Differential qd algorithm for totally nonnegative Hessenberg matrices: introduction of origin shifts and relationship with the discrete hungry Lotka-Volterra system

Yusaku Yamamoto¹ and Takeshi Fukaya²

¹ Department of Computer Science and Systems Engineering, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan

² Department of Computational Science and Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

E-mail yamamoto@cs.kobe-u.ac.jp

Received February 27, 2010, Accepted April 21, 2010

Abstract

We propose an approach for introducing the origin shift into the multiple dqd algorithm for computing the eigenvalues of a totally nonnegative matrix. Numerical experiments show that the shift speeds up the convergence while retaining the accuracy of the computed eigenvalue.

Keywords eigenvalue, totally nonnegative, Hessenberg matrix, qd algorithm, origin shift

Research Activity Group Algorithms for Matrix / Eigenvalue Problems and their Applications

1. Introduction

Let A be an $m \times m$ matrix. A is called *totally nonnegative* (TN) if all of its minors are nonnegative. TN matrices have many applications in areas such as combinatorics and statistics [1]. Recently, there is a growing interest in numerical algorithms for TN matrices, and several algorithms for eigenvalue computation, singular value decomposition and linear equation solving with TN coefficient matrices have been developed [1–3].

In [2], we proposed an algorithm for computing the eigenvalues of a totally nonnegative band matrix. The algorithm, called the multiple dqd (differential quotient-difference) algorithm, is a natural extension of the dqd algorithm for computing the eigenvalues of a symmetric positive-definite tridiagonal matrix [4]. It has two features: first, it exploits the fact that TN matrices can be represented as a product of positive bidiagonal factors and works directly on these bidiagonal factors. Second, it preserves the total nonnegativity throughout the iterations. These features enable us to show that the algorithm can compute all the eigenvalues of a TN matrix to high relative accuracy [2]. Unfortunately, due to the structure of the algorithm, it seemed difficult to introduce the origin shift to accelerate the convergence into the multiple dqd algorithm. For this reason, the order of convergence of this algorithm remained only linear.

In this paper, we consider the case where A is a TN Hessenberg matrix. For this type of matrices, we show that we can introduce the origin shift while retaining the above two features. Our preliminary numerical experiment shows that the resulting algorithm exhibits faster convergence and can compute the smallest eigenvalue to the same accuracy as the shiftless algorithm, although

theoretical proof of high relative accuracy has yet to be established. We also point out the close relationship between our shifted multiple dqd algorithm and the discrete hungry Lotka-Volterra system [5].

2. The multiple dqd algorithm

In this section, we define our target problem and review the unshifted multiple dqd algorithm.

Let L and R_i ($i = 1, \dots, M$) be $m \times m$ lower and upper bidiagonal matrices defined by

$$L = \begin{pmatrix} q_1 & & & & \\ 1 & q_2 & & & \\ & 1 & q_3 & & \\ & & \ddots & \ddots & \\ & & & 1 & q_m \end{pmatrix} \quad \text{and} \quad R_i = \begin{pmatrix} 1 & e_{i1} & & & \\ & 1 & e_{i2} & & \\ & & 1 & \ddots & \\ & & & \ddots & e_{i,m-1} \\ & & & & 1 \end{pmatrix}, \quad (1)$$

respectively, where q_k ($1 \leq k \leq m$) and e_{ik} ($1 \leq i \leq M$, $1 \leq k \leq m-1$) are some positive numbers. We consider the problem of computing the eigenvalues of a matrix defined as the product of these bidiagonal factors:

$$A = LR_1R_2 \cdots R_M. \quad (2)$$

A is a Hessenberg matrix with upper bandwidth M . Furthermore, A is totally nonnegative because it is a product of positive bidiagonal factors [1]. Then, from the general theory of the TN matrix, we know that all of the

eigenvalues of A are simple, real and positive.

The multiple dqd algorithm for A is a variant of the LR algorithm that performs the LR step solely on the bidiagonal factors. Let $M = 3$ and consider applying an LR step to A . Noting that (2) already gives the LR decomposition of A , we can compute the next iterate $\hat{A}$ by making the product $R_1 R_2 R_3 L$ and computing its LR decomposition. In the multiple dqd algorithm, we do this by repeating the LR decomposition of a product of upper and lower bidiagonal factors as follows:

$$\begin{aligned}\hat{A} &= R_1 R_2 R_3 L \\ &= R_1 R_2 L^{(1)} \hat{R}_3 \\ &= R_1 L^{(2)} \hat{R}_2 \hat{R}_3 \\ &= L^{(3)} \hat{R}_1 \hat{R}_2 \hat{R}_3 \equiv \hat{L} \hat{R}_1 \hat{R}_2 \hat{R}_3.\end{aligned}\quad (3)$$

Here, the products such as $R_1 R_2 R_3 L$ are not computed explicitly and the LR transformations such as $R_3 L = L^{(1)} \hat{R}_3$ are done with the dqd algorithm [4]. This algorithm has the following two features:

- (i) It works on the bidiagonal factors directly and never forms their products explicitly.
- (ii) Positivity of the bidiagonal factors is preserved due to the characteristics of the dqd algorithm.

By combining these features with mixed error analysis of the dqd algorithm [4] and relative perturbation theory on the eigenvalues of a TN matrix [1], one can show that the multiple dqd algorithm can compute all the eigenvalues of a TN matrix to high relative accuracy [2].

3. Introduction of the origin shift

3.1 The shifted LR algorithm for a general matrix

As a preparation for introducing the origin shift into the multiple dqd algorithm, we first explain the shifted LR algorithm. Let s be some properly chosen shift and denote the $m \times m$ identity matrix by I . Then, one step of the shifted LR algorithm can be written as follows:

$$A - sI = L^{(0)} R^{(0)}, \quad (4)$$

$$\hat{A} = R^{(0)} L^{(0)} + sI \left(= (L^{(0)})^{-1} A L^{(0)} \right). \quad (5)$$

Here, we define the LU decomposition in (4) so that the diagonals of $R^{(0)}$ are 1's, in accordance with (1).

Now, assume that A is given in the factored form as $A = LR$ and we want $\hat{A}$ also in the factored form $\hat{A} = \hat{L} \hat{R}$. Then, the above formulae can be rewritten as

$$LR - sI = L^{(0)} R^{(0)}, \quad (6)$$

$$\hat{L} \hat{R} = (L^{(0)})^{-1} L R L^{(0)}. \quad (7)$$

This shows that the computation of $\hat{L}$ and $\hat{R}$ from L and R can be done in the following two steps.

- Obtain the lower triangular factor $L^{(0)}$ from the LR decomposition of $LR - sI$.
- Apply similarity transformation by $L^{(0)}$ to LR .

Based on this observation, we introduce the origin shift into the multiple dqd algorithm in the next subsection.

3.2 Introduction of the origin shift into the multiple dqd algorithm

Assume that A is given as in (2). In considering the shifted version of the multiple dqd algorithm, it would be natural to require that the next iterate $\hat{A}$ is also given as a product of bidiagonal factors:

$$\hat{A} = \hat{L} \hat{R}_1 \hat{R}_2 \cdots \hat{R}_M, \quad (8)$$

where $\hat{L}$ is a lower bidiagonal matrix whose lower subdiagonal elements are all 1's and $\hat{R}_i$ is a unit upper bidiagonal matrix. We denote the diagonal elements of $\hat{L}$ by $\hat{q}_k$ and the upper subdiagonal elements of $\hat{R}_i$ by $\hat{e}_{ik}$.

We want to design the shifted algorithm in such a way that the conditions (i) and (ii) explained in Section 2 are satisfied. This is because then we will be able to perform relative error analysis of the algorithm as in the unshifted case. In the present subsection, we concentrate on constructing the shifted algorithm so that the condition (i) is satisfied. Positivity of the variables will be discussed in the next subsection. Also, in this subsection, we assume that breakdown of the algorithm, such as division by zero, does not occur. A sufficient condition for this will be given in the next subsection.

Inserting (2) and (8) into (6) and (7), we have

$$L R_1 \cdots R_M - sI = L^{(0)} R^{(0)}, \quad (9)$$

$$\hat{L} \hat{R}_1 \cdots \hat{R}_M = (L^{(0)})^{-1} L R_1 \cdots R_M L^{(0)}. \quad (10)$$

In (9), $L^{(0)}$ is a lower bidiagonal matrix whose subdiagonal elements are all 1's. Now, suppose that we know $L^{(0)}$. Then we can compute $\hat{L}, \hat{R}_1, \dots, \hat{R}_M$ by applying the LR transformations repeatedly to the right-hand side of (10) as follows:

$$R_M L^{(0)} = L^{(1)} \hat{R}_M, \quad (11)$$

$$R_{M-1} L^{(1)} = L^{(2)} \hat{R}_{M-1}, \quad (12)$$

$$\vdots$$

$$R_1 L^{(M-1)} = L^{(M)} \hat{R}_1, \quad (13)$$

$$L L^{(M)} = L^{(0)} \hat{L}. \quad (14)$$

Here, the last equation is obtained by rewriting the equation $\hat{L} = (L^{(0)})^{-1} L L^{(M)}$. Eqs. (11) through (14) show that once we know $L^{(0)}$, all the subsequent computations can be done by working only on the bidiagonal factors. The remaining problem is how to compute $L^{(0)}$. From (9), it seems that one needs to form $LR_1 \cdots R_M - sI$ explicitly and compute its LR decomposition. But this approach would violate the condition (i).

Fortunately, we can compute $L^{(0)}$ simultaneously with $\hat{L}, \hat{R}_1, \dots, \hat{R}_M$ by rearranging the computations. To see this, first compare the $(1, 1)$ elements of both sides of (9) to obtain

$$q_1^{(0)} = q_1 - s. \quad (15)$$

Next, by writing down (11) through (13) element by element, we have the following set of equalities:

$$\hat{e}_{M,k-1} = e_{M,k-1} \frac{q_k^{(0)}}{q_{k-1}^{(1)}} \quad (2 \leq k \leq m), \quad (16)$$

$$q_k^{(1)} = e_{M,k} + q_k^{(0)} - \hat{e}_{M,k-1} \quad (1 \leq k \leq m), \quad (17)$$

$$\hat{e}_{M-1,k-1} = e_{M-1,k-1} \frac{q_k^{(1)}}{q_{k-1}^{(2)}} \quad (2 \leq k \leq m), \quad (18)$$

$$q_k^{(2)} = e_{M-1,k} + q_k^{(1)} - \hat{e}_{M-1,k-1} \quad (1 \leq k \leq m), \quad (19)$$

$\vdots$

$$\hat{e}_{1,k-1} = e_{1,k-1} \frac{q_k^{(M-1)}}{q_{k-1}^{(M)}} \quad (2 \leq k \leq m), \quad (20)$$

$$q_k^{(M)} = e_{1,k} + q_k^{(M-1)} - \hat{e}_{1,k-1} \quad (1 \leq k \leq m), \quad (21)$$

where we adopted the convention that

$$\hat{e}_{M,0} = \hat{e}_{M-1,0} = \cdots = \hat{e}_{1,0} = 0, \quad (22)$$

$$e_{M,m} = e_{M-1,m} = \cdots = e_{1,m} = 0. \quad (23)$$

Finally, we write down (14) element-wise to obtain

$$\hat{q}_k = q_k \frac{q_k^{(M)}}{q_k^{(0)}} \quad (1 \leq k \leq m), \quad (24)$$

$$q_{k+1}^{(0)} = q_{k+1} + q_k^{(M)} - \hat{q}_k \quad (1 \leq k \leq m-1). \quad (25)$$

Eqs. (15) through (25) determine $L^{(0)}, \hat{L}, \hat{R}_1, \dots, \hat{R}_M$ completely. Indeed, starting from $q_1^{(0)}$ given by (15) and using (17), (19), (21), (24) and (25), we can compute $q_1^{(1)}, q_1^{(2)}, \dots, q_1^{(M)}, \hat{q}_1$ and $q_2^{(0)}$ in this order (Note that we need not compute $\hat{e}_{i,k-1}$ when $k = 1$). Once $q_2^{(0)}$ has been computed, we can proceed to the second round; we compute $\hat{e}_{M,1}, q_2^{(1)}, \hat{e}_{M-1,1}, q_2^{(2)}, \dots, \hat{e}_{1,1}, q_2^{(M)}, \hat{q}_2$ and $q_3^{(0)}$ in this order using (16) through (25). By continuing this process, all the elements of $\hat{L}, \hat{R}_1, \dots, \hat{R}_M$, along with the elements of $L^{(0)}$, can be computed. We call this process the *shifted multiple dqd algorithm*. Obviously, this algorithm obviates the need to form the product $LR_1R_2 \cdots R_M$ explicitly and works only on the bidiagonal factors. Thus the condition (i) is satisfied.

3.3 The condition for positivity of the variables

In the previous subsection, we derived the equations (15) through (25) of the shifted multiple dqd algorithm without asking whether the variables are positive or not. We also assumed that breakdown of the algorithm does not occur. In this subsection, we show that all the variables remain positive and breakdown does not occur if the shift satisfies a certain condition. More specifically, we have the following theorem.

Theorem 1 *Let A be a TN matrix given by (2) and denote its smallest eigenvalue by λ_m . If $s < \lambda_m$, then breakdown does not occur in one step of the shifted multiple dqd algorithm. Moreover, all the elements of $L^{(i)}$ ($0 \leq i \leq M$), $\hat{R}_i$ ($1 \leq i \leq M$) and $\hat{L}$ are positive.*

Proof We first show the positivity of $L^{(0)}$. Let $C \equiv A - sI$ and denote the $k \times k$ leading principal submatrices of A and C by $A_{1:k}^{1:k}$ and $C_{1:k}^{1:k}$, respectively. Since A is a TN matrix, its eigenvalues are all real and positive. Moreover, from the interlace inequality concerning the eigenvalues of a leading principal submatrix of a TN matrix [6], the smallest eigenvalue of $A_{1:k}^{1:k}$ is larger than or equal to the smallest eigenvalue of A . Hence all the eigen-

values of $C_{1:k}^{1:k} = A_{1:k}^{1:k} - sI_k$ are positive and accordingly $|C_{1:k}^{1:k}| > 0$ ($1 \leq k \leq m$). In this case, the left-hand side of (9) is LR-decomposable and the diagonal elements of the lower triangular factor are given as

$$q_k^{(0)} = \frac{|C_{1:k}^{1:k}|}{|C_{1:k-1}^{1:k-1}|} > 0, \quad (26)$$

where $C_{1:0}^{1:0} \equiv 1$. This shows the positivity of $L^{(0)}$.

Next, we show the positivity of $L^{(i)}$ and $\hat{R}_i$ ($1 \leq i \leq M$). First, observe that $L^{(1)}$ and $\hat{R}_M$ are computed from $L^{(0)}$ and R_M by the LR transformation (11). It is well known that the LR transformation can be done with the differential qd algorithm that uses only addition, multiplication and division. Hence if the input bidiagonal factors are positive, there is no breakdown and the output bidiagonal factors are also positive. Since both $L^{(0)}$ and R_M are positive in (11), we know that both $L^{(1)}$ and $\hat{R}_M$ are positive. By repeating this reasoning, we know that $L^{(2)}, \dots, L^{(M)}$ and $\hat{R}_{M-1}, \dots, \hat{R}_1$ are positive.

Finally, since $L, L^{(0)}$ and $L^{(M)}$ are positive, we know from (24) that $\hat{L}$ is also positive. Also, the positivity of $L^{(0)}, L^{(1)}, \dots, L^{(M)}$ guarantees the absence of breakdown, because the denominators on the right-hand sides of (16), (18), (20) and (24) are positive. (QED)

From Theorem 1, we know that the shifted multiple dqd algorithm also satisfies the condition (ii) if $s < \lambda_m$.

4. Relationship with the discrete hungry Lotka-Volterra system

The shifted multiple dqd algorithm given in the previous section has a close relationship with the discrete hungry Lotka-Volterra system. To see this, we rename the variables in (15) through (25) as follows:

$$q_k \rightarrow U_{(k-1)(M+1)+1}^{(n)}, \quad (27)$$

$$e_{M-i+1,k} \rightarrow U_{(k-1)(M+1)+i+1}^{(n)}, \quad (28)$$

$$\hat{q}_k \rightarrow U_{(k-1)(M+1)+1}^{(n+1)}, \quad (29)$$

$$\hat{e}_{M-i+1,k} \rightarrow U_{(k-1)(M+1)+i+1}^{(n+1)}, \quad (30)$$

$$q_k^{(0)} \rightarrow \bar{V}_{(k-1)(M+1)+1}^{(n)}, \quad (31)$$

$$q_k^{(i)} \rightarrow \bar{V}_{(k-1)(M+1)+i+1}^{(n)}, \quad (32)$$

$$s \rightarrow -\frac{1}{\delta^{(n)}}. \quad (33)$$

Then (16) through (21), (24) and (25) can be rewritten as follows:

$$U_{(k-2)(M+1)+2}^{(n+1)} = \frac{U_{(k-2)(M+1)+2}^{(n)} \bar{V}_{(k-1)(M+1)+1}^{(n)}}{\bar{V}_{(k-2)(M+1)+2}^{(n)}}, \quad (34)$$

$$\begin{aligned} \bar{V}_{(k-1)(M+1)+2}^{(n)} &= U_{(k-1)(M+1)+2}^{(n)} + \bar{V}_{(k-1)(M+1)+1}^{(n)} \\ &\quad - U_{(k-2)(M+1)+2}^{(n+1)}, \end{aligned} \quad (35)$$

$$U_{(k-2)(M+1)+3}^{(n+1)} = \frac{U_{(k-2)(M+1)+3}^{(n)} \bar{V}_{(k-1)(M+1)+2}^{(n)}}{\bar{V}_{(k-2)(M+1)+3}^{(n)}}, \quad (36)$$

$$\begin{aligned} \bar{V}_{(k-1)(M+1)+3}^{(n)} &= U_{(k-1)(M+1)+3}^{(n)} + \bar{V}_{(k-1)(M+1)+2}^{(n)} \\ &\quad - U_{(k-2)(M+1)+3}^{(n+1)}, \end{aligned} \quad (37)$$

$\vdots$

$$U_{(k-1)(M+1)}^{(n+1)} = \frac{U_{(k-1)(M+1)}^{(n)} \bar{V}_{(k-1)(M+1)+M}^{(n)}}{\bar{V}_{(k-1)(M+1)}^{(n)}}, \quad (38)$$

$$\bar{V}_{k(M+1)}^{(n)} = U_{k(M+1)}^{(n)} + \bar{V}_{(k-1)(M+1)+M}^{(n)} - U_{(k-1)(M+1)}^{(n+1)}, \quad (39)$$

$$U_{(k-1)(M+1)+1}^{(n+1)} = \frac{U_{(k-1)(M+1)+1}^{(n)} \bar{V}_{k(M+1)}^{(n)}}{\bar{V}_{(k-1)(M+1)+1}^{(n)}}, \quad (40)$$

$$\bar{V}_{k(M+1)+1}^{(n)} = U_{k(M+1)+1}^{(n)} + \bar{V}_{k(M+1)}^{(n)} - U_{(k-1)(M+1)+1}^{(n+1)}. \quad (41)$$

Here, the ranges of k in (34) through (41) are the same as those in (16) through (21), (24) and (25). Corresponding to (15), (22) and (23), we make the following definitions:

$$\bar{V}_1^{(n)} = U_1^{(n)} + \frac{1}{\delta^{(n)}}, \quad (42)$$

$$U_{-M+1}^{(n+1)} = \dots = U_0^{(n+1)} = 0, \quad (43)$$

$$U_{(m-1)(M+1)+2}^{(n)} = \dots = U_{m(M+1)}^{(n)} = 0. \quad (44)$$

Eqs. (34) through (41) can be written succinctly as

$$\bar{V}_k^{(n)} = U_k^{(n)} + \bar{V}_{k-1}^{(n)} - \frac{U_{k-M-1}^{(n)} \bar{V}_{k-1}^{(n)}}{\bar{V}_{k-M-1}^{(n)}}, \quad (45)$$

$$U_k^{(n+1)} = \frac{U_k^{(n)} \bar{V}_{k+M}^{(n)}}{\bar{V}_k^{(n)}}. \quad (46)$$

where $1 \leq k \leq m(M+1)$ in (45) and $1 \leq k \leq (m-1)(M+1)+1$ in (46). Moreover, (42) and (43) can be transformed into the following conditions:

$$\bar{V}_{-M}^{(n)} = \bar{V}_{-M+1}^{(n)} = \dots = \bar{V}_0^{(n)} = \frac{1}{\delta^{(n)}}, \quad (47)$$

$$U_{-M}^{(n)} = U_{-M+1}^{(n)} = \dots = U_0^{(n)} = 0. \quad (48)$$

Eqs. (45) through (48) are nothing but the discrete hungry Lotka-Volterra (dhLV) system [5], if we set $V_k^{(n)} = \delta^{(n)} \bar{V}_k^{(n)}$. However, in the context of the dhLV system, the parameter $\delta^{(n)}$ has the meaning of step size and must be positive. On the other hand, in the multiple dqd algorithm, we want to set $\delta^{(n)}$ negative (that is, set s positive) to accelerate the convergence. We showed that positivity of the bidiagonal factors is retained in this case if s is smaller than the smallest eigenvalue of A .

5. Numerical results

To confirm our analysis in Section 3, we performed a preliminary numerical experiment. We set $m = 10$ and $M = 3$ and generated e_{ik} and q_k using random numbers in $(0, 1]$. For this matrix, the smallest eigenvalue is $\lambda_m = 1.24375120694785 \times 10^{-4}$ [2]. We applied the multiple dqd algorithm with and without the origin shift to this matrix. Ideally, the shift s should be updated at each iteration. However, we fixed s to one of $0.9\lambda_m$, $0.99\lambda_m$,

Fig. 1. Convergence of $\sum_{i=1}^3 e_{i,m-1}^{(0)}$.

or $0.999\lambda_m$ here, because we have not devised an efficient shifting strategy yet. So the convergence of the shifted algorithm should still be linear, though it should be faster than that of the unshifted algorithm.

In Fig. 1, we plotted the sum of $e_{i,m-1}$ ($1 \leq i \leq 3$) as a function of the iteration number. As can be seen, this quantity decays faster as s approaches λ_m , showing the effect of the shift in accelerating the convergence.

In the shifted algorithm, e_{ik} and q_k remained positive throughout the computation. This confirms the prediction of Theorem 1. Also, the computed smallest eigenvalue agreed to the correct value to 15 decimal digits, indicating relative accuracy of the shifted algorithm.

6. Conclusion

In this paper, we showed that the origin shift can be incorporated into the multiple dqd algorithm when the target matrix is a TN Hessenberg matrix. It remains as the subject of our future research to prove the relative accuracy of the algorithm theoretically.

Acknowledgments

We are grateful to the anonymous referee, whose comments helped us to improve the quality of this paper. We also would like to thank Prof. Masashi Iwasaki, Prof. Satoshi Tsujimoto, Prof. Yoshimasa Nakamura, Ms. Akiko Fukuda and Mr. Kensuke Aishima for valuable discussion. This work is partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-aid for Scientific Research.

References

- [1] P. Koev, Accurate eigenvalues and SVDs of totally nonnegative matrices, *SIAM J. Matrix Anal. Appl.*, **27** (2005), 1–23.
- [2] Y. Yamamoto and T. Fukaya, Differential qd algorithm for totally nonnegative band matrices: convergence properties and error analysis, *JSIAM Letters*, **1** (2009), 56–59.
- [3] M. Gasca and J. M. Peña, Total positivity and Neville elimination, *Lin. Alg. Appl.*, **165** (1992), 25–44.
- [4] K. V. Fernando and B. N. Parlett, Accurate singular values and differential qd algorithms, *Numer. Math.*, **67** (1994), 191–229.
- [5] A. Fukuda, E. Ishiwata, M. Iwasaki and Y. Nakamura, The discrete hungry Lotka-Volterra system and a new algorithm for computing matrix eigenvalues, *Inverse Problems*, **25** (2009), 015007.
- [6] C. Li and R. Mathias, Interlacing inequalities for totally nonnegative matrices, *Lin. Alg. Appl.*, **341** (2002), 35–44.