Evaluation of a New Eigen Decomposition Algorithm for Symmetric Tridiagonal Matrices

Hiroaki Tsuboi *, Taro Konda *, Masami Takata †, Kinji Kimura ‡, Masashi Iwasaki * and Yoshimasa Nakamura *

*SORST, JST and Graduate School of Informatics Kyoto University Kyoto 606-8501, JAPAN

†Graduate School of Humanity and Science Nara Women's University Nara 630-8506, JAPAN

‡ CREST, JST and College of Science Rikkyo University Tokyo 171-8501, JAPAN

Abstract ***This paper focuses on a new extension version of double Divide and Conquer (dDC) algorithm to eigen decomposition. Recently, dDC was proposed for singular value decomposition (SVD) of rectangular matrix. The dDC for SVD consists of two parts. One is Divide and Conquer (D&C) for singular value and the other is twisted factorization for singular vector. The memory usage of dDC is smaller than that of D&C. Both theoretical and running time are also shorter than those of D&C. In this paper, a new dDC for eigen decomposition is proposed. A shift of origin is introduced into our dDC. By some numerical tests, dDC is evaluated with respect to running time and accuracy.***

Keywords: Eigen Decomposition, double Divide and Conquer, Simplified Divide and Conquer, Twisted Factorization, Shift of Origin

1 Introduction

Any $n \times n$ symmetric matrix is transformed into a symmetric tridiagonal matrix by using a sequence of Householder transformations [2]. This preconditioning process help us to shorten execution time drastically. Eigen decomposition algorithms of symmetric tridiagonal matrices are also important. Divide and Conquer(D&C) [1, 6] is one of the standard algorithms for symmetric tridiagonal eigen decomposition. D&C requires $O(n^2)$ memory capacity, and then large scale eigen decomposition may fail by lack of memory. Theoretical time in D&C depends on frequency of deflation and is between $O(n^2)$ and $O(n^3)$.

In 2006, double Divide and Conquer (dDC) [7] for singular value decomposition (SVD) has been proposed. In dDC, singular values and singular vectors are computed by using a part of D&C and twisted factorization, respectively. In this paper, we design a new eigen decomposition algorithm which is an extension of dDC for SVD. The memory usage and the theoretical time are $O(n)$ and $O(n^2)$, respectively. Our dDC is also parallelizable easily.

In section 2, we design a new dDC for eigen decomposition. By some numerical tests in section 3, we discuss dDC with respect to theoretical time and accuracy in section 4.

2 double Divide and Conquer

We first adopt a part of D&C and twisted factorization for computing eigenvalues and eigenvectors of positive-definite T. In this paper, twisted factorization is done with the help of the dstqds and dqds transformations [5]. These qd-type transformations for positive-definite matrix have robustness [3]. It is of significance to note that $\bar{T} \equiv T - sI$ with shift s has the same eigenvectors of T. Namely, we may compute the eigenvectors of positive-definite $\bar{T}$ instead of nonpositive-definite T. By combining simplified D&C, twisted factorization and shift of origin, we design a new dDC for eigen decomposition.

Standard D&C is an algorithm for not only eigenvalues but also eigenvectors. It is known that D&C algorithm can be simplified if only eigenvalues are desired [6]. Let $T \in \mathbb{R}^{n \times n}$ be divided as

$$T = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + \theta\beta \begin{pmatrix} e_l \\ \theta^{-1} e_f \end{pmatrix} \begin{pmatrix} e_l^\top & \theta^{-1} e_f^\top \end{pmatrix},$$

$$T_1 \in \mathbb{R}^{n_1 \times n_1},\ e_l \equiv (0, \cdots, 0, 1)^\top \in \mathbb{R}^{n_1},$$

$$T_1 \in \mathbb{R}^{n_2 \times n_2},\ e_f \equiv (1, 0, \cdots, 0)^\top \in \mathbb{R}^{n_2}$$

where θ is nonzero constant and β corresponds to an (n_1, n_1+1)-element of T. Let $T_k, k = 1,2$ be decomposed as $T_k = Q_k D_k Q_k^{\mathsf{T}}$, where Q_k are eigenvector matrices and D_k are diagonal matrices with eigenvalues of T_k. Simplified D&C is different from the standard version that a large part of Q_k are not necessary for eigenvalues of original matrix T. Simplified D&C requires only the first row $\boldsymbol{q}_f$ of Q_2 and the last row $\boldsymbol{q}_l$ of Q_1. We can obtain eigenvalues of T by computing the roots λ of the secular equation

$$1 + \beta(\theta + \theta^{-1})\boldsymbol{z}^{\mathsf{T}}(D_{12} - \lambda I)^{-1}\boldsymbol{z} = 0,$$
$$D_{12} \equiv \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}, \boldsymbol{z} \equiv \frac{1}{\sqrt{1+\theta^{-2}}}\begin{pmatrix} \boldsymbol{q}_l^{\mathsf{T}} \\ \theta^{-1}\boldsymbol{q}_f^{\mathsf{T}} \end{pmatrix}.$$

Moreover the first row $\boldsymbol{q}_{T;f}$ or the last row $\boldsymbol{q}_{T;l}$ of eigenvector matrix of T are necessary for eigenvalues of original matrix if T is not original. Let $\tilde{Q}$ be the eigenvector matrix of $D_{12} + \beta(\theta+\theta^{-1})\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}$. Then $\boldsymbol{q}_{T;f}$ and $\boldsymbol{q}_{T;l}$ are given as

$$\boldsymbol{q}_{T;f} = \boldsymbol{q}_{12;f}\tilde{Q}, \boldsymbol{q}_{T;l} = \boldsymbol{q}_{12;l}\tilde{Q}$$

in terms of the first row $\boldsymbol{q}_{12;f}$ and the last row $\boldsymbol{q}_{12;l}$ of $Q_{12} \equiv \mathrm{diag}\{Q_1, Q_2\}$, respectively. Obviously, theoretical time decreases in comparison with standard D&C which computes both eigenvalues and eigenvectors. Simplified D&C runs for $O(n^2)$ time. Especially, if $\boldsymbol{z}$ includes a zero-element or any elements of D_{12} are mutually equal, theoretical time of simplified D&C is shortened by deflation.

If eigenvalues are given, twisted factorization is a useful algorithm for computing eigenvectors. Let T be a symmetric tridiagonal matrix with (k,k)-element a_k, $(k,k+1)$-element b_k and an eigenvalue λ. Of course, $(k+1,k)$-element of T is b_k. Then a corresponding eigenvector $\boldsymbol{v}$ to λ satisfies $(T - \lambda I)\boldsymbol{v} = 0$. Generally, we obtain an approximate eigenvalue $\hat{\lambda}$ with some error by numerical algorithms. To compute a higher accurate eigenvector from $\hat{\lambda}$, we had better find $\hat{\boldsymbol{v}}$ such that

$$(T - \hat{\lambda}I)\hat{\boldsymbol{v}} = \gamma_\varrho \boldsymbol{e}_\varrho \qquad (1)$$

where γ_ϱ is a residual parameter and $\boldsymbol{e}_\varrho$ is an identity vector with ϱ-th element 1. The vector $\hat{\boldsymbol{v}}$ in Eq.(1) with $\gamma_\varrho \neq 0$ approaches to the correct eigenvector $\boldsymbol{v}$. Moreover, a twisted factorization of $T - \hat{\lambda}I$

$$T - \hat{\lambda}I = N_\varrho D_\varrho N_\varrho^{\mathsf{T}}, \qquad (2)$$
$$N_\varrho \equiv \begin{pmatrix} 1 & & & & & & 0 \\ l_1 & 1 & & & & & \\ & \ddots & \ddots & & & & \\ & & l_{\varrho-1} & 1 & u_\varrho & & \\ & & & & 1 & \ddots & \\ & & & & & \ddots & u_{n-1} \\ 0 & & & & & & 1 \end{pmatrix},$$
$$D_\varrho \equiv \mathrm{diag}(d_1^+, \cdots, d_{\varrho-1}^+, \gamma_\varrho, d_{\varrho+1}^-, \cdots, d_n^-),$$
$$\gamma_\varrho \equiv d_\varrho^+ + d_\varrho^- + \hat{\lambda} - a_\varrho.$$

through LDL and UDU decompositions of $T - \hat{\lambda}I$. By substituting Eq.(2) for Eq.(1), we have

$$N_\varrho D_\varrho N_\varrho^{\mathsf{T}}\hat{\boldsymbol{v}} = \gamma_\rho \boldsymbol{e}_\varrho. \qquad (3)$$

Since it is obvious that $D_\varrho \boldsymbol{e}_\varrho = \gamma_\rho \boldsymbol{e}_\varrho$ and $N_\varrho \boldsymbol{e}_\varrho = \boldsymbol{e}_\varrho$, Eq.(3) is transformed to

$$N_\varrho^{\mathsf{T}}\hat{\boldsymbol{v}} = \boldsymbol{e}_\varrho. \qquad (4)$$

By choosing a twist index ϱ such that $|\gamma_\varrho| = \min_k |\gamma_k|$, computed $\hat{\boldsymbol{v}}$ in Eq.(4) becomes a good approximation of the correct eigenvector $\boldsymbol{v}$. The element $\hat{v}(k)$ of $\hat{\boldsymbol{v}} = (\hat{v}(1), \hat{v}(2), \cdots, \hat{v}(n))^{\mathsf{T}}$ is sequentially computed by $\hat{v}(k) = 1$ (if $k = \varrho$) or $-l_k\hat{v}(k+1)$ (if $\varrho - 1 \geq k \geq 1$) or $-u_{k-1}\hat{v}(k-1)$ (if $\varrho + 1 \leq k \leq n$). If $d_{k_0}^+ = 0$ or $d_{k_0}^- = 0$ for some k_0, then $\hat{v}(k_0)$ is exceptionally given as $\hat{v}(k_0) = b_{k_0+1}\hat{v}(k_0+2)/b_{k_0}$ (if $k_0 < \varrho$) or $b_{k_0-2}\hat{v}(k_0-2)/b_{k_0-1}$ (if $k_0 > \varrho$). It is shown in [4] that theoretical time of twisted factorization is $O(n^2)$.

By combining $O(n)$-shift with simplified D&C and twisted factorization, our dDC takes $O(n^2)$ time for eigen decomposition of original T.

3 Numerical tests

In this section, we compare the following algorithms by some numerical tests.

- D&C: Divide and Conquer (DSTEDC in LAPACK[8])
- dDC: double Divide and Conquer
- dDC+I: double Divide and Conquer with an inverse iteration

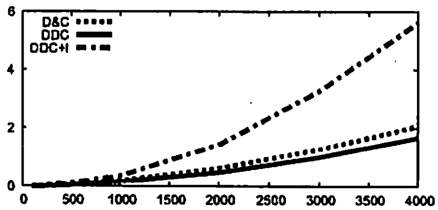

Figure 1: A graph of the Type 1 matrix size (x-axis) and the running time (sec) for eigen decomposition (y-axis)

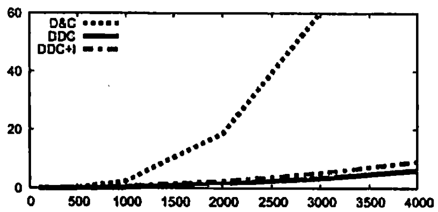

Figure 2: A graph of the Type 2 matrix size (x-axis) and the running time (sec) for eigen decomposition (y-axis)

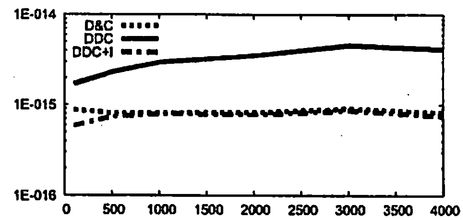

Figure 3: A graph of the Type 1 matrix size (x-axis) and the relative gap c_1 (y-axis)

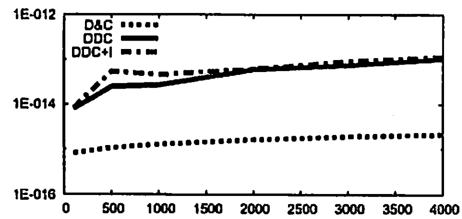

Figure 4: A graph of the Type 2 matrix size (x-axis) and the relative gap c_1 (y-axis)

Let us introduce two kinds of test matrices with symmetric tridiagonal form. Diagonal and subdiagonal elements of Type 1 are given randomly. Type 2 matrix with random eigenvalues is constructed by using Lanczos method [2]. We prepare 50 and 10 different test matrices for Type 1 and Type 2, respectively. Numerical tests were carried out by our computer with CPU:Intel Pentium 4 2.66GHz, Memory:1GB, Compiler:GNU g77 ver.3.4.5 (option:-O3) and OS:Linux ver.2.4.27-2-386.

Figure 1 and 2 show the averaged running time for computing eigen decompositions of Type 1 and Type 2 matrices, respectively. According to Figure 1 and 2, dDC runs the fastest for eigen decomposition of both Type 1 and Type 2. Obviously, running time of dDC+I increases by adding an inverse iteration to dDC. The eigen decomposition of Type 1 is computed with great shorter than that of Type 2 if we use D&C. The eigen decomposition of Type 1 is accelerated by some deflations [1]. Hardly any deflations happen in the case of Type 2. Namely, running time of D&C is extremely variable.

Let $\hat{D}$ denote a diagonal matrix with an approximate eigenvalues $\hat{\lambda}_k$ of T. Let $\hat{Q} = \{\hat{Q}_{i,j}\}_{1\le i,j\le n}$ be an approximate eigenvector matrix to the correct matrix $Q = \{Q_{i,j}\}_{1\le i,j\le n}$. Then we introduce the following criteria for accuracy

$$c_1 = \frac{\|\hat{Q}\hat{D}\hat{Q}^\top - T\|_\infty}{\|T\|_\infty},\quad c_2 = \|\hat{Q}\hat{Q}^\top - I\|_\infty,$$

$$c_3 = \sum_{k=1}^{n} \frac{|\lambda_k - \hat{\lambda}_k|}{|\lambda_k|},\quad c_4 = \sum_{1\le i,j\le n} |Q_{i,j} - \hat{Q}_{i,j}|$$

where $\{\lambda_k\}_{k=1,2,\ldots,n}$ are the correct eigenvalues of T and $\|\ \|_\infty$ denotes the infinity norm. Figure 3 and 4 describe the relative gap c_1 between the original matrix T and computed eigen decomposition $T \approx \hat{Q}\hat{D}\hat{Q}^\top$. Figure 5 and 6 illustrate the orthogonality c_2 of computed eigenvector matrix $\hat{Q}$. Especially, for Type 2 matrix, the relative gap of $\hat{\lambda}_k$ and the absolute gap of $\hat{Q}$ are shown in Figure 7 and 8, respectively. Figure 3 and 4 suggest that dDC+I and D&C should be applied to the eigen decomposition of Type 1 and Type 2, respectively, if the relative gap c_1 is desired to be small. It is emphasized here that $\hat{Q}$ is given by product of several orthogonal matrices if we use D&C. While every eigenvector is separately computed by twisted factorization of dDC and dDC+I. As shown in Figure 5 and 6, $\hat{Q}$ with the highest orthogonality is accordingly computed by using D&C. Moreover, we see from Figure 7 that $\hat{\lambda}_k$ are computed with the almost same accuracy if we use either D&C or dDC. Obviously, computed $\hat{\lambda}_k$ by dDC+I are equal to

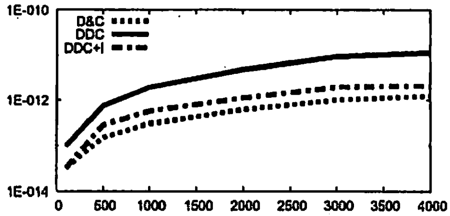

Figure 5: A graph of the Type 1 matrix size (x-axis) and the orthogonality c_2 of computed eigenvector matrix (y-axis)

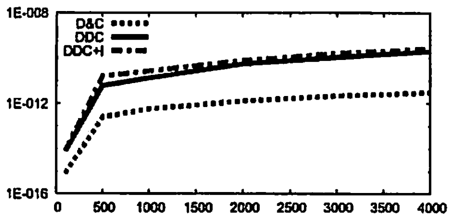

Figure 6: A graph of the Type 2 matrix size (x-axis) and the orthogonality c_2 of computed eigenvector matrix (y-axis)

that by dDC. This is because D&C, dDC and dDC+I adopt the same algorithm for eigenvalues. In Figure 8, computed $\hat{Q}$ by dDC+I has smaller gaps c_4 than that by dDC. Compared with dDC, computed eigenvectors approach to the correct vectors by adding an inverse iteration. Figure 6 and 8 also implies that the orthogonality c_2 of computed eigenvector matrix does not have much concern with the gap c_4 from the correct eigenvector matrix.

4 Conclusion

In this paper, we proposed double Divide and Conquer (dDC) algorithm for eigen decomposition of symmetric tridiagonal matrix. Our new eigen decomposition algorithm is designed based on dDC for SVD. By some numerical tests, it is shown that dDC is as fast as or even faster than D&C for eigen decomposition. Though computed eigenvector matrix $\hat{Q}$ by dDC or dDC+I is with less orthogonality than that by D&C, $\hat{Q}$ is with smaller gap from the correct matrix. The eigenvector also approaches to more accurate vector if we use only an inverse iteration. As a future work, we will design a parallel version of dDC.

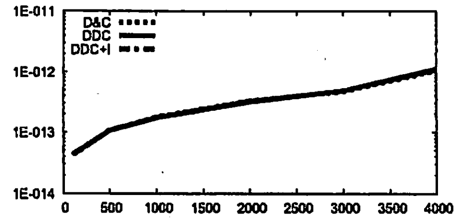

Figure 7: A graph of the Type 2 matrix size (x-axis) and the relative gap c_3 of computed eigenvalues (y-axis)

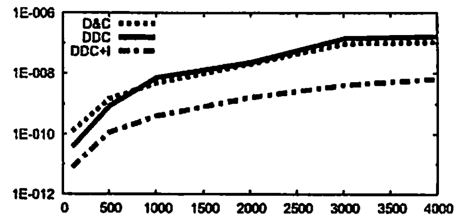

Figure 8: A graph of the Type 2 matrix size (x-axis) and the absolute gap c_4 of computed eigenvector matrix (y-axis)

References

[1] J. J. M. Cuppen: *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., Vol. 36 (1981), pp.177–195.

[2] J. Demmel: *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[3] J. Demmel and W. Kahan: *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Sta. Comput., Vol. 11. No. 5, (1990), pp.873–912.

[4] I. S. Dhillon: A New O(n2) Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem, Ph. D. Thesis, Univ. of California, Berkeley, (1997).

[5] I. S. Dhillon and B. N. Parlett: Orthogonal eigenvectors and relative gaps, SIAM J. Matrix Anal. Appl., Vol. 25, No. 3 (2004), pp.858–899.

[6] M. Gu and S. C. Eisenstat: *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., Vol. 16, No. 1 (1995), pp.172–191.

[7] T. Konda, M. Takata, M. Iwasaki and Y. Nakamura: *A New SVD Algorithm Suited to Parallelization and Preliminary Results*, Proceedings of the IASTED International Conference on Advances in Computer Science and Technology, (2006), pp.79–84.

[8] LAPACK, http://www.netlib.org/lapack/