Chapter Eight

A Constellation of Superlinear Algorithms

The Newton method (Algorithm 5 in Chapter 6) applied to the gradient of a real-valued cost is the archetypal superlinear optimization method. The Newton method, however, suffers from a lack of global convergence and the prohibitive numerical cost of solving the Newton equation (6.2) necessary for each iteration. The trust-region approach, presented in Chapter 7, provides a sound framework for addressing these shortcomings and is a good choice for a generic optimization algorithm. Trust-region methods, however, are algorithmically complex and may not perform ideally on all problems. A host of other algorithms have been developed that provide lower-cost numerical iterations and stronger global convergence properties than the Newton iteration while still approximating the second-order properties of the Newton algorithm sufficiently well to obtain superlinear local convergence. The purpose of this chapter is to briefly review some of these techniques and show how they can be generalized to manifolds. These techniques admit so many variations that we have no pretention of being exhaustive. Most available optimization schemes in $\mathbb{R}^n$ have never been formulated on abstract manifolds. Considering each algorithm in detail is beyond the scope of this book. We will instead focus on resolving a common issue underlying most of these algorithms—approximating derivatives by finite differences on manifolds. To this end, we introduce the concept of vector transport, which relaxes the computational requirements of parallel translation in very much the same way as the concept of retraction relaxes the computational requirements of exponential mapping. Vector transport is a basic ingredient in generalizing the class of finite-difference and conjugate-gradient algorithms on manifolds.

We conclude the chapter by considering the problem of determining a solution, or more generally a least-squares solution, of a system of equations $F(x) = 0$, where F is a function on a manifold into $\mathbb{R}^n$. Although this problem is readily rewritten as the minimization of the squared norm of F, its particular structure lends itself to specific developments.

8.1 VECTOR TRANSPORT

In Chapter 4, on first-order algorithms, the notion of retraction was introduced as a general way to take a step in the direction of a tangent vector. (The tangent vector was, typically, the steepest-descent direction for the cost function.) In second-order algorithms, when the second-order information is not readily available through a closed-form Jacobian or Hessian, it will be necessary to approximate second derivatives by "comparing" first-order information (tangent vectors) at distinct points on the manifold. The notion of *vector transport* $\mathcal{T}$ on a manifold $\mathcal{M}$, roughly speaking, specifies how to transport a tangent vector ξ from a point $x \in \mathcal{M}$ to a point $R_x(\eta) \in \mathcal{M}$.

Vector transport, as defined below, is not a standard concept of differential geometry. (Neither is the notion of retraction.) However, as we will see, it is closely related to the classical concept of parallel translation. The reason for considering the more general notion of vector transport is similar to the reason for considering general retractions rather than the specific exponential mapping. Parallel translation along geodesics is a vector transport that is associated with any affine connection in a natural way. Conceptually appealing (like the exponential mapping), it can, however, be computationally demanding or cumbersome in numerical algorithms. Another vector transport may reduce (in some cases dramatically) the computational effort while retaining the convergence properties of the algorithm.

Let $T\mathcal{M} \oplus T\mathcal{M}$ denote the set

$$T\mathcal{M} \oplus T\mathcal{M} = \{(\eta_x, \xi_x) : \eta_x, \xi_x \in T_x\mathcal{M},\ x \in \mathcal{M}\}.$$

This set admits a natural manifold structure for which the mappings

$$(\eta_x, \xi_x) \in T\mathcal{M} \oplus T\mathcal{M} \mapsto (\varphi_1(x), \ldots, \varphi_d(x), \eta_x\varphi_1, \ldots, \eta_x\varphi_d, \xi_x\varphi_1, \ldots, \xi_x\varphi_d)$$

are charts whenever φ is a chart of the manifold $\mathcal{M}$. The operation $\oplus$ is called the *Whitney sum*.

We refer to Figure 8.1 for an illustation of the following definition.

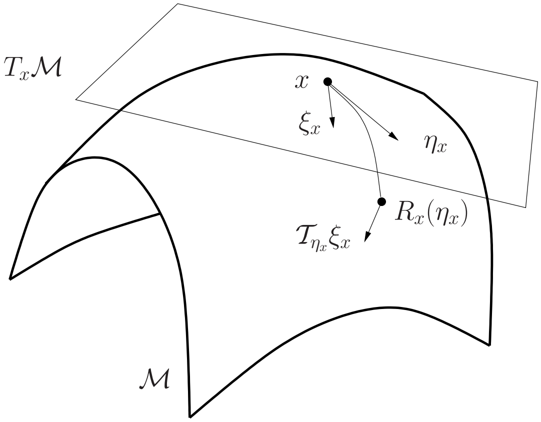

Figure 8.1 Vector transport.

Definition 8.1.1 (vector transport) *A* vector transport *on a manifold $\mathcal{M}$ is a smooth mapping*

$$T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x) \in T\mathcal{M}$$

satisfying the following properties for all $x \in \mathcal{M}$:

(i) (Associated retraction) There exists a retraction R, called the retraction associated with $\mathcal{T}$, *such that the following diagram commutes*

$$\begin{tikzcd}
(\eta_x, \xi_x) \arrow[r, "\mathcal{T}"] \arrow[d] & \mathcal{T}_{\eta_x}(\xi_x) \arrow[d, "\pi"] \\
\eta_x \arrow[r, "R"'] & \pi\left(\mathcal{T}_{\eta_x}(\xi_x)\right)
\end{tikzcd}$$

where $\pi\left(\mathcal{T}_{\eta_x}(\xi_x)\right)$ denotes the foot of the tangent vector $\mathcal{T}_{\eta_x}(\xi_x)$.

(ii) (Consistency) $\mathcal{T}_{0_x}\xi_x = \xi_x$ for all $\xi_x \in T_x\mathcal{M}$;

(iii) (Linearity) $\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}(\xi_x) + b\mathcal{T}_{\eta_x}(\zeta_x)$.

The first point in Definition 8.1.1 means that $\mathcal{T}_{\eta_x}\xi_x$ is a tangent vector in $T_{R_x(\eta_x)}\mathcal{M}$, where R is the retraction associated with $\mathcal{T}$. When it exists, $(\mathcal{T}_{\eta_x})^{-1}(\xi_{R_x(\eta_x)})$ belongs to $T_x\mathcal{M}$. If η and ξ are two vector fields on $\mathcal{M}$, then $(\mathcal{T}_\eta)^{-1}\xi$ is naturally defined as the vector field satisfying

$$\left((\mathcal{T}_\eta)^{-1}\xi\right)_x = (\mathcal{T}_{\eta_x})^{-1}\,(\xi_{R_x(\eta_x)}).$$

8.1.1 Vector transport and affine connections

There is a close relationship between vector transport and affine connections.

If $\mathcal{T}$ is a vector transport and R is the associated retraction, then

$$\nabla_{\eta_x}\xi := \left.\frac{\mathrm{d}}{\mathrm{d}t}\,\mathcal{T}_{t\eta_x}^{-1}\xi_{R(t\eta_x)}\right|_{t=0} \tag{8.1}$$

defines an affine connection. The properties are readily checked from the definition.

Conversely, parallel translation is a particular vector transport that can be associated with any affine connection. Let $\mathcal{M}$ be a manifold endowed with an affine connection ∇ and recall from Section 5.4 the notation $t \mapsto P_\gamma^{t\leftarrow a}\xi(a)$ for the parallel vector field on the curve γ that satisfies $P_\gamma^{a\leftarrow a} = \gamma(a)$ and

$$\frac{\mathrm{D}}{\mathrm{d}t}\left(P_\gamma^{t\leftarrow a}\xi(a)\right) = 0.$$

Proposition 8.1.2 *If ∇ is an affine connection and R is a retraction on a manifold $\mathcal{M}$, then*

$$\mathcal{T}_{\eta_x}(\xi_x) := P_\gamma^{1\leftarrow 0}\xi_x \tag{8.2}$$

is a vector transport with associated retraction R, where P_γ denotes the parallel translation induced by ∇ along the curve $t \mapsto \gamma(t) = R_x(t\eta_x)$. Moreover, $\mathcal{T}$ and ∇ satisfy (8.1)*.*

Proof. It is readily checked that (8.2) defines a vector transport. For the second claim, let R be a retraction and let $\mathcal{T}$ be defined by the parallel translation induced by ∇, i.e.,

$$\frac{\mathrm{D}}{\mathrm{d}t}\left(\mathcal{T}_{t\eta_x}\xi_x\right) = 0 \tag{8.3}$$

with $\pi(\mathcal{T}_{t\eta_x}\xi_x) = R(t\eta_x)$ and $\mathcal{T}_{0_x}\xi_x = \xi_x$. Let $\hat{\nabla}$ be defined by

$$\hat{\nabla}_{\eta_x}\xi := \frac{\mathrm{d}}{\mathrm{d}t}\,\mathcal{T}^{-1}_{t\eta_x}\xi_{R(t\eta_x)}\bigg|_{t=0}.$$

We want to show that $\nabla_{\eta_x}\xi = \hat{\nabla}_{\eta_x}\xi$ for all η_x, ξ. Let $\tilde{\xi}$ denote the vector field defined by $\tilde{\xi}_y = \mathcal{T}_{R_x^{-1}y}\xi_x$ for all y sufficiently close to x. We have

$$\hat{\nabla}_{\eta_x}\xi = \hat{\nabla}_{\eta_x}(\xi - \tilde{\xi}) + \hat{\nabla}_{\eta_x}\tilde{\xi} = \hat{\nabla}_{\eta_x}(\xi - \tilde{\xi}) = \nabla_{\eta_x}(\xi - \tilde{\xi}) = \nabla_{\eta_x}\xi,$$

where we have used the identities $\hat{\nabla}_{\eta_x}\tilde{\xi} = 0$ (which holds in view of the definitions of $\hat{\nabla}$ and $\tilde{\xi}$), $\hat{\nabla}_{\eta_x}(\xi - \tilde{\xi}) = \nabla_{\eta_x}(\xi - \tilde{\xi})$ (in view of $\xi_x - \tilde{\xi}_x = 0$), and $\nabla_{\eta_x}\tilde{\xi} = 0$ (since $\nabla_{\eta_x}\tilde{\xi} = \frac{\mathrm{D}}{\mathrm{d}t}\,\tilde{\xi}_{R(t\eta_x)}\Big|_{t=0} = \frac{\mathrm{D}}{\mathrm{d}t}\,\mathcal{T}_{t\eta_x}\xi_x\big|_{t=0} = 0$). □

We also point out that if $\mathcal{M}$ is a Riemannian manifold, then the parallel translation defined by the Riemannian connection is an isometry, i.e.,

$$\langle P_\gamma^{t\leftarrow a}\xi(a), P_\gamma^{t\leftarrow a}\zeta(a)\rangle = \langle \xi(a), \zeta(a)\rangle.$$

Example 8.1.1 ***Sphere***

We consider the sphere S^{n-1} *with its structure of Riemannian submanifold of* $\mathbb{R}^n$. *Let* $t \mapsto x(t)$ *be a geodesic for the Riemannian connection* (5.16) *on* S^{n-1}*; see* (5.25). *Let* u *denote* $\frac{1}{\|\dot{x}(0)\|}\dot{x}(0)$. *The parallel translation (associated with the Riemannian connection) of a vector* $\xi(0) \in T_{x(0)}$ *along the geodesic is given by*

$$\xi(t) = -x(0)\sin(\|\dot{x}(0)\|t)u^T\xi(0) + u\,\cos(\|\dot{x}(0)\|t)x^T(0)\xi(0) + (I - uu^T)\xi(0). \tag{8.4}$$

Example 8.1.2 ***Stiefel manifold***

There is no known closed form for the parallel translation along geodesics for the Stiefel manifold $\mathrm{St}(p,n)$ *endowed with the Riemannian connection inherited from the embedding in* $\mathbb{R}^{n\times p}$.

Example 8.1.3 ***Grassmann manifold***

Consider the Grassmann manifold viewed as a Riemannian quotient manifold of $\mathbb{R}^{n\times p}$ *with the inherited Riemannian connection. Let* $t \mapsto \mathcal{Y}(t)$ *be a geodesic for this connection, with* $\mathcal{Y}(0) = \mathrm{span}(Y_0)$ *and* $\overline{\dot{\mathcal{Y}}(0)}_{Y_0} = U\Sigma V^T$, *a thin singular value decomposition (i.e.,* U *is* $n \times p$ *orthonormal,* V *is* $p \times p$ *orthonormal, and* Σ *is* $p \times p$ *diagonal with nonnegative entries). We assume for simplicity that* Y_0 *is chosen orthonormal. Let* $\xi(0)$ *be a tangent vector at* $\mathcal{Y}(0)$. *Then the parallel translation of* $\xi(0)$ *along the geodesic is given by*

$$\overline{\xi(t)}_{Y(t)} = -Y_0V\sin(\Sigma t)U^T\overline{\xi(0)}_{Y_0} + U\cos(\Sigma t)U^T\overline{\xi(0)}_{Y_0} + (I - UU^T)\overline{\xi(0)}_{Y_0}. \tag{8.5}$$

Parallel translation is not the only way to achieve vector transport. As was the case with the choice of retraction, there is considerable flexibility in how a vector translation is chosen for a given problem. The approach taken will depend on the problem considered and the resourcefulness of the scientist designing the algorithm. In the next three subsections we present three approaches that can be used to generate computationally tractable vector translation mappings for the manifolds associated with the class of applications considered in this book.

8.1.2 Vector transport by differentiated retraction

Let $\mathcal{M}$ be a manifold endowed with a retraction R. Then a vector transport on $\mathcal{M}$ is defined by

$$\mathcal{T}_{\eta_x}\xi_x := \mathrm{D}R_x\left(\eta_x\right)\left[\xi_x\right]; \tag{8.6}$$

i.e.,

$$\mathcal{T}_{\eta_x}\xi_x = \frac{\mathrm{d}}{\mathrm{d}t}\,R_x(\eta_x + t\xi_x)\bigg|_{t=0};$$

see Figure 8.2. Notice in particular that, in view of the local rigidity condition $\mathrm{D}R_x(0_x) = \mathrm{id}$, the condition $\mathcal{T}_{0_x}\xi = \xi$ for all $\xi \in T_x\mathcal{M}$ is satisfied.

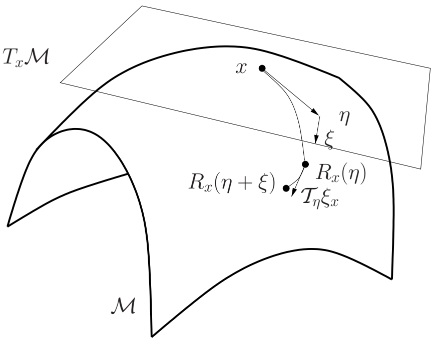

Figure 8.2 The vector transport $\mathcal{T}_\eta(\xi) := \mathrm{D}R_x\left(\eta\right)\left[\xi\right]$.

The definition (8.6) also provides a way to associate an affine connection with a retraction using (8.6) and (8.1).

We also point out that the vector transport (8.6) of a tangent vector along itself is given by

$$\mathcal{T}_{\eta_x}\eta_x = \frac{\mathrm{d}}{\mathrm{d}t}(R_x(t\eta_x))\bigg|_{t=1}.$$

Example 8.1.4 ***Sphere***

On the sphere S^{n-1} *with the projection retraction*

$$R_x(\xi_x) = (x+\xi_x)/||x+\xi_x||,$$

the vector transport (8.6) *yields*

$$\begin{aligned}\mathcal{T}_{\eta_x}\xi_x &= \frac{1}{\|x+\eta_x\|}\mathrm{P}_{x+\eta_x}\xi_x \\ &= \frac{1}{\|x+\eta_x\|}\left(I - \frac{1}{\|x+\eta_x\|^2}(x+\eta_x)(x+\eta_x)^T\right)\xi_x,\end{aligned}$$

where, as usual, we implicitly use the natural inclusion of $T_x S^{n-1}$ in $\mathbb{R}^n$.

Example 8.1.5 ***Stiefel manifold***

Consider the QR-based retraction (4.8) *on the Stiefel manifold:*

$$R_X(Z) = \mathrm{qf}(X+Z).$$

We need a formula for $\mathrm{D}\,\mathrm{qf}\,(Y)\,[U]$ *with* $Y \in \mathbb{R}_*^{n\times p}$ *and* $U \in T_Y \mathbb{R}_*^{n\times p} = \mathbb{R}^{n\times p}$*. Let* $t \mapsto W(t)$ *be a curve on* $\mathbb{R}_*^{n\times p}$ *with* $W(0) = Y$ *and* $\dot{W}(0) = U$ *and let* $W(t) = X(t)R(t)$ *denote the QR decomposition of* $W(t)$*. We have*

$$\dot{W} = \dot{X}R + X\dot{R}. \tag{8.7}$$

Since $XX^T + (I - XX^T) = I$*, we have the decomposition*

$$\dot{X} = XX^T\dot{X} + (I - XX^T)\dot{X}. \tag{8.8}$$

Multiplying (8.7) *by* $I - XX^T$ *on the left and by* R^{-1} *on the right yields the expression* $(I - XX^T)\dot{X} = (I - XX^T)\dot{W}R^{-1}$ *for the second term of* (8.8)*. It remains to obtain an expression for* $X^T\dot{X}$*. Multiplying* (8.7) *on the left by* X^T *and on the right by* R^{-1} *yields*

$$X^T\dot{W}R^{-1} = X^T\dot{X} + \dot{R}R^{-1}. \tag{8.9}$$

In view of the form

$$T_X\,\mathrm{St}(p,n) = \{X\Omega + X_\perp K : \Omega^T = -\Omega,\ K \in \mathbb{R}^{(n-p)\times p}\}$$

for the tangent space to the Stiefel manifold at a point X*, it follows that the term* $X^T\dot{X}$ *in* (8.9) *belongs to the set of skew-symmetric* $p\times p$ *matrices, while the term* $\dot{R}R^{-1}$ *belongs to the set of upper triangular matrices. Let* $\rho_{\mathrm{skew}}(B)$ *denote the the skew-symmetric term of the decomposition of a square matrix* B *into the sum of a skew-symmetric term and an upper triangular term, i.e,*

$$(\rho_{\mathrm{skew}}(B))_{i,j} = \begin{cases} B_{i,j} & \text{if } i > j, \\ 0 & \text{if } i = j, \\ -B_{j,i} & \text{if } i < j. \end{cases}$$

From (8.9)*, we have* $X^T\dot{X} = \rho_{\mathrm{skew}}(X^T\dot{W}R^{-1})$*. Replacing these results in* (8.8) *gives*

$$\dot{X} = XX^T\dot{X} + (I - XX^T)\dot{X} = X\rho_{\mathrm{skew}}(X^T\dot{W}R^{-1}) + (I - XX^T)\dot{W}R^{-1},$$

hence

$$\begin{aligned}\mathrm{D}\,\mathrm{qf}\,(Y)\,[U] = {} & \mathrm{qf}(Y)\rho_{\mathrm{skew}}(\mathrm{qf}(Y)^T U(\mathrm{qf}(Y)^T Y)^{-1}) \\ & + (I - \mathrm{qf}(Y)\mathrm{qf}(Y)^T)U(\mathrm{qf}(Y)^T Y)^{-1}.\end{aligned}$$

Finally, we have, for $Z, U \in T_X\,\mathrm{St}(p,n)$*,*

$$\begin{aligned}\mathcal{T}_Z U &= \mathrm{D}R_X\,(Z)\,[U] \\ &= \mathrm{D}\,\mathrm{qf}\,(X+Z)\,[U] \\ &= R_X(Z)\rho_{\mathrm{skew}}(R_X(Z)^T U(R_X(Z)^T(X+Z))^{-1}) \\ &\quad + (I - R_X(Z)R_X(Z)^T)U(R_X(Z)^T(X+Z))^{-1}.\end{aligned}$$

Example 8.1.6 ***Grassmann manifold***

As previously, we view the Grassmann manifold $\mathrm{Grass}(p,n)$ *as a Riemannian quotient manifold of* $\mathbb{R}_*^{n\times p}$. *We consider the retraction*

$$R_{\mathcal{Y}}(\eta) = \mathrm{span}(Y + \overline{\eta}_Y).$$

We obtain

$$\overline{\mathrm{D}R_{\mathcal{Y}}(\eta)\,[\xi]}_{R_{\mathcal{Y}}(\eta)} = P^h_{Y+\overline{\eta}_Y}\overline{\xi}_Y,$$

where P^h_Y *denotes the orthogonal projection onto the orthogonal complement of the span of* Y*; see* (3.41).

8.1.3 Vector transport on Riemannian submanifolds

If $\mathcal{M}$ is an embedded submanifold of a Euclidean space $\mathcal{E}$ and $\mathcal{M}$ is endowed with a retraction R, then we can rely on the natural inclusion $T_y\mathcal{M} \subset \mathcal{E}$ for all $y \in \mathcal{N}$ to simply define the vector transport by

$$\mathcal{T}_{\eta_x}\xi_x := \mathrm{P}_{R_x(\eta_x)}\xi_x, \qquad (8.10)$$

where P_x denotes the orthogonal projector onto $T_x\mathcal{N}$.

Example 8.1.7 ***Sphere***

On the sphere S^{n-1} *endowed with the retraction* $R(\eta_x) = (x+\eta_x)/\|x+\eta_x\|$, (8.10) *yields*

$$\mathcal{T}_{\eta_x}\xi_x = \left(I - \frac{(x+\eta_x)(x+\eta_x)^T}{\|x+\eta_x\|^2}\right)\xi_x \quad \in T_{R(\eta_x)}S^{n-1}.$$

Example 8.1.8 ***Orthogonal Stiefel manifold***

Let R *be a retraction on the Stiefel manifold* $\mathrm{St}(p,n)$. *(Possible choices of* R *are given in Section 4.1.1.) Formula* (8.10) *yields*

$$\mathcal{T}_{\eta_X}\xi_X = (I - YY^T)\xi_X + Y\,\mathrm{skew}(Y^T\xi_X) \quad \in T_Y\mathrm{St}(p,n),$$

where $Y := R_X(\eta_X)$.

8.1.4 Vector transport on quotient manifolds

Let $\mathcal{M} = \overline{\mathcal{M}}/\sim$ be a quotient manifold, where $\overline{\mathcal{M}}$ is an open subset of a Euclidean space $\mathcal{E}$ (this includes the case where $\overline{\mathcal{M}}$ itself is a Euclidean space). Let $\mathcal{H}$ be a horizontal distribution on $\overline{\mathcal{M}}$ and let $\mathrm{P}^h_{\overline{x}} : T_{\overline{x}}\overline{\mathcal{M}} \to \mathcal{H}_{\overline{x}}$ denote the projection parallel to the vertical space $\mathcal{V}_{\overline{x}}$ onto the horizontal space $\mathcal{H}_{\overline{x}}$. Then (using the natural identification $T_{\overline{y}}\overline{\mathcal{M}} \simeq \mathcal{E}$ for all $\overline{y} \in \overline{\mathcal{M}}$),

$$\overline{(\mathcal{T}_{\eta_x}\xi_x)}_{\overline{x}+\overline{\eta}_{\overline{x}}} := \mathrm{P}^h_{\overline{x}+\overline{\eta}_{\overline{x}}}\overline{\xi}_{\overline{x}} \qquad (8.11)$$

defines a vector transport on $\mathcal{M}$.

Example 8.1.9 ***Projective space***
As in Section 3.6.2, we view the projective space $\mathbb{RP}^{n-1}$ *as a Riemannian quotient manifold of* $\mathbb{R}^n_*$. *Equation* (8.11) *yields*

$$\overline{(\mathcal{T}_{\eta_{x\mathbb{R}}}\xi_{x\mathbb{R}})}_{x+\overline{\eta}_x} = \mathrm{P}^h_{x+\overline{\eta}_x}\overline{\xi}_x,$$

where $\mathrm{P}^h_y z = z - yy^T z$ *denotes the projection onto the horizontal space at* y.

Example 8.1.10 ***Grassmann manifold***
Again as in Section 3.6.2, we view the Grassmann manifold $\mathrm{Grass}(p, n)$ *as the Riemannian quotient manifold* $\mathbb{R}^{n\times p}_*/\mathrm{GL}_p$. *Equation* (8.11) *leads to*

$$\overline{(\mathcal{T}_{\eta_{\mathcal{Y}}}\xi_{\mathcal{Y}})}_{Y+\overline{\eta}_Y} = \mathrm{P}^h_{Y+\overline{\eta}_Y}\overline{\xi}_Y, \tag{8.12}$$

where $\mathrm{P}^h_Y Z = Z - Y(Y^TY)^{-1}Y^TZ$ *denotes the projection onto the horizontal space at* Y.

8.2 APPROXIMATE NEWTON METHODS

Let $\mathcal{M}$ be a manifold equipped with a retraction R and an affine connection ∇. Let ξ be a vector field on $\mathcal{M}$ and consider the problem of seeking a zero of ξ. The Newton equation (6.1) reads

$$\nabla_{\eta_x}\xi = -\xi_x$$

for the unknown $\eta_x \in T_x\mathcal{M}$. In Chapter 6, it was assumed that a procedure for computing $\nabla_{\eta_x}\xi$ is available at all $x \in \mathcal{M}$. In contrast, approximate Newton methods seek to relax the solution of Newton's equation in a way that retains the superlinear convergence of the algorithm. The kth iteration of the algorithm thus replaces (6.1) with the solution $\eta_k \in T_{x_k}\mathcal{M}$ of a relaxed equation

$$(J(x_k) + E_k)\eta_k = \ -\xi_{x_k} + \varepsilon_k, \tag{8.13}$$

where $J(x_k)$ is the Jacobian of ξ defined by

$$J(x_k) : T_{x_k}\mathcal{M} \to T_{x_k}\mathcal{M} : \eta_k \mapsto \nabla_{\eta_k}\xi.$$

The operator E_k denotes the approximation error on the Jacobian, while the tangent vector ε_k denotes the residual error in solving the (inexact) Newton equation.

The next result gives sufficiently small bounds on E_k and ε_k to preserve the fast local convergence of the exact Newton method.

Theorem 8.2.1 (local convergence of inexact Newton) *Suppose that at each step of Newton's method (Algorithm 4), the Newton equation* (6.1) *is replaced by the inexact equation* (8.13). *Assume that there exists* $x_* \in \mathcal{M}$ *such that* $\xi_{x_*} = 0$ *and* $J(x_*)$ *is invertible. Let* $(\mathcal{U}', \varphi)$, $x_* \in \mathcal{U}'$, *be a chart of* *the manifold $\mathcal{M}$ and let the coordinate expressions be denoted by $\hat{\cdot}$. Assume that there exist constants β_J and β_η such that*

$$\|\hat{E}_k\| \leq \beta_J \|\hat{\xi}_k\| \tag{8.14}$$

and

$$\|\hat{\varepsilon}_k\| \leq \min\{\|\hat{\xi}_k\|^\theta, \kappa\}\|\hat{\xi}_k\| \tag{8.15}$$

for all k, with $\theta > 0$. Then there exists a neighborhood $\mathcal{U}$ of x_* in $\mathcal{M}$ such that, for all $x_0 \in \mathcal{U}$, the inexact algorithm generates an infinite sequence $\{x_k\}$ converging superlinearly to x_*.

Proof. (Sketch.) The assumptions and notation are those of the proof of Theorem 6.3.2, and we sketch how that proof can be adapted to handle Theorem 8.2.1. By a smoothness argument,

$$\|\hat{\xi}_{\hat{x}}\| \leq \gamma_\xi \|\hat{x} - \hat{x}_*\|.$$

It follows from Lemma 6.3.1 that

$$\begin{aligned}
&\|(\hat{J}(\hat{x}_k) + \hat{E}_k)^{-1}\| \leq \|\hat{J}(\hat{x}_k)^{-1}\| \|(I - (\hat{J}(\hat{x}_k))^{-1}\hat{E}_k)^{-1}\| \\
&\leq 2\beta \frac{1}{1 - \|\hat{J}(\hat{x}_k))^{-1}\hat{E}_k\|} \leq 2\beta \frac{1}{1 - 2\beta\|\hat{E}_k\|} \leq 2\beta \frac{1}{1 - 2\beta\beta_J\gamma_\xi\|\hat{x}_k - \hat{x}_*\|}.
\end{aligned}$$

Consequently, by choosing $\mathcal{U}$ sufficiently small, $\|(\hat{J}(\hat{x}_k) + \hat{E}_k))^{-1}\|$ is bounded by a constant, say $2\beta'$, for all $x \in \mathcal{U}$. From there, it is direct to update the end of the proof of Theorem 6.3.2 to obtain again a bound

$$\begin{aligned}
\|\hat{x}_{k+1} - \hat{x}_*\| \leq &(\beta'(\gamma_J + \gamma_\Gamma) + 2\beta'\gamma_\Gamma + 2\beta'\beta_J\gamma_\xi + \gamma_R)\|\hat{x}_k - \hat{x}_*\|^2 \\
&+ 2\beta'\gamma_\xi^{\theta+1}\|\hat{x}_k - \hat{x}_*\|^{\theta+1}
\end{aligned}$$

for all x_k in some neighborhood of x_*. □

Condition (8.15) on the residual in the Newton equation is easily enforced by using an iterative solver that keeps track of the residual of the linear system of equations; the inner iteration is merely stopped as soon as the required precision is reached. Pointers to the literature on iterative solvers for linear equations can be found in Notes and References. Enforcing condition (8.14), on the other hand, involves differential geometric issues; this is the topic of the next section.

8.2.1 Finite difference approximations

A standard way to approximate the Jacobian $J(x_k)$ without having to compute second-order derivatives is to evaluate finite differences of the vector field ξ. On manifolds, the idea of evaluating finite differences on ξ is hindered by the fact that when $y \neq z$, the quantity $\xi_y - \xi_z$ is ill-defined, as the two tangent vectors belong to two different abstract Euclidean spaces $T_y\mathcal{M}$ and $T_z\mathcal{M}$. In practice, we will encounter only the case where a tangent vector η_y is known such that $z = R(\eta_y)$. We can then compare ξ_y and $\xi_{R(\eta_y)}$ using a

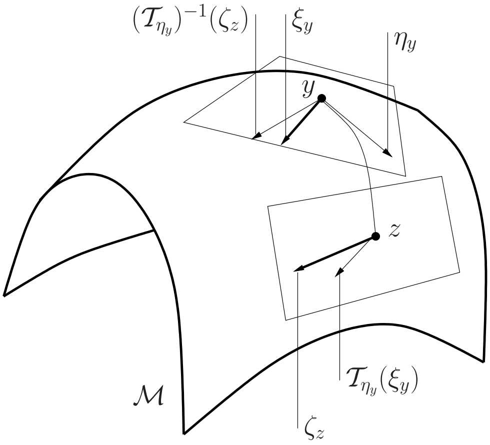
$(\mathcal{T}_{\eta_y})^{-1}(\zeta_z)$
ξ_y
η_y
y
z
$\mathcal{M}$
$\mathcal{T}_{\eta_y}(\xi_y)$
ζ_z

Figure 8.3 To compare a tangent vector $\xi_y \in T_y\mathcal{M}$ with a tangent vector $\zeta_z \in T_z\mathcal{M}$, $z = R(\eta_y)$, it is possible to transport ξ_y to $T_z\mathcal{M}$ through the mapping $\mathcal{T}_{\eta_y}$ or to transport ζ_z to $T_y\mathcal{M}$ through the mapping $(\mathcal{T}_{\eta_y})^{-1}$.

vector transport, as introduced in Section 8.1. Depending on the situation, we may want to compare the vectors in any of the two tangent spaces; see Figure 8.3.

To define finite differences in a neighborhood of a point x_* on a manifold $\mathcal{M}$ endowed with a vector transport $\mathcal{T}$, pick (smooth) vector fields E_i, $i = 1, \ldots, d$, such that $((E_1)_x, \ldots, (E_d)_x)$ forms a basis of $T_x\mathcal{M}$ for all x in a neighborhood $\mathcal{U}$ of x_*. Let R denote the retraction associated with the vector transport $\mathcal{T}$. Given a smooth vector field ξ and a real constant $h > 0$, let $A(x) : T_x\mathcal{M} \to T_x\mathcal{M}$ be the linear operator that satisfies, for $i = 1, \ldots, d$,

$$A(x)[E_i] = \frac{(\mathcal{T}_{h(E_i)_x})^{-1}\xi_{R(h(E_i)_x)} - \xi_x}{h}. \tag{8.16}$$

We thus have $A(x)[\eta_x] = \sum_{i=1}^{d} \eta^i|_x A(x)[E_i]$, where $\eta_x = \sum_{i=1}^{d} \eta^i|_x (E_i)_x$ is the decomposition of η_x in the basis $((E_1)_x, \ldots, (E_d)_x)$.

The next lemma gives a bound on how well $A(x)$ approximates the Jacobian $J(x) : \eta_x \mapsto \nabla_{\eta_x}\xi$ in a neighborhood of a zero of ξ. This result is instrumental in the local convergence analysis of the finite-difference quasi-Newton method introduced below.

Lemma 8.2.2 (finite differences) *Let ξ be a smooth vector field on a manifold $\mathcal{M}$ endowed with a vector transport $\mathcal{T}$ (Definition 8.1.1). Let x_* be a nondegenerate zero of ξ and let $(E_1, \ldots, E_d)$ be a basis of $\mathfrak{X}(\mathcal{U})$, where $\mathcal{U}$ is a neighborhood of x_*. Let A be defined by finite differences as in* (8.16). *Then there is $c > 0$ such that, for all x sufficiently close to x_* and all h sufficiently small, it holds that*

$$\|A(x)[E_i] - J(x)[E_i]\| \leq c(h + \|\xi_x\|). \tag{8.17}$$

Proof. This proof uses notation and conventions from the proof of Theorem 6.3.2. We work in local coordinates and denote coordinate expressions with a hat. (For example, $\hat{J}(\hat{x})$ denotes the coordinate expression of the operator $J(x)$.) There is a neighborhood $\mathcal{U}$ of x_* and constants $c_1, \ldots, c_6$ such that, for all $x \in \mathcal{U}$ and all $h > 0$ sufficiently small, the following bounds hold:

$$\begin{aligned}
&\|hA(x)[E_i] - J(x)[hE_i]\| \\
\leq\, & c_1 \|h\widehat{A(x)[E_i]} - \widehat{J(x)[hE_i]}\| \\
=\, & \|(\widehat{\mathcal{T}_{hE_i})^{-1}}\hat{\xi}_{\hat{R}_{\hat{x}}(h\hat{E}_i)} - \hat{\xi}_{\hat{x}} - \mathrm{D}\hat{\xi}\,(\hat{x})\left[h\hat{E}_i\right] - \hat{\Gamma}_{\hat{x},\hat{\xi}}[h\hat{E}_i]\| \\
\leq\, & \|\hat{\xi}_{\hat{x}+h\hat{E}_i} - \hat{\xi}_{\hat{x}} - \mathrm{D}\hat{\xi}\,(\hat{x})\left[h\hat{E}_i\right]\| + \|(\widehat{\mathcal{T}_{hE_i})^{-1}}\hat{\xi}_{\hat{R}_{\hat{x}}(h\hat{E}_i)} - \hat{\xi}_{\hat{R}_{\hat{x}}(h\hat{E}_i)}\| \\
&\quad + \|\hat{\xi}_{\hat{R}_{\hat{x}}(h\hat{E}_i)} - \hat{\xi}_{\hat{x}+h\hat{E}_i}\| + \|\hat{\Gamma}_{\hat{x},\hat{\xi}}[h\hat{E}_i]\| \\
\leq\, & c_2 h^2 + c_3 h(\|\hat{x} - \hat{x}_*\| + h) + c_4 h^2 + c_5 \|\hat{x} - \hat{x}_*\| h \\
\leq\, & c_6 h(h + \|\xi_x\|).
\end{aligned}$$

(A bound of the form $\|\hat{x} - \hat{x}_*\| \leq c\|\xi_x\|$ comes from the fact that x_* is a nondegenerate zero of ξ.) The claim follows. □

In the classical case, where $\mathcal{M}$ is a Euclidean space and the term

$$(\mathcal{T}_{h(E_i)_x})^{-1}\xi_{R(h(E_i)_x)}$$

in (8.16) reduces to ξ_{x+hE_i}, the bound (8.17) can be replaced by

$$\|A(x)[E_i] - J(x)[E_i]\| \leq ch, \tag{8.18}$$

i.e., $\|\xi_x\|$ no longer appears. The presence of $\|\xi_x\|$ is the counterpart to the fact that our definition of vector transport is particularly lenient. Fortunately, the perturbation $\|\xi_x\|$ goes to zero sufficiently fast as x goes to a zero of ξ. Indeed, using Lemma 8.2.2 and Theorem 8.2.1, we obtain the following result.

Proposition 8.2.3 *Consider the geometric Newton method (Algorithm 4) where the exact Jacobian $J(x_k)$ is replaced by the operator $A(x_k)$ defined in* (8.16) *with $h := h_k$. If*

$$\lim_{k\to\infty} h_k = 0,$$

then the convergence to nondegenerate zeros of ξ is superlinear. If, moreover, there exists some constant c such that

$$h_k \leq c\|\xi_{x_k}\|$$

for all k, then the convergence is (at least) quadratic.

8.2.2 Secant methods

An approximate Jacobian at $x \in \mathcal{M}$ is a linear operator in the d-dimensional tangent space $T_x\mathcal{M}$. Secant methods in $\mathbb{R}^n$ construct an approximate Jacobian A_{k+1} by imposing the secant equation

$$\xi_{x_{k+1}} - \xi_{x_k} = A_{k+1}\eta_k, \tag{8.19}$$

which can be seen as an underdetermined system of equations with d^2 unknowns. The remaining degrees of freedom in A_{k+1} are specified according to some algorithm that uses prior information where possible and also preserves or even improves the convergence properties of the underlying Newton method.

The generalization of the secant condition (8.19) on a manifold $\mathcal{M}$ endowed with a vector transport $\mathcal{T}$ is

$$\xi_{x_{k+1}} - \mathcal{T}_{\eta_k}\xi_{x_k} = A_{k+1}[\mathcal{T}_{\eta_k}\eta_k], \tag{8.20}$$

where η_k is the update vector at the iterate x_k, i.e., $R_{x_k}(\eta_k) = x_{k+1}$.

In the case where the manifold is Riemannian and ξ is the gradient of a real-valued function f of which a minimizer is sought, it is customary to require the following additional properties. Since the Hessian $J(x) = \operatorname{Hess} f(x)$ is symmetric (with respect to the Riemannian metric), one requires that the operator A_k be symmetric for all k. Further, in order to guarantee that η_k remains a descent direction for f, the updating formula should generate a positive-definite operator A_{k+1} whenever A_k is positive-definite. A well-known updating formula in $\mathbb{R}^n$ that aims at satisfying these properties is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) scheme. On a manifold $\mathcal{M}$ endowed with a vector transport $\mathcal{T}$, the BFGS scheme generalizes as follows. With the notation

$$\begin{aligned} s_k &:= \mathcal{T}_{\eta_k}\eta_k \in T_{x_{k+1}}\mathcal{M}, \\ y_k &:= \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\eta_k}(\operatorname{grad} f(x_k)) \in T_{x_{k+1}}\mathcal{M}, \end{aligned}$$

we define the operator $A_{k+1} : T_{x_{k+1}}\mathcal{M} \mapsto T_{x_{k+1}}\mathcal{M}$ by

$$A_{k+1}\eta = \tilde{A}_k\eta - \frac{\langle s_k, \tilde{A}_k\eta\rangle}{\langle s_k, \tilde{A}_k s_k\rangle}\tilde{A}_k s_k + \frac{\langle y_k, \eta\rangle}{\langle y_k, s_k\rangle} y_k \quad \text{for all } p \in T_{x_{k+1}}\mathcal{M},$$

with

$$\tilde{A}_k = \mathcal{T}_{\eta_k} \circ A_k \circ (\mathcal{T}_{\eta_k})^{-1}.$$

Note that the inner products are taken with respect to the Riemannian metric. Assume that A_k is symmetric positive-definite on $T_{x_k}\mathcal{M}$ (with respect to the inner product defined by the Riemannian metric) and that $\mathcal{T}_{\eta_k}$ is an isometry (i.e., the inverse of $\mathcal{T}_{\eta_k}$ is equal to its adjoint). Then $\tilde{A}_k$ is symmetric positive-definite, and it follows from the classical BFGS theory that A_{k+1} is symmetric positive-definite on $T_{x_{k+1}}\mathcal{M}$ if and only if $\langle y_k, s_k\rangle > 0$. The advantage of A_k is that it requires only first-order information that has to be computed anyway to provide the right-hand side of the Newton equation.

The local and global convergence analysis of the BFGS method in $\mathbb{R}^n$ is not straightforward. A careful generalization to manifolds, in the vein of the work done in Chapter 7 for trust-region methods, is beyond the scope of the present treatise.

8.3 CONJUGATE GRADIENTS

In this section we depart the realm of quasi-Newton methods to briefly consider conjugate gradient algorithms. We first summarize the principles of CG in $\mathbb{R}^n$.

The *linear* CG algorithm can be presented as a method for minimizing the function

$$\phi(x) = \tfrac{1}{2} x^T A x - x^T b, \tag{8.21}$$

where $b \in \mathbb{R}^n$ and A is an $n \times n$ symmetric positive-definite matrix. One of the simplest ways to search for the minimizer of ϕ is to use a steepest-descent method, i.e., search along

$$-\operatorname{grad} \phi(x_k) = b - Ax_k := r_k,$$

where r_k is called the *residual* of the iterate x_k. Unfortunately, if the matrix A is ill-conditioned, then the steepest-descent method may be very slow. (Recall that the convergence factor r in Theorem 4.5.6 goes to 1 as the ratio between the smallest and the largest eigenvalues of A—which are the eigenvalues of the constant Hessian of ϕ—goes to zero.) Conjugate gradients provide a remedy to this drawback by modifying the search direction at each step. Let x_0 denote the initial iterate and let $p_0, \dots, p_k$ denote the successive search directions that can be used to generate x_{k+1}. A key observation is that, writing x_{k+1} as

$$x_{k+1} = x_0 + P_{k-1} y + \alpha p_k,$$

where $P_{k-1} = [p_1 | \dots | p_{k-1}]$, $y \in \mathbb{R}^{k-1}$, and $\alpha \in \mathbb{R}$, we have

$$\phi(x_{k+1}) = \phi(x_0 + P_{k-1} y) + \alpha y^T P_{k-1}^T A p_k + \frac{\alpha^2}{2} p_k^T A p_k - \alpha p_k^T r_0.$$

Hence the minization of $\phi(x_{k+1})$ splits into two independent minimizations—one for y and one for α—when the search direction p_k is chosen to be A-orthogonal to the previous search directions, i.e.,

$$P_{k-1}^T A p_k = 0.$$

It follows that if the search directions $p_0, \dots, p_k$ are *conjugate* with respect to A, i.e.,

$$p_i^T A p_j = 0 \qquad \text{for all } i \neq j,$$

then an algorithm, starting from x_0 and performing successive exact line-search minimizations of ϕ along $p_0, \dots, p_k$, returns a point x_{k+1} that is the minimizer of ϕ over the set $x_0 + \operatorname{span}\{p_0, \dots, p_k\}$.

Thus far we have only required that the search directions be conjugate with respect to A. The linear CG method further relates the search directions to the gradients by selecting each p_k to be in the direction of the minimizer of $\|p - r_k\|_2$ over all vectors p satisfying the A-orthogonality condition $[p_1 | \dots | p_{k-1}]^T A p = 0$. It can be shown that this requirement is satisfied by

$$p_k = r_k + \beta_k p_{k-1}, \tag{8.22}$$

where

$$\beta_k = -\frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}. \tag{8.23}$$

Summarizing, the linear CG iteration is

$$x_{k+1} = x_k + \alpha_k p_k,$$

where α_k is chosen as

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$$

to achieve exact minimization of ϕ along the line $x_k + \alpha p_k$ and where p_k is selected according to (8.22), (8.23). The first search direction p_0 is simply chosen as the steepest-descent direction at x_0. This algorithm is usually presented in a mathematically equivalent but numerically more efficient formulation, which is referred to as *the* (linear) CG algorithm. Notice that, since the minimizer of ϕ is $x = A^{-1}b$, the linear CG algorithm can also be used to solve systems of equations whose matrices are symmetric positive-definite.

Several generalizations of the linear CG algorithm have been proposed for cost functions f that are not necessarily of the quadratic form (8.21) with $A = A^T$ positive-definite. These algorithms are termed *nonlinear CG* methods. Modifications with respect to the linear CG algorithm occur at three places: (i) the residual r_k becomes the negative gradient $-\operatorname{grad} f(x_k)$, which no longer satisfies the simple recursive formula $r_{k+1} = r_k + \alpha_k A p_k$; (ii) computation of the line-search step α_k becomes more complicated and can be achieved approximately using various line-search procedures; (iii) several alternatives are possible for β_k that yield different nonlinear CG methods but nevertheless reduce to the linear CG method when f is strictly convex-quadratic and α_k is computed using exact line-search minimization. Popular choices for β_k in the formula

$$p_k = -\operatorname{grad} f(x_k) + \beta_k p_{k-1} \tag{8.24}$$

are

$$\beta_k = \frac{(\operatorname{grad} f(x_k))^T \operatorname{grad} f(x_k)}{(\operatorname{grad} f(x_{k-1}))^T \operatorname{grad} f(x_{k-1})} \quad \text{(Fletcher-Reeves)}$$

and

$$\beta_k = \frac{(\operatorname{grad} f(x_k))^T (\operatorname{grad} f(x_k) - \operatorname{grad} f(x_{k-1}))}{(\operatorname{grad} f(x_{k-1}))^T \operatorname{grad} f(x_{k-1})} \quad \text{(Polak-Ribière).}$$

When generalizing nonlinear CG methods to manifolds, we encounter a familiar difficulty: in (8.24), the right-hand side involves the sum of an element $\operatorname{grad} f(x_k)$ of $T_{x_k}\mathcal{M}$ and an element p_{k-1} of $T_{x_{k-1}}\mathcal{M}$. Here again, the concept of vector transport provides an adequate and flexible solution. We are led to propose a "meta-algorithm" (Algorithm 13) for the conjugate gradient.

Algorithm 13 Geometric CG method

```
Require: Riemannian manifold $\mathcal{M}$; vector transport $\mathcal{T}$ on $\mathcal{M}$ with associated retraction $R$; real-valued function $f$ on $\mathcal{M}$.
Goal: Find a local minimizer of $f$.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Sequence of iterates $\{x_k\}$.
Set $\eta_0 = -\operatorname{grad} f(x_0)$.
for $k = 0, 1, 2, \ldots$ do
    Compute a step size $\alpha_k$ and set
        $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$.    (8.25)
    Compute $\beta_{k+1}$ and set
        $\eta_{k+1} = -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}(\eta_k)$.    (8.26)
end for
```

In Step 3 of Algorithm 13, the computation of α_k can be done, for example, using a line-search backtracking procedure as described in Algorithm 1. If the numerical cost of computing the exact line-search solution is not prohibitive, then the minimizing value of α_k should be used. Exact line-search minimization yields $0 = \frac{\mathrm{d}}{\mathrm{d}t} f(R_{x_k}(t\eta_k))\big|_{t=\alpha_k} = \mathrm{D}f\,(x_{k+1})\left[\frac{\mathrm{d}}{\mathrm{d}t} R_{x_k}(t\eta_k)\big|_{t=\alpha_k}\right]$. Assuming that $\mathcal{T}_{\alpha_k\eta_k}(\eta_k)$ is collinear with $\frac{\mathrm{d}}{\mathrm{d}t} R_{x_k}(t\eta_k)\big|_{t=\alpha_k}$ (see Section 8.1.2), this leads to $\langle \operatorname{grad} f(x_{k+1}), \mathcal{T}_{\alpha_k\eta_k}(\eta_k)\rangle = \mathrm{D}f\,(x_{k+1})\,[\mathcal{T}_{\alpha_k\eta_k}(\eta_k)] = 0$. In view of (8.26), one finds that

$$\langle \operatorname{grad} f(x_{k+1}), \eta_{k+1}\rangle = -\langle \operatorname{grad} f(x_{k+1}), \operatorname{grad} f(x_{k+1})\rangle < 0,$$

i.e., η_{k+1} is a descent direction for f.

Several choices are possible for β_{k+1} in Step 4 of Algorithm 13. Imposing the condition that η_{k+1} and $\mathcal{T}_{\alpha_k\eta_k}(\eta_k)$ be conjugate with respect to $\operatorname{Hess} f(x_{k+1})$ yields

$$\beta_{k+1} = \frac{\langle \mathcal{T}_{\alpha_k\eta_k}(\eta_k), \operatorname{Hess} f(x_{k+1})[\operatorname{grad} f(x_{k+1})]\rangle}{\langle \mathcal{T}_{\alpha_k\eta_k}(\eta_k), \operatorname{Hess} f(x_{k+1})[\mathcal{T}_{\alpha_k\eta_k}(\eta_k)]\rangle}. \tag{8.27}$$

The β of Fletcher-Reeves becomes

$$\beta_{k+1} = \frac{\langle \operatorname{grad} f(x_{k+1}), \operatorname{grad} f(x_{k+1})\rangle}{\langle \operatorname{grad} f(x_k), \operatorname{grad} f(x_k)\rangle}, \tag{8.28}$$

whereas the β of Polak-Ribière naturally generalizes to

$$\beta_{k+1} = \frac{\langle \operatorname{grad} f(x_{k+1}), \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k\eta_k}(\operatorname{grad} f(x_k))\rangle}{\langle \operatorname{grad} f(x_k), \operatorname{grad} f(x_k)\rangle}. \tag{8.29}$$

Whereas the convergence theory of linear CG is well understood, nonlinear CG methods have convergence properties that depend on the choice of α_k and β_k, even in the case of $\mathbb{R}^n$. We do not further discuss such convergence issues in the present framework.

8.3.1 Application: Rayleigh quotient minimization

As an illustration of the geometric CG algorithm, we apply Algorithm 13 to the problem of minimizing the Rayleigh quotient function (2.1) on the Grassmann manifold. For simplicity, we consider the standard eigenvalue problem (namely, $B := I$), which leads to the cost function

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

where A is an arbitrary $n\times n$ symmetric matrix. As usual, we view $\mathrm{Grass}(p, n)$ as a Riemannian quotient manifold of $\mathbb{R}_*^{n\times p}$ (see Section 3.6.2). Formulas for the gradient and the Hessian of f can be found in Section 6.4.2. For Step 3 of Algorithm 13 (the line-search step), we select x_{k+1} as the Armijo point (Definition 4.2.2) with $\overline{\alpha} = 1$, $\sigma = 0.5$, and $\beta = 0.5$. For Step 4 (selection of the next search direction), we use the Polak-Ribière formula (8.29). The retraction is chosen as in (4.11), and the vector transport is chosen according to (8.12). The algorithm further uses a restart strategy that consists of choosing $\beta_{k+1} := 0$ when k is a multiple of the dimension $d = p(n - p)$ of the manifold. Numerical results are presented in Figures 8.4 and 8.5.

The resulting algorithm appears to be an efficient method for computing an extreme invariant subspace of a symmetric matrix. One should bear in mind, however, that this is only a brute-force application of a very general optimization scheme to a very specific problem. As such, the algorithm admits several enhancements that exploit the simple structure of the Rayleigh quotient cost function. A key observation is that it is computationally inexpensive to optimize the Rayleigh quotient over a low-dimensional subspace since this corresponds to a small-dimensional eigenvalue problem. This suggests a modification of the nonlinear CG scheme where the next iterate x_{k+1} is obtained by minimizing the Rayleigh quotient over the space spanned by the columns of x_k, η_{k-1} and $\mathrm{grad}\, f(x_k)$. The algorithm obtained using this modification, barring implementation issues, is equivalent to the locally optimal CG method proposed by Knyazev (see Notes and References in Chapter 4).

An interesting point of comparison between the numerical results displayed in Figures 7.1 and 7.2 for the trust-region approach and in Figures 8.4 and 8.5 is that the trust-region algorithm reaches twice the precision of the CG algorithm. The reason is that, around a minimizer v of a smooth cost function f, one has $f(R_v(\eta)) = f(v) + O(\|\eta\|^2)$, whereas $\|\mathrm{grad}\, f(R_v(\eta))\| = O(\|\eta\|)$. Consequently, the numerical evaluation of $f(x_k)$ returns exactly $f(v)$ as soon as the distance between x_k and v is of the order of the square root of the machine epsilon, and the line-search process in Step 3 of Algorithm 13 just returns $x_{k+1} = x_k$. In contrast, the linear CG method used in the inner iteration of the trust-region method, with its exact minimization formula for α_k, makes it possible to obtain accuracies of the order of the machine epsilon. Another potential advantage of the trust-region approach over nonlinear CG methods is that it requires significantly fewer evaluations of the cost function f since it relies only on its local model m_{x_k} to carry out the inner iteration process. This is important when the cost function is expensive to compute.

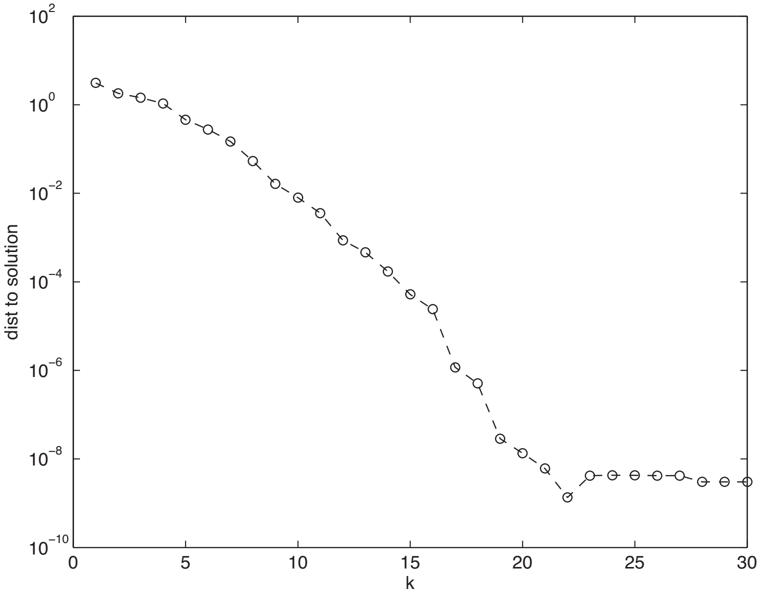

Figure 8.4 Minimization of the Rayleigh quotient (2.1) on Grass(p, n), with $n = 100$ and $p = 5$. $B = I$ and A is chosen with p eigenvalues evenly spaced on the interval $[1, 2]$ and the other $(n - p)$ eigenvalues evenly spaced on the interval $[10, 11]$; this is a problem with a large eigenvalue gap. The distance to the solution is defined as the square root of the sum of the canonical angles between the current subspace and the leftmost p-dimensional invariant subspace of A. (This distance corresponds to the geodesic distance on the Grassmann manifold endowed with its canonical metric (3.44).)

8.4 LEAST-SQUARE METHODS

The problem addressed by the geometric Newton method presented in Algorithm 4 is to compute a zero of a vector field on a manifold $\mathcal{M}$ endowed with a retraction R and an affine connection ∇. A particular instance of this method is Algorithm 5, which seeks a critical point of a real-valued function f by looking for a zero of the gradient vector field of f. This method itself admits enhancements in the form of line-search and trust-region methods that ensure that f decreases at each iteration and thus favor convergence to local minimizers.

In this section, we consider more particularly the case where the real-valued function f takes the form

$$f : \mathcal{M} \to \mathbb{R} : x \mapsto \tfrac{1}{2}\|F(x)\|^2, \tag{8.30}$$

where

$$F : \mathcal{M} \to \mathcal{E} : x \mapsto F(x)$$

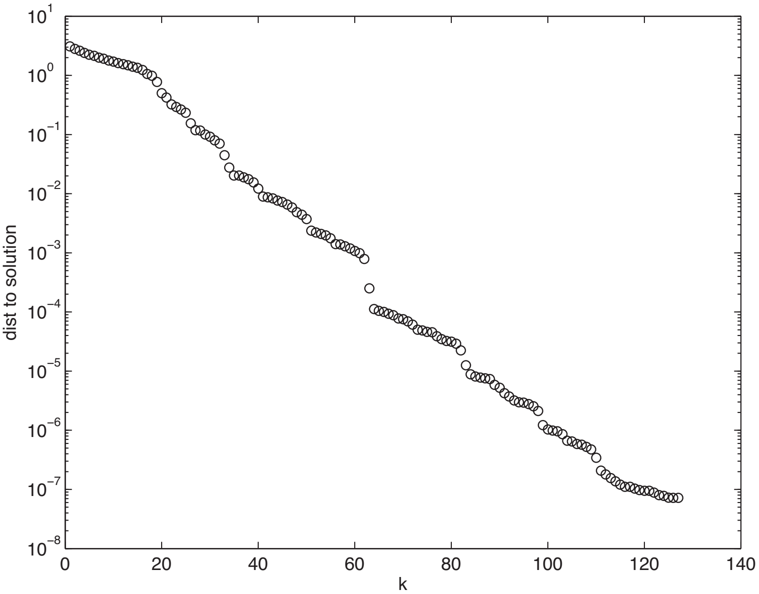

Figure 8.5 Same situation as in Figure 8.4 but now with $B = I$ and $A = \text{diag}(1, \ldots, n)$.

is a function on a Riemannian manifold $(\mathcal{M}, g)$ into a Euclidean space $\mathcal{E}$. The goal is to minimize $f(x)$. This is a *least-squares problem* associated with the least-squares cost $\sum_i (F_i(x))^2$, where $F_i(x)$ denotes the ith component of $F(x)$ in some orthonormal basis of $\mathcal{E}$. We assume throughout that $\dim(\mathcal{E}) \geq \dim(\mathcal{M})$, in other words, there are at least as many equations as "unknowns". Minimizing f is clearly equivalent to minimizing $\|F(x)\|$. Using the squared cost is important for regularity purposes, whereas the $\frac{1}{2}$ factor is chosen to simplify the equations.

Recall that $\|F(x)\|^2 := \langle F(x), F(x)\rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathcal{E}$. We have, for all $\xi \in T_x\mathcal{M}$,

$$\mathrm{D}f\,(x)\,[\xi] = \langle \mathrm{D}F\,(x)\,[\xi]\,, F(x)\rangle = \langle \xi, (\mathrm{D}F(x))^*[F(x)]\rangle,$$

where $(\mathrm{D}F(x))^*$ denotes the adjoint of the operator $\mathrm{D}F(x) : T_x\mathcal{M} \to \mathcal{E}$, i.e.,

$$\langle y, \mathrm{D}F\,(x)\,[\xi]\rangle = g((\mathrm{D}F(x))^*[y], \xi)$$

for all $y \in T_{F(x)}\mathcal{E} \simeq \mathcal{E}$ and all $\xi \in T_x\mathcal{M}$. Hence

$$\operatorname{grad} f(x) = (\mathrm{D}F(x))^*[F(x)].$$

Further, we have, for all $\xi, \eta \in T_x\mathcal{M}$,

$$\nabla^2 f(x)[\xi, \eta] = \langle \mathrm{D}F\,(x)\,[\xi]\,, \mathrm{D}F\,(x)\,[\eta]\rangle + \langle F(x), \nabla^2 F(x)[\xi, \eta]\rangle, \qquad (8.31)$$

where $\nabla^2 f(x)$ is the $(0, 2)$-tensor defined in Section 5.6.

8.4.1 Gauss-Newton methods

Recall that the geometric Newton method (Algorithm 5) computes an update vector $\eta \in T_x\mathcal{M}$ by solving the equation

$$\operatorname{grad} f(x) + \operatorname{Hess} f(x)[\eta] = 0,$$

or equivalently,

$$\mathrm{D}f\,(x)\,[\xi] + \nabla^2 f(x)[\xi,\eta] = 0 \quad \text{for all } \xi \in T_x\mathcal{M}.$$

The *Gauss-Newton* method is an approximation of this geometric Newton method for the case where $f(x) = \|F(x)\|^2$ as in (8.30). It consists of approximating $\nabla^2 f(x)[\xi,\eta]$ by the term $\langle \mathrm{D}F\,(x)\,[\xi]\,, \mathrm{D}F\,(x)\,[\eta]\rangle$; see (8.31). This yields the Gauss-Newton equation

$$\langle \mathrm{D}F\,(x)\,[\xi]\,, F(x)\rangle + \langle \mathrm{D}F\,(x)\,[\xi]\,, \mathrm{D}F\,(x)\,[\eta]\rangle = 0 \quad \text{for all } \xi \in T_x\mathcal{M},$$

or equivalently,

$$(\mathrm{D}F(x))^*[F(x)] + ((\mathrm{D}F(x))^* \circ \mathrm{D}F(x))[\eta] = 0.$$

The geometric Gauss-Newton method is given in Algorithm 14. (Note that the affine connection ∇ is not required to state the algorithm.)

Algorithm 14 Riemannian Gauss-Newton method

```
Require: Riemannian manifold $\mathcal{M}$; retraction $R$ on $\mathcal{M}$; function $F : \mathcal{M} \to \mathcal{E}$ where $\mathcal{E}$ is a Euclidean space.
Goal: Find a (local) least-squares solution of $F(x) = 0$.
Input: Initial iterate $x_0 \in \mathcal{M}$.
Output: Sequence of iterates $\{x_k\}$.
for $k = 0, 1, 2, \dots$ do
    Solve the Gauss-Newton equation
        $((\mathrm{D}F(x_k))^* \circ \mathrm{D}F(x_k))\,[\eta_k] = -(\mathrm{D}F(x_k))^*[F(x_k)]$    (8.32)
    for the unknown $\eta_k \in T_{x_k}\mathcal{M}$.
    Set
        $x_{k+1} := R_{x_k}(\eta_k)$.
end for
```

In the following discussion, we assume that the operator $\mathrm{D}F(x_k)$ is injective (i.e., full rank, since we have assumed $n \geq d$). The Gauss-Newton equation (8.32) then reads

$$\eta_k = ((\mathrm{D}F(x_k))^* \circ (\mathrm{D}F(x_k)))^{-1}\,[(\mathrm{D}F(x_k))^*[F(x_k)]];$$

i.e.,

$$\eta_k = (\mathrm{D}F(x_k))^\dagger[F(x_k)], \tag{8.33}$$

where $(\mathrm{D}F(x_k))^\dagger$ denotes the *Moore-Penrose inverse* or *pseudo-inverse* of the operator $\mathrm{D}F(x_k)$.

Key advantages of the Gauss-Newton method over the plain Newton method applied to $f(x) := \|F(x)\|^2$ are the lower computational complexity of producing the iterates and the property that, as long as $\mathrm{D}F(x_k)$ has full rank, the Gauss-Newton direction is a descent direction for f. Note also that the update vector η_k turns out to be the least-squares solution

$$\arg\min_{\eta \in T_{x_k}\mathcal{M}} \|\mathrm{D}F(x_k)[\eta] + F(x_k)\|^2.$$

In fact, instead of finding the critical point of the quadratic model of f, the Gauss-Newton method computes the minimizer of the norm of the "model" $F(x_k) + \mathrm{D}F(x_k)[\eta]$ of F.

Usually, Algorithm 14 is used in combination with a line-search scheme that ensures a sufficient decrease in f. If the sequence $\{\eta_k\}$ generated by the method is gradient-related, then global convergence follows from Theorem 4.3.1.

The Gauss-Newton method is in general not superlinearly convergent. In view of Theorem 8.2.1, on the convergence of inexact Newton methods, it *is* superlinearly convergent to a nondegenerate minimizer x_* of f when the neglected term $\langle F(x), \nabla^2 F(x)[\xi, \eta]\rangle$ in (8.31) vanishes at x_*. In particular, this is the case when $F(x_*) = 0$, i.e., the (local) least-squares solution x_* turns out to be a zero of F.

8.4.2 Levenberg-Marquardt methods

An alternative to the line-search enhancement of Algorithm 14 (Gauss-Newton) is to use a trust-region approach. The model is chosen as

$$\begin{aligned} m_{x_k}(\eta) = \tfrac{1}{2}\|F(x_k)\|^2 &+ g(\eta, (\mathrm{D}F(x)_k)^*[F(x_k)]) \\ &+ \tfrac{1}{2} g(\eta, ((\mathrm{D}F(x))^* \circ \mathrm{D}F(x))[\eta]]) \end{aligned}$$

so that the critical point of the model is the solution η_k of the Gauss-Newton equation (8.32). (We assume that $\mathrm{D}F(x_k)$ is full rank for simplicity of the discussion.) All the convergence analyses of Riemannian trust-region methods apply.

In view of the characterization of the solutions of the trust-region subproblems in Proposition 7.3.1, the minimizer of $m_{x_k}(\eta)$ within the trust region $\|\eta\| \leq \Delta_k$ is either the solution of the Gauss-Newton equation (8.32) when it falls within the trust region, or the solution of

$$((\mathrm{D}F(x_k))^* \circ \mathrm{D}F(x_k) + \mu_k \,\mathrm{id})\eta = -(\mathrm{D}F(x_k))^* F(x_k), \tag{8.34}$$

where μ_k is such that the solution η_k satisfies $\|\eta_k\| = \Delta_k$. Equation (8.34) is known as the *Levenberg-Marquard* equation.

Notice that the presence of $\mu\,\mathrm{id}$ as a modification of the approximate Hessian $(\mathrm{D}F(x))^* \circ \mathrm{D}F(x)$ of f is analogous to the idea in (6.6) of making the modified Hessian positive-definite by adding a sufficiently positive-definite perturbation to the Hessian.

8.5 NOTES AND REFERENCES

On the Stiefel manifold, it is possible to obtain a closed form for the parallel translation along geodesics associated with the Riemannian connection obtained when viewing the manifold as a Riemannian quotient manifold of the orthogonal group; see Edelman *et al.* [EAS98]. We refer the reader to Edelman *et al.* [EAS98] for more information on the geodesics and parallel translations on the Stiefel manifold. Proof that the Riemannian parallel translation is an isometry can be found in [O'N83, Lemma 3.20].

More information on iterative methods for linear systems of equations can be found in, e.g., Axelsson [Axe94], Saad [Saa96], van der Vorst [vdV03], and Meurant [Meu06].

The proof of Lemma 8.2.2 is a generalization of the proof of [DS83, Lemma 4.2.1].

For more information on quasi-Newton methods in $\mathbb{R}^n$, see, e.g., Dennis and Schnabel [DS83] or Nocedal and Wright [NW99]. An early reference on quasi-Newton methods on manifolds (more precisely, on submanifolds of $\mathbb{R}^n$) is Gabay [Gab82]. The material on BFGS on manifolds comes from [Gab82], where we merely replaced the usual parallel translation by the more general notion of vector transport. Hints for the convergence analysis of BFGS on manifolds can also be found in [Gab82].

The linear CG method is due to Hestenes and Stiefel [HS52]. Major results for nonlinear CG algorithms are due to Fletcher and Reeves [FR64] and Polak and Ribiere [PR69]. More information can be found in, e.g., [NW99]. A counterexample showing lack of convergence of the Polak-Ribière method can be found in Powell [Pow84].

Smith [Smi93, Smi94] proposes a nonlinear CG algorithm on Riemannian manifolds that corresponds to Algorithm 13 with the retraction R chosen as the Riemannian exponential map and the vector transport $\mathcal{T}$ defined by the parallel translation induced by the Riemannian connection. Smith points out that the Polak-Ribière version of the algorithm has n-step quadratic convergence towards nondegenerate local minimizers of the cost function.

The Gauss-Newton method on Riemannian manifolds can be found in Adler *et al.* [ADM$^+$02] in a formulation similar to Algorithm 14.

The original Levenberg-Marquardt algorithm [Lev44, Mar63] did not make the connection with the trust-region approach; it proposed heuristics to adapt μ directly.

More information on the classical version of the methods presented in this chapter can be found in textbooks on numerical optimization such as [Fle01, DS83, NS96, NW99, BGLS03].