SIAM J. MATRIX ANAL. APPL.
Vol. 45, No. 3, pp. 1669–1688
© 2024 Society for Industrial and Applied Mathematics

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

# LOW-RANK PLUS DIAGONAL APPROXIMATIONS FOR RICCATI-LIKE MATRIX DIFFERENTIAL EQUATIONS*

SILVÈRE BONNABEL†, MARC LAMBERT‡, AND FRANCIS BACH§

**Abstract.** We consider the problem of computing tractable approximations of time-dependent $d \times d$ large positive semidefinite (PSD) matrices defined as solutions of a matrix differential equation. We propose to use "low-rank plus diagonal" PSD matrices as approximations that can be stored with a memory cost being linear in the high dimension $d$. To constrain the solution of the differential equation to remain in that subset, we project the derivative at all times onto the tangent space to the subset, following the methodology of dynamical low-rank approximation. We derive a closed-form formula for the projection and show that after some manipulations, it can be computed with a numerical cost being linear in $d$, allowing for tractable implementation. Contrary to previous approaches based on pure low-rank approximations, the addition of the diagonal term allows for our approximations to be invertible matrices that can moreover be inverted with linear cost in $d$. We apply the technique to Riccati-like equations, then to two particular problems: first, a low-rank approximation to our recent Wasserstein gradient flow for Gaussian approximation of posterior distributions in approximate Bayesian inference and, second, a novel low-rank approximation of the Kalman filter for high-dimensional systems. Numerical simulations illustrate the results.

**Key words.** matrix differential equation, Riccati equation, low-rank approximation, high dimension, matrix manifolds, Kalman filter, Gaussian variational inference

**MSC codes.** 68Q25, 68R10, 68U05

**DOI.** 10.1137/23M1587610

## 1. Introduction.

Positive semidefinite (PSD) matrices $X \in \mathbb{R}^{d\times d}$ have storage cost in $d^2$, and the computation cost associated with typical matrix operations is in $d^3$. Those two aspects may come as a limitation in various applications where the dimension $d$ is very large. A sensible approach is thus to work with low-rank approximations instead. A rank $p$ approximation of a PSD matrix $X$ can be factored as $Y = URU^T$, where $U \in \mathbb{R}^{d\times p}$ has orthonormal columns and $R$ is a $p \times p$ positive definite (PD) matrix of much-reduced size, letting $p \ll d$. We denote by $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ the set of rank-$p$ PSD matrices. A somewhat richer set that will be the object of this paper is the set of "low-rank plus diagonal" matrices, which we denote by $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$, composed of matrices of the form $Y = URU^T + \psi$ with $\psi$ a diagonal matrix with strictly positive diagonal elements, making $Y$ invertible. We then speak of FA decomposition in relation to the problem of factor analysis (FA) [13]. In the case where the diagonal matrix $\psi$ is made isotropic, that is, is taken of the form $\psi = sI_d$, $s > 0$, we speak of PPCA decomposition in relation to the problem of probabilistic principal component analysis (PPCA) [29] and denote this smaller set by $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$. A well-known "static"

*Received by the editors July 18, 2023; accepted for publication (in revised form) by B. Vandereycken May 23, 2024; published electronically September 6, 2024.
https://doi.org/10.1137/23M1587610
**Funding:** The work of the authors was supported by the French Defence procurement agency (DGA), the French government under the management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program (grant ANR-19-P3IA-0001(PRAIRIE 3IA Institute)), and the European Research Council (grant SEQUOIA 724063).
†Centre for Robotics, Mines Paris, Université PSL, Paris, France (silvere.bonnabel@mines-paristech.fr).
‡DGA, Inria, ENS, Université PSL, Paris, France (marc.lambert@inria.fr).
§Inria, ENS, Université PSL, Paris, France (francis.bach@inria.fr).



Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

approximation problem consists in computing the closest approximating matrix $Y$ to some large PSD matrix $X$ in the sense of the Frobenius norm, that is, solving

$$\min_{Y \in \mathcal{M}} ||Y - X||, \tag{1.1}$$

where $\mathcal{M}$ may denote $\mathrm{S}^+(\mathrm{p},\mathrm{d}), \mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$, or $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$. If $\mathcal{M} = \mathrm{S}^+(\mathrm{p},\mathrm{d})$, the problem is solved by SVD, where eigenvalues are truncated after the $p$th one, a classical result of matrix analysis known as the Eckart–Young–Mirsky theorem [10]; see also [15]. If $\mathcal{M} = \mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$, we recover the problem known as "minimum residual FA"; see [27, 3]. The problem is not solvable in closed form, but numerical methods have been developed; see, e.g., [9].

In this paper, however, we consider a different problem. We consider PSD matrices $X(t) \in \mathbb{R}^{d \times d}$ defined as solutions of matrix differential equations in high dimension. Instead of attacking (1.1) directly, we try to solve at all times

$$\min_{\dot{Y}(t) \in \mathcal{T}_{Y(t)}\mathcal{M}} ||\dot{Y}(t) - \dot{X}(t)||. \tag{1.2}$$

Technically, the problem differs from (1.1), as the derivatives live in a different space, namely, the tangent spaces. The rationale is as follows. To handle possibly high dimension $d$, we would like to maintain a storage cost being linear in $d$—which is achieved by letting $Y(t) \in \mathcal{M}$ at all times—and a computation cost being also linear in $d$ when incrementally evolving our approximation $Y(t)$, hence the need to solve (1.2) efficiently. The solution to

$$\dot{X}(t) = F(X(t)) \tag{1.3}$$

may then be approximated by letting at all times $\dot{Y}(t)$ be the solution to (1.2), where $\dot{X}(t) = F(X(t))$ is replaced by $F(Y(t))$. Indeed, even if $Y(t) \in \mathcal{M}$, the matrix $F(Y(t))$ generally points to a direction that makes $Y(t)$ step out of $\mathcal{M}$, hence the need to project it onto the tangent space $\mathcal{T}_{Y(t)}\mathcal{M}$. Although $\mathcal{T}_{Y(t)}\mathcal{M}$ is a vector space for each $Y(t)$, solving (1.2) exactly while maintaining operations being linear in $d$ is not straightforward, and this is the object of the present paper.

Let us first consider $\mathcal{M} = \mathrm{S}^+(\mathrm{p},\mathrm{d})$. Low-rank approximations $Y(t)$ to $X(t)$ conveniently write as $Y(t) = U(t)R(t)U(t)^T$ with $U(t) \in \mathbb{R}^{d \times p}$ being a matrix whose columns are orthonormal and $R(t) \in \mathrm{S}^+(\mathrm{p},\mathrm{p})$ being a small-size PD symmetric matrix, as advocated in [16, 4, 19]. [16, 19] address the projection problem

$$\min_{\dot{Y}(t) \in \mathcal{T}_{Y(t)}\mathrm{S}^+(\mathrm{p},\mathrm{d})} ||\dot{Y}(t) - \dot{X}(t)|| \tag{1.4}$$

in the sense of the Frobenius norm, where $\mathcal{T}_Y\mathrm{S}^+(\mathrm{p},\mathrm{d})$ denotes the tangent space to $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ at $Y \in \mathrm{S}^+(\mathrm{p},\mathrm{d})$. [16] derives expressions for $\dot{U}(t)$ and $\dot{R}(t)$ such that $\dot{Y}(t)$ corresponds to the solution of (1.4) indeed.

Historically, [5, 7, 19] proposed approximations to the continuous-time low-rank Riccati equation using differential geometry which were in particular applied to approximating the Linblad equations in quantum physics. The work [7, 19] is in line with prior results of [16], where essentially one seeks to orthogonally project the tangent vector in the sense of the Euclidean norm, whereas the approach of [5] builds on the geometry of [4]. A judicious numerical time integrator was proposed in [22] with recent extensions to matrices and tensors that preserve symmetry in [8].

The present paper builds on [16, 7, 19], providing an invertible covariance matrix, by approximating derivatives on $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ and $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ instead. This has not been

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

addressed to our knowledge, and we provide closed-form formulas for the projection of the tangent vector. Although the method is general, implementing the projection depends on the underlying differential equation being approximated. Of particular interest is the Riccati differential equation, which appears in a number of applications, e.g., Kalman filtering, linear quadratic control, stability analysis via the Lyapunov function, or computational statistics to estimate covariance matrices; see [17]. The interests of the "low-rank plus diagonal" framework in the context of matrix dynamical approximation are as follows:

- It allows for manipulation of *invertible* matrices, whereas low-rank approximations are inherently noninvertible. Moreover, using Woodbury's lemma, inversion of such an approximation is feasible at all times; see (4.1) below.
- Approximation with a full rank matrix is desirable in estimation problems where $Y$ encodes a covariance matrix. A rank-deficient covariance may lead to overconfident estimates. In Kalman filtering, this may lead the state variable estimates to spuriously drift, as shall be shown in simulations.
- It ensures more flexibility while retaining a storage cost linear in $d$. Notably, it can capture the individual variances of the variables for a large covariance matrix $Y$.
- $Y$ admits a probabilistic interpretation: $URU^T + sI_d$ is the covariance of $x = Uz + \nu$, with $z \sim \mathcal{N}(0, R)$ a small-dimensional latent variable mapped via $U$ into a subspace of $\mathbb{R}^d$, plus $\nu$ an isotropic noise of magnitude $s$ [29, 13].

Our contributions and the organization of the paper are as follows. Previous work on low-rank approximation of matrix differential equations [16, 19] is recapped in section 2. Section 3 extends the results to the low-rank plus diagonal case: We solve exactly in closed form for each $t$ the extension of problem (1.4),

$$\min_{\dot{Y}(t) \in \mathcal{T}_{Y(t)} \mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})} ||\dot{Y}(t) - \dot{X}(t)||, \tag{1.5}$$

for the set $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ and also for $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$. In section 4, we discuss implementation in high dimension and show that the approximation can be obtained exactly with a computational cost being linear in $d$. In section 5, we leverage the result to provide a tractable approximation to the Riccati equation. This is applied to two problems. First, we derive a low-rank approximation to our Wasserstein gradient flow of [18]. Then we propose a novel low-rank plus diagonal Kalman filter and illustrate the results on a tutorial example inspired by robotics. The code is made publicly available.

As concerns low-rank techniques for Kalman filtering, we note that the problem has arisen in data assimilation for weather forecasting and oceanography [11], [30], where the state is driven by a PDE and is thus encoded by a high-dimensional vector (say, up to 1 million). Since the Kalman filter needs to store the covariance matrix of the estimates, it meets the computer's memory limits when the dimension is high. Different variants have been proposed to tackle this memory problem. The SEEK filter [30] is based on SVD of the covariance matrix and provides a low-rank Riccati update in discrete time. The ensemble Kalman filter [11, 20] approximates the Riccati equation using Monte Carlo sampling. Very recently and closely related to the present paper, a method for efficient, approximate Gaussian (Kalman) filtering, including smoothing and marginal likelihood computation, was developed using low-rank approximations on $\mathrm{S}^+(\mathrm{p},\mathrm{d})$; see [26]. In robotics, [28] proposed the sparse extended information filter to process large maps for simultaneous localization and mapping.

**2. Reminders on low-rank approximation.** We start with the geometry of $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ and then recap existing results on dynamical low-rank approximation.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

**2.1. Geometry of $\mathbf{S^+(p,d)}$.** Any matrix $Y \in \mathrm{S}^+(\mathrm{p},\mathrm{d})$ may be written as $Y = URU^T$, where $R \in \mathrm{S}^+(\mathrm{p},\mathrm{p})$ is a small-size PD matrix, and $U \in \mathrm{V}(\mathrm{p},\mathrm{d})$, where $\mathrm{V}(\mathrm{p},\mathrm{d})$ denotes the set of $d \times p$ matrices with $p$ orthonormal columns, hence satisfying $U^TU = I_p$, called the Stiefel manifold. Letting $\Pi_U = UU^T$ be the projector onto the span of $U$ and $\Pi_U^\perp = I_d - UU^T$ the projector onto the orthogonal subspace, we have $(\Pi_U)^2 = \Pi_U$, $(\Pi_U^\perp)^2 = \Pi_U^\perp$, $\Pi_U^\perp U = 0$, $U^T\Pi_U^\perp = 0$.

The quotient geometry of $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ is thoroughly studied in [4]. Any tangent vector to $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ may be represented by the infinitesimal variation $(\delta U, \delta R) \in \mathcal{T}_{(U,R)}\mathrm{S}^+(\mathrm{p},\mathrm{d})$ with $(\delta U, \delta R) \in \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times p}$ of the following form:

$$\delta U = (I_d - UU^T)\Gamma,\ \Gamma \in \mathbb{R}^{d\times p};\ \delta R \in \mathbb{R}^{p\times p},\ \delta R^T = \delta R. \tag{2.1}$$

This may be interpreted as follows: An infinitesimal variation of $U$ makes its columns move in the subspace of $\mathbb{R}^d$ orthogonal to span(U), as we have $(\delta U)^\top U = 0$, ensuring that $\delta(U^TU) = 0$, in accordnace with the constraint $U^TU = I_p$. And an infinitesimal variation of $R$ necessarily remains symmetric [4]. The corresponding tangent vector $\delta Y = \mathcal{T}_Y\mathrm{S}^+(\mathrm{p},\mathrm{d})$ at $Y = URU^T$ writes as

$$\delta Y = (\delta U)RU^T + U(\delta R)U^T + UR(\delta U)^T. \tag{2.2}$$

**2.2. Optimal approximations on the tangent space.** Any tangent vector to the set of full $d \times d$ PSD matrices at $X$ is encoded by a matrix $H \in \mathbb{R}^{d\times d}$, with $H$ symmetric; see, e.g., [4]. Hence, given matrices $X(t)$ depending smoothly on the parameter $t \in \mathbb{R}$, one may write at each time $\dot{X}(t) = H$ and turn the problem (1.2), which specifies to (1.4) in the present case, into the generic problem of solving

$$\min_{\dot{Y}(t)\in\mathcal{T}_{Y(t)}\mathrm{S}^+(\mathrm{p},\mathrm{d})} ||\dot{Y}(t) - H|| \tag{2.3}$$

with $H$ symmetric and arbitrary. Replacing $H$ with $F(Y(t))$ at each time $t$ and completing it with an initial condition $Y(t_0)$ will yield an approximation $\big(Y(t)\big)_{t\geq t_0}$ to the solution of the ODE $\dot{X} = F(X)$. We have the following.

PROPOSITION 1 (from [16, 19]). *The orthogonal projection of a symmetric matrix $H$ onto $\mathcal{T}_Y\mathrm{S}^+(\mathrm{p},\mathrm{d})$ at $Y = URU^T$ is, in the retained form of tangent vectors* (2.1)*,*

$$\delta Y = P_{U,R,s}(H) = \delta URU^T + U\delta RU^T + UR\delta U^T,$$

*where the matrices are given by*

$$\delta U = (I - UU^T)HUR^{-1}, \tag{2.4}$$

$$\delta R = U^THU. \tag{2.5}$$

*The tangent vector then writes as*

$$P_{U,R,s}(H) = HUU^T + UU^TH - UU^THUU^T.$$

*This choice solves problem* (2.3)*; that is, it minimizes over matrices of the form $\delta Y = \delta URU^T + U\delta RU^T + UR\delta U^T$ with constraints* (2.1) *the cost*

$$J_{H;U,R}(\delta U, \delta R) = \mathrm{Tr}((H - \delta Y)^2) = ||H - \delta Y||^2, \tag{2.6}$$

*whose minimum is given by* $\min J_{H;U,R}(\delta U, \delta R) = \mathrm{Tr}((I - UU^T)H(I - UU^T)H)$ *and where we used the trace form of Frobenius norm* $||A||^2 = \mathrm{Tr}(A^TA) = \mathrm{Tr}(A^2)$ *for symmetric matrices.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Given a differential equation $\frac{d}{dt}X(t) = F(X(t))$ over PSD matrices of full size $d \times d$, (2.4)–(2.5) readily yield $\dot{U}(t)$ and $\dot{R}(t)$ to implement its low-rank approximation at all times, letting $H = F(Y(t))$ with $Y(t) = U(t)R(t)U(t)^T$.

**3. Optimal low-rank plus diagonal approximation.** We now turn to our main theoretical results, which consist of extensions to the low-rank plus diagonal case.

**3.1. Geometry of $\mathbf{S^+_{diag}(p,d)}$ and $\mathbf{S^+_{isot}(p,d)}$.** Take $Y = URU^T + \psi$ with $\psi$ diagonal and having all its diagonal elements strictly positive. Recalling (2.1), an element $Y \in \mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$, along with tangent vectors $\delta Y \in \mathcal{T}_Y \mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ to this element, writes as

$$\left.\begin{aligned} Y &= URU^T + \psi, \\ \delta Y &= \delta U R U^T + U \delta R U^T + U R \delta U^T + \delta\psi \end{aligned}\right\} \quad \text{(FA)} \tag{3.1}$$

with $\delta U, \delta R$ as in (2.1) and $\delta\psi$ diagonal.

*Remark* 1. In terms of geometric structure, $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ is a smooth embedded submanifold albeit not straightforward to prove. The interested reader is referred to [31], which contains two proofs: Proposition 3.11 and another from [14], which is completed therein. Unfortunately, if we wonder whether this extends to $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$, we discover that $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ is not a submanifold: The dimension of the tangent space is not constant, as can be seen by considering $Y = xx^T + \psi$ with $x = (1,0,0)^T$ on the one hand, leading to a dimension of 5 for all $\delta Y$, and $x = (1,1,1)^T$ on the other, leading to a dimension of 6. Thus, $\mathrm{S}^+_{\mathrm{diag}}(1,3)$ bears no manifold structure, but this shall not prevent one from using (3.1) to represent the matrices and their derivatives, and we refer to the set of all such $\delta Y$ as the tangent space at $Y$.

Let us now turn to the PPCA decomposition of the form $Y = URU^T + sI_d$. So far, it has been introduced as a subset of $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ based on the constraint $\varphi = sI_d$ for the sake of simplicity of exposition. However, because some degrees of freedom are removed by this constraint, the eigenvalues of $Y$ are necessarily lower bounded by $s > 0$, as $Y \succeq sI_d$, since $URU^T$ is PSD. Thus, $Y$ cannot have arbitrarily small eigenvalues in span(U). This motivates the use of a slightly different, richer parameterization. From now on, we define $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ as matrices of the form

$$Y = U(R - sI)U^T + sI = URU^T + s(I - UU^T) = URU^T + s\Pi^{\perp}_U \tag{3.2}$$

and impose that $R \in \mathrm{S}^+(\mathrm{p},\mathrm{p})$, $U \in \mathrm{V}(\mathrm{p},\mathrm{d})$, and $s > 0$. This allows for $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ to be a bigger approximating set while retaining that $Y$ is PD.

This leads to the following geometry for $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$. An element $Y \in \mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ with associated tangent vector $\delta Y \in \mathcal{T}\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ in the PPCA form writes as

$$\left.\begin{aligned} Y &= URU^T + s(I - UU^T) = U(R - sI)U^T + sI, \\ \delta Y &= \delta U(R - sI)U^T + U(\delta R - \delta s I)U^T, \\ &\quad + U(R - sI)\delta U^T + \delta s I \end{aligned}\right\} \text{(PPCA)} \tag{3.3}$$

with $\delta U, \delta R$ as in (2.1) and $\delta s \in \mathbb{R}$.

**3.2. Optimal approximation in the PPCA form.** Let us start with the simpler PPCA decomposition.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

PROPOSITION 2. *The orthogonal projection of a symmetric matrix* $H \in \mathbb{R}^{d\times d}$ *onto* $\mathcal{T}_Y \in \mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ *is* $\delta Y = \delta U(R-sI)U^T + U(\delta R - \delta s I)U^T + U(R-sI)\delta U^T + \delta s I$, *where the matrices are given by*

$$\begin{aligned} \delta s &= [\mathrm{Tr}(H) - \mathrm{Tr}(U^T H U)]/(d-p), \\ \delta U &= (I - UU^T)HU(R-sI)^{-1}, \\ \delta R &= U^T H U. \end{aligned} \tag{3.4}$$

*The tangent vector then writes as*

$$P_{U,R,s}(H) = HUU^T + UU^T H - UU^T HUU^T + \delta s(I - UU^T).$$

*This choice minimizes over matrices of the form* (3.3) *the following cost:*

$$C_{H;U,R,s}(\delta U, \delta R, \delta s) = \mathrm{Tr}((H - \delta Y)^2). \tag{3.5}$$

*Proof.* The cost (3.5), that is, the squared Frobenius norm between $H$ and the tangent vector, rewrites as

$$\mathrm{Tr}((\tilde{H} - [\delta U \tilde{R} U^T + U\delta R U^T + U\tilde{R}\delta U^T])^2)$$

with $\tilde{H} = H - \delta s(I - UU^T)$ and $\tilde{R} = R - sI$. In other terms,

$$C_{H;U,R,s}(\delta U, \delta R, \delta s) = J_{\tilde{H};U,\tilde{R}}(\delta U, \delta R)$$

with $J$ as in (2.6). The next step consists in writing

$$\min_{\delta U, \delta R, \delta s} C_{H;U,R,s} = \min_{\delta s}\big(\min_{\delta U, \delta R} C_{H;U,R,s}\big) \tag{3.6}$$

$$= \min_{\delta s}\big(\min_{\delta U, \delta R} J_{\tilde{H};U,\tilde{R}}\big) \tag{3.7}$$

$$= \min_{\delta s} \mathrm{Tr}\big((I - UU^T)\tilde{H}(I - UU^T)\tilde{H}\big), \tag{3.8}$$

using Proposition 1, which also shows that in the present case, $\min_{\delta U,\delta R} J_{\tilde{H};U,\tilde{R}}$ corresponds to

$$\delta U = (I - UU^T)\tilde{H}U\tilde{R}^{-1} = (I - UU^T)HU\tilde{R}^{-1}, \tag{3.9}$$

$$\delta R = U^T \tilde{H} U = U^T H U. \tag{3.10}$$

To conclude, we still have to solve (3.8). Recalling $\tilde{H} = H - \delta s(I - UU^T)$, we differentiate $\mathrm{Tr}\big((I - UU^T)\tilde{H}(I - UU^T)\tilde{H}\big)$ with respect to $\delta s$,

$$\begin{aligned} J'(\delta s) &= -2\mathrm{Tr}\big((I - UU^T)\tilde{H}\big) \\ &= -2\mathrm{Tr}\big((I - UU^T)(H - \delta s(I - UU^T))\big), \end{aligned}$$

and we find $\delta s = [\mathrm{Tr}(H) - \mathrm{Tr}(U^T H U)]/(d-p)$. ∎

**3.3. Optimal approximation in the FA form.** The problem of projecting onto $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ is more difficult than the latter, as in the proof of Proposition 2, we extensively used the isotropy of the diagonal term $sI_d$, namely, $U(sI_d)U^T = sUU^T$. However, it turns out that we may find a closed-form expression also in the FA decomposition. This allows for capturing different individual variances in the subspace orthogonal to span(U).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The first step is to transform the optimization problem over the matrix $\psi$ into a least-squares problem for a vector consisting of its diagonal. The cost we want to minimize is

$$\begin{aligned}
&C_{H;U,R,\psi}(\delta U, \delta R, \delta\psi)\\
&:= \mathrm{Tr}((H - [\delta U R U^T + U\delta R U^T + UR\delta U^T + \delta\psi])^2)\\
&= \mathrm{Tr}((H - \delta\psi - [\delta U R U^T + U\delta R U^T + UR\delta U^T])^2)\\
&= \mathrm{Tr}((\tilde{H} - [\delta U R U^T + U\delta R U^T + UR\delta U^T])^2)\\
&= J_{\tilde{H},U,R}(\delta U, \delta R),
\end{aligned}$$

where now $\tilde{H}$ denotes $\tilde{H} = H - \delta\psi$. We write

$$\min_{\delta U,\delta R,\delta\psi} C(\delta U, \delta R, \delta\psi) = \min_{\delta\psi}\big(\min_{\delta U,\delta R} J_{\tilde{H};U,\tilde{R}}(\delta U, \delta R)\big),$$

which is equal to $\min_{\delta\psi} J_{\tilde{H};U,R}(\delta U, \delta R)$ with

$$\delta U = (I - UU^T)\tilde{H}UR^{-1}, \tag{3.11}$$

$$\delta R = U^T\tilde{H}U, \tag{3.12}$$

using (2.4), (2.5). Using Proposition 1, we see that the cost function to minimize becomes

$$\begin{aligned}
\tilde{C}(\delta\psi) &:= \mathrm{Tr}\big((I - UU^T)\tilde{H}(I - UU^T)\tilde{H}\big)\\
&= \mathrm{Tr}\big(\tilde{H}^2\big) - 2\mathrm{Tr}\big(UU^T\tilde{H}^2\big) + \mathrm{Tr}\big(UU^T\tilde{H}UU^T\tilde{H}\big).
\end{aligned} \tag{3.13}$$

Equation (3.13) may be rewritten as a vector least-squares problem for the vector $\overline{\delta\psi} := (\delta\psi_{11}, \delta\psi_{22}, \ldots, \delta\psi_{dd})^T = \mathrm{diag}(\delta\psi) \in \mathbb{R}^{\mathrm{d}}$. This leads to the following optimal approximation.

PROPOSITION 3. *The orthogonal projection of a symmetric matrix $H \in \mathbb{R}^{d\times d}$ onto $\mathcal{T}_Y \in \mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ is $\delta Y = \delta U R U^T + U\delta R U^T + UR\delta U^T + \delta\psi$, where the matrices are given by*

$$\begin{aligned}
&\mathrm{diag}(\delta\psi) = \overline{\delta\psi},\\
&\delta U = (I - UU^T)(H - \delta\psi)UR^{-1},\\
&\delta R = U^T(H - \delta\psi)U
\end{aligned} \tag{3.14}$$

*and where the vector $\overline{\delta\psi} \in \mathbb{R}^d$ that encodes the diagonal matrix $\delta\psi \in \mathbb{R}^{d\times d}$ is defined by*

$$\overline{\delta\psi} = ((I - UU^T)^{\circ 2})^+ \mathrm{diag}((\mathrm{I} - \mathrm{UU^T})\mathrm{H}(\mathrm{I} - \mathrm{UU^T})), \tag{3.15}$$

*letting $+$ denote the Moore–Penrose inverse and where $\circ$ denotes the elementwise (Hadamard) matrix product.*

*Proof.* Recall the notation $\Pi_U^\perp = I_d - UU^T$. We may rewrite the cost (3.13) in terms of the vector $\overline{\delta\psi}$, i.e., $\tilde{C}(\delta\psi) = \bar{C}(\overline{\delta\psi})$, and we find

$$\begin{aligned}
\bar{C}(\overline{\delta\psi}) &= \mathrm{Tr}\big((\Pi_{\mathrm{U}}^\perp\mathrm{H} - \Pi_{\mathrm{U}}^\perp\mathrm{diag}(\overline{\delta\psi}))^2\big)\\
&= c - 2\mathrm{Tr}(\Pi_{\mathrm{U}}^\perp\mathrm{H}\Pi_{\mathrm{U}}^\perp\mathrm{diag}(\overline{\delta\psi})) + \mathrm{Tr}(\mathrm{diag}(\overline{\delta\psi})\Pi_{\mathrm{U}}^\perp\mathrm{diag}(\overline{\delta\psi})\Pi_{\mathrm{U}}^\perp),
\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where $c$ is a constant with respect to $\overline{\delta\psi}$. This may be rewritten as a vector least-squares problem for the vector $\overline{\delta\psi}$ using the relation $\mathrm{Tr}(\mathrm{diag}(\mathrm{x})\mathrm{A}\mathrm{diag}(\mathrm{y})\mathrm{B}^{\mathrm{T}}) = \mathrm{x}^{\mathrm{T}}(\mathrm{A}\circ \mathrm{B})\mathrm{y}$, where $\circ$ is the elementwise matrix product. If we note $1_{\mathrm{d}}$ the vector of ones, the cost becomes

$$\tilde{C}(\delta\psi) = c - 2{1_{\mathrm{d}}}^{\mathrm{T}}(\Pi_{\mathrm{U}}^{\perp}\mathrm{H}\Pi_{\mathrm{U}}^{\perp}\circ \mathrm{I})\overline{\delta\psi} + \overline{\delta\psi}^{\mathrm{T}}(\Pi_{\mathrm{U}}^{\perp}\circ\Pi_{\mathrm{U}}^{\perp})\overline{\delta\psi}.$$

Taking the derivative with respect to $\overline{\delta\psi}$, we obtain

$$\nabla\bar{C}(\overline{\delta\psi}) = -2\mathrm{diag}(\Pi_{\mathrm{U}}^{\perp}\mathrm{H}\Pi_{\mathrm{U}}^{\perp}) + 2(\Pi_{\mathrm{U}}^{\perp}\circ\Pi_{\mathrm{U}}^{\perp})\overline{\delta\psi}.$$

Zeroing the gradient yields (3.15) indeed. ∎

**4. Implementation.** We now discuss the interest of the results for the high-dimensional setting. Our goal, for tractability, is to elicit operations and storage costs being linear in the dimension $d$. We prove that both methods lead to projections that require linear computation cost in the dimension $d$ and discuss the numerical cost in detail. A key result to our analysis is that any matrix $Y$ of $\mathrm{S}^{+}_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ or $\mathrm{S}^{+}_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ is easily inverted via the Woodbury lemma,

$$(URU^T + \psi)^{-1} = \psi^{-1} - \psi^{-1}U(R^{-1} + U^T\psi^{-1}U)^{-1}U^T\psi^{-1}, \tag{4.1}$$

a fact we will extensively use. Note that it also proves that matrices of $\mathrm{S}^{+}_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ and $\mathrm{S}^{+}_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$, contrary to low-rank matrices of $\mathrm{S}^{+}(\mathrm{p},\mathrm{d})$, can be inverted, and this with a numerical cost of inversion is $O(p^3d)$ and hence linear in the dimension $d$.

**4.1. Numerically efficient formulation for the FA form.** Equation (3.15) necessitates (pseudo)inversion of a $d\times d$ matrix $(I - UU^T)^{\circ 2}$, which may hinder its use in large dimension. However, it is amenable to a linear computation cost.

Proposition 4. *The orthogonal projection* (3.15) *can be performed with linear computation cost in $d$. Indeed, the required vector $\overline{\delta\psi}\in\mathbb{R}^d$ is given by* (4.4) *below and can be advantageously computed via* (4.10).

To prove the result, we start with a lemma whose proof is technical and hence postponed to Appendix A.

Lemma 1. *The cost $\tilde{C}(\delta\psi)$ may be rewritten as a function of the vector $\overline{\delta\psi}\in\mathbb{R}^d$ as follows:*

$$\begin{aligned}\tilde{C}(\delta\psi) = \bar{C}(\overline{\delta\psi}) := \alpha &+ (\bar{h} - \overline{\delta\psi})^T(\bar{h} - \overline{\delta\psi})\\ &+ 4\bar{h}_U^T\overline{\delta\psi} - 2\overline{\delta\psi}^T\bar{D}\overline{\delta\psi} - 2\Lambda^T\overline{\delta\psi} + \overline{\delta\psi}^T\Upsilon\Upsilon^T\overline{\delta\psi},\end{aligned} \tag{4.2}$$

*where $\alpha$ is a constant, $\bar{h},\bar{h}_U,\Lambda$ are all vectors of $\mathbb{R}^d$, $\bar{D}$ is a diagonal $d\times d$ matrix, and $\Upsilon$ is a matrix, all being given below. Hence, the gradient writes as*

$$\frac{1}{2}\nabla\bar{C}(\overline{\delta\psi}) = \big(I - 2\bar{D} + \Upsilon\Upsilon^T\big)\overline{\delta\psi} - \Lambda + 2\bar{h}_U - \bar{h}, \tag{4.3}$$

*and the optimizer is given by*

$$\overline{\delta\psi} = \big(I - 2\bar{D} + \Upsilon\Upsilon^T\big)^{+}\big(\bar{h} - 2\bar{h}_U + \Lambda\big). \tag{4.4}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

*The parameters in the equations above are*

$$\bar{h} = \mathrm{diag}(\mathrm{H}), \qquad \bar{\mathrm{h}}_{\mathrm{U}} := \mathrm{diag}(\mathrm{UU}^{\mathrm{T}}\mathrm{H}), \tag{4.5}$$

$$\Lambda = (\Lambda_1, \ldots, \Lambda_d)^T \in \mathbb{R}^d, \tag{4.6}$$

$$\Lambda_k = \sum_{1 \le i,j \le r} (U^T H U)_{ij} U_{ki} U_{kj}, \quad 1 \le k \le d, \tag{4.7}$$

$$\Gamma_{ij} = ((\Gamma_{ij})_1, \ldots, (\Gamma_{ij})_d)^T \in \mathbb{R}^d, \tag{4.8}$$

$$(\Gamma_{ij})_k = U_{ki} U_{kj}, \quad 1 \le k \le d, \tag{4.9}$$

*with* $\Upsilon$ *a matrix of size* $d \times \frac{p(p+1)}{2}$ *whose first* $\frac{p(p-1)}{2}$ *columns are given by the vectors* $\sqrt{2}\Gamma_{ij}$, $i < j$, *and whose last* $r$ *columns are* $\Gamma_{ii}$ *and where* $\bar{D}$ *is the diagonal matrix defined by* $\bar{D}_{ii} = \sum_{j=1}^r U_{ij}^2$. *It may be checked that* (4.4) *coincides with* (3.15).

The problem with (4.4) is that it requires inverting a $d \times d$ matrix, which is not affordable computationally. When $I - 2\bar{D} + \Upsilon\Upsilon^T$ is invertible, we can use the Woodbury matrix identity instead to express the solution (4.4) as

$$\overline{\delta\psi} = \big(\varphi - \varphi\Upsilon\big(I_{\frac{p(p+1)}{2}} + \Upsilon^T\varphi\Upsilon\big)^{-1}\Upsilon^T\varphi\big)\big(\bar{h} - 2\bar{h}_U + \Lambda\big), \tag{4.10}$$

where $\varphi := (I_d - 2\bar{D})^{-1}$ is the inverse of a diagonal matrix, which may be efficiently computed using only vectors of $\mathbb{R}^d$, and where the $\frac{p(p+1)}{2} \times \frac{p(p+1)}{2}$ matrix $I_{\frac{p(p+1)}{2}} + \Upsilon^T\psi\Upsilon$ needs be inverted.

**4.2. Computation cost.** As a preliminary remark, we see that whatever the chosen low-rank decomposition, computing the approximation requires storing and projecting the $d \times d$ symmetric matrix $H$. To perform this in a high dimension, we need to assume some sort of sparsity or low-rank structure for $H$ of the kind $H = GG^T$ with $G \in \mathbb{R}^{d \times r}$, where $r \ll d$ or $H \in \mathrm{S}_\psi^+(\mathrm{r}, \mathrm{d})$. This problem is already present in the literature on low-rank approximations and has proved not to be a limitation in applications; see [16, 19] and articles that followed. For our complexity analysis, we will henceforth suppose the computational cost of computing $U^T H U$ is $O(dp^2)$. Note that to project the matrix in the PPCA or FA form, all equations must be coded "vectorially"; that is, diagonal matrices are encoded by vectors, and we carefully choose the order of operations to never multiply full-size matrices, e.g., in (4.10).

Propositions 2–4 give the formulas that one needs to evolve the factors $U(t), R(t)$, $s(t)$ or $U(t), R(t), \psi(t)$, where their derivatives are given by (3.4) and (3.14), (4.10), respectively. In the PPCA form, to compute matrices (3.4), we see we need to perform $O(dp^2)$ operations. In the FA form, implementation of (3.14) retains linear complexity in $d$. However, in the efficient implementation (4.10) of (3.15), we need to invert the $\frac{p(p+1)}{2} \times \frac{p(p+1)}{2}$ matrix $I_{\frac{p(p+1)}{2}} + \Upsilon^T\psi\Upsilon$. This yields a computational cost of order $O(dp^2) + O(p^6)$. We thus see that FA decomposition, albeit richer, comes at a price: The number of "factors" $p$ that may be used is more limited than in the PPCA decomposition.

To illustrate how the methods compare in terms of numerical cost, we propose to apply the three methods for the approximation of a large-scale matrix $H = GG^T$, where $G$ is a matrix of size $d \times r$ generated randomly, where $r > p$ is low but sufficiently large for the $p$-rank approximation not to be exact. Using a product form for $H$ allows for low-memory-cost operations; in particular, we have $UHU^T = (UG)(UG)^T$, yielding a linear cost in $d$. The execution times of the different algorithms are shown

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 1
*Matrix projection with $d = 10^6$, $p = 10$, and $r = 100$.*

| Set | Projection method | Execution time (s) | Memory cost |
|---|---|---|---|
| $\mathrm{S}^+(\mathrm{p},\mathrm{d})$ | "Low-rank" | 3.5 | $dp$ |
| $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ | "PPCA" | 7 | $dp$ |
| $\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ | "FA" | 35 | $dp + p^4/4$ |

in Table 1 in dimension $d = 10^6$, where the computations are performed on a standard laptop with Octave. All algorithms scale well to very-high-dimensional problems. However, FA has a higher computation time since it needs to invert a $\frac{p(p+1)}{2} \times \frac{p(p+1)}{2}$ matrix.

**5. Application to the Riccati equation.** In this section, we consider the following classical continuous-time Riccati equation in high dimension $d$:

$$\frac{d}{dt}P = AP + PA^T + Q - PC^T N^{-1} CP, \tag{5.1}$$

where $P$, $A$, and $Q$ are $d \times d$ high-dimensional matrices, $N$ is a $k \times k$ matrix, and $C$ is of size $k \times d$. $P$, $Q$, and $N$ are PSD matrices.

To store this covariance matrix $P$ in high dimension, a first solution is to project it onto the low-rank manifold $\mathrm{S}^+(\mathrm{p},\mathrm{d})$. To do so, we replace $P$ with $Y = URU^T$ in the right-hand side (5.1), and we project the obtained matrix. Proposition 1 yields

$$\begin{aligned} \dot{U} &= (I - UU^T)(AU + QUR^{-1}), \\ \dot{R} &= U^T AUR + RU^T A^T U + U^T QU - RU^T C^T N^{-1} CUR, \end{aligned} \tag{5.2}$$

termed a low-rank Riccati equation in [7]. This may be used as a low-rank proxy to the Kalman filter. The obtained filter then performs inference in the low-dimensional subspace spanned by the dominant eigenvectors [20].

To approximate (5.1) in $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ instead, we replace $P$ with $Y = URU^T + s(I - UU^T)$ in the right-hand side of (5.1), and we project the obtained matrix using (3.4).

PROPOSITION 5 (PPCA-Riccati). *The lift on $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$ of the orthogonal projection of the vector field defined by* (5.1) *writes as*

$$\dot{s} = \frac{1}{d-p}\mathrm{Tr}\big((I - UU^T)(2sA + Q - s^2 C^T N^{-1} C)\big), \tag{5.3}$$

$$\dot{U} = (I - UU^T)(AUR + QU + sA^T U - sC^T N^{-1} CUR)(R - sI)^{-1}, \tag{5.4}$$

$$\dot{R} = U^T AUR + RU^T A^T U + U^T QU - RU^T C^T N^{-1} CUR. \tag{5.5}$$

We have the following interpretation: (5.5) is the original Riccati equation (5.1), where matrices are projected onto the subspace encoded by $U$; (5.4) resembles the Oja flow [24], which tracks the dominant subspace of a symmetric matrix; and (5.3) provides an adaptation of parameter $s$ to reflect the inflation of the covariance under the vector field in the subspace orthogonal to span(U). Note that (5.3) offers guarantees that $s(t)$ remains positive at all times since as long as $(I - UU^T)Q \neq 0$, we have $s = 0 \Rightarrow \dot{s} > 0$. Thus, the lifted equations (5.3)–(5.5) preserve our PPCA form at all times.

In terms of implementation, we will need a numerical integration method for the obtained differential equations. With $h$ being the time step and $\dot{s}(t) = \delta s, \dot{U}(t) = \delta U, \dot{R}(t) = \delta R$, one needs to define $s(t+h), U(t+h), R(t+h)$ so that $Y(t+h) \in \mathrm{S}^+_{\mathrm{isot}}(\mathrm{p},\mathrm{d})$. This can be performed using a retraction; see [1]. We let

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$$s(t+h) \leftarrow s(t) + h\delta s, \tag{5.6}$$

$$U(t+h) \leftarrow \text{retraction\_qr}(\mathrm{U(t)}, \mathrm{h}\delta\mathrm{U}), \tag{5.7}$$

$$R(t+h) \leftarrow R^{1/2} \exp(hR^{-1/2}\delta R R^{-1/2})R^{1/2}, \tag{5.8}$$

where (5.7) is based on the retraction "retraction_qr" of the Stiefel manifold from the Manopt toolbox [6], (5.8) is a natural retraction on PD matrices [4], and "exp" denotes the usual matrix exponential, whose computation is tractable in the small dimension $p$. $\mathrm{S}^{+}_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ is similarly treated. Note that alternative schemes have been proposed on $\mathrm{S}^{+}(\mathrm{p},\mathrm{d})$; see [16, 19, 23]. [22] even proposes a numerical integration method that avoids inverting the factor $R$; see [8] for recent developments.

Regarding the FA approach, we can approximate (5.1) in $\mathrm{S}^{+}_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ as follows. We write $H := \dot{P} = AP + PA^T + Q - PSP$ with $S = C^T N^{-1} C$ and $P = URU^T + \psi$. After some calculations, this yields the following.

PROPOSITION 6 (FA-Riccati). *The lift on* $\mathrm{S}^{+}_{\mathrm{diag}}(\mathrm{p},\mathrm{d})$ *of the orthogonal projection of the vector field defined by* (5.1) *writes as*

$$\begin{aligned}
&\mathrm{diag}(\dot{\psi}) = (\mathrm{I} - 2\bar{\mathrm{D}} + \Upsilon\Upsilon^{\mathrm{T}})^{+}\,\mathrm{diag}((\mathrm{I} - \mathrm{UU}^{\mathrm{T}})\mathrm{M}(\mathrm{I} - \mathrm{UU}^{\mathrm{T}})),\\
&\dot{U} = (I - UU^T)(M - \delta\psi)UR^{-1} + (I - UU^T)(AU - \psi SU),\\
&\dot{R} = U^T(M - \delta\psi)U - U^T\psi SUR - RU^T S\psi U - RU^T SUR + U^T AUR + RU^T A^T U,
\end{aligned}$$

*where* $M = A\psi + \psi A^T + Q - \psi S\psi$, $S = C^T N^{-1} C$ *and* $\bar{D}, \Upsilon$ *as in* (4.4).

The pseudoinversion may be avoided when $p \ll d$ using the Woodbury formula as in (4.10), leading to an update having a memory cost linear in $d$ if the order of operations is taken carefully.

We now propose two applications for the equations just obtained: one in the field of computational statistics, where one seeks to approximate a fixed posterior distribution, for which we provide a tractable approximation in high dimension through a Riccati flow, and one in the field of Kalman filtering, with applications to robotics. The former is essentially an illustration of the applicability of the method, while the latter aims at comparing the various approaches through numerical simulations.

**5.1. Application to computational statistics.** We seek to approximate a distribution in high dimension which is the stationary solution of a Langevin equation. In this context, Markov chain Monte Carlo (MCMC) has been extensively studied in statistical physics and machine learning [2]. Recently, variational inference [18] has been used in this framework as an alternative form to MCMC to approximate the stationary distribution with a Gaussian distribution. The final approximation is defined as the asymptotic limit of a flow over the set of Gaussian distributions, parameterized by their mean and covariance matrix. The flow for the covariance matrix has the form of a Riccati equation. To illustrate the applicability of the methods developed in the present paper, we now show that the PPCA projection can be directly applied to these flows to provide a memory-efficient and computationally tractable algorithm in high dimension.

**5.1.1. Wasserstein gradient flow for Gaussian variational inference.** In Bayesian statistics, one is often faced with the problem of approximating a target posterior distribution $\pi$ that is known up to a normalizing constant; that is, we have access to $\nabla V(x)$, where $\pi(x) \propto \exp(-V(x)/\epsilon)$. To compute a Gaussian approximation of $\pi$, we may proceed as follows. The stochastic differential equation (Langevin dynamics)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$$dx_t = -\nabla V(x_t)dt + \sqrt{2\varepsilon}dB_t \tag{5.9}$$

is such that, under suitable assumptions, the marginal distribution of $x_t$ satisfies the Fokker–Planck (partial differential) equation,

$$\frac{\partial p_t}{\partial t} = \mathrm{div}(\mathrm{V p_t}) + \varepsilon \Delta \mathrm{p_t}, \tag{5.10}$$

which has stationary distribution $\pi$. Thus, if one can approximate the solution to the latter equation by a Gaussian density, one may hope that the latter tends asymptotically to a Gaussian approximation of $\pi$. In the recent work [18], the solution of this PDE has been approximated with a Gaussian $p_t \approx q_t = \mathcal{N}(\mu_t, P_t)$ using variational inference. The Gaussian flow

$$\begin{aligned} &\dot{\mu}_t = -\mathbb{E}_{X\sim\mathcal{N}(\mu_t,P_t)}[\nabla V(X)], \\ &\dot{P}_t = A_t P_t + P_t A_t^T + Q, \\ &Q := 2\varepsilon I; \quad A_t := -\mathbb{E}_{X\sim\mathcal{N}(\mu_t,P_t)}[\nabla^2 V(X)] \end{aligned} \tag{5.11}$$

is shown to converge with exponential rate to the Gaussian distribution being the closest to the target distribution $\pi$ (in the sense of Kullback–Leibler divergence) if $V$ is strongly convex; see [18, Appendix A]. For large-scale problems, the differential equation on $P_t$ in (5.11) is problematic because it requires storing at each time a $d \times d$ matrix. Moreover, the computation cost is (roughly) cubic in the dimension due to matrix products, even if we overlook the difficulty of computing the required expectations. This motivates a low-rank plus diagonal approximation to this Riccati-like equation. We readily see that the Wasserstein gradient flow (5.11) lends itself to the present low-rank approximation framework, as it is of the form (5.1), provided that the expectations which define $A_t$ may be computed and that they can be written in a factorized form compatible with the high dimension.

**5.1.2. Particular case of a Gaussian target.** To fix ideas, let us see what the equations boil down to in the case where the target is Gaussian, that is, $\pi \sim \mathcal{N}(m, \epsilon M)$, to be consistent with $\pi \propto \exp(-V/\epsilon)$ so that $V(x) = \frac{1}{2}(x-m)^T M^{-1}(x-m)$. In this case, the expectations can be computed analytically, and the variational Gaussian flow (5.11) writes as

$$\begin{aligned} \dot{\mu}_t &= -M^{-1}(\mu_t - m), \\ \dot{P}_t &= 2\varepsilon I - M^{-1}P_t - P_t M^{-1}. \end{aligned}$$

The equations have as stationary point $(\mu, P) = (m, \epsilon M)$, which are the parameters of $\pi$, as expected.

*Remark* 2. We see that the "pure" low-rank projection, defined by (2.4), of the covariance $P_t$ writes as

$$\dot{U} = -(I - UU^T)M^{-1}U,$$

and we recover minus the Oja flow [24], that is, the flow in a form that tracks the eigenvectors corresponding to the smallest eigenvalues of $M^{-1}$, thus the dominant eigenspace of $M$.

**5.1.3. PPCA approximation in the large-scale case.** We now come back to a general target posterior $\pi \propto \exp(-V/\epsilon)$ and seek to approximate the Gaussian

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

flow (5.11) in a tractable form with high $d$. The first step is to approximate the expectation under the Gaussian distribution appearing in the Gaussian flow (5.11) with $K$ being Monte Carlo samples. To this aim, we may let appear an outer product of the form

$$\begin{aligned}
&\mathbb{E}_{\mathcal{N}(\mu_t,P_t)}[\nabla^2 V(X)]P_t = \mathbb{E}_{\mathcal{N}(\mu_t,P_t)}[\nabla V(X)(X-\mu_t)^T] \\
&\quad \approx \frac{1}{K}\sum_{k=1}^{K} \nabla V(x_t^k)(x_t^k-\mu_t)^T = D_t B_t^T, \quad \text{where} \quad x_t^k \sim q_t, \\
&D_t := \frac{1}{\sqrt{K}}\left(\nabla V(x_t^1),\dots,\nabla V(x_t^N)\right) \quad B_t := \frac{1}{\sqrt{K}}\left(x_t^1-\mu_t,\dots,x_t^N-\mu_t\right),
\end{aligned}$$

where the first equality comes from integration by parts (Stein lemma [21]). We now only need to store the matrices $B_t$ and $D_t$ of size $d \times K$. In practice, the sampling is done after the discretization of these ODEs at each integration step.

As $D_t$ and $B_t$ depend on the samples $x_t^k$ generated from the distribution $q_t \sim \mathcal{N}(\mu_t, P_t)$, where in our PPCA approximation we let $P_t = URU^T + s(I - UU^T)$, we need to be able to sample from a Gaussian distribution having such a factorized covariance matrix. A simple way to do so is as follows. We first note that if we take $Z_1 \sim \mathcal{N}(0, URU^T)$ and $Z_1 \sim \mathcal{N}(0, s(I-UU^T))$ independent, then $Z_1 + Z_2$ has the desired covariance. To sample $Z_1$, one may sample a small-dimensional variable $Z_3 \sim \mathcal{N}(0, R)$ and let $Z_1 = UZ_3$. To sample $Z_2$, we may sample $Z_4 \sim \mathcal{N}(0, sI)$ and let $Z_2 = (I - UU^T)Z_4$.

The ODE we seek to approximate now takes the form

$$\dot{P}_t = 2\varepsilon I - (D_t B_t^T + B_t D_t^T). \tag{5.12}$$

We project this equation onto the PPCA subset. This is achieved thanks to Proposition 5, and it yields

$$\begin{aligned}
\dot{s} &= \frac{1}{d-p}\mathrm{Tr}\big(2(I-UU^T)(\varepsilon I - D_t B_t^T)\big), \\
\dot{U} &= (I-UU^T)(2\varepsilon I - D_t B_t^T - B_t D_t^T)U(R-sI)^{-1}, \\
\dot{R} &= U^T(2\varepsilon I - D_t B_t^T - B_t D_t^T)U.
\end{aligned}$$

These operations can be computed in a memory-efficient way using the relation $\mathrm{Tr} D_t B_t^T = \sum_{i=1}^{d} D_t[i,:]B_t[:,i]$ and $U^T D_t B_t^T U = (U^T D_t)(U^T B_t)^T$ to only manipulate vectors or matrices of size $p \times K$.

We have briefly shown that using the results of the present paper, the recent Wasserstein gradient flow of [18] for variational inference is amenable to a PPCA approximation being compatible with the high dimension. We now turn to another application and compare numerically the various low-rank approaches.

**5.2. Application to Kalman filtering.** In this section, we consider the more standard problem of the Kalman filter in continuous time known as the Kalman–Bucy filter. In Kalman filtering, we seek to estimate hidden physical quantities that evolve over time and that are partially observed through a linear model. The covariance of the state $P$ satisfies the Riccati equation (5.1).

To assess the various low-rank approximations, we consider an example inspired by robotics. We consider a swarm of $d/2$ agents in the two-dimensional plane, each equipped with motion sensors and governed by the following dynamics:

$$\forall\, 1 \le i \le d/2, \qquad \frac{d}{dt}X_i = u_i(t) + w_i(t), \tag{5.13}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

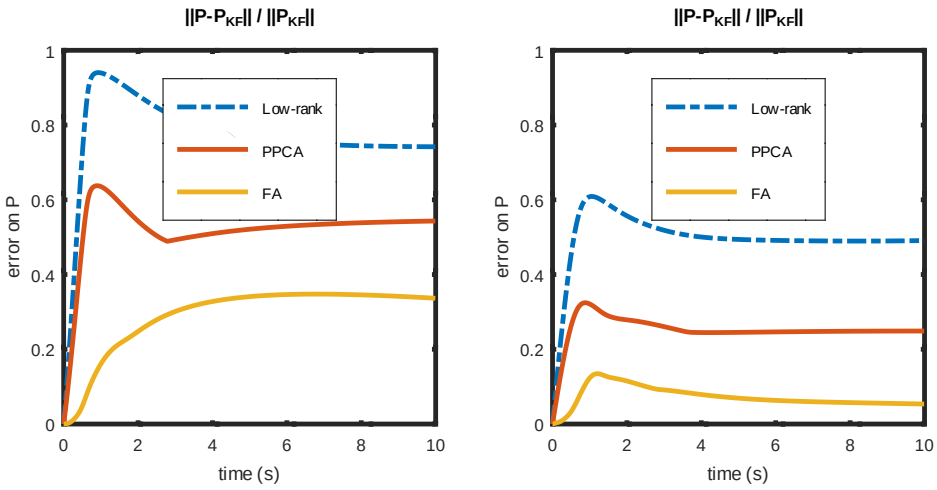


FIG. 1. *$d = 200$ with $p = 8$ (left) and $p = 50$ (right). Normalized distance between the covariance matrix computed from the true full-rank Riccati equation and the ones computed from the low-rank, low-rank + diagonal (FA), and low-rank + isotropic diagonal (PPCA) for the swarm example.*

where $X_i \in \mathbb{R}^2$ denotes the position of agent $i$, $w_i(t)$ is white noise, and $u_i(t)$ is a control input. Measurements consist of relative position between some agents being neighbors in a visibility graph, corrupted by noise $v_{ij}(t)$, i.e.,

$$y_{ij}(t) = X_j(t) - X_i(t) + v_{ij}(t), \quad (i,j) \in \Gamma. \tag{5.14}$$

Moreover, there is a "queen," say, agent 1, having relatively more computational capacity onboard—albeit limited—and equipped with a GPS; that is, we also measure $y_1(t) = X_1(t) + v_1(t)$. The queen receives the dynamical motions $u_i(t)$ as well as the measurements $y_{ij}(t)$ and estimates the state of the whole swarm onboard through a low-rank filter compatible with its modest computational capabilities.

We integrate the Riccati equation with an Euler scheme (with step 0.01 seconds) during 10 seconds and consider an initial covariance matrix factorized as purely low-rank $P_0 = URU^T$ with $R = 2I_p$ and $U$ a random Stiefel matrix to allow for a common starting point. We set noise covariance to be $N = 2I$ and process noise covariance $Q = D$, where $D$ is a diagonal matrix whose diagonal consists of positive values dispersed around 1, reflecting discrepancies in the accuracy of motion sensors of the agents. In our experiments, each agent sees one other agent (that might be the queen), randomly picked at the beginning. We tested changing the visibility graph over time, but that does not change the nature of the following results.

We compare the low-rank approximation [16, 19] with the two variants of our low-rank + diagonal approximation in dimension $d = 200$, that is, 100 agents, and a latent dimension $p = 8$ or $p = 50$ in Figure 1. We clearly observe that the proposed approximations outperform the former in terms of distance to the true covariance matrix. Moreover, the order of the curves is as expected: Projections onto an increasing sequence of submanifolds yield in turn increasing accuracy for the matrix differential equation approximation.

To assess the effect of approximating the Riccati equation on the Kalman filter's state estimates, we have compared the full-rank KF's optimal estimates with those obtained by its computationally cheaper variants for a randomly distributed initial error in a noise-free setting, that is, when using the Kalman filter as an observer, to

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

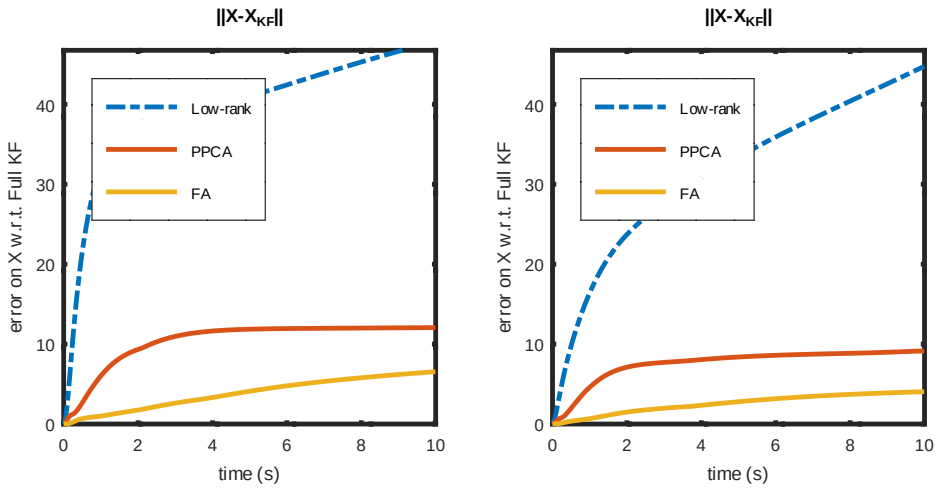


FIG. 2. $d = 200$ *with* $p = 8$ *(left) and* $p = 50$ *(right). Norm of the error over time between the filters' state estimates and the full KF's optimal estimate, for a randomly picked initial error in a noise-free setting.*

get more legible curves. The results are given in Figure 2. We see that the deviation to the optimal estimates is much more contained when adding a diagonal matrix.

It is striking to see that the PPCA method with $p = 8$ competes with the low-rank method of [19] (arXiv preprint version) with $p = 50$ in terms of covariance matrix approximation at the expense of only one more scalar parameter. We also observe that PPCA with $p = 8$ outperforms low-rank with $p = 50$ in terms of state estimate accuracy. More generally, the experiments show that adding a diagonal term provides an efficient alternative to increasing the rank.

Two remarks are in order. First, we note that although the curves indicate a clear improvement of FA over PPCA, this is in fact largely due to the choice of anisotropic process noise, reflecting a discrepancy in the motion sensors' accuracy. When making the problem more "isotropic," the differences between the two diminish. Then we also observed that the initial condition $P(0)$ plays an important role. Although we took $P(0) \in \mathrm{S}^+(\mathrm{p}, \mathrm{d})$ to allow for a common starting point, we noted that departing from $P(0) \in \mathrm{S}^+_{\mathrm{isot}}(\mathrm{p}, \mathrm{d})$ naturally reduces the gap between FA and PPCA.

**5.2.1. Source code.** The code is available on Github for Octave or MATLAB at https://github.com/marc-h-lambert/Riccati-PPCA. It provides projections onto $\mathcal{T}\mathrm{S}^+(\mathrm{p}, \mathrm{d})$, $\mathcal{T}\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p}, \mathrm{d})$, and $\mathcal{T}\mathrm{S}^+_{\mathrm{diag}}(\mathrm{p}, \mathrm{d})$ of matrices of the form $H = GG^T$ or diagonal or a linear combination. Scripts are provided to redo the numerical experiments.

**5.2.2. Interest of invertible approximations in Kalman filtering.** We conclude this section with a discussion about the relevance of low-rank approximations in Kalman filtering. To illustrate the shortcomings of purely low-rank–based methods, e.g., [19, 30, 5, 16, 20] for Kalman filtering, we consider a tutorial case; see Example 6.2.10 of [25]. Consider noisy observations of a Brownian motion in a large dimension

$$dX = dw, \quad dY = dX + dv, \quad P(t_0) = \bar{P}$$

with $w, v$ Wiener process noises with covariance matrices $Q = \lambda I$, $N = \nu I$. Assume that one wants to filter the noise out.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

*Low-rank KF.* The low-rank Riccati equation of [4, 19, 20] writes as

$$\dot{U} = 0, \quad \dot{R} = U^T Q U - R U^T N^{-1} U R,$$

as we have $\dot{U} = (I - UU^T)QUR^{-1} = \lambda(I - UU^T)UR^{-1}$. As a result, $R$ stabilizes to the steady-state solution $\sqrt{\lambda\nu} I_p$. Let us see how the corresponding steady-state Kalman filter updates the state. The static Kalman gain writes as $K = P_\infty C^T N^{-1} = \sqrt{\lambda/\nu}\, U_0 U_0^T$, where $U_0$ is the initial value of $U$. The steady-state KF equations are

$$d\hat{X} = K(dY - \hat{X}dt) = \sqrt{\lambda/\nu}\, UU^T(dY - \hat{X}dt).$$

We see that $P$ is a degenerate low-rank matrix and that the innovation vector $dY - \hat{X}dt$ is always projected onto the same subspace by the operator $UU^T$.

*Low-rank plus diagonal KF in PPCA form.* By contrast, (5.3)–(5.5) write

$$\dot{U} = 0, \quad \dot{R} = \lambda I_p - \frac{1}{\nu} R^2, \quad \dot{s} = \lambda - \frac{s^2}{\nu}.$$

The steady-state solution gives $R = \sqrt{\lambda\nu} I_p$ and $s = \sqrt{\lambda\nu}$. The associated KF equations write now as

$$P = URU^T + s(I - UU^T) = \sqrt{\lambda\nu} I,$$
$$d\hat{X} = K(dY - \hat{X}dt) = \sqrt{\lambda/\nu}\,(dY - \hat{X}dt).$$

In this particular example, we see that we recover the equations of the full-rank steady-state Kalman filter.

PROPOSITION 7. *Using the low-rank KF, the estimation error incurred grows unbounded* $E(||X - \hat{X}||^2) \to +\infty$. *By contrast, using our low-rank plus diagonal approximation,* $E(||X - \hat{X}||^2) \to \sqrt{\lambda/\nu} d$. *Hence, the error is here optimally contained.*

*Proof.* For the low-rank KF, we have $d[(I - UU^T)(X - \hat{X})] = (I - UU^T)dw$. Thus, $\frac{d}{dt} E(||X - \hat{X}||^2) \geq (d - p)\lambda$. By contrast, for the proposed KF, the estimation error is an Ornstein–Uhlenbeck process whose norm stabilizes at $\sqrt{\lambda/\nu} d$. ∎

Contrary to the robotics example, the latter problem obviously lends itself to the proposed approximation due to $C, Q, N$ being diagonal and isotropic, leading to a diagonal and isotropic (stationary) covariance that is fully recovered on $\mathrm{S}^+_{\mathrm{isot}}(\mathrm{p}, \mathrm{d})$. However, it clearly illustrates the interest of maintaining an invertible covariance matrix in Kalman filtering.

**6. Conclusion.** We have derived the orthogonal projection of any tangent vector to the set of PSD matrices onto the tangent space to low-rank plus diagonal PSD matrices. This allows for an attractive alternative to previous low-rank approximations of differential equations defined on the set of PSD matrices in that it retains their computational efficiency while allowing for more flexible, full-rank approximations, leading to better results for large-scale filtering. Our conclusion is as follows:

1. Existing (pure) low-rank PSD approximations have the shortcoming of having many null eigenvalues, which comes with consequences. If they model a Gaussian covariance matrix, we cannot compute the precision matrix, and there is no density. In Kalman filtering, this results in overconfidence, potentially leading to large error drift in turn.
2. The proposed PPCA approximation is about as efficient in terms of memory and computational cost while having none of those drawbacks. We recommend its systematic use.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

3. The proposed FA approximation additionally closely captures the diagonal elements. However, it comes at a greater price albeit retaining linearity in $d$. Its use shall be reserved for problems known to have a strongly nonisotropic diagonal, which one wants to capture well.

As a perspective, we see that while PPCA is computationally efficient, it does not reach the error accuracies achieved by the FA format. It could prove useful to seek a practical criterion for switching between these two formats. Besides, a criterion for choosing the rank $p$ (or adapting it dynamically over time) may also prove useful. A first step to address those perspectives is to observe that the approximation error $\|H - P(H)\|^2$ can be computed at all times with a cost being linear in $d$ too, allowing for approximation quality assessment, as shown in Appendix B. An other interesting perspective would be to explore how one could bring the proposed method to bear on the dynamic FA problem; see e.g., [12].

**Appendix A. Proof of Lemma 1.** Letting $\delta\psi_1, \cdots, \delta\psi_d$ denote the diagonal elements of $\delta\psi$, the cost (3.13) writes as

$$\begin{aligned}\bar{C}(\delta\psi) &= \sum_{i=1}^{n}(h_{ii} - \delta\psi_i)^2 + 4\mathrm{Tr}(UU^T H\delta\psi) - 2\mathrm{Tr}(UU^T\delta\psi^2)\\ &\quad - 2\mathrm{Tr}\big(U^T HUU^T\delta\psi U\big) + \mathrm{Tr}\big(U^T\delta\psi UU^T\delta\psi U\big)\\ &:= ① + ② + ③ + ④ + ⑤.\end{aligned}$$

Let us analyze each term and write them as functions of $\overline{\delta\psi} := (\delta\psi_1, \cdots, \delta\psi_d)^T = \mathrm{diag}(\delta\psi)$,

$$\begin{aligned}① &= \sum_{i=1}^{n}(h_{ii} - \delta\psi_i)^2 = (\bar{h} - \overline{\delta\psi})^T(\bar{h} - \overline{\delta\psi}), \quad \bar{h} := \mathrm{diag(H)},\\ ② &= 4\mathrm{Tr}(UU^T H\delta\psi) = 4\bar{h}_U^T\overline{\delta\psi}, \quad \bar{h}_U := \mathrm{diag(UU^TH)},\\ ③ &= -2\mathrm{Tr}(UU^T\delta\psi^2) = -2\sum_{k=1}^{d}\delta\psi_k^2\left(\sum_{j=1}^{r}U_{kj}^2\right) = -2\overline{\delta\psi}^T\bar{D}\overline{\delta\psi},\end{aligned}$$

where we let $\bar{D}$ be the diagonal matrix defined by $\bar{D}_{ii} = \sum_{j=1}^{r}U_{ij}^2$. As concerns the fourth term, we have

$$\begin{aligned}④ &= -2\mathrm{Tr}\big(U^T HUU^T\delta\psi U\big)\\ &= -2\sum_{1\le i,j\le r}(U^T HU)_{ij}\left(\sum_{k=1}^{d}\delta\psi_k U_{ki}U_{kj}\right)\\ &= -2\sum_{k=1}^{d}\delta\psi_k\sum_{1\le i,j\le r}(U^T HU)_{ij}U_{ki}U_{kj}\\ &= -2\Lambda^T\overline{\delta\psi}, \quad \Lambda = (\Lambda_1, \cdots, \Lambda_d)^T \in \mathbb{R}^d\end{aligned}$$

with $\Lambda_k = \sum_{1\le i,j\le r}(U^T HU)_{ij}U_{ki}U_{kj}$ for each $k$. Finally,

$$\begin{aligned}⑤ &= \mathrm{Tr}\big(U^T\delta\psi UU^T\delta\psi U\big) = \sum_{1\le i,j\le r}\left(\sum_{k=1}^{d}\delta\psi_k U_{ki}U_{kj}\right)^2\\ &= \sum_{1\le i,j\le r}(\Gamma_{ij}^T\overline{\delta\psi})^2,\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where each $\Gamma_{ij} \in \mathbb{R}^d$ is a vector whose $k$th component is $U_{ki}U_{kj}$. Thus, we seek to minimize

$$\bar{C}(\overline{\delta\psi}) = (\bar{h} - \overline{\delta\psi})^T(\bar{h} - \overline{\delta\psi}) + 4\bar{h}_U^T\overline{\delta\psi} - 2\overline{\delta\psi}^T\bar{D}\overline{\delta\psi} - 2\Lambda^T\overline{\delta\psi} + \sum_{1\le i,j\le r}(\Gamma_{ij}^T\overline{\delta\psi})^2.$$

Thus,

$$\nabla_{\overline{\delta\psi}}\bar{C}(\overline{\delta\psi}) = 2(\overline{\delta\psi} - \bar{h}) + 4\bar{h}_U - 4\bar{D}\overline{\delta\psi} - 2\Lambda + 2\sum_{1\le i,j\le r}\Gamma_{ij}\Gamma_{ij}^T\overline{\delta\psi}$$

$$\Rightarrow \frac{1}{2}\nabla_{\overline{\delta\psi}}\bar{C}(\overline{\delta\psi}) = \left(I - 2\bar{D} + \sum_{1\le i,j\le r}\Gamma_{ij}\Gamma_{ij}^T\right)\overline{\delta\psi} - \Lambda + 2\bar{h}_U - \bar{h}.$$

And the optimizer is given by

$$\overline{\delta\psi}^* = \left(I - 2\bar{D} + \sum_{1\le i,j\le r}\Gamma_{ij}\Gamma_{ij}^T\right)^{-1}\big(\bar{h} - 2\bar{h}_U + \Lambda\big).$$

Finally, we note that $\sum_{1\le i,j\le r}\Gamma_{ij}\Gamma_{ij}^T = \Upsilon\Upsilon^T$, where $\Upsilon$ is a $d \times p^2$ matrix whose columns consist of the $p^2$ vectors $\Gamma_{ij}$. To avoid double counting, we may take for $\Upsilon$ a matrix of size $d \times \frac{p(p+1)}{2}$ whose first $\frac{p(p-1)}{2}$ columns are given by the vectors $\sqrt{2}\Gamma_{ij}$, $i < j$, and whose last $r$ columns are the $\Gamma_{ii}$'s.

**Appendix B. Monitoring the quality of approximation.** An advantage of the method is that one may monitor the quality of the approximation by computing the discrepancy $\|H - P(H)\|^2$ at all times, with a cost being linear in $d$. This opens up for methods that could dynamically adapt the parameter $p$ or switch between PPCA and FA. Indeed, we have the following result.

PROPOSITION 8. *In the PPCA case, we have*

$$\|H - P_{U,R,s}(H)\|^2 = \mathrm{Tr}\big((I - UU^T)[H - \delta sI]^T[H - \delta sI]\big).$$

*This may be compared to* $\|H\|^2 = \|H - P(H)\|^2 + \|P(H)\|^2$ *to assess the quality of the approximation.*

As for the FA case, we have advantageously rewritten $\|H - P_{U,R,\psi}(H)\|^2$ as the vector cost (4.2), allowing for the following result.

PROPOSITION 9. *In the FA case,* $\|H - P_{U,R,\psi}(H)\|^2$ *may be computed as* (4.2) *letting*

$$\alpha = \mathrm{Tr}\big(H^2\big) - 2\mathrm{Tr}\big(UU^TH^2\big) + \mathrm{Tr}\big(UU^THUU^TH\big) - \bar{h}^T\bar{h}.$$

*This may be compared to* $\|H\|^2 = \|H - P(H)\|^2 + \|P(H)\|^2$ *to assess the quality of the approximation.*

*Proof.* To compute the constant $\alpha$ in (4.2), we write $\tilde{C}(0) = \alpha + \bar{h}^T\bar{h}$. This is also equal to (3.13) at $\delta\psi = 0$, i.e., letting $\tilde{H} = H$ in the expression. This yields $\alpha$. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

**Acknowledgments.** We thank Pierre-Antoine Absil and Guillaume Olikier for enlightening discussions about the submanifold structure of $S^+(p, d)$. We also thank the anonymous reviewers for their suggestions.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Vol. 78, Princeton University Press, Princeton, NJ, 2008.
[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, *An introduction to MCMC for machine learning*, Mach. Learn., 50 (2003), pp. 5–43.
[3] D. Bertsimas, M. S. Copenhaver, and R. Mazumder, *Certifiably optimal low rank factor analysis*, J. Mach. Learn. Res., 18 (2017), pp. 907–959.
[4] S. Bonnabel and R. Sepulchre, *Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1055–1070.
[5] S. Bonnabel and R. Sepulchre, *The geometry of low-rank Kalman filters*, in Matrix Information Geomentry, F. Nielsen and R. Bhatia, eds., Springer, Berlin, 2013, pp. 53–68.
[6] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a Matlab toolbox for optimization on manifolds*, J. Mach. Learn. Res., 15 (2014), pp. 1455–1459.
[7] C. L. Bris and P. Rouchon, *Low rank approximation for the numerical simulation of high dimensional Lindblad and Riccati equations*, Phys. Rev. A, 87 (2013), 022125.
[8] G. Ceruti and C. Lubich, *Time integration of symmetric and anti-symmetric low-rank matrices and Tucker tensors*, BIT, 60 (2020), pp. 591–614.
[9] V. Ciccone, A. Ferrante, and M. Zorzi, *An alternating minimization algorithm for factor analysis*, Kybernetika, 55 (2019), pp. 740–754.
[10] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
[11] G. Evensen, *Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics*, J. Geophys. Res, 99 (1994), pp. 10143–10162.
[12] L. Falconi, A. Ferrante, and M. Zorzi, *A robust approach to ARMA factor modeling*, IEEE Trans. Automat. Control, 69 (2024), pp. 828–841.
[13] H. Harman, *Modern Factor Analysis*, University of Chicago Press, Chicago, 1967.
[14] U. Helmke and M. A. Shayman, *Critical points of matrix least square distance functions*, in System Structure and Control 1992, Elsevier, New York, 1992, pp. 116–118.
[15] N. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J .C. Gover and S. Barnett, eds., Oxford University Press, Oxford, 1989, pp. 1–27.
[16] O. Koch and C. Lubich, *Dynamical low-rank approximation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 434–454.
[17] M. Lambert, S. Bonnabel, and F. Bach, *The recursive variational Gaussian approximation (R-VGA)*, Stat. Comput., 32 (2022), 10.
[18] M. Lambert, S. Chewi, F. Bach, S. Bonnabel, and P. Rigollet, *Variational inference via Wasserstein gradient flows*, in Advances in Neural Information Processing Systems (NeurIPS), 2022.
[19] C. Le Bris and P. Rouchon, *Low-rank numerical approximations for high-dimensional Lindblad equations*, Phys. Rev. A, 87 (2013), 022125.
[20] M. Le Provost, R. Baptista, Y. Marzouk, and J. D. Eldredge, *A low-rank ensemble Kalman filter for elliptic observations*, Proc. Roy. Soc. A, 478 (2022), 20220182.
[21] W. Lin, M. Khan, and M. Schmidt, *Stein's Lemma for the Reparameterization Trick with Exponential Family Mixtures*, preprint, arXiv:1910.13398, 2019.
[22] C. Lubich and I. V. Oseledets, *A projector-splitting integrator for dynamical low-rank approximation*, BIT, 54 (2014), pp. 171–188.
[23] A. Nonnenmacher and C. Lubich, *Dynamical low-rank approximation: Applications and numerical experiments*, Math. Comput. Simul., 79 (2008), pp. 1346–1357.
[24] E. Oja, *Simplified neuron model as a principal component analyzer*, J. Math. Biol., 15 (1982), pp. 267–273.
[25] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer Science & Business Media, New York, 2013.
[26] J. Schmidt, P. Hennig, J. Nick, and F. Tronarp, *The rank-reduced Kalman filter: Approximate dynamical-low-rank filtering in high dimensions*, in Advances in Neural Information Processing Systems (NIPS) Conference, Vol. 36, 2023, pp. 61364–61376.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[27] A. Shapiro, *Rank-reducibility of a symmetric matrix and sampling theory of minimum trace factor analysis*, Psychometrika, 47 (1982), pp. 187–199.
[28] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, *Simultaneous localization and mapping with sparse extended information filters*, Int. J. Robot. Res., 23 (2004), pp. 693–716.
[29] M. E. Tipping and C. M. Bishop, *Probabilistic principal component analysis*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 61 (1999), pp. 611–622.
[30] D. Tuan Pham, J. Verron, and M. C. Roubaud, *A singular evolutive extended Kalman filter for data assimilation in oceanography*, J. Marine Syst., 16 (1998), pp. 323–340.
[31] B. Vandereycken, *Riemannian and Multilevel Optimization for Rank-Constrained Matrix Problems*, Faculty of Engineering, Katholieke Universiteit Leuven, 2010.

Downloaded 09/09/24 to 109.221.135.229 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.