IMA Journal of Numerical Analysis (2005) Page 1 of 24
doi: 10.1093/imanum/dri017

An implicit trust-region method on Riemannian manifolds *

C. G. BAKER† ‡, P.-A. ABSIL§, K. A. GALLIVAN‡

† *Computational Mathematics and Algorithms, Sandia National Laboratories, Albuquerque, NM 87185-1320, USA*

‡ *School of Computational Science, Florida State University, Tallahassee, FL 32306-4120, USA*

§ *Département d'ingénierie mathématique, Université catholique de Louvain, Avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium*

[Received on 31 March 2007, Revised on 31 December, 2007]

We propose and analyze an "implicit" trust-region method in the general setting of Riemannian manifolds. The method is implicit in that the trust-region is defined as a superlevel set of the ρ ratio of the actual over predicted decrease in the objective function. Since this method potentially requires the evaluation of the objective function at each step of the inner iteration, we do not recommend it for problems where the objective function is expensive to evaluate. However, we show that on some instances of a very structured problem—the extreme symmetric eigenvalue problem, or equivalently the optimization of the Rayleigh quotient on the unit sphere—the resulting numerical method outperforms state-of-the-art algorithms. Moreover, the new method inherits the detailed convergence analysis of the generic Riemannian trust-region method.

Keywords: optimization on manifolds, trust-region methods, Newton's method, symmetric generalized eigenvalue problem

1. Introduction

Trust-region methods are widely used in the unconstrained optimization of smooth functions. Much of the reason for their popularity is the superposition of strong global convergence, fast local convergence, and ease of implementation. In Powell (1970b), Powell helped to establish a following for this family of methods. In addition to proving global convergence of the method under mild conditions, the work showed that the method was competitive with state-of-the-art algorithms for unconstrained optimization. This launched a period of great interest in the methods; see Conn et al. (2000) and references therein.

Recently, there has been a significant interest in the theory and practice of the optimization of functions defined on Riemannian manifolds. The goal of much of this effort has been the transfer of algorithms and theory from classical (Euclidean) optimization to a Riemannian setting. These approaches typically assume analogous conditions on the objective function and attempt to preserve the convergence properties of their Euclidean counterparts. The Riemannian setting comes with some additional overhead, but this effort is worthwhile. First, applications and theory developed in the generalized Riemannian setting are immediately applicable to unconstrained Euclidean optimization; Euclidean space

*This work was supported by NSF Grant ACI0324944. The first author was in part supported by the CSRI, Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy; contract/grant number: DE-AC04-94AL85000. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

IMA Journal of Numerical Analysis © Institute of Mathematics and its Applications 2005; all rights reserved.

is, after all, a very friendly Riemannian manifold. Second, while some Riemannian optimization problems can be described as constrained Euclidean optimization problems, the theoretical overhead of the (unconstrained) Riemannian setting negates the need for the complications associated with constrained optimization, e.g., modifications to the conditions on optimality; adaptation of methods to handle the constraints; the possibility of returning unfeasible points in the case of early termination from some methods. Third, certain Riemannian optimization problems of significant interest cannot be easily recast as constrained Euclidean optimization problems. In particular, Section 4.2 concerns the computation of extreme eigenspaces of a positive definite matrix pencil.

There is a significant and growing literature regarding optimization on Riemannian manifolds. Much work concerns the application of familiar methods from Euclidean optimization. The first Riemannian optimization methods to be considered were steepest descent and Newton's method; see, e.g., Luenberger (1972); Gabay (1982); Smith (1993); Helmke and Moore (1994); Udrişte (1994); Smith (1994); Edelman et al. (1998); Adler et al. (2002); Dedieu et al. (2003) and references therein. A Riemannian conjugate gradient method was presented in Smith (1993, 1994). This paper builds off of Absil et al. (2007), which describes a trust-region framework for Riemannian optimization, the Riemannian Trust Region method.

Similar to Euclidean trust-region methods, the Riemannian Trust-Region (RTR) method ensures strong global convergence properties while allowing superlinear local convergence. The trust-region mechanism is a heuristic, whereby the performance of the last update dictates the constraints on the next update. The trust-region mechanism makes it possible to disregard the (potentially expensive) objective function during the inner iteration by relying instead on a model restricted to a trust region, i.e., a region where the model is tentatively trusted to be a sufficiently accurate approximation of the objective function. A downside lies in the difficulty of adjusting the trust-region size. When the trust-region radius is too large, valuable time may be spent proposing a new iterate that may be rejected. Alternatively, when the trust-region radius is too small, the algorithm progresses unnecessarily slowly.

The inefficiencies resulting from the trust-region mechanism can be addressed by disabling the trust-region mechanism in such a way as to preserve the desired convergence properties. For example, in Gould et al. (2005), the authors describe a filter-trust-region method, where a modified acceptance criterion seeks to encourage convergence to first-order critical points. Other approaches adjust the trust-region radius according to dynamic measures such as objective function improvement and step size lengths; see Conn et al. (2000).

Instead of relaxing the acceptance criterion, this paper proposes that the trust-region be identified as that set of points that would have been accepted under the classical mechanism. Therefore, as long as the update returned from the model minimization is feasible, i.e., it belongs to the trust-region, then acceptance is automatic. In addition to avoiding the discarding of valuable updates, this method eliminates the explicit trust-region radius and its heuristic mechanism, in exchange for a meaningful measure of performance. We refer to this new trust-region concept as the implicit trust-region and to the resulting method as the Implicit Riemannian Trust-Region (IRTR) method.

The description of the algorithm and the analysis of convergence consider the optimization of a smooth real function f whose domain is a differentiable manifold M with Riemannian metric g, i.e., a Riemannian manifold (M, g). Briefly, we exploit an intrinsic property of the manifold known as the tangent plane at the current iterate x, denoted by T_xM. This space, coupled with g_x, is an abstract Euclidean space (finite-dimensional, real, inner product space) where most of the effort of the solution occurs via a mapping R_x from T_xM to M (called a retraction). The retraction is used to lift the cost function from the manifold to the tangent plane, where classical algorithms may be more easily and efficiently applied.

Section 2 reviews the workings of the RTR and describes the IRTR modification. Section 3 presents the global and local convergence properties for the IRTR method. Section 4 reviews the conditions required for an efficient application of the IRTR method and demonstrates the value of the algorithm for the symmetric generalized eigenvalue problem.

2. Implicit Riemannian Trust-Region Method

This section briefly reviews the workings of the Riemannian Trust-Region (RTR) method and introduces the Implicit Riemannian Trust-Region (IRTR) method. The attempt is made to limit the amount of background material from differential geometry and Euclidean optimization. Interested readers are recommended to see Boothby (1975) or do Carmo (1992) for theory on Riemannian manifolds. Trust-region information can be found in most books on unconstrained optimization, for example Nocedal and Wright (1999) or Conn et al. (2000). Readers interested in optimization on Riemannian manifolds and the RTR are recommended Absil et al. (2007), Absil et al. (2008) and the references therein.

We assume that M is a differentiable manifold and g is a Riemannian metric on M. Together, (M,g) describes a Riemannian manifold. For all $x \in M$, the restriction

$$g_x : T_xM \times T_xM \to \mathbb{R}.$$

of g to the tangent plane T_xM defines an inner product on the vector space T_xM.

Assume that f is a real-valued differentiable function defined on M. Let $\operatorname{grad} f(x)$ and $\operatorname{Hess} f(x)$ denote the Riemannian gradient and Hessian, respectively. The Riemannian gradient $\operatorname{grad} f(x)$ is a tangent vector in T_xM, while the Riemannian Hessian $\operatorname{Hess} f(x)$ is a linear mapping from T_xM to T_xM that is symmetric under g_x.

The goal of the IRTR, like that of the RTR, is to find a local minimizer of the objective function

$$f : M \to \mathbb{R}.$$

The RTR method, like Euclidean trust-region methods, computes iterates by solving a minimization problem on a model of the objective function. However, the RTR performs this model minimization, not on the manifold M, but on the tangent bundle TM. This is achieved through the use of a mapping called a retraction.

Initially described in Shub (1986) and Adler et al. (2002) and further employed in Absil et al. (2007), a retraction maps the tangent bundle TM to the manifold M. More specifically, a retraction R on M is a differentiable mapping from TM to M such that $R_x(0_x) = x$ for all $x \in M$ (where 0_x denotes the origin of T_xM) and that $\frac{\mathrm{d}}{\mathrm{d}t} R(t\xi)\big|_{t=0} = \xi$ for all $\xi \in T_xM$.

The retraction is used to define a "lifted" cost function

$$\hat{f} = f \circ R : TM \to \mathbb{R}.$$

At a single point $x \in M$, we can restrict the domain of $\hat{f}$ to yield $\hat{f}_x = f \circ R_x : T_xM \to \mathbb{R}$. The definition of a retraction ensures that the usual optimality conditions are transferred form f to $\hat{f}$: if $\operatorname{grad} \hat{f}_x(0) = 0$ and $\operatorname{Hess} \hat{f}_x(0_x)$ is positive definite, then $\operatorname{grad} f(x) = 0$ and $\operatorname{Hess} f(x)$ is positive definite, so that x is a local minimizer of f on M. This result is fundamental to the practice of retraction-based Riemannian optimization.

It is important to note that the material that follows, since it is described for a general Riemannian manifold, is immediately applicable to standard Euclidean space. Identify $M = \mathbb{R}^d$ along with the

canonical identification $T_xM = \mathbb{R}^d$. Choose g as the canonical Euclidean inner product

$$g_x(\xi, \zeta) = \xi^T \zeta,$$

along with the exponential retraction $R_x(\eta) = \mathrm{Exp}_x \eta = x + \eta$. Then the approach described in this paper results in an implicit trust-region method for functions defined on $\mathbb{R}^d$, naturally endowed with all of the accompanying theory.

The benefit of using a retraction is that the tangent plane, coupled with the Riemannian metric g, is an abstract Euclidean space, and therefore a more familiar and convenient arena for conducting numerical optimization. The RTR method follows the example of Euclidean trust-region methods by constructing a model m_x of $\hat{f}_x$ and solving the trust-region subproblem using m_x:

$$\text{minimize } m_x(\xi), \quad \text{subject to } g_x(\xi, \xi) \leqslant \Delta^2, \tag{2.1}$$

where Δ is the trust-region radius. We assume through the paper that the model m_x is a quadratic model of $\hat{f}_x$ which approximates $\hat{f}_x$ to at least the first order:

$$m_x(\xi) = \hat{f}_x(0_x) + g_x\left(\xi, \mathrm{grad}\, \hat{f}_x(0_x)\right) + \frac{1}{2} g_x\left(\xi, H_x[\xi]\right), \tag{2.2}$$

where $H_x[\xi]$ is a symmetric operator on T_xM and 0_x is the additive identity in T_xM.

The tangent vector ξ is used to generate a new iterate, which is accepted depending on the value of the quotient

$$\rho_x(\xi) = \frac{\hat{f}_x(0_x) - \hat{f}_x(\xi)}{m_x(0_x) - m_x(\xi)}. \tag{2.3}$$

This quantity measures the ratio between the decrease in the objective function and the decrease predicted by the model. In addition to accepting/rejecting proposed iterates, $\rho_x(\xi)$ is also used to expand or shrink the trust-region radius. For convenience, the RTR algorithm is restated in Algorithm 2.1.

ALGORITHM 2.1 **Require:** Complete Riemannian manifold (M, g); scalar field f on M; retraction R
Input: $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, and $\rho' \in [0, \frac{1}{4})$, initial iterate $x_0 \in M$
Output: Sequences of iterates $\{x_k\}$

for $k = 0, 1, 2, \ldots$ **do**
  — ***Model-based Minimization*** —
  Obtain η_k by approximately solving (2.1)
  Evaluate $\rho_k = \rho_{x_k}(\eta_k)$ as in (2.3)
  — ***Adjust trust region*** —
  if $\rho_k < \frac{1}{4}$ **then**
    Set $\Delta_{k+1} = \frac{1}{4}\Delta_k$
  else if $\rho_k > \frac{3}{4}$ and $\|\eta_k\| = \Delta_k$ **then**
    Set $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
  else
    Set $\Delta_{k+1} = \Delta_k$
  end if
  — ***Compute next iterate*** —
  if $\rho_k > \rho'$ **then**
    Set $x_{k+1} = R_{x_k}(\eta_k)$

```
    else
        Set x_{k+1} = x_k
    end if
end for
```

The general algorithm does not state how (2.1) should be solved. We have previously advocated the use of the truncated conjugate gradient method of Steihaug and Toint; see Steihaug (1983) or Toint (1981) or Conn et al. (2000). This method has the benefit of requiring very little memory and returning a point inside the trust-region. It also benefits in the ability to exploit a preconditioner when solving the model minimization. Algorithm 2.2 states a preconditioned truncated conjugate gradient method for solving the model minimization on the tangent plane. Note that superscripts are used on η to differentiate between inner iterations; outer iterations of η are denoted by subscripts, as in Algorithm 2.1.

ALGORITHM 2.2 **Input:** Iterate $x \in M$, $\operatorname{grad} f(x) \neq 0$; trust-region radius Δ; convergence criteria $\kappa \in (0,1)$, $\theta > 0$; model m_x as in (2.2); symmetric/positive definite preconditioner $N : T_xM \to T_xM$

```
Set η^0 = 0_x, r_0 = grad f(x), z_0 = N^{-1} r_0, d_0 = -z_0
for j = 0, 1, 2, ... do
        — Check κ/θ stopping criterion —
    if ||r_j|| ⩽ ||r_0|| min{κ, ||r_0||^θ} then
        return η^j
    end if
        — Check curvature of current search direction —
    if g_x(H_x[d_j], d_j) ⩽ 0 then
        Compute τ > 0 such that η = η^j + τ d_j satisfies ||η||_N = Δ
        return η
    end if
        — Generate next inner iterate —
    Set α_j = g_x(z_j, r_j) / g_x(H_x[d_j], d_j)
    Set η^{j+1} = η^j + α_j d_j
        — Check trust-region —
    if ||η^{j+1}||_N > Δ then
        Compute τ > 0 such that η = η^j + τ d_j satisfies ||η||_N = Δ
        return η
    end if
        — Use CG recurrences to update residual and search direction —
    Set r_{j+1} = r_j + α_j H_x[d_j]
    Set z_{j+1} = N^{-1} r_{j+1}
    Set β_{j+1} = g_x(z_{j+1}, r_{j+1}) / g_x(z_j, r_j)
    Set d_{j+1} = -z_{j+1} + β_{j+1} d_j
end for
```

The simplest stopping criterion for Algorithm 2.2 would be to stop after a fixed number of iterations. As above, we have previously elected to stop as soon as an iteration j is reached where

$$\|r_j\| \leqslant \|r_0\| \min\left\{\kappa, \|r_0\|^{\theta}\right\}. \tag{2.4}$$

As will be discussed in Section 3.2, this strategy allows for an improved rate of convergence, by seeking linear convergence early on and superlinear convergence as the algorithm progresses.

The classical trust-region mechanism has many favorable features, including global convergence to a critical point, stable convergence only to local minimizers, and superlinear local convergence (depending on the choice of quadratic model and the method used to minimize it). These properties were retained for the RTR method; see Absil et al. (2007). The trust-region heuristic is self-tuning, such that an appropriate trust-region radius will eventually be discovered by the algorithm. In practice, however, this adjustment can result in wasted iterations, as proposed iterates are rejected do to poor scores under ρ.

We propose a modification to the trust-region method. This modification bypasses the step size heuristic and directly addresses the model performance. The implicit trust-region at x is defined as a superlevel set of ρ_x:

$$\left\{\xi \in T_xM : \rho_x(\xi) \geqslant \rho'\right\}. \tag{2.5}$$

The model minimization now consists of

$$\text{minimize } m_x(\xi), \quad \text{subject to } \rho_x(\xi) \geqslant \rho'. \tag{2.6}$$

The implicit trust-region contains exactly those points that would have been accepted by the classical trust-region mechanism. The result is that there is no trust-region radius to adjust and no explicit acceptance/rejection scheme. The IRTR algorithm is stated in Algorithm 2.3.

REMARK 2.1 A more careful examination reveals that a satisfactory value of ρ does not ensure that the next iterate produces a decrease in the objective function: an update η which increases the objective function is in the implicit trust-region as long it produces a similar increase in the model. This is in keeping with the classical trust-region presentation, which delayed the guarantee of model decrease to the discussion of global convergence, at which point it becomes necessary. Note that the implicit trust-region mechanism does ensure a decrease of any point in the trust-region, as long as there is also decrease in the model. Furthermore, the truncated conjugate gradient method recommended in this paper always produces a decrease in the model.

ALGORITHM 2.3 **Require:** Complete Riemannian manifold (M,g); scalar field f on M; retraction R
Input: $\rho' \in (0,1)$, initial iterate $x_0 \in M$
Output: Sequences of iterates $\{x_k\}$

for $k = 0,1,2,\ldots$ **do**
  — ***Model-based Minimization*** —
  Obtain η_k by approximately solving (2.6)
  — ***Compute next iterate*** —
  Set $x_{k+1} = R_{x_k}(\eta_k)$
end for

The new trust-region definition modifies the model minimization, and these modifications must be reflected in the truncated conjugate gradient solver. The trust-region definition occurs in the solver in two cases: when testing that the CG iterates remain inside the trust-region and when moving along a search direction to the edge of the trust-region. In the case of a trust-region collision during the model minimization, the truncated CG for RTR (Algorithm 2.2, lines 7 and 13) would move along the prescribed search direction to the edge of the trust-region, performing the search:

$$\text{find } \tau \geqslant 0 \text{ such that } \eta = \eta^j + \tau d_j \text{ satisfies } \|\eta\|_N.$$

Due to the simple description of the trust-region, this search is easily performed. It requires only the

solution of a quadratic equation in one variable (namely, τ); see, for example, (Conn et al., 2000, pg. 206).

The analogous operation for the implicit trust-region is the following:

$$\text{find } \tau \geqslant 0 \text{ such that } \eta = \eta^j + \tau d_j \text{ satisfies } \rho_x(\eta) = \rho'.$$

In general, this may not be as easily accomplished. Later theorems and lemmas prove the existence of satisfactory points. However, finding them may require a search of ρ_x along directions of interest. In such a case, it is desirable to relax the search, so that we require only a point inside the trust-region (not necessarily on its edge). Therefore, our presentation of truncated CG for the IRTR assumes only this. The updated truncated conjugate gradient algorithm is displayed in Algorithm 2.4.

ALGORITHM 2.4 **Input:** Iterate $x \in M$, $\operatorname{grad} f(x) \neq 0$; trust-region parameter $\rho' \in (0,1)$; convergence criteria $\kappa \in (0,1)$, $\theta \geqslant 0$; model m_x as in (2.2); symmetric/positive definite preconditioner $N : T_xM \to T_xM$

Set $\eta^0 = 0_x$, $r_0 = \operatorname{grad} f(x)$, $z_0 = N^{-1} r_0$, $d_0 = -z_0$
for $j = 0, 1, 2, \ldots$ **do**
  — ***Check κ/θ stopping criterion*** —
  if $\|r_j\| \leqslant \|r_0\| \min\left\{\kappa, \|r_0\|^\theta\right\}$ **then**
    return η^j
  end if
  — ***Check curvature of current search direction*** —
  if $g_x(H_x[d_j], d_j) \leqslant 0$ **then**
    Compute $\tau > 0$ such that $\eta = \eta^j + \tau d_j$ satisfies $\rho_x(\eta) \geqslant \rho'$
    return η
  end if
  — ***Generate next inner iterate*** —
  Set $\alpha_j = g_x(z_j, r_j) / g_x(H_x[d_j], d_j)$
  Set $\eta^{j+1} = \eta^j + \alpha_j d_j$
  — ***Check trust-region*** —
  if $\rho_x(\eta^{j+1}) < \rho'$ **then**
    Compute $\tau > 0$ such that $\eta = \eta^j + \tau d_j$ satisfies $\rho_x(\eta) \geqslant \rho'$
    return η
  end if
  — ***Use CG recurrences to update residual and search direction*** —
  Set $r_{j+1} = r_j + \alpha_j H_x[d_j]$
  Set $z_{j+1} = N^{-1} r_{j+1}$
  Set $\beta_{j+1} = g_x\left(z_{j+1}, r_{j+1}\right) / g_x(z_j, r_j)$
  Set $d_{j+1} = -z_{j+1} + \beta_{j+1} d_j$
end for

The benefit of the classical trust-region definition is that trust-region membership is easily determined, requiring only a norm calculation. The implicit trust-region, on the other hand, requires checking the value of the update vector under ρ. Furthermore, there are two occasions in the truncated CG method that require following a search direction to the edge of the trust-region. In the case of the implicit trust-region, this will not in general admit an analytical solution and may require a search of ρ along the direction of interest. In general, each evaluation of ρ will require evaluating the objective function f, which will be unallowable in many applications.

In the case that ρ admits an analytical solution, it may be possible to easily and efficiently search for a satisfactory value of ρ along a tangent vector, in order to evaluate step 12 and satisfy steps 7 and 13 of Algorithm 2.4. If there is simply an efficient method for testing or even bounding below ρ, a backtracking or binary search may be used to satisfy steps 7 and 13. Therefore, it is technically possible to apply the IRTR method to any objective function; it bears restating that the efficiency of the method is tied to the efficiency of evaluating and searching ρ.

We show in Section 4.2 that in a specific but very important application—computing the leftmost eigenvector of a generalized eigenvalue problem—the IRTR algorithm can be implemented in a remarkably efficient way and yields an algorithm that outperforms state-of-the-art methods on certain instances of the problem. In addition to providing an efficient application of the IRTR, this analysis will provide a new look at an existing eigensolver, the Trace Minimization method Sameh and Wisniewski (1982) and Sameh and Tong (2000). Before this, Section 3 will show that the IRTR inherits all of the convergence properties of the RTR.

3. Convergence Analysis for IRTR

The mechanisms of the IRTR method are sufficiently different from those of the RTR method that we must construct a separate convergence theory. We first study the global convergence properties of the IRTR method (Algorithm 2.3). As in Absil et al. (2007), we assume mild conditions on the cost function and the retraction; no assumptions are made concerning the method used to solve the model minimization (2.6), except that there is a "sufficient decrease" on the model. For the RTR and Euclidean trust-region methods, this is tied to the so-called Cauchy decrease. The modification of the trust-region definition in the IRTR scheme requires revisiting the concept of the Cauchy point, and this endeavor constitutes most of the effort in the global convergence analysis of Section 3.1.

We then analyze the convergence of the proposed method around nondegenerate local minima. Specifically, this analysis is conducted in the context of Algorithm 2.3/2.4, referring to the IRTR method where the trust-region subproblems are solved using the tCG algorithm with stopping criterion (2.4). It is shown that the iterates of the algorithm converge to nondegenerate stationary points with an order of convergence $\min(\theta+1,2)$.

3.1 *Global Convergence*

The main objective of this section is to show that the sequence $\{x_k\}$ generated by Algorithm 2.3 satisfies $\lim_{k\to\infty}\|\operatorname{grad} f(x_k)\| = 0$. This is the stronger of two global convergence results shown for the RTR presented in Absil et al. (2007).

In the discussion that follows, (M,g) is a complete Riemannian manifold of dimension d and R is a retraction on M, as defined in Absil et al. (2007). We assume that the domain of R is the whole of TM. We denote by $P_\gamma^{\tau\leftarrow 0}v$ the vector of $T_{\gamma(\tau)}M$ obtained by parallel transporting the vector $v \in T_{\gamma(0)}M$ along the curve γ. We denote by ∇ the Riemannian connection on M and by $\operatorname{dist}(x,y)$ the distance between two points on the manifold:

$$\operatorname{dist}(x,y) = \inf_\gamma \left\{ \int_0^1 \|\dot\gamma(t)\| \mathrm{d}t \right\},$$

where γ is a curve on M such that $\gamma(0) = x$ and $\gamma(1) = y$.

We define

$$\hat f : TM \to \mathbb{R} : \xi \mapsto f(R(\xi)), \tag{3.1}$$

and denote by $\hat{f}_x$ the restriction of $\hat{f}$ to T_xM, with gradient $\operatorname{grad}\hat{f}_x(0_x)$ abbreviated $\operatorname{grad}\hat{f}_x$. Recall from (2.2) that m_x has the form

$$m_x(\xi) = \hat{f}_x(0_x) + g_x\left(\xi, \operatorname{grad}\hat{f}_x\right) + \frac{1}{2} g_x\left(\xi, H_x[\xi]\right),$$

with a direction of steepest descent at the origin given by

$$p_x^{\mathrm{S}} = -\frac{\operatorname{grad}\hat{f}_x}{\|\operatorname{grad}\hat{f}_x\|}. \tag{3.2}$$

The first-order convergence results for trust-region methods typically assume that $m_{x_k}(\eta_k)$ is a sufficiently good approximation of $\hat{f}_{x_k}(\eta_k)$. In (Conn et al., 2000, Theorem 6.4.6), this is guaranteed by the assumption that the Hessian of the cost function is bounded. As in Absil et al. (2007), we will weaken this assumption and assume that the cost function is radially Lipschitz continuously differentiable. This concept is defined here for convenience.

Definition 3.1 (Radially L-C^1 Function) Let $\hat{f} : TM \to \mathbb{R}$ be as in (3.1). We say that $\hat{f}$ is radially Lipschitz continuously differentiable if there exist reals $\beta_{\mathrm{RL}} > 0$ and $\delta_{\mathrm{RL}} > 0$ such that, for all $x \in M$, for all $\xi \in T_xM$ with $\|\xi\| = 1$, and for all $t < \delta_{\mathrm{RL}}$, it holds

$$\left| \frac{d}{d\tau}\hat{f}_x(\tau\xi)\,|_{\tau=t} - \frac{d}{d\tau}\hat{f}_x(\tau\xi)\,|_{\tau=0} \right| \leqslant \beta_{\mathrm{RL}} t. \tag{3.3}$$

The main effort here regards the concept of the Cauchy point. Introduced by Powell in his early papers on the convergence of trust-region methods (Powell (1970a,b, 1975)), the Cauchy point is defined as the point inside the current trust-region which minimizes the quadratic model m_x along the direction of steepest descent of m_x. In trust-region methods employing a spherical or elliptical definitions of the trust-region, the Cauchy point is easily computed. This follows from the fact that moving along a tangent vector (in this case, the gradient of m_x) will cause you exit the trust-region only once and never re-enter it. However, for the IRTR method, depending on the function ρ_x, it may be possible to move along a tangent vector, exiting and re-entering the trust-region numerous times. Therefore, it may be difficult to compute the Cauchy point; in some cases, the Cauchy point may be at infinity.

One solution is to restrict consideration to a local trust region. Definition 3.2 defines the relevant segment along the direction of steepest descent, and Definition 3.3 defines the local Cauchy point. Theorem 3.1 describes the form of the local Cauchy point, while Theorem 3.2 gives a bound on its decrease under the model m_x. All of these results are analogous to theorems and concepts from classical trust-region theory; see Nocedal and Wright (1999); Conn et al. (2000).

Definition 3.2 (Local Trust-Region) Consider an iterate $x \in M$, $\operatorname{grad}\hat{f}_x \neq 0$, and a model m_x as in (2.2). Let ρ_x be defined as in (2.3) and let p_x^{S} be the direction of steepest descent of m_x, given in (3.2). The local trust-region along p_x^{S} is given by the following set:

$$\left\{\tau p_x^{\mathrm{S}} : 0 < \tau \leqslant \Delta_x\right\},$$

where Δ_x specifies the distance to the edge of the trust-region along p_x^{S}, given by

$$\Delta_x = \inf\left\{\tau > 0 : \rho_x\left(\tau p_x^{\mathrm{S}}\right) < \rho'\right\}. \tag{3.4}$$

The local Cauchy point will fulfill the same role as the Cauchy point, except that it is confined to the local trust-region instead of the entirety of the trust-region. The formal definition follows.

Definition 3.3 (Local Cauchy Point) Consider an iterate $x \in M$, $\operatorname{grad} \hat{f}_x \neq 0$, and a model m_x. The local Cauchy point p_x^{L} is the point

$$p_x^{\mathrm{L}} = \tau_x p_x^{\mathrm{S}}, \tag{3.5}$$

where

$$\tau_x = \operatorname*{argmin}_{0 \leqslant \tau \leqslant \Delta_x} m_x(\tau p_x^{\mathrm{S}}),$$

and where Δ_x and p_x^{S} are from Definition 3.3.

The local Cauchy point can be computed without leaving the trust-region. This makes it an attractive target when solving the trust-region subproblem using a feasible point method. In fact, the truncated conjugate gradient described earlier in the paper (Algorithm 2.4) begins with the local Cauchy point and makes reduction from there. As such, the global convergence result for IRTR will require that every solution to the trust-region subproblem produces at least as much decrease in m_x as the local Cauchy point. Therefore, we wish to describe this decrease. Before that, we present some helpful properties of the local Cauchy point.

THEOREM 3.1 Consider an iterate $x \in M$, $\operatorname{grad} \hat{f}_x \neq 0$, and $\rho' \in (0,1)$. Then the local Cauchy point takes the form

$$p_x^{\mathrm{L}} = \tau_x p_x^{\mathrm{S}},$$

where

$$\begin{aligned} \tau_x &= \begin{cases} \Delta_x, & \text{if } \gamma_x \leqslant 0 \\ \min\left\{\Delta_x, \frac{\|\operatorname{grad} \hat{f}_x\|^3}{\gamma_x}\right\} & \text{otherwise} \end{cases} \\ \gamma_x &= g_x\left(\operatorname{grad} \hat{f}_x, H_x[\operatorname{grad} \hat{f}_x]\right). \end{aligned}$$

Furthermore, if $\hat{f}_x$ is bounded below, then $\tau_x < \infty$.

Proof. Assume first that $\gamma_x \leqslant 0$. Then m_x monotonically decreases as we move along p_x^{S}, so that the minimizer along p_x^{S} inside $[0, \Delta_x]$ is $\tau_x p_x^{\mathrm{S}} = \Delta_x p_x^{\mathrm{S}}$.

Assume instead that $\gamma_x > 0$. Then m_x has a global minimizer along p_x^{S} at $\tau_* p_x^{\mathrm{S}}$, where

$$\tau_* = \frac{g_x\left(-p_x^{\mathrm{S}}, \operatorname{grad} \hat{f}_x\right)}{g_x\left(p_x^{\mathrm{S}}, H_x[p_x^{\mathrm{S}}]\right)} = \frac{\|\operatorname{grad} \hat{f}_x\|^3}{\gamma_x}.$$

If $\tau_* \in (0, \Delta_x)$, then $\tau_* = \min\{\Delta_x, \tau_*\} = \tau_x$ is the minimizer of m_x along p_x^{S} in the local trust-region, and $\tau_x p_x^{\mathrm{S}}$ is the local Cauchy point. Otherwise, $\Delta_x \leqslant \tau_*$. Note that m_x monotonically decreases along p_x^{S} between $[0, \tau_*]$, so that the minimizer of m_x along p_x^{S} between $[0, \Delta_x]$ occurs at $\Delta_x = \min\{\Delta_x, \tau_*\} = \tau_x$, and $\tau_x p_x^{\mathrm{S}}$ is the local Cauchy point.

Assume now that $\hat{f}$ is bounded below. We will show that $\tau_x < \infty$. First consider when $\gamma > 0$. We have that $\tau_x = \min\{\tau_*, \Delta_x\}$. But τ_* is finite, so that τ_x is finite as well.

Consider now that $\gamma \leqslant 0$. Assume for the purpose of contradiction that $\tau_x = \infty$. Then $\Delta_x = \infty$, and

for all $\tau > 0$, $\rho_x(\tau p_x^{\mathrm{S}}) \geqslant \rho'$. Then

$$\begin{aligned}\lim_{\tau\to\infty} \hat{f}_x(0) - \hat{f}_x(\tau p_x^{\mathrm{S}}) &= \lim_{\tau\to\infty} \rho_x(\tau p_x^{\mathrm{S}}) \left(m_x(0) - m_x(\tau p_x^{\mathrm{S}})\right) \\ &\geqslant \lim_{\tau\to\infty} \rho' \left(m_x(0) - m_x(\tau p_x^{\mathrm{S}})\right) \\ &= \infty.\end{aligned}$$

But this contradicts the assumption that $\hat{f}$ is bounded below. Therefore, our initial assumption is false and τ_x is finite. □

The next theorem concerns the decrease in m_x associated with the local Cauchy point, as described above. The proof is a straightforward modification of the classical result; see (Nocedal and Wright, 1999, Lemma 4.5) or (Conn et al., 2000, Theorem 6.3.1).

THEOREM 3.2 Take an iterate $x \in M$, $\operatorname{grad} \hat{f}_x \neq 0$, and $\rho' \in (0,1)$. Then the local Cauchy point p_x^{L} (as given in Theorem 3.1) has a decrease in m_x satisfying

$$m_x(0) - m_x(p_x^{\mathrm{L}}) \geqslant \frac{1}{2} \|\operatorname{grad} \hat{f}_x\| \min\left\{\Delta_x, \frac{\|\operatorname{grad} \hat{f}_x\|}{\|H_x\|}\right\}.$$

The last result needed before presenting the global convergence result proves that, under the radially Lipschitz continuous assumption on $\hat{f}$, our local trust-region in the direction of steepest descent always maintains a certain size. This property is necessary because the decrease in the local Cauchy point is tied to the size of the local trust-region. The local trust-region cannot be allowed to shrink to zero if we are to obtain a sufficient decrease of the model under the local Cauchy point. The following lemmas guarantee that this situation does not occur.

LEMMA 3.1 Assume that $\hat{f}$ is radially L-C^1. Assume that there exists $\beta_H \in (0,\infty)$ such that $\|H_x\| \leqslant \beta_H$ for all $x \in M$. Then for all $\rho' \in (0,1)$, there exists $\beta_\Delta > 0$ such that, for all $x \in M$, $\operatorname{grad} \hat{f}_x \neq 0$, and all $t \in (0,1]$,

$$\rho_x \left(t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} p_x^{\mathrm{S}}\right) \geqslant \rho'.$$

Proof. As a consequence of the radially L-C^1 property, we have that

$$\left|\hat{f}_x(\xi) - \hat{f}_x(0) - g_x\left(\operatorname{grad} \hat{f}_x, \xi\right)\right| \leqslant \frac{1}{2}\beta_{\mathrm{RL}} \|\xi\|^2, \tag{3.6}$$

for all $x \in M$ and all $\xi \in T_xM$ such that $\|\xi\| \leqslant \delta_{\mathrm{RL}}$.

Note that

$$\rho_x(\xi) = \frac{\hat{f}_x(0) - \hat{f}_x(\xi)}{m_x(0) - m_x(\xi)} = 1 - \frac{\hat{f}_x(\xi) - m_x(\xi)}{m_x(0) - m_x(\xi)}.$$

Let $t \in (0,1]$. Let ξ be defined

$$\xi = t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} p_x^{\mathrm{S}}.$$

Since

$$\hat{f}_x(\xi) - m_x(\xi) = \hat{f}_x(\xi) - \hat{f}_x(0) - g_x\left(\operatorname{grad} \hat{f}_x, \xi\right) - \frac{1}{2} g_x\left(\xi, H_x[\xi]\right)$$

it follows from (3.6) and from the bound on $\|H_x\|$ that

$$\begin{aligned}\left|\hat{f}_x(\xi) - m_x(\xi)\right| \leqslant{}& \frac{1}{2}\beta_{\mathrm{RL}} t^2 \min^2\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} \\ &+ \frac{1}{2}\beta_H t^2 \min^2\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\}.\end{aligned} \tag{3.7}$$

Also note that

$$m_x(0) - m_x(\xi) = t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} \|\operatorname{grad} \hat{f}_x\| - g_x(\xi, H_x[\xi])$$

and

$$\begin{aligned} |m_x(0) - m_x(\xi)| \geqslant\ & t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} \|\operatorname{grad} \hat{f}_x\| \\ & - t^2 \min{}^2\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} \beta_H. \end{aligned} \tag{3.8}$$

Then combining (3.7) and (3.8), we have

$$\begin{aligned} \frac{\left|\hat{f}_x(\xi) - m_x(\xi)\right|}{|m_x(0) - m_x(\xi)|} &\leqslant \frac{1}{2} \frac{(\beta_{\mathrm{RL}} + \beta_H) t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\}}{\|\operatorname{grad} \hat{f}_x\| - t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} \beta_H} \\ &\leqslant \frac{1}{2} \frac{(\beta_{\mathrm{RL}} + \beta_H) \beta_\Delta \|\operatorname{grad} \hat{f}_x\|}{\|\operatorname{grad} \hat{f}_x\| - \beta_\Delta \|\operatorname{grad} \hat{f}_x\| \beta_H} \\ &= \frac{1}{2} \frac{(\beta_{\mathrm{RL}} + \beta_H) \beta_\Delta}{1 - \beta_\Delta \beta_H}, \end{aligned}$$

because $t \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\} \leqslant \beta_\Delta \|\operatorname{grad} \hat{f}_x\|$. Then it is easy to see that there exists $\beta_\Delta > 0$ such that

$$\frac{1}{2} \frac{(\beta_{\mathrm{RL}} + \beta_H) \beta_\Delta}{1 - \beta_\Delta \beta_H} < 1 - \rho'.$$

□

COROLLARY 3.1 (BOUND ON Δ_x) It follows from Lemma 3.1 that, under the conditions required for the lemma, $\Delta_x \geqslant \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}\right\}$.

The convergence theory of the RTR method Absil et al. (2007) provides two results on global convergence. The stronger of these results states that the accumulation points of any series generated by the algorithm are critical points of the objective function. The definition of the implicit trust-region allows to immediately prove this result, without passing first via the weaker result. The result and approach are analogous to a classical result from Euclidean trust-region theory originally given in Shultz et al. (1985); see (Nocedal and Wright, 1999, Theorem 4.8) or (Conn et al., 2000, Theorem 6.4.6) for modern representations. Theorem 3.3 proves this for the IRTR method described in Algorithm 2.3.

THEOREM 3.3 (GLOBAL CONVERGENCE) Let $\{x_k\}$ be a sequence of iterates produced by Algorithm 2.3, each $\operatorname{grad} \hat{f}_x \neq 0$, with $\rho' \in (0,1)$. Suppose that there exists $\beta_H \in (0,\infty)$ such that each $\|H_{x_k}\| \leqslant \beta_H$. Suppose that each $\hat{f}_{x_k}$ is C^1, and that $\hat{f}$ is radially L-C^1 and bounded below on the level set

$$\{x : f(x) \leqslant f(x_0)\}.$$

Further suppose that each update η_k produces at least as much decrease in m_{x_k} as a fixed fraction of the local Cauchy point. That is, for some constant $c_1 > 0$,

$$m_{x_k}(0) - m_{x_k}(\eta_k) \geqslant c_1 \|\operatorname{grad} \hat{f}_{x_k}\| \min\left\{\Delta_{x_k}, \frac{\|\operatorname{grad} \hat{f}_{x_k}\|}{\beta_H}\right\},$$

where the terms in this inequality are from Theorem 3.2.

Then

$$\lim_{k \to \infty} \|\operatorname{grad} f(x_k)\| = 0.$$

Proof. Assume for the purpose of contradiction that the theorem does not hold. Then there exists $\varepsilon > 0$ such that, for all $K > 0$, there exists $k \geqslant K$ such that

$$\|\operatorname{grad} f(x_k)\| > \varepsilon.$$

From the workings of Algorithm 2.3,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= \hat{f}_{x_k}(0) - \hat{f}_{x_k}(\eta_k) = \rho_{x_k}(\eta_k)\left(m_{x_k}(0) - m_{x_k}(\eta_k)\right) \\ &\geqslant \rho'\left(m_{x_k}(0) - m_{x_k}(\eta_k)\right) \\ &\geqslant \rho' c_1 \|\operatorname{grad} \hat{f}_{x_k}\| \min\left\{\Delta_{x_k}, \frac{\|\operatorname{grad} \hat{f}_{x_k}\|}{\beta_H}\right\} \\ &\geqslant \rho' c_1 \|\operatorname{grad} \hat{f}_{x_k}\| \min\left\{\beta_\Delta \|\operatorname{grad} \hat{f}_x\|, \delta_{\mathrm{RL}}, \frac{\|\operatorname{grad} \hat{f}_{x_k}\|}{\beta_H}\right\}, \end{aligned}$$

where the last inequality results from Corollary 3.1. Then for all $K > 0$, there exists $k \geqslant K$ such that

$$f(x_k) - f(x_{k+1}) \geqslant \rho' c_1 \varepsilon \min\left\{\beta_\Delta \varepsilon, \delta_{\mathrm{RL}}, \frac{\varepsilon}{\beta_H}\right\} > 0.$$

But because f is bounded below and decreases monotonically with the iterates produced by the algorithm, we know that

$$\lim_{k\to\infty} (f(x_k) - f(x_{k+1})) = 0,$$

and we have reached a contradiction. Hence, our original assumption must be false, and the desired result is achieved. □

3.2 *Local Convergence*

The local convergence results for the IRTR require significantly less modification from the RTR than did the global convergence results. For the sake of brevity, only original proofs will be provided. Neglected proofs may be found in Absil et al. (2007).

First, we ask one additional constraint be placed upon the retraction, in addition to the definition of retraction from Absil et al. (2007). This is that that there exists some $\mu > 0$ and δ_μ such that

$$\|\xi\| \geqslant \mu \operatorname{dist}(x, R_x(\xi)), \quad \text{for all } x \in M, \text{ for all } \xi \in T_xM, \|\xi\| \leqslant \delta_\mu. \tag{3.9}$$

In particular, the exponential retraction satisfies (3.9) as an equality, with $\mu = 1$. The bound is also satisfied when R is smooth and M is compact.

We will state a few preparatory lemmas before moving on the to local convergence results.

LEMMA 3.2 (TAYLOR) Let $x \in M$, let V be a normal neighborhood of x, and let ζ be a C^1 tangent vector field on M. Then, for all $y \in V$,

$$P_\gamma^{0\leftarrow 1}\zeta_y = \zeta_x + \nabla_\xi \zeta + \int_0^1 \left(P_\gamma^{0\leftarrow\tau}\nabla_{\gamma'(\tau)}\zeta - \nabla_\xi \zeta\right) d\tau,$$

where γ is the unique minimizing geodesic satisfying $\gamma(0) = x$ and $\gamma(1) = y$, and $\xi = \operatorname{Exp}_x^{-1} y = \gamma'(0)$.

LEMMA 3.3 Let $v \in M$ and let f be a C^2 cost function such that $\operatorname{grad} f(v) = 0$ and $\operatorname{Hess} f(v)$ is positive definite with maximal and minimal eigenvalues $\lambda_{\max}$ and $\lambda_{\min}$. Then, given $c_0 < \lambda_{\min}$ and $c_1 > \lambda_{\max}$, there exists a neighborhood V of v such that, for all $x \in V$, it holds that

$$c_0 \operatorname{dist}(v,x) \leqslant \|\operatorname{grad} f(x)\| \leqslant c_1 \operatorname{dist}(v,x).$$

The first local convergence result states that the nondegenerate local minima are attractors of Algorithm 2.3/2.4. This theorem is unmodified from the same result for the RTR; see (Absil et al., 2007, Theorem 4.12), which itself is closely related to the Capture Theorem (Bertsekas, 1995, Theorem 1.2.5).

THEOREM 3.4 (LOCAL CONVERGENCE TO LOCAL MINIMIMA) Consider Algorithm 2.3/2.4–i.e., the Implicit Riemannian Trust-Region algorithm where the trust-region subproblem (2.1) is solved using the modified truncated CG algorithm–with all the assumptions of Theorem 3.3 (Global Convergence). Let v be a nondegenerate local minimizer of f, i.e., $\operatorname{grad} f(v) = 0$ and $\operatorname{Hess} f(v)$ is positive definite. Assume that $x \to \|H_x^{-1}\|$ is bounded on a neighborhood of v and that (3.9) holds for some $\mu > 0$ and $\delta_\mu > 0$. Then there exists a neighborhood V of v such that, for all $x_0 \in V$, the sequence $\{x_k\}$ generated by Algorithm 2.3/2.4 converges to v.

Now we study the order of convergence of the sequences that converge to a nondegenerate local minimizer. This result is the same as for the RTR; see (Absil et al., 2007, Theorem 4.13). However, the proof is slightly modified. The previous proof showed that the trust-region eventually becomes inactive as a stopping condition on the truncated CG; this requires review under the new trust-region definition.

THEOREM 3.5 (ORDER OF LOCAL CONVERGENCE) Consider Algorithm 2.3/2.4. Suppose that R is C^2 retraction, that f is a C^2 cost function on M, and that

$$\|H_{x_k} - \operatorname{Hess} \hat{f}_{x_k}(0_{x_k})\| \leqslant \beta_H \|\operatorname{grad} f(x_k)\|, \tag{3.10}$$

that is, H_{x_k} is a sufficiently good approximation of $\operatorname{Hess} \hat{f}_{x_k}(0_{x_k})$. Let $v \in M$ be a nondegenerate local minimizer of f, (i.e., $\operatorname{grad} f(v) = 0$ and $\operatorname{Hess} f(v)$ is positive definite). Further assume that $\operatorname{Hess} \hat{f}_x(0_x)$ is Lipschitz-continuous at 0_x uniformly in a neighborhood of v, i.e., there exist β_{L2}, $\delta_1 > 0$ and $\delta_2 > 0$ such that, for all $x \in B_{\delta_1}(v)$ and all $\xi \in B_{\delta_2}(0_x)$, there holds

$$\|\operatorname{Hess} \hat{f}_x(\xi) - \operatorname{Hess} \hat{f}_x(0_x)\| \leqslant \beta_{\mathrm{L2}} \|\xi\|. \tag{3.11}$$

Then there exists $c \geqslant 0$ such that, for all sequences $\{x_k\}$ generated by the algorithm converging to v, there exists $K > 0$ such that for all $k > K$,

$$\operatorname{dist}(x_{k+1}, v) \leqslant c\,(\operatorname{dist}(x_k, v))^{\min\{\theta+1,2\}}.$$

Proof. We will show below that there exist $\tilde{\Delta}$, c_0, c_1, c_2, c_3, c_3', c_4, and c_5 such that, for all sequences $\{x_k\}$ satisfying the conditions asserted, all $x \in M$, all ξ with $\|\xi\| \leqslant \tilde{\Delta}$, and all k greater than some K, there holds

$$c_0 \operatorname{dist}(v, x_k) \leqslant \|\operatorname{grad} f(x_k)\| \leqslant c_1 \operatorname{dist}(v, x_k), \tag{3.12}$$

$$\|\eta_k\| \leqslant c_4 \|\operatorname{grad} m_{x_k}(0_{x_k})\| \leqslant \tilde{\Delta}, \tag{3.13}$$

$$\|\operatorname{grad} f(R_{x_k}(\xi))\| \leqslant c_5 \|\operatorname{grad} \hat{f}_{x_k}(\xi)\|, \tag{3.14}$$

$$\|\operatorname{grad} m_{x_k}(\xi) - \operatorname{grad} \hat{f}_{x_k}(\xi)\| \leqslant c_3 \|\xi\|^2 + c_3' \|\operatorname{grad} f(x_k)\| \|\xi\|, \tag{3.15}$$

$$\|\operatorname{grad} m_{x_k}(\eta_k)\| \leqslant c_2 \|\operatorname{grad} m_{x_k}(0)\|^{\theta+1}, \tag{3.16}$$

where $\{\eta_k\}$ is the sequence of update vectors corresponding to $\{x_k\}$. With these results at hand, the proof is concluded as follows. For all $k > K$, it follows from (3.12) that

$$c_0 \mathrm{dist}(v, x_{k+1}) \leqslant \|\operatorname{grad} f(x_{k+1})\| = \|\operatorname{grad} f(R_{x_k}(\eta_k))\|,$$

and from (3.14) that

$$\|\operatorname{grad} f(R_{x_k}(\eta_k))\| \leqslant c_5 \|\operatorname{grad} \hat{f}_{x_k}(\eta_k)\|,$$

and from (3.13) and (3.15) and (3.16) that

$$\begin{aligned}\|\operatorname{grad} \hat{f}_{x_k}(\eta_k)\| &\leqslant \|\operatorname{grad} m_{x_k}(\eta_k) - \operatorname{grad} \hat{f}_{x_k}(\eta_k)\| + \|\operatorname{grad} m_{x_k}(\eta_k)\| \\ &\leqslant (c_3 c_4^2 + c_3' c_4)\|\operatorname{grad} m_{x_k}(0)\|^2 + c_2 \|\operatorname{grad} m_{x_k}(0)\|^{\theta+1},\end{aligned}$$

and from (3.12) that

$$\|\operatorname{grad} m_{x_k}(0)\| = \|\operatorname{grad} f(x_k)\| \leqslant c_1 \mathrm{dist}(v, x_k).$$

Consequently, taking K larger if necessary so that $\mathrm{dist}(v, x_k) < 1$ for all $k > K$, it follows that

$$\begin{aligned}c_0 \mathrm{dist}(v, x_{k+1}) &\leqslant \|\operatorname{grad} f(x_{k+1})\| \\ &\leqslant c_5 (c_3 c_4^2 + c_3' c_4)\|\operatorname{grad} f(x_k)\|^2 + c_5 c_2 \|\operatorname{grad} f(x_k)\|^{\theta+1} \\ &\leqslant c_5 ((c_3 c_4^2 + c_3' c_4) c_1^2 (\mathrm{dist}(v, x_k))^2 + c_2 c_1^{\theta+1} (\mathrm{dist}(v, x_k))^{\theta+1}) \\ &\leqslant c_5 ((c_3 c_4^2 + c_3' c_4) c_1^2 + c_2 c_1^{\theta+1}) (\mathrm{dist}(v, x_k))^{\min\{2, \theta+1\}}\end{aligned}$$

for all $k > K$, which is the desired result. It remains to prove the bounds (3.12)-(3.16).

Equation (3.12) comes from Lemma 3.3 and is due to the fact that v is a nondegenerate critical point. Equations (3.13)-(3.15) are proved in Absil et al. (2007).

It remains only to prove (3.16). Let γ_k denote $\|\operatorname{grad} f(x_k)\|$. It follows from the definition of ρ_k that

$$\rho_k - 1 = \frac{m_{x_k}(\eta_k) - \hat{f}_{x_k}(\eta_k)}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)}. \tag{3.17}$$

From Taylor's theorem (3.2), there holds

$$\hat{f}_{x_k}(\eta_k) = \hat{f}_{x_k}(0_{x_k}) + g_{x_k}(\operatorname{grad} f(x_k), \eta_k) + \int_0^1 g_{x_k}\left(\mathrm{Hess}\hat{f}_{x_k}(\tau\eta_k)[\eta_k], \eta_k\right)(1-\tau)d\tau.$$

It follows that

$$\begin{aligned}\left|m_{x_k}(\eta_k) - \hat{f}_{x_k}(\eta_k)\right| &= \left|\int_0^1 \left(g_{x_k}\left(H_{x_k}[\eta_k], \eta_k\right) - g_{x_k}\left(\mathrm{Hess}\hat{f}_{x_k}(\tau\eta_k)[\eta_k], \eta_k\right)\right)(1-\tau)d\tau\right| \\ &\leqslant \int_0^1 \left|g_{x_k}\left((H_{x_k} - \mathrm{Hess}\hat{f}_{x_k}(0_{x_k}))[\eta_k], \eta_k\right)\right|(1-\tau)d\tau \\ &\quad + \int_0^1 \left|g_{x_k}\left((\mathrm{Hess}\hat{f}_{x_k}(0_{x_k}) - \mathrm{Hess}\hat{f}(\tau\eta_k))[\eta_k], \eta_k\right)\right|(1-\tau)d\tau \\ &\leqslant \frac{1}{2}\beta_H \gamma_k \|\eta_k\|^2 + \frac{1}{6}\beta_{\mathrm{L2}}\|\eta_k\|^3.\end{aligned}$$

It then follows from (3.17), using the bound on the Cauchy decrease, that

$$\|\rho_k - 1\| \leqslant \frac{(3\beta_H \gamma_k + \beta_{\mathrm{L2}}\|\eta_k\|)\|\eta_k\|^2}{6\gamma_k \min\{\Delta_k, \gamma_k/\beta\}},$$

where β is an upper bound on the norm of H_{x_k}. Since $\Delta_k \geqslant \min\{\beta_\Delta \gamma_k, \delta_{\mathrm{RL}}\}$ (Corollary 3.1) and $\lim_{k\to\infty} \gamma_k = 0$ (in view of Theorem 3.3), we can choose K large enough that $\Delta_k \geqslant \beta_\Delta \gamma_k$, for all $k > K$. This and $\|\eta_k\| \leqslant c_4 \gamma_k$ yield

$$\|\rho_k - 1\| \leqslant \frac{(3\beta_H + \beta_{\mathrm{L2}} c_4) c_4^2 \gamma_k^3}{6 \min\left\{\beta_\Delta, \frac{1}{\beta}\right\} \gamma_k^2}.$$

Since $\lim_{k\to\infty} \gamma_k = 0$, it follows that $\lim_{k\to\infty} \rho_k = 1$.

Therefore, the trust-region eventually becomes inactive as a stopping criterion for the truncated CG. Furthermore, because $\{x_k\}$ converges to v and $\operatorname{Hess} f(v)$ is positive definite, it follows that H_{x_k} is positive definite for all k greater than a certain K. This eliminates negative curvature of the Hessian as a stopping criterion for truncated CG.

This means that the truncated CG loop terminates only after sufficient reduction has been made in $\|\operatorname{grad} m_{x_k}(\eta_k)\|$ with respect to $\|\operatorname{grad} m_{x_k}(0_{x_k})\|$:

$$\|\operatorname{grad} m_{x_k}(\eta_k)\| \leqslant \|\operatorname{grad} m_{x_k}(0_{x_k})\|^{\theta+1},$$

(choosing K large enough that $\|\operatorname{grad} m_{x_k}(0_{x_k})\|^\theta < \kappa$ for all $k > K$), or the model minimization has been solved exactly, in which case $\operatorname{grad} m_{x_k}(\eta_k) = 0$. In either case, we have satisfied (3.16). □

4. Applications

In this section, we review and discuss the essential ingredients for applying the IRTR-tCG method, Algorithm 2.3/Algorithm 2.4. We then apply the IRTR method to the solution of generalized symmetric eigenvalue problems. Numerical experiments show that the IRTR modification can improve on the efficiency of the classical trust-region mechanism.

4.1 *Checklist*

The following ingredients are required for applying the IRTR method to optimizing a cost function f on a Riemannian manifold (M, g):

1. a tractable numerical representation for points x on M, for tangent vectors in T_xM, and for the inner products $g_x(\cdot,\cdot)$ on T_xM,

2. a tractable retraction $R_x : T_xM \to M$,

3. formulas for $f(x)$, $\operatorname{grad} f(x)$, and an approximate Hessian $H_x[\xi]$ that satisfy the properties required for convergence in Section 3,

4. an efficient formula for evaluating or bounding $\rho_x(\xi)$ and an efficient method for searching along $\rho_x(t\xi)$, as needed by steps 7, 12 and 13 of Algorithm 2.4.

The first three of these are requirements of the RTR method, with the third being required of trust-region methods in general. The fourth requirement is unique to the IRTR.

All trust-region methods assume the ability to evaluate ρ for the purpose of accepting/rejecting candidate iterates, as well as updating the trust-region radius. However, the occurrence of this evaluation is relatively rare, occurring once per outer iteration. Each evaluation of ρ in general requires evaluating the objective function f. The rarity of evaluating the objective function is one of the attractions of trust-region methods; for many problems, evaluating the objective function is significantly more expensive than evaluating the surrogate m.

However, in the case that ρ can be efficiently computed, either directly via f and m or indirectly via some other formula or bound, it is possible to implement the IRTR. With simply an efficient formula for ρ, Armijo-style backtracking searches can be employed to find a point in the implicit trust-region which satisfy sufficient decrease conditions. This is similar to the technique employed in the Generic RTR package (Baker et al. (2007)). This is an important point, as the global convergence result Theorem 3.3 requires a sufficient decrease with respect to the local Cauchy point.

The application considered in the next section provide formulas for evaluating ρ which are sufficiently efficient to demonstrate the benefits of the IRTR method.

4.2 *Application: Extreme Symmetric Generalized Eigenspaces*

The generalized eigenvalue problem is often used as an example in Riemannian optimization, because of its familiarity, its importance in numerous application, and its position as a function over a non-trivial Riemannian manifold; see Helmke and Moore (1994); Edelman et al. (1998); Lundström and Eldén (2002); Absil et al. (2002, 2004, 2007). In this section, we will demonstrate the applicability of the IRTR for the solution of generalized eigenvalue problems. Along the way, our analysis will provide a novel derivation of the Trace Minimization method of Sameh and Wisniewski (1982) and Sameh and Tong (2000). Furthermore, this problem will illustrate the potential efficiency of the IRTR method over the RTR method.

Given two $n \times n$ matrices, λ is an eigenvalue if there exists a non-zero vector v such that

$$Av = Bv\lambda.$$

If A is symmetric and B is symmetric/positive definite then the generalized eigenvalue problem is said to be symmetric/positive definite. In this case, the eigenvalues are all real and the eigenvectors are B-orthogonal (and can be chosen B-orthonormal).

Let the eigenvalues of the pencil (A,B) be $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$. Consider the p leftmost eigenvalues, $\lambda_1, \ldots, \lambda_p$, and corresponding eigenvectors, $v_1, \ldots, v_p$. We will assume below, though it is not strictly necessary, that $\lambda_p < \lambda_{p+1}$. It is known that the $n \times p$ matrix containing the leftmost eigenvectors is a global minimizer of the generalized Rayleigh quotient

$$f : \mathbb{R}_*^{n\times p} \to \mathbb{R} : X \mapsto \operatorname{trace}\left((X^T BX)^{-1}(X^T AX)\right),$$

where $\mathbb{R}_*^{n\times p}$ is the set of $n \times p$ real matrices of full column rank.

It is easily shown that the generalized Rayleigh quotient depends only on the subspace spanned by the columns of X, denoted $\operatorname{colsp}(X)$. Therefore, f induces a real-valued function on the set of p-dimensional subspaces of $\mathbb{R}^n$. This set is the Grassmann manifold $\operatorname{Grass}(p,n)$, and it can be endowed with a Riemannian structure.

As in Absil et al. (2004, 2007), we will treat the Grassmann manifold $\mathrm{Grass}(p,n)$ as the quotient manifold $\mathbb{R}_*^{n\times p}/\mathrm{GL}_p$ of the noncompact Stiefel manifold by the set of transformations that preserves column space. In this approach, a subspace in $\mathrm{Grass}(p,n)$ is represented by any $n\times p$ matrix whose columns span the subspace. A real function h on $\mathrm{Grass}(p,n)$ is represented by its lift $h_\uparrow(X) = h(\mathrm{colsp}(X))$. This flexibility in representing points on the Grassmann means that some extra consideration has to be made when looking for a unique representation of tangent vectors. To represent a tangent vector ξ to $\mathrm{Grass}(p,n)$ at a point $\mathscr{X} = \mathrm{colsp}(X)$, we will define a horizontal space $\mathscr{H}_X$. Then ξ is uniquely represented by its horizontal lift $\xi_{\uparrow X}$, which is in turn defined by the following conditions: (i) $\xi_{\uparrow X} \in \mathscr{H}_X$ and (ii) $\mathrm{D}\, h(\mathscr{X})[\xi] = \mathrm{D}\, h_\uparrow(X)[\xi_{\uparrow X}]$ for all real functions h on $\mathrm{Grass}(p,n)$. In this way, the horizontal space $\mathscr{H}_X$ represents the tangent space $T_{\mathscr{X}}\mathrm{Grass}(p,n)$.

To simplify the derivation of the gradient and Hessian of the Rayleigh cost function, we define the horizontal space at X as

$$\mathscr{H}_X = \left\{ Z \in \mathbb{R}^{n\times p} : Z^T BX = 0 \right\} \tag{4.1}$$

and employ the non-canonical Riemannian metric

$$g_{\mathscr{X}}(\xi,\zeta) = \mathrm{trace}\left((X^T BX)^{-1} \xi_{\uparrow X}^T \zeta_{\uparrow X} \right), . \tag{4.2}$$

We use the retraction

$$R_{\mathscr{X}}(\xi) = \mathrm{colsp}(X + \xi_{\uparrow X}) \tag{4.3}$$

The objective function is the generalized Rayleigh quotient, defined from this point forward as follows:

$$f : \mathrm{Grass}(p,n) \to \mathbb{R} : \mathrm{colsp}(X) \mapsto \mathrm{trace}\left((X^T BX)^{-1} (X^T AX) \right). \tag{4.4}$$

The retraction is used to lift this function from the manifold to the tangent plane, yielding

$$\hat{f} : T\mathrm{Grass}(p,n) \to \mathbb{R} : \xi \mapsto f(R(\xi)), \tag{4.5}$$

and, as before, $\hat{f}_{\mathscr{X}}$ is this function restricted to $T_{\mathscr{X}}\mathrm{Grass}(p,n)$.

We denote by P_{BX} the orthogonal projector onto the horizontal space $\mathscr{H}_X$:

$$P_{BX} := I - BX(X^T B^2 X)^{-1} X^T B$$

An expansion of $\hat{f}_{\mathscr{X}}$ yields:

$$\begin{aligned}
\hat{f}_{\mathscr{X}}(\xi) &= \mathrm{trace}\left(\left((X+\xi_{\uparrow X})^T B (X+\xi_{\uparrow X}) \right)^{-1} (X+\xi_{\uparrow X})^T A (X+\xi_{\uparrow X}) \right) \\
&= \mathrm{trace}\left((X^T BX)^{-1} X^T AX \right) + 2\mathrm{trace}\left((X^T BX)^{-1} \xi_{\uparrow X}^T AX \right) \\
&\quad + \mathrm{trace}\left((X^T BX)^{-1} \xi_{\uparrow X}^T \left(A\xi_{\uparrow X} - B\xi_{\uparrow X} (X^T BX)^{-1} X^T AX \right) \right) + HOT \\
&= \mathrm{trace}\left((X^T BX)^{-1} X^T AX \right) + 2\mathrm{trace}\left((X^T BX)^{-1} \xi_{\uparrow X}^T P_{BX} AX \right) \\
&\quad + \mathrm{trace}\left((X^T BX)^{-1} \xi_{\uparrow X}^T P_{BX} \left(A\xi_{\uparrow X} - B\xi_{\uparrow X} (X^T BX)^{-1} X^T AX \right) \right) + HOT,
\end{aligned} \tag{4.6}$$

where the introduction of the projectors does not modify the expression since $P_{BX}\xi_{\uparrow X} = \xi_{\uparrow X}$. Then using the Riemannian metric (4.2), we can make the following identifications:

$$(\mathrm{grad}\, f(\mathscr{X}))_{\uparrow X} = \left(\mathrm{grad}\, \hat{f}_{\mathscr{X}}(0_{\mathscr{X}}) \right)_{\uparrow X} = 2P_{BX}AX, \tag{4.7}$$

$$\left(\mathrm{Hess}\hat{f}_{\mathscr{X}}(0_{\mathscr{X}})[\xi] \right)_{\uparrow X} = 2P_{BX}\left(A\xi_{\uparrow X} - B\xi_{\uparrow X}(X^T BX)^{-1} X^T AX \right). \tag{4.8}$$

An efficient implementation of the implicit RTR requires an understanding of the improvement ratio ρ, repeated here:

$$\rho_{\mathscr{X}}(\xi) = \frac{\hat{f}_{\mathscr{X}}(0_{\mathscr{X}}) - \hat{f}_{\mathscr{X}}(\xi)}{m_{\mathscr{X}}(0_{\mathscr{X}}) - m_{\mathscr{X}}(\xi)},$$

for all $\xi \in T_{\mathscr{X}}\mathrm{Grass}(p,n)$ and where $m_{\mathscr{X}}$ is the quadratic model chosen to approximate $\hat{f}_{\mathscr{X}}$:

$$m_{\mathscr{X}}(\xi) = f(\mathscr{X}) + g_{\mathscr{X}}\left(\mathrm{grad}\, f(\mathscr{X}), \xi\right) + \frac{1}{2} g_{\mathscr{X}}\left(H_{\mathscr{X}}[\xi], \xi\right). \tag{4.9}$$

Note that the model Hessian $H_{\mathscr{X}}$ has been left unspecified, as its effect on $m_{\mathscr{X}}$, and therefore on $\rho_{\mathscr{X}}$, may cause us to prefer one form over another. In the discussion that follows, we will examine two choices for the model Hessian.

4.2.1 *Case 1:* TRACEMIN *Model* We assumed above the matrices A and B are both symmetric and that B is positive definite. Consider the case now where A is positive semi-definite. Consider also the following choice for the model Hessian:

$$\left(H_{\mathscr{X}}[\xi]\right)_{\uparrow X} = 2P_{BX}AP_{BX}\xi_{\uparrow X}.$$

This operator is symmetric/positive definite, hence the model (2.2) admits a unique unconstrained minimizer. For simplicity, assume also that the basis X representing $\mathscr{X}$ is B-orthonormal (this is easily enforced in the retraction). The trust-region subproblem (2.6) now consists of the following:

$$\begin{aligned} &\text{minimize trace}\left(X^TAX + 2\xi_{\uparrow X}^T P_{BX}AX + \xi_{\uparrow X}^T P_{BX}AP_{BX}\xi_{\uparrow X}\right), \\ &\text{such that } \xi_{\uparrow X}^T BX = 0 \text{ and } \rho_{\mathscr{X}}(\xi) \geqslant \rho'. \end{aligned} \tag{4.10}$$

If we neglect the trust-region requirement, then we are left with the following problem:

$$\begin{aligned} &\text{minimize trace}\left(X^TAX + 2\xi_{\uparrow X}^T P_{BX}AX + \xi_{\uparrow X}^T P_{BX}AP_{BX}\xi_{\uparrow X}\right), \\ &\text{such that } \xi_{\uparrow X}^T BX = 0. \end{aligned} \tag{4.11}$$

This precisely is one of the approaches suggested for the Trace Minimization Algorithm (TRACEMIN) in Sameh and Wisniewski (1982) and Sameh and Tong (2000). The authors show in those works the following inequality:

$$\begin{aligned} \hat{f}_{\mathscr{X}}(\xi) &= \text{trace}\left(\left(I + \xi_{\uparrow X}\right)^T B\xi_{\uparrow X}\right)^{-1}\left(X + \xi_{\uparrow X}\right)^T A\left(X + \xi_{\uparrow X}\right)\Big) \\ &\leqslant \text{trace}\left(\left(X + \xi_{\uparrow X}\right)^T A\left(X + \xi_{\uparrow X}\right)\right) \\ &= m_{\mathscr{X}}(\xi). \end{aligned} \tag{4.12}$$

Recall from the model definition that $m_{\mathscr{X}}(0_{\mathscr{X}}) = \hat{f}_{\mathscr{X}}(0_{\mathscr{X}})$. Inserting this into Equation (4.12), we yield the following:

$$\hat{f}_{\mathscr{X}}(0_{\mathscr{X}}) - \hat{f}_{\mathscr{X}}(\xi) \geqslant m_{\mathscr{X}}(0_{\mathscr{X}}) - m_{\mathscr{X}}(\xi).$$

Then any $\xi \in T_{\mathscr{X}}\mathrm{Grass}(p,n)$ produces at least as much decrease in the objective function as in the model. Returning to the context of the implicit trust-region, this means that ρ satisfies the following:

$$\rho_{\mathscr{X}}(\xi) = \frac{\hat{f}_{\mathscr{X}}(0_{\mathscr{X}}) - \hat{f}_{\mathscr{X}}(\xi)}{m_{\mathscr{X}}(0_{\mathscr{X}}) - m_{\mathscr{X}}(\xi)} \geqslant 1.$$

As a result of this and the assumption that $\rho' \leqslant 1$, the implicit trust region (2.6) is the whole of $T_{\mathscr{X}}\mathrm{Grass}(p,n)$, and the solution of (4.11) is the unique solution of (4.10). In this way, the Trace Minimization method is equivalent to the Implicit Riemannian Trust-Region method for a particular choice of the model Hessian. The Trace Minimization method inherits the convergence analysis of the IRTR, in addition to that provided by its authors.

Note that because the entirety of the tangent plane lies inside the implicit trust-region and because the model Hessian is positive definite, the truncated CG algorithm will terminate only when it has sufficiently reduced the gradient of the model, i.e., the residual of the linear system

$$H_{\mathscr{X}}[\xi] = -\operatorname{grad} f(\mathscr{X}).$$

Then these methods could be described as quasi-Newton approaches, with the added benefit that the implicit trust-region mechanism provides stable convergence only to local minimizers.

However, an unfortunate consequence of the Hessian choice $(H_{\mathscr{X}}[\xi])_{\uparrow X} = 2P_{BX}AP_{BX}\xi_{\uparrow X}$ is that it does not adequately approximate the actual Hessian of $\hat{f}$. As a result, the method yields only a linear rate of convergence. This result was known by the authors of TRACEMIN, due to the relationship between optimal TRACEMIN and the subspace iteration method; see Sameh and Wisniewski (1982) or Sameh and Tong (2000). The approach in the following subsection addresses the slow convergence by using a more accurate model Hessian.

4.2.2 *Case 2: Newton Model* Relax the TRACEMIN assumption that A is positive definite. Consider the case where the quadratic model $m_{\mathscr{X}}$ is chosen as the Newton model, i.e., the quadratic Taylor expansion of $\hat{f}_{\mathscr{X}}$:

$$m_{\mathscr{X}}(\xi) = f(\mathscr{X}) + g_{\mathscr{X}}(\operatorname{grad} f(\mathscr{X}), \xi) + \frac{1}{2} g_{\mathscr{X}}\left(\operatorname{Hess}\hat{f}_{\mathscr{X}}(0_{\mathscr{X}})[\xi], \xi\right).$$

We wish to perform an analysis of $\rho_{\mathscr{X}}$ for the Newton model just as we did for the TRACEMIN model. Assume as before that $\mathscr{X}$ is represented by a B-orthonormal basis, i.e. $X^TBX = I$. Take some tangent vector $\eta \in T_{\mathscr{X}}\mathrm{Grass}(p,n)$. Consider the denominator of $\rho_{\mathscr{X}}(\eta)$:

$$\begin{aligned} m_{\mathscr{X}}(0_{\mathscr{X}}) - m_{\mathscr{X}}(\eta) &= -g_{\mathscr{X}}(\operatorname{grad} f(\mathscr{X}), \eta) - \frac{1}{2} g_{\mathscr{X}}(H_{\mathscr{X}}[\eta], \eta) \\ &= -2\operatorname{trace}\left(\eta_{\uparrow X}^T AX\right) - \operatorname{trace}\left(\eta_{\uparrow X}^T A\eta_{\uparrow X} - \eta_{\uparrow X}^T B\eta_{\uparrow X} X^T AX\right) \\ &= \operatorname{trace}\left(\eta_{\uparrow X}^T B\eta_{\uparrow X} X^T AX - 2\eta_{\uparrow X}^T AX - \eta_{\uparrow X}^T A\eta_{\uparrow X}\right) \\ &= \operatorname{trace}\left(\hat{M}\right), \end{aligned} \tag{4.13}$$

for $\hat{M} = \eta_{\uparrow X}^T B\eta_{\uparrow X} X^T AX - 2\eta_{\uparrow X}^T AX - \eta_{\uparrow X}^T A\eta_{\uparrow X}$. Consider the numerator:

$$\begin{aligned} \hat{f}_{\mathscr{X}}(0_{\mathscr{X}}) - \hat{f}_{\mathscr{X}}(\eta) &= f(\operatorname{colsp}(X)) - f(\operatorname{colsp}(X + \eta_{\uparrow X})) \\ &= \operatorname{trace}\left(X^TAX - \left((X+\eta_{\uparrow X})^T B(X+\eta_{\uparrow X})\right)^{-1}(X+\eta_{\uparrow X})^T A(X+\eta_{\uparrow X})\right) \\ &= \operatorname{trace}\left(\left(I + \eta_{\uparrow X}^T B\eta_{\uparrow X}\right)^{-1}\left(\eta_{\uparrow X}^T B\eta_{\uparrow X} X^T AX - 2\eta_{\uparrow X}^T AX - \eta_{\uparrow X}^T A\eta_{\uparrow X}\right)\right) \\ &= \operatorname{trace}\left((I + \eta_{\uparrow X}^T B\eta_{\uparrow X})^{-1}\hat{M}\right). \end{aligned} \tag{4.14}$$

Combining Equations (4.13) and (4.14) allows $\rho_{\mathscr{X}}(\eta)$ to be written as follows:

$$\rho_{\mathscr{X}}(\eta) = \frac{\operatorname{trace}\left((I + \eta_{\uparrow X}^T B \eta_{\uparrow X})^{-1}\hat{M}\right)}{\operatorname{trace}\left(\hat{M}\right)}. \tag{4.15}$$

Note that in the specific case of $p = 1$, i.e., the solution for a single eigenpair, the Equation (4.15) simplifies to

$$\rho_{\mathscr{X}}(\eta) = \frac{1}{1 + \eta_{\uparrow X}^T B \eta_{\uparrow X}}. \tag{4.16}$$

This formula provides both of the ingredients for an efficient implementation of Algorithm 2.3/2.4: a trivial evaluation of $\rho_{\mathscr{X}}(\eta)$, and an efficient search along η for some $\rho_{\mathscr{X}}(t\eta) = \rho'$. Contrast this with the formula (4.15), which currently defies efficient evaluation and/or search.

This formula for $\rho_{\mathscr{X}}$ enables the two actions required to efficiently implement the IRTR: an efficient method for evaluating $\rho_{\mathscr{X}}(\eta)$; and the ability to efficiently move along a search direction to the edge of the trust-region. The result can be thought of as an inexact Newton iteration along with stopping criterion that ensure strong global convergence results and a fast rate of local convergence.

This technique, like that resulting from applying the RTR to this problem, has many similarities to the Jacobi-Davidson method Sleijpen and Van der Vorst (1996). In Notay (2002), the author developed an analysis which (inexpensively) provides knowledge of the residual of the outer (eigenvalue) iteration based on the conjugate gradient coefficients used to solve the Jacobi-Davidson correction equation. Notay suggests exploiting this information as a stopping criterion for the inner iteration. His suggestion involves stopping the inner iteration when the marginal decrease in the outer residual norm is less than some fraction of the marginal decrease in the inner residual norm. The implicit trust-region, on the other hand, is comprised of strictly those points where the decrease under the objective function is some fraction of the decrease of the quadratic model. In this regard, both approaches strive to stop the inner iteration when it becomes inefficient or irrelevant with regard to the outer iteration, though the IRTR does this in a way that yields strong global convergence results.

4.2.3 *Numerical Results* This section illustrates the potential efficiency of the IRTR method over the RTR method for the problem of computing the leftmost eigenpair of a symmetric/positive definite matrix pencil. The IRTR is also compared against the LOBPCG method from Knyazev (2001), as implemented in Hetmaniuk and Lehoucq (2006). This method was chosen because it implements a state-of-the-art optimization-oriented, CG-based eigensolver. Both methods were implemented in C++ using the Anasazi eigensolver package of the Trilinos package (see Baker et al. (2005) and Heroux et al. (2003)). Tests were conducted in serial (i.e., one processor).

The pencil used for experimentation derives from a finite element discretization (with linear basis functions) of a one-dimensional Laplacian. The parameter n refers to the number of elements in the discretization. The parameter ρ' is the acceptance parameter for the RTR and the trust-region parameter for IRTR. IRTR was evaluated for multiple values of ρ', to illustrate the effect of the parameter on the efficiency of the method. Table 1 lists the results of the comparison.

This testing shows that the IRTR has the potential to exceed the performance of the RTR, while maintaining competitiveness against methods designed specifically for solving this class of problems. One benefit of the IRTR method is to reduce the number of parameters controlling the iteration by eliminating those parameters corresponding to the trust-region radius. However, the performance parameter ρ' still must be selected by the user. The results in Table 1 indicate that the performance of the method

is tied to this parameter. Future research and experiments are necessary to determine how this parameter should be selected.

5. Concluding remarks

We presented the Implicit Riemannian Trust-Region method, a modification to the classical trust-region mechanism. This work shows that the explicit radius of the classical trust-region mechanism can be relaxed, as long as the objective function decrease is some fixed fraction of the model decrease. The resulting method has the potential for increased performance.

The algorithm was described in the context of Riemannian optimization. The Riemannian setting adds a small amount of overhead relative to a Euclidean setting, especially for those readers not familiar with Riemannian geometry. On the other hand, a Riemannian manifold structure can be thought of as the most basic structure that the optimization domain must possess for smooth optimization techniques to be applicable. We believe that the greater generality of the approach is worth the effort.

The capability for increased performance was demonstrated via the computation of an extreme eigenvector of a symmetric matrix pencil. On this problem, the algorithm outperformed the RTR method as well as a state-of-the-art eigensolver. Future research is necessary to identify other applications appropriate for the IRTR method.

ACKNOWLEDGMENTS Useful discussions with Andreas Stathopoulos and Denis Ridzal are gratefully acknowledged. We are indebted to the referees for their careful reading and many helpful comments.

REFERENCES

Absil, P.-A., Mahony, R., and Sepulchre, R. (2008). Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ.

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

Absil, P.-A., Mahony, R., and Sepulchre, R. (2004). Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. Acta Appl. Math., **80**(2), 199–220.

Absil, P.-A., Baker, C. G., and Gallivan, K. A. (2007). Trust-region methods on Riemannian manifolds. Foundations of Computational Mathematics, **7**(3), 303–330.

Adler, R. L., Dedieu, J.-P., Margulies, J. Y., Martens, M., and Shub, M. (2002). Newton's method on Riemannian manifolds and a geometric model for the human spine. IMA J. Numer. Anal., **22**(3), 359–390.

TABLE 1 *Experimental comparison of* IRTR, RTR *and* LOBPCG *methods. n denotes the problem size, while the numbers in the table indicate runtime in seconds. The parameter to RTR and IRTR denotes the value of* ρ'.

n	RTR(.1)	IRTR(.1)	IRTR(.45)	IRTR(.90)	LOBPCG
100	0.040	0.038	**0.027**	0.031	0.103
500	0.227	0.448	0.298	**0.185**	2.414
1,000	0.524	0.596	0.592	**0.383**	1.794
10,000	36.38	54.36	**27.66**	42.25	130.3
50,000	**850**	2985	1221	1202	3819

Baker, C. G., Hetmaniuk, U. L., Lehoucq, R. B., and Thornquist, H. K. (2005). The Anasazi block eigensolvers package. See http://software.sandia.gov/trilinos/packages/anasazi/.

Baker, C. G., Absil, P.-A., and Gallivan, K. A. (2007). Generic Riemannian trust-region package. Website: http://www.scs.fsu.edu/∼cbaker/GenRTR/.

Bertsekas, D. P. (1995). Nonlinear Programming. Athena Scientific, Belmont, Massachusetts.

Boothby, W. M. (1975). An introduction to Differentiable Manifolds and Riemannian Geometry. Academic Press.

Conn, A. R., Gould, N. I. M., and Toint, P. L. (2000). Trust-region methods. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, and Mathematical Programming Society (MPS), Philadelphia, PA.

Dedieu, J.-P., Priouret, P., and Malajovich, G. (2003). Newton's method on Riemannian manifolds: convariant alpha theory. IMA J. Numer. Anal., **23**(3), 395–419.

do Carmo, M. P. (1992). Riemannian geometry. Mathematics: Theory & Applications. Birkhäuser Boston Inc., Boston, MA. Translated from the second Portuguese edition by Francis Flaherty.

Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl., **20**(2), 303–353.

Gabay, D. (1982). Minimizing a differentiable function over a differential manifold. Journal of Optimization Theory and Applications, **37**(2), 177–219.

Gould, N. I. M., Sainvitu, C., and Toint, P. L. (2005). A filter-trust-region method for unconstrained optimization. SIAM J. Optimization, **16**(2), 341–357.

Helmke, U. and Moore, J. B. (1994). Optimization and Dynamical Systems. Springer.

Heroux, M., Bartlett, R., Howle, V., Hoekstra, R., Hu, J., Kolda, T., Lehoucq, R., Long, K., Pawlowski, R., Phipps, E., Salinger, A., Thornquist, H., Tuminaro, R., Willenbring, J., and Williams, A. (2003). An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories.

Hetmaniuk, U. and Lehoucq, R. (2006). Basis selection in LOBPCG. J. Comput. Phys., **218**(1), 324–332.

Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. SIAM J. Sci. Comput., **23**(2), 517–541.

Luenberger, D. G. (1972). The gradient projection method along geodesics. Management Sci., **18**, 620–631.

Lundström, E. and Eldén, L. (2002). Adaptive eigenvalue computations using Newton's method on the Grassmann manifold. SIAM J. Matrix Anal. Appl., **23**(3), 819–839.

Nocedal, J. and Wright, S. J. (1999). Numerical Optimization. Springer Series in Operations Research. Springer-Verlag, New York.

Notay, Y. (2002). Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem. Numer. Linear Algebra Appl., **9**(1), 21–44.

Powell, M. J. D. (1970a). A hybrid method for nonlinear equations. In P. Rabinowitz, editor, Numerical Methods for Nonlinear Algebraic Equations, pages 87–114. Gordon and Breach, London.

Powell, M. J. D. (1970b). A new algorithm for unconstrained optimization. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, editors, Nonlinear Programming, pages 31–65. Academic Press, London.

Powell, M. J. D. (1975). Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, Nonlinear Programming 2, pages 1–27. Academic Press, London.

Sameh, A. and Tong, Z. (2000). The trace minimization method for the symmetric generalized eigenvalue problem. J. Comput. Appl. Math., **123**, 155–175.

Sameh, A. H. and Wisniewski, J. A. (1982). A trace minimization algorithm for the generalized eigenvalue problem. SIAM J. Numer. Anal., **19**(6), 1243–1259.

Shub, M. (1986). Some remarks on dynamical systems and numerical analysis. In L. Lara-Carrero and J. Lewowicz, editors, Proc. VII ELAM., pages 69–92. Equinoccio, U. Simón Bolívar, Caracas.

Shultz, G. A., Schnabel, R. B., and Byrd, R. H. (1985). A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. SIAM J. Numer. Anal., **22**(1), 47–67.

Sleijpen, G. L. G. and Van der Vorst, H. A. (1996). A Jacobi-Davidson iteration method for linear eigenvalue

problems. SIAM J. Matrix Anal. Appl., **17**(2), 401–425.

Smith, S. T. (1993). Geometric optimization methods for adaptive filtering. Ph.D. thesis, Division of Applied Sciences, Harvard University, Cambridge, Massachusetts.

Smith, S. T. (1994). Optimization techniques on Riemannian manifolds. In Hamiltonian and gradient flows, algorithms and control, volume 3 of Fields Inst. Commun., pages 113–136. Amer. Math. Soc., Providence, RI.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal., **20**, 626–637.

Toint, P. L. (1981). Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, Sparse Matrices and Their Uses, pages 57–88. Academic Press, London.

Udrişte, C. (1994). Convex functions and optimization methods on Riemannian manifolds. Kluwer Academic Publishers.