

# A Riemannian approach to low-rank algebraic Riccati equations\*

Bamdev Mishra<sup>1</sup> and Bart Vandereycken<sup>2</sup>

**Abstract**—We propose a Riemannian optimization approach for computing low-rank solutions of the algebraic Riccati equation. The scheme alternates between fixed-rank optimization and rank-one updates. The fixed-rank optimization is on the set of fixed-rank symmetric positive definite matrices which is endowed with a particular Riemannian metric (and geometry) that is tuned to the structure of the cost function. We specifically discuss the implementation of a Riemannian trust-region algorithm that is potentially scalable to large-scale problems. The rank-one update is based on a descent direction that ensures a monotonic decrease of the cost function. Preliminary numerical results on standard small-scale benchmarks show that we obtain solutions to the Riccati equation at lower ranks than the standard approaches.

## I. INTRODUCTION

We look at the following low-rank algebraic Riccati equation in  $\mathbf{X} \in \mathbb{R}^{n \times n}$

$$\mathbf{A}^T \mathbf{X} + \mathbf{X} \mathbf{A} + \mathbf{X} \mathbf{B} \mathbf{B}^T \mathbf{X} = \mathbf{C}^T \mathbf{C}, \quad (1)$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is a full rank matrix,  $\mathbf{B} \in \mathbb{R}^{n \times p}$ , and  $\mathbf{C} \in \mathbb{R}^{s \times n}$  and  $s, p \ll n$ . The solution of (1) is expected to be symmetric positive semidefinite and low-rank (the rank is  $\ll n$ ) and is of specific interest in a number of disciplines as noted in [SSM13], [LS13], [BS13] and references therein. Uniqueness of the solution results from additional assumptions that  $-\mathbf{A}$  and  $-\mathbf{A} - \mathbf{B} \mathbf{B}^T \mathbf{X}$  are stable, i.e., all their eigenvalues have negative real part [SSM13].

Although conventional solvers for solving (1), e.g., Matlab's ARE function, exist, they are computationally expensive and cost at least  $O(n^3)$ . To circumvent the scaling issue in large-scale problems (large  $n$ ), low-rank solvers are of particular interest. Two state-of-the-art approaches that are tuned to the problem (1) are the Kleinman inexact approach [BS13, Section 5] and the projection type Krylov subspace approach or the Galerkin Projection approach [BS13, Section 3], [SSM13]. The Kleinman inexact approach relaxes the quadratic nature of the equation (1) into a sequence linear Lyapunov equations which are solved approximately by a low-rank Lyapunov solver at every iteration to guarantee a Newton like convergence to the solution. The projection type approach, on the other hand, is based on a series of

smaller subspace projections. At each step, the equation (1) is projected onto a smaller subspace which is then solved using standard solver like the Matlab's ARE. The subspace is incremented in dimension as iterations progress to get accurate solutions. Finally, the rank of the solution is truncated to obtain low-rank solutions.

In contrast, we view computing a low-rank solution of (1) as computing a solution (critical point) of the problem

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}^{n \times n}} \quad & \frac{1}{4} \|\mathbf{A}^T \mathbf{X} + \mathbf{X} \mathbf{A} + \mathbf{X} \mathbf{B} \mathbf{B}^T \mathbf{X} - \mathbf{C}^T \mathbf{C}\|_F^2 \\ \text{subject to} \quad & \mathbf{X} \succeq 0, \end{aligned} \quad (2)$$

where  $\mathbf{X} \succeq 0$  is the positive semidefiniteness constraint and  $\|\cdot\|_F$  is the Frobenius norm of a matrix. The quantity  $\|\mathbf{A}^T \mathbf{X} + \mathbf{X} \mathbf{A} + \mathbf{X} \mathbf{B} \mathbf{B}^T \mathbf{X} - \mathbf{C}^T \mathbf{C}\|_F$  is referred to as *residual error*. To find a critical point of (2), we propose a scheme that alternates between fixed-rank optimization and rank-one updates [JBAS10].

The fixed-rank reformulation of (2) is defined as

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}^{n \times n}} \quad & \frac{1}{4} \|\mathbf{A}^T \mathbf{X} + \mathbf{X} \mathbf{A} + \mathbf{X} \mathbf{B} \mathbf{B}^T \mathbf{X} - \mathbf{C}^T \mathbf{C}\|_F^2 \\ \text{subject to} \quad & \mathbf{X} = \mathbf{S}_+(r, n), \end{aligned} \quad (3)$$

where  $\mathbf{S}_+(r, n)$  denotes the set of rank- $r$  symmetric positive semidefinite matrices of size  $n \times n$ . We tackle the above fixed-rank problem in the framework of Riemannian optimization [EAS98], [AMS08]. The Riemannian optimization framework embeds the constraint into the search space thereby providing an unconstrained optimization setup on the nonlinear search space  $\mathbf{S}_+(r, n)$ . A few Riemannian approaches on  $\mathbf{S}_+(r, n)$  are discussed in [JBAS10], [VV10], [MBS11] which also list various *ingredients* that enables us to minimize any smooth cost function in a numerically efficient manner.

A critical component in invoking the Riemannian framework on the search space is the choice of a *Riemannian metric*, a smoothly varying inner product. Selecting the metric leads to equipping the search space with a *Riemannian structure*. Choice of the metric has profound impact on the performance of optimization algorithms. Tuning the metric then amounts to *preconditioning* the optimization problems by incorporating the Hessian information in the metric. Designing a tuned Riemannian metric on the search space by taking a weighted  $L_2$  norm (using a symmetric positive definite part of the Hessian of a specific problem) has been explored in [MAAS12]. This is, for example, done for the low-rank matrix completion problem with good success in [MAAS12], [NS12]. For the problem (3) of interest, we exploit the parameterization of the search space discussed

\*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. This work was also partly supported by the Belgian FRFC (Fonds de la Recherche Fondamentale Collective). Bamdev Mishra is a research fellow of the Belgian National Fund for Scientific Research (FNRS).

<sup>1</sup>Department of Electrical Engineering and Computer Science, University of Liège, Liège, 4000 Belgium B.Mishra@ulg.ac.be

<sup>2</sup>Department of Mathematics, Fine Hall, Princeton University, USA Bartv@math.princeton.edu

in [JBAS10] and follow the developments in [MAAS12] to design a novel Riemannian metric in section II-B for (3).

As an algorithm for the fixed-rank problem (3), we implement a Riemannian trust-region algorithm that has a provably quadratic rate of convergence near the optimum (refer [AMS08, Chapter 7] for a convergence analysis). In addition, we combine this fixed-rank optimization with a rank increasing outer iteration. The overall algorithm converges superlinearly to a critical point of (2). This provides a way to keep a tighter control over the rank of the solution while better minimizing the residual error.

Our main contribution is a meta scheme, shown in Table I, for (1) that is based on a novel Riemannian metric (6) for the fixed-rank optimization problem (3), discussed in section II. The optimization-related ingredients that are required to implement an off-the-shelf Riemannian trust-region algorithm [AMS08, Section 7] for (3) are discussed in section II-C. In particular, the numerical complexity (per iteration) of this approach is discussed in section II-D which shows that our proposed algorithm is potentially scalable to large problems. Preliminary simulations show encouraging results on standard small-scale benchmarks where we obtain lower residual errors at lower ranks with our scheme than the standard approaches.

## II. THE PROPOSED RIEMANNIAN GEOMETRY ON THE SET OF FIXED-RANK SYMMETRIC POSITIVE SEMIDEFINITE MATRICES

Any rank- $r$  symmetric positive semidefinite matrix  $\mathbf{X} \in \mathbb{R}^{n \times n}$  is parameterized as

$$\mathbf{X} = \mathbf{Y}\mathbf{Y}^T,$$

where  $\mathbf{Y} \in \mathbb{R}_*^{n \times r}$ , the set of full column rank matrices of size  $n \times r$ . This factorization, however, is not unique as  $\mathbf{X}$  remains unchanged under the transformation  $\mathbf{Y} \mapsto \mathbf{Y}\mathbf{O}$  for all  $\mathbf{O} \in \mathcal{O}(r)$ , where  $\mathcal{O}(r)$  is the set of matrices of size  $r \times r$  such that  $\mathbf{O}\mathbf{O}^T = \mathbf{O}^T\mathbf{O} = \mathbf{I}$ . The search space is, therefore, identified with the quotient space  $S_+(r, n) \simeq \mathbb{R}_*^{n \times r} / \mathcal{O}(r)$ . In other words, the search space is the set of equivalence classes  $[\mathbf{Y}] = \{\mathbf{Y}\mathbf{O} : \mathbf{O} \in \mathcal{O}(r)\}$ , where  $\mathbf{Y} \in \mathbb{R}_*^{n \times r}$ .

Based on the above observation, we reformulate the optimization problem (3) into an optimization problem of a smooth function  $\phi : \mathcal{M} \rightarrow \mathbb{R}$  on a quotient manifold [EAS98], [AMS08]

$$\begin{aligned} \min_x \quad & \phi(x) \\ \text{subject to} \quad & x \in \mathcal{M} = \overline{\mathcal{M}} / \mathcal{O}(r), \end{aligned} \quad (4)$$

where  $\overline{\mathcal{M}} = \mathbb{R}_*^{n \times r}$  (the set of full column rank matrices of size  $n \times r$ ) is the computational space,  $\mathcal{O}(r)$  is the set of orthogonal matrices of size  $r \times r$ . We represent an element of the total space  $\overline{\mathcal{M}}$  by  $\bar{x}$  and its corresponding equivalence class by  $x$  such that  $x = [\bar{x}]$ , i.e.,  $\bar{x}$  has the matrix representation  $\mathbf{Y}$  and  $x$  refers to the equivalence class  $[\mathbf{Y}]$ . Similarly, the function  $\phi : \mathcal{M} \rightarrow \mathbb{R}$  on the quotient manifold  $\mathcal{M}$  is induced by the function  $\bar{\phi} : \overline{\mathcal{M}} \rightarrow \mathbb{R} : \bar{x} \mapsto \bar{\phi}(\bar{x})$  on the total space  $\overline{\mathcal{M}}$ . For our case  $\bar{\phi}(\bar{x}) = \|\mathbf{A}^T\mathbf{Y}\mathbf{Y}^T + \mathbf{Y}\mathbf{Y}^T\mathbf{A} +$

$\mathbf{Y}\mathbf{Y}^T\mathbf{B}\mathbf{B}^T\mathbf{Y}\mathbf{Y}^T - \mathbf{C}^T\mathbf{C}\|_F^2/4$  which is obtained from the cost function in (3) with the parameterization  $\mathbf{X} = \mathbf{Y}\mathbf{Y}^T$ .

In this section, first, we derive a novel tuned Riemannian metric on  $\overline{\mathcal{M}}$  which induces a Riemannian metric on the quotient space  $\mathcal{M}$ . Second, the metric structure leads to concrete ideas of implementing a trust-region algorithm on the quotient manifold  $\mathcal{M}$  for which we list all the required ingredients. Third, we discuss the per-iteration numerical complexity of the trust-region algorithm and show the potential scalability of the setup.

### A. A symmetric positive definite Hessian approximation

Our tuned metric is based on preconditioning the Hessian by a symmetric positive definite approximation of it. Computation of the symmetric positive definite approximation of the Hessian of the cost function  $\bar{\phi}(\mathbf{Y}) = \|\mathbf{A}^T\mathbf{Y}\mathbf{Y}^T + \mathbf{Y}\mathbf{Y}^T\mathbf{A} + \mathbf{Y}\mathbf{Y}^T\mathbf{B}\mathbf{B}^T\mathbf{Y}\mathbf{Y}^T - \mathbf{C}^T\mathbf{C}\|_F^2/4$  follows from the computation of the gradient  $\text{Grad}_{\mathbf{Y}}\bar{\phi}$  at  $\mathbf{Y} \in \mathbb{R}_*^{n \times r}$  and its directional derivative in the direction  $\mathbf{Z} \in \mathbb{R}^{n \times r}$ . Shown below are few steps that lead to identifying a symmetric positive definite part of the Hessian.

$$\begin{aligned} \text{DGrad}_{\mathbf{Y}}\bar{\phi}[\mathbf{Z}] &= (\mathbf{A}\mathbf{A}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Y}) \\ &+ \mathbf{Z}(\mathbf{Y}^T\mathbf{A}\mathbf{A}^T\mathbf{Y}) \\ &+ (\mathbf{B}\mathbf{B}^T\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{B}\mathbf{B}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Y}) \\ &+ \mathbf{Z}(\mathbf{Y}^T\mathbf{B}\mathbf{B}^T\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{B}\mathbf{B}^T\mathbf{Y}) + \text{other terms} \\ \Leftrightarrow \text{Euclidean Hess}[\text{vec}(\mathbf{Z})] &= \underbrace{((\mathbf{Y}^T\mathbf{Y}) \otimes \mathbf{A}\mathbf{A}^T)}_{\text{positive definite}} \text{vec}(\mathbf{Z}) \\ &+ \underbrace{(\mathbf{Y}^T\mathbf{A}\mathbf{A}^T\mathbf{Y} \otimes \mathbf{I})}_{\text{positive definite}} \text{vec}(\mathbf{Z}) \\ &+ \underbrace{((\mathbf{Y}^T\mathbf{Y}) \otimes (\mathbf{B}\mathbf{B}^T\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{B}\mathbf{B}^T))}_{\text{positive semidefinite}} \text{vec}(\mathbf{Z}) \\ &+ \underbrace{((\mathbf{Y}^T\mathbf{B}\mathbf{B}^T\mathbf{Y}\mathbf{Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{B}\mathbf{B}^T\mathbf{Y}) \otimes \mathbf{I})}_{\text{positive semidefinite}} \text{vec}(\mathbf{Z}) \\ &+ (\text{other negative semidefinite terms})\text{vec}(\mathbf{Z}), \end{aligned} \quad (5)$$

where  $\otimes$  denotes the Kronecker product of matrices,  $\text{vec}(\cdot)$  vectorizes a matrix by stacking the columns of the matrix on top of each other, and  $\text{DGrad}_{\mathbf{Y}}\bar{\phi}[\mathbf{Z}]$  is the standard Euclidean directional derivative of  $\text{Grad}_{\mathbf{Y}}\bar{\phi}$  in the direction  $\mathbf{Z}$ , i.e.,  $\text{DGrad}_{\mathbf{Y}}\bar{\phi}[\mathbf{Z}] = \lim_{t \rightarrow 0} (\text{Grad}_{\mathbf{Y}+t\mathbf{Z}}\bar{\phi} - \text{Grad}_{\mathbf{Y}}\bar{\phi})/t$ . It should be stated that when  $\mathbf{B} = 0$ , the approximation in (5) extracts the *dominant* component of the Hessian.

### B. The proposed Riemannian metric

Our choice of the metric is a weighted  $L_2$  metric using the above symmetric positive semidefinite approximation of the Hessian. Consequently, the proposed metric  $\bar{g}_{\bar{x}} : T_{\bar{x}}\overline{\mathcal{M}} \times T_{\bar{x}}\overline{\mathcal{M}} \rightarrow \mathbb{R}$  on  $\overline{\mathcal{M}}$  at  $\bar{x}$  is  $\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\zeta}_{\bar{x}}) = \text{vec}(\bar{\xi}_{\bar{x}})\mathcal{L}\text{vec}(\bar{\zeta}_{\bar{x}})$ , where  $\mathcal{L}$  is the symmetric positive semidefinite approximation of the Hessian derived in (5),  $\text{vec}(\cdot)$  vectorizes a matrix, and  $\bar{\xi}_{\bar{x}}, \bar{\zeta}_{\bar{x}}$  are any vectors in the tangent space  $T_{\bar{x}}\overline{\mathcal{M}}$  (the linearization of the manifold  $\overline{\mathcal{M}}$  at  $\bar{x}$ ) at  $\bar{x} \in \overline{\mathcal{M}}$  with the matrix representation  $\bar{x} = \mathbf{Y} \in \mathbb{R}_*^{n \times r}$  and  $T_{\bar{x}}\overline{\mathcal{M}} = \mathbb{R}^{n \times r}$  [AMS08, Example 3.6.4]. Equivalently in matrix form, the

metric proposed is

$$\bar{g}_x(\bar{\xi}_x, \bar{\zeta}_x) = \text{Trace}(\bar{\xi}_x^T \mathbf{A}_1 \bar{\zeta}_x \mathbf{M}_1) + \text{Trace}(\bar{\xi}_x^T \bar{\zeta}_x \mathbf{M}_2), \quad (6)$$

where the auxiliary variables  $\mathbf{M}_1 = \mathbf{Y}^T \mathbf{Y} \succ 0$ ,  $\mathbf{A}_1 = (\mathbf{A}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T \mathbf{Y}\mathbf{Y}^T \mathbf{Y}\mathbf{Y}^T \mathbf{B}\mathbf{B}^T) \succ 0$ , and  $\mathbf{M}_2 = (\mathbf{Y}^T \mathbf{A}\mathbf{A}^T \mathbf{Y} + \mathbf{Y}^T \mathbf{B}\mathbf{B}^T \mathbf{Y}\mathbf{Y}^T \mathbf{Y}\mathbf{Y}^T \mathbf{B}\mathbf{B}^T \mathbf{Y}) \succ 0$  are introduced as shorthand notations.

Following [JBAS10], [AMS08], it can be readily checked that the proposed metric (6) respects the invariance by the group action of  $\mathcal{O}(r)$ , induces a Riemannian metric on the quotient space  $\mathcal{M}$ , and gives a Riemannian submersion structure to  $\mathcal{M}$ . Invariance with respect to the action of  $\mathcal{O}(r)$  is critical to define a valid metric on the quotient space  $\mathcal{M}$  [AMS08, Section 3.6.2]. Observe that the metric (6) can be interpreted as a *preconditioner* for the Hessian of  $\bar{\phi}$  since the metric is proposed from the Hessian information.

### C. Ingredients of a trust-region algorithm on $S_+(r, n)$

Once the Riemannian geometry on  $S_+(r, n) \simeq \mathcal{M} = \overline{\mathcal{M}}/\mathcal{O}(r)$  is decided, it is conceptually straightforward to implement a Riemannian trust-region algorithm on  $S_+(r, n)$  following [EAS98], [AMS08]. At each iteration, the trust-region algorithm builds a locally quadratic model around  $x$  and minimizes the function in a neighborhood to obtain a candidate search direction [AMS08, Chapter 7]. In the Riemannian setup, we list the following ingredients that enable us to implement a Riemannian trust-region algorithm [AMS08, Algorithm 10 and Section 7.5.1].

- Matrix representation of an element  $x$  on the quotient manifold  $\mathcal{M}$  and its tangent space  $T_x \mathcal{M}$  at  $x$ . For  $\mathcal{M} = \overline{\mathcal{M}}/\mathcal{O}(r)$ , these are identified from the matrix representations in the computational space  $\overline{\mathcal{M}}$ . See [AMS08, Section 3.6.2].
- A way to “move” on the manifold given a search direction  $\xi_x \in T_x \mathcal{M}$ . This is accomplished with a retraction mapping [AMS08, Definition 4.1.1] on  $\mathcal{M}$  that maps a tangent vector onto the manifold. The retraction mapping on  $\mathcal{M} = \overline{\mathcal{M}}/\mathcal{O}(r)$  is shown in [AMS08, Example 4.1.5].
- Matrix representation of the Riemannian gradient of  $\phi$ .
- Matrix representation of a Riemannian connection on the manifold that captures the *covariant derivative* of a vector field  $\xi_x$  in the direction of a vector field  $\eta_x$ . Once the Riemannian connection is defined, the application of Riemannian Hessian along a vector field is directly obtained in terms of the Riemannian connection of the Riemannian gradient along that vector field.

### D. Numerical complexity

A carefully study of the ingredients mentioned in section II-C reveals that all operations cost  $O(|\mathbf{A}|r + nr^2 + r^3)$ , except the computation of the Riemannian gradient and the Riemannian Hessian operator, where  $|\mathbf{A}|$  is the number of non-zero elements in  $\mathbf{A}$ . Each computation of the Riemannian gradient and the Riemannian Hessian operator needs (only once for every iteration) to solve the linear system

$$\mathbf{A}_1 \xi \mathbf{M}_1 + \xi \mathbf{M}_2 = \mathbf{Z}, \quad \xi \in \mathbb{R}^{n \times r} \quad (7)$$

TABLE I  
A META SCHEME FOR (1).

Given	<ul style="list-style-type: none"> <li>• Initialize <math>r</math> to <math>r_0</math>, say <math>r_0 = 1</math>.</li> <li>• Initialize the iterate <math>\mathbf{Y}_0 \in \mathbb{R}_*^{n \times r_0}</math>.</li> </ul>
Scheme	We alternate between the following two steps until convergence.
Step i)	Compute a stationary point $\mathbf{Y} \in \mathbb{R}_*^{n \times r}$ of the fixed-rank optimization problem (3) with the Riemannian trust-region algorithm proposed in section II initialized from $\mathbf{Y}_0$ .
Step ii)	Update the rank to $r + 1$ and initialize $\mathbf{Y}_0 = [\mathbf{Y} \ tu]$ , where $u \in \mathbb{R}^n$ is the descent direction proposed in section III and $t > 0$ is an appropriate step-size computed by backtracking

where  $\mathbf{Z} \in \mathbb{R}^{n \times r}$  is given and the auxiliary variables  $\mathbf{A}_1, \mathbf{M}_1, \mathbf{M}_2 \succ 0$  are defined in section II-B. By means of the generalized eigenvalue decomposition of  $(\mathbf{M}_1, \mathbf{M}_2)$ , the system (7) can be transformed into  $r$  decoupled linear systems. It has an overall computational cost of  $O(C_{\text{solve}}r + nr^2 + r^3)$ , where  $C_{\text{solve}}$  is the cost of solving shifted systems of  $\mathbf{A}_1$  which has a *sparse + low-rank* structure. In many large-scale problems,  $C_{\text{solve}}$  can be obtained in  $O(n)$ , although a robust numerical implementation is work in progress. Finally, the computational cost per iteration of the trust-region algorithm is  $O(C_{\text{solve}}r + nr^2 + r^3)$ .

## III. A META-SCHEME FOR THE ALGEBRAIC LOW-RANK RICCATI EQUATION

We propose the meta scheme shown in Table I for (1) that alternates between fixed-rank optimization (with a trust-region algorithm) and rank-one updates [JBAS10]. The scheme monotonically decreases the residual error. Convergence of this scheme to a critical point of (2) is established directly from the analysis in [JBAS10]. The rank-one update is based on finding a *descent* search direction that decreases the residual error. If  $\mathbf{Y} \in \mathbb{R}_*^{n \times r}$  is the output of the Riemannian trust-region algorithm that minimizes (3), then the descent direction is obtained computing the eigenvector  $u \in \mathbb{R}^n$  corresponding to the smallest eigenvalue of the *gradient* of the residual  $\mathbf{A}^T \mathbf{X} + \mathbf{X}\mathbf{A} + \mathbf{X}\mathbf{B}\mathbf{B}^T \mathbf{X} - \mathbf{C}^T \mathbf{C}$  at  $\mathbf{X} = \mathbf{Y}\mathbf{Y}^T$ , which is of rank  $4r + 2s$ , where  $s$  is the number of columns of  $\mathbf{C}$ . The cost of computing the descent direction is  $O(|\mathbf{A}|r + nr^2 + r^3)$ , where  $|\mathbf{A}|$  is the cardinality of the matrix  $\mathbf{A}$ . The initialization iterate for the rank  $r + 1$  subproblem is obtained by the concatenation  $[\mathbf{Y} \ tu] \in \mathbb{R}^{n \times (r+1)}$  where  $t > 0$  is an appropriate step-size that produces a sufficient decrease in the residual error.

## IV. NUMERICAL COMPARISONS

To illustrate the notions presented in the paper, we implement our proposed scheme of Table I on the open-source Matlab toolbox Manopt [BMAS14]. The toolbox provides an off-the-shelf implementation of the Riemannian trust-region algorithm. Our implementation is available from <http://www.montefiore.ulg.ac.be/~mishra/codes/Riccati.html>. As stopping criteria for our scheme, the fixed-rank optimization is stopped when the norm of the Riemannian gradient norm is below  $10^{-10}$  and the rank-one updating is stopped when the relative residual  $\|\mathbf{A}^T \mathbf{X} + \mathbf{X}\mathbf{A} + \mathbf{X}\mathbf{B}\mathbf{B}^T \mathbf{X} - \mathbf{C}^T \mathbf{C}\|_F / \|\mathbf{C}^T \mathbf{C}\|_F$  is

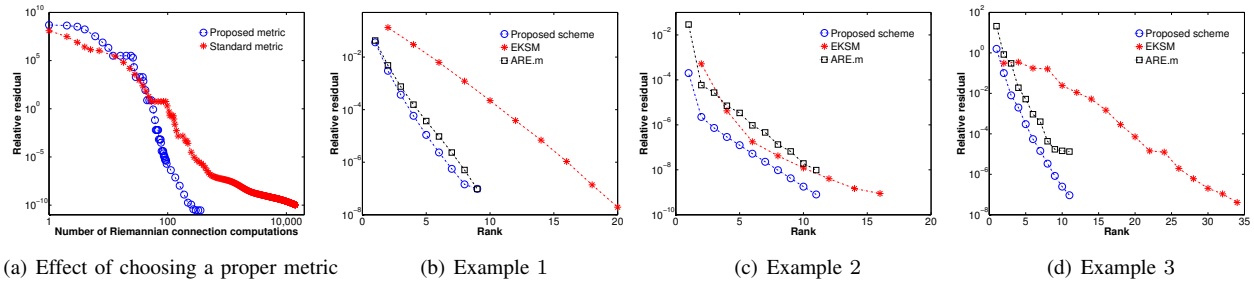


Fig. 1. The benefits of having a tailored Riemannian metric and an optimization-based approach to (1). The proposed scheme in Table I leads to smaller relative residual errors at lower ranks for each of the three considered examples. The values on the ARE plot at different ranks are obtained by taking the truncated singular value decomposition of the solution from the Matlab function ARE.

less than  $10^{-7}$ . The relative residual is computed efficiently by exploiting the low-rank structure [LS13, Section 5.3]. For the Riemannian trust-region algorithm, we also limit the number of inner iterations (to solve the trust-region subproblem) to 30 and the number of outer iterations to 500.

We first show the effectiveness of the proposed Riemannian metric (6) vis-a-vis the choice of the standard (not tuned) metric in [JBAS10] which is  $\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\zeta}_{\bar{x}}) = \text{Trace}(\bar{\xi}_{\bar{x}}^T \bar{\zeta}_{\bar{x}})$ , where  $\bar{\zeta}_{\bar{x}}, \bar{\xi}_{\bar{x}}$  are any vectors in the tangent space at  $\bar{x} \in \mathcal{M}$ . To this end, we compare the Riemannian trust-region implementations for (3) on a smaller scale Riccati equation corresponding to [LS13, Example 7.1] with  $(n, r) = (100, 5)$ . Figure 1(a) clearly shows the benefits of a tuned metric that results in a fewer number of Riemannian connection computations (note the log-log scale). The conclusion remains the same across other instances.

We also compare our scheme with the Matlab function ARE and state-of-the-art EKS.M algorithm of [SSM13] that is based on the Galerkin Projection method. Three examples are considered. Example 1 corresponds to a smaller scale,  $n = 400$ , instance of [LS13, Example 7.1]. Example 2 corresponds to the un-normalized case of [LS13, Example 7.3] with  $n = 500$ . Finally, Example 3 is similar to Example 1 except  $n = 500$  and the matrix  $\mathbf{A}$  is a tridiagonal matrix with diagonal entries equal to 2 and the off-diagonal entries equal to  $-1$  that is derived from the discretization of one-dimensional heat equation with Dirichlet boundary. Figures 1(b), 1(c), and 1(d) show that the proposed scheme leads to smaller residual errors at lower ranks for each of the considered examples. With respect to EKS.M, it should be stated that the proposed scheme is not competitive in terms of timing. The reason for this is that we traverse through all the ranks one by one minimizing the residual at each rank. This iterative process, while it results in smaller residual errors, is computationally more intensive than EKS.M.

## V. CONCLUSION

We have discussed a Riemannian optimization-based approach to low-rank algebraic Riccati equation (1). It leads to the scheme in Table I that alternates between fixed-rank optimization and rank-one updates, converging to a critical point of (2). The fixed-rank optimization problem (3) is solved with a Riemannian trust-region algorithm on the set

of rank- $r$  symmetric positive definite matrices endowed with a novel tuned Riemannian metric (6) that can be seen as a preconditioner for the Riemannian optimization problem. Limited preliminary investigation shows that our approach results in smaller residual errors at lower ranks on standard problem instances. Finding suitable cost functions for (2) and extending the analogy to other matrix equations will be a topic of future research, as well as having a competitive numerical implementation.

## ACKNOWLEDGMENT

We thank Valeria Simoncini for providing the codes for the Galerkin subspace methods.

## REFERENCES

- [AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, NJ, 2008.
- [BMAS14] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt: a Matlab toolbox for optimization on manifolds*, Journal of Machine Learning Research (2014), Accepted for publication.
- [BS13] P. Benner and J. Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: A state of the art survey*, Tech. report, MPI Magdeburg, 2013.
- [EAS98] A. Edelman, T.A. Arias, and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis and Applications **20** (1998), no. 2, 303–353.
- [JBAS10] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM Journal on Optimization **20** (2010), no. 5, 2327–2351.
- [LS13] Y. Lin and V. Simoncini, *A new subspace iteration method for the algebraic Riccati equation*, Tech. report, arXiv:1307.3843, 2013.
- [MAAS12] B. Mishra, K. Adithya Apuroop, and R. Sepulchre, *A Riemannian geometry for low-rank matrix completion*, Tech. report, arXiv:1211.1550, 2012.
- [MBS11] G. Meyer, S. Bonnabel, and R. Sepulchre, *Regression on fixed-rank positive semidefinite matrices: a Riemannian approach*, Journal of Machine Learning Research **11** (2011), no. Feb, 593–625.
- [NS12] T. T. Ngo and Y. Saad, *Scaled gradients on Grassmann manifolds for matrix completion*, Neural Information Processing Systems conference, NIPS, 2012, pp. 1421–1429.
- [SSM13] V. Simoncini, D. B. Szyld, and M. Monslave, *On two numerical methods for the solution of large-scale algebraic Riccati equations*, IMA Journal of Numerical Analysis (2013), doi: 10.1093/imanum/drt015.
- [VV10] B. Vandereycken and S. Vandewalle, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM Journal on Matrix Analysis and Applications **31** (2010), no. 5, 2553–2579.