Chapter Six

Newton's Method

This chapter provides a detailed development of the archetypal second-order optimization method, Newton's method, as an iteration on manifolds. We propose a formulation of Newton's method for computing the zeros of a vector field on a manifold equipped with an affine connection and a retraction. In particular, when the manifold is Riemannian, this geometric Newton method can be used to compute critical points of a cost function by seeking the zeros of its gradient vector field. In the case where the underlying space is Euclidean, the proposed algorithm reduces to the classical Newton method. Although the algorithm formulation is provided in a general framework, the applications of interest in this book are those that have a matrix manifold structure (see Chapter 3). We provide several example applications of the geometric Newton method for principal subspace problems.

6.1 NEWTON'S METHOD ON MANIFOLDS

In Chapter 5 we began a discussion of the Newton method and the issues involved in generalizing such an algorithm on an arbitrary manifold. Section 5.1 identified the task as computing a zero of a vector field ξ on a Riemannian manifold $\mathcal{M}$ equipped with a retraction R. The strategy proposed was to obtain a new iterate x_{k+1} from a current iterate x_k by the following process.

1. Find a tangent vector $\eta_k \in T_{x_k}\mathcal{M}$ such that the "directional derivative" of ξ along η_k is equal to $-\xi$.
2. Retract η_k to obtain x_{k+1}.

In Section 5.1 we were unable to progress further without providing a generalized definition of the directional derivative of ξ along η_k. The notion of an affine connection, developed in Section 5.2, is now available to play such a role, and we have all the tools necessary to propose Algorithm 4, a geometric Newton method on a general manifold equipped with an affine connection and a retraction.

By analogy with the classical case, the operator

$$J(x) : T_x\mathcal{M} \to T_x\mathcal{M} : \eta \mapsto \nabla_\eta \xi$$

involved in (6.1) is called the *Jacobian of* ξ *at* x. Equation (6.1) is called the *Newton equation*, and its solution $\eta_k \in T_{x_k}\mathcal{M}$ is called the *Newton vector*.

Algorithm 4 Geometric Newton method for vector fields

Require: Manifold $\mathcal{M}$; retraction R on $\mathcal{M}$; affine connection ∇ on $\mathcal{M}$; vector field ξ on $\mathcal{M}$.
Goal: Find a zero of ξ, i.e., $x \in \mathcal{M}$ such that $\xi_x = 0$.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Sequence of iterates $\{x_k\}$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Solve the Newton equation

$$J(x_k)\eta_k = -\xi_{x_k} \tag{6.1}$$

for the unknown $\eta_k \in T_{x_k}\mathcal{M}$, where $J(x_k)\eta_k := \nabla_{\eta_k}\xi$.
3: Set

$$x_{k+1} := R_{x_k}(\eta_k).$$

4: **end for**

In Algorithm 4, the choice of the retraction R and the affine connection ∇ is not prescribed. This freedom is justified by the fact that superlinear convergence holds for every retraction R and every affine connection ∇ (see forthcoming Theorem 6.3.2). Nevertheless, if $\mathcal{M}$ is a Riemannian manifold, there is a natural connection—the Riemannian connection—and a natural retraction—the exponential mapping. From a computational viewpoint, choosing ∇ as the Riemannian connection is generally a good choice, notably because it admits simple formulas on Riemannian submanifolds and on Riemannian quotient manifolds (Sections 5.3.3 and 5.3.4). In contrast, instead of choosing R as the exponential mapping, it is usually desirable to consider alternative retractions that are computationally more efficient; examples are given in Section 4.1.

When $\mathcal{M}$ is a Riemannian manifold, it is often advantageous to wrap Algorithm 4 in a line-search strategy using the framework of Algorithm 1. At the current iterate x_k, the search direction η_k is computed as the solution of the Newton equation (6.1), and x_{k+1} is computed to satisfy the descent condition (4.12) in which the cost function f is defined as

$$f := \langle \xi, \xi \rangle.$$

Note that the global minimizers of f are the zeros of the vector field ξ. Moreover, if ∇ is the Riemannian connection, then, in view of the compatibility with the Riemannian metric (Theorem 5.3.1.ii), we have

$$\mathrm{D}\langle \xi, \xi \rangle (x_k)[\eta_k] = \langle \nabla_{\eta_k}\xi, \xi \rangle + \langle \xi, \nabla_{\eta_k}\xi \rangle = -2\langle \xi, \xi \rangle_{x_k} < 0$$

whenever $\xi \neq 0$. It follows that the Newton vector η_k is a descent direction for f, although $\{\eta_k\}$ is not necessarily gradient-related. This perspective provides another motivation for choosing ∇ in Algorithm 4 as the Riemannian connection.

Note that an analytical expression of the Jacobian $J(x)$ in the Newton equation (6.1) may not be available. The Jacobian may also be singular

or ill-conditioned, in which case the Newton equation cannot be reliably solved for η_k. Remedies to these difficulties are provided by the quasi-Newton approaches presented in Section 8.2.

6.2 RIEMANNIAN NEWTON METHOD FOR REAL-VALUED FUNCTIONS

We now discuss the case $\xi = \operatorname{grad} f$, where f is a cost function on a Riemannian manifold $\mathcal{M}$. The Newton equation (6.1) becomes

$$\operatorname{Hess} f(x_k)\eta_k = -\operatorname{grad} f(x_k), \tag{6.2}$$

where

$$\operatorname{Hess} f(x) : T_x\mathcal{M} \to T_x\mathcal{M} : \eta \mapsto \nabla_\eta \operatorname{grad} f \tag{6.3}$$

is the Hessian of f at x for the affine connection ∇. We formalize the method in Algorithm 5 for later reference. Note that Algorithm 5 is a particular case of Algorithm 4.

Algorithm 5 Riemannian Newton method for real-valued functions

Require: Riemannian manifold $\mathcal{M}$; retraction R on $\mathcal{M}$; affine connection ∇ on $\mathcal{M}$; real-valued function f on $\mathcal{M}$.
Goal: Find a critical point of f, i.e., $x \in \mathcal{M}$ such that $\operatorname{grad} f(x) = 0$.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Sequence of iterates $\{x_k\}$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Solve the Newton equation

$$\operatorname{Hess} f(x_k)\eta_k = -\operatorname{grad} f(x_k) \tag{6.4}$$

for the unknown $\eta_k \in T_{x_k}\mathcal{M}$, where $\operatorname{Hess} f(x_k)\eta_k := \nabla_{\eta_k} \operatorname{grad} f$.
3: Set

$$x_{k+1} := R_{x_k}(\eta_k).$$

4: **end for**

In general, the Newton vector η_k, solution of (6.2), is not necessarily a descent direction of f. Indeed, we have

$$\mathrm{D}f(x_k)[\eta_k] = \langle \operatorname{grad} f(x_k), \eta_k \rangle = -\langle \operatorname{grad} f(x_k), (\operatorname{Hess} f(x_k))^{-1} \operatorname{grad} f(x_k) \rangle, \tag{6.5}$$

which is not guaranteed to be negative without additional assumptions on the operator $\operatorname{Hess} f(x_k)$. A sufficient condition for η_k to be a descent direction is that $\operatorname{Hess} f(x_k)$ be *positive-definite* (i.e., $\langle \xi, \operatorname{Hess} f(x_k)[\xi] \rangle > 0$ for all $\xi \neq 0_{x_k}$). When ∇ is a symmetric affine connection (such as the Riemannian connection), $\operatorname{Hess} f(x_k)$ is positive-definite if and only if all its eigenvalues are strictly positive.

In order to obtain practical convergence results, quasi-Newton methods have been proposed that select an update vector η_k as the solution of

$$(\mathrm{Hess}\, f(x_k) + E_k)\eta_k = -\mathrm{grad}\, f(x_k), \tag{6.6}$$

where the operator E_k is chosen so as to make the operator $(\mathrm{Hess}\, f(x_k) + E_k)$ positive-definite. For a suitable choice of operator E_k this guarantees that the sequence $\{\eta_k\}$ is gradient-related, thereby fulfilling the main hypothesis in the global convergence result of Algorithm 1 (Theorem 4.3.1). Care should be taken that the operator E_k does not destroy the superlinear convergence properties of the pure Newton iteration when the desired (local minimum) critical point is reached.

6.3 LOCAL CONVERGENCE

In this section, we study the local convergence of the plain geometric Newton method as defined in Algorithm 4. Well-conceived globally convergent modifications of Algorithm 4 should not affect its superlinear local convergence.

The convergence result below (Theorem 6.3.2) shows quadratic convergence of Algorithm 4. Recall from Section 4.3 that the notion of quadratic convergence on a manifold $\mathcal{M}$ does not require any further structure on $\mathcal{M}$, such as a Riemannian structure. Accordingly, we make no such assumption. Note also that since Algorithm 5 is a particular case of Algorithm 4, the convergence analysis of Algorithm 4 applies to Algorithm 5 as well.

We first need the following lemma.

Lemma 6.3.1 *Let $\|\cdot\|$ be any consistent norm on $\mathbb{R}^{n\times n}$ such that $\|I\| = 1$. If $\|E\| < 1$, then $(I - E)^{-1}$ exists and*

$$\|(I - E)^{-1}\| \leq \frac{1}{1 - \|E\|}.$$

If A is nonsingular and $\|A^{-1}(B - A)\| < 1$, then B is nonsingular and

$$\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}(B - A)\|}.$$

Theorem 6.3.2 (local convergence of Newton's method) *Under the requirements and notation of Algorithm 4, assume that there exists $x_* \in \mathcal{M}$ such that $\xi_{x_*} = 0$ and $J(x_*)^{-1}$ exists. Then there exists a neighborhood $\mathcal{U}$ of x_* in $\mathcal{M}$ such that, for all $x_0 \in \mathcal{U}$, Algorithm 4 generates an infinite sequence $\{x_k\}$ converging superlinearly (at least quadratically) to x_*.*

Proof. Let $(\mathcal{U}, \varphi)$, $x_* \in \mathcal{U}$, be a coordinate chart. According to Section 4.3, it is sufficient to show that the sequence $\{\varphi(x_k)\}$ in $\mathbb{R}^d$ converges quadratically to $\varphi(x_*)$. To simplify the notation, coordinate expressions are denoted by hat quantities. In particular, $\hat{x}_k = \varphi(x_k)$, $\hat{\xi}_{\hat{x}_k} = \mathrm{D}\varphi(x_k)[\xi]$, $\hat{J}(\hat{x}_k) = (\mathrm{D}\varphi(x_k)) \circ J(x_k) \circ (\mathrm{D}\varphi(x_k))^{-1}$, $\hat{R}_{\hat{x}}\hat{\zeta} = \varphi(R_x\zeta)$. Note that $\hat{J}(\hat{x}_k)$ is a linear operator

from $\mathbb{R}^d$ to $\mathbb{R}^d$, i.e., a $d \times d$ matrix. Note also that $\hat{R}_{\hat{x}}$ is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$ whose differential at zero is the identity.

The iteration defined by Algorithm 4 reads

$$\hat{x}_{k+1} = \hat{R}_{\hat{x}_k}(-\hat{J}(\hat{x}_k)^{-1}\hat{\xi}_{\hat{x}_k}), \tag{6.7}$$

whereas the classical Newton method applied to the function $\hat{\xi} : \mathbb{R}^d \mapsto \mathbb{R}^d$ would yield

$$\hat{x}_{k+1} = \hat{x}_k + (-\mathrm{D}\hat{\xi}(\hat{x}_k)^{-1}\hat{\xi}_{\hat{x}_k}). \tag{6.8}$$

The strategy in this proof is to show that (6.7) is sufficiently close to (6.8) that the superlinear convergence result of the classical Newton method is preserved. We are going to prove more than is strictly needed, in order to obtain information about the multiplicative constant in the quadratic convergence result.

Let $\beta := \|\hat{J}(\hat{x}_*)^{-1}\|$. Since ξ is a (smooth) vector field, it follows that $\hat{J}$ is smooth, too, and therefore there exists $r_J > 0$ and $\gamma_J > 0$ such that

$$\|\hat{J}(\hat{x}) - \hat{J}(\hat{y})\| \leq \gamma_J \|\hat{x} - \hat{y}\|$$

for all $\hat{x}, \hat{y} \in B_{r_J}(\hat{x}_*) := \{\hat{x} \in \mathbb{R}^d : \|\hat{x} - \hat{x}_*\| < r_J\}$. Let

$$\epsilon = \min\left\{r_J, \frac{1}{2\beta\gamma_J}\right\}.$$

Assume that $\hat{x}_k \in B_\epsilon(\hat{x}_*)$. It follows that

$$\begin{aligned}\|\hat{J}(\hat{x}_*)^{-1}(\hat{J}(\hat{x}_k) - \hat{J}(\hat{x}_*))\| &\leq \|\hat{J}(\hat{x}_*)\|^{-1}\|\hat{J}(\hat{x}_k) - \hat{J}(\hat{x}_*)\| \\ &\leq \beta\gamma_J\|\hat{x}_k - \hat{x}_*\| \leq \beta\gamma_J\epsilon \leq \frac{1}{2}.\end{aligned}$$

It follows from Lemma 6.3.1 that $\hat{J}(\hat{x}_k)$ is nonsingular and that

$$\|\hat{J}(\hat{x}_k)^{-1}\| \leq \frac{\|\hat{J}(\hat{x}_*)^{-1}\|}{1 - \|\hat{J}(\hat{x}_*)^{-1}(\hat{J}(\hat{x}_k) - \hat{J}(\hat{x}_*))\|} \leq 2\|\hat{J}(\hat{x}_*)^{-1}\| \leq 2\beta.$$

It also follows that for all $\hat{x}_k \in B_\epsilon(\hat{x}_*)$, the Newton vector $\hat{\eta}_k := \hat{J}(\hat{x}_k)^{-1}\hat{\xi}_{\hat{x}_k}$ is well defined. Since R is a retraction (thus a smooth mapping) and $\hat{x}_*$ is a zero of $\hat{\xi}$, it follows that there exists r_R and $\gamma_R > 0$ such that

$$\|\hat{R}_{\hat{x}_k}\hat{\eta}_k - (\hat{x}_k + \hat{\eta}_k)\| \leq \gamma_R\|\hat{x}_k - \hat{x}_*\|^2$$

for all $\hat{x}_k \in B_\epsilon(\hat{x}_*)$. (Indeed, since $\|\hat{J}(\hat{x}_k)^{-1}\|$ is bounded on $B_\epsilon(\hat{x}_*)$, and $\hat{\xi}$ is smooth and $\hat{\xi}_{\hat{x}_*} = 0$, we have a bound $\|\hat{\eta}_k\| \leq c\|\hat{x}_k - \hat{x}_*\|$ for all x_k in a neighborhood of x_*; and in view of the local rigidity property of R, we have $\|\hat{R}_{\hat{x}_k}\hat{\eta}_k - (\hat{x}_k + \hat{\eta}_k)\| \leq c\|\hat{\eta}_k\|^2$ for all x_k in a neighborhood of x_* and all η_k sufficiently small.)

Define $\hat{\Gamma}_{\hat{x},\hat{\xi}}$ by $\hat{\Gamma}_{\hat{x},\hat{\xi}}\hat{\zeta} := \hat{J}(\hat{x})\hat{\zeta} - \mathrm{D}\hat{\xi}(\hat{x})\left[\hat{\zeta}\right]$; see (5.7). Note that $\hat{\Gamma}_{\hat{x},\hat{\xi}}$ is a linear operator. Again by a smoothness argument, it follows that there exists r_Γ and γ_Γ such that

$$\|\hat{\Gamma}_{\hat{x},\hat{\xi}} - \hat{\Gamma}_{\hat{y},\hat{\xi}}\| \leq \gamma_\Gamma\|\hat{x} - \hat{y}\|$$

for all $\hat{x}, \hat{y} \in B_{r_\Gamma}(\hat{x}_*)$. In particular, since $\hat{\xi}_{\hat{x}_*} = 0$, it follows from the uniqueness of the connection at critical points that $\hat{\Gamma}_{\hat{x}_*,\hat{\xi}} = 0$, hence

$$\|\hat{\Gamma}_{\hat{x},\hat{\xi}}\| \leq \gamma_\Gamma \|\hat{x}_k - \hat{x}_*\|$$

for all $\hat{x}_k \in B_\epsilon(\hat{x}_*)$.

We need a Lipschitz constant for $\mathrm{D}\hat{\xi}$. For all $\hat{x}, \hat{y} \in B_{\min\{r_J, r_\Gamma\}}(\hat{x}_*)$, we have

$$\begin{aligned}
&\|\mathrm{D}\hat{\xi}(\hat{x}) - \mathrm{D}\hat{\xi}(\hat{y})\| - \|\hat{\Gamma}_{\hat{x},\hat{\xi}} - \hat{\Gamma}_{\hat{y},\hat{\xi}}\| \\
&\quad \leq \|\mathrm{D}\hat{\xi}(\hat{x}) + \hat{\Gamma}_{\hat{x},\hat{\xi}} - \left(\mathrm{D}\hat{\xi}(\hat{y}) + \hat{\Gamma}_{\hat{y},\hat{\xi}}\right)\| = \|\hat{J}(\hat{x}) - \hat{J}(\hat{y})\| \leq \gamma_J \|\hat{x} - \hat{y}\|,
\end{aligned}$$

hence

$$\|\mathrm{D}\hat{\xi}(\hat{x}) - \mathrm{D}\hat{\xi}(\hat{y})\| \leq (\gamma_J + \gamma_\Gamma)\|\hat{x} - \hat{y}\|.$$

From (6.7) we have

$$\hat{x}_{k+1} - \hat{x}_* = \hat{R}_{\hat{x}_k}(-\hat{J}(\hat{x}_k)^{-1}\hat{\xi}_{\hat{x}_k}) - \hat{x}_*.$$

Applying the bounds developed above, one obtains

$$\begin{aligned}
\|\hat{x}_{k+1} - \hat{x}_*\| \leq& \|\hat{x}_k - \hat{J}(\hat{x}_k)^{-1}\hat{\xi}_{\hat{x}_k} - \hat{x}_*\| + \gamma_R \|\hat{x}_k - \hat{x}_*\|^2 \\
\leq& \|\hat{J}(\hat{x}_k)^{-1}\left(\hat{\xi}_{\hat{x}_*} - \hat{\xi}_{\hat{x}_k} - \hat{J}(\hat{x}_k)(\hat{x}_* - \hat{x}_k)\right)\| + \gamma_R \|\hat{x}_k - \hat{x}_*\|^2 \\
\leq& \|\hat{J}(\hat{x}_k)^{-1}\| \|\hat{\xi}_{\hat{x}_*} - \hat{\xi}_{\hat{x}_k} - \mathrm{D}\hat{\xi}\,(\hat{x}_k)\,[\hat{x}_* - \hat{x}_k]\,\| \\
&+ \|\hat{J}(\hat{x}_k)^{-1}\| \|\hat{\Gamma}_{\hat{x}_k,\hat{\xi}}(\hat{x}_* - \hat{x}_k)\| + \gamma_R \|\hat{x}_k - \hat{x}_*\|^2 \\
\leq& 2\,\beta\,\frac{1}{2}(\gamma_J + \gamma_\Gamma)\|\hat{x}_k - \hat{x}_*\|^2 \\
&+ 2\,\beta\,\gamma_\Gamma \|\hat{x}_k - \hat{x}_*\|^2 + \gamma_R \|\hat{x}_k - \hat{x}_*\|^2
\end{aligned}$$

whenever $\|\hat{x}_k - \hat{x}_*\| \leq \min\{\epsilon, r_\Gamma, r_R\}$, where we have used Proposition A.6.1. This completes the proof. □

It is interesting to note that in the classical Euclidean case, the proof holds with $\gamma_R = 0$ (because $R_x\zeta := x + \zeta$) and $\gamma_\Gamma = 0$ (because $J(x)\zeta \equiv \nabla_\zeta \xi := \mathrm{D}\xi\,(x)\,[\zeta]$).

In the case where $\mathcal{M}$ is a Riemannian metric and the Riemannian connection is used along with a second-order retraction (e.g., the exponential retraction), it is also possible to obtain a better bound. Consider normal coordinates around the point x_*. The Christoffel symbols Γ^i_{jk} vanish at $\hat{x}_*$, and the constant γ_Γ can be replaced by $O(\|\hat{x}_k - \hat{x}_*\|)$. Since we are working in normal coordinates around x_*, it follows that the Christoffel symbols at x_* vanish, hence the acceleration condition $\frac{\mathrm{D}^2}{\mathrm{d}t^2} R_{x_*}(t\zeta_{x_*})\Big|_{t=0} = 0$ yields $\frac{\mathrm{d}^2}{\mathrm{d}t^2} \hat{R}_{\hat{x}_*}(t\hat{\zeta}_{\hat{x}_*})\Big|_{t=0} = 0$ and, by the smoothness of R, we have $\mathrm{D}^2\hat{R}_{\hat{x}_k} = O(\|\hat{x}_k - \hat{x}_*\|)$. It follows that γ_R may be replaced by $O(\|\hat{x}_k - \hat{x}_*\|)$. Thus, the convergence bound becomes

$$\begin{aligned}
\|\hat{x}_{k+1} - \hat{x}_*\| \leq& 2\beta\frac{1}{2}(\gamma_J + \gamma_\Gamma)\|\hat{x}_k - \hat{x}_*\|^2 + 2\beta\gamma_\Gamma\|\hat{x}_k - \hat{x}_*\|^2 + \gamma_R\|\hat{x}_k - \hat{x}_*\|^2 \\
\leq& \beta\gamma_J\|\hat{x}_k - \hat{x}_*\|^2 + O(\|\hat{x}_k - \hat{x}_*\|^3).
\end{aligned}$$

In normal coordinates at x_* one has that $\mathrm{dist}(x_k, x_*) = \|\hat{x}_k - \hat{x}_*\|$.

6.3.1 Calculus approach to local convergence analysis

Theorem 6.3.2 provides a strong convergence analysis of the geometric Newton method along with explicit convergence bounds. A weaker quadratic convergence result can be obtained from a local coordinate analysis of the Newton iteration using the calculus-based convergence result of Theorem 4.5.3.

Let x_* be a critical point of a vector field ξ with a nondegenerate Jacobian at x_*. Choose a coordinate chart around x^* and use the hat notation to represent the coordinate expression of geometric objects. Without loss of generality we choose $\hat{x}_* = 0$. The iteration defined in Algorithm 4 reads

$$\hat{x}_{k+1} = \hat{R}_{\hat{x}_k}(\hat{\eta}_k), \tag{6.9}$$

$$\hat{\nabla}_{\hat{\eta}_k}\hat{\xi} = -\hat{\xi}_{\hat{x}_k}. \tag{6.10}$$

Since the vector field ξ and the retraction R are smooth by assumption, this defines a smooth iteration mapping $\hat{x}_k \mapsto \hat{x}_{k+1}(\hat{x}_k)$. Evaluating the Newton equation (6.10) at $\hat{x}_k = \hat{x}_* = 0$ yields

$$\hat{\nabla}_{\hat{\eta}_0}\hat{\xi} = 0$$

and thus $\hat{\eta}_0 = 0$ because the Jacobian $J(x_*) : \zeta \in T_{x_*}\mathcal{M} \mapsto \nabla_\zeta \xi \in T_{x_*}\mathcal{M}$ is assumed to be nondegenerate. Since R satisfies the consistency property $R_x(0_x) = x$ for all x, it follows that $\hat{x}_* = 0$ is a fixed point of the iteration mapping. Recalling Theorem 4.5.3, it is sufficient to show that $\mathrm{D}\hat{x}_{k+1}(x_*) = 0$ to prove local quadratic convergence. For clarity, we use the notation $\hat{R}(\hat{x}, \hat{\eta})$ for $\hat{R}_{\hat{x}}(\hat{\eta})$, and we let $\mathrm{D}_1\hat{R}$ and $\mathrm{D}_2\hat{R}$ denote the differentials with respect to the first and second arguments of the function $\hat{R}$. (Note that $\hat{R}$ is a function from $\mathbb{R}^d \times \mathbb{R}^d$ into $\mathbb{R}^d$, where d is the dimension of the manifold $\mathcal{M}$.) Differentiating the iteration mapping $\hat{x}_k \mapsto \hat{x}_{k+1}(\hat{x}_k)$ at 0 along $\hat{\zeta}$, one obtains

$$\mathrm{D}\hat{x}_{k+1}(0)[\hat{\zeta}] = \mathrm{D}_1\hat{R}(0,0)[\hat{\zeta}] + \mathrm{D}_2\hat{R}(0,0)[\mathrm{D}\hat{\eta}(0)[\hat{\zeta}]], \tag{6.11}$$

where $\hat{x} \mapsto \hat{\eta}(\hat{x})$ is the function implicitly defined by the Newton equation

$$\hat{\nabla}_{\hat{\eta}(\hat{x})}\hat{\xi} = -\hat{\xi}_{\hat{x}}. \tag{6.12}$$

We have $\mathrm{D}_1\hat{R}(0,0)[\hat{\zeta}] = \hat{\zeta}$ because of the consistency condition $R(0_x) = x$. Moreover, the local rigidity condition $\mathrm{D}R_x(0_x) = \mathrm{id}_{T_x\mathcal{M}}$ (see Definition 4.1.1) ensures that $\mathrm{D}_2\hat{R}(0,0)[\mathrm{D}\hat{\eta}(0)[\hat{\zeta}]] = \mathrm{D}\hat{\eta}(0)[\hat{\zeta}]$. Hence (6.11) yields

$$\mathrm{D}\hat{x}_{k+1}(0)[\hat{\zeta}] = \hat{\zeta} + \mathrm{D}\hat{\eta}(0)[\hat{\zeta}]. \tag{6.13}$$

Using the local expression (5.7) for the affine connection, the Newton equation (6.12) reads

$$\mathrm{D}\hat{\xi}(\hat{x})[\hat{\eta}(\hat{x})] + \hat{\Gamma}_{\hat{x},\hat{\xi}_{\hat{x}}}\hat{\eta}(\hat{x}) = -\hat{\xi}_{\hat{x}}.$$

(Recall that $\hat{\Gamma}_{\hat{x},\hat{\xi}_{\hat{x}}}$ is a matrix and $\hat{\Gamma}_{\hat{x},\hat{\xi}_{\hat{x}}}\hat{\eta}(\hat{x})$ is a matrix-vector product.) Differentiating this equation with respect to $\hat{x}$ along $\hat{\zeta}$, one obtains

$$\mathrm{D}^2\hat{\xi}(\hat{x})[\hat{\eta}(\hat{x}), \hat{\zeta}] + \mathrm{D}\hat{\xi}(\hat{x})[\mathrm{D}\hat{\eta}(\hat{x})[\hat{\zeta}]] + \mathrm{D}\hat{\Gamma}_{\cdot,\hat{\xi}_\cdot}(\hat{x})[\hat{\zeta}]\hat{\eta}(\hat{x}) + \hat{\Gamma}_{\hat{x},\hat{\xi}_{\hat{x}}}\mathrm{D}\hat{\eta}(\hat{x})[\hat{\zeta}] = -\mathrm{D}\hat{\xi}(\hat{x})[\hat{\zeta}].$$

Most of the terms in this equation vanish when evaluated at $\hat{x} = 0$ since $\hat{\xi}_0 = 0$ and $\hat{\eta}_0 = 0$. (In particular, observe that $\hat{\Gamma}_{0,0} = 0$ in view of (5.8).) This leaves us with

$$\mathrm{D}\hat{\xi}(0)[\mathrm{D}\hat{\eta}(0)[\hat{\zeta}]] = -\mathrm{D}\hat{\xi}(0)[\hat{\zeta}]. \tag{6.14}$$

Since $J(x_*)$ is nonsingular and $\hat{\Gamma}_{\hat{x}_*,\hat{\xi}_{\hat{x}_*}} = 0$, it follows in view of (5.7) that the linear operator $\mathrm{D}\hat{\xi}(0) = \hat{J}(\hat{x}_*)$ is nonsingular. Hence (6.14) reduces to

$$\mathrm{D}\hat{\eta}(0)[\hat{\zeta}] = -\hat{\zeta}.$$

Using this result in (6.13) yields $D\hat{x}_{k+1}(x_*) = 0$. From Theorem 4.5.3, it follows that the iteration $\hat{x}_k \mapsto \hat{x}_{k+1}(\hat{x}_k)$ converges locally quadratically to $\hat{x}_*$. Since quadratic convergence is independent of coordinate representation, this property holds for the Newton iteration on the manifold.

6.4 RAYLEIGH QUOTIENT ALGORITHMS

In this section we show how the geometric Newton algorithms can be turned into practical numerical algorithms for the optimization of various cost functions of the Rayleigh quotient type.

6.4.1 Rayleigh quotient on the sphere

Recall the example of the Rayleigh quotient on the sphere first considered in Section 4.6. The main points were summarized in Table 4.1. The cost function is the Rayleigh quotient

$$f : S^{n-1} \to \mathbb{R} : x \mapsto x^T A x, \tag{6.15}$$

on the unit sphere S^{n-1}, viewed as a Riemannian submanifold of the Euclidean space $\mathbb{R}^n$. We also use the extension

$$\overline{f} : \mathbb{R}^n \to \mathbb{R} : x \mapsto x^T A x,$$

whose restriction to S^{n-1} is f. In Section 4.6, we obtained

$$\operatorname{grad} f(x) = 2\,\mathrm{P}_x(Ax) = 2\,(Ax - xx^T Ax),$$

where P_x is the orthogonal projector onto $T_x S^{n-1}$, i.e.,

$$\mathrm{P}_x z = z - xx^T z.$$

(Note that P_x can also be viewed as the matrix $(I - xx^T)$.) We also expressed a preference for the retraction

$$R_x(\xi) := \frac{x + \xi}{\|x + \xi\|}. \tag{6.16}$$

The geometric Newton method (Algorithm 4) requires an affine connection ∇. There is no reason not to pick the natural choice, the Riemannian

connection. Since S^{n-1} is a Riemannian submanifold of the Euclidean space $\mathbb{R}^n$, it follows from the material in Section 5.3.3 that

$$\nabla_\eta \xi = \mathrm{P}_x \left(\mathrm{D}\xi \left(x \right) [\eta] \right)$$

for every η in the tangent space $T_x S^{n-1} = \{z \in \mathbb{R}^n : x^T z = 0\}$ and every vector field ξ on S^{n-1}.

We are ready to apply the geometric Newton method (Algorithm 4) to the vector field $\xi := \operatorname{grad} f$, where f is the Rayleigh quotient (6.15). For every $\eta \in T_{x_k} S^{n-1}$, we have

$$\begin{aligned} \nabla_\eta \operatorname{grad} f(x) &= 2\mathrm{P}_x \left(\mathrm{D} \operatorname{grad} f(x)[\eta] \right) \\ &= 2\mathrm{P}_x (A\eta - \eta x^T A x) \\ &= 2(\mathrm{P}_x A \mathrm{P}_x \eta - \eta x^T A x), \end{aligned}$$

where we took into account that $\mathrm{P}_x x = 0$ and $\mathrm{P}_x \eta = \eta$. The last expression underscores the symmetry of the Hessian operator (in the sense of Proposition 5.5.3). Consequently, the Newton equation (6.1) reads

$$\begin{cases} \mathrm{P}_x A \mathrm{P}_x \eta - \eta x^T A x = -\mathrm{P}_x A x, \\ x^T \eta = 0, \end{cases} \tag{6.17}$$

where the second equation is the expression of the requirement $\eta \in T_x S^{n-1}$.

In conclusion, application of the geometric Newton method to $\xi := \operatorname{grad} f$, where f is the Rayleigh quotient (6.15) on the sphere S^{n-1}, viewed as a Riemannian submanifold of $\mathbb{R}^n$ endowed with its Riemannian connection and with the retraction (6.16), yields the matrix algorithm displayed in Algorithm 6. Since Algorithm 6 is a particular case of the forthcoming Algorithms 7 and 8, we postpone its analysis to Section 6.5.1.

Algorithm 6 Riemannian Newton method for the Rayleigh quotient on S^{n-1}

Require: Symmetric matrix A.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Sequence of iterates $\{x_k\}$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Solve the linear system (6.17), i.e.,

$$\begin{cases} \mathrm{P}_{x_k} A \mathrm{P}_{x_k} \eta_k - \eta_k x_k^T A x_k = -\mathrm{P}_{x_k} A x_k, \\ x_k^T \eta_k = 0, \end{cases} \tag{6.18}$$

for the unknown $\eta_k \in \mathbb{R}^n$.
3: Set

$$x_{k+1} := R_{x_k} \eta_k,$$

with R defined in (6.16).
4: **end for**

6.4.2 Rayleigh quotient on the Grassmann manifold

Consider the cost function

$$f : \mathrm{Grass}(p, n) \to \mathbb{R} : \mathrm{span}(Y) \mapsto \mathrm{tr}\left((Y^TY)^{-1}Y^TAY\right). \tag{6.19}$$

The first-order geometry of this cost function was investigated in Section 4.9 (see Table 4.3). The Grassmann manifold $\mathrm{Grass}(p, n)$ was viewed as a Riemannian quotient manifold of $(\mathbb{R}_*^{n\times p}, \overline{g})$ with

$$\overline{g}_Y(Z_1, Z_2) = \mathrm{tr}\left((Y^TY)^{-1}Z_1^TZ_2\right). \tag{6.20}$$

The horizontal distribution is

$$\mathcal{H}_Y = \{Z \in \mathbb{R}^{n\times p} : Y^TZ = 0\}, \tag{6.21}$$

the projection onto the horizontal space is

$$\mathrm{P}_Y^h = (I - Y(Y^TY)^{-1}Y^T), \tag{6.22}$$

and we obtained

$$\overline{\mathrm{grad}\, f}_Y = 2\mathrm{P}_Y^hAY = 2\left(AY - Y(Y^TY)^{-1}Y^TAY\right).$$

It follows from this expression that if $\mathrm{grad}\, f(\mathrm{span}(Y)) = 0$, then $\mathrm{span}(Y)$ is an invariant subspace of A. Conversely, if $\mathrm{span}(Y)$ is an invariant subspace of A, then there exists an M such that $AY = YM$; premultiplying this equation by $(Y^TY)^{-1}Y^T$ yields $M = (Y^TY)^{-1}Y^TAY$, and we obtain $\mathrm{grad}\, f(\mathrm{span}(Y)) = 0$. In conclusion, the critical points of f are the invariant subspaces of A.

In Section 5.3.4, we established the formula

$$\overline{\nabla_\eta \xi} = \mathrm{P}_Y^h\left(\mathrm{D}\overline{\xi}\,(Y)\,[\overline{\eta}_Y]\right) \tag{6.23}$$

for the Riemannian connection on the Grassmann manifold. This yields the following expression for the Hessian of the Rayleigh quotient cost function f:

$$\overline{\nabla_\eta \mathrm{grad}\, f} = \mathrm{P}_Y^h\left(\mathrm{D}\overline{\mathrm{grad}\, f}\,(Y)\,[\overline{\eta}_Y]\right) = 2\,\mathrm{P}_Y^h\left(A\overline{\eta}_Y - \overline{\eta}_Y(Y^TY)^{-1}Y^TAY\right),$$

where we have utilized the identity $\mathrm{P}_Y^h(YM) = (\mathrm{P}_Y^hY)M = 0$. Taking the horizontal lift of the Newton equation $\nabla_\eta \mathrm{grad}\, f = -\mathrm{grad}\, f(x)$ yields the equation

$$\mathrm{P}_Y^h\left(A\overline{\eta}_Y - \overline{\eta}_Y(Y^TY)^{-1}Y^TAY\right) = -\mathrm{P}_Y^hAY, \qquad \overline{\eta}_Y \in \mathcal{H}_Y,$$

whose solution $\overline{\eta}_Y$ in the horizontal space $\mathcal{H}_Y$ is the horizontal lift of the solution η of the Newton equation.

In conclusion, the geometric Newton method in Algorithm 4, for the Rayleigh quotient cost function (6.19), with the affine connection ∇ chosen as the Riemannian connection on $\mathrm{Grass}(p, n)$ seen as the Riemannian quotient manifold of $(\mathbb{R}_*^{n\times p}, \overline{g})$ with $\overline{g}$ in (6.20), and with the retraction R chosen as (4.40) yields the matrix algorithm displayed in Algorithm 7. The notation $Z_k = \overline{\eta}_{Y_k}$ is used to make the matrix expression resemble contemporary algorithms from the field of numerical linear algebra. The expression

Algorithm 7 Riemannian Newton method for the Rayleigh quotient on $\mathrm{Grass}(p, n)$

Require: Symmetric matrix A.
Input: Initial iterate $Y_0 \in \mathbb{R}_*^{n\times p}$.
Output: Sequence of iterates $\{Y_k\}$ in $\mathbb{R}_*^{n\times p}$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Solve the linear system

$$\begin{cases} \mathrm{P}^h_{Y_k}\left(AZ_k - Z_k(Y_k^TY_k)^{-1}Y_k^TAY_k\right) = -\mathrm{P}^h_{Y_k}(AY_k) \\ Y_k^TZ_k = 0 \end{cases} \tag{6.24}$$

for the unknown Z_k, where P^h_Y is the orthogonal projector defined in (6.22). (The condition $Y_k^TZ_k$ expresses that Z_k belongs to the horizontal space $\mathcal{H}_{Y_k}$.)
3: Set

$$Y_{k+1} = (Y_k + Z_k)N_k$$

where N_k is a nonsingular $p \times p$ matrix chosen for normalization purposes.
4: **end for**

in (6.24) can be simplified since $\mathrm{P}^h_{Y_k}Z_k = Z_k$. We will tend not to simplify such expressions in the matrix equations in order that the equations clearly reveal the underlying geometric structure (e.g., the quantity considered belongs to the range of $\mathrm{P}^h_{Y_k}$) or to emphasize the symmetry of certain operators.

Note that Algorithm 7 is the matrix expression of an algorithm defined on $\mathrm{Grass}(p, n)$. In other words, if $\{Y_k\}$ and $\{\check{Y}_k\}$ are two sequences generated by Algorithm 7 (with same matrix A) and if $\mathrm{span}(Y_0) = \mathrm{span}(\check{Y}_0)$, then $\mathrm{span}(Y_k) = \mathrm{span}(\check{Y}_k)$ for all k. Algorithm 7 thus could be written formally as an algorithm generating a sequence on $\mathrm{Grass}(p, n)$, by taking as input an element $\mathcal{Y}_0$ of $\mathrm{Grass}(p, n)$, picking $Y_0 \in \mathbb{R}_*^{n\times p}$ with $\mathrm{span}(Y_0) = \mathcal{Y}_0$, proceeding as in Algorithm 7, and returning the sequence $\{\mathrm{span}(Y_k)\}$.

Note also that when $p = 1$ and N_k (now a scalar) is chosen as $\|Y_k + Z_k\|^{-1}$, Algorithm 7 reduces to Algorithm 6.

6.4.3 Generalized eigenvalue problem

We assume that A and B are $n \times n$ symmetric matrices with B positive-definite, and we consider the generalized eigenvalue problem

$$Av = \lambda Bv$$

described in Section 2.1. With a view towards computing eigenspaces of a pencil (A, B), we consider the Rayleigh quotient function

$$f(\mathrm{span}(Y)) = \mathrm{tr}((Y^TAY)(Y^TBY)^{-1}), \tag{6.25}$$

where Y is a full-rank $n \times p$ matrix and $\mathrm{span}(Y)$ denotes the column space of Y. It is readily checked that the right-hand side depends only on $\mathrm{span}(Y)$, so that f is a well-defined real-valued function on the Grassmann manifold $\mathrm{Grass}(p, n)$.

As in the previous section, we view $\mathrm{Grass}(p, n)$ as a Riemannian quotient manifold of $(\mathbb{R}_*^{n\times p}, \overline{g})$ with

$$\overline{g}_Y(Z_1, Z_2) = \mathrm{tr}((Y^TY)^{-1}Z_1^TZ_2). \tag{6.26}$$

With a view towards applying the Riemannian Newton method given in Algorithm 5, we need formulas for the gradient and the Hessian of the Rayleigh cost function (6.25). Mimicking the calculations in Section 4.9, we obtain

$$\tfrac{1}{2}\overline{\mathrm{grad}\, f}_Y = P_{BY,Y}AY(Y^TBY)^{-1}Y^TY, \tag{6.27}$$

where

$$P_{U,V} = I - U(V^TU)^{-1}V^T$$

denotes the projector parallel to the span of U onto the orthogonal complement of the span of V. Note that the projection P_Y^h onto the horizontal space $\mathcal{H}_Y$ is given by

$$\mathrm{P}_Y^h = P_{Y,Y}.$$

Using the result in Proposition 5.3.4 (on the Riemannian connection on Riemannian quotient manifolds) and the definition $\mathrm{Hess}\, f[\zeta] = \nabla_\zeta \mathrm{grad}\, f$, we also have

$$\tfrac{1}{2}\overline{\mathrm{Hess}\, f[\zeta]}_Y = \tfrac{1}{2}P_{Y,Y}\mathrm{D}\overline{\mathrm{grad}\, f}\,(Y)\left[\overline{\zeta}_Y\right]. \tag{6.28}$$

Expanding this expression is possible but tedious and leads to a complicated Newton equation. Fortunately, simpler Newton equations can be obtained by exploiting the freedom in (i) the choice of the Riemannian metric $\overline{g}$; (ii) the choice of the horizontal spaces $\mathcal{H}_Y$, which need not be orthogonal to the vertical spaces $\mathcal{V}_Y$ with respect to $\overline{g}$; (iii) the choice of the affine connection ∇, which need not be the Riemannian connection induced by $\overline{g}$.

We first consider an alternative Riemannian metric. We still view the Grassmann manifold $\mathrm{Grass}(p, n)$ as the quotient $\mathbb{R}_*^{n\times p}/\sim$, where the equivalence classes of $\sim$ are the sets of elements of $\mathbb{R}_*^{n\times p}$ that have the same column space. However, instead of (6.26), we consider on $\mathbb{R}_*^{n\times p}$ the metric

$$\overline{g}_Y(Z_1, Z_2) = \mathrm{tr}\left((Y^TBY)^{-1}Z_1^TBZ_2\right), \tag{6.29}$$

where B is the symmetric positive-definite matrix that appears in the definition of f in (6.25). Defining again the horizontal space as the orthogonal complement—with respect to the new inner product (6.29)—of the vertical space

$$\mathcal{V}_Y := T_Y(\pi^{-1}(\pi(Y))) = \{YM : M \in \mathbb{R}^{p\times p}\},$$

we obtain

$$\mathcal{H}_Y = \{Z \in \mathbb{R}^{n\times p} : Y^TBZ = 0\}. \tag{6.30}$$

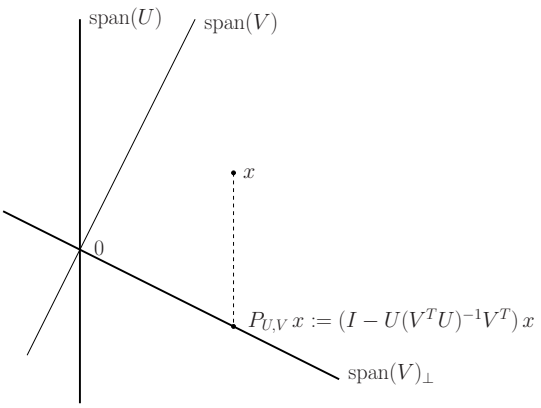

Figure 6.1 The projector $P_{U,V}$.

The orthogonal projection onto $\mathcal{H}_Y$ is given by

$$\mathrm{P}_Y^h = P_{Y,BY} = I - Y(Y^TBY)^{-1}Y^TB.$$

The homogeneity property

$$\overline{\xi}_{YM} = \overline{\xi}_Y M$$

of Proposition 3.6.1 still holds with the new Riemannian metric (6.29). Moreover,

$$\overline{g}_{YM}(\overline{\xi}_{YM}, \overline{\zeta}_{YM}) = \overline{g}_Y(\overline{\xi}_Y, \overline{\zeta}_Y).$$

Therefore, the Grassmann manifold $\mathrm{Grass}(p,n)$ admits a unique Riemannian metric

$$g(\xi,\zeta) := \overline{g}_Y(\overline{\xi}_Y, \overline{\zeta}_Y), \tag{6.31}$$

that makes $(\mathrm{Grass}(p,n), g)$ a Riemannian quotient manifold of $(\mathbb{R}_*^{n\times p}, \overline{g})$ with $\overline{g}$ defined in (6.29).

Before proceeding to obtain formulas for the gradient and the Hessian of f in $(\mathrm{Grass}(p,n), g)$, we first point out some useful properties of the projector

$$P_{U,V} = I - U(V^TU)^{-1}V^T. \tag{6.32}$$

Recall that $P_{U,V}$ is the projector that projects parallel to $\mathrm{span}(U)$ onto $\mathrm{span}(V)$; see Figure 6.1. Therefore, we have the identities

$$P_{U,V}UM = 0 \quad \text{and} \quad P_{U,V}VM = VM.$$

We also have the identity

$$P_{U,V}K = KP_{K^{-1}U,K^TV}.$$

Using the above identities and the technique of Section 3.6.2, we obtain

$$\overline{\mathrm{grad}\, f}_Y = P_{Y,BY}B^{-1}AY = B^{-1}P_{BY,Y}AY. \tag{6.33}$$

It can be checked that the new Riemannian metric (6.29) is horizontally invariant. Consequently, it follows from Proposition 5.3.4 that the Riemannian Hessian is given by

$$\begin{aligned}\overline{\operatorname{Hess} f(\mathcal{Y})[\eta]}_Y &= \overline{\nabla_\eta \operatorname{grad} f}_Y \\ &= \mathrm{P}_Y^h \mathrm{D}\overline{\operatorname{grad} f}\,(Y)\,[\overline{\eta}_Y] \\ &= P_{Y,BY} B^{-1} A \overline{\eta}_Y - P_{Y,BY} \overline{\eta}_Y (Y^T BY)^{-1} Y^T AY \\ &= B^{-1} P_{BY,Y} \left(A\overline{\eta}_Y - B\overline{\eta}_Y (Y^T BY)^{-1} Y^T AY \right). \end{aligned} \tag{6.34}$$

The Newton equation $\nabla_\eta \operatorname{grad} f = -\operatorname{grad} f(x)$ thus yields the equation

$$B^{-1} P_{BY,Y} \left(A\overline{\eta}_Y - B\overline{\eta}_Y (Y^T BY)^{-1} Y^T AY \right) = -B^{-1} P_{BY,Y} AY,$$

or equivalently,

$$P_{BY,Y} \left(A\overline{\eta}_Y - B\overline{\eta}_Y (Y^T BY)^{-1} Y^T AY \right) = -P_{BY,Y} AY.$$

In conclusion, the geometric Newton method in Algorithm 4, for the Rayleigh quotient cost function (6.25), with the affine connection ∇ chosen as the Riemannian connection on $\operatorname{Grass}(p, n)$ seen as the Riemannian quotient manifold of $(\mathbb{R}_*^{n\times p}, \overline{g})$ with $\overline{g}$ defined in (6.29), and with the retraction R chosen as (4.40), yields the matrix algorithm displayed in Algorithm 8. The notation $Z_k = \overline{\eta}_{Y_k}$ is used so that the algorithm resembles contemporary algorithms from the field of numerical linear algebra.

Algorithm 8 Riemannian Newton method for the Rayleigh quotient on $\operatorname{Grass}(p, n)$

Require: Symmetric matrix A, symmetric positive-definite matrix B.
Input: Initial iterate $Y_0 \in \mathbb{R}_*^{n\times p}$.
Output: Sequence of iterates $\{Y_k\}$ in $\mathbb{R}_*^{n\times p}$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Solve the linear system

$$\begin{cases} P_{BY_k,Y_k} \left(AZ_k - BZ_k (Y_k^T BY_k)^{-1} Y_k^T AY_k \right) = -P_{BY_k,Y_k} (AY_k) \\ Y_k^T BZ_k = 0 \end{cases} \tag{6.35}$$

for the unknown Z_k, where $P_{BY,Y} = I - BY(Y^T BY)^{-1} Y^T$. (The condition $Y_k^T BZ_k$ expresses that Z_k belongs to the horizontal space $\mathcal{H}_{Y_k}$ (6.30).)
3: Set

$$Y_{k+1} = (Y_k + Z_k) N_k$$

where N_k is a nonsingular $p \times p$ matrix chosen for normalization purposes.
4: **end for**

Algorithm 8 is related to several eigenvalues methods; see Notes and References.

A concern with the Newton equation (6.35) is that the domain $\{Z \in \mathbb{R}^{n\times p} : Y^TBZ = 0\}$ of the map

$$F : Z \mapsto P_{BY,Y}\left(AZ - BZ(Y^TBY)^{-1}Y^TAY\right)$$

differs from its range $\{Z \in \mathbb{R}^{n\times p} : Y^TZ = 0\}$. Hence, powers of F cannot be formed, and linear equation solvers based on Krylov subspaces cannot be applied directly to (6.35). A remedy based on preconditioners is discussed in Section 6.5.2. Another remedy is to exploit the freedom in the choice of the affine connection ∇, which, according to Algorithm 5, need not be the Riemannian connection. To this end, let us view $\mathrm{Grass}(p, n)$ as a Riemannian quotient manifold of $(\mathbb{R}^{n\times p}_*, \overline{g})$ with

$$\overline{g}_Y(Z_1, Z_2) = \mathrm{tr}((Y^TBY)^{-1}Z_1^TZ_2). \tag{6.36}$$

Note that this Riemannian metric is different from the canonical Riemannian metric (6.26). The horizontal space defined as the orthogonal complement of the vertical space is still given by (6.21), but the expression of the gradient becomes

$$\tfrac{1}{2}\overline{\mathrm{grad}\, f}_Y = P_{BY,Y}AY, \tag{6.37}$$

which is simpler than (6.27). Now, instead of choosing the affine connection ∇ as the Riemannian connection, we define ∇ by

$$\left(\overline{\nabla_\eta \xi}\right)_Y = P_{BY,Y}\mathrm{D}\overline{\xi}\,(Y)\,[\overline{\eta}_Y]\,. \tag{6.38}$$

It is readily checked that (6.38) defines a horizontal lift, i.e., $\left(\overline{\nabla_\eta \xi}\right)_{YM} = \left(\overline{\nabla_\eta \xi}\right)_Y M$, and that ∇ is indeed an affine connection (see Section 5.2). With this affine connection, the horizontal lift of the Newton equation $\nabla_\eta \mathrm{grad}\, f = -\mathrm{grad}\, f(\mathcal{Y})$ reads

$$P_{BY,Y}\left(AZ - BZ(Y^TBY)^{-1}Y^TAY\right) = P_{BY,Y}AY, \quad Y^TZ = 0, \tag{6.39}$$

where Z stands for $\overline{\eta}_Y$. Observe that the map

$$Z \mapsto P_{BY,Y}\left(AZ - BZ(Y^TBY)^{-1}Y^TAY\right)$$

involved in (6.39) is now from $\{Z \in \mathbb{R}^{n\times p}_* : Y^TZ = 0\}$ into itself. The resulting iteration is still guaranteed by Theorem 6.3.2 to converge locally at least quadratically to the spectral invariant subspaces of $B^{-1}A$ (see Section 6.5.1 for details).

Note that in this section we have always chosen the horizontal space as the orthogonal complement of the vertical space. The possibility of choosing other horizontal spaces is exploited in Section 7.5.3.

6.4.4 The nonsymmetric eigenvalue problem

The Rayleigh quotient

$$f : \mathrm{Grass}(p, n) \to \mathbb{R} : \mathrm{span}(Y) \mapsto \mathrm{tr}\left((Y^TY)^{-1}Y^TAY\right)$$

depends only on the symmetric part of A; it is thus clear that when A is nonsymmetric, computing critical points of f in general does not produce invariant subspaces of A. A way to tackle the nonsymmetric eigenvalue

problem is to consider instead the tangent vector field on the Grassmann manifold defined by

$$\overline{\xi}_Y := \mathrm{P}_Y^h AY, \tag{6.40}$$

where P_Y^h denotes the projector (6.22) onto the horizontal space (6.21). This expression is homogeneous ($\overline{\xi}_{YM} = \overline{\xi}_Y M$) and horizontal; therefore, as a consequence of Proposition 3.6.1, it is a well-defined horizontal lift and defines a tangent vector field given by $\xi_Y = \mathrm{D}\pi(Y)\left[\overline{\xi}_Y\right]$ on the Grassmann manifold. Moreover, $\xi_{\mathcal{Y}} = 0$ if and only if $\mathcal{Y}$ is an invariant subspace of A. Obtaining the Newton equation (6.1) for ξ defined in (6.40) is straightforward: formula (6.23), giving the horizontal lift of the connection, leads to

$$\overline{\nabla_\eta \xi}_Y = \mathrm{P}_Y^h \left(A\overline{\eta}_Y - \overline{\eta}_Y (Y^T Y)^{-1} Y^T AY\right)$$

and the Newton equation (6.1) reads

$$\begin{cases} \mathrm{P}_Y^h \left(A\overline{\eta}_Y - \overline{\eta}_Y (Y^T Y)^{-1} Y^T AY\right) = -\mathrm{P}_Y^h AY, \\ Y^T \overline{\eta}_Y = 0, \end{cases} \tag{6.41}$$

where the second equation expresses that $\overline{\eta}_Y$ is in the horizontal space. The resulting Newton iteration turns out to be identical to Algorithm 7, except that A is no longer required to be symmetric.

6.4.5 Newton with subspace acceleration: Jacobi-Davidson

The Jacobi-Davidson approach is a powerful technique for solving a variety of eigenproblems. It has recently become widely popular among chemists and solid-state physicists for computing a few extreme eigenpairs of large-scale eigenvalue problems. In this section, the principles of the Jacobi-Davidson approach are briefly reviewed and the method is interpreted as a Rayleigh-based Riemannian Newton method within a sequential subspace optimization scheme.

For simplicity we focus on the standard eigenvalue problem and let A be a symmetric $n \times n$ matrix. Central to the Jacobi-Davidson approach is the *Jacobi correction equation*

$$(I - x_k x_k^T)(A - \tau_k I)(I - x_k x_k^T) s_k = -(A - \tau_k I) x_k, \quad x_k^T s_k = 0, \tag{6.42}$$

which, for the usual choice of shift $\tau_k = x_k^T A x_k$, reduces to the Newton equation for the Rayleigh quotient on the sphere (see Algorithm 6).

In the Riemannian Newton method the update vector s_k is retracted onto the manifold to produce the next iterate $x_{k+1} = R_{x_k} s_k$; for example, the choice (6.16) of the retraction R yields $x_{k+1} = (x_k + s_k)/\|x_k + s_k\|$. Instead, in the Jacobi-Davidson approach, the update vector is used to expand a low-dimensional search space on which the given eigenproblem is projected. This is the standard Rayleigh-Ritz procedure that underlies all Davidson-like methods, as well as the Lanczos and Arnoldi methods. The small projected problem is solved by standard techniques, and this leads to approximations

Algorithm 9 Jacobi-Davidson

Require: Symmetric matrix A.

Input: Select a set of k_0 (≥ 1) orthonomal vectors $v_1, \ldots, v_{k_0}$ and set $V_1 = [v_1|\ldots|v_{k_0}]$.

Output: Sequence of iterates $\{x_k\}$.

1: **for** $k = 1, 2, \ldots$ **do**
2: Compute the interaction matrix $H_k = V_k^T A V_k$.
3: Compute the leftmost eigenpair (ρ_k, y_k) of H_k (with $\|y_k\| = 1$).
4: Compute the Ritz vector $x_k = V_k y_k$.
5: If needed, shrink the search space: compute the $j_{\min}$ leftmost eigenpairs $(\rho_k^{(j)}, y_k^{(j)})$ of H_k and reset $V_k := V_k[y_k^{(1)}|\cdots|y_k^{(j_{\min})}]$.
6: Obtain s_k by solving (approximately) the Jacobi equation (6.42).
7: Orthonormalize $[V_k|s_k]$ into V_{k+1}.
8: **end for**

for the wanted eigenvector and eigenvalues of the given large problem. The procedure is described in Algorithm 9 for the case where the leftmost eigenpair of A is sought.

Practical implementations of Algorithm 9 vary widely depending on the methods utilized to solve the Jacobi equation approximately and to reduce the search space.

Concerning the solution of the Jacobi equation, anticipating the development in Chapter 7, we point out that the solution s_k of (6.42) is the critical point of the model

$$\widehat{m}_{x_k}(s) := x_k^T A x_k + 2 s^T A x_k + s^T (A - x_k^T A x_k I) s, \quad x_k^T s = 0.$$

This model is the quadratic Taylor expansion of the cost function

$$f \circ R_{x_k} : T_{x_k} S^{n-1} \to \mathbb{R} : s \mapsto \frac{(x_k + s)^T A (x_k + s)}{(x_k + s)^T (x_k + s)}$$

around the origin 0 of the Euclidean space $T_{x_k} S^{n-1}$, where f denotes the Rayleigh quotient on the sphere and R denotes the retraction (6.16). When the goal of the algorithm is to minimize the Rayleigh quotient (in order to find the leftmost eigenpair), the idea of solving the Jacobi equation, which amounts to computing the critical point s_* of the model $\widehat{m}_{x_k}(s)$, presents two drawbacks: (i) the critical point is not necessarily a minimizer of the model, it may be a saddle point or a maximizer; (ii) even when the critical point is a minimizer, it may be so far away from the origin of the Taylor expansion that there is an important mismatch between the model $\widehat{m}_{x_k}(s_*)$ and the cost function $f \circ R_{x_k}(s_*)$. The trust-region approach presented in Chapter 7 remedies these drawbacks by selecting the update vector s_k as an approximate minimizer of the model $\widehat{m}_{x_k}$, constrainted to a region around $s = 0$ where its accuracy is trusted. Therefore, algorithms for approximately solving trust-region subproblems (see Section 7.3) can be fruitfully used as "intelligent" approximate solvers for the Jacobi equation that are aware of the underlying Rayleigh quotient optimization problem.

Concerning the sequential subspace approach, if the sequence of computed s_k's is gradient-related, then the Jacobi-Davidson method fits within the framework of Algorithm 1 (an accelerated line search) and the convergence analysis of Section 4.3 applies. In particular, it follows from Theorem 4.3.1 that every accumulation point of the sequence $\{x_k\}$ is a critical point of the Rayleigh quotient, and thus an eigenvector of A. A simple way to guarantee that $\{x_k\}$ stems from a gradient-related sequence is to include $\operatorname{grad} f(x_k) = Ax_k - x_k x_k^T Ax_k$ as a column of the new basis matrix V_{k+1}.

6.5 ANALYSIS OF RAYLEIGH QUOTIENT ALGORITHMS

In this section, we first formally prove quadratic convergence of the Newton algorithms for the Rayleigh quotient developed in the previous section. After this, the remainder of the section is devoted to a discussion of the numerical implementation of the proposed algorithms. Efficiently solving the Newton equations is an important step in generating numerically tractable algorithms. The structured matrix representation of the Newton equations that result from the approach taken in this book means that we can exploit the latest tools from numerical linear algebra to analyze and solve these equations.

6.5.1 Convergence analysis

For the convergence analysis, we focus on the case of Algorithm 8 (iteration on the Grassmann manifold for the generalized eigenvalue problem). The convergence analysis of Algorithm 7 (standard eigenvalue problem, on the Grassmann manifold) follows by setting $B = I$. These results also apply to Algorithm 6 (on the sphere) since it fits in the framework of Algorithm 7.

Since Algorithm 8 is a particular instance of the general geometric Newton method (Algorithm 4), the convergence analysis in Theorem 6.3.2 applies to Algorithm 8. A p-dimensional subspace $\operatorname{span}(Y_*)$ is a critical point of the Rayleigh quotient (6.25) if and only if $\operatorname{span}(Y_*)$ is an invariant subspace of the pencil (A, B). The condition in Theorem 6.3.2 that the Jacobian (here, the Hessian of f) at $\operatorname{span}(Y_*)$ be invertible becomes the condition that the Hessian operator

$$Z \in \mathcal{H}_{Y_*} \;\mapsto\; B^{-1}P_{BY_*,Y_*}\left(AZ - BZ(Y_*^T BY_*)^{-1}Y_*^T AY_*\right) \;\in \mathcal{H}_{Y_*}$$

given in (6.34) be invertible. It can be shown that this happens if and only if the invariant subspace $\operatorname{span}(Y_*)$ is *spectral*, i.e., for every eigenvalue λ of $B^{-1}A|_{\operatorname{span}(Y_*)}$ the multiplicities of λ as an eigenvalue of $B^{-1}A|_{\operatorname{span}(Y_*)}$ and as an eigenvalue of $B^{-1}A$ are identical. (To prove this, one chooses a basis where B is the identity and A is diagonal with the eigenvalues of $B^{-1}A|_{\operatorname{span}(Y_*)}$ in the upper left block. The operator reduced to $\mathcal{H}_{Y_*}$ turns out to be diagonal with all diagonal elements different from zero.)

Theorem 6.5.1 (local convergence of Algorithm 8) *Under the requirements of Algorithm 8, assume that there is a* $Y_* \in \mathbb{R}^{n\times p}$ *such that* $\operatorname{span}(Y_*)$ *is a spectral invariant subspace of* $B^{-1}A$. *Then there exists a neighborhood* $\mathcal{U}$ *of* $\operatorname{span}(Y_*)$ *in* $\operatorname{Grass}(p,n)$ *such that, for all* $Y_0 \in \mathbb{R}^{n\times p}$ *with* $\operatorname{span}(Y_0) \in \mathcal{U}$, *Algorithm 8 generates an infinite sequence* $\{Y_k\}$ *such that* $\{\operatorname{span}(Y_k)\}$ *converges superlinearly (at least quadratically) to* $\mathcal{Y}_*$ *on* $\operatorname{Grass}(p,n)$.

Concerning the algorithm for the nonsymmetric eigenvalue problem presented in Section 6.4.4, it follows by a similar argument that the iterates of the method converge locally superlinearly to the spectral invariant subspaces of A.

6.5.2 Numerical implementation

A crucial step in a numerical implementation of the Newton algorithms lies in solving the Newton equations. We first consider the Grassmann case with $B = I$ (standard eigenvalue problem). For clarity, we drop the subscript k. The Newton equation (6.24) reads

$$\begin{cases} \mathrm{P}_Y^h \left(AZ - Z(Y^TY)^{-1}Y^TAY\right) = -\mathrm{P}_Y^h(AY) \\ Y^TZ = 0, \end{cases} \tag{6.43}$$

where Z is the unknown and $P_Y^h = (I - Y(Y^TY)^{-1}Y^T)$. In order to make this equation simpler, the first thing to do is to choose Y orthonormal, so that $Y^TY = I$. Since Y^TAY is symmetric, it is possible to further choose Y such that Y^TAY is diagonal. This can be done by computing a matrix M such that $(YM)^TAYM \equiv M^T(Y^TAY)M$ is diagonal and making $Y \leftarrow YM$. This corresponds to solving a small-scale $p \times p$ eigenvalue problem. The diagonal elements $\rho_1, \ldots, \rho_p$ of the diagonal matrix Y^TAY are called the *Ritz values* related to $(A, \operatorname{span}(Y))$, and the columns of Y are the corresponding *Ritz vectors*. This decouples (6.43) into p independent systems of linear equations of the form

$$\begin{cases} \mathrm{P}_Y^h(A - \rho_i I)\mathrm{P}_Y^h z_i = -\mathrm{P}_Y^h A y_i, \\ Y^Tz_i = 0, \end{cases} \tag{6.44}$$

where $z_1, \ldots, z_p \in \mathbb{R}^p$ are the columns of Z. Note that (6.44) resembles a parallel implementation of p Newton methods (6.17). However, the projection operator P_Y^h in (6.44) is equal to $(I - Y(Y^TY)^{-1}Y^T)$, whereas the parallel implementation of (6.17) would lead to

$$\begin{cases} \mathrm{P}_{y_i}(A - \rho_i I)\mathrm{P}_{y_i} z_i = -\mathrm{P}_{y_i} A y_i, \\ y_i^Tz_i = 0, \end{cases}$$

$i = 1, \ldots, p$, where $\mathrm{P}_{y_i} = (I - y_i(y_i^Ty_i)^{-1}y_i^T)$. Methods for solving (6.44) include Krylov-based methods that naturally enforce the constraint Y^Tz_i. Another approach is to transform (6.44) into the saddle-point problem

$$\begin{bmatrix} A - \rho_i I & Y \\ Y^T & 0 \end{bmatrix} \begin{bmatrix} z_i \\ \ell \end{bmatrix} = \begin{bmatrix} -(A - f(y_i)I)y_i \\ 0 \end{bmatrix}, \tag{6.45}$$

a structured linear system for which several algorithms have been proposed in the literature.

We now look specifically into the $p = 1$ case, in the form of the Newton equation (6.17) on the sphere, repeated here for convenience:

$$\begin{cases} \mathrm{P}_x A \mathrm{P}_x \eta - \eta x^T A x = -\mathrm{P}_x A x, \\ x^T \eta = 0, \end{cases}$$

where $P_x = (I - xx^T)$. The Newton equation (6.17) is a system of linear equations in the unknown $\eta_k \in \mathbb{R}^n$. It admits several equivalent formulations. For example, using the fact that $\mathrm{P}_x x = 0$, (6.17) can be rewritten as

$$\mathrm{P}_x (A - f(x)I)(x + \eta) = 0, \quad x^T \eta = 0,$$

where $f(x)$ still stands for the Rayleigh quotient (6.15). Equation (6.17) is also equivalent to the *saddle-point problem*

$$\begin{bmatrix} A - f(x)I & x \\ x^T & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \ell \end{bmatrix} = \begin{bmatrix} -(A - f(x)I)y \\ 0 \end{bmatrix}. \tag{6.46}$$

If $A - f(x)I$ is nonsingular, then the solution η of (6.17) is given explicitly by

$$\eta = -x + (A - f(x)I)^{-1} x \frac{1}{x^T (A - f(x)I)^{-1} x}. \tag{6.47}$$

This points to an interesting link with the Rayleigh quotient iteration: with the retraction defined in (6.16), the next iterate constructed by Algorithm 6 is given by

$$\frac{x + \eta}{\|x + \eta\|} = \frac{(A - f(x)I)^{-1} x}{\|(A - f(x)I)^{-1} x\|},$$

which is the formula defining the Rayleigh quotient iteration.

With $U \in \mathbb{R}^{n \times (n-1)}$ chosen such that $[x|U]^T [x|U] = I$, (6.17) is also equivalent to

$$(U^T A U - f(x)I)s = -U^T A x, \quad \eta = Us, \tag{6.48}$$

which is a linear system in classical form in the unknown s. Note that when A is a large sparse matrix, the matrix U is large and dense, and there may not be enough memory space to store it. Moreover, the sparsity of A is in general not preserved in the reduced matrix $U^T AU$. The approach (6.48) should thus be avoided in the large sparse case. It is preferable to solve the system in the form (6.17) or (6.46) using an iterative method.

We now briefly discuss the generalized case $B \neq I$. The Newton equation (6.35) can be decoupled into p independent equations of the form

$$P_{BY,Y}(A - \rho_i I) P_{Y,BY} z_i = -P_{BY,Y} A y_i, \; Y^T B z_i = 0. \tag{6.49}$$

The corresponding saddle-point formulation is

$$\begin{bmatrix} A - \rho_i B & BY \\ (BY)^T & 0 \end{bmatrix} \begin{bmatrix} z_i \\ \ell \end{bmatrix} = \begin{bmatrix} -(A - \rho_i B) y_i \\ 0 \end{bmatrix}, \tag{6.50}$$

where $\rho_i := y_i^T A y_i / (y_i^T B y_i)$. For the purpose of applying a Krylov-like iterative method, the Newton equations (6.49) present the difficulty that the operator on the left-hand side sends $z_i \in (\mathrm{span}(BY))^\perp$ to the subspace $(\mathrm{span}(Y))^\perp$. A remedy is to solve the equivalent equation obtained by applying the projector $P_{BY,BY}$ to (6.49).

This trick is a particular instance of a more general technique called *preconditioning*. Assume that we have a matrix K that is a reasonably good approximation of $(A - \rho_i B)$ and such that the operator K^{-1} is easily available (in other words, systems $Kx = b$ are easy to solve). Let the superscript † denote the pseudo-inverse. Then (6.49) is equivalent to

$$\begin{aligned}(P_{BY,Y} K P_{Y,BY})^\dagger \, P_{BY,Y}(A - \rho_i I) P_{Y,BY} z_i \\ = -(P_{BY,Y} K P_{Y,BY})^\dagger \, P_{BY,Y} A y_i, \quad Y^T B z_i = 0. \quad (6.51)\end{aligned}$$

The advantage is that the operator acting on z_i on the left-hand side of (6.51) is close to the identity, which improves the speed of convergence of Krylov-based iterative solvers. In practice, applying this operator is made possible by the Olsen formula

$$\left(P_{\widetilde{Q},Q} K P_{U,\widetilde{U}}\right)^\dagger = P_{U,U} P_{K^{-1}\widetilde{Q},\widetilde{U}} K^{-1} P_{Q,Q} = P_{U,U} K^{-1} P_{\widetilde{Q},K^{-T}\widetilde{U}} P_{Q,Q}$$

since we have assumed that the operator K^{-1} is easily available.

6.6 NOTES AND REFERENCES

The history of Newton's method on manifolds can be traced back to Luenberger [Lue72], if not earlier. Gabay [Gab82] proposed a Newton method on embedded submanifolds of $\mathbb{R}^n$. Smith [Smi93, Smi94] and Udrişte [Udr94] formulated the method on general Riemannian manifolds, and Smith [Smi93, Smi94] provided a proof of quadratic convergence. Mahony's thesis [Mah94, Mah96, MM02] develops a Newton method on Lie groups and homogeneous spaces. Related work includes [Shu86, EAS98, OW00, Man02, MM02, ADM+02, FS02, DPM03, HT04, ABM06].

Smith [Smi93, Smi94] proposes a geometric Newton method that seeks a zero of a one-form (instead of a vector field). The underlying idea is that affine connections can be extended to general tensors, and in particular to one-forms. This approach makes it possible to define a Newton method that seeks a critical point of a cost function defined on a manifold equipped with an affine connection and a retraction (cf. the requirements of Algorithm 4) but not necessarily with a Riemannian metric since the Riemannian metric is no longer needed to define the gradient vector field. (Smith nevertheless requires the manifold to be Riemannian, notably because his algorithms use the Riemannian exponential to perform the update.)

Our convergence analysis (Theorem 6.3.2) of the geometric Newton method, which was built from the $\mathbb{R}^n$ proof given by Dennis and Schnabel [DS83, Th. 5.2.1] and is strongly based on coordinates. A coordinate-free

approach can be found in Smith [Smi93, Smi94] for the case of a real-valued function on a Riemannian manifold with its Riemannian connection and exponential retraction; this elegant proof exploits bounds on the second and third covariant derivatives of the cost function and on the curvature of the manifold in a neighborhood of the critical point.

The calculus-based local analysis in Section 6.3.1 was inspired by the work of Hüper; see, e.g., [HH00, Hüp02, HT04].

In general, it cannot be guaranteed that the Jacobian $J(x_k)$ in Newton's method (Algorithm 4) is nonsingular. In other words, the Newton equation (6.1) may not admit one and only one solution. Even if it does, the Jacobian operator may be poorly conditioned, so that the linear system (6.1) cannot be reliably solved. If this happens while x_k is far away from the solution, a possible remedy is to fall back to a first-order, steepest-descent-like method. Several other remedies exist that pertain to globally convergent modifications of Newton's method; see, e.g., Dennis and Schnabel [DS83] and Nocedal and Wright [NW99].

Several ways to combine Newton and line-search approaches are discussed in Dennis and Schnabel [DS83, Ch. 6]. For more information on positive-definite modifications of the Hessian, see Nocedal and Wright [NW99, §6.3].

Theorem 6.3.2 states that the sequences $\{x_k\}$ generated by Algorithm 4 (the geometric Newton method) converge to any nondegenerate zero x_* of the vector field whenever the initial point x_0 belongs to *some* neighborhood of x_*; however, Theorem 6.3.2 is silent about the *size* of this neighborhood. For Newton's method in $\mathbb{R}^n$ applied to finding a zero of a function F, Kantorovich's theorem [Kan52] (or see [Den71, DS83]) states that if the product of a Lipschitz constant for the Jacobian times a bound on the inverse of the Jacobian at x_0 times a bound on the first Newton vector is smaller than $\frac{1}{2}$, then the function F has a unique zero x_* in a ball around x_0 larger than a certain bound and the iterates of Newton's method converge to x_*. This is a very powerful result, although in applications it is often difficult to ensure that the Kantorovich condition holds. Kantorovich's theorem was generalized to the Riemannian Newton method by Ferreira and Svaiter [FS02]. Another way to obtain information about the basins of attraction for Newton's method is to use Smale's γ and α theorems, which were generalized to the Riemannian Newton method by Dedieu *et al.* [DPM03].

For the application to the computation of invariant subspaces of a matrix, Theorem 6.5.1 states that the sequences $\{\mathrm{span}(Y_k)\}$ produced by Algorithm 7 converge locally to any p-dimensional spectral invariant subspace $\mathcal{V}$ of A provided that the initial point is in a basin of attraction that contains an open ball around $\mathcal{V}$ in the Grassmann manifold, but it does not give any information about the size of the basin of attraction. This is an important issue since a large basin of attraction means that the iteration converges to the target invariant subspace even if the initial estimate is quite imprecise. It

has been shown for previously available methods that the basins of attraction are prone to deteriorate when some eigenvalues are clustered. Batterson and Smillie [BS89] have drawn the basins of attraction of the Rayleigh quotient iteration (RQI) for $n = 3$ and have shown that they deteriorate when two eigenvalues are clustered. The bounds involved in the convergence results of the methods analyzed by Demmel [Dem87] blow up when the external gap vanishes. It was shown in [ASVM04], analytically and numerically, that the Riemannian Newton method applied to the Rayleigh quotient on the Grassmann manifold suffers from a similar dependence on the eigenvalue gap. It was also shown how this drawback can be remedied by considering a Levenberg-Marquardt-like modification of the Newton algorithm. The modified algorithm depends on a real parameter whose extreme values yield the Newton method on the one hand, and the steepest-descent method for the cost function $\|\xi\|$ with ξ defined in (6.40) on the other hand. A specific choice for this parameter was proposed that significantly improves the size of the basins of attraction around each invariant subspace.

The formula for the Hessian of the Brockett cost function (4.32) on the Stiefel manifold is straightforward using formula (5.15) for the Riemannian connection on Riemannian submanifolds of Euclidean spaces. The resulting Newton equation, however, is significantly more complex than the Newton equation (6.24) for the Rayleigh quotient on the Grassmann manifold. This outcome is due to the fact that the projection (3.35) onto the tangent space to the Stiefel manifold has one more term than the projection (3.41) onto the horizontal space of the Grassmann manifold (viewed as a quotient of the noncompact Stiefel manifold). The extra complexity introduced into the Newton equation significantly complicates the evaluation of each iterate and does not significantly add to the performance of the method since the Grassmann Newton method identifies an invariant p-dimensional subspace and the numerical cost of identifying the Ritz vectors of this subspace is a negligible additional cost on top of the subspace problem.

The term "*spectral* invariant subspace" is used by Rodman *et al.* [GLR86, RR02]; Stewart [Ste01] uses the term "*simple* invariant subspace". The material in Section 6.4.4 comes from [AMS04]. There is a vast literature on saddle-point problems such as (6.46); see Benzi *et al.* [BGL05] for a survey. For the numerical computation of U in (6.48), we refer the reader to [NW99, §16.2], for example. Saad [Saa96] is an excellent reference on iterative solvers for linear systems of equations. Practical implementation issues for Newton methods applied to the Rayleigh quotient are further discussed in Absil *et al.* [ASVM04].

Several methods proposed in the literature are closely related to the eigenvalue algorithms proposed in this chapter. These methods differ on three points: (i) the matrix BY in the structured matrix involved in the Newton equation (6.50). (ii) the shifts ρ_i. (iii) the way the z_i's are used to compute the new iterate.

The modified block newton method proposed by Lösche *et al.* [LST98] corresponds to Algorithm 8 with $B = I$. The authors utilize formula (6.50) and prove quadratic convergence. In fact, the order of convergence is even cubic [AMS04].

The Newton method discussed by Edelman *et al.* [EAS98, p. 344] corresponds to Algorithm 5 applied to the Rayleigh quotient (6.19) on the Grassmann manifold with ∇ chosen as the Riemannian connection and R chosen as the exponential mapping.

Smith [Smi93, Smi94] mentions Algorithm 6 but focuses on the version where the retraction R is the exponential mapping.

The shifted Tracemin algorithm of Sameh and Wisniewski [SW82, ST00] can be viewed as a modification of Algorithm 8 where the shifts ρ_i in (6.49)—or equivalently (6.50)—are selected using a particular strategy. The simple (unshifted) version of Tracemin corresponds to $\rho_i = 0$. This algorithm is mathematically equivalent to a direct subspace iteration with matrix $A^{-1}B$. The choice $\rho_i = 0$ is further discussed and exploited in the context of a trust-region method with an adaptive model in [ABGS05].

Equation (6.47) corresponds to equation (2.10) in Sameh and Tong [ST00] and to algorithm 2 in Lundström and Elden [LE02, p. 825] to some extent.

Relations between the RQI and various Newton-based approaches are mentioned in several references, e.g., [PW79, Shu86, Smi94, ADM+02, MA03]. This equivalence still holds when certain Galerkin techniques are used to approximately solve the Newton and RQI equations [SE02]. A block generalization of the RQI is proposed in [Smi97, AMSV02]. The connection with the Newton method does not hold for the block version [AMSV02].

The method proposed by Fattebert [Fat98] is connected with (6.50). The idea is to replace BY in (6.50) by thinner matrices where some columns that are not essential for the well conditioning of the linear system are omitted. This approach is thus midway between that of the Newton method and the RQI.

As discussed in [EAS98, AMSV02], the Newton method proposed by Chatelin [Cha84, Dem87] corresponds to performing a classical Newton method in a fixed coordinate chart of the Grassmann manifold. In contrast, the algorithms proposed in this chapter can be viewed as using an adaptive coordinate chart, notably because the retraction used, for example, in Step 3 of Algorithm 6 depends on the current iterate x_k.

For more information on the Jacobi-Davidson approach and related generalized Davidson methods, see Sleijpen *et al.* [SVdV96, SvdVM98], Morgan and Scott [MS86], Stathopoulos *et al.* [SS98, Sta05, SM06], Notay [Not02, Not03, Not05], van den Eshof [vdE02], Brandts [Bra03], and references therein. Methods for (approximately) solving the Jacobi (i.e., Newton) equation can also be found in these references. The Newton algorithm on the sphere for the Rayleigh quotient (Algorithm 6) is very similar to the simplified Jacobi-Davidson algorithm given in [Not02]; see the discussion in [ABG06b]. The Newton equations (6.24), (6.35), and (6.39) can be thought of as block versions of particular instances of the Jacobi correction

equations found in [SBFvdV96, HS03]. References related to sequential subspace optimization include Hager [Hag01, HP05], Absil and Gallivan [AG05], and Narkiss and Zibulevsky [NZ05].