SIAM J. MATRIX ANAL. APPL.
Vol. 41, No. 2, pp. 852–870

COMPUTATIONALLY EFFICIENT DECOMPOSITIONS OF OBLIQUE PROJECTION MATRICES*

JOHANNES J. BRUST†, ROUMMEL F. MARCIA‡, AND COSMIN G. PETRA§

Abstract. Oblique projection matrices arise in problems in weighted least squares, signal processing, and optimization. While these matrices can be potentially very large, their low-rank structure can be exploited for efficient computation. We propose fast and scalable algorithms for computing their eigendecomposition and singular value decomposition (SVD). Numerical experiments that compare our proposed approaches to existing methods, including randomized SVD, are presented. In addition, we test their accuracy on linear systems from equality constrained optimization problems.

Key words. oblique projection matrices, eigendecomposition, singular value decomposition, SVD, randomized singular value decomposition, projections

AMS subject classifications. 65F15, 65F30, 15B99

DOI. 10.1137/19M1288115

1. Introduction.

An oblique projection matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ is defined by the properties

$$\mathbf{WW} = \mathbf{W} \quad \text{and} \quad \mathbf{W} \neq \mathbf{W}^\top. \tag{1.1}$$

Such matrices arise, for example, in weighted least-squares problems of the form

$$\mathbf{b}^* = \underset{\mathbf{b}\in\mathbb{R}^m}{\arg\min} \ \|\mathbf{D}^{1/2}(\mathbf{y}_{\text{obs}} - \mathbf{Xb})\|_2^2, \tag{1.2}$$

where $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive diagonal entries, $\mathbf{X} \in \mathbb{R}^{n\times m}$ has full column rank, and $\mathbf{y}_{\text{obs}} \in \mathbb{R}^n$ is a known vector of data observations (see [6]). The projection of the solution is given by $\mathbf{Xb}^* = \mathbf{W}\mathbf{y}_{\text{obs}}$, where $\mathbf{W}$ is the oblique projection matrix

$$\mathbf{W} = \mathbf{X}(\mathbf{X}^\top\mathbf{DX})^{-1}\mathbf{X}^\top\mathbf{D}. \tag{1.3}$$

In this paper, we consider the more general oblique projection

$$\mathbf{W} = \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top, \tag{1.4}$$

where we assume $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times m}$ have full column rank and $\mathbf{Y}^\top\mathbf{X}$ is invertible. Such oblique projections appear, for example, in block quasi-Newton matrices from optimization (see [3]). We present the eigendecomposition of oblique (nonsymmetric)

*Received by the editors September 18, 2019; accepted for publication (in revised form) by J. Liesen March 6, 2020; published electronically June 3, 2020.
https://doi.org/10.1137/19M1288115
Funding: The work of the authors was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research under grant DE-AC02-06CH11357 at Argonne National Laboratory through the project "Multifaceted Mathematics for Complex Energy Systems." The work of the second author was partially supported by the National Science Foundation grant IIS 1741490. The work of the third author was supported by the LDRD Program of Lawrence Livermore National Laboratory under projects 16-ERD-025, 17-SI-005.

†Argonne National Laboratory, Mathematics and Computer Science Division, Lemont, IL 60439 (jbrust@anl.gov).
‡Department of Applied Mathematics, University of California, Merced, CA 95343 (rmarcia@ucmerced.edu).
§Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 (petra1@llnl.gov).

projection matrices and develop efficient algorithms to compute their singular values and singular vectors.

1.1. Related work. Oblique projection matrices arise in contexts such as systems of linear inequalities, constrained optimization, and signal processing [1, 2, 4]. Stewart [14] and O'Leary [13] proposed bounds on the spectral norm of the oblique projection in (1.3). In [1], Behrens and Scharf considered related oblique projections, which are defined by a matrix $\mathbf{Z}$, whose columns form a basis for the nullspace of $\mathbf{W}$. In particular, $\mathbf{W}$ in [1] is represented as $\mathbf{W} = \mathbf{X}(\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{P}$, where $\mathbf{P}$ is the orthogonal projection onto the orthogonal complement of the range space of $\mathbf{Z}$ (i.e., $\mathbf{P}\mathbf{Z} = \mathbf{0}$). Instead of bounding the spectral norm of $\mathbf{W}$, Behrens and Scharf showed a general procedure to compute the singular values of $\mathbf{W}$. However, how to compute the singular values of the oblique complement $\widehat{\mathbf{W}} = \mathbf{I} - \mathbf{W}$ remains open.

In this article, we derive the expressions for the singular values of $\widehat{\mathbf{W}}$ and computationally efficient expressions for the ones of $\mathbf{W}$. We derive the singular values under the assumption that $\mathbf{Y}^\top\mathbf{X}$ may not be symmetric. In addition, we develop explicit formulas for *singular vectors* and propose efficient methods for computing a SVD, even for large n.

In numerical experiments, we compare our methods on matrices with general $\mathbf{Y}^\top\mathbf{X}$ to the symmetric cases of Stewart and O'Leary and Behrens and Scharf. For large values of n, we compare our proposed algorithms to the randomized singular value decomposition (SVD) (see [9] for probabilistic algorithms), which is often used for approximating the SVD of large matrices. Finally, we apply our factorizations to decompose block BFGS quasi-Newton matrices from large-scale optimization (see [3]).

2. Eigendecomposition. In this section, we describe an eigendecomposition of $\mathbf{W}$ and of its *oblique complement*, $\widehat{\mathbf{W}}$, which is defined as

$$\widehat{\mathbf{W}} = \mathbf{I} - \mathbf{W} = \mathbf{I} - \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top. \tag{2.1}$$

While $\mathrm{Range}(\mathbf{W}) = \mathrm{Range}(\mathbf{X})$ and $\mathrm{Null}(\mathbf{W}) = \mathrm{Null}(\mathbf{Y}^\top)$, in contrast, $\mathrm{Range}(\widehat{\mathbf{W}}) = \mathrm{Null}(\mathbf{Y}^\top)$ and $\mathrm{Null}(\widehat{\mathbf{W}}) = \mathrm{Range}(\mathbf{X})$. Note that $\widehat{\mathbf{W}}$ is also an oblique projection matrix. Subsequently, observe that $\mathbf{W}\mathbf{X} = \big(\mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\big)\mathbf{X} = \mathbf{X}$, which implies that $\mathrm{Range}(\mathbf{X})$ is the eigenspace of $\mathbf{W}$ that corresponds to the eigenvalue 1. Next we derive an expression for the remaining eigenvectors of $\mathbf{W}$.

Let $\mathbf{X} = \mathbf{Q}_\parallel\mathbf{R}_\parallel$ be the "thin" QR factorization of $\mathbf{X}$, where $\mathbf{Q}_\parallel \in \mathbb{R}^{n\times m}$ has orthonormal columns and $\mathbf{R}_\parallel \in \mathbb{R}^{m\times m}$ is upper triangular and nonsingular. Let $\mathbf{Q}_\perp \in \mathbb{R}^{n\times(n-m)}$ be a matrix whose columns are orthonormal (i.e., $\mathbf{Q}_\perp^\top\mathbf{Q}_\perp = \mathbf{I}$) and are orthogonal to the columns of $\mathbf{Q}_\parallel$ (i.e., $\mathbf{Q}_\perp^\top\mathbf{Q}_\parallel = \mathbf{0}$). Let $\mathbf{Q} = \begin{bmatrix}\mathbf{Q}_\parallel & \mathbf{Q}_\perp\end{bmatrix}$ so that

$$\mathbf{I} = \mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top + \mathbf{Q}_\perp\mathbf{Q}_\perp^\top. \tag{2.2}$$

In this section, we will show that the columns of the matrix

$$\mathbf{S} = \begin{bmatrix}\mathbf{X} & \big(\mathbf{I} - \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\big)\mathbf{Q}_\perp\end{bmatrix} \tag{2.3}$$

are eigenvectors of $\mathbf{W}$. First, we show how to express its inverse explicitly through the following lemma.

LEMMA 2.1. *The square matrix* $\mathbf{S}$ *in* (2.4) *has the explicit inverse*

$$\mathbf{S}^{-1} = \begin{bmatrix}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top \\ \mathbf{Q}_\perp^\top\end{bmatrix}. \tag{2.4}$$

Proof. Using the expressions $\mathbf{Q}_\perp\mathbf{Q}_\perp^\top = \left(\mathbf{I} - \mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top\right)$ from (2.2) and $\mathbf{X}\mathbf{R}_\parallel^{-1} = \mathbf{Q}_\parallel$ from the QR factorization of $\mathbf{X}$ and labeling the right-hand side in (2.4) as $\mathbf{S}^{\text{inv}}$, we see that

$$\begin{aligned}\mathbf{S}\mathbf{S}^{\text{inv}} &= \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top + \left(\mathbf{I} - \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\right)\mathbf{Q}_\perp\mathbf{Q}_\perp^\top \\ &= \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top + \left(\mathbf{I} - \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\right)\left(\mathbf{I} - \mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top\right) \\ &= \mathbf{I} - \mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top + \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top \\ &= \mathbf{I} - \mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top + \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\mathbf{X}\mathbf{R}_\parallel^{-1}\mathbf{Q}_\parallel^\top \\ &= \mathbf{I}.\end{aligned}$$

Thus, the expression in (2.4) is the inverse of $\mathbf{S}$ in (2.3). ∎

Having defined $\mathbf{S}$ and its inverse, we can now express an eigendecomposition of $\mathbf{W}$ and $\widehat{\mathbf{W}}$.

THEOREM 2.2. *The oblique projection matrix* $\mathbf{W} = \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top$ *has an eigendecomposition of the form*

$$\mathbf{W} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}, \tag{2.5}$$

where $\mathbf{S}$ *and* $\mathbf{S}^{-1}$ *are defined in* (2.3) *and* (2.4)*, respectively, and*

$$\mathbf{\Lambda} = \begin{bmatrix}\mathbf{I}_m & \\ & \mathbf{0}_{n-m}\end{bmatrix}. \tag{2.6}$$

Moreover, the oblique complement $\widehat{\mathbf{W}} = \mathbf{I} - \mathbf{W}$ *has an eigendecomposition of the form*

$$\widehat{\mathbf{W}} = \mathbf{S}\begin{bmatrix}\mathbf{0}_m & \\ & \mathbf{I}_{n-m}\end{bmatrix}\mathbf{S}^{-1}. \tag{2.7}$$

Proof. The proof follows from straightforward computations. ∎

Note that the decompositions from Theorem 2.2 are not unique because replacing $\mathbf{Q}_\perp$ by, say, $\mathbf{Q}_\perp\mathbf{U}_\mathbf{Q}$, where $\mathbf{U}_\mathbf{Q} \in \mathbb{R}^{(n-m)\times(n-m)}$ is any orthogonal matrix, yields an eigendecomposition with different eigenvectors.

3. Singular value decomposition. In this section, we first describe how to obtain the singular values of $\mathbf{W}$ and $\widehat{\mathbf{W}}$. Then given the singular values, we demonstrate how singular vectors can be computed. Finally, we show the relationships between the singular vectors of $\mathbf{W}$ and $\widehat{\mathbf{W}}$. Throughout the article we will assume that $(n-m) \geq m$.

3.1. Singular values of W and $\widehat{\mathbf{W}}$. To compute the singular values of $\mathbf{W}$ and $\widehat{\mathbf{W}}$, we will use the matrix

$$\mathbf{\Omega} = \mathbf{Q}_\parallel^\top\mathbf{W}\mathbf{Q}_\perp, \tag{3.1}$$

where $\mathbf{\Omega} \in \mathbb{R}^{m\times(n-m)}$, and whose SVD is

$$\mathbf{\Omega} = \mathbf{U}_\mathbf{\Omega}\mathbf{\Gamma}\mathbf{V}_\mathbf{\Omega}^\top \tag{3.2}$$

with $\mathbf{U}_\mathbf{\Omega} \in \mathbb{R}^{m\times m}$ and $\mathbf{V}_\mathbf{\Omega} \in \mathbb{R}^{(n-m)\times(n-m)}$ orthogonal and $\mathbf{\Gamma} \in \mathbb{R}^{m\times(n-m)}$ a rectangular diagonal with diagonal entries $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_m > 0$. We now explicitly define the singular values of $\mathbf{W}$ and $\widehat{\mathbf{W}}$.

THEOREM 3.1. *The singular values of the oblique projection matrix* $\mathbf{W}$ *from* (1.4) *are*

$$\sigma_i = \begin{cases} \sqrt{1+\gamma_i^2}, & 1 \le i \le m, \\ 0 & \textit{otherwise}, \end{cases} \tag{3.3}$$

while the singular values of the oblique complement $\widehat{\mathbf{W}} = \mathbf{I} - \mathbf{W}$ *from* (2.1) *are*

$$\widehat{\sigma}_i = \begin{cases} \sqrt{1+\gamma_i^2}, & 1 \le i \le m, \\ 1, & m+1 \le i \le n-m, \\ 0 & \textit{otherwise}, \end{cases} \tag{3.4}$$

where the γ_i*'s in* (3.3) *and* (3.4) *are the nonzero singular values of* $\mathbf{\Omega}$ *from* (3.2).

Proof. First, note that $\mathbf{Q}_{\|}^{\top}\mathbf{W}\mathbf{Q}_{\|} = \mathbf{Q}_{\|}^{\top}\mathbf{X}(\mathbf{Y}^{\top}\mathbf{X})^{-1}\mathbf{Y}^{\top}\mathbf{X}\mathbf{R}_{\|}^{-1} = \mathbf{I}$ since $\mathbf{Q}_{\|} = \mathbf{X}\mathbf{R}_{\|}^{-1}$. Then, using the orthogonal matrix $\mathbf{Q}_{\mathbf{\Omega}} = [\,\mathbf{Q}_{\|}\mathbf{U}_{\mathbf{\Omega}} \quad \mathbf{Q}_{\perp}\mathbf{V}_{\mathbf{\Omega}}\,]$, we have

$$\mathbf{Q}_{\mathbf{\Omega}}^{\top}\mathbf{W}\mathbf{Q}_{\mathbf{\Omega}} = \begin{bmatrix} \mathbf{I}_m & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_{\mathbf{\Omega}}^{\top}\widehat{\mathbf{W}}\mathbf{Q}_{\mathbf{\Omega}} = \begin{bmatrix} \mathbf{0} & -\mathbf{\Gamma} \\ \mathbf{0} & \mathbf{I}_{n-m} \end{bmatrix}. \tag{3.5}$$

Because eigenvalues are invariant with respect to orthogonal transformations, we deduce that the nonzero eigenvalues of $\mathbf{W}\mathbf{W}^{\top}$ are the diagonal elements of the diagonal matrix $\mathbf{I}_m + \mathbf{\Gamma}\mathbf{\Gamma}^{\top}$. Similarly, the nonzero eigenvalues of $\widehat{\mathbf{W}}^{\top}\widehat{\mathbf{W}}$ are the diagonal elements of $\mathbf{I}_{n-m} + \mathbf{\Gamma}^{\top}\mathbf{\Gamma}$. It follows that the nonzero singular values of $\mathbf{W}$ are $\sqrt{1+\gamma_i^2}$ for $1 \le i \le m$, and the nonzero singular values of $\widehat{\mathbf{W}}$ are $\sqrt{1+\gamma_i^2}$ for $1 \le i \le m$, and 1 with multiplicity $n-2m$, which completes the proof. ∎

Note that the proof from Theorem 3.1 explicitly defines the orthogonal matrix $\mathbf{Q}_{\mathbf{\Omega}}$ and is consistent with the analysis in [11, Problem 2.6, p. 18].

Observe that Theorem 3.1 shows that $\mathbf{W}$ and $\widehat{\mathbf{W}}$ share the nonzero singular values $\sigma_i = \sqrt{1+\gamma_i^2}$ for $1 \le i \le m$, which implies that

$$\|\mathbf{W}\|_2 = \sigma_1 = \sqrt{1+\gamma_1^2} = \|\widehat{\mathbf{W}}\|_2,$$

but

$$\kappa\left(\widehat{\mathbf{W}}\right) = \frac{\sigma_1}{1} \ge \frac{\sigma_1}{\sigma_m} = \kappa\left(\mathbf{W}\right),$$

where $\kappa(\cdot)$ is the condition number of a matrix.

3.2. Singular vectors. In addition to the singular values, we also develop representations of the singular vectors of oblique projections. Let $\mathbf{\Sigma} = \operatorname{diag}(\{\sigma_i\}) \in \mathbb{R}^{m\times m}$ be the diagonal matrix of the singular values of $\mathbf{W}$, where σ_i is defined in (3.3). Singular vectors of $\mathbf{W}$ and $\widehat{\mathbf{W}}$ are given by the following theorems.

THEOREM 3.2. *A reduced SVD of* $\mathbf{W}$ *in* (1.4) *is given as*

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}, \tag{3.6}$$

where

$$\mathbf{U} = \mathbf{Q}_{\|}\mathbf{U}_{\mathbf{\Omega}} \quad \textit{and} \quad \mathbf{V} = \mathbf{Y}(\mathbf{X}^{\top}\mathbf{Y})^{-1}\mathbf{X}^{\top}\mathbf{Q}_{\|}\mathbf{U}_{\mathbf{\Omega}}\mathbf{\Sigma}^{-1} \tag{3.7}$$

with $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times m}$.

Proof. Note that $\mathbf{U}^\top\mathbf{U} = \mathbf{U}_{\boldsymbol{\Omega}}^\top\mathbf{Q}_\parallel^\top\mathbf{Q}_\parallel\mathbf{U}_{\boldsymbol{\Omega}} = \mathbf{I}$, and from (3.5)

$$\mathbf{V}^\top\mathbf{V} \;=\; \boldsymbol{\Sigma}^{-1}\mathbf{U}_{\boldsymbol{\Omega}}^\top\mathbf{Q}_\parallel^\top\mathbf{W}\mathbf{W}^\top\mathbf{Q}_\parallel\mathbf{U}_{\boldsymbol{\Omega}}\boldsymbol{\Sigma}^{-1} \;=\; \boldsymbol{\Sigma}^{-1}(\mathbf{I}+\boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top)\boldsymbol{\Sigma}^{-1} \;=\; \mathbf{I}.$$

Therefore, $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns. Then, using the expressions in (3.7), we obtain

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \mathbf{Q}_\parallel\mathbf{Q}_\parallel^\top\mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top = \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top = \mathbf{W}$$

since $\mathbf{X} = \mathbf{Q}_\parallel\mathbf{R}_\parallel$, which demonstrates the desired result. ∎

We note that by the definition of $\mathbf{U}$ and $\mathbf{V}$ in (3.7), the columns of $\mathbf{U}$ form an orthonormal basis for Range($\mathbf{X}$) and the columns of $\mathbf{V}$ form an orthonormal basis for Range($\mathbf{Y}$). In addition, we observe that

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \mathbf{W} = \mathbf{W}\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top,$$

which means that

$$\mathbf{V}^\top\mathbf{U} = \boldsymbol{\Sigma}^{-1} = \mathbf{U}^\top\mathbf{V}. \tag{3.8}$$

An SVD of $\widehat{\mathbf{W}}$ is given as follows. We assume that singular vectors $\mathbf{U}$ and $\mathbf{V}$ of $\mathbf{W}$ are available (cf. Theorem 3.2).

THEOREM 3.3. *A full SVD of* $\widehat{\mathbf{W}}$ *in* (2.1) *is given as*

$$\widehat{\mathbf{W}} = \widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}}\widehat{\mathbf{V}}^\top = \begin{bmatrix} \widehat{\mathbf{U}}_1 & \widehat{\mathbf{U}}_2 & \mathbf{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma} & & \\ & \mathbf{I}_{n-2m} & \\ & & \mathbf{0}_{m\times m} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{V}}_1^\top \\ \widehat{\mathbf{V}}_2^\top \\ \mathbf{U}^\top \end{bmatrix}, \tag{3.9}$$

where

$$\widehat{\mathbf{U}}_1 = (\mathbf{V}\boldsymbol{\Sigma}^{-1} - \mathbf{U})\mathbf{D}^{-1}, \qquad \widehat{\mathbf{U}}_2 = \widehat{\mathbf{Q}}_\perp, \tag{3.10a}$$

$$\widehat{\mathbf{V}}_1 = (\mathbf{V} - \mathbf{U}\boldsymbol{\Sigma}^{-1})\mathbf{D}^{-1}, \qquad \widehat{\mathbf{V}}_2 = \widehat{\mathbf{Q}}_\perp, \tag{3.10b}$$

with $\widehat{\mathbf{U}} = [\ \widehat{\mathbf{U}}_1 \quad \widehat{\mathbf{U}}_2\ ] \in \mathbb{R}^{n\times(n-m)}$, $\widehat{\mathbf{V}} = [\ \widehat{\mathbf{V}}_1 \quad \widehat{\mathbf{V}}_2\ ] \in \mathbb{R}^{n\times(n-m)}$, *and* $\widehat{\boldsymbol{\Sigma}} = \operatorname{diag}(\boldsymbol{\Sigma}, \mathbf{I}_{n-2m}) \in \mathbb{R}^{(n-m)\times(n-m)}$, *and where* $\widehat{\mathbf{Q}}_\perp \in \mathbb{R}^{n\times(n-2m)}$ *is a matrix whose columns form an orthonormal basis for the orthogonal complement of the space spanned by the columns of* $\mathbf{X}$ *and* $\mathbf{Y}$. *The diagonal matrix* $\mathbf{D} \in \mathbb{R}^{m\times m}$ *satisfies*

$$\mathbf{D}^2 = \mathbf{I}_m - \boldsymbol{\Sigma}^{-2}.$$

Proof. We first verify that $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ have orthonormal columns. Since the columns of $\mathbf{U}$ and $\mathbf{V}$ form an orthonormal basis for Range($\mathbf{X}$) and Range($\mathbf{Y}$), respectively, it follows that

$$\widehat{\mathbf{U}}_1^\top\widehat{\mathbf{U}}_2 = \mathbf{D}^{-1}(\boldsymbol{\Sigma}^{-1}\mathbf{V}^\top - \mathbf{U}^\top)\widehat{\mathbf{Q}}_\perp = \mathbf{0},$$
$$\widehat{\mathbf{V}}_1^\top\widehat{\mathbf{V}}_2 = \mathbf{D}^{-1}(\mathbf{V}^\top - \boldsymbol{\Sigma}^{-1}\mathbf{U}^\top)\widehat{\mathbf{Q}}_\perp = \mathbf{0}$$

from the definition of $\widehat{\mathbf{Q}}_\perp$. Now, recall $\mathbf{V}^\top\mathbf{U} = \boldsymbol{\Sigma}^{-1} = \mathbf{U}^\top\mathbf{V}$ from (3.8). Therefore,

$$\widehat{\mathbf{U}}_1^\top\widehat{\mathbf{U}}_1 = \mathbf{D}^{-1}(\boldsymbol{\Sigma}^{-1}\mathbf{V}^\top - \mathbf{U}^\top)(\mathbf{V}\boldsymbol{\Sigma}^{-1} - \mathbf{U})\mathbf{D}^{-1} = \mathbf{D}^{-1}(-\boldsymbol{\Sigma}^{-2} + \mathbf{I})\mathbf{D}^{-1} = \mathbf{I},$$
$$\widehat{\mathbf{V}}_1^\top\widehat{\mathbf{V}}_1 = \mathbf{D}^{-1}(\mathbf{V}^\top - \boldsymbol{\Sigma}^{-1}\mathbf{U}^\top)(\mathbf{V} - \mathbf{U}\boldsymbol{\Sigma}^{-1})\mathbf{D}^{-1} = \mathbf{D}^{-1}(\ \mathbf{I} - \boldsymbol{\Sigma}^{-2})\mathbf{D}^{-1} \ = \mathbf{I}.$$

Since, by definition, $\widehat{\mathbf{U}}_2^\top \widehat{\mathbf{U}}_2 = \widehat{\mathbf{V}}_2^\top \widehat{\mathbf{V}}_2 = \widehat{\mathbf{Q}}_\perp^\top \widehat{\mathbf{Q}}_\perp = \mathbf{I}$, we conclude that $\widehat{\mathbf{U}}^\top \widehat{\mathbf{U}} = \mathbf{I}$ and $\widehat{\mathbf{V}}^\top \widehat{\mathbf{V}} = \mathbf{I}$.

Noting that $\mathbf{I} = \widehat{\mathbf{V}}_1 \widehat{\mathbf{V}}_1^\top + \widehat{\mathbf{Q}}_\perp \widehat{\mathbf{Q}}_\perp^\top + \mathbf{U}\mathbf{U}^\top$, and using the definition of $\widehat{\mathbf{U}}_1$ and $\widehat{\mathbf{V}}_1$ in (3.10a) and (3.10b), respectively, we have

$$\begin{aligned}
\widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top &= \widehat{\mathbf{U}}_1 \mathbf{\Sigma} \widehat{\mathbf{V}}_1^\top + \widehat{\mathbf{Q}}_\perp \widehat{\mathbf{Q}}_\perp^\top \\
&= \widehat{\mathbf{U}}_1 \mathbf{\Sigma} \widehat{\mathbf{V}}_1^\top + \mathbf{I} - \mathbf{U}\mathbf{U}^\top - \widehat{\mathbf{V}}_1 \widehat{\mathbf{V}}_1^\top \\
&= \mathbf{I} - \mathbf{U}\mathbf{U}^\top + (\widehat{\mathbf{U}}_1 \mathbf{\Sigma} - \widehat{\mathbf{V}}_1) \widehat{\mathbf{V}}_1^\top \\
&= \mathbf{I} - \mathbf{U}\mathbf{U}^\top + \left((\mathbf{V} - \mathbf{U}\mathbf{\Sigma}) - (\mathbf{V} - \mathbf{U}\mathbf{\Sigma}^{-1}) \right) \mathbf{D}^{-2} (\mathbf{V}^\top - \mathbf{\Sigma}^{-1}\mathbf{U}^\top) \\
&= \mathbf{I} - \mathbf{U}\mathbf{U}^\top - \mathbf{U}\mathbf{\Sigma} \left(\mathbf{I} - \mathbf{\Sigma}^{-2} \right) \mathbf{D}^{-2} \left(\mathbf{V}^\top - \mathbf{\Sigma}^{-1}\mathbf{U}^\top \right) \\
&= \mathbf{I} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \\
&= \widehat{\mathbf{W}},
\end{aligned}$$

which is the desired result. ∎

We note that in practice, we do not explicitly compute $\widehat{\mathbf{U}}_2 = \widehat{\mathbf{V}}_2 = \widehat{\mathbf{Q}}_\perp$ because they are large $n \times (n - 2m)$ matrices.

3.3. Practical implementation. In our implementation for computing the SVD of $\mathbf{W}$ and $\widehat{\mathbf{W}}$, we do not compute the matrix $\mathbf{\Omega}$ in (3.1) because it requires explicitly computing the orthogonal complement matrix $\mathbf{Q}_\perp$. Instead, to obtain the singular values, we compute the following. First, we perform the Cholesky factorization of $\mathbf{X}^\top \mathbf{X}$ to obtain $\mathbf{R}_\parallel$ since

$$\mathbf{X}^\top \mathbf{X} = \mathbf{R}_\parallel^\top \mathbf{Q}_\parallel^\top \mathbf{Q}_\parallel \mathbf{R}_\parallel = \mathbf{R}_\parallel^\top \mathbf{R}_\parallel. \tag{3.11}$$

Next, by noting that

$$\mathbf{Q}_\parallel^\top \mathbf{W}\mathbf{W}^\top \mathbf{Q}_\parallel = \mathbf{R}_\parallel (\mathbf{Y}^\top \mathbf{X})^{-1} (\mathbf{Y}^\top \mathbf{Y}) (\mathbf{X}^\top \mathbf{Y})^{-1} \mathbf{R}_\parallel^\top,$$

we can obtain the matrix of singular values $\mathbf{\Sigma}$ from the eigendecomposition of the small $m \times m$ matrix

$$\mathbf{R}_\parallel (\mathbf{Y}^\top \mathbf{X})^{-1} (\mathbf{Y}^\top \mathbf{Y}) (\mathbf{X}^\top \mathbf{Y})^{-1} \mathbf{R}_\parallel^\top = \mathbf{U}_{\mathbf{\Omega}} \mathbf{\Sigma}^2 \mathbf{U}_{\mathbf{\Omega}}^\top.$$

Algorithm 3.1 describes our approach, Reduced Oblique Singular Value Decomposition (RObSVD) for computing a reduced SVD of $\mathbf{W}$ in (3.6). In particular, we compute singular vectors of $\mathbf{W}$ using (3.7) from Theorem 3.2 by noting that $\mathbf{Q}_\parallel = \mathbf{X}\mathbf{R}_\parallel^{-1}$. Algorithm 3.1 can be used to efficiently compute a reduced SVD of $\widehat{\mathbf{W}}$ using (3.10a) and (3.10b) from Theorem 3.3. Excluding the singular vectors in $\widehat{\mathbf{Q}}_\perp$, Algorithm 3.2 describes our approach for computing a reduced SVD of $\widehat{\mathbf{W}}$. Algorithm 3.2 does not compute $\widehat{\mathbf{U}}_2 = \widehat{\mathbf{Q}}_\perp = \widehat{\mathbf{V}}_2$ because this matrix is large and potentially very expensive to compute. Instead, note that the SVD of $\widehat{\mathbf{W}}$ in Theorem 3.3 is defined by the orthogonal projection $\widehat{\mathbf{Q}}_\perp \widehat{\mathbf{Q}}_\perp^\top = \widehat{\mathbf{U}}_2 \widehat{\mathbf{V}}_2^\top$, which we describe how to economically compute in section 4.2.

4. Numerical experiments. This section describes five sets of numerical experiments that compare our proposed algorithms to existing methods:

1. singular values of $\mathbf{W}$ from [1],
2. SVD of $\mathbf{W}$ and $\widehat{\mathbf{W}} = \mathbf{I} - \mathbf{W}$ from [1],

Algorithm 3.1 Reduced Oblique Singular Value Decomposition of $\mathbf{W}$.

1: Input: $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{Y} \in \mathbb{R}^{n\times m}$;
2: $\mathbf{R}_{\|}^{\top}\mathbf{R}_{\|} = \mathbf{X}^{\top}\mathbf{X}$; {Cholesky factorization}
3: $\mathbf{U}_{\mathbf{\Omega}}\mathbf{\Sigma}^2\mathbf{U}_{\mathbf{\Omega}}^{\top} = \mathbf{R}_{\|}(\mathbf{Y}^{\top}\mathbf{X})^{-1}(\mathbf{Y}^{\top}\mathbf{Y})(\mathbf{X}^{\top}\mathbf{Y})^{-1}\mathbf{R}_{\|}^{\top}$; {Eigendecomposition}
4: $\mathbf{U} = \mathbf{X}\mathbf{R}_{\|}^{-1}\mathbf{U}_{\mathbf{\Omega}}$;
5: $\mathbf{V} = \mathbf{Y}(\mathbf{X}^{\top}\mathbf{Y})^{-1}\mathbf{X}^{\top}\mathbf{U}\mathbf{\Sigma}^{-1}$;

Algorithm 3.2 Reduced Oblique Singular Value Decomposition of $\widehat{\mathbf{W}}$.

1: Input: $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{Y} \in \mathbb{R}^{n\times m}$;
2: Compute $\mathbf{U}, \mathbf{\Sigma}$, and $\mathbf{V}$ using Algorithm 3.1;
3: $\mathbf{D} = \sqrt{\mathbf{I} - \mathbf{\Sigma}^{-2}}$;
4: $\widehat{\mathbf{U}}_1 = (\mathbf{V}\mathbf{\Sigma}^{-1} - \mathbf{U})\mathbf{D}^{-1}$;
5: $\widehat{\mathbf{V}}_1 = (\mathbf{V} - \mathbf{U}\mathbf{\Sigma}^{-1})\mathbf{D}^{-1}$;

3. comparisons with the randomized SVD [9],
4. comparisons of the spectral norm $\|\mathbf{W}\|_2$ with the bound from [14] and [13],
5. eigendecomposition and SVD of block quasi-Newton matrices from [3].

The algorithms are implemented in MATLAB and have been tested in MATLAB R2014a (64 bit) and MATLAB R2018b. The codes are available at https://github.com/johannesbrust/AOP. The numerical experiments are carried out on a Dell Precision T1700 desktop computer with Intel i5-4590 CPU @ 3.30GHz x 4 processors, 8 GB RAM, and Linux Ubuntu 14.04, 64-bit, and a Mac Book Pro laptop with Intel Core i7 CPU @ 2.6 GHz processor, 32 GB RAM, and MacOS High Sierra operating system.

4.1. Experiment I. This experiment compares the performance of our proposed method (Algorithm 3.1), the built-in MATLAB function, `svds(W,m)`, which computes the m largest singular values of $\mathbf{W}$, and an approach developed by Behrens and Scharf [1], which we describe in Appendix A.

We compare the performance of our proposed Oblique SVD approach (Algorithm 3.1) with the original Behrens and Scharf approach (Algorithm A.1) and the modified Behrens and Scharf approach (Algorithm A.2) on randomly generated matrices with various dimensions. In the experiments, the full rank matrices $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Z} \in \mathbb{R}^{n\times(n-m)}$ are generated by randomly drawing from a zero-mean normal distribution. Moreover, $\mathbf{Y} = (\mathbf{I} - \mathbf{Z}(\mathbf{Z}^{\top}\mathbf{Z})^{-1}\mathbf{Z}^{\top})\mathbf{X}$ is computed as an input for Algorithm 3.1 and Algorithm A.2. The test matrices have rank $m = 150$ and dimensions n ranging from 800 to $5,000$. For each n, the experiment was performed 10 times with different randomly generated matrices each time. To compare the accuracy of the three approaches, we define the measure

$$\text{Error} = \frac{1}{n}\sqrt{\sum_{i=1}^{m}(\sigma_i - \sigma_i^*)^2}, \tag{4.1}$$

where the σ_i^*'s are the singular values computed using the built-in `svds` MATLAB function and the σ_i's are the singular values computed using the various algorithms. The results of the experiment are reported in Tables 1 and 2.

4.2. Experiment II. In this experiment, we compute reduced SVD factorizations of $\mathbf{W}$ and $\widehat{\mathbf{W}}$ using Algorithms 3.1 and 3.2, respectively. For comparison we

TABLE 1

Experiment I. *Computational times (in seconds) of our proposed RObSVD approach (Algorithm* 3.1*), Behrens and Scharf's approach in Algorithm* A.1*, the modified Behrens and Scharf (Modified Beh-Sch) approach in Algorithm* A.2*, and the built-in MATLAB* `svds` *function for computing the singular values of* $\mathbf{W}$*. For each* n*, the experiment was performed* 10 *times with different randomly generated matrices each time. The average computational times among the* 10 *experiments are reported. Note that Algorithms* 3.1 *and* A.2 *compute the singular values the fastest while the built-in MATLAB function takes the longest.*

n	Time (in sec.)			
	RObSVD	Behrens and Scharf	Modified Beh-Sch	MATLAB `svds`
800	0.0278	0.6636	0.0266	1.8371
1000	0.0284	0.8991	0.0269	2.5957
2000	0.0304	2.2598	0.0277	9.9797
3000	0.0325	3.9002	0.0291	18.4856
4000	0.0346	5.7370	0.0304	31.4324
5000	0.0376	7.9405	0.0316	46.4987

TABLE 2

Experiment I. *Comparison using the error metric in* (4.1) *between our proposed RObSVD approach (Algorithm* 3.1*), Behrens and Scharf's approach in Algorithm* A.1*, and the modified Behrens and Scharf (Modified Beh-Sch) approach in Algorithm* A.2 *for computing the singular values of* $\mathbf{W}$*. We use the singular values from the built-in MATLAB function* `svds` *as the benchmark values. For each* n*, the experiment was performed* 10 *times with different randomly generated matrices each time. The average and maximum error values (among the* 10 *experiments) are reported. Note that the singular values computed from Algorithms* 3.1 *and* A.2 *are closer to the benchmark than those from Algorithm* A.1*.*

n	Average error			Maximum error		
	RObSVD	Behrens and Scharf	Modified Beh-Sch	RObSVD	Behrens and Scharf	Modified Beh-Sch
800	6.865e-11	4.383e-06	8.988e-11	2.830e-10	3.450e-05	4.164e-10
1000	8.058e-07	6.661e-04	1.888e-06	6.438e-06	5.271e-03	1.509e-05
2000	4.906e-11	1.306e-06	9.044e-11	1.982e-10	5.929e-06	5.058e-10
3000	9.923e-10	9.226e-04	3.434e-09	7.386e-09	7.334e-03	2.690e-08
4000	6.672e-11	5.758e-06	3.905e-11	4.456e-10	4.065e-05	2.457e-10
5000	2.724e-09	6.964e-06	1.041e-09	2.148e-08	5.037e-05	7.483e-09

relate them also to the built-in MATLAB function `svds`. The matrices $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n\times m}$ are drawn randomly from a zero-mean normal distribution. To compute the error for $\mathbf{W}$, we use the metric

$$\text{Error} = \|\mathbf{W} - \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\|_F, \tag{4.2}$$

where the matrices $\mathbf{U}$, $\boldsymbol{\Sigma}$, and $\mathbf{V}$ are computed using Algorithm 3.1. To compute the error for $\widehat{\mathbf{W}}$, we use the metric

$$\text{Error} = \|\widehat{\mathbf{W}} - (\widehat{\mathbf{U}}_1\boldsymbol{\Sigma}\widehat{\mathbf{V}}_1^\top + \widehat{\mathbf{U}}_2\widehat{\mathbf{V}}_2^\top)\|_F \tag{4.3}$$

from the SVD of $\widehat{\mathbf{W}}$ in Theorem 3.3 (see (3.9)). Since Algorithm 3.2 does not compute $\widehat{\mathbf{Q}}_\perp$, we need a way of forming the matrix $\widehat{\mathbf{U}}_2\widehat{\mathbf{V}}_2^\top = \widehat{\mathbf{Q}}_\perp\widehat{\mathbf{Q}}_\perp^\top$, which is the projection

TABLE 3

Experiment II.A. *Comparison of the performance of our proposed RObSVD approach (Algorithm* 3.1*) to the built-in MATLAB function* `svds` *for computing an SVD of* $\mathbf{W}$*, on moderately sized matrices (*$m = 150$ *and* $800 \leq n \leq 5,000$*). For each* n*, the experiment was performed* 10 *times with different randomly generated matrices each time. The average computational times, average error values, and maximum error values (among the* 10 *experiments) are reported, where the error is measured using* (4.2)*. Note that our proposed approach uses significantly less time than* `svds`*.*

n	Time		Average error		Maximum error	
	RObSVD	svds	RObSVD	svds	RObSVD	svds
800	0.0079	0.3712	4.834e-09	3.769e-12	1.870e-08	1.370e-11
1000	0.0076	0.6582	4.612e-08	4.499e-12	1.873e-07	1.629e-11
2000	0.0086	3.4918	9.858e-08	6.898e-12	4.996e-07	1.599e-11
3000	0.0092	7.9335	1.564e-06	6.259e-11	1.218e-05	1.519e-10
4000	0.0098	14.1811	1.823e-07	2.383e-10	1.417e-06	1.590e-09
5000	0.0099	21.2731	9.427e-09	6.250e-11	2.288e-08	3.195e-10

TABLE 4

Experiment II.A. *Comparison of the performance of our proposed RObSVD approach (Algorithm* 3.2*) to the built-in MATLAB function* `svds` *for computing an SVD of* $\widehat{\mathbf{W}}$*, on moderately sized matrices (*$m = 150$ *and* $800 \leq n \leq 5,000$*). For each* n*, the experiment was performed 10 times with different randomly generated matrices each time. The average computational times, average error values, and maximum error values (among the* 10 *experiments) are reported, where the error is measured using* (4.2)*. Note that our proposed approach uses significantly less time than* `svds`*.*

n	Time		Average error		Maximum error	
	RObSVD	svds	RObSVD	svds	RObSVD	svds
800	0.0243	0.0587	4.837e-09	3.315e-12	1.871e-08	1.049e-11
1000	0.0289	0.1051	4.615e-08	1.103e-11	1.874e-07	6.119e-11
2000	0.0594	2.2020	9.861e-08	7.077e-12	4.998e-07	1.828e-11
3000	0.1060	9.0947	1.565e-06	8.491e-11	1.219e-05	6.012e-10
4000	0.1836	22.4527	1.824e-07	1.229e-11	1.418e-06	5.489e-11
5000	0.2556	44.7147	9.428e-09	8.417e-12	2.290e-08	2.200e-11

matrix onto the space orthogonal to the span of the columns of $\mathbf{X}$ and $\mathbf{Y}$. Thus, we define the matrix $\mathbf{G} = [\ \mathbf{X}\ \ \mathbf{Y}\ ] \in \mathbb{R}^{n \times 2m}$ and note that

$$\widehat{\mathbf{U}}_2 \widehat{\mathbf{V}}_2^\top = \widehat{\mathbf{Q}}_\perp \widehat{\mathbf{Q}}_\perp^\top = \mathbf{I} - \mathbf{G}(\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top.$$

Experiment II is divided into two subexperiments: Experiment II.A is for moderately sized matrices ($m = 150$ and $800 \leq n \leq 5,000$), while Experiment II.B is for large-scale matrices ($m = 150$ and $7,500 \leq n \leq 300,000$). For each n, we repeated the experiment 10 times with different randomly generated matrices each time and report the average and maximum errors. The results for Experiment II.A for $\mathbf{W}$ and $\widehat{\mathbf{W}}$ are reported in Tables 3 and 4, respectively. For Experiment II.B, since $\mathbf{W} \in \mathbb{R}^{n \times n}$ does not fit into MATLAB's memory when its dimension becomes large, we modify our metric for computing errors. In particular, a fixed random number of columns, r, is used in the comparisons of the form $\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top - \mathbf{W}\|_F$, instead of comparing all columns. Specifically, the errors are computed by the modified metric

$$\text{Error} = \|(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top - \mathbf{W})\mathbf{E}_r\|_F, \tag{4.4}$$

where the columns of the matrix $\mathbf{E}_r \in \mathbb{R}^{n \times r}$ are r randomly selected columns of $\mathbf{I}$, and where the matrices $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$ are computed using Algorithm 3.1 for $\mathbf{W}$ and Algorithm 3.2 for $\widehat{\mathbf{W}}$. The results of the experiments are reported in Table 5.

TABLE 5

Experiment II.B. *The performance of our proposed RObSVD approaches (Algorithms* 3.1 *and* 3.2*) for computing an SVD of* $\mathbf{W}$ *and* $\widehat{\mathbf{W}}$*, repectively, on large-scale problems (*$m = 150$ *and* $7,500 \leq n \leq 300,000$*). For each* n*, the experiment was performed* 10 *times with different randomly generated matrices each time. The average computational times and average and maximum error values using* (4.4) *are reported. The errors are computed by comparing* $r = 8$ *randomly selected columns of* $\mathbf{W}$ *and* $\widehat{\mathbf{W}}$*, respectively, with the factorizations from Algorithms* 3.1 *and* 3.2*. Note that the factorizations are obtained within one second and the average errors are relatively small.*

n	RObSVD($\mathbf{W}$)			RObSVD($\widehat{\mathbf{W}}$)		
	Time	Avg. Err.	Max. Err.	Time	Avg. Err.	Max. Err.
7500	0.0204	1.009e-07	8.013e-07	0.0350	9.750e-08	7.741e-07
10000	0.0228	2.195e-10	8.584e-10	0.0418	2.228e-10	8.825e-10
50000	0.0753	5.688e-10	2.130e-09	0.1562	5.711e-10	2.161e-09
100000	0.1395	5.637e-10	2.541e-09	0.2963	5.634e-10	2.543e-09
200000	0.2606	9.443e-10	3.320e-09	0.5756	9.471e-10	3.343e-09
300000	0.3764	2.625e-10	9.264e-10	0.8654	2.610e-10	9.211e-10

TABLE 6

Experiment III.A. *Comparison of the performance of our proposed RObSVD approach (Algorithm* 3.1*) to the built-in MATLAB function* `svds` *and the randomized SVD (r-SVD) for computing the SVD of* $\mathbf{W}$ *on moderately sized matrices (*$m = 20$ *and* $800 \leq n \leq 5,000$*). The test matrices are deterministically generated from columns of a Chebyshev Vandermonde-like matrix. The parameter for the r-SVD is set to* $p = m = 20$*. The* `svds` *function takes the longest computational times while all errors are all comparably small.*

n	Time (in sec.)			Error		
	RObSVD	svds	r-SVD	RObSVD	svds	r-SVD
800	0.0555	0.2732	0.0284	2.6322e-12	6.5832e-12	6.7252e-13
1000	0.0042	0.0453	0.0046	1.6636e-12	5.0556e-13	1.1908e-12
2000	0.0006	0.3521	0.0135	1.2446e-12	3.4958e-11	7.388e-13
3000	0.0007	0.8625	0.0216	2.5664e-12	8.4791e-12	6.771e-13
4000	0.0028	1.1939	0.0375	3.3174e-12	5.9945e-11	8.3006e-13
5000	0.0007	2.3692	0.0588	1.8405e-12	6.0833e-11	6.2313e-13

4.3. Experiment III. In this experiment we compare our proposed algorithms to the randomized SVD (r-SVD) from Halko, Martinsson, and Tropp [9] described in Appendix B. In particular we use an implementation available in the public domain [12]. The experiment is divided into two parts. In Experiment III.A we compare Algorithms 3.1 and 3.2 to the built-in MATLAB function `svds` and the r-SVD function `rsvd`. The problems in Experiment III.A are moderately sized ($m = 20$ and $800 \leq n \leq 5,000$). We use deterministic test matrices from Higham [10]. In particular we select the first m columns of a Chebyshev Vandermonde-like matrix defined by

$$\mathbf{A}_{ij} = \cos\left(\frac{(i-1)(j-1)\pi}{n}\right).$$

In MATLAB R2018b this matrix is generated via the command `A=gallery('orthog', n,-1)`. Subsequently, we define $\mathbf{X} = \mathbf{A}[\ \mathbf{e}_1\ \mathbf{e}_2\ \cdots\ \mathbf{e}_m\ ]$, where $\mathbf{e}_i$ are columns of the identity matrix. $\mathbf{Y}$ is defined as $\mathbf{DX}$, where the elements of the diagonal matrix $\mathbf{D}$ are the reciprocals of the squared row lengths of $\mathbf{X}$. The results of the factorizations of $\mathbf{W}$ and $\widehat{\mathbf{W}}$ for moderately sized problems are summarized in Tables 6 and 7, respectively.

In Experiment III.B we compare Algorithms 2 and 3 to the r-SVD on large-scale problems ($m = 20$ and $7,500 \leq n \leq 1,000,000$). For these problems, the built-in

TABLE 7

Experiment III.A. *Comparison of the performance of our proposed RObSVD approach (Algorithm* 3.2*) to the built-in MATLAB function* `svds` *and the randomized SVD (r-SVD) for computing the SVD of* $\widehat{\mathbf{W}}$ *on moderately sized matrices (*$m = 20$ *and* $800 \leq n \leq 5,000$*). The test matrices are deterministically generated from columns of a Chebyshev Vandermonde-like matrix. The parameter for the r-SVD is set to* $p = n - m$*. In this experiment the computational times of the r-SVD significantly increases because the parameter p becomes relatively large. In contrast, the computational times for our proposed method remain consistently low.*

n	Time (in sec.)			Error		
	RObSVD	svds	r-SVD	RObSVD	svds	r-SVD
800	0.0638	0.0549	0.4494	2.6320e-12	7.1624e-13	1.4599e-12
1000	0.0086	0.1215	0.7673	1.6668e-12	7.8639e-13	1.4513e-12
2000	0.0102	2.2008	8.8362	1.2580e-12	1.4958e-12	2.0955e-12
3000	0.0214	8.9106	33.4525	2.5619e-12	1.0375e-12	1.8820e-12
4000	0.0444	18.4744	83.9922	3.3205e-12	9.6857e-13	2.0762e-12
5000	0.0913	44.4121	165.4607	1.8421e-12	1.0600e-12	2.2228e-12

TABLE 8

Experiment III.B. *Comparison of the performance of our proposed RObSVD approach (Algorithm* 3.1*) to the randomized SVD (r-SVD) for computing the SVD of* $\mathbf{W}$ *on large-scale matrices (*$m = 20$ *and* $7,500 \leq n \leq 1,000,000$*). For each n, the experiment was performed* 10 *times with different randomly generated matrices each time. The errors are computed using* (4.4) *by comparing* $r = 8$ *randomly selected columns of* $\mathbf{W}$*. The parameter for the r-SVD is set to* $p = m = 20$*. The average and maximum errors for r-SVD are slightly lower than those for ObSVD. However, the computational times for r-SVD are significantly higher.*

n	Time (in sec.)		Average error		Maximum error	
	RObSVD	r-SVD	RObSVD	r-SVD	RObSVD	r-SVD
7500	0.0007	0.0093	2.966e-10	1.638e-13	2.359e-09	5.673e-13
10000	0.0006	0.0105	9.112e-12	2.461e-13	2.863e-11	6.565e-13
50000	0.0046	0.0770	1.054e-12	6.416e-14	3.535e-12	1.197e-13
100000	0.0108	0.1761	1.672e-12	1.557e-13	5.234e-12	2.616e-13
200000	0.0211	0.3694	8.609e-12	2.461e-13	3.700e-11	7.098e-13
300000	0.0320	0.5885	7.610e-10	4.227e-13	4.493e-09	1.482e-12
500000	0.0515	1.1112	6.238e-12	1.788e-13	2.168e-11	3.098e-13
1000000	0.0900	2.5642	7.095e-13	9.671e-14	2.210e-12	2.750e-13

MATLAB function cannot be used because of out-of-memory errors and excessive compute times, which is why it is omitted in the comparisons. The matrices $\mathbf{X}$ and $\mathbf{Y}$ are generated by drawing randomly from a zero-mean normal distribution. The experiments are repeated 10 times and average values are reported. The results of the factorizations of $\mathbf{W}$ and $\widehat{\mathbf{W}}$ for large problems are summarized in Tables 8 and 9, respectively.

4.4. Experiment IV. Stewart [14] and O'Leary [13] analyzed the spectral norm of the oblique projection in (1.3) and proposed the following bound:

$$\left\|\mathbf{X}(\mathbf{X}^\top \mathbf{D}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{D}\right\|_2 \leq \rho^{-1} \equiv \Big(\min_I \inf\nolimits_+ (\mathbf{U}_I)\Big)^{-1}, \tag{4.5}$$

where $\inf_+ (\mathbf{U}_I)$ symbolizes the smallest nonzero singular value of any submatrix $\mathbf{U}_I$ of an orthonormal basis $\mathbf{U} \in \mathbb{R}^{n\times m}$ to $\mathbf{X}$. In this experiment, we compute the largest singular value σ_1 obtained from Algorithm 3.1 and compare it to the bound in (4.5). Obtaining ρ explicitly is computationally intensive because the SVD for all possible submatrices $\mathbf{U}_I$ are computed. For this reason the size n of the matrices used in

TABLE 9

Experiment III.B. *Comparison of the performance of our proposed ROhSVD approach (Algorithm* 3.2*) to the randomized SVD (r-SVD) for computing the SVD of* $\widehat{\mathbf{W}}$*, on large-scale matrices (*$m = 20$ *and* $7,500 \leq n \leq 1,000,000$*). For each* n*, the experiment was performed* 10 *times with different randomly generated matrices each time. The errors are computed using* (4.4) *by comparing* $r = 8$ *randomly selected columns of* $\widehat{\mathbf{W}}$*. The parameter for the r-SVD is set to* $p = m = 20$*. The computational times for r-SVD are slightly higher than those for our proposed method. However, the average and maximum errors for r-SVD are significantly higher because* $p \ll n - m$*.*

n	Time (in sec.)		Average error		Maximum error	
	RObSVD	r-SVD	RObSVD	r-SVD	RObSVD	r-SVD
7500	0.0021	0.0119	3.032e-10	3.406e+00	2.412e-09	3.473e+00
10000	0.0021	0.0134	9.169e-12	3.460e+00	2.900e-11	3.570e+00
50000	0.0172	0.0937	1.052e-12	3.453e+00	3.512e-12	3.605e+00
100000	0.0390	0.2230	1.675e-12	3.529e+00	5.244e-12	3.716e+00
200000	0.0756	0.4593	8.603e-12	3.441e+00	3.699e-11	3.536e+00
300000	0.1171	0.7279	7.614e-10	3.454e+00	4.497e-09	3.581e+00
500000	0.1943	1.4901	6.236e-12	3.462e+00	2.167e-11	3.556e+00
1000000	0.5561	3.1999	7.105e-13	3.432e+00	2.211e-12	3.575e+00

TABLE 10

Experiment IV. *Comparison of the bound* ρ *from* (4.5) *with the spectral norm* σ_1 *from Algorithm* 3.1 *and the built-in* `svd` *MATLAB function (*$\widehat{\sigma}_1$*). For each pair* (n, m)*, the experiments are repeated* 10 *times, and one representative result is reported. We note that the bound* ρ *is generally not tight.*

n	m	ρ	σ_1	$\widehat{\sigma}_1$	$\lvert\sigma_1 - \widehat{\sigma}_1\rvert$
6	2	5.8085e+01	1.1265e+00	1.1265e+00	2.2204e-16
7	2	1.7034e+02	1.1395e+00	1.1395e+00	2.2204e-16
8	2	6.9771e+01	1.4313e+00	1.4313e+00	4.4409e-16
9	3	5.9314e+03	1.1560e+00	1.1560e+00	0.0000e+00
10	3	3.0740e+02	1.6386e+00	1.6386e+00	0.0000e+00
11	3	8.7337e+03	1.9385e+00	1.9385e+00	1.1102e-15
12	4	1.5304e+03	1.5533e+00	1.5533e+00	2.2204e-16
20	6	1.6388e+06	1.3767e+00	1.3767e+00	0.0000e+00

the comparisons is limited to small dimensions. However, note that computing the spectral norm for large dimensions of n can be practically done with Algorithm 3.1 (see Experiment I). The results of the experiment are reported in Table 10.

4.5. Experiment V. In this experiment we compute an eigendecomposition and an SVD of block BFGS quasi-Newton matrices found in [3] and described in Appendix C. The data matrices $\mathbf{X}$ and $\mathbf{Y}$ correspond to the matrices $\mathbf{S}_k$ and $\mathbf{Y}_k$, respectively, in (2.1) and the scalar γ corresponds to γ_k in (3.12) in [3]. In this experiment, $\mathbf{X}, \mathbf{Y}$, and γ in the definition of $\mathbf{H}_+$ from (C.1) are generated by the optimization algorithm LTRSC-LEC (cf. Brust, Marcia, and Petra [2]), which is available in the public domain: https://github.com/johannesbrust/LTR_LEC. The optimization algorithm is applied to 53 large-scale problems from the CUTEst problem collection of Gould, Orban, and Toint [8] and stopped after $k = 10$ iterations to store the matrices $\mathbf{X} = \mathbf{S}_{10}$ and $\mathbf{Y} = \mathbf{Y}_{10}$ and the scalar $\gamma = \gamma_{10}$. In order to compute the products $\widehat{\mathbf{Q}}_\perp \widehat{\mathbf{Q}}_\perp^\top$ we use the same approach as in section 4.2. The results of the experiments are reported in Table 11. The error is computed like in (4.3) using $\widehat{\mathbf{Q}}_\perp \widehat{\mathbf{Q}}_\perp^\top$, and the pairs (n, m) vary for each problem between $50 \leq n \leq 10,000$ and $1 \leq m \leq 5$.

TABLE 11

Experiment V *(continued). Computational time and error for computing the eigendecomposition and SVD of* $\mathbf{H}_+$ *from* 53 *problems of the CUTEst optimization problem collection using our proposed methods, Algorithms* C.1 *and* C.2*. Note that our proposed approaches compute the factorizations quickly and accurately.*

Problem	n	m	Eigendecomposition		SVD	
			Time	Error	Time	Error
ARWHEAD	5000	1	0.0444	2.7763e-13	0.0470	5.5634e-13
BDQRTIC	5000	1	0.0498	1.9232e-12	0.0510	1.7542e-12
BOX	10000	1	0.0233	1.2860e-09	0.0095	1.0288e-09
BROYDN7D	5000	4	0.0477	4.1295e-11	0.0398	4.1270e-11
BRYBND	5000	4	0.0523	2.0564e-10	0.0638	2.0463e-10
CRAGGLVY	5000	4	0.0510	9.4554e-10	0.0602	9.3106e-10
CURLY10	10000	4	0.0111	8.0504e-11	0.0114	4.7789e-11
CURLY20	10000	5	0.0065	2.9187e-08	0.0011	1.0684e-08
CURLY30	10000	4	0.0039	8.3556e-08	0.0008	6.8244e-08
DIXMAANE	3000	4	0.0224	1.1637e-13	0.0252	4.1552e-14
DIXMAANF	3000	4	0.0246	5.3705e-13	0.0221	3.9153e-13
DIXMAANG	3000	5	0.0200	1.3656e-11	0.0214	1.3645e-11
DIXMAANH	3000	5	0.0201	9.5251e-12	0.0213	9.5145e-12
DIXMAANI	3000	4	0.0197	1.7801e-13	0.0218	1.5156e-13
DIXMAANJ	3000	4	0.0199	4.1461e-12	0.0209	4.2514e-12
DIXMAANK	3000	4	0.0191	1.9109e-10	0.0204	1.9144e-10
DIXMAANL	3000	5	0.0203	2.1932e-11	0.0208	2.1928e-11
DIXON3DQ	10000	4	0.0097	0.0000e-00	0.0078	0.0000e-00
DQDRTIC	5000	2	0.0446	4.3788e-12	0.0455	5.2888e-13
DQRTIC	5000	4	0.0485	3.7456e-05	0.0464	1.4585e-05
EDENSCH	2000	4	0.0200	5.2502e-11	0.0206	5.2442e-11
EIGENALS	2550	4	0.0196	8.1763e-10	0.0199	5.9582e-11
EIGENBLS	2550	4	0.0156	4.0069e-12	0.0176	4.1260e-12
ENGVAL1	5000	3	0.0444	2.2560e-11	0.0492	2.2534e-11
EXTROSNB	1000	3	0.0147	3.4905e-12	0.0117	3.3191e-12
FLETCHCR	1000	2	0.0012	1.4978e-10	0.0015	4.4783e-12
FMINSRF2	5625	5	0.0037	1.7530e-13	0.0023	2.4287e-14
FREUROTH	5000	3	0.0388	3.4666e-08	0.0410	3.4687e-08
GENHUMPS	5000	4	0.0486	6.4212e-11	0.0568	4.5296e-11
JIMACK	3549	5	0.0395	1.2783e-09	0.0438	1.2778e-09
LIARWHD	5000	1	0.0644	1.1383e-09	0.0623	6.4407e-11
MSQRTALS	1024	4	0.0168	3.0400e-11	0.0164	2.3044e-11
MSQRTBLS	1024	4	0.0032	1.4819e-11	0.0017	1.4756e-11
NCB20	5010	5	0.0034	4.2942e-12	0.0023	4.0284e-12
NCB20B	5000	4	0.0390	3.0889e-10	0.0377	2.0352e-10
NONCVXU2	5000	4	0.0484	8.6949e-13	0.0532	8.4630e-13
NONCVXUN	5000	5	0.0495	1.3930e-11	0.0611	1.3625e-11
NONDIA	5000	1	0.0503	1.7467e-06	0.0575	1.1639e-06
NONDQUAR	5000	2	0.0506	9.9618e-14	0.0557	7.8464e-14
POWELLSG	5000	2	0.0477	5.4592e-13	0.0551	2.1225e-13
POWER	10000	3	0.0091	3.4627e-01	0.0104	3.8937e-03
QUARTC	5000	4	0.0408	3.7456e-05	0.0500	1.4585e-05
SCHMVETT	5000	4	0.0497	2.0139e-12	0.0594	1.9257e-12
SINQUAD	5000	1	0.0492	4.9521e-11	0.0567	9.0686e-11
SPARSINE	5000	5	0.0493	7.9502e-07	0.0580	3.9332e-07
SPARSQUR	10000	4	0.0396	2.0341e-11	0.0105	1.9669e-11
SPMSRTLS	4999	5	0.1937	2.5457e-10	0.0527	2.5444e-10
SROSENBR	5000	1	0.0473	8.3040e-12	0.0617	8.1919e-12
TESTQUAD	5000	4	0.0499	1.7930e-04	0.0591	1.6347e-07
TQUARTIC	5000	1	0.0470	9.7513e-12	0.0554	7.7932e-12
TRIDIA	5000	5	0.0482	4.0554e-07	0.0596	1.5046e-07
VAREIGVL	50	5	0.0078	4.4573e-11	0.0089	4.4563e-11
WOODS	4000	2	0.0387	3.0907e-09	0.0426	2.3302e-09

5. Conclusions. In this article we develop eigenvalue and SVDs of oblique projection matrices. Algorithms for efficiently computing these factorizations are proposed, and their implementations are tested and compared on moderately and large-scale matrices. The proposed methods compare well with both standard deterministic factorizations as well as with randomized methods. We apply the factorizations to decompose block quasi-Newton matrices arising in 53 large-scale problems from a benchmarking collection of optimization problems. The numerical experiments show that our proposed approaches are fast, efficient, and accurate.

Appendix A. The approach of Behrens and Scharf. In [1], Behrens and Scharf proposed an approach for computing the singular values of $\mathbf{W} = \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top$ for a specific choice of $\mathbf{Y}$, namely, $\mathbf{Y} = \mathbf{P}^\top\mathbf{X}$:

$$\mathbf{W} = \mathbf{X}(\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{P}, \tag{A.1}$$

where $\mathbf{P}$ is matrix that projects onto the orthogonal complement of Range($\mathbf{Z}$), and $\mathbf{Z} \in \mathbb{R}^{n\times\ell}$ has full-column rank. In particular, if $\tilde{\mathbf{Q}} \in \mathbb{R}^{n\times\ell}$ is a matrix whose columns form an orthonormal basis for Range($\mathbf{Z}$), then $\mathbf{P} = \mathbf{I} - \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top = \mathbf{I} - \tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^\top$. Behrens and Scharf use the principal angles between $\mathbf{Q}_\parallel$ (from the "thin" QR factorization $\mathbf{X} = \mathbf{Q}_\parallel\mathbf{R}_\parallel$) and $\tilde{\mathbf{Q}}$. Specifically, if $\mathbf{Q}_\parallel^\top\tilde{\mathbf{Q}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^\top$ is the SVD of $\mathbf{Q}_\parallel^\top\tilde{\mathbf{Q}}$ and $\tilde{\sigma}_i$ is the i^{th} singular value in $\tilde{\mathbf{\Sigma}}$, then the principal angles, α_i, are defined by $\cos(\alpha_i) = \tilde{\sigma}_i$ (see [7]). Observe that $\mathbf{W}$ from (A.1) satisfies

$$\begin{aligned}
\mathbf{W}\mathbf{W}^\top &= \mathbf{X}(\mathbf{X}^\top\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^\top \\
&= \mathbf{Q}_\parallel(\mathbf{Q}_\parallel^\top\mathbf{P}\mathbf{Q}_\parallel)^{-1}\mathbf{Q}_\parallel^\top \\
&= \mathbf{Q}_\parallel(\tilde{\mathbf{U}}(\mathbf{I} - \tilde{\mathbf{\Sigma}}^2)\tilde{\mathbf{U}}^\top)^{-1}\mathbf{Q}_\parallel^\top \\
&= \mathbf{Q}_\parallel\tilde{\mathbf{U}}(\mathbf{I} - \tilde{\mathbf{\Sigma}}^2)^{-1}\tilde{\mathbf{U}}^\top\mathbf{Q}_\parallel^\top,
\end{aligned}$$

so that the singular values of $\mathbf{W}$ are given by

$$\sigma_i = \sqrt{(1-\tilde{\sigma}_i^2)^{-1}} = \sqrt{\frac{1}{1-\cos^2(\alpha_i)}} = \frac{1}{\sin(\alpha_i)}.$$

Algorithm A.1 describes the approach of Behrens and Scharf for computing the singular values of $\mathbf{W}$.

Algorithm A.1 Singular Values of $\mathbf{W}$ (Behrens and Scharf).

1: Input: $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{Z} \in \mathbb{R}^{n\times(n-m)}$;
2: $\mathbf{R}_\parallel^\top\mathbf{R}_\parallel = \mathbf{X}^\top\mathbf{X}$; {Cholesky factorization}
3: $\tilde{\mathbf{R}}^\top\tilde{\mathbf{R}} = \mathbf{Z}^\top\mathbf{Z}$; {Cholesky factorization}
4: $\mathbf{Q}_\parallel = \mathbf{X}\mathbf{R}_\parallel^{-1}$;
5: $\tilde{\mathbf{Q}} = \mathbf{Z}\tilde{\mathbf{R}}^{-1}$;
6: $\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^\top = \mathbf{Q}_\parallel^\top$; {SVD}
7: $\mathbf{\Sigma}^2 = (\mathbf{I} - \tilde{\mathbf{\Sigma}}^2)^{-1}$;

Note that Algorithm A.1 requires computing both $\mathbf{Q}_\parallel$ and $\tilde{\mathbf{Q}}$. Because forming $\tilde{\mathbf{Q}}$ can be computationally expensive when the nullspace of $\mathbf{W}$ in (A.1) is large, we

propose a modification of the Behrens and Scharf approach, which does not require the computation of $\tilde{\mathbf{Q}}$. Specifically, consider

$$\begin{aligned}\mathbf{R}_{\|}^{-\top}\mathbf{X}^{\top}\mathbf{Y}\mathbf{R}_{\|}^{-1} &= \mathbf{Q}_{\|}^{\top}\mathbf{Y}\mathbf{R}_{\|}^{-1}\\ &= \mathbf{Q}_{\|}^{\top}\mathbf{P}^{\top}\mathbf{X}\mathbf{R}_{\|}^{-1}\\ &= \mathbf{Q}_{\|}^{\top}(\mathbf{I}-\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^{\top})\mathbf{X}\mathbf{R}_{\|}^{-1}\\ &= \mathbf{I}-\mathbf{Q}_{\|}^{\top}\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^{\top}\mathbf{Q}_{\|}\\ &= \tilde{\mathbf{U}}(\mathbf{I}-\tilde{\mathbf{\Sigma}}^{2})\tilde{\mathbf{U}}^{\top}.\end{aligned}$$

Taking the inverse of both sides yields

$$\mathbf{R}_{\|}(\mathbf{X}^{\top}\mathbf{Y})^{-1}\mathbf{R}_{\|}^{\top} = \tilde{\mathbf{U}}(\mathbf{I}-\tilde{\mathbf{\Sigma}}^{2})^{-1}\tilde{\mathbf{U}}^{\top} = \tilde{\mathbf{U}}\mathbf{\Sigma}^{2}\tilde{\mathbf{U}}^{\top},$$

and therefore, the singular values of $\mathbf{W}$ are efficiently computed from an eigendecomposition of the small $m \times m$ matrix $\mathbf{R}_{\|}(\mathbf{X}^{\top}\mathbf{Y})^{-1}\mathbf{R}_{\|}^{\top}$. Algorithm A.2 describes this modifed approach of Behrens and Scharf for computing the singular values of $\mathbf{W}$.

Algorithm A.2 Singular Values of $\mathbf{W}$ (Modified Behrens and Scharf).

1: Input: $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{Y} = (\mathbf{I}-\mathbf{Z}(\mathbf{Z}^{\top}\mathbf{Z})^{-1}\mathbf{Z}^{\top})\mathbf{X} \in \mathbb{R}^{n\times m}$;
2: $\mathbf{R}_{\|}^{\top}\mathbf{R}_{\|} = \mathbf{X}^{\top}\mathbf{X}$ {Cholesky factorization}
3: $\tilde{\mathbf{U}}\mathbf{\Sigma}^{2}\tilde{\mathbf{U}}^{\top} = \mathbf{R}_{\|}(\mathbf{X}^{\top}\mathbf{Y})^{-1}\mathbf{R}_{\|}^{\top}$ {Eigendecomposition}

Appendix B. Randomized SVD. An alternative way to approximately factor large matrices is the randomized SVD (r-SVD) from, among others, Halko, Martinsson, and Tropp [9]. These probabilistic factorizations are based on two stages. In the first stage an approximate orthonormal basis to the range of the target matrix, say, $\mathbf{W} \in \mathbb{R}^{n\times n}$, is computed. This is achieved through a technique called sampling (sketching). The sampling technique is based on a "small" random (possibly normally distributed) matrix, $\mathbf{\Omega} \in \mathbb{R}^{n\times p}$, with $p \ll n$. First, a matrix $\mathbf{B} \in \mathbb{R}^{n\times p}$, whose columns lie in the range of $\mathbf{W}$, is formed as

$$\mathbf{B} = \mathbf{W}\mathbf{\Omega}.$$

Second, an approximate orthonormal basis of the range of $\mathbf{W}$ is constructed via, e.g., a QR factorization of $\mathbf{B}$:

$$\mathbf{B} = \mathbf{Q}\mathbf{R},$$

where $\mathbf{Q} \in \mathbb{R}^{n\times p}$ has orthonormal columns and $\mathbf{R} \in \mathbb{R}^{p\times p}$ is upper triangular. Then an SVD of the $p \times n$ matrix $\mathbf{Q}^{\top}\mathbf{W}$ is computed:

$$\mathbf{Q}^{\top}\mathbf{W} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^{\top},$$

where $\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}} \in \mathbb{R}^{p\times p}$ and $\tilde{\mathbf{V}} \in \mathbb{R}^{n\times p}$. An approximate randomized SVD of $\mathbf{W}$ is computed as

$$\mathbf{W} \approx \mathbf{Q}\mathbf{Q}^{\top}\mathbf{W} = (\mathbf{Q}\tilde{\mathbf{U}})\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^{\top}.$$

Appendix C. Block BFGS matrices. For optimization problems or systems of nonlinear equations, quasi-Newton matrices are effective ways to approximate derivatives of multivariate functions [5]. The most widely used quasi-Newton matrix is

the BFGS matrix. In [3], Byrd, Nocedal, and Schnabel propose recursive block-BFGS formulas of the form

$$\mathbf{H}_+ = (\mathbf{I} - \mathbf{X}(\mathbf{Y}^\top\mathbf{X})^{-1}\mathbf{Y}^\top)\mathbf{H}(\mathbf{I} - \mathbf{Y}(\mathbf{X}^\top\mathbf{Y})^{-1}\mathbf{X}^\top) + \mathbf{X}(\mathbf{X}^\top\mathbf{Y})^{-1}\mathbf{X}^\top, \tag{C.1}$$

where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times m}$. The matrix $\mathbf{H}$ is often defined as $\mathbf{H} = \gamma\mathbf{I}$. Using our previous notation for $\mathbf{W}$ in (1.4) and $\widehat{\mathbf{W}}$ in (2.1), the block BFGS formula takes the form

$$\begin{aligned}\mathbf{H}_+ &= \widehat{\mathbf{W}}\mathbf{H}\widehat{\mathbf{W}}^\top + \mathbf{W}\mathbf{X}(\mathbf{X}^\top\mathbf{Y})^{-1}\mathbf{X}^\top\mathbf{W}^\top \\ &= \gamma\widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}^2\widehat{\mathbf{U}}^\top + \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{Y})^{-1}\mathbf{X}^\top\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top \\ &\equiv \gamma\widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}^2\widehat{\mathbf{U}}^\top + \mathbf{U}\mathbf{\Sigma}\mathbf{M}\mathbf{\Sigma}\mathbf{U}^\top, \end{aligned} \tag{C.2}$$

where we used the factorizations of $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and $\widehat{\mathbf{W}} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top$ from Theorems 3.2 and 3.3, and where

$$\mathbf{M} \equiv \mathbf{V}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{Y})^{-1}\mathbf{X}^\top\mathbf{V}, \tag{C.3}$$

and $\mathbf{M} \in \mathbb{R}^{m\times m}$. We present an eigendecomposition of $\mathbf{H}_+$ in the following theorem.

THEOREM C.1. *Consider an eigendecomposition of the $2m \times 2m$ matrix*

$$\begin{bmatrix} \gamma\mathbf{\Sigma}^2 + \mathbf{\Sigma}\mathbf{M}\mathbf{\Sigma} & \mathbf{\Sigma}\mathbf{M} \\ -\gamma\mathbf{\Sigma} & \mathbf{0} \end{bmatrix} = \mathbf{C}\mathbf{\Lambda}\mathbf{C}^{-1},$$

where $\mathbf{M}$ *is from* (C.3). *Let* $\mathbf{\Psi} \equiv [\ \mathbf{U}\quad \mathbf{V}\ ]$, *and let*

$$\mathbf{S}_1 \equiv \mathbf{\Psi}\mathbf{C} \quad \textit{and} \quad \widehat{\mathbf{S}}_1^\top \equiv \mathbf{C}^{-1}(\mathbf{\Psi}^\top\mathbf{\Psi})^{-1}\mathbf{\Psi}^\top.$$

Then the eigendecomposition of $\mathbf{H}_+$ *from* (C.2) *is*

$$\mathbf{H}_+ = \begin{bmatrix} \mathbf{S}_1 & \widehat{\mathbf{Q}}_\perp \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda} & \\ & \gamma\mathbf{I}_{n-2m} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{S}}_1^\top \\ \widehat{\mathbf{Q}}_\perp^\top \end{bmatrix}, \tag{C.4}$$

where $\widehat{\mathbf{Q}}_\perp \in \mathbb{R}^{n\times(n-2m)}$ *is from Theorem* 3.3, *i.e.,* $\widehat{\mathbf{Q}}_\perp$ *is a matrix whose columns form an orthonormal basis for the orthogonal complement of the space spanned by the columns of* $\mathbf{U}$ *and* $\mathbf{V}$.

Proof. By the definition of $\widehat{\mathbf{Q}}_\perp$, we have $\mathbf{U}^\top\widehat{\mathbf{Q}}_\perp = \mathbf{0}$. In addition, from Theorem 3.3, it holds that

$$\widehat{\mathbf{U}}^\top\widehat{\mathbf{Q}}_\perp = \begin{bmatrix} \widehat{\mathbf{U}}_1^\top \\ \widehat{\mathbf{U}}_2^\top \end{bmatrix} \widehat{\mathbf{Q}}_\perp = \begin{bmatrix} \mathbf{D}^{-1}(\mathbf{\Sigma}^{-1}\mathbf{V}^\top - \mathbf{U}^\top) \\ \widehat{\mathbf{Q}}_\perp^\top \end{bmatrix} \widehat{\mathbf{Q}}_\perp = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}.$$

Thus

$$\mathbf{H}_+\widehat{\mathbf{Q}}_\perp = (\gamma\widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}^2\widehat{\mathbf{U}}^\top + \mathbf{U}\mathbf{\Sigma}\mathbf{M}\mathbf{\Sigma}\mathbf{U}^\top)\widehat{\mathbf{Q}}_\perp = \gamma\widehat{\mathbf{U}} \begin{bmatrix} \mathbf{\Sigma}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} = \gamma\widehat{\mathbf{Q}}_\perp. \tag{C.5}$$

Therefore the columns of $\widehat{\mathbf{Q}}_\perp$ form an orthonormal basis for eigenvectors associated with the eigenvalue γ, which has multiplicity $(n - 2m)$.

Recalling from (3.10a) that $\widehat{\mathbf{U}}_1 = (\mathbf{V}\boldsymbol{\Sigma}^{-1} - \mathbf{U})\mathbf{D}^{-1}$, where $\mathbf{D}^2 = \mathbf{I} - \boldsymbol{\Sigma}^{-2}$, and that $\mathbf{U}^\top\mathbf{V} = \boldsymbol{\Sigma}^{-1}$ from (3.8), we observe that

$$\begin{aligned}\widehat{\mathbf{U}}^\top\boldsymbol{\Psi} &= \begin{bmatrix} \widehat{\mathbf{U}}_1^\top \\ \widehat{\mathbf{U}}_2^\top \end{bmatrix} [\ \mathbf{U} \quad \mathbf{V}\ ] \\ &= \begin{bmatrix} \mathbf{D}^{-1}(\boldsymbol{\Sigma}^{-1}\mathbf{V}^\top\mathbf{U} - \mathbf{I}) & \mathbf{D}^{-1}(\boldsymbol{\Sigma}^{-1} - \mathbf{U}^\top\mathbf{V}) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.\end{aligned}$$

Next, we note that

$$\begin{aligned}\mathbf{H}_+\boldsymbol{\Psi} &= \{\gamma\widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}}^2\widehat{\mathbf{U}}^\top + \mathbf{U}\boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma}\mathbf{U}^\top\}\boldsymbol{\Psi} \\ &= \gamma[\ \widehat{\mathbf{U}}_1 \quad \widehat{\mathbf{U}}_2\ ] \begin{bmatrix} \boldsymbol{\Sigma}^2 & \\ & \mathbf{I} \end{bmatrix} \begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \mathbf{U}\boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma}\mathbf{U}^\top[\ \mathbf{U} \quad \mathbf{V}\ ] \\ &= -\gamma\widehat{\mathbf{U}}_1\mathbf{D}[\ \boldsymbol{\Sigma}^2 \quad \mathbf{0}\ ] + \mathbf{U}\boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma}[\ \mathbf{I} \quad \boldsymbol{\Sigma}^{-1}\ ] \\ &= [\ \mathbf{U} \quad \mathbf{V}\ ] \left\{ \gamma \begin{bmatrix} \mathbf{I} \\ -\boldsymbol{\Sigma}^{-1} \end{bmatrix} [\ \boldsymbol{\Sigma}^2 \quad \mathbf{0}\ ] + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma}[\ \mathbf{I} \quad \boldsymbol{\Sigma}^{-1}\ ] \right\} \\ &= \boldsymbol{\Psi} \begin{bmatrix} \gamma\boldsymbol{\Sigma}^2 + \boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma} & \boldsymbol{\Sigma}\mathbf{M} \\ -\gamma\boldsymbol{\Sigma} & \mathbf{0} \end{bmatrix}. \qquad \text{(C.6)}\end{aligned}$$

Subsequently, we obtain

$$\mathbf{H}_+\mathbf{S} = \mathbf{H}_+\boldsymbol{\Psi}\mathbf{C} = \boldsymbol{\Psi} \begin{bmatrix} \gamma\boldsymbol{\Sigma}^2 + \boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma} & \boldsymbol{\Sigma}\mathbf{M} \\ -\gamma\boldsymbol{\Sigma} & \mathbf{0} \end{bmatrix} \mathbf{C} = \boldsymbol{\Psi}\mathbf{C}\boldsymbol{\Lambda} = \mathbf{S}\boldsymbol{\Lambda}.$$

Therefore $\mathbf{S}_1$ contains eigenvectors associated to the eigenvalues in $\boldsymbol{\Lambda}$. Finally, note that

$$\begin{bmatrix} \widehat{\mathbf{S}}_1^\top \\ \widehat{\mathbf{Q}}_\perp^\top \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \widehat{\mathbf{Q}}_\perp \end{bmatrix} = \begin{bmatrix} \mathbf{C}^{-1}(\boldsymbol{\Psi}^\top\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\mathbf{C} & \mathbf{C}^{-1}(\boldsymbol{\Psi}^\top\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^\top\widehat{\mathbf{Q}}_\perp \\ \widehat{\mathbf{Q}}_\perp^\top\boldsymbol{\Psi}\mathbf{C} & \widehat{\mathbf{Q}}_\perp^\top\widehat{\mathbf{Q}}_\perp \end{bmatrix} = \mathbf{I}$$

since $\widehat{\mathbf{Q}}_\perp^\top\boldsymbol{\Psi} = \widehat{\mathbf{Q}}_\perp^\top[\ \mathbf{U} \quad \mathbf{V}\ ] = \mathbf{0}$, which completes the proof. ∎

Next, we present an SVD of $\mathbf{H}_+$ in the following theorem.

THEOREM C.2. *Let* $\boldsymbol{\Psi} \equiv [\ \mathbf{U} \quad \mathbf{V}\ ]$*, and consider the Cholesky factorization* $\boldsymbol{\Psi}^\top\boldsymbol{\Psi} = \mathbf{R}^\top\mathbf{R}$*. Furthermore, consider the SVD*

$$\mathbf{R} \begin{bmatrix} \gamma\boldsymbol{\Sigma}^2 + \boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\Sigma} & \boldsymbol{\Sigma}\mathbf{M} \\ -\gamma\boldsymbol{\Sigma} & \mathbf{0} \end{bmatrix} \mathbf{R}^{-1} = \widetilde{\mathbf{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\mathbf{V}}^\top,$$

and let $(\mathbf{U}_+)_1 = \boldsymbol{\Psi}\mathbf{R}^{-1}\widetilde{\mathbf{U}} \in \mathbb{R}^{n\times 2m}$ *and* $(\mathbf{V}_+)_1 = \boldsymbol{\Psi}\mathbf{R}^{-1}\widetilde{\mathbf{V}} \in \mathbb{R}^{n\times 2m}$*. An SVD of* $\mathbf{H}_+$ *in* (C.2) *is given as*

$$\mathbf{H}_+ = \mathbf{U}_+\boldsymbol{\Sigma}_+\mathbf{V}_+^\top = \begin{bmatrix} (\mathbf{U}_+)_1 & \widehat{\mathbf{Q}}_\perp \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\Sigma}} & \\ & \gamma\mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{V}_+)_1^\top \\ \widehat{\mathbf{Q}}_\perp^\top \end{bmatrix}, \qquad \text{(C.7)}$$

where $\widehat{\mathbf{Q}}_\perp \in \mathbb{R}^{n\times(n-2m)}$ *is from Theorem* 3.3*, i.e.,* $\widehat{\mathbf{Q}}_\perp$ *is a matrix whose columns form an orthonormal basis for the orthogonal complement of the space spanned by the columns of* $\mathbf{U}$ *and* $\mathbf{V}$*.*

Proof. It can easily be verified that $\mathbf{U}_+^\top \mathbf{U}_+ = \mathbf{I}$ and $\mathbf{V}_+^\top \mathbf{V}_+ = \mathbf{I}$ from how $(\mathbf{U}_+)_1, (\mathbf{V}_+)_1$, and $\widehat{\mathbf{Q}}_\perp$ are defined. Next we demonstrate that $\mathbf{U}_+^\top \mathbf{H}_+ \mathbf{V}_+ = \mathbf{\Sigma}_+$. First, using (C.6), we have that

$$\begin{aligned}
(\mathbf{U}_+)_1^\top \mathbf{H}_+ (\mathbf{V}_+)_1 &= (\mathbf{U}_+)_1^\top \mathbf{H}_+ \mathbf{\Psi} \mathbf{R}^{-1} \widetilde{\mathbf{V}} \\
&= (\mathbf{U}_+)_1^\top \mathbf{\Psi} \begin{bmatrix} \gamma \mathbf{\Sigma}^2 + \mathbf{\Sigma} \mathbf{M} \mathbf{\Sigma} & \mathbf{\Sigma} \mathbf{M} \\ -\gamma \mathbf{\Sigma} & \mathbf{0} \end{bmatrix} \mathbf{R}^{-1} \widetilde{\mathbf{V}} \\
&= (\mathbf{U}_+)_1^\top \mathbf{\Psi} \mathbf{R}^{-1} \mathbf{R} \begin{bmatrix} \gamma \mathbf{\Sigma}^2 + \mathbf{\Sigma} \mathbf{M} \mathbf{\Sigma} & \mathbf{\Sigma} \mathbf{M} \\ -\gamma \mathbf{\Sigma} & \mathbf{0} \end{bmatrix} \mathbf{R}^{-1} \widetilde{\mathbf{V}} \\
&= (\mathbf{\Psi} \mathbf{R}^{-1} \widetilde{\mathbf{U}})^\top \mathbf{\Psi} \mathbf{R}^{-1} \left(\widetilde{\mathbf{U}} \widetilde{\mathbf{\Sigma}} \widetilde{\mathbf{V}}^\top \right) \widetilde{\mathbf{V}} \\
&= \widetilde{\mathbf{\Sigma}}.
\end{aligned}$$

Next, note that from (C.5), $\widehat{\mathbf{Q}}_\perp^\top \mathbf{H}_+ \widehat{\mathbf{Q}}_\perp = \gamma \mathbf{I}_{n-2m}$. Finally, since $\mathbf{H}_+ \widehat{\mathbf{Q}}_\perp = \gamma \widehat{\mathbf{Q}}_\perp$ from (C.5) and $\mathbf{\Psi}^\top \widehat{\mathbf{Q}}_\perp = \mathbf{0}$ by definition of $\widehat{\mathbf{Q}}_\perp$, we have that $(\mathbf{U}_+)_1^\top \mathbf{H}_+ \widehat{\mathbf{Q}}_\perp = \gamma \widetilde{\mathbf{U}}^\top \mathbf{R}^{-\top} \mathbf{\Psi}^\top \widehat{\mathbf{Q}}_\perp = \mathbf{0}$. Similarly, $\widehat{\mathbf{Q}}_\perp^\top \mathbf{H}_+ (\mathbf{V}_+)_1 = \gamma \widehat{\mathbf{Q}}_\perp^\top \mathbf{\Psi} \mathbf{R}^{-1} \widetilde{\mathbf{V}} = \mathbf{0}$. Therefore, we conclude that

$$\mathbf{U}_+^\top \mathbf{H}_+ \mathbf{V}_+ = \begin{bmatrix} \widetilde{\mathbf{\Sigma}} & \\ & \gamma \mathbf{I}_{n-2m} \end{bmatrix},$$

which is equivalent to (C.7) . ∎

We present two practical algorithms to compute an eigendecomposition and an SVD of block BFGS matrices, corresponding to the eigenvalues and singular values other than γ.

Algorithm C.1 Eigendecomposition of $\mathbf{H}_+$.

Input: $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\mathbf{Y} \in \mathbb{R}^{n \times m}$, $\gamma \in \mathbb{R}$;
Compute $\mathbf{U}, \mathbf{\Sigma}$, and $\mathbf{V}$ using Algorithm 3.1 and form $\mathbf{\Psi} = [\ \mathbf{U} \quad \mathbf{V}\ ]$;
$\mathbf{M} = \mathbf{V}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{Y})^{-1} \mathbf{X}^\top \mathbf{V}$; {Intermediate matrix}
$\mathbf{C} \mathbf{\Lambda} \mathbf{C}^{-1} = \begin{bmatrix} \gamma_k \mathbf{\Sigma}^2 + \mathbf{\Sigma} \mathbf{M} \mathbf{\Sigma} & \mathbf{\Sigma} \mathbf{M} \\ -\gamma_k \mathbf{\Sigma} & \mathbf{0} \end{bmatrix}$; {Eigendecomposition}
$\mathbf{S}_1 = \mathbf{\Psi} \mathbf{C}$;
$\widehat{\mathbf{S}}_1 = \mathbf{\Psi} (\mathbf{\Psi}^\top \mathbf{\Psi})^{-1} \mathbf{C}^{-\top}$;

Algorithm C.2 Singular Value Decomposition of $\mathbf{H}_+$.

Input: $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\mathbf{Y} \in \mathbb{R}^{n \times m}$, $\gamma \in \mathbb{R}$;
Compute $\mathbf{U}, \mathbf{\Sigma}$, and $\mathbf{V}$ using Algorithm 3.1 and form $\mathbf{\Psi} = [\ \mathbf{U} \quad \mathbf{V}\ ]$;
$\mathbf{M} = \mathbf{V}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{Y})^{-1} \mathbf{X}^\top \mathbf{V}$; {Intermediate matrix}
$\mathbf{R}^\top \mathbf{R} = \mathbf{\Psi}^\top \mathbf{\Psi}$; {Cholesky factorization}
$\widetilde{\mathbf{U}} \widetilde{\mathbf{\Sigma}} \widetilde{\mathbf{V}}^\top = \mathbf{R} \begin{bmatrix} \gamma \mathbf{\Sigma}^2 + \mathbf{\Sigma} \mathbf{M} \mathbf{\Sigma} & \mathbf{\Sigma} \mathbf{M} \\ -\gamma \mathbf{\Sigma} & \mathbf{0} \end{bmatrix} \mathbf{R}^{-1}$; {SVD}
$(\mathbf{U}_+)_1 = \mathbf{\Psi} \mathbf{R}^{-1} \widetilde{\mathbf{U}}$;
$(\mathbf{V}_+)_1 = \mathbf{\Psi} \mathbf{R}^{-1} \widetilde{\mathbf{V}}$;

Acknowledgments. We thank the editor and referee for the careful reading of the manuscript, which improved the article and enabled a simplified proof of Theorem 3.1.

REFERENCES

[1] R. T. Behrens and L. L. Scharf, *Signal processing applications of oblique projection operators*, IEEE Trans. Signal Process., 42 (1994), https://doi.org/10.1109/78.286957.

[2] J. J. Brust, R. F. Marcia, and C. G. Petra, *Large-scale quasi-newton trust-region methods with low-dimensional linear equality constraints*, Comput. Optim. Appl., 74 (2019), pp. 669–701, https://doi.org/10.1007/s10589-019-00127-4.

[3] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi-Newton matrices and their use in limited-memory methods*, Math. Program., 63 (1994), pp. 129–156, https://doi.org/10.1007/BF01582063.

[4] Y. Censor and T. Elfving, *Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 40–58, https://doi.org/10.1137/S089547980138705X.

[5] J. Dennis and J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89, https://doi.org/10.1137/1019005.

[6] G. Golub and C. van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893, https://doi.org/10.1137/0717073.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[8] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394, https://doi.org/10.1145/962437.962439.

[9] N. Halko, P. Martinsson, and J. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806.

[10] N. J. Higham, *Algorithm* 694: *A collection of test matrices in MATLAB*, ACM Trans. Math. Software, 17 (1991), pp. 289–305, https://doi.org/10.1145/114697.116805.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed., Cambridge University Press, Cambridge, 2012.

[12] A. Liutkus, *Software: Randomized Singular Value Decomposition*, Version 1.0.0.0, 2014, https://www.mathworks.com/matlabcentral/fileexchange/47835-randomized-singular-value-decomposition.

[13] P. O'Leary, *On bounds for scaled projections and pseudoinverses*, Linear Algebra Appl., 132 (1990), pp. 115–117, https://doi.org/10.1016/0024-3795(90)90056-I.

[14] G. Stewart, *On scaled projections and pseudoinverses*, Linear Algebra Appl., 112 (1989), pp. 189–193, https://doi.org/10.1016/0024-3795(89)90594-6.