right)^H F_k s_j}{\beta_k s_{kj}^{(k)}} = \frac{\epsilon_{jj}^{(k)}}{\beta_k s_{kj}^{(k)}},$$

which clearly reveals that the loss of orthogonality is in direction of **(left) Ritz vectors** of *converging* (right) Ritz pairs. This is Theorem 4.18, page 154. The nice thing about this equation is that we can bound the quantities $\epsilon_{jj}^{(k)}$, at least in the symmetric case.

Our new analysis focused on the similar relation

$$\hat{v}_i^H q_{k+1} = \frac{\left(\lambda_i - \theta_j^{(k)}\right) \hat{v}_i^H y_j^{(k)}}{c_{k+1,k} s_{kj}^{(k)}} + \frac{\hat{v}_i^H F_k s_j^{(k)}}{c_{k+1,k} s_{kj}^{(k)}}.$$

that holds in general for *all* Krylov subspace methods. This relation clearly reveals that a deviation occurs in direction of **(left) eigenvectors** of *converging* (right) Ritz pairs. This is Theorem 4.32, page 172. This time, the numerator can not be bounded this nicely. Nevertheless, the gain lies in the new point of view that compares fixed (unknown) quantities with the computed ones.

Paige used and refined the eigenvector – eigenvalue relations obtained by Thompson and McEnteggert. We gave new, comparable results on *general* eigenvector – eigenvalue relations which turn out to be particularly useful in case of Hessenberg matrices. We showed how the new relation can be re-written using these relations and the theory of polynomial interpolation. We gave some numerical experiments explaining in detail the occurrence of multiple Ritz values and the delay in the convergence.

We have not investigated and used similar relations based on singular values and singular vectors of the matrices $C_k$ and, more interesting, $\underline{C}_k$. An analysis of this type may be more insightful when we want to discuss the aspects of Krylov methods of OR and MR-type for the approximation of the solution of a linear system.

We have enlightened several aspects of Krylov subspace methods that usually are only presented with some kind of mystique. The *general* approach based on the presentation of the floating point algorithm as perturbed Krylov/Hessenberg decomposition without referring to extensive error analysis shows:

- *All main aspects* and the *characteristic behaviour* can be understood in full generality *without the necessity* to discuss
  - implementational details,
  - algorithmic variants,
  - different kinds of error models.
- The necessity to *distinguish* between long-term, short-term and product-type Krylov subspace methods and to *group* the error analysis accordingly.
- The differences in the error analysis between three-term and coupled two-term methods lies in the *relative* form of the error matrix of the Hessenberg decomposition in case of coupled recurrences.
- Krylov methods, especially short-term Krylov methods, can not be treated with the aid of forward, backward and mixed error analysis of classical type. New terminology and a new notion of stability have to be coined.
- *Every* Krylov subspace method has the same *limitations*, namely that the method will
  - return backward stable eigenpairs or approximate solutions only as long as the error term in the scaled Hessenberg decomposition describing the recurrence is moderate,
  - compute multiple eigenpairs when the method is based on *short-term* recurrences and orthogonality is *not forced* by (full or partial) re-orthogonalisation.

These comments summarise the limitations of finite precision Krylov methods. There is room left for improvement of results. We mention the Lanczos-type product methods that have no (refined, appropriate) error analysis up to now. Little room is left in case of basic methods, here mostly in case of the non-symmetric Lanczos method. The main work lies in determining the 'best' variant with respect to stability. Plenty of room is left for improvement with respect to *new* methods closing the gap between short-term and long-term methods, i.e., methods based on truncation, restart and look-ahead, and their error analysis.

The new point of view that 'sees' finite precision Krylov methods as perturbed matrix equations, has resulted in a new point of view concerning the stability. We interprete *one finite precision Krylov method* as an overlap of *several Krylov subspace methods* with a wrong normalisation. This becomes apparent when we take a closer look at the construction of the basis vectors, in terms of the *actual Ritz values* this construction is described in Theorem 4.37, page 176:

$$q_{k+1} = \frac{\chi_{C_k}(A)}{\prod_{p=1}^{k} c_{p+1,p}}\, q + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(A)}{\prod_{p=l+1}^{k} c_{p+1,p}} \frac{f_l}{c_{l+1,l}} \right],$$

and in terms of the *actual OR residuals* this construction is described in Corollary 4.48, page 194:

$$q_{k+1} = \frac{\chi_{C_k}(0)}{\prod_{p=1}^{k} c_{p+1,p}} \left( \frac{r_k}{\|r_0\|} \right) + \sum_{l=1}^{k} \left[ \frac{\chi_{C_{l+1:k}}(0)}{\prod_{p=l+1}^{k} c_{p+1,p}} \frac{f_l}{c_{l+1,l}} \right].$$

We have re-written the results slightly to reveal the influences of the errors on the recurrence. In the new point of view, in every step, a new method with starting vector is invoked. The overlay of all methods with starting vectors $q$ (wanted part) and

$f_l/c_{l+1,l}$, $l \in \underline{k}$ (unwanted part) is normalised with erroneous normalising constant $c_{k+1,k}$.

We have refrained to use the results obtained to derive *bounds* on the deviation in terms of bounds on the convergence. This, in contrast to the before mentioned, *has* to be based on the characteristics of the machine, the actual method and the algorithmic implementation. Concerning bounds, the results might differ greatly between *two* implementations of *one* method. This becomes obvious when considering the MGS and Householder variants of the GMRES (Arnoldi) method. The new results on the deviation can be used to devise new, more appropriate stopping criteria and to develop new semi-orthogonalisation methods.

# Bibliography


[Arn51] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9: 17–29, 1951. CODEN QAMAAY. ISSN 0033-569X.

[ARRY01] Götz Alefeld, Jiří Rohn, Siegfried Rump, and Tetsuro Yamamoto, editors. *Symbolic Algebraic Methods and Verification Methods*. Springer-Verlag Wien New York, 2001. ISBN 3-211-83593-8.

[Bai94] Zhaojun Bai. Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem. *Mathematics of Computation*, 62(205): 209–226, January 1994. CODEN MCMPAF. ISSN 0025-5718.

[BDM89] Z. Bai, J. Demmel, and A. McKenney. On the conditioning of the nonsymmetric eigenproblem: Theory and software. Technical Report CS-89-86, University of Tennessee, October 1989. Available as LAPACK Working Note 13, see [BDM91].

[BDM91] Z. Bai, J. Demmel, and A. McKenney. On the conditioning of the nonsymmetric eigenproblem: Theory and software. LAPACK Working Note 13, Courant Institute, 251 Mercer Str., New York, NY 10012, December 1991. URL `http://www.netlib.org/lapack/lawns/lawn13.ps; http://www.netlib.org/lapack/lawnspdf/lawn13.pdf`. Originally appeared as University of Tennessee Technical Report CS-89-86, see [BDM89].

[BDY99] Zhaojun Bai, David Day, and Qiang Ye. ABLE: an adaptive block Lanczos method for non-Hermitian eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 20(4):1060–1082, October 1999. CODEN SJMAEL. ISSN 0895-4798 (print), 1095-7162 (electronic). URL `http://epubs.siam.org/sam-bin/dbq/article/31780`.

[BF60] F. L. Bauer and C. T. Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2:137–141, 1960. CODEN NUMMA7. ISSN 0029-599X (print), 0945-3245 (electronic).

[Bot] Mikhail A. Botchev. A. N. Krylov: a short biography. URL `http://www.math.uu.nl/people/vorst/kryl.html`.

[BP92] Å. Björck and C. C. Paige. Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm. *SIAM Journal on Matrix Analysis and Applications*, 13(1):176–190, January 1992. CODEN SJMAEL. ISSN 0895-4798 (print), 1095-7162 (electronic).

[BS97] Z. Bai and G. W. Stewart. Algorithm 776: SRRIT: a Fortran subroutine to calculate the dominant invariant subspace of a nonsymmetric matrix. *ACM Transactions on Mathematical Software*, 23(4):494–513,

December 1997. CODEN ACMSCU. ISSN 0098-3500. URL http://doi.acm.org/10.1145/279232.279234.

[CC00] F. Chaitin-Chatelin. Comprende les mèthodes de Krylov en précision finie : le programme du Groupe Qualitative Computing au CERFACS. CERFACS Technical Report TR/PA/00/11, CERFACS, 42, Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France, 2000. URL http://aton.cerfacs.fr/algor/reports/2000/TR_PA_00_11.ps.gz.

[CCF96] Françoise Chaitin-Chatelin and Valérie Frayssé. *Lectures on Finite Precision Computations*, volume 1 of *Software, Environments, and Tools*. SIAM, Philadelphia, 1996. ISBN 0-89871-358-7. URL http://www.ec-securehost.com/SIAM/SE01.html.

[CCTP00] F. Chaitin-Chatelin, E. Traviesas, and L. Plantiè. Understanding Krylov methods in finite precision. CERFACS Technical Report TR/PA/00/40, CERFACS, 42, Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France, 2000. URL http://aton.cerfacs.fr/algor/reports/2000/TR_PA_00_40.ps.gz. Also in the Proceedings of the Second Conference on Numerical Analysis and Applications, Rousse, Bulgaria, pp. 187–197, Springer-Verlag.

[CG96] Jane Cullum and Anne Greenbaum. Relations between Galerkin and norm-minimizing iterative methods for solving linear systems. *SIAM Journal on Matrix Analysis and Applications*, 17(2):223–247, April 1996. CODEN SJMAEL. ISSN 0895-4798 (print), 1095-7162 (electronic). URL http://epubs.siam.org/sam-bin/dbq/article/24676.

[Cha93] Françoise Chatelin. *Eigenvalues of Matrices*. John Wiley & Sons Ltd., Chichester, New York, Brisbane, Toronto, Singapore, 1993. ISBN 0-471-93538-7. With exercises by Mario Ahués and Françoise Chatelin. Translated from the French with additional material by Walter Ledermann.

[Cod81] William J. Cody, Jr. Analysis of proposals for the floating-point standard. *Computer*, 14(3):63–69, March 1981. CODEN CPTRB4. ISSN 0018-9162. See [IEE85b, IEE85c].

[Coo80] J. T. Coonen. An implementation guide to a proposed standard for floating-point arithmetic. *Computer*, 13(1):68–79, January 1980. CODEN CPTRB4. ISSN 0018-9162. See errata in [Coo81a]. See [IEE85b, IEE85c].

[Coo81a] Jerome T. Coonen. Errata: An implementation guide to a proposed standard for floating point arithmetic. *Computer*, 14(3):62, March 1981. CODEN CPTRB4. ISSN 0018-9162. See [Coo80, IEE85b, IEE85c].

[Coo81b] Jerome T. Coonen. Underflow and the denormalized numbers. *Computer*, 14(3):75–87, March 1981. CODEN CPTRB4. ISSN 0018-9162. See [IEE85b, IEE85c].

[CW80] Jane Cullum and Ralph A. Willoughby. The Lanczos phenomenon – an interpretation based upon conjugate gradient optimization. *Linear Algebra and its Applications*, 29:63–90, 1980. CODEN LAAPAW. ISSN 0024-3795.

[CW85a] Jane K. Cullum and Ralph A. Willoughby. *Lanczos algorithms for large symmetric eigenvalue computations. Volume I: Theory*, volume 3 of *Progress in Scientific Computing*. Birkhäuser, Boston, Basel, Stuttgart, first edition, 1985. ISBN 3-7643-3058-9, 0-8176-3058-9.

[CW85b] Jane K. Cullum and Ralph A. Willoughby. *Lanczos algorithms for large symmetric eigenvalue computations. Volume II: Programs*, volume 4 of *Progress in Scientific Computing*. Birkhäuser, Boston, Basel, Stuttgart, first edition, 1985. ISBN 3-7643-3294-8, 0-8176-3294-8. URL `http://www.netlib.org/lanczos`.

[Day93] David Minot Day, III. *Semi-duality in the two-sided Lanczos algorithm*. PhD thesis, University of California, Berkeley, Berkeley, November 1993.

[Day97] David Day. An efficient implementation of the nonsymmetric Lanczos algorithm. *SIAM Journal on Matrix Analysis and Applications*, 18 (3):566–589, July 1997. CODEN SJMAEL. ISSN 0895-4798 (print), 1095-7162 (electronic). URL `http://epubs.siam.org/sam-bin/dbq/article/29250`.

[DEG72] Germund Dahlquist, Stanley C. Eisenstat, and Gene H. Golub. Bounds for the error of linear systems of equations using the theory of moments. *Journal of Mathematical Analysis and Applications*, 37:151–166, 1972. CODEN JMANAK. ISSN 0022-247X.

[Dem] James Demmel. Personal communication.

[Dem86] James Weldon Demmel. Computing stable eigendecompositions of matrices. *Linear Algebra and its Applications*, 79:163–193, 1986. CODEN LAAPAW. ISSN 0024-3795.

[Dem97] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997. ISBN 0-89871-389-7. URL `http://www.ec-securehost.com/SIAM/ot56.html`; `http://www.cs.berkeley.edu/~demmel/ma221/SIAM.html`.

[Dem00] James Demmel. The inherent complexity of accurate floating point computation. Talk at IWASEP3, III International Workshop on Accurate Solution of Eigenvalue Problems, Arcadeon Conference Center, Hagen, Germany, July 2000.

[DGG+] J.-G. Dumas, T. Gautier, M. Giesbrecht, P. Giorgi, B. Hovinen, E. Kaltofen, B.D. Saunders, W.J. Turner, and G. Villard. LinBox: A generic library for exact linear algebra. URL `citeseer.nj.nec.com/511043.html`. submitted to World Scientific.

[DGN79] Germund Dahlquist, Gene H. Golub, and Stephen G. Nash. Bounds for the error in linear systems. In Hettich [Het79], pages 154–172. ISBN 3-540-09479-2.

[DH93] Peter Deuflhard and Andreas Hohmann. *Numerische Mathematik I. Eine algorithmisch orientierte Einführung. (Numerical mathematics I. An algorithmically oriented introduction). 2., überarb. Aufl.* de Gruyter Lehrbuch. Berlin: Walter de Gruyter,. xv, 371 S., 1993.

[Dhi97] Inderjit Singh Dhillon. *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*. PhD thesis, Graduate Division of the University of California, Berkeley, 1997.

[Don] Jack Dongarra. Freely available software for linear algebra on the web. URL http://www.netlib.org/utk/people/JackDongarra/la-sw.html.

[DS93] I. S. Duff and J. A. Scott. Computing selected eigenvalues of sparse unsymmetric matrices using subspace iteration. *ACM Transactions on Mathematical Software*, 19(2):137–159, June 1993. CODEN ACMSCU. ISSN 0098-3500. URL http://doi.acm.org/10.1145/152613.152614. See [DS95].

[DS95] Iain S. Duff and Jennifer A. Scott. Corrigendum: Computing selected eigenvalues of sparse unsymmetric matrices using subspace iteration. *ACM Transactions on Mathematical Software*, 21(4):490, December 1995. CODEN ACMSCU. ISSN 0098-3500. URL http://doi.acm.org/10.1145/212066.215254. See [DS93].

[EE99] Michael Eiermann and Oliver G. Ernst. Geometric aspects in the theory of Krylov subspace methods. Internal publication ano394, Laboratoire ANO Université des Sciences et Technologies de Lille, 1999. URL http://ano.univ-lille1.fr/pub/1999/ano394.ps.Z.

[ER80] Thomas Ericsson and Axel Ruhe. The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems. *Mathematics of Computation*, 35(152):1251–1268, October 1980.

[Fac00] Axel Facius. *Iterative Solution of Linear Systems with Improved Arithmetic and Result Verification*. PhD thesis, Universität Karlsruhe (TH), 2000.

[FG93] Bernd Fischer and Gene Golub. On the error computation for polynomial based iteration methods. In Gene Golub, A. Greenbaum, and M. Luskin, editors, *Recent Advances in Iterative Methods*, volume 60 of *The IMA Volumes in Mathematics and its Applications*, pages 59–67. IMA, Springer, 1993.

[FM84] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM Journal on Numerical Analysis*, 21:352–362, 1984.

[FSvdV96] Diederik R. Fokkema, Gerard L. G. Sleijpen, and Henk A. van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils. Preprint 941, Department of Mathematics, Utrecht University, January 1996. URL http://www.math.ruu.nl/publications/preprints/941.ps.gz. Appeared as [FSvdV99].

[FSvdV99] Diederik R. Fokkema, Gerard L. G. Sleijpen, and Henk A. van der Vorst. Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM Journal on Scientific Computing*, 20(1):94–125, January 1999. CODEN SJOCE3. ISSN 1064-8275 (print), 1095-7197 (electronic). URL http://epubs.siam.org/sam-bin/dbq/article/30007.

[Fuc96] B. Fuchssteiner. *MuPAD User's Manual*. John Wiley & Sons Ltd., Chichester, New York, Wenheim, Brisbane, Singapore, Toronto, first edition, March 1996. ISBN Wiley: 0-471-96716-5, Teubner: 3-519-02114-5. Includes a CD for Apple Macintosh and UNIX.

[FW98] Andreas Frommer and André Weinberg. Verified error bounds for linear systems through the Lanczos process. Preprint BUGHW-SC 98/9, Bergische Universität GH Wuppertal Fachbereich Mathematik, December 1998. URL `http://www.math.uni-wuppertal/SciComp/`.

[GL02a] L. Giraud and J. Langou. Robust selective Gram-Schmidt reorthogonalization. Technical Report TR/PA/02/52, CERFACS, Toulouse, France, 2002. Submitted to SIAM J. Scientific Computing Copper Mountain Special Issue.

[GL02b] L. Giraud and J. Langou. When modified Gram-Schmidt generates a well-conditioned set of vectors. *IMA Journal of Numerical Analysis*, 22 (4):521–528, 2002.

[GLR02a] L. Giraud, J. Langou, and M. Rozložník. On the loss of orthogonality in the Gram-Schmidt orthogonalization process. Submitted for Proceedings of a conference in Miskolc, 2002.

[GLR02b] L. Giraud, J. Langou, and M. Rozložník. On the round-off error analysis of the Gram-Schmidt algorithm with reorthogonalization. Research Report TR/PA/02/33, CERFACS, Toulouse, France, April 2002.

[GM94a] Gene H. Golub and Gérard Meurant. Matrices, moments and quadrature. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1993*, volume 303 of *Pitman Research Notes in Mathematics*, pages 105–156. Longman Scientific & Technical, 1994. Also available as [GM94b].

[GM94b] Gene H. Golub and Gérard Meurant. Matrices, moments and quadrature. SCCM Technical Report SCCM-93-07, SCCM, Stanford, September 1994. URL `http://www-sccm.stanford.edu/pub/sccm/sccm93-07.ps.gz`. Appeared as [GM94a].

[GM96] Gene H. Golub and Gérard Meurant. Matrices, moments and quadrature II *or* How to compute the norm of the error in iterative methods. SCCM Technical Report SCCM-96-18, SCCM, Stanford, September 1996. URL `http://www-sccm.stanford.edu/pub/sccm/sccm96-18.ps.gz`. Appeared as [GM97].

[GM97] Gene H. Golub and Gérard Meurant. Matrices, moments and quadrature II: how to compute the norm of the error in iterative methods. *BIT (Nordisk tidskrift for informationsbehandling)*, 37(3):687–705, October 1997. Also available as [GM96].

[GNU] The GNU Compiler Collection. URL `http://www.gnu.org/software/gcc/gcc.html`.

[Grc81] Joseph Frank Grcar. *Analyses of the Lanczos Algorithm and of the Approximation Problem in Richardson's Method*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 1981.

[Gre81] Anne Greenbaum. *Convergence Properties of the Conjugate Gradient Algorithm in Exact and Finite Precision Arithmetic*. PhD thesis, University of California, Berkeley, 1981.

[Gre89] A. Greenbaum. Behaviour of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra and its Applications*, 113:7–63, 1989.

[Gre97] Anne Greenbaum. *Iterative methods for solving linear systems*. Frontiers in Applied Mathematics. SIAM, Philadelphia, 1997.

[GS92] A. Greenbaum and Z. Strakoš. Predicting the behaviour of finite precision Lanczos and conjugate gradient computations. *SIAM Journal on Matrix Analysis and Applications*, 13(1):121–137, January 1992.

[GS94a] Gene H. Golub and Zdeněk Strakoš. Estimates in quadratic formulas. *Numerical Algorithms*, 8:241–268, August 1994.

[GS94b] Gene H. Golub and Zdeněk Strakoš. Estimates in quadratic formulas. SCCM Technical Report SCCM-93-08, Department of Computer Science, Stanford University, August 1994. URL `http://www-sccm.stanford.edu/pub/sccm/sccm93-08.ps.gz`. Appeared as [GS94a].

[GS97] Martin H. Gutknecht and Zdeněk Strakoš. Accuracy of three-term and two-term recurrences for Krylov space solvers. Technical Report TR-97-21, Swiss Center for Scientific Computing ETH Zürich, December 1997.

[GS00] Martin H. Gutknecht and Zdeněk Strakoš. Accuracy of two three-term and three-two term recurrences for Krylov space solvers. *SIAM Journal on Matrix Analysis and Applications*, 22(1):213–229, 2000.

[GvL96] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 3rd edition, 1996.

[Het79] Rainer Hettich, editor. *Semi-Infinite Programming: Proceedings of a Workshop, Bad Honnef, August 30–September 1, 1978*, volume 15 of *Lecture notes in control and information sciences*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1979. ISBN 3-540-09479-2.

[Hig93] Nicholas J. Higham. Perturbation theory and backward error for $AX - XB = C$. *BIT*, 33:124–136, 1993. URL `http://www.maths.man.ac.uk/~nareports/narep211.ps.gz`.

[Hig96] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 1996.

[HJ85] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985. ISBN 0-521-30586-1 (hardback), 0-521-38632-2 (paperback).

[HJ94] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1994.

[HL98] Marlis Hochbruck and Christian Lubich. Error analysis of Krylov methods in a nutshell. *SIAM Journal on Scientific Computing*, 19(2):695–701, March 1998. Timely Communication.

[Hou75] Alston S. Householder. *The Theory of Matrices in Numerical Analysis*. Dover Books on Elementary and Intermediate Mathematics. Dover Publications, New York, 2nd edition, 1975.

[Hou81] David Hough. Applications of the proposed IEEE-754 standard for floating point arithmetic. *Computer*, 14(3):70–74, March 1981. CODEN CPTRB4. ISSN 0018-9162. See [IEE85b, IEE85c].

[HS52] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December 1952.

[IA85] IEEE Computer Society Standards Committee. Working group of the Microprocessor Standards Subcommittee and American National Standards Institute. *IEEE standard for binary floating-point arithmetic*. ANSI/IEEE Std 754-1985. IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1985. 18 pp. See [IEE85b].

[IEE85a] IEEE. IEEE standard for binary floating-point arithmetic. *ACM SIGPLAN Notices*, 22(2):9–25, February 1985. CODEN SINODQ. ISSN 0362-1340. See [IEE85b].

[IEE85b] IEEE Task P754. *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*. IEEE, New York, NY, USA, August 12, 1985. ISBN 1-55937-653-8. 20 pp. US$35.00. URL `http://standards.ieee.org/reading/ieee/std_public/description/busarch/754-1985_desc.html; http://www.iec.ch/cgi-bin/procgi.pl/www/iecwww.p?wwwlang=E&wwwprog=cat-det.p&wartnum=019113`. Revised 1990. A preliminary draft was published in the January 1980 issue of IEEE Computer, together with several companion articles [Cod81, Coo81b, Coo80, Coo81a, Hou81, Ste81a, Ste81b]. The final version was republished in [IEE85a, IA85]. See also [WF82]. Also standardized as *IEC 60559 (1989-01) Binary floating-point arithmetic for microprocessor systems*.

[IEE85c] IEEE Task P754. *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*. IEEE, New York, August 12 1985. A preliminary draft was published in the January 1980 issue of IEEE Computer, together with several companion articles [Cod81, Coo81b, Coo80, Coo81a, Hou81, Ste81a, Ste81b]. Available from the IEEE Service Center, Piscataway, NJ, USA.

[IEE87] IEEE. *854-1987 (R1994) IEEE Standard for Radix-Independent Floating-Point Arithmetic*. IEEE, New York, NY, USA, 1987. ISBN 1-55937-859-X. 16 pp. US$44.00. URL `http://standards.ieee.org/reading/ieee/std_public/description/busarch/854-1987_desc.html`. Revised 1994.

[Int] Interval Computations Homepage. URL `cs.utep.edu/interval-comp/main.html`.

[Ips96a] I. C. F. Ipsen. Helmut Wielandt's contributions to the numerical solution of complex eigenvalue problems. In B. Huppert and H. Schneider, editors, *Helmut Wielandt, Mathematische Werke, Mathematical Works*, volume 2: Linear Algebra and Analysis, pages 453–463. Walter de Gruyter, New York, 1996.

[Ips96b] I. C. F. Ipsen. A history of inverse iteration. In B. Huppert and H. Schneider, editors, *Helmut Wielandt, Mathematische Werke, Mathematical Works*, volume 2: Linear Algebra and Analysis, pages 464–472. Walter de Gruyter, New York, 1996.

[Ips97] Ilse C. F. Ipsen. Computing an eigenvector with inverse iteration. *SIAM Review*, 39(2):254–291, 1997. CODEN SIREAD. ISSN 0036-1445 (print), 1095-7200 (electronic).

[Jar] Dick Jardine. Difference equation chronology. URL `http://wcb.keene.edu/~djardine/web/IMHT/diffeqnchr.htm`.

[Kat66] Tosio Kato. *Perturbation Theory for Linear Operators*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1966. xix + 592 pp. LCCN QA320 .K33.

[KGL97] Philip A. Knight, Michael Grinfeld, and Harbir Lamba. Nonnormality and finite precision arithmetic in power method dynamics. Technical report, Numerical Analysis Group, Department of Mathematics, University of Strathclyde, February 1997.

[KM81] Ulrich W. Kulisch and Willard L. Miranker. *Computer Arithmetic in Theory and Practice.* Computer Science and Applied Mathematics. New York etc.: Academic Press, a Subsidiary of Harcourt Brace Jovanovich, Publishers. XV, 249 p. , 1981.

[Knü94] Olaf Knüppel. PROFIL/BIAS - a fast interval library. *Computing*, 53: 277–287, 1994.

[Knu98] Donald E. Knuth. *The art of computer programming. Vol. 2: Seminumerical algorithms. 3rd ed.* Bonn: Addison-Wesley. xiii, 762 p. , 1998.

[Koe00] Erik Koelink. Spectral theory and special functions. Lecture Notes for a four hour course in the SIAM Activity Group "Orthogonal Polynomials and Special Functions", Summer School, Laredo, Spain, July 2000. Last revision: May 7, 2002.

[KPJ82] W. Kahan, B. N. Parlett, and E. Jiang. Residual bounds on approximate eigensystems of nonnormal matrices. *SIAM Journal on Numerical Analysis*, 19(3):470–484, June 1982. CODEN SJNAAM. ISSN 0036-1429 (print), 1095-7170 (electronic).

[Кры31] Алексей Николаевич Крылов. О численном решении уравнения, которым в технических вопросах определяются частоты малых колебаний материальных систем. *Известия Академии Наук СССР. Отделение математических и естественных наук. Ser. VII*, 4:491–539, 1931.

[Lan50] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, October 1950.

[Lan52] Cornelius Lanczos. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49 (1):33–53, July 1952.

[Lan02] Julien Langou. A reorthogonalization procedure for MGS applied to a low rank deficient matrix. Talk at the Conference: Computational Linear Algebra with Applications, Milovy, Czech Republic, August 4–10 2002.

[Li85] Ren-Cang Li. On perturbation bounds for eigenvalues of a matrix. Computing Center, Academia Sinica, Beijing, P. R. China, November 1985. URL `http://www.cs.uky.edu/~rcli/papers/bafijordan.ps`. Preprint.

[LSY98] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK user's guide. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.* SIAM, 1998.

[LV74] M. La Porte and J. Vignes. Étude statistique des erreurs dans l'arithmétique des ordinateurs; application au contrôle des Résultats d'algorithmes numériques. (French) [Statistical study of errors in computer arithmetic: Application to the control of the results of numerical algorithms]. *Numerische Mathematik*, 23(1):63–72, 1974. CODEN NUMMA7. ISSN 0029-599X (print), 0945-3245 (electronic).

[MBO97] Julio Moro, James V. Burke, and Michael L. Overton. On the Lidskii–Vishik–Lyusternik perturbation theory for eigenvalues of matrices with arbitrary Jordan structure. *SIAM Journal on Matrix Analysis and Applications*, 18(4):793–817, 1997. URL `citeseer.nj.nec.com/325913.html`.

[Meu99] Gérard Meurant. *Computer Solution of Large Linear Systems*, volume 28 of *Studies in Mathematics and its Applications*. North Holland, Amsterdam – Lausanne – New York – Oxford – Shannon – Singapore – Tokyo, 1999. ISBN 0-444-50169-X.

[Moo66] R. E. Moore. *Interval analysis (Prentice-Hall Series in Automatic Computation)*. Englewood Cliffs, N. J.: Prentice-Hall, Inc., XI, 145 p. , 1966.

[Net] Software. URL `http://www.netlib.org`.

[oMa] The MacTutor History of Mathematics archive. Aleksei Nikolaevich Krylov. URL `http://www-groups.dcs.st-andrews.ac.uk/~history/Mathematicians/Krylov_Aleksei.html`.

[OP64] W. Oettli and W. Prager. Computability of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. *Numerische Mathematik*, 6:405–409, 1964.

[Pai69] C. C. Paige. Error analysis of the symmetric Lanczos processes for the eigenproblem. Tech. Note ICSI 209, London University Insitute of Computer Science, Insitute of Computer Science 44 Gordon Square London WCI, July 1969.

[Pai70] C. C. Paige. Practical use of the symmetric Lanczos process with reorthogonalization. *BIT (Nordisk tidskrift for informationsbehandling)*, 10:183–195, 1970.

[Pai71] Christopher Conway Paige. *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*. PhD thesis, London University Institute of Computer Science, April 1971.

[Pai72] C. C. Paige. Computational variants of the Lanczos method for the eigenproblem. *J. Inst. Maths Applics*, 10:373–381, 1972.

[Pai76] C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *J. Ins. Maths Applics*, 18:341–349, 1976.

[Pai80] C. C. Paige. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear Algebra and its Applications*, 34: 235–258, 1980.

[Par98] Beresford N. Parlett. *The symmetric eigenvalue problem*. Classics in Applied Mathematics. SIAM, Philadelphia, unabridged, corrected republication of 1980 edition, 1998.

[Rei71] John K. Reid. On the method of conjugate directions for the solution of large sparse systems of equations. In J.K. Reid, editor, *Large Sparse Sets of Linear Equations*, pages 231–254. Academic Press London New York, 1971.

[Rel] Reliable Computing Mailing List. URL `reliable_computing@interval.louisiana.edu`.

[RG67] J. L. Rigal and J. Gaches. On the computability of a given solution with the data of a linear system. *Journal of the Association for Computing Machinery*, 14(3):543–548, July 1967.

[Roz97] Miroslav Rozložník. *Numerical Stability of the GMRES Method*. PhD thesis, Institute of Computer Science, Czech Technical University Prague, Prague, April 1997.

[Rum98] S. M. Rump. INTLAB - INTerval LABoratory. Berichte des Forschungsschwerpunktes Informations- und Kommunikationstechnik Bericht 98.4, Technische Universität Hamburg-Harburg, Inst. f. Informatik III, Oktober 1998. URL `http://www.ti3.tu-harburg.de/rump/intlab/index.html`.

[Rum99] Siegfried M. Rump. Fast and parallel interval arithmetic. *BIT (Nordisk tidskrift for informationsbehandling)*, 39(3):534–554, January 1999.

[Saa80] Youcef Saad. On the rates of convergence of the Lanczos and the block-Lanczos methods. *SIAM Journal on Numerical Analysis*, 17(5):687–706, February 1980.

[Saa92] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Halsted Press, Div. of John Wiley & Sons, Inc., New York, 1992. online available.

[Saa96] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. PWS, Boston, 1996. ISBN 053494776X. online available.

[Sco78] David St. Clair Scott. *Analysis of the Symmetric Lanczos Process*. PhD thesis, University of California, Berkeley, Berkeley, December 1978.

[Sco79] D. S. Scott. How to make the Lanczos algorithm converge slowly. *Mathematics of Computation*, 33(145):239–247, January 1979.

[Sim82] Horst D. Simon. *The Lanczos Algorithm for Solving Linear Systems*. PhD thesis, Dept. of Methematics, University of California, Berkeley, April 1982.

[Sim84] Horst D. Simon. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Linear Algebra and its Applications*, 61: 101–131, 1984.

[Sin98] Christof Sinn. *Zur verifizierten Lösung großer linearer Gleichungssysteme mit spärlich besetzten Matrizen beliebiger Bandbreite*. PhD thesis, Universität Basel, 1998.

[SJ81] William J. Stewart and Alan Jennings. Algorithm 570: LOPSI: A simultaneous iteration method for real matrices [F2]. *ACM Transactions on Mathematical Software*, 7(2):230–232, June 1981. CODEN ACMSCU. ISSN 0098-3500.

[SS86] Youcef Saad and Martin H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Statistic Computing*, 7(3):856–869, July 1986.

[SS90] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, Boston, 1990.

[Ste73] G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, October 1973. CODEN SIREAD. ISSN 0036-1445 (print), 1095-7200 (electronic).

[Ste81a] David Stevenson. A proposed standard for binary floating-point arithmetic. *Computer*, 14(3):51–62, March 1981. CODEN CPTRB4. ISSN 0018-9162. See [IEE85b, IEE85c].

[Ste81b] David Stevenson. *A proposed standard for binary floating-point arithmetic: draft 8.0 of IEEE Task P754*. IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1981. 36 pp. See [IEE85b, IEE85c].

[Ste98] G. W. Stewart. On the adjugate matrix. *Linear Algebra and its Applications*, 283:151–164, 1998.

[Ste01] G. W. Stewart. Backward error bounds for approximate Krylov subspaces. Technical Report UMIACS TR-2001-32, CMSC TR-4247, University of Maryland, 2001. URL `ftp://thales.cs.umd.edu/pub/reports/bebaks.ps`. Appeared as [Ste02].

[Ste02] G. W. Stewart. Backward error bounds for approximate Krylov subspaces. *Linear Algebra and its Applications*, 340:115–120, 2002.

[SvdV95] G. L. G. Sleijpen and H. A. van der Vorst. The Jacobi-Davidson method for eigenvalue problems and ist relation with accelerated inexact newton schemes. Technical report, Mathematical Institute, Utrecht University, Budapestlaan 6, Utrecht, the Netherlands, February 1995. URL `http://www.math.uu.nl/people/sleijpen/JDaNEWTON.ps.gz`.

[SvdVM97] Gerard L.G. Sleijpen, Henk A. van der Vorst, and Jan Modersitzki. The main effects of rounding errors in Krylov solvers for symmetric linear systems. Preprint 1006, Universiteit Utrecht, Department of Mathematics, 1997. URL `http://www.math.uu.nl/publications/preprints/1006.ps.gz`.

[SvdVM01] Gerard L. G. Sleijpen, Henk A. van der Vorst, and Jan Modersitzki. Differences in the effects of rounding errors in Krylov solvers for symmetric indefinite linear systems. *SIAM Journal on Matrix Analysis and Applications*, 22(3):726–751, July 2001. CODEN SJMAEL. ISSN 0895-4798 (print), 1095-7162 (electronic). URL `http://epubs.siam.org/sam-bin/dbq/article/32308`.

[SW95] Willi Schönauer and Rüdiger Weiss. An engineering approach to generalized conjugate gradient methods and beyond. Applied numerical mathematics 1995, Rechenzentrum der Universität Karlsruhe, 1995. URL http://www.uni-karlsruhe.de/~ne03/litps/ANM-95.1.ps.gz.

[TB97] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997. ISBN 0-89871-361-7. URL http://www.siam.org/books/ot50/; http://www.ec-securehost.com/SIAM/ot50.html.

[Tho66] R. C. Thompson. Principal submatrices of normal and Hermitian matrices. *Illinois J. Math.*, 10:296–308, January 1966.

[TM68] R. C. Thompson and P. McEnteggert. Principal submatrices II: The upper and lower quadratic inequalities. *Linear Algebra and its Applications*, 1:211–243, 1968.

[Tre93] Lloyd N. Trefethen. Non-normal operators and pseudospectra. Handout to a talk given at the University of Hamburg, 1993.

[Tre97] Lloyd N. Trefethen. Pseudospectra of linear operators. *SIAM Review*, 39(3):383–406, September 1997. CODEN SIREAD. ISSN 0036-1445 (print), 1095-7200 (electronic). URL http://epubs.siam.org/sam-bin/dbq/article/29528.

[Tre99] Lloyd Nicholas Trefethen. Computation of pseudospectra. *Acta Numerica*, 8:247–295, 1999. URL http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/acta.ps.gz.

[TT94] Kim-Chuan Toh and Lloyd N. Trefethen. Calculation of pseudospectra by the Arnoldi iteration. Technical report CTC94TR179, Cornell Theory Center, Advanced Computing Research Institute, May 1994.

[TT96] Kim-Chuan Toh and Lloyd N. Trefethen. Calculation of pseudospectra by the Arnoldi iteration. *SIAM Journal on Scientific Computing*, 17 (1):1–15, January 1996. CODEN SJOCE3. ISSN 1064-8275 (print), 1095-7197 (electronic). Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994).

[TTA00] Top Ten Algorithms. NA Digest Sunday, July 23, 2000 Volume 00 : Issue 30, July 2000. URL http://netlib3.cs.utk.edu/cgi-bin/mfs/02/00/v00n30.html#1.

[TY95] Charles H. Tong and Qiang Ye. Analysis of the finite precision biconjugate gradient algorithm for nonsymmetric linear systems. Stanford SCCM Technical Report 95–11, Stanford University, Stanford, CA, October 1995.

[TY00] Charles H. Tong and Qiang Ye. Analysis of the finite precision biconjugate gradient algorithm for nonsymmetric linear systems. *Mathematics of Computation*, 69(232):1559–1575, 2000.

[Und75] Richard Underwood. *An iterative block Lanczos method for the solution of large sparse symmetric eigenproblems*. PhD thesis, Computer Science Department, School of Humanities and Sciences, Stanford University, 1975.

[Vin76] P. K. W. Vinsome. ORTHOMIN, an iterative method for solving sparse sets of simultaneous linear equations. 4th symposium of numerical simulation of reservoir performance. Technical report, Society of Petroleum Engineers of the AIME, Los Angeles, 1976.

[vNG47] John von Neumann and Herman H. Goldstine. Numerical inverting of matrices of high order. *Bull. Am. Math. Soc.*, 53:1021–1099, 1947.

[Voß93] Heinrich Voß. Iterative methods for linear systems of equations. Hamburger Beiträge zur Angewandten Mathematik, Reihe B, Bericht 27, Universität Hamburg, September 1993.

[Wei94] Rüdiger Weiss. Relations between smoothing and QMR. Interner Bericht Nr. 53/94, Universität Karlsruhe, Rechenzentrum, Abteilung "Numerikforschung für Supercomputer", Januar 1994. URL `http://www.uni-karlsruhe.de/~ne03/litps/ib53-94.ps.gz`.

[WF82] Shlomo Waser and Michael J. Flynn. *Introduction to Arithmetic for Digital Systems Designers*. Holt, Reinhart, and Winston, New York, NY, USA, 1982. ISBN 0-03-060571-7. xvii + 308 pp. LCCN TK7895 A65 W37 1982. Master copy output on Alphatype CRS high-resolution phototypesetter. This book went to press while the IEEE 754 Floating-Point Standard was still in development; consequently, some of the material on that system was invalidated by the final Standard (1985) [IEE85b].

[Wil63] J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. Dover Books on Advanced Mathematics. Dover Publications, Inc., 1994 reprinted edition, 1963.

[Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Numerical Mathematics and Scientific Computation. Oxford Science Publications, 1996 reprinted edition, 1965. ISBN 0-19-853418-3.

[XSC] XSC languages. URL `http://www.xsc.de`.

[YJ80] D. M. Young and K. C. Jea. Generalized conjugate–gradient acceleration of nonsymmetrizable iterative methods. *Linear Algebra and its Applications*, 34:159 – 194, 1980.

[Zem97] Jens-Peter M. Zemke. Symmetrien bei der numerischen Behandlung von linearen und nichtlinearen Gleichungssystemen. Master's thesis, Universität Hamburg, 1997.

[Zem99] Jens Zemke. b4m - A free interval arithmetic toolbox for Matlab based on BIAS. Berichte des Forschungschwerpunktes Informations- und Kommunikationstechnik Bericht 99.2, Technische Universität Hamburg-Harburg, Inst. f. Informatik III, März 1999. URL `http://www.ti3.tu-harburg.de/zemke/b4m/index.html`.

[Zem01] Jens-Peter M. Zemke. How orthogonality is lost in Krylov methods. In Alefeld et al. [ARRY01], pages 255–266. ISBN 3-211-83593-8.