Journal of Math-for-Industry, Vol. 3 (2011A-4), pp. 47–52

A note on eigenvalue computation for a tridiagonal matrix with real eigenvalues

Akiko Fukuda

Received on October 6, 2010 / Revised on February 7, 2011

Abstract. The target matrix of the dhLV algorithm is already shown to be a class of nonsymmetric band matrix with complex eigenvalues. In the case where the band width $M = 1$ in the dhLV algorithm, it is applicable to a tridiagonal matrix, with real eigenvalues, whose upper and lower subdiagonal entries are restricted to be positive and 1, respectively. In this paper, we first clarify that the dhLV algorithm is also applicable to the eigenvalue computation of nonsymmetric tridiagonal matrix with relaxing the restrictions for subdiagonal entries. We also demonstrate that the well-known packages are not always desirable for computing nonsymmetric eigenvalues with respect to numerical accuracy. Through some numerical examples, it is shown that the tridiagonal eigenvalues computed by the dhLV algorithm are to high relative accuracy.

Keywords. matrix eigenvalues, tridiagonal matrix, discrete hungry Lotka-Volterra system

1. Introduction

Several routines for nonsymmetric eigenvalues are provided in MATLAB [1] which is an interactive software for matrix-based computation, LAPACK [2] which is the famous numerical linear algebra package, and so on. The QR algorithm [3, 4] is the most standard algorithm for nonsymmetric eigenvalues, so is adopted as the LAPACK routines. However, it is not difficult to find an example such that the eigenvalues computed by LAPACK are not to high relative accuracy.

The author designs in [5] an algorithm, named the dhLV algorithm, for complex eigenvalues of a certain nonsymmetric band matrix. The dhLV algorithm is based on the integrable discrete hungry Lotka-Volterra (dhLV) system

$$(1) \qquad u_k^{(n+1)} = u_k^{(n)} \prod_{j=1}^{M} \frac{1+\delta^{(n)} u_{k+j}^{(n)}}{1+\delta^{(n+1)} u_{k-j}^{(n+1)}},$$
$$k = 1, 2, \ldots, M_m, \quad M_k := (k-1)M + k,$$
$$u_{1-j}^{(n)} \equiv 0, \quad u_{M_m+j}^{(n)} \equiv 0, \quad j = 1, 2, \ldots, M,$$

where M, m are given positive integers, and $\delta^{(n)} > 0$, $u_k^{(n)}$ denote the values of δ and u_k at the discrete step n, respectively. The dhLV system (1) is a discretized version of the continuous-time hungry Lotka-Volterra system [6], which is the prey-predator model in mathematical biology. The parameter M originally denotes the number of species which a species can prey, whereas, in the dhLV algorithm, M gives the location of positive entries in the target matrix. In the case where $M = 1$, the target matrix of the dhLV algorithm becomes a nonsymmetric tridiagonal matrix whose lower subdiagonal entries are 1 and eigenvalues are real. Additionally, in [5], the upper subdiagonal entries are required to be positive, in order to guarantee convergence of the dhLV algorithm. It is noted here that such tridiagonal matrices always have real eigenvalues.

The first purpose of this paper is to expand the applicable range of the dhLV algorithm with $M = 1$. The target matrix of the dhLV algorithm with $M = 1$ is shown to be the nonsymmetric tridiagonal matrix whose upper and lower subdiagonal entries are not restricted to be positive and 1, respectively. The second is to demonstrate that the computed eigenvalues by the dhLV algorithm are to high relative accuracy, which is not discussed precisely in [5], in the case where the eigenvalues are both real and complex, through some numerical examples.

In this paper, we first review in Section 2 the dhLV algorithm with $M = 1$ briefly, and then we clarify that it is also applicable to the eigenvalue computation of nonsymmetric tridiagonal matrix which is not a class of the target matrix in [5]. In Section 3, by some numerical examples, we observe that MATLAB and LAPACK are not always desirable with respect to the numerical accuracy of the computed eigenvalues. In the comparison with the numerical results by MATLAB and LAPACK, the computed eigenvalues by the dhLV algorithm are shown to have almost high relative accuracy. Finally, in Section 4, we give concluding remarks.

2. The dhLV algorithm for nonsymmetric tridiagonal matrix

We first explain the dhLV algorithm with $M = 1$.

Let us introduce two kinds of $2m \times 2m$ matrices,

$$L^{(n)} := \begin{pmatrix} 0 & U_1^{(n)} & & & \\ 1 & 0 & U_2^{(n)} & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & U_{2m-1}^{(n)} \\ & & & 1 & 0 \end{pmatrix}, \tag{2}$$

$$R^{(n)} := \begin{pmatrix} V_1^{(n)} & & & & \\ 0 & V_2^{(n)} & & & \\ \delta^{(n)} & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \delta^{(n)} & 0 & V_{2m}^{(n)} \end{pmatrix}, \tag{3}$$

where

$$U_k^{(n)} := u_k^{(n)}(1 + \delta^{(n)} u_{k-1}^{(n)}),$$
$$V_k^{(n)} := (1 + \delta^{(n)} u_k^{(n)})(1 + \delta^{(n)} u_{k-1}^{(n)}).$$

Then, the matrix representation for the dhLV system with $M = 1$ is given by

$$R^{(n)} L^{(n+1)} = L^{(n)} R^{(n)}. \tag{4}$$

The equality in each entry of (4) leads to the dhLV system (1) with $M = 1$. Let us assume that $u_1^{(0)} > 0, u_2^{(0)} > 0, \ldots, u_{2m-1}^{(0)} > 0$, then it is obvious from (1) that, for all n, $u_1^{(n)} > 0, u_2^{(n)} > 0, \ldots, u_{2m-1}^{(n)} > 0$. By taking account that $V_k^{(n)} > 1$, we see that $R^{(n)}$ is nonsingular. So, (4) can be transformed as $L^{(n+1)} = (R^{(n)})^{-1} L^{(n)} R^{(n)}$. This implies that the eigenvalues of $L^{(n)}$ are invariant under the time evolution from n to $n+1$ of the dhLV system (1) with $M = 1$. Hence, the matrices $L^{(0)}$ and $L^{(1)}, L^{(2)}, \ldots$ are similar to each other. For the unit matrix I and an arbitrary constant d, the matrices $L^{(0)} + dI$ and $L^{(1)} + dI, L^{(2)} + dI, \ldots$ are also similar.

The asymptotic behavior as $n \to \infty$ of the dhLV variables are given as

$$\lim_{n\to\infty} u_{2k-1}^{(n)} = c_k, \quad k = 1, 2, \ldots, m, \tag{5}$$

$$\lim_{n\to\infty} u_{2k}^{(n)} = 0, \quad k = 1, 2, \ldots, m-1, \tag{6}$$

where $c_1, c_2, \ldots, c_m$ are positive constants such that $c_1 > c_2 > \cdots > c_m$. See [5] for the proof of (5) and (6). From (5) and (6), we see that the characteristic polynomial of $L^* := \lim_{n\to\infty} L^{(n)} + dI$ coincides with that of the block diagonal matrix

$$\mathrm{diag}(L_1, L_2, \ldots, L_m),$$

where L_k is the 2×2 matrix

$$L_k = \begin{pmatrix} d & c_k \\ 1 & d \end{pmatrix}.$$

Hence, the characteristic polynomial of L^* becomes

$$\det(L^* - \lambda I) = \prod_{k=1}^{m} [(d-\lambda)^2 - c_k].$$

Consequently, $2m$ eigenvalues of $L^{(0)} + dI$ are given by

$$d \pm \sqrt{c_k}, \quad k = 1, 2, \ldots, m. \tag{7}$$

Namely, all the eigenvalues are real for nonsymmetric tridiagonal matrix $L^{(0)} + dI$. If each entry $U_k^{(0)}$ in $L^{(0)} + dI$ is given, the initial value $u_k^{(0)}$ in the dhLV system (1) with $M = 1$ is set as $U_k^{(0)}/(1 + \delta^{(n)} u_{k-1}^{(0)})$. Since, for sufficiently large N, $u_{2k-1}^{(N)}$ is an approximation of c_k, $d + \sqrt{u_{M_k}^{(n)}}$ leads to the approximation of the eigenvalues of $L^{(0)} + dI$.

We here expand the applicable range of the dhLV algorithm. Let us introduce the diagonal matrix

$$D := \mathrm{diag}(1, \alpha_1, \alpha_1\alpha_2, \ldots, (\alpha_1\alpha_2 \cdots \alpha_{2m-1})),$$

with arbitrary constants $\alpha_1, \alpha_2, \ldots, \alpha_{2m-1}$. Then the similarity transformation by D yields

$$\hat{L}^{(n)} + dI := D(L^{(n)} + dI)D^{-1}$$

$$= \begin{pmatrix} d & \hat{U}_1^{(n)} & & & \\ \alpha_1 & d & \hat{U}_2^{(n)} & & \\ & \alpha_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \hat{U}_{2m-1}^{(n)} \\ & & & \alpha_{2m-1} & d \end{pmatrix}, \tag{8}$$

where $\hat{U}_k^{(n)} = U_k^{(n)}/\alpha_k$. Obviously, the eigenvalues of $\hat{L}^{(n)} + dI$ coincide with those of $L^{(n)} + dI$. In other words, the applicable range of the dhLV algorithm with $M = 1$ is the matrix given as (8). Hence the eigenvalues of $\hat{L}^{(0)} + dI$ are given as (7), if the initial values $U_1^{(0)}, U_2^{(0)}, \ldots, U_{2m-1}^{(0)}$ are set, in accordance with $\hat{U}_1^{(0)}, \hat{U}_2^{(0)}, \ldots, \hat{U}_{2m-1}^{(0)}$ and $\alpha_1, \alpha_2, \ldots, \alpha_{2m-1}$, as

$$U_k^{(0)} = \hat{U}_k^{(0)} \alpha_k, \quad k = 1, 2, \ldots 2m-1.$$

Note that the convergence of the dhLV algorithm is guaranteed if $U_k^{(0)} > 0$, namely, $\hat{U}_k^{(0)} \alpha_k > 0$ for $k = 1, 2, \ldots, 2m-1$. The positivity $\hat{U}_k^{(0)} \alpha_k > 0$ implies that $\hat{U}_k^{(0)}$ and α_k have the same sign for each k. It is remarkable that the nonsymmetric tridiagonal matrix $\hat{L}^{(n)} + dI$ with $\hat{U}_k^{(0)} \alpha_k > 0$ has only real eigenvalues.

In Table 1, we present the dhLV algorithm for the eigenvalues of nonsymmetric tridiagonal matrix $\hat{L}^{(0)} + dI$. The lines from the 7th to the 11th are repeated until $\max_k u_{2k} \leq eps$ or $n > n_{\max}$ is satisfied for sufficiently small $eps > 0$.

3. Numerical experiments

In this section, we show some numerical results.

Table 1: The dhLV algorithm with $M = 1$.

for $k := 1, 2, \ldots, 2m-1$ **do**
  $U_k^{(0)} = \hat{U}_k^{(0)} \alpha_k$
end for
for $k := 1, 2, \ldots, 2m-1$ **do**
  $u_k^{(0)} = U_k^{(0)} / (1 + \delta^{(0)} u_{k-1}^{(0)})$
end for
for $n := 1, 2, \ldots, n_{\max}$ **do**
  for $k := 1, 2, \ldots, 2m-1$ **do**
    $u_k^{(n+1)} := u_k^{(n)} (1 + \delta^{(n)} u_{k+1}^{(n)}) / (1 + \delta^{(n+1)} u_{k-1}^{(n+1)})$
  end for
end for
for $k := 1, 2, \ldots, m$ **do**
  $\lambda_k = d \pm \sqrt{u_{2k-1}^{(n+1)}}$
end for

Totally nonnegative (TN) matrix is a class of nonsymmetric matrices whose eigenvalues are computable to high relative accuracy. Here TN matrix is a nonsymmetric matrix with real and positive eigenvalues. Koev proposes in [7] an algorithm for computing eigenvalues of a TN matrix. Watkins claims in [8] "This is the first example of a class of (mostly) nonsymmetric matrices whose eigenvalues can be determined to high relative accuracy". In other words, it is not easy to compute eigenvalues of nonsymmetric matrices to high relative accuracy, except for a TN matrix. The readers should pay attention to that the following example matrices are not TN.

Numerical experiments have been carried out on our computer with CPU: Intel (R) CPU L2400 @ 1.66GHz, RAM: 2GB. The dhLV algorithm is implemented by the compiler: Microsoft(R) C/C++ Optimizing Compiler Version 15.00.30729.01. We also use MATLAB R2009b (Version 7.9.0.529) and LAPACK-3.2.1 with complier: gcc-4.3.2. In this section, with respect to the numerical accuracy of computed eigenvalues, we compare our routine `dhLV`, which is a programming code of the dhLV algorithm, with the MATLAB routine `eig` and the LAPACK routine `dhseqr` for nonsymmetric Hessenberg matrix and `dsterf` for symmetric tridiagonal matrix. According to [1], the MATLAB routine `eig` is based on the LAPACK routine `dgeev`. In `dhLV`, we set $\delta^{(n)} = 1.0$ for $n = 0, 1, \ldots$.

Example 1. The first example is the 100×100 nonsymmetric matrix

$$T_1 = \begin{pmatrix} 0 & 1 & & & \\ \ell & 0 & 1 & & \\ & \ell & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & \ell & 0 \end{pmatrix},$$

whose eigenvalues are theoretically given by

$$2\sqrt{\ell}\cos\frac{k\pi}{101}, \quad k = 1, 2, \ldots, 100. \tag{9}$$

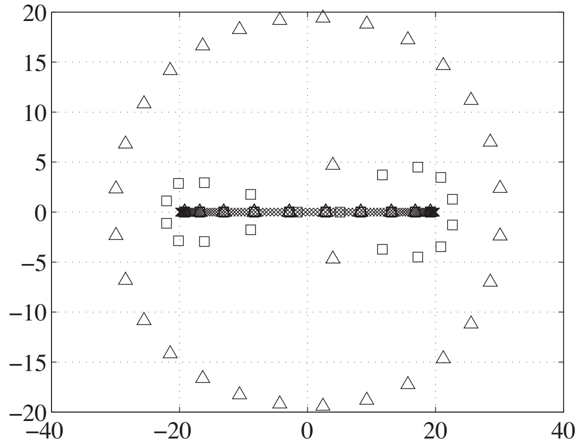

Figure 1: A graph of the real part (x-axis), and the imaginary part (y-axis) of the eigenvalues of T_1, given by (9) (plotted by $\times$), computed by `eig` (plotted by $\triangle$) and computed by `dhseqr` (plotted by $\square$).

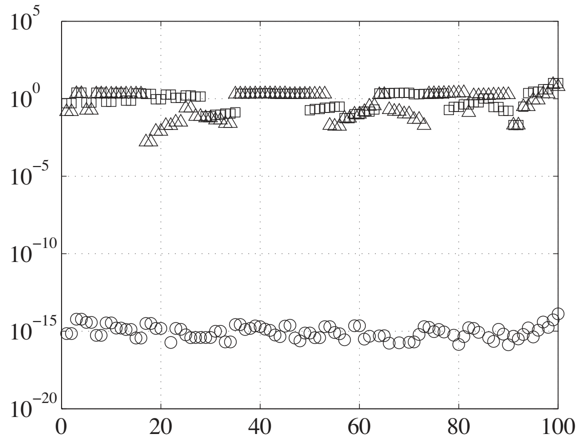

Figure 2: A graph of the index k of the computed eigenvalue $\hat{\lambda}_k$ of T_1 (x-axis), and the relative error r_k (y-axis) in `eig` (plotted by $\triangle$), `dhseqr` (plotted by $\square$) and `dhLV` (plotted by $\bigcirc$).

Figure 1 shows the eigenvalues of T_1 with $\ell = 100$, given by (9), computed by `eig`, and computed by `dhseqr` in double precision arithmetic. Obviously, all the eigenvalues of T_1 with $\ell = 100$ are real. We, however, observe that some computed eigenvalues by `eig` and `dhseqr` are not real. The routines `eig` and `dhseqr` give rise to 30 and 16 complex eigenvalues, respectively. Only in `dhLV`, all the computed eigenvalues are real. And, if the computed eigenvalues by `dhLV` are plotted by $\circ$ in Figure 1, the mark $\circ$ approximately overlaps with $\times$, given in (9). So, in Figure 2, we clarify the relative error $r_k := |\hat{\lambda}_k - \lambda_k| / |\lambda_k|$ where λ_k with $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{100}|$ and $\hat{\lambda}_k$ with $|\hat{\lambda}_1| > |\hat{\lambda}_2| > \cdots > |\hat{\lambda}_{100}|$ denote the eigenvalues given by (9) and computed by `eig`, `dhseqr` or `dhLV`, respectively. Not only Figure 1 but also Figure 2 imply that the relative errors $r_1, r_2, \ldots, r_{100}$ in `eig` and `dhseqr` are not small. Figure 2 also demonstrates that $r_1, r_2, ..., r_{100}$ in `dhLV` are much smaller than those in

Table 2: The number of irrelevant complex eigenvalues.

ℓ	10^{-10}	10^{-5}	10^{-1}	10^{0}	10^{1}	10^{5}	10^{10}
`eig`	26	22	0	0	76	20	20
`dhseqr`	26	44	14	0	92	76	52

Table 3: The average of the relative error r_{ave}.

ℓ	`eig`	`dhseqr`	`dhLV`
10^{-10}	1.18×10^{0}	1.28×10^{0}	8.60×10^{-10}
10^{-5}	1.11×10^{0}	1.13×10^{0}	4.94×10^{-13}
10^{-1}	1.48×10^{-8}	2.88×10^{0}	1.85×10^{-15}
10^{0}	6.07×10^{-16}	1.66×10^{-15}	1.40×10^{-15}
10^{1}	2.03×10^{0}	2.62×10^{0}	2.42×10^{-15}
10^{5}	1.07×10^{0}	1.39×10^{0}	1.31×10^{-15}
10^{10}	1.14×10^{0}	1.58×10^{0}	2.14×10^{-15}

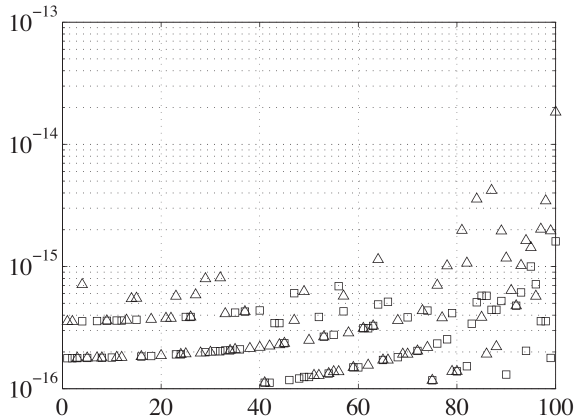

Figure 3: A graph of the index k (x-axis) of the computed eigenvalue $\hat{\lambda}_k$, and the relative error r_k (y-axis) in `eig` (plotted by $\triangle$) and `dhseqr` (plotted by $\square$).

both `eig` and `dhseqr`.

Moreover, let us consider the cases where $\ell = 10^{-10}, 10^{-5}, 10^{-1}, 10^{0}, 10^{1}, 10^{5}$ and 10^{10}. Let us recall here that T_1 with positive ℓ has only real eigenvalues. Table 2 gives the number of irrelevant complex eigenvalues by `eig` and `dhseqr`. Similarly to the case where $\ell = 100$, some computed eigenvalues by `eig` and `dhseqr` are complex in every case. Of course, all the computed eigenvalues by `dhLV` are not complex. Table 3 lists the average of relative errors $r_{\text{ave}} := \sum_{k=1}^{100} r_k/100$ in `eig`, `dhseqr` and `dhLV`. Except in the case where $\ell = 10^0$, r_{ave} in `eig` and `dhseqr` are not small. Though r_{ave} in `dhLV` tends to be larger as ℓ becomes smaller, the eigenvalues by `dhLV` have small relative errors in the comparison with those by `eig` and `dhseqr`. It is hence concluded that the eigenvalues of T_1 are computed by `dhLV` with high relative accuracy.

It is worth noting that T_1 with positive ℓ becomes the symmetric matrix by similarity transformation. For example, without changing the eigenvalues, T_1 with $\ell = 100$ is symmetrized as

$$\tilde{T}_1 := D_1 T_1 (D_1)^{-1} = \begin{pmatrix} 0 & 10 & & & \\ 10 & 0 & 10 & & \\ & 10 & \ddots & \ddots & \\ & & \ddots & \ddots & 10 \\ & & & 10 & 0 \end{pmatrix},$$

where $D_1 = \mathrm{diag}(1, \sqrt{10^{-2}}, \ldots, \sqrt{10^{-198}})$. All the computed eigenvalues by `eig` and `dsterf` are real in the case where $\tilde{T}_1$ is the target matrix. As is shown in Figure 3, the computed eigenvalues by `eig` and `dsterf` are also high relative accuracy. This is an example such that the symmetrization process is useful to improve the numerical accuracy of computed eigenvalues.

Example 2. The second example is the 100×100 nonsymmetric matrix

$$T_2 = \begin{pmatrix} 0 & 1 & & & \\ 10^8 & 0 & \ddots & & \\ & 1 & \ddots & 1 & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 0 \end{pmatrix},$$

whose two eigenvalues are ± 10000 and the others have the absolute values in $[0.03, 2.0]$. In the case where T_2 is the target matrix, the computed eigenvalues by `eig` and `dhseqr` become real without the symmetrization process. Figure 4 shows that the relative errors in `dhLV` and `eig` are almost smaller than those in `dhseqr`. On the other hand, by imposing the symmetrization for T_2 beforehand and using `dsterf` which is for symmetric tridiagonal matrices, the relative errors in `dsterf` are small, whereas those in `eig` become larger, which is shown in Figure 5.

Example 3. The third example is

$$T_3(k) = \begin{pmatrix} 0 & 1 & & & & & \\ 10^8 & 0 & \ddots & & & & \\ & \ddots & \ddots & 1 & & & \\ & & 10^8 & 0 & 1 & & \\ & & & 1 & 0 & \ddots & \\ & & & & \ddots & \ddots & 1 \\ & & & & & 1 & 0 \end{pmatrix},$$

whose matrix size is 100. The relative errors in `eig` and `dsterf` are evaluated only in the case where $T_3(k)$ is symmetrized beforehand.

First, let us consider the case where $k = 98$. Two eigenvalues of $T_3(98)$ are ± 0.14, and the others have the absolute values in $[628, 19990]$. Since the relative errors r_{97}, r_{98}, r_{99}

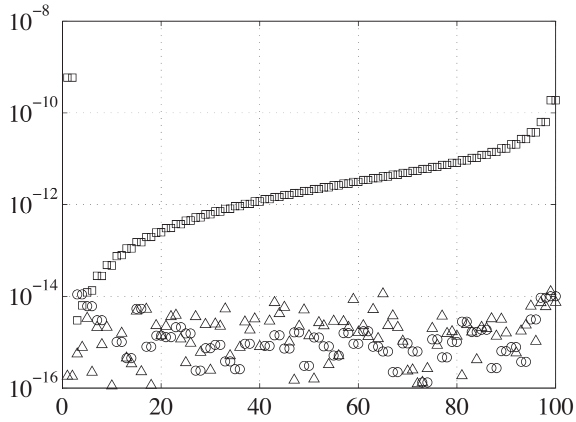

Figure 4: A graph of the index k (x-axis) of the computed eigenvalue $\hat{\lambda}_k$, and the relative error r_k (y-axis) in `eig` (plotted by $\triangle$), `dhseqr` (plotted by $\square$) and `dhLV` (plotted by $\bigcirc$).

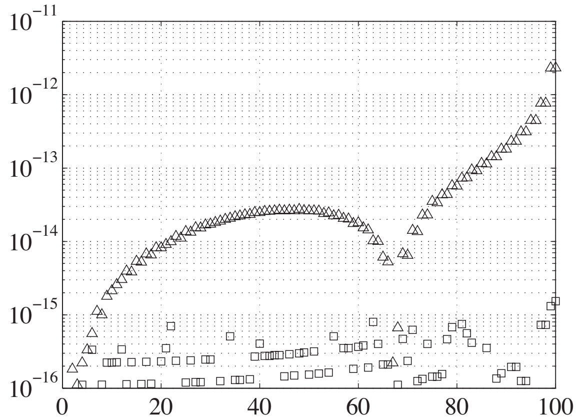

Figure 5: A graph of the index k (x-axis) of the computed eigenvalue $\hat{\lambda}_k$, and the relative error r_k (y-axis) in `eig` (plotted by $\triangle$) and `dsterf` (plotted by $\square$).

and r_{100} in `dhLV` are smaller than 1.0×10^{-16}, so they are not plotted in Figure 6. We observe from Figure 6 that $r_1, r_2, \ldots, r_{98}$ in `eig` and `dsterf` are sufficiently small, whereas r_{99} and r_{100} are not small. By comparing with Figures 2 and 4, we also see that the relative errors in `dhLV` are small similar to the case where T_1 and T_2 are the target matrices.

Next, let $k = 50$. The matrix $T_3(50)$ has the absolute values of 50 eigenvalues in $[1207, 19964]$ and the others in $[0.03, 2.0]$. Figure 7 claims that almost half of the computed eigenvalues by `eig` and `dsterf` are not high relative accuracy, even in the case of employing the symmetrization process. The routine `dhLV` brings to $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_{100}$ with $r_1, r_2, \ldots, r_{100} < 10^{-14}$.

It is numerically verified that the relative errors in `dhLV` are smaller than $O(10^{-14})$ in almost every example. It is also observed that there are some cases where the relative errors in `eig`, `dhseqr` and `dsterf` are larger than $O(10^{-12})$.

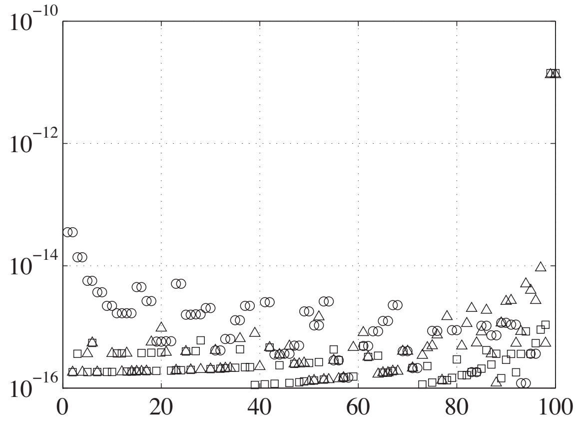

Figure 6: A graph of the index k (x-axis) of the computed eigenvalue $\hat{\lambda}_k$, and the relative error r_k (y-axis) in `eig` (plotted by $\triangle$), `dsterf` (plotted by $\square$) and `dhLV` (plotted by $\bigcirc$).

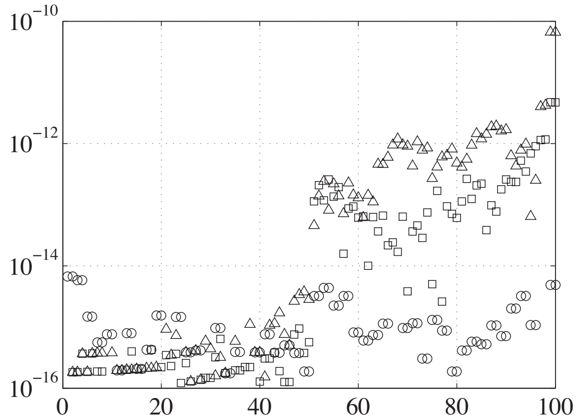

Figure 7: A graph of the index k (x-axis) of the computed eigenvalue $\hat{\lambda}_k$, and the relative error r_k (y-axis) in `eig` (plotted by $\triangle$), `dsterf` (plotted by $\square$) and `dhLV` (plotted by $\bigcirc$).

It is therefore concluded that the dhLV algorithm is better than MATLAB and LAPACK in order not to get the computed eigenvalues with larger relative errors.

4. Concluding remarks

In this paper, we first review the dhLV algorithm with $M = 1$ which is designed from the dhLV system, and then we expand the applicable class of nonsymmetric tridiagonal matrix with real eigenvalues. Even in the case where the MATLAB and the LAPACK routines are not desirable with respect to numerical accuracy of computed eigenvalues, the dhLV algorithm enables us to compute eigenvalues with high relative accuracy.

Acknowledgments

The author would like to thank Dr. E. Ishiwata, Dr. M. Iwasaki, Prof. Y. Nakamura, Prof. H. Hasegawa, Prof. G. L. G. Sleijpen and Mr. K. Yadani for many fruitful discussions and helpful advices in this work.

References

[1] The MathWorks, Inc.: MATLAB Reference Guide, Natick, MA, 1992.

[2] Anderson E., Bai Z., Bischof C., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Ostrouchov S., and Sorensen D.: LAPACK Users' Guide, 2nd ed., SIAM, Philadelphia,1995. See also LAPACK: http://www.netlib.org/lapack/.

[3] Francis J. G. F.: The QR transformation: a unitary analogue to the LR transformation I, *Comput. J.* **4** (1961/1962) 265–271.

[4] Kublanovskaya V. N.: On some algorithms for the solution of the complete eigenvalue problem, *U.S.S.R. Comput. Math. and Math. Phys.* **3** (1961) 637–657.

[5] Fukuda A., Ishiwata E., Iwasaki M., and Nakamura Y.: The discrete hungry Lotka-Volterra system and a new algorithm for computing matrix eigenvalues, *Inverse Problems* **25** (2009) 015007.

[6] Itoh Y.: Integrals of a Lotka-Volterra system of odd number of variables, *Prog. Theor. Phys.* **78** (1987) 507–510.

[7] P. Koev P.: Accurate eigenvalues and SVDs of totally nonnegative matrices, *SIAM J. Matrix Anal. Appl.* **27** (2005) 1–23.

[8] Watkins D. S.: Product eigenvalue problems, *SIAM Review* **47** (2005) 3–40.

Akiko Fukuda
Graduate School of Science, Tokyo University of Science, Tokyo 162-8601, Japan
E-mail: j1409704(at)ed.kagu.tus.ac.jp