

# qstiefel: A MATLAB Package for Riemannian Optimization over Quaternion Stiefel Manifold

Ying Wang\*, Yuning Yang\*<sup>†</sup>

## Abstract

This document describes qstiefel, a MATLAB package for Riemannian optimization over the quaternion Stiefel manifold  $\text{St}_{\mathbb{Q}}(n, p)$ , i.e., the set of  $n \times p$  partially unitary quaternion matrices. Optimization problems on  $\text{St}_{\mathbb{Q}}(n, p)$  arise in several areas, including color image processing, numerical quaternion linear algebra, and airborne direct georeferencing. To facilitate its use, the qstiefel package is written in the style of Manopt, a popular and efficient MATLAB package for general Riemannian optimization.

## 1 Files

The qstiefel package is a publicly available (<https://gitee.com/wwwyyyying/qstiefel>) MATLAB package for Riemannian optimization over the quaternion Stiefel manifold and follows the style of the well-known Manopt package [1]. The files in the qstiefel package are listed as follows:

- stiefelquaternionfactory.m

This is the main file in the package. It follows the style of Manopt and gives codes for Riemannian optimization tools over the quaternion Stiefel manifold, including Riemannian metric, Riemannian norm, tangent space projector, Riemannian gradient, Riemannian Hessian, exponential map, retractions (qr, polar, and Cayley retractions), transporter, and so on.

- the “examples” folder

This folder has five files: dominant\_invariant\_subspace\_quaternion.m, PCA\_stochastic\_quaternion.m, robust\_pca\_quaternion.m, sparse\_pca\_quaternion.m, and truncated\_svd\_quaternion.m. They solve five representative optimization problems defined over the quaternion Stiefel manifold, by calling the code stiefelquaternionfactory.m in the qstiefel package and solvers in the Manopt package. The five problems are: dominant eigenvalue decomposition of a Hermitian quaternion matrix (EVD), principal component analysis of quaternion data (QPCA), robust principal component analysis of quaternion data (robust QPCA), sparse principal component analysis of quaternion data (sparse QPCA), and dominant singular value decomposition of a quaternion matrix (SVD). Both dominant\_invariant\_subspace\_quaternion.m and PCA\_stochastic\_quaternion.m can solve the QPCA problem; the former is more accurate while the latter is more efficient for large-scale data.

- test\_all\_examples.m

This file tests the examples listed above.

---

\*College of Mathematics and Information Science, Guangxi University, Nanning, 530004, China

<sup>†</sup>Corresponding author: Yuning Yang, yyang@gxu.edu.cn

- readme.pdf

The document you are reading now.

- the “Qlab” folder

The codes in this folder are written by Chao Chang and Yuning Yang. This folder contains some functions related to quaternion matrices. In particular, we need the following ones: QLEQ.m (which solves the quaternion linear equation  $AX=b$ ), QLEQL.m (which solves the quaternion linear equation  $XA=b$ ), and normQf (which computes the Frobenius norm of a quaternion matrix).

To use the qstiefel package, you need to first download two MATLAB packages: Manopt (available at <https://www.manopt.org>) and QTFM (available at <https://qtfm.sourceforge.io>). The former is a MATLAB package for Riemannian optimization and the latter is a MATLAB package for most operations and algorithms related to quaternions and quaternion matrices. After adding the above two packages and the qstiefel package (including the Qlab folder) to MATLAB path, the user can directly call stiefelquaternionfactory.m and Manopt to solve optimization problems over the quaternion Stiefel manifold. The user can run the file test\_all\_examples.m to test all examples provided in the qstiefel package.

## 2 Tests

We present results from some preliminary tests to illustrate the effectiveness of the qstiefel package. All the numerical experiments were conducted on a PC running macOS with an Intel Core i5 2.4 GHz CPU and 8GB memory. To reproduce our results, the user can simply run the file test\_all\_examples.m (after setting up the required MATLAB packages as explained above).

**Dominant Eigenvalue Decomposition.** For a Hermitian quaternion matrix, one is often interested in computing the largest  $p$  eigenvalues and the associated eigenvectors (the dominant eigenpairs). This need arises in quaternion PCA, for example. The standard package for quaternion matrix computations, QTFM, does not provide a function that directly computes dominant eigenpairs. One has to use the function eig in QTFM to compute the full spectral decomposition and then truncate the  $p$  leading eigenpairs (in other words, there is no analogue of eigs in QTFM). This is inefficient. In our machine, we observed that for a Hermitian quaternion matrix of size  $1000 \times 1000$ , it takes **over 300 seconds** to compute the full eigenvalue decomposition by calling eig in QTFM. On the other hand, by formulating dominant eigenvalue decomposition as an optimization problem over the quaternion Stiefel manifold, our code dominant\_invariant\_subspace\_quaternion.m (adapted from dominant\_invariant\_subspace.m in Manopt) can directly compute the leading eigenpairs by calling stiefelquaternionfactory.m in the qstiefel package and the Riemannian trust region solver in Manopt. For  $p = 5$  and quaternion matrices of size  $1000 \times 1000$ , this takes **less than 10 seconds**. The improvement over QTFM is substantial. For matrices of larger size, the advantage of qstiefel over QTFM will be more pronounced. In the future, more comprehensive comparisons with other algorithms (especially those based on numerical linear algebra) are desired.

**Dominant Singular Value Decomposition.** For a quaternion matrix of size  $m \times n$ , one is often interested in computing the largest  $p$  singular values and the associated left and right singular vectors. This need arises in low-rank approximation of quaternion matrices, for example. Similarly, QTFM does not provide such a function directly. One must resort to the full SVD function in QTFM. In our machine, for a quaternion matrix of size  $1000 \times 1000$ , it typically takes **130 seconds** to compute the full SVD by calling the function svd in QTFM. On the other hand, one can formulate dominant SVD as an optimization problem over the Cartesian product of two quaternion Stiefel manifolds and solve it by calling stiefelquaternionfactory.m in the qstiefel package and the Riemannian trust region solver in Manopt. For  $p = 5$  and quaternion matrices of size  $1000 \times 1000$ , this usually takes **60 seconds**. The improvement over QTFM is clear. For matrices of larger size,

the advantage of qstiefel over QTFM will be more pronounced. We remark that our implementation of the code for quaternion SVD (i.e., the file `truncated_svd_quaternion.m`, which is adapted from `truncated_svd.m` in Manopt) is the most straightforward one and we expect it can be further customized to improve efficiency. In the future, more comprehensive comparisons with other algorithms (especially those based on numerical linear algebra) are desired.

**Principal Component Analysis of Quaternion Data.** For quaternion data (such as color images), one is interested in dimension reduction. This leads to quaternion principal component analysis (QPCA), which corresponds to computing dominant quaternion eigenvalue decomposition (or dominant quaternion SVD). So QPCA can be solved by calling `dominant_invariant_subspace_quaternion.m` (or `truncated_svd_quaternion.m`) in the qstiefel package. However, for large-scale data, this may be costly. The file `PCA_stochastic_quaternion.m` (adapted from `PCA_stochastic.m` in Manopt) implements a Riemannian stochastic gradient descent algorithm for QPCA (which is an optimization problem over the quaternion Stiefel manifold) and is more efficient for large-scale QPCA (although its returned solution is less accurate). In our machine, for a quaternion data matrix of size  $100000 \times 1000$ , it takes `PCA_stochastic_quaternion.m` about **5 seconds** to obtain a reasonably accurate solution. Meanwhile, it takes `dominant_invariant_subspace_quaternion.m` **175 seconds** to obtain a solution (which is highly accurate). Therefore for large-scale data, one might prefer the `PCA_stochastic_quaternion.m` code.

**Sparse Principal Component Analysis of Quaternion Data.** QPCA is a dimension reduction technique for quaternion data. Like its real analogue (i.e., the usual PCA), QPCA may lack interpretability. In the real case, this issue can be addressed, e.g., by the sparse PCA model proposed in [2]. One can extend this model to quaternion data. Our code `sparse_pca_quaternion.m` implements this idea and is adapted from the code `sparse_pca.m` in Manopt. This sparse QPCA model appears to be new, so there are no customized algorithms for it in the literature yet. Thus we did not include numerical comparisons for it.

**Robust Principal Component Analysis of Quaternion Data.** As mentioned, QPCA is a dimension reduction technique for quaternion data. Like its real analogue (i.e., the usual PCA), QPCA lacks robustness. In the real case, this issue is addressed by a robust PCA model. There are different variants. One can try to extend such models to quaternion data. Our code `robust_pca_quaternion.m` is adapted from the code `robust_pca.m` in Manopt. This particular robust QPCA model appears to be new, so there are no customized algorithms for it in the literature yet. Thus we did not include numerical comparisons for it.

## References

- [1] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(42):1455–1459, 2014.
- [2] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(2), 2010.