LAPACK EIGEN/SVD-SOLVERS

Zhengbo Zhou*

Abstract

This note concentrate on paper [7, 8].

1 Introduction

LAPACK provides several solvers for symmetric eigenvalue problems(SYP). However, the Jacobi algorithm is not explicitly used in the LAPACK routines for SYP. Several papers using LAPACK's singular value decomposition(SVD) routines to help computing the eigenvalues.

This note is based on the following two papers [7] and [8].

- Weiguo Gao, Yuxin Ma, and Meiyue Shao. A mixed precision Jacobi SVD algorithm. *ArXiv*, 2022.
- Zhiyuan Zhang and Zheng-Jian Bai. A mixed precision Jacobi method for the symmetric eigenvalue problem.*ArXiv*, 2022.

1.1 LAPACK Name Scheme

All drives and computational routines have names of the form **XYYZZZ**, where for some driver routine the 6th character is blank.

The first letter, **X**, indicates the data type as follows:

S	Real
D	Double Precision
C	Complex
Z	Double Complex

When we wish to refer to an LAPACK routine generically, regardless of data type, we replace the first letter by "x". Thus xGESV refers to any or all of the routines `SGESV`, `CGESV`, `DGESV` and `ZGESV`.

The next two letters, **YY**, indicate the type of matrix (or of the most significant matrix). Most of these two-letter codes apply to both read and complex matrices; a few apply specifically to one or the other, as indicated in Table 1.

When we wish to refer to a class of routines that performs the same function on different types of matrices, we replace the first three letters by "xyy". Thus `xyySVX` refers to all the expert driver routines for systems of linear equations that are listed in Table 1.

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (`zhengbo.zhou@postgrad.manchester.ac.uk`).

Table 1: Matrix types in the LAPACK naming scheme

BD	**B**i**D**iagonal
DI	**DI**agonal
GB	**G**eneral **B**and
GE	**GE**neral (i.e. unsymmetric, in some cases rectangular)
GG	**G**eneral matrices, **G**eneralized problem (i.e. a pair of general matrices)
GT	**G**eneral **T**ridiagonal
HB	(complex) **H**ermitian **B**and
HE	(complex) **HE**rmitian
HG	upper **H**essenberg matrix, **G**eneralized problem (i.e. a Hessenberg and a triangular matrix)
HP	(complex) **H**ermitian, **P**acked storage
HS	upper **H**e**S**senberg
OP	(real) **O**rthogonal, **P**acked storage
OR	(real) **OR**thogonal
PB	symmetric or Hermitian **P**ositive definite **B**and
PO	symmetric or Hermitian **PO**sitive definite
PP	symmetric or Hermitian **P**ositive definite, **P**acked storage
PT	symmetric or Hermitian **P**ositive definite **T**ridiagonal
SB	(real) **S**ymmetric **B**and
SP	**S**ymmetric, Packed storage
ST	(real) **S**ymmetric **T**ridiagonal
SY	**SY**mmetric
TB	**T**riangular Band
TG	**T**riangular matrices, **G**eneralized problem (i.e., a pair of triangular matrices)
TP	**T**riangular, **P**acked storage
TR	**TR**iangular (or in some cases quasi-triangular)
TZ	**T**rape**Z**oidal
UN	(complex) **UN**itary
UP	(complex) **U**nitary, **P**acked storage

The last three letters **ZZZ** indicate the computation performed. They are explained in [2, 1999, Sec 2.4]. For example, `SGEBRD` is a **S**ingle precision routine that performs a bidiagonal reduction (**BRD**) of a real **GE**neral matrix.

2 LAPACK Routine for SVD Problem by [7]

The paper [7] is working with the SVD problem rather the SEP. However, they point out the following LAPACK routines that they used for their implementation of the Jacobi SVD. We will list these routines and give some brief introductions to them.

The following 4 routines are used in [7] for computing SVD.

Section 2.1 `S/D/C/Z GE JSV`
Section 2.2 `S/D/C/Z GE SVJ`
Section 2.3 `S/D/C/Z GE QRF`
Section 2.4 `S/D/C/Z GE SVD`

The following 2 routines are used in [7] for computing eigendecomposition.

Section 2.5 `S/D SY EV`

2.1 `DGEJSV` : Double Precision, General Matrices, Preconditioned Jacobi SVD Algorithm

`DGEJSV` : Computes the *singular value decomposition* of a matrix $A \in \mathbb{R}^{m\times n}$, where $m \geq n$. `DGEJSV` can sometimes compute tiny singular values and their singular vectors much more accurately than other SVD routines. `DGEJSV` implements a *preconditioned Jacobi SVD algorithm*. It uses `DGEQP3`, `DGEQRF`, and `DGELQF` as a preprocessor, which in some cases results in much higher accuracy.

Example 2.1. Suppose matrix A has the structure $A = D_1CD_2$, where D_1 and D_2 are arbitrarily ill-conditioned diagonal matrices and C is well-conditioned matrix. In this case, complete pivoting in the first QR factorizations provides accuracy dependent on the condition number of C, and independent of D_1 and D_2.

Example 2.2. If A can be written as $A = BD$, with well-conditioned B and some diagonal D, then the high accuracy is guaranteed, both theoretically and in software, independent of D.

For more details, see [5, 6]

2.2 `DGESVJ` : Double Precision, General Matrices, Computing SVD using Jacobi Plane Rotations

`DGESVJ` computes the SVD of a matrix $A \in \mathbb{R}^{m\times n}$, where $M \geq N$. The SVD of A is written as

$$A = U\Sigma V^T.$$

`DGESVJ` can sometimes compute tiny singular values and their singular vectors much more accurately than other SVD routine.

The orthogonal $n \times n$ matrix V is obtained as a product of Jacobi plane rotations. The rotations are implements as fast scaled rotations of Anda and Park [1]. In case of underflow of the Jacobi angle, a modified Jacobi transformation of Drmač [4] is used. The relative accuracy of the computed singular values and the accuracy of the computed singular vectors (in angle metric) is as guaranteed by the theory of Demmel and Veselic [3]. The condition number that determines the accuracy in the full rank case is essentially $\min_{\Delta \text{ is diagonal}} \kappa_2(A\Delta)$. The best performance of this Jacobi SVD procedure is achieved if used in an accelerated version of Drmač and Veselić [5, 6].

2.3 DGEQRF : Double Precision, General Matrices, QR Factorization

DGEQRF : Computes a QR factorization of a real $m \times n$ matrix A:

$$A = Q \times \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal, $R \in \mathbb{R}^{n\times n}$ is an upper triangular matrix, and 0 is a $(m-n) \times n$ zero matrix if $m > n$. The matrix Q is represented as a product of elementary reflectors

$$Q = H_1 H_2 \cdots H_k, \quad k = \min(m, n).$$

Each H_i has the form

$$H_i = I - \tau v v^T$$

where τ is a real scalar, and v is a real vector with $v(1 : i-1) = 0$ and $v(i) = 1$.

2.4 DGESVD : Single Precision, General Matrices, Singular Value Decomposition

DGESVD computes the singular value decomposition (SVD) of a real $m \times n$ matrix A. The SVD is written

$$A = U\Sigma V^T$$

where $\Sigma \in \mathbb{R}^{m\times n}$ is an matrix which is zero except for its $\min(m, n)$ diagonal elements, and $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$ are orthogonal matrices. The diagonal elements of Σ are the singular values of A; they are real and non-negative, and are returned in descending oreder. The first $\min(m, n)$ columns of U and V are the left and right singular vectors of A.

xGESVD, where x $\in$ S,C, computes the SVD of a general matrix by

- Reducing it to bidiagonal form B via routine xGEBRD[1].
- Call xBDSQR[2] to compute the SVD of B.

2.5 DSYEV : Double Precision, Symmetric Matrices, Eigendecomposition

DSYEV : A *simple* driver computes all the eigenvalues and (optionally) eigenvectors by

- Call SSYTRD to reduce symmetric matrix to tridiagonal form.
- For eigenvalues only, call SSTERF[3]. For eigenvectors, first call SORGTR[4] to generate the orthogonal matrix, then call SSTEQR[5].

[1] xGEBRD reduce a general $m \times n$ matrix A to upper or lower bidiagonal form B by an orthogonal transformation: $Q^T AP = B$.

[2] xBDSQR computes the singular values of a real $n \times n$ bidiagonal matrix B using the implicit zero-shift QR algorithm.

[3] SSTERF computes all eigenvalues of a symmetric tridiagonal matrix using the Pal-Walker-Kahan variant of the QL or QR algorithm.

[4] SORGTR generates a real orthogonal matrix Q which is defined as the product of $n-1$ elementary reflectors of order N.

[5] SSTEQR computes all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the implicit QL or QR method. The eigenvectors of a full or band symmetric matrix can also be found if SSYTRD has been used to reduce this matrix to tridiagonal form.

Remark 2.3. Paper [8, 2022] uses `SSYEV` to compute the eigenvalues and eigenvectors of a given matrix in low precision.

References

[1] Andrew A. Anda and Haesun Park. Fast plane rotations with dynamic scaling. *SIAM Journal on Matrix Analysis and Applications*, 15(1):162–174, 1994. (Cited on p. 3.)

[2] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Third edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999. 429 pp. ISBN 0-89871-447-8. (Cited on p. 2.)

[3] James Demmel and Krešimir Veselić. Jacobi's method is more accurate than QR. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1204–1245, 1992. (Cited on p. 3.)

[4] Zlatko Drmac. Implementation of jacobi rotations for accurate singular value computation in floating point arithmetic. *SIAM Journal on Scientific Computing*, 18(4): 1200–1222, 1997. (Cited on p. 3.)

[5] Zlatko Drmač and Krešimir Veselić. New fast and accurate Jacobi SVD algorithm. I. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1322–1342, 2008. (Cited on p. 3.)

[6] Zlatko Drmač and Krešimir Veselić. New fast and accurate Jacobi SVD algorithm. II. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1343–1362, 2008. (Cited on p. 3.)

[7] Weiguo Gao, Yuxin Ma, and Meiyue Shao. A mixed precision Jacobi SVD algorithm, 2022. (Cited on pp. 1, 3, 4, and 5.)

[8] Zhiyuan Zhang and Zheng-Jian Bai. A mixed precision Jacobi method for the symmetric eigenvalue problem, November 2022. (Cited on pp. 1 and 5.)