A Calculus Approach to Matrix Eigenvalue Algorithms

Habilitationsschrift

der Fakultät für Mathematik und Informatik

der Bayerischen Julius-Maximilians-Universität Würzburg

für das Fach Mathematik vorgelegt von

Knut Hüper

Würzburg im Juli 2002

Meiner Frau Barbara

und unseren Kindern Lea, Juval und Noa gewidmet

Contents

Chapter 1

Introduction

The interaction between numerical linear algebra and control theory has crucially influenced the development of numerical algorithms for linear systems in the past. Since the performance of a control system can often be measured in terms of eigenvalues or singular values, matrix eigenvalue methods have become an important tool for the implementation of control algorithms. Standard numerical methods for eigenvalue or singular value computations are based on the QR-algorithm. However, there are a number of computational problems in control and signal processing that are not amenable to standard numerical theory or cannot be easily solved using current numerical software packages. Various examples can be found in the digital filter design area. For instance, the task of finding sensitivity optimal realizations for finite word length implementations requires the solution of highly nonlinear optimization problems for which no standard numerical solution algorithms exist.

There is thus the need for a new approach to the design of numerical algorithms that is flexible enough to be applicable to a wide range of computational problems as well as has the potential of leading to efficient and reliable solution methods. In fact, various tasks in linear algebra and system theory can be treated in a unified way as optimization problems of smooth functions on Lie groups and homogeneous spaces. In this way the powerful tools of differential geometry and Lie group theory become available to study such problems.

Higher order local convergence properties of iterative matrix algorithms are in many instances proven by means of tricky estimates. E.g., the Jacobi method, essentially, is an optimization procedure. The idea behind the proof of local quadratic convergence for the cyclic Jacobi method applied to a Hermitian matrix lies in the fact that one can estimate the amount of descent per sweep, see Henrici (1958) [Hen58]. Later on, by several authors these ideas where transferred to similar problems and even refined, e.g., Jacobi for the symmetric eigenvalue problem, Kogbetliantz (Jacobi) for SVD, skew-symmetric Jacobi, etc..

The situation seems to be similar for QR-type algorithms. Looking first at Rayleigh quotient iteration, neither Ostrowski (1958/59) [Ost59] nor Parlett [Par74] use Calculus to prove local cubic convergence.

About ten years ago there appeared a series of papers where the authors studied the global convergence properties of QR and RQI by means of dynamical systems methods, see Batterson and Smillie [BS89a, BS89b, BS90], Batterson [Bat95], and Shub and Vasquez [SV87]. To our knowledge these papers where the only ones where Global Analysis was applied to QR-type algorithms.

From our point of view there is a lack in studying the local convergence properties of matrix algorithms in a systematic way. The methodologies for different algorithms are often also different. Moreover, the possibility of considering a matrix algorithm atleast locally as a discrete dynamical system on a homogenous space is often overseen. In this thesis we will take this point of view. We are able to (re)prove higher order convergence for several wellknown algorithms and present some efficient new ones.

This thesis contains three parts.

At first we present a Calculus approach to the local convergence analysis of the Jacobi algorithm. Considering these algorithms as selfmaps on a manifold (i.e., projective space, isospectral or flag manifold, etc.) it turns out, that under the usual assumptions on the spectrum, they are differentiable maps around certain fixed points. For a wide class of Jacobi-type algorithms this is true due to an application of the Implicit Function Theorem, see [HH97, HH00, Hüp96, HH95, HHM96]. We then generalize the Jacobi approach to socalled Block Jacobi methods. Essentially, these methods are the manifold version of the socalled grouped variable approach to coordinate descent, wellknown to the optimization community.

In the second chapter we study the nonsymmetric eigenvalue problem introducing a new algorithm for which we can prove quadratic convergence. These methods are based on the idea to solve lowdimensional Sylvester equations again and again for improving estimates of invariant subspaces.

At third, we will present a new shifted QR-type algorithm, which is somehow the *true* generalization of the Rayleigh Quotien Iteration (RQI) to a full symmetric matrix, in the sense, that not only one column (row) of the matrix converges cubically in norm, but the off-diagonal part as a whole. Rather than being a scalar, our shift is matrix valued. A prerequisite for studying this algorithm, called Parallel RQI, is a detailed local analysis of the classical RQI itself. In addition, at the end of that chapter we discuss the local convergence properties of the shifted QR-algorithm. Our main result for this topic is that there cannot exist a smooth shift strategy ensuring quadratic convergence.

In this thesis we do not answer questions on global convergence. The algorithms which are presented here are all locally smooth self mappings of manifolds with vanishing first derivative at a fixed point. A standard argument using the mean value theorem then ensures that there exists an open neighborhood of that fixed point which is invariant under the iteration of the algorithm. Applying then the contraction theorem on the closed neighborhood ensures convergence to that fixed point and moreover that the fixed point is isolated. Most of the algorithms turn out to be discontinous far away from their fixed points but we will not go into this.

I wish to thank my colleagues in Würzburg, Gunther Dirr, Martin Kleinsteuber, Jochen Trumpf, and Piere-Antoine Absil for many fruitful discussions we had. I am grateful to Paul Van Dooren, for his support and the discussions we had during my visits to Louvain. Particularly, I am grateful to Uwe Helmke. Our collaboration on many different areas of applied mathematics is still broadening.

Chapter 2

Jacobi-type Algorithms and Cyclic Coordinate Descent

In this chapter we will discuss generalizations of the Jacobi algorithm well known from numerical linear algebra text books for the diagonalization of real symmetric matrices. We will relate this algorithm to socalled cyclic coordinate descent methods known to the optimization community. Under reasonable assumptions on the objective function to be minimized and on the step size selection rule to be considered, we will prove local quadratic convergence.

2.1 Algorithms

Suppose in an optimization problem we want to compute a local minimum of a smooth function

$$f : M \to \mathbb{R}, \tag{2.1}$$

defined on a smooth n-dimensional manifold M. Let denote for each $x \in M$

$$\{\gamma_1^{(x)}, \ldots, \gamma_n^{(x)}\} \tag{2.2}$$

a family of mappings,

$$\begin{aligned} \gamma_i^{(x)} &: \mathbb{R} \to M, \\ \gamma_i^{(x)}(0) &= x, \end{aligned} \tag{2.3}$$

such that the set $\{\dot{\gamma}_1^{(x)}(0),\ldots,\dot{\gamma}_n^{(x)}(0)\}$ forms a basis of the tangent space T_xM. We refer to the smooth mappings

$$\begin{aligned} G_i &: \mathbb{R}\times M \to M, \\ G_i(t,x) &:= \gamma_i^{(x)}(t) \end{aligned} \tag{2.4}$$

as the *basic transformations.*

2.1.1 Jacobi and Cyclic Coordinate Descent

The proposed algorithm for minimizing a smooth function $f : M \to \mathbb{R}$ then consists of a recursive application of socalled sweep operations. The algorithm is termed a *Jacobi-type* algorithm.

Algorithm 2.1 (Jacobi Sweep).

Given an $x_k \in M$ define

$$\begin{aligned} x_k^{(1)} &:= G_1(t_*^{(1)}, x_k) \\ x_k^{(2)} &:= G_2(t_*^{(2)}, x_k^{(1)}) \\ &\vdots \\ x_k^{(n)} &:= G_n(t_*^{(n)}, x_k^{(n-1)}) \end{aligned}$$

where for $i = 1,\ldots,n$

$$t_*^{(i)} := \arg\min_{t\in\mathbb{R}}(f(G_i(t,x_k^{(i-1)}))) \quad \text{if} \quad f(G_i(t,x_k^{(i-1)})) \not\equiv f(x_k^{(i-1)})$$

and

$$t_*^{(i)} := 0 \quad \text{otherwise.}$$

Thus $x_k^{(i)}$is recursively defined as the minimum of the smooth cost function $f : M \to \mathbb{R}$ when restricted to the i-th curve

$$\{G_i(t, x_k^{(i-1)}) \,|\, t \in \mathbb{R}\} \subset M.$$

The algorithm then consists of the iteration of sweeps.

Algorithm 2.2 (Jacobi-type Algorithm on n-dimensional Manifold).

- Let $x_0, \ldots, x_k \in M$ be given for $k \in \mathbb{N}_0$.
- Define the recursive sequence $x_k^{(1)}, \ldots, x_k^{(n)}$ as above (sweep).
- Set $x_{k+1} := x_k^{(n)}$. Proceed with the next sweep.

2.1.2 Block Jacobi and Grouped Variable Cyclic Coordinate Descent

A quite natural generalization of the Jacobi method is the following. Instead of minimizing along predetermined curves, one might minimize over the manifold using more than just one parameter at each algorithmic step.

Let denote

$$T_xM = V_1^{(x)} \oplus \cdots \oplus V_m^{(x)} \tag{2.5}$$

a direct sum decomposition of the tangent space T_xM at $x \in M$. We will not require the subspaces $V_i^{(x)}$, $\dim V_i^{(x)} = l_i$, to have equal dimension. Let denote

$$\{\gamma_1^{(x)}, \ldots, \gamma_m^{(x)}\} \tag{2.6}$$

a family of smooth mappings smoothly parameterized by x,

$$\begin{aligned} \gamma_i^{(x)} &: \mathbb{R}^{l_i} \to M, \\ \gamma_i^{(x)}(0) &= x, \end{aligned} \tag{2.7}$$

such that for all $i = 1, \dots, m$, for the image of the derivative

$$\operatorname{im} \mathrm{D}\, \gamma_i^{(x)}(0) = V_i^{(x)} \tag{2.8}$$

holds. Again we refer to

$$\begin{aligned} G_i : \mathbb{R}^{l_i} \times M &\to M, \\ G_i(t, x) &:= \gamma_i^{(x)}(t) \end{aligned} \tag{2.9}$$

as the *basic transformations*. Analogously, to the one-dimensional case above, the proposed algorithm for minimizing a smooth function $f : M \to \mathbb{R}$ then consists of a recursive application of socalled *grouped variable* sweep operations. The algorithm is termed a *Block Jacobi Algorithm*.

Algorithm 2.3 (Block Jacobi Sweep).

Given an $x_k \in M$. Define

$$\begin{aligned} x_k^{(1)} &:= G_1(t_*^{(1)}, x_k) \\ x_k^{(2)} &:= G_2(t_*^{(2)}, x_k^{(1)}) \\ &\vdots \\ x_k^{(m)} &:= G_m(t_*^{(m)}, x_k^{(m-1)}) \end{aligned}$$

where for $i = 1, \dots, m$

$$t_*^{(i)} := \arg \min_{t \in \mathbb{R}^{l_i}} (f(G_i(t, x_k^{(i-1)}))) \quad \text{if} \quad f(G_i(t, x_k^{(i-1)})) \not\equiv f(x_k^{(i-1)})$$

and

$$t_*^{(i)} := 0 \quad \text{otherwise.}$$

Thus $x_k^{(i)}$is recursively defined as the minimum of the smooth cost function $f : M \to \mathbb{R}$ when restricted to the i-th l_i-dimensional subset

$$\{G_i(t, x_k^{(i-1)}) \,|\, t \in \mathbb{R}^{l_i}\} \subset M.$$

The algorithm then consists of the iteration of sweeps.

Algorithm 2.4 (Block Jacobi Algorithm on Manifold).

- Let $x_0, \ldots, x_k \in M$ be given for $k \in \mathbb{N}_0$.
- Define the recursive sequence $x_k^{(1)}, \ldots, x_k^{(m)}$ as above (sweep).
- Set $x_{k+1} := x_k^{(m)}$. Proceed with the next sweep.

The formulation of the above algorithms suffer from several things. Without further assumptions on the objective function as well as on the mappings which lead to the basic transformations one hardly can prove anything.

For the applications we have in mind the objective function is always smooth. The art to choose suitable mappings $\gamma_i^{(x)}$ leading to the basic transformations often needs some insight into and intuition for the problem under consideration. For instance, if the manifold M is noncompact and the objective function $f : M \to \mathbb{R}^+$ is smooth and proper a good choice for the mappings $\gamma_i^{(x)}$ is clearly that one which ensures that the restriction $f|_{\gamma_i^{(x)}(\mathbb{R})}$ is also proper for all i and all $x \in M$. Moreover, if $M = \mathcal{G}$ is a compact Lie group, say $\mathcal{G} = \mathcal{SO}_n$, a good choice for $\gamma_i^{(x)} : \mathbb{R} \to \mathcal{SO}_n$ is one which ensures $\gamma_i^{(x)}([0, 2\pi]) \cong S^1 \cong \mathcal{SO}_2$. More generally, one often succeeds in finding mappings $\gamma_i^{(x)}$ such that optimizing the restriction of f to the image of these mappings is a problem of the same kind as the original one but of lower dimension being solvable in closed form. All these situations actually appear very often in practise. Some of them are briefly reviewed in the next subsection.

2.1.3 Applications and Examples for 1-dimensional Optimization

If $M = \mathbb{R}^n$ and $G_i(t, x) = x + te_i$, with e_i the i-th standard basis vector of $\mathbb{R}^n$, one gets the familiar coordinate descent method, cf. [AO82, BSS93, Lue84, LT92].

Various tasks in linear algebra and system theory can be treated in a unified way as optimization problems of smooth functions on Lie groups and homogeneous spaces. In this way the powerful tools of differential geometry and Lie group theory become available to study such problems. With Brockett's paper [Bro88] as the starting point there has been ongoing success in tackling difficult computational problems by geometric optimization methods. We refer to [HM94] and the PhD theses [Smi93, Mah94, Deh95, Hüp96] for more systematic and comprehensive state of the art descriptions. Some of the further application areas where our methods are potentially useful include diverse topics such as frequency estimation, principal component analysis, perspective motion problems in computer vision, pose estimation, system approximation, model reduction, computation of canonical forms and feedback controllers, balanced realizations, Riccati equations, and structured eigenvalue problems.

In the survey paper [HH97] a generalization of the classical Jacobi method for symmetric matrix diagonalization, see Jacobi [Jac46], is considered that is applicable to a wide range of computational problems. Jacobi-type methods have gained increasing interest, due to superior accuracy properties, [DV92], and inherent parallelism, [BL85, Göt94, Sam71], as compared to QR-based methods. The classical Jacobi method successively decreases the sum of squares of the off-diagonal elements of a given symmetric matrix to compute the eigenvalues. Similar extensions exist to compute eigenvalues or singular values of arbitrary matrices. Instead of using a special cost function such as the off-diagonal norm in Jacobi's method, other classes of cost functions are feasible as well. In [HH97] a class of perfect Morse-Bott functions on homogeneous spaces is considered that are defined by unitarily invariant norm functions or by linear trace functions. In addition to gaining further generality this choice of functions leads to an elegant theory as well as yielding improved convergence properties for the resulting algorithms.

Rather than trying to develop the Jacobi method in full generality on arbitrary homogeneous spaces in [HH97] its applicability by means of examples from linear algebra and system theory is demonstrated. New classes of Jacobi-type methods for symmetric matrix diagonalization, balanced realization, and sensitivity optimization are obtained. In comparison with standard numerical methods for matrix diagonalization the new Jacobi-method has the advantage of achieving automatic sorting of the eigenvalues. This sorting property is particularly important towards applications in signal processing; i.e., frequency estimation, estimation of dominant subspaces, independant component analysis, etc.

Let $\mathcal{G}$ be a real reductive Lie group and $\mathcal{K} \subset \mathcal{G}$ a maximal compact subgroup. Let

$$\alpha : \mathcal{G} \times V \to V, \quad (g, x) \mapsto g \cdot x \tag{2.10}$$

be a linear algebraic action of $\mathcal{G}$ on a finite dimensional vector space V. Each orbit $\mathcal{G}{\cdot}x$ of such a real algebraic group action then is a smooth submanifold of V that is diffeomorphic to the homogeneous space $\mathcal{G}/\mathcal{H}$, with $\mathcal{H} := \{g \in \mathcal{G} | g \cdot x = x\}$ the stabilizer subgroup. In [HH97] we are interested in understanding the structure of critical points of a smooth proper function $f : \mathcal{G} \cdot x \to \mathbb{R}^+$ defined on orbits $\mathcal{G} \cdot x$. Some of the interesting cases actually arise when f is defined by a norm function on V. Thus given a positive definite inner product $\langle \, , \, \rangle$ on V let $\|x\|^2 = \langle x, x \rangle$ denote the associated Hermitian norm. An Hermitian norm on V is called $\mathcal{K}-$invariant if

$$\langle k \cdot x, k \cdot y \rangle = \langle x, y \rangle \tag{2.11}$$

holds for all $x, y \in V$ and all $k \in \mathcal{K}$, for $\mathcal{K}$ a maximal compact subgroup of $\mathcal{G}$. Fix any such $\mathcal{K}-$invariant Hermitian norm on V. For any $x \in V$ we consider the smooth distance function on $\mathcal{G} \cdot x$ defined as

$$\phi \colon \mathcal{G}{\cdot}x \to \mathbb{R}^+, \quad \phi(g{\cdot}x) = \|g{\cdot}x\|^2. \tag{2.12}$$

We then have the following result due to Kempf and Ness [KN79]. For an important generalization to plurisubharmonic functions on complex homogeneous spaces, see Azad and Loeb [AL90].

Theorem 2.1. *1. The norm function $\phi \colon \mathcal{G}{\cdot}x \to \mathbb{R}^+$, $\phi(g{\cdot}x) = \|g{\cdot}x\|^2$, has a critical point if and only if the orbit $\mathcal{G} \cdot x$ is a closed subset of V.*

2. Let $\mathcal{G} \cdot x$ be closed. Every critical point of $\phi : \mathcal{G} \cdot x \to \mathbb{R}^+$ is a global minimum and the set of global minima is a single uniquely determined $\mathcal{K}-$orbit.

3. If $\mathcal{G} \cdot x$ is closed, then $\phi : \mathcal{G} \cdot x \to \mathbb{R}^+$ is a perfect Morse-Bott function. The set of global minima is connected. □

Theorem 2.1 completely characterizes the critical points of $\mathcal{K}-$invariant Hermitian norm functions on $\mathcal{G}-$orbits $\mathcal{G}x$ of a reductive Lie group $\mathcal{G}$. Similar results are available for compact groups. We describe such a result in a special situation which suffices for the subsequent examples. Thus let $\mathcal{G}$ now be a compact semisimple Lie group with Lie algebra $\mathfrak{g}$. Let

$$\alpha\colon \mathcal{G} \times \mathfrak{g} \to \mathfrak{g}, \quad (g, x) \mapsto g\cdot x = \mathrm{Ad}(g)x \tag{2.13}$$

denote the adjoint action of $\mathcal{G}$ on its Lie algebra. Let $\mathcal{G}\cdot x$ denote an orbit of the adjoint action and let

$$(x, y) := -\operatorname{tr}(\mathrm{ad}_x \circ \mathrm{ad}_y) \tag{2.14}$$

denote the Killing form on $\mathfrak{g}$. Then for any element $a \in \mathfrak{g}$ the trace function

$$f_a\colon \mathcal{G}\cdot x \to \mathbb{R}^+,\ f_a(g\cdot x) = -\operatorname{tr}(\mathrm{ad}_a \circ \mathrm{ad}_{g\cdot x}) \tag{2.15}$$

defines a smooth function on $\mathcal{G}\cdot x$. For a proof of the following result, formulated for orbits of the co-adjoint action, we refer to Atiyah [Ati82], Guillemin and Sternberg [GS82].

Theorem 2.2. *Let $\mathcal{G}$ be a compact, connected, and semisimple Lie group over $\mathbb{C}$ and let $f_a : \mathcal{G}\cdot x \to \mathbb{R}^+$ be the restriction of a linear function on a co-adjoint orbit, defined via evaluation with an element a of the Lie algebra. Then*

1. *$f_a\colon \mathcal{G}\cdot x \to \mathbb{R}$ is a perfect Morse-Bott function.*
2. *If $f_a\colon \mathcal{G}\cdot x \to \mathbb{R}$ has only finitely many critical points, then there exists a unique local=global minimum. All other critical points are saddle points or maxima.* □

Suppose now in an optimization exercise we want to compute the set of critical points of a smooth function $\phi : \mathcal{G}\cdot x \to \mathbb{R}^+$, defined on an orbit of a Lie group action. Thus let $\mathcal{G}$ denote a compact Lie group acting smoothly on a finite dimensional vector space V. For $x \in V$ let $\mathcal{G}\cdot x$ denote an orbit. Let $\{\Omega_1, \ldots, \Omega_N\}$ denote a basis of the Lie algebra $\mathfrak{g}$ of $\mathcal{G}$, with $N = \dim \mathcal{G}$. Denote by $\exp(t\Omega_i), t \in \mathbb{R}$, the associated one parameter subgroups of $\mathcal{G}$. We then refer to $G_1(t)$,...,$G_N(t)$ with $G_i(t, x) = \exp(t\Omega_i)\cdot x$ as the *basic transformations* of G as above.

Into the latter frame work also the Jacobi algorithm for the real symmetric eigenvalue problem from text books on matrix algorithms fits, cf.

[GvL89, SHS72]. If the real symmetric matrix to be diagonalized has distinct eigenvalues then the isospectral manifold of this matrix is diffeomorphic to the orthogonal group itself. Some advantages of the Jacobi-type method as compared to other optimization procedures one might see from the following example. The symmetric eigenvalue problem might be considered as a constrained optimization task in a Euclidian vector space embedding the orthogonal group, cf. [Chu88, Chu91, Chu96, CD90], implying relatively complicated lifting and projection computations in each algorithmic step. Intrinsic gradient and Newton-type methods for the symmetric eigenvalue problem were first and independently published in the Ph.D. theses [Smi93, Mah94]. The Jacobi approach, in contrast to the above- mentioned ones, uses *predetermined* directions to compute geodesics instead of directions determined by the gradient of the function or by calculations of second derivatives. One should emphasize the simple calculability of such directions: the optimization is performed only along closed curves. The bottleneck of the gradient-based or Newton-type methods with their seemingly good convergence properties is generally caused by the explicit calculation of directions, the related geodesics, and possibly step size selections. The time required for these computations may amount to the same order of magnitude as the whole of the problem. For instance, the computation of the exponential of a dense skew-symmetric matrix is comparable to the effort of determining its eigenvalues. The advantage of optimizing along circles will become evident by the fact that the complete analysis of the restriction of the function to that closed curve is a problem of considerably smaller dimension and sometimes can be solved in closed form. For instance, for the real symmetric eigenvalue problem one has to solve only a quadratic.

A whole class of further examples are developed in [Kle00] generalizing earlier results from [Hüp96]. There, generalizations of the conventional Jacobi algorithm to the problem of computing diagonalizations in compact Lie algebras are presented.

We would like two mention two additional applications, namely, (i) the computation of signature symmetric balancing transformations, being an important problem in systems and circuit theory, and (ii), the stereo matching problem without correspondence, having important applications in computer vision. The results referred to here are developed more detailed in [HHM02], respectively [HH98].

Signature Symmetric Balancing

From control theory it is well konwn that balanced realizations of symmetric transfer functions are signature symmetric. Wellknown algorithms, e.g., [LHPW87, SC89], however, do not preserve the signature symmetry and they may be sensible to numerical perturbations from the signature symmetric class. In recent years there is a tremendous interest in structure preserving (matrix) algorithms. The main motivation for this is twofold. If such a method can be constructed it usually (i) leads to reduction in complexity and (ii) often coincidently avoids that in finite arithmetic physically meaningless results are obtained. Translated to our case that means that (i) as the appropriate state space transformation group the Lie group $\mathcal{O}_{pq}^{+}$ of special pseudo-orthogonal transformations is used instead of $\mathcal{GL}_n$. Furthermore, (ii) at any stage of an algorithm the computed transformation should correspond to a signature symmetric realization if one would have started with one. Put into other words, the result of each iteration step should have some physical meaning. Let us very briefly review notions and results on balancing and signature symmetric realizations. Given any asymptotically stable linear system (A, B, C), the continuous-time controllability Gramian W_c and the observability Gramian W_o are defined, respectively, by

$$\begin{aligned} W_c &= \int\limits_0^\infty \mathrm{e}^{tA}\, BB'\, \mathrm{e}^{tA'}\, \mathrm{d}\, t, \\ W_o &= \int\limits_0^\infty \mathrm{e}^{tA'}\, C'C\, \mathrm{e}^{tA}\, \mathrm{d}\, t. \end{aligned} \tag{2.16}$$

Thus, assuming controllability and observability, the Gramians W_c, W_o are symmetric positive definite matrices. Moreover, a linear change of variables in the state space by an invertible state space coordinate transformation T leads to the co- and contravariant transformation law of Gramians as

$$(W_c, W_o) \mapsto \left(TW_cT', (T')^{-1}W_oT^{-1}\right). \tag{2.17}$$

Let $p, q \in \mathbb{N}_0$ be integers with $p + q = n$, $I_{pq} := \operatorname{diag}(I_p, -I_q)$. A realization $(A, B, C) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{m\times n}$ is called *signature symmetric* if

$$\begin{aligned}(AI_{pq})' &= AI_{pq},\\ (CI_{pq})' &= B\end{aligned} \tag{2.18}$$

holds. Note that every strictly proper symmetric rational $(m \times m)$-transfer function $G(s) = G(s)'$ of McMillan degree n has a minimal signature symmetric realization and any two such minimal signature symmetric realizations are similar by a unique state space similarity transformation $T \in \mathcal{O}_{pq}$. The set

$$\mathcal{O}_{pq} := \{T \in \mathbb{R}^{n\times n} | TI_{pq}T' = I_{pq}\}$$

is the real Lie group of pseudo-orthogonal $(n \times n)$-matrices stabilizing I_{pq} by congruence. The set $\mathcal{O}_{pq}^{+}$ denotes the identity component of $\mathcal{O}_{pq}$. Here $p - q$ is the Cauchy-Maslov index of $G(s)$, see [AB77] and [BD82]. For any stable signature symmetric realization the controllability and observability Gramians satisfy

$$W_o = I_{pq}W_cI_{pq}. \tag{2.19}$$

As usual, a realization (A, B, C) is called balanced if

$$W_c = W_o = \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \tag{2.20}$$

where the $\sigma_1, \ldots, \sigma_n$ are the Hankel singular values. In the sequel we assume that they are pairwise distinct.

Let

$$M(\Sigma) := \{T\Sigma T' \,|\, T \in \mathcal{O}_{pq}^{+}\}, \tag{2.21}$$

with Σ as in (2.20) assuming pairwise distinct Hankel singular values. Thus $M(\Sigma)$ is an orbit of $\mathcal{O}_{pq}^{+}$ and therefore a smooth and connected manifold. Note that the stabilizer subgroup of a point $X \in M(\Sigma)$ is finite and therefore $M(\Sigma)$ is diffeomorphic to $\mathcal{O}_{pq}^{+}$ which as a pseudo-orthogonal group of order $n = p + q$ has dimension $n(n-1)/2$.

Let $N := \operatorname{diag}(\mu_1, \ldots, \mu_p, \nu_1, \ldots, \nu_q)$ with $0 < \mu_1 < \cdots < \mu_p$ and $0 < \nu_1 < \cdots < \nu_q$. We then consider the smooth cost function

$$\begin{aligned}f_N :\ & M(\Sigma) \to \mathbb{R},\\ f_N(W) &:= \operatorname{tr}(NW).\end{aligned} \tag{2.22}$$

This choice is motivated by our previous work on balanced realizations [HH00], where we studied the smooth function $\operatorname{tr}(N(W_c+W_o))$ with diagonal positive definite N having distinct eigenvalues. Now

$$\begin{aligned}\operatorname{tr}(N(W_c+W_o)) &= \operatorname{tr}(N(W_c+I_{pq}W_cI_{pq})) \\ &= 2\operatorname{tr}(NW_c)\end{aligned}$$

by the above choice of a diagonal N. The following result summarizes the basic properties of the cost function f_N.

Theorem 2.3. *Let* $N := \operatorname{diag}(\mu_1, \ldots, \mu_p, \nu_1, \ldots, \nu_q)$ *with* $0 < \mu_1 < \cdots < \mu_p$ *and* $0 < \nu_1 < \cdots < \nu_q$. *For the smooth cost function* $f_N : M(\Sigma) \to \mathbb{R}$, *defined by* $f_N(W) := \operatorname{tr}(NW)$, *the following holds true.*

1. $f_N : M(\Sigma) \to \mathbb{R}$ *has compact sublevel sets and a minimum of* f_N *exists.*
2. $X \in M(\Sigma)$ *is a critical point for* $f_N : M(\Sigma) \to \mathbb{R}$ *if and only if* X *is diagonal.*
3. *The global minimum is unique and it is characterized by* $X = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, *where* $\sigma_1 > \cdots > \sigma_p$ *and* $\sigma_{p+1} > \cdots > \sigma_n$ *holds.*
4. *The Hessian of the function* f_N *at a critical point is nondegenerate.*

□

The constraint set for our cost function $f_N : M(\Sigma) \to \mathbb{R}$ is the Lie group $\mathcal{O}^+_{p,q}$ with Lie algebra $\mathfrak{o}_{pq}$. We choose a basis of $\mathfrak{o}_{pq}$ as

$$\Omega_{ij} := e_j e_i' - e_i e_j' \tag{2.23}$$

where $1 \leq i < j \leq p$ or $p+1 \leq i < j \leq n$ holds and

$$\Omega_{kl} := e_l e_k' + e_k e_l' \tag{2.24}$$

where $1 \leq k \leq p < l \leq n$ holds. These basis elements are defined via the standard basis vectors $e_1, \ldots, e_n$ of $\mathbb{R}^n$. Thus $\exp(t\Omega_{ij})$ is an orthogonal rotation with $(i,j)-$th sub matrix

$$\begin{bmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{bmatrix} \tag{2.25}$$

and $\exp(t\Omega_{kl})$ is a hyperbolic rotation with $(k,l)-$th sub matrix

$$\begin{bmatrix} \cosh t & \sinh t \\ \sinh t & \cosh t \end{bmatrix}. \tag{2.26}$$

Let N as in Theorem 2.3 above and let W be symmetric positive definite. Consider the smooth function

$$\begin{aligned} &\phi : \mathbb{R} \to \mathbb{R}, \\ &\phi(t) := \operatorname{tr}\left(N\,\mathrm{e}^{t\Omega}\,W\,\mathrm{e}^{t\Omega'}\right) \end{aligned} \tag{2.27}$$

where Ω denotes a fixed element of the above basis of $\mathfrak{o}_{pq}$. We have

Lemma 2.1. *1. For $\Omega = \Omega_{kl} = (\Omega_{kl})'$ as in (2.24) the function $\phi : \mathbb{R} \to \mathbb{R}$ defined by (2.27) is proper and bounded from below.*

2. A minimum

$$t_\Omega := \arg\min_{t\in\mathbb{R}} \phi(t) \in \mathbb{R} \tag{2.28}$$

exists for all $\Omega = \Omega_{ij} = -(\Omega_{ij})'$ where $1 \leq i < j \leq p$ or $p+1 \leq i < j \leq n$ holds, and exists as well for all $\Omega = \Omega_{kl} = (\Omega_{kl})'$ where $1 \leq k \leq p < l \leq n$ holds.

□

In [HHM02] the details are figured out. Moreover, a Jacobi method is presented for which local quadratic convergence is shown.

A Problem From Computer Vision

The Lie group $\mathcal{G}$ under consideration is the semidirect product $\mathcal{G} = \mathbb{R} \ltimes \mathbb{R}^2$. Here $\mathcal{G}$ acts linearly on the projective space $\mathbb{R}P^2$. A Jacobi-type method is formulated to minimize a smooth cost function $f : M \to \mathbb{R}$.

Consider the Lie algebra

$$\mathfrak{g} := \left\{ B = \begin{bmatrix} b_1 & b_2 & b_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} ;\ b_1, b_2, b_3 \in \mathbb{R} \right\} \tag{2.29}$$

with Lie bracket the matrix commutator. Exponentiating a general $B \in \mathfrak{g}$ gives us the representation of a general Lie group element

$$\exp(B) = I_3 + h(b_1)B \quad \text{with} \quad h(b_1) := \begin{cases} \dfrac{\mathrm{e}^{b_1} - 1}{b_1} & \text{for } b_1 \neq 0 \\ \\ 1 & \text{for } b_1 = 0 \end{cases} . \tag{2.30}$$

A one-parameter subgroup of

$$\mathcal{G} = \{A \in \mathbb{R}^{3\times 3} | A = I_3 + h(b_1)B, B \in \mathfrak{g}\} \tag{2.31}$$

is the smooth curve

$$\exp(tB) = I_3 + t \cdot h(t \cdot b_1)B. \tag{2.32}$$

Given a 3×3-matrix $N = N' > 0$ and let $M = \{X = ANA' | A \in \mathcal{G}\}$. Then M is a smooth and connected manifold. The tangent space of M at $X \in M$ is $T_X M = \{BX + XB' | B \in \mathfrak{g}\}$.

The stereo matching problem without correspondences can be formulated mathematically in the following way. Given two symmetric matrices $X, Q \in \mathbb{R}^{3\times 3}$

$$\begin{aligned} X &= \sum_{i=1}^{k} \begin{bmatrix} x_{1,i} \\ y_{1,i} \\ 1 \end{bmatrix} [x_{1,i},\ y_{1,i},\ 1], \\ Q &= \sum_{i=1}^{k} \begin{bmatrix} x_{2,i} \\ y_{2,i} \\ 1 \end{bmatrix} [x_{2,i}, y_{2,i}, 1]. \end{aligned} \tag{2.33}$$

In the sequel we will always assume that X and Q are positive definite. This assumption corresponds to a generic situation in the stereo matching problem. In the noise free case one can assume that there exists a group element $A \in \mathcal{G}$ such that

$$Q - AXA' = 0_3. \tag{2.34}$$

Our task then is to find such a matrix $A \in \mathcal{G}$. A convenient way to do so is using a variational approach as follows. Define the smooth cost function

$$\begin{aligned} f &: M \to \mathbb{R}, \\ f(X) &= \|Q - X\|^2, \end{aligned} \tag{2.35}$$

where $\|Y\|^2 := \sum_{i,j=1}^{3} y_{ij}^2$. The critical points of f are given by

Lemma 2.2. *The unique global minimum X_c of the function $f : M \to \mathbb{R}$, $f(X) = \|Q - X\|^2$ is characterized by $Q = X_c$. There are no further critical points.* □

Following the above approach we fix a basis of the Lie algebra $\mathfrak{g} = \langle B_1, B_2, B_3 \rangle$ with corresponding one-parameter subgroups of $\mathcal{G}$

$$A_i(t) = \mathrm{e}^{tB_i}, \ t \in \mathbb{R}, \quad i = 1, 2, 3. \tag{2.36}$$

Using an arbitrary ordering of the $A_1(t), A_2(t), A_3(t)$ the proposed algorithm then consists of a recursive application of sweep operations. In [HH98] it is shown that under reasonable assumptions this algorithm will converge quadratically. Moreover, numerical experiments indicate that only about five iterations are enough to reach the minimum.

2.1.4 Applications and Examples for Block Jacobi

If $M = \mathbb{R}^n$ one gets the socalled grouped variable version of the cyclic coordinate descent method, cf. [BHH$^+$87].

For applications with $M = \mathcal{O}_n \cdot x$ or $M = (\mathcal{O}_n \times \mathcal{O}_m) \cdot x$, cf. [Hüp96]. There, Kogbetliantz algorithms for singular value decompositions (2-dimensional optimization) and Block Jacobi for the real skewsymmetric eigenvalue problem (4-dimensional optimization) are considered. In contrast to the socalled onesided Jacobi methods for singular value computations, twosided methods essentially solve in each iteration step an optimization problem with two parameters. Similarly, as for the real symmetric eigenvalue problem, the subsets the cost function is restricted to in each step are compact, moreover, solving the restricted optimization problem is possible in closed form. The same holds true if one goes one step further, cf. [Hac93, Lut92, Mac95, Meh02, Paa71, RH95] or section 8.5.11 on Block Jacobi procedures in [GvL89] and references cited therein. The idea behind applying Block Jacobi methods to matrix eigenvalue problems is the following. Instead of zeroing out exactly one offdiagonal element (resp. two in the symmetric case) in each step, one produces a whole block of zeroes simultaneously outside the diagonal. Moreover, each such block is visited once per sweep operation. For all the papers cited above there exits a reinterpretation by the grouped variable approach, but this will not figured out here.

2.2 Local Convergence Analysis

We now come to the main result (Theorem 2.4) of this chapter, giving, under reasonable smoothness assumptions, sufficient conditions for a Jacobi-type algorithm to be efficient, i.e., being locally at least *quadratically* convergent.

Assumption 2.1.

1. The cost function $f : M \to \mathbb{R}$ is smooth. The cost f has a local minimum, say x_f, with nondegenerate Hessian at this minimum. The function f attains an isolated global minimum when restricting it to the image of the mappings $\gamma_i^{(x)}$.
2. All the partial algorithmic steps of the algorithm have x_f as a fixed point.
3. All the partial algorithmic steps are smooth mappings in an open neighborhood of the fixed point x_f. For this we require the (multi-)step size selection rule, i.e., computation of the set of t-parameters, to be smooth around x_f.

Remark 2.1. In the sequel of this chapter we will not assume less than C^∞-smoothness properties on mappings involved. This would sometimes obscure notation, moreover, for applications we have in mind, C^∞-smoothness is often guaranteed.

Theorem 2.4. *Consider the Block Jacobi Algorithm 2.4. Assume that Assumption 2.1 is fulfilled. Then this algorithm is locally quadratically convergent if the vector subspaces V_i from the direct sum decomposition*

$$T_{x_f}M = V_1 \oplus \cdots \oplus V_m$$

are mutually orthonormal with respect to the Hessian of the cost function f at the fixed point x_f.

Proof. The Block Jacobi Algorithm is defined as

$$\begin{aligned} s &: M \to M, \\ s(x) &= (r_m \circ \cdots \circ r_1)(x), \end{aligned}$$

i.e., a sweep consists of block minimzation steps, m in number. To be more precise, each partial algorithmic step is defined by a basic transformation

$$x \mapsto r_i(x) = G_i(t, x)\big|_{t=t_*^{(i)}}. \tag{2.37}$$

For each partial step $r_i : M \to M$ the fixed point condition holds

$$r_i(x_f) = x_f, \quad i = 1, \ldots, m. \tag{2.38}$$

The smoothness properties of each r_i around the fixed point x_f allows us to do analysis on M around x_f. The derivative of a sweep at $x \in M$ is the linear map

$$\mathrm{D}\, s(x) : T_x M \to T_{s(x)} M \tag{2.39}$$

assigning to any $\xi \in T_x M$ by the chain rule the value

$$\mathrm{D}\, s(x) \cdot \xi = \mathrm{D}\, r_m\big((r_{m-1} \circ \ldots \circ r_1)(x)\big) \cdot \ldots \cdot \mathrm{D}\, r_1(x) \cdot \xi. \tag{2.40}$$

That is, by the fixed point condition

$$\begin{aligned} \mathrm{D}\, s(x_f) &: T_{x_f} M \to T_{x_f} M, \\ \mathrm{D}\, s(x_f) \cdot \xi &= \mathrm{D}\, r_m(x_f) \cdot \ldots \cdot \mathrm{D}\, r_1(x_f) \cdot \xi \end{aligned} \tag{2.41}$$

holds. Let us take a closer look to the linear maps

$$\mathrm{D}\, r_i(x_f) : T_{x_f} M \to T_{x_f} M. \tag{2.42}$$

Omitting for a while any indexing, consider as before the maps of basic transformations

$$
\begin{aligned}
G : \mathbb{R}^l \times M &\to M, \\
G(t,x) &:= \gamma^{(x)}(t).
\end{aligned}
\tag{2.43}
$$

Now

$$
\mathrm{D}\, r(x_f) \cdot \xi = \Big(\mathrm{D}_1\, G(t,x) \cdot \mathrm{D}\, t(x) \cdot \xi + \mathrm{D}_2\, G(t,x) \cdot \xi \Big)_{x=x_f,\, t=t(x_f)}. \tag{2.44}
$$

Consider the smooth function

$$
\begin{aligned}
\psi : \mathbb{R}^{l_i} \times M &\to \mathbb{R}^{l_i}, \\
\psi(t,x) : &= \begin{bmatrix} \frac{\partial}{\partial t_1} \\ \vdots \\ \frac{\partial}{\partial t_{l_i}} \end{bmatrix} f(G(t,x)).
\end{aligned}
\tag{2.45}
$$

By definition of the multi-step size selection rule it follows that

$$
\psi(t(x),x) \equiv 0. \tag{2.46}
$$

Applying the Implicit Function Theorem to (2.46) one can get an expression for the derivative of the multi-step size, $\mathrm{D}\, t(x_f) \cdot \xi$. We will use the following abbreviations:

$$
\widetilde{\xi}_j := \dot{\gamma}_j^{(x_f)}(0) \quad \text{for all} \quad j = 1, \ldots, l_i,
$$

$$
\mathsf{H}(\widetilde{\xi}_j, \widetilde{\xi}_i) := \mathrm{D}^2 f(x_f) \cdot (\widetilde{\xi}_j, \widetilde{\xi}_i),
$$

$$
\mathcal{H} := \big(h_{ij}\big), \quad h_{ij} := \mathsf{H}(\widetilde{\xi}_i, \widetilde{\xi}_j).
$$

Finally, we get, using a hopefully not too awkward notation,

$$\begin{aligned}\mathrm{D}\, r_i(x_f)\cdot\xi &= \xi - \sum_{j=1}^{l_i} \widetilde{\xi}_j (\mathcal{H}^{-1})_{j-\text{th row}} \cdot \begin{bmatrix}\mathsf{H}(\widetilde{\xi}_1,\xi)\\ \vdots \\ \mathsf{H}(\widetilde{\xi}_{l_i},\xi)\end{bmatrix}\\ &= \underbrace{\left(\mathrm{id} - \begin{bmatrix}\widetilde{\xi}_1 & \dots & \widetilde{\xi}_{l_i}\end{bmatrix}\mathcal{H}^{-1}\begin{bmatrix}\mathsf{H}(\widetilde{\xi}_1,(\,\cdot\,))\\ \vdots \\ \mathsf{H}(\widetilde{\xi}_{l_i},(\,\cdot\,))\end{bmatrix}\right)}_{=:Q_i}\xi\end{aligned} \tag{2.47}$$

Note that $\widetilde{\xi}_i := \dot{\gamma}_i^{(x_f)}(0) \in V_i^{(x_f)} \subset T_{x_f}M$. Therefore, by the chain rule, the derivative of one sweep acting on an arbitrary tangent vector $\xi \in T_{x_f}M$ evaluated at the fixed point (minimum) x_f is as

$$\mathrm{D}\, s(x_f)\cdot\xi = Q_m \cdot \ldots \cdot Q_1 \cdot \xi. \tag{2.48}$$

For convenience we will switch now to ordinary matrix vector notation,

$$\begin{aligned}\widetilde{\xi}_{l_1}, \xi \in T_{x_f}M \quad &\longleftrightarrow \quad \widetilde{\xi}_{l_1}, \xi \in \mathbb{R}^n,\\ \mathrm{D}_2\, f(x_f) : T_{x_f}M \times T_{x_f}M \to \mathbb{R} \quad &\longleftrightarrow \quad \mathsf{H} = \mathsf{H}^\top \in \mathbb{R}^{n\times n},\\ \mathrm{id} : T_{x_f}M \to T_{x_f}M \quad &\longleftrightarrow \quad I_n.\end{aligned}$$

That is, rewriting the right hand side of (2.47)

$$\begin{aligned}Q_i\xi &= \left(I_n - \underbrace{\begin{bmatrix}\widetilde{\xi}_1 & \dots & \widetilde{\xi}_{l_i}\end{bmatrix}}_{=:\widetilde{\Xi}_i}\begin{bmatrix}\mathsf{H}(\widetilde{\xi}_1,\widetilde{\xi}_1) & \cdots & \mathsf{H}(\widetilde{\xi}_1,\widetilde{\xi}_{l_i})\\ \vdots & & \vdots\\ \mathsf{H}(\widetilde{\xi}_{l_i},\widetilde{\xi}_1) & \cdots & \mathsf{H}(\widetilde{\xi}_{l_i},\widetilde{\xi}_{l_i})\end{bmatrix}^{-1}\begin{bmatrix}\mathsf{H}(\widetilde{\xi}_1,(\,\cdot\,))\\ \vdots \\ \mathsf{H}(\widetilde{\xi}_{l_i},(\,\cdot\,))\end{bmatrix}\right)\xi\\ &= \left(I_n - \widetilde{\Xi}_i(\widetilde{\Xi}_i^\top \mathsf{H}\widetilde{\Xi}_i)^{-1}\widetilde{\Xi}_i^\top\mathsf{H}\right)\xi.\end{aligned}$$

We want to examine under which conditions $\mathrm{D}\, s(x_f) = 0$, i.e., we want to examine to which conditions on the subspaces $V_i^{(x_f)}$ the condition

$$Q_m \cdot \ldots \cdot Q_1 \equiv 0$$

is equivalent to. It is easily seen that for all $i = 1, \ldots, m$

$$\begin{aligned} P_i &:= \mathsf{H}^{\frac{1}{2}} Q_i \mathsf{H}^{-\frac{1}{2}} \\ &= I_n - (\mathsf{H}^{\frac{1}{2}}\widetilde{\Xi}_i)\Big((\mathsf{H}^{\frac{1}{2}}\widetilde{\Xi}_i)^\top(\mathsf{H}^{\frac{1}{2}}\widetilde{\Xi}_i)\Big)^{-1}(\mathsf{H}^{\frac{1}{2}}\widetilde{\Xi}_i)^\top, \end{aligned} \tag{2.49}$$

are orthogonal projection operators, i.e.,

$$P_i = P_i^2 = P_i^\top, \quad \text{for all} \quad i = 1, \ldots, m, \tag{2.50}$$

holds true. Therefore,

$$Q_m \cdot \ldots \cdot Q_1 = 0 \quad \Leftrightarrow \quad P_m \cdot \ldots \cdot P_1 = 0. \tag{2.51}$$

To proceed we need a lemma.

Lemma 2.3. *Consider $\mathbb{R}^n$ with usual inner product. Consider orthogonal projection matrices $P_i = P_i^\top = P_i^2$, $i = 1, \ldots, m$ with $m \leq n$. We require*

$$\operatorname{rk} P_i = n - k_i \quad \textit{and} \quad \sum_{j=1}^{m} k_j = n. \tag{2.52}$$

Then the following holds true

$$P_m \cdot P_{m-1} \cdot \ldots \cdot P_2 \cdot P_1 = 0 \tag{2.53}$$

$$\Leftrightarrow$$

$$\ker P_i \perp \ker P_j \quad \textit{for all} \quad i \neq j.$$

Proof of Lemma 2.3. We prove the “only if”-part, the “if”-part is immediate. Each projection matrix can be represented as

$$P_i = I_n - X_i X_i^\top \tag{2.54}$$

with $X_i \in \mathrm{St}_{k_i,n}$, i.e., a full rank matrix $X_i \in \mathbb{R}^{n \times k_i}$ with orthonormal columns, $X_i^\top X_i = I_{k_i}$.

Claim 2.1. *The equation (2.53)*

$$P_m \cdot P_{m-1} \cdot \ldots \cdot P_2 \cdot P_1 = 0$$

holds if and only if there exists $\Theta \in \mathcal{O}_n$*, such that*

$$\widetilde{P}_i = \Theta P_i \Theta^\top, \quad \textit{for all} \quad i = 1, \ldots, m, \tag{2.55}$$

satisfy

1. $$\widetilde{P}_m \cdot \ldots \cdot \widetilde{P}_1 = 0, \tag{2.56}$$

2. $$\widetilde{P}_i = \begin{bmatrix} * & 0 \\ 0 & I_{n-k_1-\ldots-k_i} \end{bmatrix}. \tag{2.57}$$

Proof of Claim 2.1. Without loss of generality (2.57) holds for $i = 1$. To see that (2.57) holds also for $i = 2$ consider an orthogonal matrix $\Theta_2 \in \mathcal{O}_n$ of block diagonal form

$$\Theta_2 := \begin{bmatrix} I_{k_1} & 0 \\ 0 & U_2 \end{bmatrix},$$

with orthogonal submatrix $U_2 \in \mathcal{O}_{n-k_1}$. Clearly, such a Θ_2 stabilizes $\widetilde{P}_1$, i.e.,

$$\Theta_2 \widetilde{P}_1 \Theta_2^\top = \widetilde{P}_1. \tag{2.58}$$

Moreover, Θ_2, (respectively, U_2) can be chosen such as to block diagonalize P_2 as

$$\Theta_2 P_2 \Theta_2^\top = I_n - \Theta_2 X_2 X_2^\top \Theta_2^\top = \begin{bmatrix} * & 0 \\ 0 & I_{n-k_1-k_2} \end{bmatrix} = \widetilde{P}_2, \tag{2.59}$$

by requiring the product $\Theta_2 X_2$ to be as

$$\Theta_2 X_2 = \begin{bmatrix} I_{k_1} & 0 \\ 0 & U_2 \end{bmatrix} X_2 = \begin{bmatrix} * \\ 0_{(n-k_1-k_2)\times(k_2)} \end{bmatrix} \in \mathrm{St}_{k_2,n}\,. \tag{2.60}$$

Recall that $n - k_1 \geq k_2$. Now proceeding inductively using

$$\Theta_l X_l = \begin{bmatrix} I_{k_{l-1}} & 0 \\ 0 & U_l \end{bmatrix} X_l = \begin{bmatrix} * \\ 0_{(n-k_1-\cdots-k_l)\times(k_l)} \end{bmatrix} \in \mathrm{St}_{k_l,n} \tag{2.61}$$

for $l = 3, \ldots, m-1$, with suitably chosen $U_l \in \mathcal{O}_{n-k_1-\cdots-k_{l-1}}$ proves (2.57). By defining $\Theta \in \mathcal{O}_n$ as

$$\Theta := \Theta_{m-1} \cdots \Theta_1$$

where

$$\Theta_1 P_1 \Theta_1^\top = \begin{bmatrix} 0_{k_1} & 0 \\ 0 & I_{n-k_1} \end{bmatrix} = \widetilde{P}_1,$$

Claim 2.1 follows. □

Proof of Lemma 2.3 continued. By Claim 2.1 the product $\widetilde{P}_{m-1} \cdot \ldots \cdot \widetilde{P}_1$ takes the block diagonal form

$$\begin{aligned} \widetilde{P}_{m-1} \cdots \widetilde{P}_1 &= \begin{bmatrix} * & 0 \\ 0 & I_{n-k_1-\cdots-k_{m-1}} \end{bmatrix} \cdot \ldots \cdot \begin{bmatrix} * & 0 \\ 0 & I_{n-k_1-k_2} \end{bmatrix} \cdot \begin{bmatrix} 0_{k_1} & 0 \\ 0 & I_{n-k_1} \end{bmatrix} \\ &= \begin{bmatrix} * & 0 \\ 0 & I_{n-k_1-\cdots-k_{m-1}} \end{bmatrix} \\ &= \begin{bmatrix} * & 0 \\ 0 & I_{k_m} \end{bmatrix}. \end{aligned} \tag{2.62}$$

Recall that $\operatorname{rk} \widetilde{P}_m = n - k_m$. Therefore we have the implications

$$\begin{aligned} \widetilde{P}_m \cdot (\widetilde{P}_{m-1} \cdots \widetilde{P}_1) = 0 \Rightarrow \quad \widetilde{P}_m &= \begin{bmatrix} * & 0 \\ 0 & 0_{k_m} \end{bmatrix} \\ \Rightarrow \quad \widetilde{P}_m &= \begin{bmatrix} I_{n-k_m} & 0 \\ 0 & 0_{k_m} \end{bmatrix}. \end{aligned} \tag{2.63}$$

Now we proceed by working off the remaining product $\widetilde{P}_{m-1} \cdot (\widetilde{P}_{m-2} \cdots \widetilde{P}_1)$ from the left.

Analogously to (2.62) we have

$$\widetilde{P}_{m-2}\cdots\widetilde{P}_1 = \begin{bmatrix} * & 0 & 0 \\ 0 & I_{k_{m-1}} & 0 \\ 0 & 0 & I_{k_m} \end{bmatrix}, \tag{2.64}$$

and similarly to (2.63) we have the implications

$$\widetilde{P}_{m-1}(\widetilde{P}_{m-2}\cdots\widetilde{P}_1) = \begin{bmatrix} 0_{n-k_m} & 0 \\ 0 & I_{k_m} \end{bmatrix} \Rightarrow \widetilde{P}_{m-1} = \begin{bmatrix} * & 0 & 0 \\ 0 & 0_{k_{m-1}} & 0 \\ 0 & 0 & I_{k_m} \end{bmatrix}$$

$$\Rightarrow \widetilde{P}_{m-1} = \begin{bmatrix} I_{k_1+\ldots+k_{m-2}} & 0 & 0 \\ 0 & 0_{k_{m-1}} & 0 \\ 0 & 0 & I_{k_m} \end{bmatrix}.$$

The result of Lemma 2.3 follows then by induction, i.e.,

$$\widetilde{P}_i = \begin{bmatrix} I_{k_1+\ldots+k_{i-1}} & 0 & 0 \\ 0 & 0_{k_i} & 0 \\ 0 & 0 & I_{k_{i+1}+\ldots+k_m} \end{bmatrix}$$

holds true for all $i = 2, \ldots, m-1$,

$$\widetilde{P}_1 = \begin{bmatrix} 0_{k_1} & 0 \\ 0 & I_{n-k_1} \end{bmatrix},$$

and

$$\widetilde{P}_m = \begin{bmatrix} I_{n-k_m} & 0 \\ 0 & 0_{k_m} \end{bmatrix}.$$

□

Proof of Theorem 2.4 continued. Finishing the proof of our theorem we therefore can state that

$$\mathrm{D}\, s(x_f)\cdot\xi = \mathrm{D}\, r_m(x_f)\cdot\ldots\cdot\mathrm{D}\, r_1(x_f) = 0$$

holds true if the direct sum decomposition

$$T_{x_f}M = V_1 \oplus \cdots \oplus V_m$$

is also orthonormal with respect to the Hessian of our objective function f at the fixed point (minimum) x_f. The result follows by the Taylor-type argument

$$\|x_{k+1} - x_f\| \leq \sup_{z \in \overline{U}} \| \mathrm{D}^2 \, s(z) \| \cdot \|x_k - x_f\|^2.$$

□

2.3 Discussion

From our point of view there are several advantages of the calculus approach we have followed here. It turns out that the ordering partial algorithmic steps are worked off do not play a role for the quadratic convergence. Forinstance for the symmetric eigenvalue problem several papers have been published to show that row-cyclic and column-cyclic strategies both ensure quadratic convergence. Our approach now shows that the convergence properties do not depend on the ordering in general.

Exploiting the differentiability properties of the algorithmic maps offers a much more universal methodology for showing quadratic convergence than sequences of tricky estimates usually do. It is e.g. often the case that estimates used for $\mathcal{O}_n$-related problems may not be applicable to $\mathcal{GL}_n$-related ones and vice versa. On the other hand computing the derivative of an algorithm is always the same type of calculation. But the most important point seems to be the fact that our approach shows quadratic convergence of a matrix algorithm itself. If one looks in text books on matrix algorithms usually higher order convergence is understood as a property of a scalar valued cost function (which can even just the norm of a subblock) rather than being a property of the algorithm itself considered as a selfmap of some manifold.

Chapter 3

Refining Estimates of Invariant Subspaces

We are interested in refining estimates of invariant subspaces of real nonsymmetric matrices which are already “nearly” block upper triangular. The idea is the following. The Lie group of real unipotent lower block triangular $(n \times n)$-matrices acts by similarity on such a given nearly block upper triangular matrix. We will develop several algorithms consisting on similarity transformations, such that after each algorithmic step the matrix is closer to perfect upper block triangular form. We will show that these algorithms are efficient, meaning that under certain assumptions on the starting matrix, the sequence of similarity transformed matrices will converge locally quadratically fast to a block upper triangular matrix. The formulation of these algorithms, as well as their convergence analysis, are presented in a way, such that the concrete block sizes chosen initially do not matter. Especially, in applications it is often desirable for complexity reasons that a real matrix which is close to its *real* Schur form, cf. p.362 [GvL89], is brought into *real* Schur form by using exclusively *real* similarities instead of switching to complex ones.

In this chapter we always work over $\mathbb{R}$. The generalization to $\mathbb{C}$ is immediate and we state without proof that all the results from this chapter directly apply to the complex case.

The outline of this chapter is as follows. After introducing some notation we will focus on an algorithm consisting on similarity transformations by unipotent lower block triangular matrices. Then we refine this approach by using orthogonal transformations instead, to improve numerical accuracy.

The convergence properties of the orthogonal algorithm then will be an immediate consequence of the former one.

3.1 Lower Unipotent Block Triangular Transformations

Let denote $V \subset \mathbb{R}^{n\times n}$ the subvector space of real block upper triangular $(n \times n)-$matrices

$$V := \{X \in \mathbb{R}^{n\times n} | X_{ij} = 0_{n_i\times n_j} \; \forall\, 1 \leq j < i \leq r\}\}, \tag{3.1}$$

i.e., an arbitrary element $X \in V$ looks like

$$X = \begin{bmatrix} X_{11} & \cdots & \cdots & X_{1r} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & X_{rr} \end{bmatrix} \tag{3.2}$$

the diagonal subblocks $X_{ii} \in \mathbb{R}^{n_i\times n_i}$, $i = 1, \ldots, r$, being square and therefore

$$\sum_{i=1}^{r} n_i = n.$$

Let denote $\mathcal{L}_n$ the Lie group of real unipotent lower block triangular $(n \times n)$-matrices with partitioning according to V

$$\mathcal{L}_n := \left\{ X \in \mathbb{R}^{n\times n} \;\middle|\; \begin{array}{ll} X_{kk} = I_{n_k} & \forall\, 1 \leq k \leq r, \\ X_{ij} = 0_{n_i\times n_j} & \forall\, 1 \leq i < j \leq r \end{array} \right\}, \tag{3.3}$$

i.e., an arbitrary element $X \in \mathcal{L}_n$ looks like

$$X = \begin{bmatrix} I_{n_1} & 0 & \cdots & 0 \\ X_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ X_{n_r,1} & \cdots & X_{n_r,n_{r-1}} & I_{n_r} \end{bmatrix}. \tag{3.4}$$

Given a real block upper triangular matrix

$$A = \begin{bmatrix} A_{11} & \cdots & \cdots & A_{1r} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{rr} \end{bmatrix} \tag{3.5}$$

consider the orbit $M_{\mathcal{L}_n}$ of A under similarity action σ of $\mathcal{L}_n$.

$$\begin{aligned} \sigma : \mathcal{L}_n \times \ V &\to \mathbb{R}^{n\times n}, \\ (L, X) &\mapsto LXL^{-1}, \end{aligned} \tag{3.6}$$

and

$$M_{\mathcal{L}_n} := \{X \in \mathbb{R}^{n\times n} \,|\, X = LAL^{-1},\ L \in \mathcal{L}_n\}. \tag{3.7}$$

In this chapter we will make the following assumptions:

Assumption 3.1. Let A as in (3.5). The spectra of the diagonal subblocks A_{ii}, for $i = 1, \ldots, r$, of A are mutually disjoint.

Our first result shows that any matrix lying in a sufficiently small neighborhood of A which fulfils Assumption 3.1, is then element of an $\mathcal{L}_n$-orbit of some other matrix, say B, which also fulfils Assumption 3.1.

Let $A \in \mathbb{R}^{n\times n}$ fulfil Assumption 3.1. Consider the smooth mapping

$$\begin{aligned} \sigma : \mathcal{L}_n \times V &\to \mathbb{R}^{n\times n}, \\ \sigma(L, X) &= LXL^{-1}. \end{aligned} \tag{3.8}$$

Lemma 3.1. *The mapping σ defined by (3.8) is locally surjective around (I, A).*

Proof. Let denote $\mathfrak{l}_n$ the Lie algebra of real lower block triangular $(n \times n)$-matrices

$$\mathfrak{l}_n := \left\{ X \in \mathbb{R}^{n\times n} \,\middle|\, \begin{matrix} X_{kk} = 0_{n_k} & \forall\, 1 \le k \le r, \\ X_{ij} = 0_{n_i \times n_j} & \forall\, 1 \le i < j \le r \end{matrix} \right\}, \tag{3.9}$$

i.e., an arbitrary element $X \in \mathfrak{l}_n$ looks like

$$X = \begin{bmatrix} 0_{n_1} & 0 & \cdots & 0 \\ X_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ X_{n_r,1} & \cdots & X_{n_r,n_r-1} & 0_{n_r} \end{bmatrix}. \tag{3.10}$$

It is sufficient to show that the derivative

$$\mathrm{D}\,\sigma(I,A) : \mathfrak{l}_n \times V \to \mathbb{R}^{n\times n} \tag{3.11}$$

is locally surjective. For arbitrary $l \in \mathfrak{l}_n$ and for arbitrary $a \in V$ the following holds true

$$\mathrm{D}\,\sigma(I,A)\cdot(l,a) = lA - Al + a. \tag{3.12}$$

We show that for any $h \in \mathbb{R}^{n\times n}$ the linear system

$$lA - Al + a = h \tag{3.13}$$

has a solution in terms of $l \in \mathfrak{l}_n$ and $a \in V$. By decomposing into block upper triangular and *strictly* block lower triangular parts

$$h = h_{\text{bl.upp.}} + h_{\text{str.bl.low.}} \tag{3.14}$$

and because $a \in V$ is already block upper triangular it remains to show that the strictly lower block triangular part of (3.13)

$$(lA - Al)_{\text{str.bl.low}} = h_{\text{str.bl.low.}} \tag{3.15}$$

can be solved for $l \in \mathfrak{l}_n$. We partition into "blocks of subblocks"

$$l = \begin{bmatrix} l_{11} & 0 \\ l_{\widetilde{21}} & l_{\widetilde{22}} \end{bmatrix},$$

$$A = \begin{bmatrix} A_{11} & A_{\widetilde{12}} \\ 0 & A_{\widetilde{22}} \end{bmatrix},$$

$$h_{\text{str.low.bl.}} = \begin{bmatrix} (h_{11})_{\text{str.low.bl.}} & 0 \\ h_{\widetilde{21}} & (h_{\widetilde{22}})_{\text{str.low.bl.}} \end{bmatrix},$$

accordingly, i.e., $A_{11} \in \mathbb{R}^{n_1\times n_1}$ and $l_{11} = 0_{n_1}$ as before. Thus one has to solve for $l_{\widetilde{21}}$ and $l_{\widetilde{22}}$. Considering the $(\widetilde{21})-$block of (3.15) gives

$$l_{\widetilde{21}}A_{11} - A_{\widetilde{22}}l_{\widetilde{21}} = h_{\widetilde{21}}, \tag{3.16}$$

By Assumption 3.1, the Sylvester equation (3.16) can be solved uniquely for $l_{\widetilde{21}}$, i.e., the block $l_{\widetilde{21}}$ is therefore fixed now. Applying an analogous argumentation to the $(\widetilde{22})-$block of (3.15)

$$l_{\widetilde{22}}A_{\widetilde{22}} - A_{\widetilde{22}}l_{\widetilde{22}} = -l_{\widetilde{21}}A_{\widetilde{12}} + (h_{\widetilde{22}})_{\text{str.low.}}, \tag{3.17}$$

and by continuing inductively ($l := l_{\widetilde{22}}$, $A := A_{\widetilde{22}}$, etc.) by partitioning into smaller blocks of subblocks of the remaining diagonal blocks A_{ii}, $i = 2, \ldots, r$, gives the result. □

Let $A \in \mathbb{R}^{n\times n}$ fulfil Assumption 3.1. Let

$$M_{\mathcal{L}_n} := \big\{X \in \mathbb{R}^{n\times n} | X = LAL^{-1}, L \in \mathcal{L}_n\big\}. \tag{3.18}$$

The next lemma characterizes the $\mathcal{L}_n$-orbit of the matrix A.

Lemma 3.2. *$M_{\mathcal{L}_n}$ is diffeomorphic to $\mathcal{L}_n$.*

Proof. The set $M_{\mathcal{L}_n}$ is a smooth manifold, because it is the orbit of a semi-algebraic group action, see p.353 [Gib79]. We will show that the stabilizer subgroup $\operatorname{stab}(A) \subset \mathcal{L}_n$ equals the identity $\{I\}$ in $\mathcal{L}_n$, i.e., the only solution in terms of $L \in \mathcal{L}_n$ for

$$LAL^{-1} = A \quad \Longleftrightarrow \quad [L, A] = 0 \tag{3.19}$$

is $L = I$. Partition L into blocks of blocks

$$L = \begin{bmatrix} I_{n_1} & 0 \\ L_{\widetilde{21}} & L_{\widetilde{22}} \end{bmatrix}$$

where the second diagonal block $L_{\widetilde{22}} \in \mathcal{L}_{n-n_1}$. Let

$$A = \begin{bmatrix} A_{11} & A_{\widetilde{12}} \\ 0 & A_{\widetilde{22}} \end{bmatrix}$$

be accordingly to L partitioned. The $(\widetilde{21})-$block of the equation $[L, A] = 0$ is as

$$L_{\widetilde{21}}A_{11} - A_{\widetilde{22}}L_{\widetilde{21}} = 0. \tag{3.20}$$

By Assumption 3.1 on the spectrum of A, equation (3.20) implies $L_{\widetilde{21}} = 0$. By recursive application of this argumentation to the $(\widetilde{22})-$block of (3.19) the result follows. Therefore, $L = I$ implies $\operatorname{stab}(A) = \{I\}$ and hence

$$M_{\mathcal{L}_n} \cong \mathcal{L}_n / \operatorname{stab}(A) = \mathcal{L}_n \tag{3.21}$$

□

3.2 Algorithms

3.2.1 Main Ideas

The algorithms presented in this chapter for the iterative refinement of invariant subspaces of nonsymmetric real matrices are driven by the following ideas.

Let the matrix A be partitioned as in

$$A = \begin{bmatrix} A_{11} & \cdots & \cdots & A_{1r} \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & A_{rr} \end{bmatrix} \tag{3.22}$$

and fulfilling Assumption 3.1. Consider an $X \in M_{\mathcal{L}_n}$, $M_{\mathcal{L}_n}$ the A-orbit of the similarity action by $\mathcal{L}_n$. Assume X is sufficiently close to A, i.e.,

$$\|X - A\| < \Delta_\lambda \tag{3.23}$$

holds, where $\|Z\| := \sqrt{\operatorname{tr}(ZZ^\top)}$ and Δ_λ denotes the absolute value of the smallest difference of two eigenvalues of A which correspond to *different* diagonal subblocks of A. Obviously,

$$\operatorname{span}\left(\begin{bmatrix} 0_{(n_1+\ldots+n_{i-1})\times n_i} \\ I_{n_i} \\ 0_{(n_{i+1}+\ldots+n_r)\times n_i} \end{bmatrix}\right) \tag{3.24}$$

is then for all $i = 1, \ldots, n$ a good approximation for an n_i-dimensional right invariant subspace of X, because by assumption (3.23) on X, for all $j > i$

$$\|X_{ji}\| \quad \text{is small.} \tag{3.25}$$

Consider an $L^{(\alpha)} \in \mathcal{L}_n$ of the following partitioned form

$$L^{(\alpha)} := \begin{bmatrix} I_{n_1} & & & & \\ & \ddots & & & \\ & & I_{n_\alpha} & & \\ & & p^{(\alpha+1,\alpha)} & \ddots & \\ & & \vdots & & \ddots \\ & & p^{(r,\alpha)} & & & I_{n_r} \end{bmatrix}, \tag{3.26}$$

where empty blocks are considered to be zero ones. We want to compute

$$P^{(\alpha)} := \begin{bmatrix} p^{(\alpha+1,\alpha)} \\ \vdots \\ p^{(r,\alpha)} \end{bmatrix} \in \mathbb{R}^{(n_{\alpha+1}+\ldots+n_r)\times n_\alpha}, \tag{3.27}$$

such that

$$L_\alpha X L_\alpha^{-1} = \begin{bmatrix} I_{n_1+\ldots+n_{\alpha-1}} & 0 & 0 \\ 0 & I_{n_\alpha} & 0 \\ 0 & P^{(\alpha)} & I_{n_{\alpha+1}+\ldots+n_r} \end{bmatrix} X \begin{bmatrix} I_{n_1+\ldots+n_{\alpha-1}} & 0 & 0 \\ 0 & I_{n_\alpha} & 0 \\ 0 & -P^{(\alpha)} & I_{n_{\alpha+1}+\ldots+n_r} \end{bmatrix}$$

$$= Z, \tag{3.28}$$

where Z is of the form

$$Z = \begin{bmatrix} Z_{11} & \cdots & \cdots & \cdots & \cdots & \cdots & Z_{1,r} \\ \vdots & \ddots & & & & & \vdots \\ \vdots & & Z_{\alpha-1,\alpha-1} & & & & \vdots \\ \vdots & & \vdots & Z_{\alpha,\alpha} & & & \vdots \\ \vdots & & \vdots & 0 & Z_{\alpha+1,\alpha+1} & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ Z_{r1} & \cdots & Z_{r,\alpha-1} & 0 & Z_{r,\alpha+1} & \cdots & Z_{r,r} \end{bmatrix}, \tag{3.29}$$

i.e., the blocks below the diagonal block $Z_{\alpha,\alpha}$ are zero. For convenience we assume for a while without loss of generality that $r = 2$. Therefore, we want to solve the (21)-block of

$$\begin{bmatrix} I & 0 \\ P^{(1)} & I \end{bmatrix} \cdot \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ -P^{(1)} & I \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix} \tag{3.30}$$

in terms of $P^{(1)}$, i.e., we want to solve the matrix valued algebraic Riccati equation

$$P^{(1)}X_{11} + X_{21} - P^{(1)}X_{12}P^{(1)} - X_{22}P^{(1)} = 0. \tag{3.31}$$

As a matter of fact, (3.31) is in general not solvable in closed form. As a consequence authors have suggested several different approaches to solve (3.31) iteratively. See [Cha84] for Newton-type iterations on the noncompact Stiefel manifold and [DMW83, Ste73] for iterations like

$$P_{i+1}X_{11} - X_{22}P_{i+1} = P_iX_{12}P_i - X_{21}, \quad P_0 = 0. \tag{3.32}$$

Moreover, see [Dem87] for a comparison of the approaches of the former three papers. For quantitative results concerning Newton-type iterations to solve Riccati equations see also [Nai90].

A rather natural idea to solve (3.31) approximately is to ignore the second order term, $-P^{(1)}X_{12}P^{(1)}$, and solve instead the Sylvester equation

$$P^{(1)}X_{11} + X_{21} - X_{22}P^{(1)} = 0. \tag{3.33}$$

Note that by Assumption 3.1 equation (3.33) is uniquely solvable.

Now we switch back to the general case where the number r of invariant subspaces to be computed is not necessarily equal to 2. Having in mind sweep-type algorithms it is natural to formulate an algorithm which solves an equation like (3.33) for $P^{(1)}$, respecting (3.26)-(3.29), say, then transform X according to $X \mapsto L_1XL_1^{-1}$, do the same for $P^{(2)}$, and so forth. One can show that such an algorithm would be a differentiable map around A. Moreover, local quadratic convergence could be proved by means of analysis. But the story will not end here as we will see now.

Instead of solving a Sylvester equation for

$$P^{(\alpha)} = \begin{bmatrix} p^{(\alpha+1,\alpha)} \\ \vdots \\ p^{(r,\alpha)} \end{bmatrix}, \tag{3.34}$$

i.e., solving for the corresponding block of (3.28), one could refine the algorithm reducing complexity by solving Sylvester equations of lower dimension in a cyclic manner, i.e., perform the algorithm block wise on each $p^{(ij)} \in \mathbb{R}^{n_i \times n_j}$. In principle one could refine again and again reaching finally the scalar case but then, not necessarily all Sylvester equations could be solved, because within a diagonal block we did not assume anything on the spectrum. On the other hand, if the block sizes were 1×1, e.g., if one already knew that all the eigenvalues of A were distinct, then the resulting scalar algebraic Riccati equations were solvable in closed form, being just quadratics. We would like to mention that such an approach would come rather close to [BGF91, CD89, Ste86] where the authors studied Jacobi-type methods for solving the nonsymmetric (gerneralized) eigenvalue problem.

3.2.2 Formulation of the Algorithm

The following algorithm will be analyzed. Given an $X \in M_{\mathcal{L}_n}$ and let A fulfil Assumption 3.1. Assume further that X is sufficiently close to A. Consider the index set

$$\mathcal{I} := \{(ij)\}_{i=2,\ldots,r;j=1,\ldots,r-1} \tag{3.35}$$

and fix an ordering, i.e., a surjective map

$$\beta : \mathcal{I} \to \left\{1, \ldots, \binom{r}{2}\right\}. \tag{3.36}$$

For convenience we rename double indices in the discription of the algorithm by simple ones by means of $X_{ij} \mapsto X_{\beta(ij)}$ respecting the ordering β.

Algorithm 3.1 (Sylvester Sweep).

Given an $X \in M_{\mathcal{L}_n}$. Define

$$\begin{aligned}
X_k^{(1)} &:= L_1 X L_1^{-1}\\
X_k^{(2)} &:= L_2 X_k^{(1)} L_2^{-1}\\
&\ \vdots\\
X_k^{\binom{r}{2}} &:= L_{\binom{r}{2}} X_k^{\binom{r}{2}-1} L_{\binom{r}{2}}^{-1}
\end{aligned}$$

where for $l = 1, \dots, \binom{r}{2}$, the transformation matrix $L_l \in \mathcal{L}_n$ differs from the identity matrix I_n only by the ij-th block, say p_l.
Here $p_l \in \mathbb{R}^{n_j \times n_i}$, $\beta((ij)) = l$, and p_l solves the Sylvester equation

$$p_l \left(X_k^{(l-1)}\right)_{jj} - \left(X_k^{(l-1)}\right)_{ii} p_l + \left(X_k^{(l-1)}\right)_{ij} = 0.$$

The overall algorithm then consists of the iteration of sweeps.

Algorithm 3.2 (Refinement of Estimates of Subspaces).

- Let $X_0, \dots, X_k \in M_{\mathcal{L}_n}$ be given for $k \in \mathbb{N}_0$.
- Define the recursive sequence $X_k^{(1)}, \dots, X_k^{\binom{r}{2}}$ as above (sweep).
- Set $X_{k+1} := X_k^{\binom{r}{2}}$. Proceed with the next sweep.

For convenience let us write down one sweep for $r = 3$. Fixing the ordering

$$\beta\big((21)\big) = 1, \quad \beta\big((31)\big) = 2, \quad \beta\big((32)\big) = 3, \tag{3.37}$$

we have for

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix}, \qquad X = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{23} & A_{33} \end{bmatrix},$$

$$X_1 = \begin{bmatrix} I & 0 & 0 \\ p_1 & I & 0 \\ 0 & 0 & I \end{bmatrix} X \begin{bmatrix} I & 0 & 0 \\ -p_1 & I & 0 \\ 0 & 0 & I \end{bmatrix},$$

$$X_2 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ p_2 & 0 & I \end{bmatrix} X_1 \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -p_2 & 0 & I \end{bmatrix},$$

$$X_3 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & p_3 & I \end{bmatrix} X_2 \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & -p_3 & I \end{bmatrix}.$$

In contrast to our Jacobi approach in Chapter 2, where the analysis showed that the ordering, the partial algorithmic steps where worked off, did not influence the quadratic convergence properties, the ordering in the present case will do.

For the index set

$$\mathcal{I} := \{(ij)\}_{i=2,\dots,r;j=1,\dots,r-1}$$

we fix the following two different orderings. The first one is as

$$\begin{aligned}
\beta_{\text{col}} : \mathcal{I} \to &\left\{1, \dots, \binom{r}{2}\right\}, \\
\beta_{\text{col}}\big((r1)\big) &= 1, \\
&\vdots \\
\beta_{\text{col}}\big((21)\big) &= r-1, \\
\beta_{\text{col}}\big((r2)\big) &= r, \\
&\vdots \\
\beta_{\text{col}}\big((32)\big) &= r-1+r-2 = 2r-3, \\
&\vdots \\
&\vdots \\
\beta_{\text{col}}\big((r, r-2)\big) &= \binom{r}{2} - 2, \\
\beta_{\text{col}}\big((r-1, r-2)\big) &= \binom{r}{2} - 1, \\
\beta_{\text{col}}\big((r, r-1)\big) &= \binom{r}{2},
\end{aligned}$$

clarified by the following diagram

▨	▨	▨	▨	▨	▨
r-1	▨	▨	▨	▨	▨
$\vdots$	$2r-3$	▨	▨	▨	▨
$\vdots$	$\vdots$	$\ddots$	▨	▨	▨
$\uparrow$	$\uparrow$	$\uparrow$	$\binom{r}{2}-1$	▨	▨
1	r	$\cdots$	$\binom{r}{2}-2$	$\binom{r}{2}$	▨

.

The second ordering is as

$$\beta_{\text{row}} : \mathcal{I} \to \left\{1, \dots, \binom{r}{2}\right\},$$

▨	▨	▨	▨	▨	▨
$\binom{r}{2}$	▨	▨	▨	▨	▨
$\binom{r}{2}-2$	$\binom{r}{2}-1$	▨	▨	▨	▨
$\vdots$	$\rightarrow$	$\ddots$	▨	▨	▨
r	$\rightarrow$	$\cdots$	$2r-3$	▨	▨
1	$\rightarrow$	$\cdots$	$\cdots$	$r-1$	▨

.

Obviously, the two orderings are mapped into each other by just transposing the diagrams with respect to the antidiagonal.

3.2.3 Local Convergence Analysis

The next result shows that our algorithm is locally a smooth map.

Theorem 3.1. *Algorithm 3.2*

$$s : M_{\mathcal{L}_n} \to M_{\mathcal{L}_n} \tag{3.38}$$

is a smooth mapping locally around A.

Proof. The algorithm is a composition of partial algorithmic steps

$$r_i : M_{\mathcal{L}_n} \to M_{\mathcal{L}_n}, \tag{3.39}$$

with $r_i(A) = A$ for all i. It therefore suffices to show smoothness for each r_i around the fixed point A. Typically, for one partial iteration step one has to compute the subblock p of the unipotent lower block triangular matrix

$$L = \begin{bmatrix} I & 0 \\ p & I \end{bmatrix}$$

fulfilling the equality

$$LXL^{-1} = \begin{bmatrix} I & 0 \\ p & I \end{bmatrix} \cdot \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ -p & I \end{bmatrix} = \begin{bmatrix} * & * \\ -pX_{12}p & * \end{bmatrix}$$

i.e., p has to solve the Sylvester equation

$$pX_{11} + X_{21} - X_{22}p = 0.$$

By assumption on the spectra of X_{11} and X_{22}, respectively, the solution of this Sylvester equation exists and is unique. Moreover, applying the Implicit Function Theorem to the function

$$\begin{aligned} &(X, p) \mapsto f(X, p), \\ &f(X, p) = pX_{11} + X_{21} - X_{22}p = 0 \end{aligned} \tag{3.40}$$

implies that $X \mapsto p(X)$ is smooth around A. Hence all partial iteration steps are smooth, the result follows. □

The above Theorem 3.1 justifies to use calculus for proving higher order convergence of our algorithm. We show next that the first derivative of our algorithm s at the fixed point A vanishes identically implying quadratic convergence if the chosen ordering is either β_{row} or β_{col}.

Theorem 3.2. *Algorithm 3.2 converges locally quadratically fast if as an ordering β_{row} or β_{col} is chosen.*

Proof. We will show that the first derivative $\mathrm{D}\, s(A)$ of the algorithm s at the fixed point A vanishes identically if β_{col} or β_{row} is chosen. By the chain rule we therefore have to compute the $\mathrm{D}\, r_{ij}(A)$ for all $i > j$ with $2 \leq i \leq l$ and $1 \leq j \leq m-1$. To be more precise, we have to study the effect of applying the linear map

$$\mathrm{D}\, r_{ij}(A) : T_A M_{\mathcal{L}_n} \to T_A M_{\mathcal{L}_n}$$

to those tangent vectors $[l, A] \in T_A M_{\mathcal{L}_n}$ onto which the "earlier" linear maps $\mathrm{D}\, r_{pq}(A)$ have been already applied to

$$\mathrm{D}\, s(A) \cdot [l, A] = \mathrm{D}\, r_{\text{last}}(A) \cdot \ldots \cdot \mathrm{D}\, r_{\text{first}}(A) \cdot [l, A], \quad l \in \mathfrak{l}_n.$$

Notice that A is not only a fixed point of s but also one of each individual r.

For simplicity but without loss of generality we may assume that the partitioning consists of 5 by 5 blocks. Typically, an $r_{ij}(X) = L_{ij} X L_{ij}^{-1}$ looks like

$$r_{ij}(X) = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & p_{ij} & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \cdot X \cdot \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & -p_{ij} & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}. \tag{3.41}$$

Therefore,

$$\mathrm{D}\, r_{ij}(A) \cdot [l, A] = \mathrm{D}(L_{ij} X L_{ij}^{-1}) \cdot [l, X]|_{X=A} = [L'_{ij}, A] + [l, A]$$

where

$$L'_{ij} := \mathrm{D}\, L_{ij}(A) \cdot [l, A],$$

and typically

$$L'_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p'_{ij} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with

$$p'_{ij} := \mathrm{D}\, p_{ij}(X) \cdot [l, X]|_{X=A}.$$

We already know that p_{ij} solves a Sylvester equation, namely

$$p_{ij}(X) X_{jj} + X_{ij} - X_{ii} p_{ij}(X) = 0, \tag{3.42}$$

with

$$p_{ij}(X)|_{X=A} = 0. \tag{3.43}$$

Taking the derivative of the Sylvester equation (3.42) acting on $[l, X]$ evaluated at $X = A$ gives

$$p'_{ij}(A) A_{jj} + [l, A]_{ij} - A_{ii} p'_{ij}(A) = 0. \tag{3.44}$$

An easy computation verifies that the commutator $[L'_{ij}, A]$ is of the following form

$$[L'_{ij}, A] = \begin{bmatrix} 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & p'_{ij} A_{jj} - A_{ii} p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

i.e., it differs from zero only by the (ij)-th block as well as by those blocks, which are to the right to, or which are above to this (ij)-th block. Therefore by (3.44), for the derivative of the (ij)-th partial step r_{ij} we get

$$\mathrm{D}\, r_{ij}(A) \cdot [l, A] = \underbrace{\begin{bmatrix} 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & p'_{ij} A_{jj} - A_{ii} p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{[L'_{ij}, A]} + \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & [l, A]_{ij} & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{[l, A]} .$$

That is, by (3.44) the first derivative annihilates the $(ij)-$th block, altering those blocks which are above or to the right to this $(ij)-$th block, but it leaves invariant all the other remaining blocks. Apparently, both ordering strategies now ensure, that after a whole iteration step all those blocks of the tangent vector $[l, A]$ lying below the main diagonal of blocks are eliminated. We therefore can conclude that

$$\mathrm{D}\, r_{ij}(A) \cdot [l, A] = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix} . \tag{3.45}$$

But we can even conclude more, namely

$$\mathrm{D}\, r_{ij}(A) \cdot [l, A] = 0. \tag{3.46}$$

This is easily proved following the argumentation in the proof of Lemma 3.2. Essentially, Assumption 3.1 ensures that the only Lie algebra element of $\mathfrak{l}_n$, which commutes with A into a block upper triangular matrix like A itself, is the zero matrix.

The result follows by the Taylor-type argument

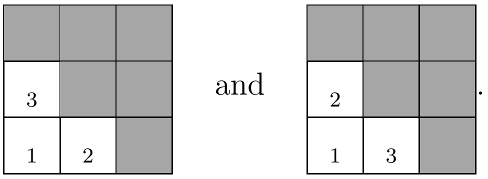
$$\|X_{k+1} - A\| \leq \sup_{Z \in \overline{U}} \| \mathrm{D}^2\, s(Z)\| \cdot \|X_k - A\|^2.$$

□

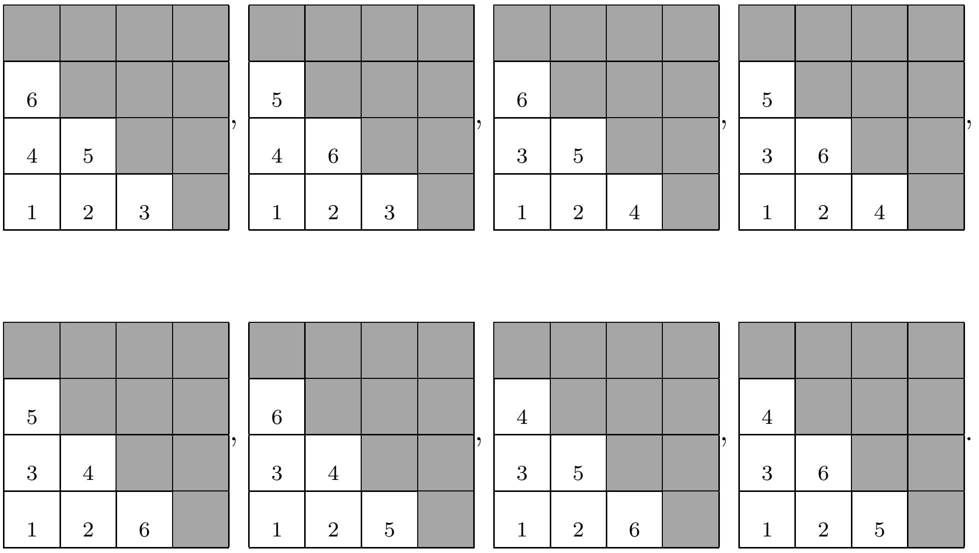
3.2.4 Further Insight to Orderings

Quite naturally one might ask if the two orderings β_{row} and β_{col} are the only possible ones ensuring quadratic convergence. The answer is no, because somehow "mixtures" of both strategies will also suffice as we will demonstrate by a few low dimensional examples.

Example 3.1. For $r = 3$ there are two possible orderings ensuring quadratic convergence for Algorithm 3.2:



Example 3.2. For $r = 4$ there are eight possible orderings together with its "conjugate" counterparts (transposing with respect to the antidiagonal) ensuring quadratic convergence for Algorithm 3.2:



Remark 3.1. The possible orderings are related to Young tableaux, or to be more precise, to standard tableaux. See [Ful97] for the connections between geometry of flag manifolds, representation theory of $\mathcal{GL}_n$, and calculus of tableaux.

Consequently, as a corollary of Theorem 3.2 we get the following result.

Corollary 3.1. *Algorithm 3.2 is quadratic convergent if the ordering is specified by the following two rules. The integers* $1, \dots, \binom{r}{2}$ *to be filled in*

1. *are strictly increasing across each row,*

2. *are strictly increasing up each column.*

□

We did not comment yet on orderings which are definitely not leading to quadratic convergence. It seems to be a cumbersome combinatorial problem to decide weather some ordering which does not respect Corollary 3.1 is always bad. To answer this question one needs also information on the fixed point as we see now by some further examples.

For the following series of examples let the fixed point be given by

$$A := \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}. \tag{3.47}$$

The partitioning will be according to $n = r = 3$, i.e., the block sizes are always 1×1. Therefore, an arbitrary tangent element $\xi \in T_A M_{\mathcal{L}_3}$ is of the form

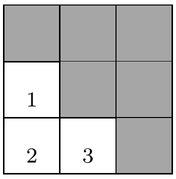
$$\xi = [l, A]$$

$$= \begin{bmatrix} -(a_{12}\,l_{21}) - a_{13}\,l_{31} & -(a_{13}\,l_{32}) & 0 \\ a_{11}\,l_{21} - a_{22}\,l_{21} - a_{23}\,l_{31} & a_{12}\,l_{21} - a_{23}\,l_{32} & a_{13}\,l_{21} \\ a_{11}\,l_{31} - a_{33}\,l_{31} & a_{12}\,l_{31} + a_{22}\,l_{32} - a_{33}\,l_{32} & a_{13}\,l_{31} + a_{23}\,l_{32} \end{bmatrix}$$

with

$$l := \begin{bmatrix} 0 & 0 & 0 \\ l_{21} & 0 & 0 \\ l_{31} & l_{32} & 0 \end{bmatrix}. \tag{3.48}$$

Example 3.3. Here we study explicitly the effect of the ordering



We get

$$\mathrm{D}\, r_1(A)\xi =$$

$$\begin{bmatrix} -\left(\frac{(a_{11}\, a_{13} - a_{13}\, a_{22} + a_{12}\, a_{23})\, l_{31}}{a_{11} - a_{22}}\right) & -\left(a_{13}\, l_{32}\right) & 0 \\ 0 & \frac{a_{23}\, (a_{12}\, l_{31} + (-a_{11} + a_{22})\, l_{32})}{a_{11} - a_{22}} & \frac{a_{13}\, a_{23}\, l_{31}}{a_{11} - a_{22}} \\ \left(a_{11} - a_{33}\right)\, l_{31} & a_{12}\, l_{31} + \left(a_{22} - a_{33}\right)\, l_{32} & a_{13}\, l_{31} + a_{23}\, l_{32} \end{bmatrix},$$

$$\mathrm{D}\, r_2(A)\, \mathrm{D}\, r_1(A)\xi = \begin{bmatrix} \frac{a_{12}\, a_{23}\, l_{31}}{-a_{11} + a_{22}} & -\left(a_{13}\, l_{32}\right) & 0 \\ a_{23}\, l_{31} & \frac{a_{23}\, (a_{12}\, l_{31} + (-a_{11} + a_{22})\, l_{32})}{a_{11} - a_{22}} & \frac{a_{13}\, a_{23}\, l_{31}}{a_{11} - a_{22}} \\ 0 & \left(a_{22} - a_{33}\right)\, l_{32} & a_{23}\, l_{32} \end{bmatrix},$$

$$\mathrm{D}\, r_3(A)\, \mathrm{D}\, r_2(A)\, \mathrm{D}\, r_1(A)\xi = \begin{bmatrix} \frac{a_{12}\, a_{23}\, l_{31}}{-a_{11} + a_{22}} & 0 & 0 \\ a_{23}\, l_{31} & \frac{a_{12}\, a_{23}\, l_{31}}{a_{11} - a_{22}} & \frac{a_{13}\, a_{23}\, l_{31}}{a_{11} - a_{22}} \\ 0 & 0 & 0 \end{bmatrix}$$

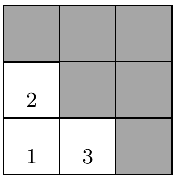
$$\neq 0.$$

But if one assumes that for the entry a_{23} of the fixed point A

$$a_{23} = 0 \tag{3.49}$$

holds, then even this ordering results in a quadratic convergent algorithm. □

Example 3.4. Here quadratic convergence is ensured according to Theorem 3.2:



We get

$$\mathrm{D}\, r_1(A)\xi = \begin{bmatrix} -\left(a_{12}\, l_{21}\right) & -\left(a_{13}\, l_{32}\right) & 0 \\ \left(a_{11} - a_{22}\right)\, l_{21} & a_{12}\, l_{21} - a_{23}\, l_{32} & a_{13}\, l_{21} \\ 0 & \left(a_{22} - a_{33}\right)\, l_{32} & a_{23}\, l_{32} \end{bmatrix},$$

$$\mathrm{D}\, r_2(A)\, \mathrm{D}\, r_1(A)\xi = \begin{bmatrix} 0 & -\left(a_{13}\, l_{32}\right) & 0 \\ 0 & -\left(a_{23}\, l_{32}\right) & 0 \\ 0 & \left(a_{22} - a_{33}\right)\, l_{32} & a_{23}\, l_{32} \end{bmatrix},$$

$$\mathrm{D}\, r_3(A)\, \mathrm{D}\, r_2(A)\, \mathrm{D}\, r_1(A)\xi = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

□

Example 3.5. This is another example which ensures quadratic convergence only under additional assumptions on the structure of the fixed point.

▨	▨	▨
1	▨	▨
3	2	▨

We get

$$\mathrm{D}\, r_1(A)\xi =$$

$$\begin{bmatrix} -\left(\frac{(a_{11}\, a_{13} - a_{13}\, a_{22} + a_{12}\, a_{23})\, l_{31}}{a_{11} - a_{22}}\right) & -\left(a_{13}\, l_{32}\right) & 0 \\ 0 & \frac{a_{23}\, (a_{12}\, l_{31} + (-a_{11} + a_{22})\, l_{32})}{a_{11} - a_{22}} & \frac{a_{13}\, a_{23}\, l_{31}}{a_{11} - a_{22}} \\ \left(a_{11} - a_{33}\right)\, l_{31} & a_{12}\, l_{31} + \left(a_{22} - a_{33}\right)\, l_{32} & a_{13}\, l_{31} + a_{23}\, l_{32} \end{bmatrix},$$

$$\mathrm{D}\, r_2(A)\, \mathrm{D}\, r_1(A)\xi =$$

$$\begin{bmatrix} -\left(\frac{(a_{11}\, a_{13} - a_{13}\, a_{22} + a_{12}\, a_{23})\, l_{31}}{a_{11} - a_{22}}\right) & \frac{a_{12}\, a_{13}\, l_{31}}{a_{22} - a_{33}} & 0 \\ 0 & \frac{a_{12}\, a_{23}\, (a_{11} - a_{33})\, l_{31}}{(a_{11} - a_{22})\, (a_{22} - a_{33})} & \frac{a_{13}\, a_{23}\, l_{31}}{a_{11} - a_{22}} \\ \left(a_{11} - a_{33}\right)\, l_{31} & 0 & \frac{(-(a_{12}\, a_{23}) + a_{13}\, (a_{22} - a_{33}))\, l_{31}}{a_{22} - a_{33}} \end{bmatrix},$$

$$\mathrm{D}\,r_3(A)\,\mathrm{D}\,r_2(A)\,\mathrm{D}\,r_1(A)\xi = \begin{bmatrix} \frac{a_{12}\,a_{23}\,l_{31}}{-a_{11}+a_{22}} & \frac{a_{12}\,a_{13}\,l_{31}}{a_{22}-a_{33}} & 0 \\ a_{23}\,l_{31} & \frac{a_{12}\,a_{23}\,(a_{11}-a_{33})\,l_{31}}{(a_{11}-a_{22})\,(a_{22}-a_{33})} & \frac{a_{13}\,a_{23}\,l_{31}}{a_{11}-a_{22}} \\ 0 & -\left(a_{12}\,l_{31}\right) & \frac{a_{12}\,a_{23}\,l_{31}}{-a_{22}+a_{33}} \end{bmatrix}$$

$$\neq 0.$$

But if one assumes that

$$a_{23} = a_{12} = 0 \tag{3.50}$$

holds, then even this ordering results in a quadratic convergent algorithm.

□

3.3 Orthogonal Transformations

For numerical reasons it makes more sense to use orthogonal transformations instead of unipotent lower triangular ones. We therefore reformulate Algorithm 3.2 in terms of orthogonal transformations. The convergence analysis for this new algorithm will greatly benefit from the calculations we already did.

For convenience we assume for a while that $r = 5$. Given

$$L = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & p & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix},$$

a quite natural idea is to use instead of L the orthogonal Q-factor from L after performing Gram-Schmidt, i.e., $L = RQ$, to the rows of subblocks of L. We have

$$R = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ & (I+p^\top p)^{-\frac{1}{2}} & 0 & p^\top (I+pp^\top)^{-\frac{1}{2}} & 0 \\ & & I & 0 & 0 \\ & & & (I+pp^\top)^{\frac{1}{2}} & 0 \\ & & & & I \end{bmatrix} \tag{3.51}$$

and

$$Q = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & (I+p^\top p)^{-\frac{1}{2}} & 0 & -(I+p^\top p)^{-\frac{1}{2}}p^\top & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & (I+pp^\top)^{-\frac{1}{2}}p & 0 & (I+pp^\top)^{-\frac{1}{2}} & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}. \tag{3.52}$$

Before we proceed to formulate the orthogonal version of Algorithm 3.2 we need some preliminaries. Namely we have to fix the manifold such an algorithm is “living” on.

Consider an “Iwasawa Decomposition” of the Lie group $\mathcal{L}_n$. The set of orthogonal matrices Q coming from an RQ-decomposition as in (3.51) do in general not generate an orthogonal group with group operation the ordinary matrix product. To see this we look at the simple 2 × 2-case

$$\begin{bmatrix} 1 & 0 \\ p & 1 \end{bmatrix} = \begin{bmatrix} (I+p^\top p)^{-\frac{1}{2}} & p^\top(I+pp^\top)^{-\frac{1}{2}} \\ 0 & (I+pp^\top)^{\frac{1}{2}} \end{bmatrix} \cdot \begin{bmatrix} (I+p^\top p)^{-\frac{1}{2}} & -(I+p^\top p)^{-\frac{1}{2}}p^\top \\ (I+pp^\top)^{-\frac{1}{2}}p & (I+pp^\top)^{-\frac{1}{2}} \end{bmatrix}. \tag{3.53}$$

Obviously, the set of orthogonal Q-matrices does not include

$$\widetilde{Q} := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{3.54}$$

Note that

$$\lim_{p\to\pm\infty} L \notin \mathcal{L}_2. \tag{3.55}$$

Nevertheless, we are able to construct atleast locally the space an orthogonal version of Algorithm 3.2 can be defined on. This construction will then allow us to use again analysis to prove quadratic convergence.

Consider an arbitrary element $L \in \mathcal{L}_n$ in a sufficiently small neighborhood $U_{\mathcal{L}_n}(I_n)$ of the identity I_n in $\mathcal{L}_n$, such that L can be parameterized by exponential coordinates of the second kind, cf. p.86, [Var84]. Let

$$\begin{aligned} L &= L_{\binom{r}{2}} \cdot \ldots \cdot L_1 \\ &= R_{\binom{r}{2}} Q_{\binom{r}{2}} \cdot \ldots \cdot R_1 Q_1. \end{aligned} \tag{3.56}$$

Here the L_i are defined as in (3.41). Each L_i, for $i = 1, \dots, \binom{r}{2}$, is represented as

$$L_i = \mathrm{e}^{l_i} \tag{3.57}$$

with, e.g., using β_{row} as an ordering,

$$l_1 = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & \ddots & \vdots \\ p_1 & 0 & \cdots & 0 \end{bmatrix}, l_2 = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & & & \ddots & \vdots \\ 0 & p_2 & 0 & \cdots & 0 \end{bmatrix}, \quad \text{etc.} \dots \tag{3.58}$$

We can therefore study the map

$$\begin{aligned} \sigma : \mathcal{L}_n \supset U_{\mathcal{L}_n}(I_n) &\to \mathcal{SO}_n, \\ L &\mapsto Q_{\binom{r}{2}}(L) \cdot \ldots \cdot Q_1(L). \end{aligned} \tag{3.59}$$

Note that

$$Q_i(I_n) = I_n \quad \text{for all} \quad i = 1, \dots, \binom{r}{2}$$

holds true. The following series of lemmata characterizes the mapping σ.

Lemma 3.3. *The mapping σ defined by (3.59) is smooth.*

Proof. See the explicit form of the Q_i given as in (3.51). □

Lemma 3.4. *The mapping σ defined by (3.59) is an immersion at I_n.*

Proof. We have to show that the derivative

$$\mathrm{D}\,\sigma(I_n) : \mathfrak{l}_n \to \mathfrak{so}_n$$

is injective.

For arbitrary $l = \sum_{i=1}^{\binom{r}{2}} l_i \in \mathfrak{l}_n$ the following holds true

$$\begin{aligned} \mathrm{D}\,\sigma(I_n) \cdot l &= \sum_{i=1}^{\binom{r}{2}} \mathrm{D}\,Q_i(I_n) \cdot l_i \\ &= \sum_{i=1}^{\binom{r}{2}} (l_i - l_i^\top) \\ &= l - l^\top, \end{aligned} \tag{3.60}$$

where we have used

$$\frac{\mathrm{d}}{\mathrm{d}\,\varepsilon}(I + \varepsilon^2 p^\top p)^{-\frac{1}{2}}\bigg|_{\varepsilon=0} = 0$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\,\varepsilon}(I + \varepsilon^2 p p^\top)^{-\frac{1}{2}}\bigg|_{\varepsilon=0} = 0.$$

Equation (3.60) implies injectivity in an obvious manner. □

Now we can apply the Immersion Theorem, cf. [AMR88] p.199.

Lemma 3.5. *The mapping σ as defined by (3.59) is a diffeomorphism of $U_{\mathcal{L}_n}(I_n)$ onto the image $\sigma(U_{\mathcal{L}_n}(I_n))$.* □

Consider the isospectral manifold

$$M_{\mathcal{SO}_n} := \{X \in \mathbb{R}^{n\times n} \,|\, X = QAQ^\top, Q \in \mathcal{SO}_n\} \tag{3.61}$$

with A as in (3.5) fulfilling Assumption 3.1. Define

$$\begin{aligned} \alpha : \sigma(U_{\mathcal{L}_n}(I_n)) &\to M_{\mathcal{SO}_n}, \\ Q &\mapsto QAQ^\top. \end{aligned} \tag{3.62}$$

Lemma 3.6. *The mapping α defined as in (3.62) is smooth.*

Proof. The result follows by the explicit construction of an arbitrary Q by using exponential coordinates of the second kind. □

Lemma 3.7. *The mapping α defined as in (3.62) is an immersion at I_n.*

Proof. We have to show that the derivative

$$\mathrm{D}\,\alpha(I_n) : T_{I_n}\sigma(U_{\mathcal{L}_n}(I_n)) \to T_A M_{\mathcal{SO}_n}$$

is injective. Arbitrary elements of the tangent space $T_{I_n}\sigma(U_{\mathcal{L}_n}(I_n))$ have the form

$$\sum_{i=1}^{\binom{r}{2}} (l_i - l_i^\top) = l - l^\top,$$

whereas those of the tangent space $T_A M_{\mathcal{SO}_n}$ look like

$$[l - l^\top, A].$$

To show injectivity of

$$\begin{aligned} \mathrm{D}\,\alpha(I_n) : T_{I_n}\sigma(U_{\mathcal{L}_n}(I_n)) &\to T_A M_{\mathcal{SO}_n}, \\ l - l^\top &\mapsto [l - l^\top, A], \end{aligned}$$

we partition $l - l^\top$ accordingly to A, i.e.,

$$A = \begin{bmatrix} A_{11} & \cdots & A_{rr} \\ & \ddots & \vdots \\ & & A_{rr} \end{bmatrix}, \quad l - l^\top = \begin{bmatrix} 0 & -p_{21}^\top & \cdots & -p_{r1}^\top \\ p_{21} & \ddots & & \vdots \\ \vdots & & \ddots & -p_{r,r-1}^\top \\ p_{r1} & \cdots & p_{r,r-1} & 0 \end{bmatrix}.$$

Note that

$$[l - l^\top, A]_{r1} = p_{r1}A_{11} - A_{rr}p_{r1}.$$

Assume the converse, i.e.,

$$[l - l^\top, A] = [\widetilde{l} - \widetilde{l}^\top, A] \tag{3.63}$$

holds for some $\widetilde{l} \neq l$ with

$$\widetilde{l} := \begin{bmatrix} 0 & & & \\ \widetilde{p}_{21} & \ddots & & \\ \vdots & & \ddots & \\ \widetilde{p}_{r1} & \cdots & \widetilde{p}_{r,r-1} & 0 \end{bmatrix} \in \mathfrak{l}_n.$$

Looking at the $(r1)$-block of (3.63) implies

$$(p_{r1} - \widetilde{p}_{r1})A_{11} - A_{rr}(p_{r1} - \widetilde{p}_{r1}) = 0. \tag{3.64}$$

By Assumption 3.1 on the spectra of A_{11} and A_{rr}, respectively, (3.64) implies

$$p_{r1} = \widetilde{p}_{r1}.$$

Now by induction on the subdiagonals of blocks, i.e., going from the lower left corner block of (3.63) to the first subdiagonal of blocks, and continuing to apply recursively the same arguments on the $(r-1,1)$-block of (3.63), as well as on the $(r2)$-block of (3.63), then imply

$$p_{r2} = \widetilde{p}_{r2} \quad \text{and} \quad p_{r-1,1} = \widetilde{p}_{r-1,1}.$$

Finally, we get

$$[l - l^\top, A] = [\widetilde{l} - \widetilde{l}^\top, A] \quad \Longrightarrow \quad l = l^\top,$$

a contradiction. Therefore, $\mathrm{D}\,\alpha(I_n)$ is injective, hence α is an immersion at I_n. □

Consequently, we have

Lemma 3.8. *The composition mapping $\alpha \circ \sigma : U_{\mathcal{L}_n}(I_n) \to M_{\mathcal{SO}_n}$ is a diffeomorphism of $U_{\mathcal{L}_n}(I_n)$ onto the image $(\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n))$.*

□

3.3.1 The Algorithm

The following algorithm will be analyzed. Given an $X \in (\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n))$ and let A fulfil Assumption 3.1. For convenience we abbreviate in the sequel

$$M := (\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n)). \tag{3.65}$$

Consider the index set

$$\mathcal{I} := \{(ij)\}_{i=2,\dots,r;j=1,\dots,r-1} \tag{3.66}$$

and fix an ordering β. For convenience we again rename double indices in the description of the algorithm by simple ones by means of $X_{ij} \mapsto X_{\beta(ij)}$ respecting the ordering β.

Algorithm 3.3 (Orthogonal Sylvester Sweep).

Given an $X \in (\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n)) = M$. Define

$$\begin{aligned} X_k^{(1)} &:= Q_1 X Q_1^\top \\ X_k^{(2)} &:= Q_2 X_k^{(1)} Q_2^\top \\ &\vdots \\ X_k^{\binom{r}{2}} &:= Q_{\binom{r}{2}} X_k^{\binom{r}{2}-1} Q_{\binom{r}{2}}^\top \end{aligned}$$

where for $l = 1, \ldots, \binom{r}{2}$ the transformation matrix $Q_l \in \mathcal{SO}_n$ differs from the identity matrix I_n only by 4 subblocks. Namely, the

$$\begin{aligned} jj - \text{th block is equal to} &\quad (I + p^\top p)^{-\frac{1}{2}} \\ ji - \text{th block is equal to} &\quad -(I + p^\top p)^{-\frac{1}{2}} p^\top \\ ij - \text{th block is equal to} &\quad (I + pp^\top)^{-\frac{1}{2}} p \\ ii - \text{th block is equal to} &\quad (I + pp^\top)^{-\frac{1}{2}}. \end{aligned}$$

Here $p_l \in \mathbb{R}^{n_j \times n_i}$, $\beta((ij)) = l$, and p_l solves the Sylvester equation

$$p_l \left(X_k^{(l-1)} \right)_{jj} - \left(X_k^{(l-1)} \right)_{ii} p_l + \left(X_k^{(l-1)} \right)_{ij} = 0.$$

The overall algorithm then consists of the iteration of orthogonal sweeps.

Algorithm 3.4 (Orthogonal Refinement of Estimates of Subspaces).

- Let $X_0, \ldots, X_k \in M$ be given for $k \in \mathbb{N}_0$.
- Define the recursive sequence $X_k^{(1)}, \ldots, X_k^{\binom{r}{2}}$ as above (sweep).
- Set $X_{k+1} := X_k^{\binom{r}{2}}$. Proceed with the next sweep.

3.3.2 Local Convergence Analysis

Analogously to Theorem 3.1 we have

Theorem 3.3. *Algorithm 3.4*

$$s : M \to M \tag{3.67}$$

is a smooth mapping locally around A.

Proof. The algorithm is a composition of partial algorithmic steps r_i. Smoothness of these partial algorithmic steps follows from the smoothness of each p_i already shown. □

Theorem 3.4. *Algorithm 3.4 converges locally quadratically fast if for working off the partial algorithmic steps an ordering is chosen which respects Corollary 3.1.*

Proof. We will compute $\mathrm{D}\, r_{ij}(A)$ for all $i > j$ with $2 \leq i \leq l$ and $1 \leq j \leq m-1$. Without loss of generality we may assume that the partitioning consists of 5 by 5 blocks. Typically, a transformation matrix Q_{ij} for $r_{ij}(X) = Q_{ij}XQ_{ij}^\top$ looks like

$$Q_{ij}(X) = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & S_{ij}(X) & 0 & -S_{ij}(X)p_{ij}^\top(X) & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & T_{ij}(X)p_{ij}(X) & 0 & T_{ij}(X) & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}, \tag{3.68}$$

where

$$S_{ij}(X) = S_{ij}^\top(X) := \Big(I + p^\top(X)p(X)\Big)^{-\frac{1}{2}}$$

and

$$T_{ij}(X) = T_{ij}^\top(X) := \Big(I + p(X)p^\top(X)\Big)^{-\frac{1}{2}}.$$

Moreover,

$$S_{ij}(A) = I_{n_i}$$

and

$$T_{ij}(A) = I_{n_j}.$$

An arbitrary

$$\Omega \in \mathfrak{so}_n/(\mathfrak{so}_{n_1} \oplus \ldots \oplus \mathfrak{so}_{n_r})$$

looks like

$$\Omega = \begin{bmatrix} 0 & -\Omega_{21}^\top & \cdots & -\Omega_{r1}^\top \\ \Omega_{21} & \ddots & & \vdots \\ \vdots & & \ddots & -\Omega_{r,r-1}^\top \\ \Omega_{r1} & \cdots & \Omega_{r,r-1} & 0 \end{bmatrix}.$$

The derivative of one partial algorithmic step acting on $[\Omega, A] \in T_A M$ is as

$$\mathrm{D}\, r_{ij}(A) \cdot [\Omega, A] = [Q'_{ij}, A] + [\Omega, A],$$

where

$$Q'_{ij} := \mathrm{D}\, Q_{ij}(A) \cdot [\Omega, A],$$

and typically

$$Q'_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & S'_{ij}(A) & 0 & -(p_{ij}^\top)'(A) & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p'_{ij}(A) & 0 & T'_{ij}(A) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with

$$p'_{ij}(A) := \mathrm{D}\, p_{ij}(X) \cdot [\Omega, X]|_{X=A}.$$

We already know that p_{ij} solves a Sylvester equation, namely

$$p_{ij}(X)X_{jj} + X_{ij} - X_{ii}p_{ij}(X) = 0, \tag{3.69}$$

with

$$p_{ij}(X)|_{X=A} = 0. \tag{3.70}$$

Taking the derivative of the Sylvester equation (3.69) acting on $[\Omega, A]$ gives

$$p'_{ij}(A)A_{jj} + [\Omega, A]_{ij} - A_{ii}p'_{ij}(A) = 0. \tag{3.71}$$

An easy computation verifies that the commutator $[Q'_{ij}, A]$ is of the following form

$$[Q'_{ij}, A] = \begin{bmatrix} 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & p'_{ij}A_{jj} - A_{ii}p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

i.e., the (ij)-th block equals $p'_{ij}A_{jj} - A_{ii}p'_{ij}$ and columns of blocks to the left as well as rows of blocks below are zero. Therefore by (3.71), for the derivative of the (ij)-th partial step r_{ij} we get

$$\mathrm{D}\, r_{ij}(A) \cdot [\Omega, A] = \underbrace{\begin{bmatrix} 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & p'_{ij}A_{jj} - A_{ii}p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{[Q'_{ij}, A]} + \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & [\Omega, A]_{ij} & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{[\Omega, A]}.$$

That is, by (3.71) the first derivative annihilates the $(ij)-$th block, altering eventually those blocks which are above, to the right, or a combination of both, to this $(ij)-$th block, but it leaves invariant all the other remaining blocks. Apparently, all ordering strategies respecting Corollary 3.1 ensure, that after a whole iteration step all those blocks lying below the main diagonal of blocks are eliminated. We therefore can conclude that

$$\mathrm{D}\, r_{ij}(A) \cdot [\Omega, A] = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}. \tag{3.72}$$

Again we can even conclude more, namely

$$\mathrm{D}\, r_{ij}(A) \cdot [\Omega, A] = 0. \tag{3.73}$$

Following the argumentation in the proof of Lemma 3.2, essentially, Assumption 3.1 ensures that the only element of $\mathfrak{so}_n/(\mathfrak{so}_{n_1} \oplus \ldots \oplus \mathfrak{so}_{n_r})$, which commutes with A into a block upper triangular matrix, is the zero matrix. This can also be seen from the fact that the above Ω equals an $l - l^\top$ where $l \in \mathfrak{l}_n$.

The result follows by the Taylor-type argument

$$\|X_{k+1} - A\| \leq \sup_{Z \in \overline{U}} \| \mathrm{D}^2\, s(Z)\| \cdot \|X_k - A\|^2.$$

□

3.3.3 Discussion and Outlook

Consider a nearly upper triangular matrix over $\mathbb{C}$ with distinct eigenvalues. Assume $n = r$, i.e., we have to solve $\binom{n}{2}$ *scalar* Sylvester equations per sweep. Our algorithm leads then to an extremely efficient algorithm for refining estimates of eigenvectors. Each partial algorithmic step requires just the solution of a scalar linear equation.

We would like to mention that the methods from this chapter can also be applied to generalized eigenproblems in a completely straight forward way. Instead of one Riccati or one Sylvester equation one has to solve a system of two coupled ones. Everything works fine under a reasonable assumption on the spectra of subblocks.

It is a challenge to apply our methods also to more structured generalized eigenvalue problems, say Hamiltonian ones.

If the matrix for which we would like to compute invariant subspaces is symmetric, our method is related to [Göt95]. There, socalled approximate Givens (or Jacobi) transformations are developed which essentially approximate an exact rotation to zero out a matrix entry. Such an approach has advantages if one is interested in VLSI-implementations.

Nevertheless, it is an open problem if our algorithm has a reinterpretation as a Jacobi-type method in the general nonsymmetric case, i.e., if there is a cost function which is minimized in each step.

Chapter 4

Rayleigh Quotient Iteration, QR-Algorithm, and Some Generalizations

A wellknown algorithm for computing a single eigenvector-eigenvalue pair of a real symmetric matrix is the Rayleigh Quotient Iteration. It was initially used to improve an approximate eigenvector, see [Par80] and references therein. Local cubic convergence was firstly shown in a series of papers by Ostrowski, cf. [Ost59].

The QR-algorithm for the symmetric eigenvalue problem is known to be closely related to RQI, see e.g., p. 441 in [GvL89]. The QR-algorithm is known to be one of the most efficient algorithms. The reason for this is mainly that one can exploit a banded structure of the matrices under consideration and furthermore one is able to bring a given matrix in a finite number of steps to such a banded form. Nevertheless, from our point of view the convergence analysis of the QR-algorithm is far from being easy to understand. Moreover, the fact that in the symmetric tridiagonal case QR using Rayleigh Quotient shifts or socalled Wilkinson shifts converges locally cubically is somewhat misleading because it is not the algorithm itself which is converging fast, merely it is a submatrix or some entry which converges cubically in norm. Essentially, deflating then is necessary to make the algorithm efficient.

In this chapter we will start showing cubic convergence of the classical Rayleigh Quotient iteration by means of Calculus. Then we will develop a new algorithm which we call parallel RQI, because its relation to the RQI is closer than the relation between QR-algorithm and RQI. Essentially, parallel RQI is an algorithm which is under some mild assumptions locally well defined, moreover, ulimately it converges in a way that all eigenvalue-eigenvector pairs converge *simultaneously* cubically.

In the last section we take a closer look to the local convergence properties of the shifted QR-algorithm when applied to a real symmetric matrix. We will show that there exists no smooth shift strategy which ensures that the algorithm itself, considered as a selfmap on the orthogonal group $\mathcal{O}_n$, converges quadratically.

4.1 Local Cubic Convergence of RQI

Given a nonsingular $A = A^\top \in \mathbb{R}^{n\times n}$ with distinct eigenvalues the iteration

$$x_{k+1} = \frac{(A - x_k^\top A x_k I_n)^{-1} x_k}{\|(A - x_k^\top A x_k I_n)^{-1} x_k\|} \tag{4.1}$$

is known to be locally cubically convergent around each eigenvector of A, cf. [Ost59, Par80]. Usually, one proves cubic convergence by using tricky estimates. The differentiability properties of the map

$$x \mapsto \frac{(A - x^\top A x I_n)^{-1} x}{\|(A - x^\top A x I_n)^{-1} x\|} \tag{4.2}$$

are not exploited for the proof. The main reason for this lack might be that the map

$$\begin{aligned} \tilde{f} : S^{n-1} &\to S^{n-1} \\ x &\mapsto \frac{(A - x^\top A x I_n)^{-1} x}{\|(A - x^\top A x I_n)^{-1} x\|} \end{aligned} \tag{4.3}$$

has discontinuities at the fixed points of the corresponding dynamical system, namely at the normalized eigenvectors of A.

By a rather simple idea we remove this discontinuities by defining another iteration, which we call again Rayleigh Quotient Iteration. Consider the map

$$\begin{aligned} f : S^{n-1} &\to S^{n-1} \\ x &\mapsto \frac{\operatorname{adj}(A - x^\top A x I_n) x}{\|\operatorname{adj}(A - x^\top A x I_n) x\|}. \end{aligned} \tag{4.4}$$

Iterating (4.4) obviously has roughly the same dynamics as iterating (4.3). The difference is just in the sign of the determinant of $(A - x^\top AxI_n)$. The big advantage of looking at (4.4) instead of (4.3) is that both have the same fixed points but (4.4) is smooth around the eigenvectors.

Theorem 4.1. *Let $A = A^\top$ be nonsingular having distinct eigenvalues. The mapping f defined by (4.4) is smooth around any eigenvector of A.*

Proof. Without loss of generality we may assume that A is diagonal, i.e.,

$$A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

The denominator in (4.4), namely

$$\|\operatorname{adj}(A - x^\top AxI_n)x\|$$

is equal to zero, if and only if x lies in the kernel of $\operatorname{adj}(A - x^\top AxI_n)$. If x is an eigenvector, namely $x = e_i$, with e_i a standard basis vector of $\mathbb{R}^n$, it follows that $x^\top Ax$ is the corresponding eigenvalue and therefore $A - x^\top AxI_n$ is singular.

Nevertheless,

$$\begin{aligned} \operatorname{adj}(A - \lambda_i I_n) \cdot e_i &= \prod_{j \neq i} (\lambda_j - \lambda_i) e_i \\ &\neq 0 \end{aligned}$$

holds true. For $x \neq e_i$ lying in a sufficiently small neighborhood of e_i the "Rayleigh Quotient" $x^\top Ax$ is never an eigenvalue. The result follows. □

By the next result we show that RQI is cubically convergent. For this we will use the differentiability properties of (4.4).

Theorem 4.2. *Let $A = A^\top$ be nonsingular having distinct eigenvalues. At any eigenvector $x \in S^{n-1}$ of A, the first and second derivatives of f defined by (4.4) vanish.*

Proof. Again without loss of generality we assume A to be diagonal. Define

$$\begin{aligned} F &: S^{n-1} \to S^{n-1} \\ F(x) &= \operatorname{adj}(A - x^\top AxI_n)x, \end{aligned} \tag{4.5}$$

and therefore

$$f(x) = \frac{F(x)}{\|F(x)\|}. \tag{4.6}$$

Furthermore, define

$$\begin{aligned} G &: \mathbb{R}^n \to \mathbb{R}^n \\ G(x) &= \operatorname{adj}(A - x^\top A x I_n)x, \end{aligned} \tag{4.7}$$

and

$$g(x) = \frac{G(x)}{\|G(x)\|}. \tag{4.8}$$

That is

$$F = G|_{S^{n-1}}, \quad f = g|_{S^{n-1}}.$$

Now for real $\alpha \neq 0$

$$\mathrm{D}\, g(x)\xi|_{x=\alpha e_i} = \Big(\operatorname{id} - g(\alpha e_i) g^\top(\alpha e_i)\Big) \frac{\mathrm{D}\, G(\alpha e_i)\xi}{\|G(\alpha e_i)\|} \tag{4.9}$$

where

$$\begin{aligned} g(\alpha e_i) &= \frac{\prod\limits_{j\neq i}(\lambda_j - \lambda_i)\alpha e_i}{\|\prod\limits_{j\neq i}(\lambda_j - \lambda_i)\alpha e_i\|} \\ &= e_i \operatorname{sign}(\alpha \prod_{j\neq i}(\lambda_j - \lambda_i)). \end{aligned} \tag{4.10}$$

Therefore,

$$\operatorname{id} - g(\alpha e_i) g^\top(\alpha e_i) = \operatorname{id} - e_i e_i^\top. \tag{4.11}$$

Moreover,

$$\begin{aligned} \mathrm{D}\, G(x)\xi|_{x=\alpha e_i} &= \mathrm{D}(\operatorname{adj}(A - x^\top A x I_n)x)\xi|_{x=\alpha e_i} \\ &= \operatorname{adj}(A - x^\top A x I_n)\xi|_{x=\alpha e_i} + (\mathrm{D}(\operatorname{adj}(A - x^\top A x I_n))\xi)x|_{x=\alpha e_i}. \end{aligned} \tag{4.12}$$

The first summand on the right hand side of the last line of (4.12) gives

$$\begin{aligned} \operatorname{adj}(A - x^\top A x I_n)\xi|_{x=\alpha e_i} &= \prod_{j\neq i}(\lambda_j - \lambda_i\alpha^2)\xi_i e_i \\ &= K_1 e_i \end{aligned} \tag{4.13}$$

with constant $K_1 \in \mathbb{R}$.

The second summand of (4.12) is as

$$\begin{aligned} (\mathrm{D}(\operatorname{adj}(A - x^\top A x I_n))\xi)x|_{x=\alpha e_i} &= \mathrm{D}\begin{bmatrix} \prod\limits_{j\neq 1}(\lambda_j - x^\top A x) & & \\ & \ddots & \\ & & \prod\limits_{j\neq n}(\lambda_j - x^\top A x) \end{bmatrix}\xi\Bigg|_{x=\alpha e_i} \alpha e_i \\ &= \mathrm{D}\prod_{j\neq i}(\lambda_j - x^\top A x)\xi|_{x=\alpha e_i}\alpha e_i \\ &= K_2 e_i \end{aligned} \tag{4.14}$$

with constant $K_2 \in \mathbb{R}$. Hence,

$$\begin{aligned} \mathrm{D}\, g(x)\xi|_{x=\alpha e_i} &= \frac{\mathrm{id} - e_i e_i^\top}{\|G(\alpha e_i)\|}(K_1 + K_2)e_i \\ &= 0. \end{aligned} \tag{4.15}$$

That is, we have the implication

$$\mathrm{D}\, g(e_i) = 0 \quad \Longrightarrow \quad \mathrm{D}\, f(e_i) = 0. \tag{4.16}$$

Now we compute the second derivative. For $h \in \mathbb{R}^n$ we have

$$\begin{aligned} \mathrm{D}^2\, g(x)(h,h)|_{x=\alpha e_i} = \mathrm{D}\,\frac{\mathrm{id} - g(x)g^\top(x)}{\|G(x)\|}h \cdot \mathrm{D}\, G(x)\cdot h|_{x=\alpha e_i} + \\ + \frac{\mathrm{id} - g(x)g^\top(x)}{\|G(x)\|}\cdot \mathrm{D}^2\, G(x)\cdot (h,h)|_{x=\alpha e_i}. \end{aligned} \tag{4.17}$$

We claim that the first summand in (4.17) is zero. By a tedious computation one gets that

$$\mathrm{D}\,\frac{\mathrm{id}-g(x)g^\top(x)}{\|G(x)\|}h|_{x=\alpha e_i} = \mathrm{const}\cdot(\mathrm{id}-e_ie_i^\top). \tag{4.18}$$

But we already know that

$$\mathrm{D}\,G(e_i)\cdot h = \mathrm{const}\cdot e_i, \tag{4.19}$$

therefore the claim is true.

The Hessian of $f = g|_{S^{n-1}}$ at e_i as a symmetric bilinear form on $T_{e_i}S^{n-1}\times T_{e_i}S^{n-1}$ can now be defined as

$$\mathrm{D}^2\,f(e_i)(\mu,\mu) = \frac{\mathrm{id}-e_ie_i^\top}{\|F(e_i)\|}\cdot\mathrm{D}^2\,G(e_i)(\mu,\mu). \tag{4.20}$$

It will turn out from the following calculation that

$$\mathrm{D}^2\,f(e_i)(\mu,\mu) = 0. \tag{4.21}$$

For this let us first evaluate for $h\in\mathbb{R}^n$ the second derivative of the extended function $G:\mathbb{R}^n\to\mathbb{R}^n$ defined by (4.7). A lengthy computation gives

$$\begin{aligned}\mathrm{D}^2\,G(x)(h,h)|_{x=\alpha e_i} &= 2(\mathrm{D}\,\mathrm{adj}(A-x^\top AxI_n)\cdot h)\cdot h|_{x=\alpha e_i}+\\ &\quad+\mathrm{D}^2\,\mathrm{adj}(A-x^\top AxI_n)(h,h)\cdot x|_{x=\alpha e_i}.\end{aligned} \tag{4.22}$$

Note that the last summand in (4.22) lies in the kernel of $\mathrm{id}-e_ie_i^\top$, therefore

$$\begin{aligned}\mathrm{D}^2\,g(x)(h,h)|_{x=e_i} &= 2\frac{\mathrm{id}-e_ie_i^\top}{\|G(e_i)\|}\cdot(\mathrm{D}\,\mathrm{adj}(A-x^\top AxI_n)\cdot h)\cdot h|_{x=e_i}\\ &= -4e_i^\top h\frac{\mathrm{id}-e_ie_i^\top}{\|G(e_i)\|}\prod_{j\neq i}(\lambda_j-\lambda_i)\begin{bmatrix}\sum\limits_{k\neq 1}\frac{1}{\lambda_k-\lambda_i} & & \\ & \ddots & \\ & & \sum\limits_{k\neq n}\frac{1}{\lambda_k-\lambda_i}\end{bmatrix}\cdot h.\end{aligned} \tag{4.23}$$

For all $\mu\in T_{e_i}S^{n-1}$ it holds

$$\mathrm{D}^2\,g(e_i)(\mu,\mu) = 0$$

because on the n-sphere $\mathrm{e}_i^\top \mu = 0$. The theorem is proved now. □

Equation (4.23) is interesting for the following reasons. Firstly, it shows cubic convergence for the RQI considered as a dynamical system on the sphere. Secondly, for arbitrary vectors $h \notin T_{e_i}S^{n-1}$ the second order derivative of g is not equal to zero.

As a consequence we can state that RQI considered as a dynamical system on $\mathbb{R}^n$ is only quadratically convergent. An even more tedious calculation shows that a third variant

$$x_{k+1} = \frac{\operatorname{adj}\left(A - \frac{x_k^\top A x_k}{x_k^\top x_k}\right) x_k}{\left\|\operatorname{adj}\left(A - \frac{x_k^\top A x_k}{x_k^\top x_k}\right) x_k\right\|} \tag{4.24}$$

is again cubically convergent. These subtleties may have consequences to the numerical implementation. For RQI considered as an iteration on $\mathbb{R}^n$ nonspherical second order perturbations near an equilibrium point may disturb cubic convergence. Whereas for the iteration (4.24) they will not. All these considerations may convince the programmer how important correct normalization might be.

4.2 Parallel Rayleigh Quotient Iteration or Matrix-valued Shifted QR-Algorithms

A quite natural question one may raise is, if one is able to formulate a QR-type algorithm which is somehow the true generalization of RQI to a full matrix. By this we mean an algorithm which ultimately does RQI on each column individually and even simultaneously. The idea is as follows.

Algorithm 4.1 (Parallel Rayleigh Quotient Iteration on $\mathrm{St}_{n,k}$). Given a nonsingular $A = A^\top \in \mathbb{R}^{n\times n}$ with distinct eigenvalues and an $X \in \mathrm{St}_{n,k}$.

1. Solve for $Z \in \mathbb{R}^{n\times k}$
$$AZ - Z\,\mathrm{Diag}(X^\top AX) = X$$
where $\mathrm{Diag}(A) := \mathrm{diag}(a_{11}, \dots, a_{nn})$ denotes the diagonal part of A.
2. Set $X = (Z)_Q$, where $(Z)_Q$ denotes the Q-factor from a QR-decomposition of Z, with $Q \in \mathrm{St}_{n,k}$ and go to 1.

Obviously, for $k = 1$ this iteration is RQI. For $k = n$ one can interpret this algorithm as a QR-type algorithm on the orthogonal group $\mathcal{O}_n$ performing *matrix valued shifts*, i.e., each column of Z is differently shifted. For $n > k \geq 1$ this algorithm is closely related to the recent work, [AMSD02]. One of the main differences between Algorithm 4.1 and the iteration presented in [AMSD02] is that we just take the orthogonal projection on the diagonal, see step 1. in Algorithm 4.1, whereas Absil *et al.* need a diagonalization instead. Moreover, we are able to show that our iteration is well defined even in the case $n = k$.

Let us analyze the local convergence properties of parallel RQI. Firstly, we want to get rid of the discontinuities at the fixed points. It is easily seen that the fixed points of the parallel RQI are those $X \in \mathrm{St}_{n,k}$ where $X^\top AX$ is diagonal, or equivalently, those points X, the colunmns of which are eigenvectors of A. We will use the same idea as for standard RQI, see above.

Let $X = [x_1, \dots, x_k]$, and $Z = [z_1, \dots, z_k]$. If the $i-$th diagonal entry of $X^\top AX$ is not equal to an eigenvalue of A, the shifted matrix $A-(X^\top AX)_{ii}I_n$ is invertible and therefore

$$AZ - Z\,\mathrm{Diag}(X^\top AX) = X \quad\iff\quad z_i = (A - (X^\top AX)_{ii}I_n)^{-1}x_i \tag{4.25}$$

for all $i = 1, \dots, k$. For our analysis we will therefore use for all i

$$\boxed{z_i = \operatorname{adj}(A - (X^\top AX)_{ii} I_n) x_i. \qquad (4.26)}$$

Scaling a column z_i by the determinant $\det(A - (X^\top AX)_{ii} I_n)$ is not necessary because this can be incorporated into the triangular factor R of the QR-decomposition of Z in the second step of the algorithm.

As a consequence of the RQI analysis from the last section we have

Theorem 4.3. *Parallel RQI considered as an iteration on the compact Stiefel manifold* $\mathrm{St}_{n,k}$ *using (4.26)*

1. *is a well defined iteration in an open neighborhood of any fixed point* X*,*
2. *is smooth around such an* X*,*
3. *converges locally cubically fast to such an* X*.*

Proof. Without loss of generality we assume $A = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$. To see that the algorithm is locally well defined it is enough to prove that the matrix Z has full rank at a fixed point, but this is trivial because each fixed point itself is of full rank. Note that for any vector $x \in \mathbb{R}^n$ being sufficiently close to an eigenvector e_i of A the expression $(\operatorname{adj}(A - x^\top AxI_n))x$ never equals zero and therefore the second part of the theorem follows also. To prove the third part we need to compute the derivative of

$$Z(X) = Q(Z(X)) \cdot R(Z(X)). \tag{4.27}$$

Here

$$\begin{aligned} Z &: \mathrm{St}_{n,k} \to \mathbb{R}^{n \times k}, \\ Q &: \mathbb{R}^{n \times k} \to \mathrm{St}_{n,k}, \\ R &: \mathbb{R}^{n \times k} \to \mathcal{U}_k, \end{aligned} \tag{4.28}$$

where $\mathcal{U}_k$ denotes the Lie group of upper triangular $(k \times k)$-matrices having positive diagonal entries. The algorithm Parallel RQI can be described by the map

$$X \mapsto Q(Z(X)). \tag{4.29}$$

To prove higher order convergence we therefore need to compute

$$\mathrm{D}\, Q(Z(X_f)) : T_{X_f}\,\mathrm{St}_{n,k} \to T_{X_f}\,\mathrm{St}_{n,k} \tag{4.30}$$

which can be done by taking the derivative of (4.27). Using the chain rule and exploiting the fact that $Z(X_f) = Q(X_f) = X_f$ and $R(X_f) = I_k$ we get

$$\mathrm{D}\, Z(X_f)\xi = \big(\mathrm{D}\, Q(X_f)\cdot \mathrm{D}\, Z(X_f)\cdot \xi\big)R(X_f) + Q(X_f)\big(\mathrm{D}\, R(X_f)\cdot \mathrm{D}\, Z(X_f)\cdot \xi\big).$$

As a matter of fact

$$\mathrm{D}\, Z(X_f) = 0 \tag{4.31}$$

holds true because each "column" of $\mathrm{D}\, Z(X_f)\cdot\xi$ is equal to zero being just the derivative of an individual RQI on the corresponding column of X evaluated at X_f. Hence

$$\mathrm{D}\, Q(Z(X_f))\cdot \mathrm{D}\, Z(X_f) = 0. \tag{4.32}$$

Consequently it makes perfectly sense also to define second derivatives. The argumentation is the same and the details are therefore omitted. The result follows. □

4.2.1 Discussion

Each iteration step of parallel RQI requires a solution of a Sylvester equation. Problems will occur if these solutions will not have full rank. As a consequence the QR-decomposition in the second step of the algorithm would not be unique. Even worse, the iteration itself would not be well defined. The following counterexample shows that the property of being well defined does not globally hold.

Example 4.1. Consider the tridiagonal symmetric matrix

$$A = \begin{bmatrix} 1 & \sqrt{2} & 0 \\ \sqrt{2} & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{4.33}$$

with eigenvalues $\lambda_1 = -1$ and $\lambda_{2,3} = \frac{3}{2} \pm \sqrt{\frac{5}{4}}$. If one starts the parallel Rayleigh Quotient Iteration for $k = 3$ with the identity matrix I_3 the columns of the matrix Z are computed as

$$z_1 = \operatorname{adj}(A - a_{11}I_3)e_1 = \begin{bmatrix} -1 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix},$$

$$z_2 = \operatorname{adj}(A - a_{22}I_3)e_2 = \begin{bmatrix} \sqrt{2} \\ 0 \\ 0 \end{bmatrix}, \tag{4.34}$$

$$z_3 = \operatorname{adj}(A - a_{33}I_3)e_3 = \begin{bmatrix} \sqrt{2} \\ -\ 1 \\ -\ 1 \end{bmatrix}.$$

That is

$$Z = \begin{bmatrix} -1 & \sqrt{2} & \sqrt{2} \\ \sqrt{2} & 0 & -1 \\ \sqrt{2} & 0 & -1 \end{bmatrix}, \tag{4.35}$$

which clearly has only rank 2.

4.3 Local Convergence Properties of the Shifted QR-Algorithm

We consider the QR-algorithm with any smooth shift strategy. Given $A = A^\top$ with distinct eigenvalues. Consider the following mapping on the orthogonal similarity orbit of A

$$A \mapsto (A - \mu(A)I_n)_Q^\top (A - \mu(A)I_n)(A - \mu(A)I_n)_Q. \tag{4.36}$$

Iterating the mapping (4.36) is usually referred to as the shifted QR-algorithm with shift strategy μ. Alternatevely, one might consider two closely related versions of the shifted QR-algorithm living on $\mathcal{O}_n$

$$X \mapsto \big((A - \mu(X)I_n)X\big)_Q \tag{4.37}$$

and

$$X \mapsto \big((A - \mu(X)I_n)^{-1}X\big)_Q. \tag{4.38}$$

Rewrite (4.38) into

$$\sigma : X \mapsto \big(\operatorname{adj}(A - \mu(X)I_n)X\big)_Q. \tag{4.39}$$

Without loss of generality we assume $A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Assume further that the dynamical system defined by iterating (4.39) on $\mathcal{O}_n$ converges to $X = I_n$. Then for $\xi \in T_I\mathcal{O}_n \cong \mathfrak{so}_n$ a tedious calculation shows that

$$\begin{aligned} \mathrm{D}\,\sigma(I_n)\xi &= \mathrm{D}\left(\operatorname{adj}(A-\mu(X)I)X\right)_Q \cdot \xi\Big|_{X=I_n} \\ &= \left(\mathrm{D}\left((\operatorname{adj}(A-\mu(X)I)X\right)\cdot\xi\right)_{\text{skewsym.}}\Big|_{X=I_n} \\ &= \left(\begin{bmatrix} \prod_{i\neq 1}(\lambda_i - \mu(I_n)) & & \\ & \ddots & \\ & & \prod_{i\neq n}(\lambda_i-\mu(I_n)) \end{bmatrix} \cdot \xi\right)_{\text{skewsym.}}, \end{aligned} \tag{4.40}$$

where $(Z)_{\text{skewsym.}}$ denotes the skewsymmetric summand from the unique additive decomposition of Z into skewsymmetric and upper triangular part. Obviously, there cannot exist a smooth function $\mu : \mathcal{O}_n \to \mathbb{R}$, such that $\mathrm{D}\,\sigma(I_n) = 0$, because this would require that $\prod_{i\neq j}(\lambda_i - \mu(I_n)) = 0$ for all $j = 1, \ldots, n$, being clearly impossible. We therefore have proved

Theorem 4.4. *There exists no smooth scalar shift strategy to ensure quadratic convergence for the QR-algorithm.* □

This theorem indicates that either deflation or a matrix valued shift strategy is necessary for the shifted QR-algorithm to be efficient.

Bibliography

[AB77] B.D.O. Anderson and R.R. Bitmead. The matrix Cauchy index: Properties and applications. *SIAM J. Appl. Math.*, 33:655–672, 1977.

[AL90] H. Azad and J.J. Loeb. On a theorem of Kempf and Ness. *Ind. Univ. Math. J.*, 39(1):61–65, 1990.

[AMR88] R. Abraham, J.E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications.* Springer, New York, second edition, 1988.

[AMSD02] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren. A Grassmann–Rayleigh quotient iteration for computing invariant subspaces. *SIAM Review*, 44(1):57–73, 2002.

[AO82] T. Abatzoglou and B. O'Donnell. Minimization by coordinate descent. *J. of Optimization Theory and Applications*, 36(2):163–174, February 1982.

[Ati82] M.F. Atiyah. Convexity and commuting Hamiltonians. *Bull. London Math. Soc.*, 14:1–15, 1982.

[Bat95] S. Batterson. Dynamical analysis of numerical systems. *Numerical Linear Algebra with Applications*, 2(3):297–309, 1995.

[BD82] C.I. Byrnes and T.W. Duncan. On certain topological invariants arising in system theory. In P.J. Hilton and G.S. Young, editor, *New Directions in Applied Mathematics*, pages 29–72. Springer, New York, 1982.

[BGF91] A. Bunse-Gerstner and H. Fassbender. On the generalized Schur decomposition of a matrix pencil for parallel computation. *SIAM J. Sci. Stat. Comput.*, 12(4):911–939, 1991.

[BHH$^+$87] J.C. Bezdek, R.J. Hathaway, R.E. Howard, C.A. Wilson, and M.P. Windham. Local convergence analysis of a grouped variable version of coordinate descent. *J. of Optimization Theory and Applications*, 1987.

[BL85] R.P. Brent and F.T. Luk. The solution of singular value and symmetric eigenvalue problems on multiprocessor arrays. *SIAM J. Sci. Stat. Comput.*, 6(1):69–84, 1985.

[Bro88] R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. In *Proc. IEEE of the 27th Conference on Decision and Control*, pages 799–803, Austin, TX, 12 1988. See also *Lin. Algebra & Applic.*, 146:79-91, 1991.

[BS89a] S. Batterson and J. Smillie. The dynamics of Rayleigh quotient iteration. *SIAM J. Num. Anal.*, 26(3):624–636, 1989.

[BS89b] S. Batterson and J. Smillie. Rayleigh quotient iteration fails for nonsymmetric matrices. *Appl. Math. Lett.*, 2(1):19–20, 1989.

[BS90] S. Batterson and J. Smillie. Rayleigh quotient iteration for nonsymmetric matrices. *Math. of Computation*, 55(191):169–178, 1990.

[BSS93] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming*. John Wiley & Sons, New York, second edition, 1993.

[CD89] J.-P. Charlier and P. Van Dooren. A Jacobi-like algorithm for computing the generalized Schur form of a regular pencil. *Journal of Computational and Applied Mathematics*, 27:17–36, 1989.

[CD90] M.T. Chu and K.R. Driessel. The projected gradient method for least squares matrix approximations with spectral constraints. *SIAM J. Num. Anal.*, 27(4):1050–1060, 1990.

[Cha84] F. Chatelin. Simultaneous Newton's iteration for the eigenproblem. *Computing, Suppl.*, 5:67–74, 1984.

[Chu88] M.T. Chu. On the continuous realization of iterative processes. *SIAM Review*, 30:375–387, 1988.

[Chu91] M.T. Chu. A continuous Jacobi-like approach to the simultaneous reduction of real matrices. *Lin. Algebra & Applic.*, 147:75–96, 1991.

[Chu96] M.T. Chu. Continuous realization methods and their applications, March 1996. Notes prepared for lecture presentations given at ANU, Canberra, Australia.

[Deh95] J. Dehaene. *Continuous-time matrix algorithms systolic algorithms and adaptive neural networks.* PhD thesis, Katholieke Universiteit Leuven, October 1995.

[Dem87] J. Demmel. Three methods for refining estimates of invariant subspaces. *Computing*, 38:43–57, 1987.

[DMW83] J.J. Dongarra, C.B. Moler, and J.H. Wilkinson. Improving the accuracy of computed eigenvalues and eigenvectors. *SIAM J. Num. Anal.*, 20(1):23–45, 1983.

[DV92] J. Demmel and K. Veselić. Jacobi's method is more accurate than QR. *SIAM J. Matrix Anal. Appl.*, 13:1204–1245, 1992.

[Ful97] W. Fulton. *Young Tableaux.* LMS Student Texts 35. Cambridge Univ. Press, 1997.

[Gib79] C.G. Gibson. *Singular points of smooth mappings.* Pitman, Boston, 1979.

[Göt94] J. Götze. On the parallel implementation of Jacobi's and Kogbetliantz's algorithms. *SIAM J. Sci. Stat. Comput.*, 15(6):1331–1348, 1994.

[Göt95] J. Götze. *Orthogonale Matrixtransformationen, Parallele Algorithmen, Architekturen und Anwendungen.* Oldenbourg, München, 1995. in German.

[GS82] V. Guillemin and S. Sternberg. Convexity properties of the moment mapping. *Inventiones Math.*, 67:491–513, 1982.

[GvL89] G. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[Hac93] D. Hacon. Jacobi's method for skew-symmetric matrices. *SIAM J. Matrix Anal. Appl.*, 14(3):619–628, 1993.

[Hen58] P. Henrici. On the speed of convergence of cyclic and quasicyclic Jacobi methods for computing eigenvalues of Hermitian matrices. *J. Soc. Indust. Appl. Math.*, 6(2):144–162, 1958.

[HH95] K. Hüper and U. Helmke. Geometrical methods for pole assignment algorithms. In *Proc. IEEE of the 34th Conference on Decision and Control*, New Orleans, USA, 1995.

[HH97] U. Helmke and K. Hüper. The Jacobi method: A tool for computation and control. In C.I. Byrnes, B.N. Datta, C.F. Martin, and D.S. Gilliam, editors, *Systems and Control in the Twenty-First Century*, pages 205–228, Boston, 1997. Birkhäuser.

[HH98] K. Hüper and U. Helmke. Jacobi-type methods in computer vision: A case study. *Z. Angew. Math. Mech.*, 78:S945–S948, 1998.

[HH00] U. Helmke and K. Hüper. A Jacobi-type method for computing balanced realizations. *Systems & Control Letters*, 39:19–30, 2000.

[HHM96] K. Hüper, U. Helmke, and J.B. Moore. Structure and convergence of conventional Jacobi-type methods minimizing the off-norm function. In *Proc. IEEE of the 35th Conference on Decision and Control*, pages 2124–2128, Kobe, Japan, 1996.

[HHM02] U. Helmke, K. Hüper, and J.B. Moore. Computation of signature symmetric balanced realizations. *Journal of Global Optimization*, 2002.

[HM94] U. Helmke and J.B. Moore. *Optimization and Dynamical Systems*. CCES. Springer, London, 1994.

[Hüp96] K. Hüper. *Structure and convergence of Jacobi-type methods for matrix computations.* PhD thesis, Technical University of Munich, June 1996.

[Jac46] C.G.J. Jacobi. Über ein leichtes Verfahren, die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen. *Crelle's J. für die reine und angewandte Mathematik*, 30:51–94, 1846.

[Kle00] M. Kleinsteuber. Das Jacobi-Verfahren auf kompakten Lie-Algebren, 2000. Diplomarbeit, Universität Würzburg.

[KN79] G. Kempf and L. Ness. The length of vectors in representation spaces. *Lect. Notes in Math. 732*, pages 233–243, 1979.

[LHPW87] A.J. Laub, M.T. Heath, C.C. Paige, and R.C. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Transactions on Automatic Control*, 32(2):115–122, 1987.

[LT92] Z.Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *J. of Optimization Theory and Applications*, 72(1):7–35, January 1992.

[Lue84] D. G. Luenberger, editor. *Linear and nonlinear programming.* Addison-Wesley, Reading, 2nd edition, 1984.

[Lut92] A. Lutoborski. On the convergence of the Euler-Jacobi method. *Numer. Funct. Anal. and Optimiz.*, 13(1& 2):185–202, 1992.

[Mac95] N. Mackey. Hamilton and Jacobi meet again: Quaternions and the eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 16(2):421–435, 1995.

[Mah94] R. Mahony. *Optimization algorithms on homogeneous spaces.* PhD thesis, Australian National University, Canberra, March 1994.

[Meh02] C. Mehl. Jacobi-like algorithms for the indefinite generalized Hermitian eigenvalue problem. Technical Report 738-02, Technische Universität Berlin, Institut für Mathematik, June 2002.

[Nai90] M.T. Nair. Computable error estimates for Newton‘s iterations for refining invariant subspaces. *Indian J. Pure and Appl. Math.*, 21(12):1049–1054, December 1990.

[Ost59] A.M. Ostrowski. On the convergence of the Rayleigh quotient iteration for the computation of characteristic roots and vectors. *Arch. rational Mech. Anal.*, 1-4:233–241,423–428,325–340,341–347,472–481,153–165, 1958/59.

[Paa71] M.H.C. Paardekooper. An eigenvalue algorithm for skew-symmetric matrices. *Num. Math.*, 17:189–202, 1971.

[Par74] B.N. Parlett. The Rayleigh quotient iteration and some generalizations for nonnormal matrices. *Math. of Computation*, 28(127):679–693, 1974.

[Par80] B.N. Parlett. *The symmetric eigenvalue problem.* Prentice Hall, 1980.

[RH95] N.H. Rhee and V. Hari. On the cubic convergence of the Paardekooper method. *BIT*, 35:116–132, 1995.

[Sam71] A. Sameh. On Jacobi and Jacobi-like algorithms for a parallel computer. *Math. of Computation*, 25:579–590, 1971.

[SC89] M.G. Safonov and R.Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Transactions on Automatic Control*, 34(7):729–733, 1989.

[SHS72] H.R. Schwarz, H.Rutishauser, and E. Stiefel. *Numerik symmetrischer matrizen.* B.G. Teubner, Stuttgart, 1972.

[Smi93] S.T. Smith. *Geometric optimization methods for adaptive filtering.* PhD thesis, Harvard University, Cambridge, May 1993.

[Ste73] G.W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, October 1973.

[Ste86] G.W. Stewart. A Jacobi-like algorithm for computing the Schur decomposition of a nonhermitian matrix. *SIAM J. Sci. Stat. Comput.*, 6(4):853–864, October 1986.

[SV87] M. Shub and A.T. Vasquez. Some linearly induced Morse-Smale systems, the QR algorithm and the Toda lattice. *Contemporary Math.*, 64:181–194, 1987.

[Var84] V.S. Varadarajan. *Lie Groups Lie Algebras, and Their Representations.* Number 102 in GTM. Springer, New York, 1984.