

Algorithm for computing Jordan basis

Kenji Kudo¹, Yoshiaki Kakinuma², Kazuyuki Hiraoka³, Hiroki Hashiguchi¹, Yutaka Kuwajima¹
and Takaomi Shigehara¹

¹ Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama City, Saitama 338-8570, Japan

² NDD Corporation, 2-46-2, Hon-cho, Nakano-ku, Tokyo 164-0012, Japan

³ General Education, Wakayama National College of Technology, 77 Noshima, Nada-cho, Gobo City, Wakayama 644-0023, Japan

E-mail sigehara@mail.saitama-u.ac.jp

Received March 31, 2010, Accepted July 7, 2010

Abstract

We propose a novel algorithm to compute a Jordan basis (JB) for an arbitrarily given square matrix. The algorithm is based on the fact that a JB for a linear transformation f is obtained by extending a JB for the restriction of f to its range $R(f)$. The main ingredient of the algorithm is singular value decomposition, and that ensures backward-stability of the algorithm. To enhance the practical utility, we also introduce an automatic mechanism into the algorithm such that it outputs all possible Jordan structures close to the exact one of the input matrix.

Keywords Jordan canonical form, Jordan basis, recursive algorithm

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

1. Introduction

Numerical algorithms to reveal the Jordan structure for a given square matrix have been constructed mainly along three lines; *unitary deflation* [1–4], *matrix powering* [2, 5], and *resolvent analysis* [6, 7]. Unitary deflation relies on the fact in the first step that any square matrix can be reduced to a block triangular form by successive applications of unitary transformations. This procedure is enough to determine the *Jordan canonical form* (JCF) of the input matrix. However, a further process including nonunitary transformations is required to construct a *Jordan basis* (JB) for the matrix. It should be stressed that, even if such a series of unitary and nonunitary transformations are established, it is inevitable to multiply all the transformation matrices in order to obtain a JB concretely. The purpose of this paper is to propose a novel numerical algorithm for revealing Jordan structure including JB for an arbitrarily given square matrix. In the proposed algorithm, only unitary deflation is required to obtain a JB as well as the JCF for the input matrix. The main tool of the algorithm is singular value decomposition (SVD), which guarantees the backward stability of the algorithm. The existence of tiny singular values may disturb the numerical stability. Therefore we introduce a mechanism into the algorithm such that, for an input square matrix, all possible solutions are automatically output. In other words, the algorithm outputs all possible Jordan structures *close to* the exact one of the input matrix, together with the information on numerical error associated with each Jordan structure.

2. Theoretical aspects

Let V be a finite-dimensional linear space over \mathbf{C} and f a linear transformation on V . The kernel and the range

of f are denoted by $N(f)$ and $R(f)$, respectively. Let μ be a complex constant. An ordered sequence (x_1, \dots, x_l) of l vectors ($l \geq 1$) of V with the property

$$(f - \mu)(x_k) = x_{k-1} \quad (k = 1, \dots, l), \quad x_0 \equiv 0 \quad (1)$$

is called a *Jordan sequence* (JS) of length l associated with the eigenvalue μ of f . The set of JSs associated with μ is denoted by J_μ . A JS in J_μ ($\mu \neq 0$) is called *regular*, while a JS in J_0 is called *singular*. A set of JSs such that the vectors in the JSs compose a basis of V is called a *Jordan basis* (JB) for f .

The following theorem gives the theoretical foundation of this paper. Although the assertion is the same as in [8], the proof is rather simplified. The restriction of f to $R(f)$ defines a linear transformation on $R(f)$, that is denoted by f' in the following.

Theorem 1 Let $\mu_j \neq 0$ ($j = 1, \dots, n_r$) be the nonzero constants. If the JSs

$$\begin{aligned} (x_{j;1}^{(r)}, \dots, x_{j;l_j^{(r)}}^{(r)}) &\in J_{\mu_j} \quad (j = 1, \dots, n_r), \\ (x_{j;1}^{(s)}, \dots, x_{j;l_j^{(s)}}^{(s)}) &\in J_0 \quad (j = 1, \dots, n'_s) \end{aligned}$$

are a JB for f' , then f has a JB such that

$$\begin{aligned} (x_{j;1}^{(r)}, \dots, x_{j;l_j^{(r)}}^{(r)}) &\in J_{\mu_j} \quad (j = 1, \dots, n_r), \\ (x_{j;1}^{(s)}, \dots, x_{j;l_j^{(s)}}^{(s)}, x_{j;l_j^{(s)}+1}^{(s)}) &\in J_0 \quad (j = 1, \dots, n'_s), \quad (2) \\ (x_{n'_s+j;1}^{(s)}) &\in J_0 \quad (j = 1, \dots, n_s), \end{aligned}$$

where the vectors $x_{n'_s+1;1}^{(s)}, \dots, x_{n'_s+n_s;1}^{(s)}$ are a basis of a complementary space of $N(f) \cap R(f)$ in $N(f)$.

Proof Since $x_{j;l_j^{(s)}}^{(s)} \in R(f)$, there exists $x_{j;l_j^{(s)}+1}^{(s)}$ such

that $x_{j;l_j^{(s)}}^{(s)} = f(x_{j;l_j^{(s)}+1}^{(s)})$. Hence there exist the middle JSs associated with $\mu = 0$ in (2). The images of the sequences

$$\begin{aligned} (x_{j;1}^{(r)}, \dots, x_{j;l_j^{(r)}}^{(r)}) \quad (j = 1, \dots, n_r), \\ (x_{j;2}^{(s)}, \dots, x_{j;l_j^{(s)}+1}^{(s)}) \quad (j = 1, \dots, n'_s) \end{aligned} \quad (3)$$

by f are

$$\begin{aligned} (\mu_j x_{j;1}^{(r)}, \mu_j x_{j;2}^{(r)} + x_{j;1}^{(r)}, \dots, \mu_j x_{j;l_j^{(r)}}^{(r)} + x_{j;l_j^{(r)}-1}^{(r)}) \\ (j = 1, \dots, n_r), \\ (x_{j;1}^{(s)}, \dots, x_{j;l_j^{(s)}}^{(s)}) \quad (j = 1, \dots, n'_s), \end{aligned} \quad (4)$$

respectively. Since $\mu_j \neq 0$, the vectors in the sequences in (4) are a basis of $R(f)$ by assumption. Thus, we conclude that the vectors in the sequences in (3) are a basis of a complementary space of $N(f)$ in V . The remaining vectors $x_{j;1}^{(s)}$ ($j = 1, \dots, n'_s$) are a basis of $N(f) \cap R(f)$. Hence, by adding the lower JSs in (2), we obtain a JB for f in (2). (QED)

3. Proposal of algorithm

3.1 Framework

The constructive proof in the previous section makes it possible to establish a recursive algorithm for computing a JB for the restriction of f to the generalized eigenspace associated with the eigenvalue zero of f . The regular JSs associated with the eigenvalue $\mu \neq 0$ of f correspond to the singular JSs of $f - \mu$, which can be obtained by replacing f by $f - \mu$ in Theorem 1. Thus we are led to an algorithm for computing a JB for the linear transformation f . Here we assume that the distinct eigenvalues μ_1, \dots, μ_m of f are separately computed in advance.

JB algorithm

input: linear transformation $f : V \rightarrow V$, and all the distinct eigenvalues μ_1, \dots, μ_m of f .

For each eigenvalue μ_i ($i = 1, \dots, m$), repeat 1)–4).

- 1) Set $f_i^{(1)} = f - \mu_i$ and $V_i^{(1)} = V$.
- 2) For $k = 1, \dots, t_i$, find the restriction $f_i^{(k+1)} : V_i^{(k+1)} \rightarrow V_i^{(k+1)}$ of $f_i^{(k)}$ to $V_i^{(k+1)} \equiv R(f_i^{(k)})$, where t_i ($t_i \geq 1$) is the minimum integer such that $f_i^{(t_i+1)}$ is bijective.
- 3) Set $q_{i;t_i} = \dim V_i^{(t_i)} - \dim V_i^{(t_i+1)}$. Define

$$S(f_i^{(t_i)}) \equiv \{(x_j) \mid j = 1, \dots, q_{i;t_i}\}$$

with a basis $x_1, \dots, x_{q_{i;t_i}}$ of $N(f_i^{(t_i)})$. Set $p = q_{i;t_i}$.

- 4) For $k = t_i - 1, \dots, 1$, repeat a)–c).

- a) For each $s_j = (x_{j;1}, \dots, x_{j;l_j}) \in S(f_i^{(k+1)})$ ($j = 1, \dots, p$), solve the linear system

$$f_i^{(k)}(x_{j;l_j+1}) = x_{j;l_j}$$

and set $\text{ext}(s_j) \equiv (x_{j;1}, \dots, x_{j;l_j}, x_{j;l_j+1})$. Define

$$S_1(f_i^{(k)}) \equiv \{\text{ext}(s_j) \mid s_j \in S(f_i^{(k+1)}), j = 1, \dots, p\}.$$

- b) Set $q_{i;k} = \dim V_i^{(k)} - \dim V_i^{(k+1)} - p$. Define

$$S_2(f_i^{(k)}) \equiv \{(x_j) \mid j = 1, \dots, q_{i;k}\}$$

with a basis $x_1, \dots, x_{q_{i;k}}$ of a complementary space of $N(f_i^{(k)}) \cap R(f_i^{(k)})$ in $N(f_i^{(k)})$.

- c) Define $S(f_i^{(k)}) \equiv S_1(f_i^{(k)}) \cup S_2(f_i^{(k)})$. Set $p = p + q_{i;k}$.

output: $S(f_1^{(1)}) \cup \dots \cup S(f_m^{(1)})$.

Note that for each $i = 1, \dots, m$, $V_i^{(k)} = R(f_i^{k-1})$ and hence $f_i^{(k)}$ is just the restriction of $f_i^{(1)} = f - \mu_i$ to $R(f_i^{k-1})$ ($k = 1, \dots, t_i + 1$), where $R(f_i^0) \equiv V$.

The output of the JB algorithm gives a JB for f . In particular, for each $i = 1, \dots, m$, the JSs in $S(f_i^{(1)})$ gives a JB for the restriction of f to the generalized eigenspace $G(\mu_i)$ associated with the eigenvalue μ_i of f . The value of $q_{i;k}$ in the algorithm means the number of JSs of length k ($k = 1, \dots, t_i$) associated with the eigenvalue μ_i , and hence $\dim G(\mu_i) = \sum_{k=1}^{t_i} (k q_{i;k})$ ($i = 1, \dots, m$).

3.2 Matrix representation

Denote the set of $l_1 \times l_2$ complex matrices by $\mathbf{C}^{l_1 \times l_2}$ in general. Let $F \in \mathbf{C}^{n \times n}$ be a singular matrix of rank $r = \dim R(F) < n$. Consider the matrix representation $F' \in \mathbf{C}^{r \times r}$ of the restriction of F to $R(F)$. First we decompose F to the form

$$F = IS. \quad (5)$$

Here $I \in \mathbf{C}^{n \times r}$ is injective, while $S \in \mathbf{C}^{r \times n}$ is surjective. The decomposition in (5) is indeed possible. In the present implementation, we set $I = U$ and $S = DV^*$. Here

$$F = UDV^* \quad (6)$$

is the singular value decomposition (SVD) of F , where

$$\begin{aligned} D &= \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbf{C}^{r \times r}, \\ U &= (\mathbf{u}_1, \dots, \mathbf{u}_r) \in \mathbf{C}^{n \times r}, \end{aligned} \quad (7)$$

$$V = (\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathbf{C}^{n \times r}$$

with the singular values $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$ and the corresponding left and right (orthonormal) singular vectors \mathbf{u}_j and \mathbf{v}_j ($j = 1, \dots, n$). The column vectors of I compose a basis of $R(F)$ and, with respect to this basis, the matrix representation F' of the restriction of F to $R(F)$ should satisfy $FI = IF'$. Hence $I(F' - SI) = O$. Since I is injective, we conclude

$$F' = SI \in \mathbf{C}^{r \times r}.$$

This procedure to construct F' from F is essentially similar to the procedure to obtain $B_{11}^{(2)}$ from $B^{(1)}$ by a unitarily-equivalent transformation in [2, Section 10], and it is used in step 2) in the JB algorithm.

Suppose that we have a singular JS $(\mathbf{x}'_1, \dots, \mathbf{x}'_l)$ for F' . Then, together with $\mathbf{x}_k \equiv I\mathbf{x}'_k$ ($k = 1, \dots, l$) as well as a solution \mathbf{x}_{l+1} of the linear system

$$F\mathbf{x}_{l+1} = \mathbf{x}_l, \quad (8)$$

the sequence $(\mathbf{x}_1, \dots, \mathbf{x}_l, \mathbf{x}_{l+1})$ is a singular JS for F . Since we already have the SVD of F in (6), a solution of (8) is given by using the Moore-Penrose inverse:

$$\mathbf{x}_{l+1} = VD^{-1}U^* \mathbf{x}_l = VD^{-1} \mathbf{x}'_l.$$

This is used in step 4)-a) in the JB algorithm.

To keep numerical accuracy in numerics, we have to remove tiny singular values in the SVD of F . This is carried out by introducing a small cut-off parameter ε : If $\sigma_{r'} \geq \sigma_1 \varepsilon > \sigma_{r'+1} \geq \dots \geq \sigma_r > 0$, then set $\sigma_{r'+1} = \dots = \sigma_r = 0$ in (7). The parameter r' corresponds to the *numerical δ -rank* of F with $\delta = \sigma_1 \varepsilon$ [9], and we numerically regard $\mathbf{u}_1, \dots, \mathbf{u}_{r'}$ as a basis of $R(F)$ while $\mathbf{v}_{r'+1}, \dots, \mathbf{v}_n$ are a basis of $N(F)$.

We may choose different values of the cut-off parameter, according to each eigenvalue as well as each recursive step associated with the eigenvalue. However, in general cases, it is hard to know appropriate magnitudes of the parameters in advance. Therefore we introduce an automatic mechanism into the algorithm which makes it possible to compute JB's in both cases whether each tiny singular value is discarded or not.

With these notices, we are led to a possible matrix representation of the JB algorithm. Here $E \in \mathbf{C}^{n \times n}$ is the identity matrix. For an input matrix $F \in \mathbf{C}^{n \times n}$, the following algorithm outputs all the possible JB's of the restriction of F to $G(\mu)$ for an assigned eigenvalue μ of F , by making a thorough investigation of the cut-off parameters within a range $[\varepsilon_{\min} \sigma_1, \varepsilon_{\max} \sigma_1]$. Here σ_1 is the maximum singular value of SVD under the consideration, and the two parameters $\varepsilon_{\min}, \varepsilon_{\max}$ ($\varepsilon_{\min} < \varepsilon_{\max}$) are chosen such that the range substantially covers the tiny singular values.

JB algorithm (matrix version)

input: square matrix $F \in \mathbf{C}^{n \times n}$, an eigenvalue μ of F , and the parameters $\varepsilon_{\min}, \varepsilon_{\max}$.

Set $\mathbf{S}(\mu) = \emptyset$.

Set $F_{n_1}^{(1)} = F - \mu E$ with $n_1 = n$.

Call `cal_JB_of_GE`($F_{n_1}^{(1)}, \varepsilon_{\max}, \varepsilon_{\min}, \mathbf{S}(\mu)$).

output: $\mathbf{S}(\mu)$.

proc cal_JB_of_GE($F_{n_k}^{(k)}, \varepsilon_{\max}, \varepsilon_{\min}, \mathbf{S}(\mu)$)

Compute the singular pairs $(\sigma_j^{(k)}, \mathbf{u}_j^{(k)}, \mathbf{v}_j^{(k)})$ ($j = 1, \dots, n_k$) of $F_{n_k}^{(k)}$, where

$$\begin{aligned} \sigma_1^{(k)} &\geq \dots \geq \sigma_{r_{\max}^{(k)}}^{(k)} \geq \varepsilon_{\max} \sigma_1^{(k)} > \dots \geq \sigma_{r_{\min}^{(k)}}^{(k)} \\ &\geq \varepsilon_{\min} \sigma_1^{(k)} > \dots \geq \sigma_{n_k}^{(k)} \geq 0. \end{aligned}$$

Repeat the following procedure for $r'_k = r_{\min}^{(k)}, \dots, r_{\max}^{(k)}$: If $r'_k < n_k$, then perform a1)–a3).

a1) Define $F_{r'_k}^{(k+1)} \equiv D_{r'_k}^{(k)} V_{r'_k}^{(k)*} U_{r'_k}^{(k)} \in \mathbf{C}^{r'_k \times r'_k}$ with

$$D_{r'_k}^{(k)} = \text{diag}(\sigma_1^{(k)}, \dots, \sigma_{r'_k}^{(k)}) \in \mathbf{C}^{r'_k \times r'_k},$$

$$U_{r'_k}^{(k)} = (\mathbf{u}_1^{(k)}, \dots, \mathbf{u}_{r'_k}^{(k)}) \in \mathbf{C}^{n_k \times r'_k},$$

$$V_{r'_k}^{(k)} = (\mathbf{v}_1^{(k)}, \dots, \mathbf{v}_{r'_k}^{(k)}) \in \mathbf{C}^{n_k \times r'_k}.$$

a2) Set $N(F_{r'_k}^{(k+1)}) = \text{span}(\mathbf{v}_{r'_k+1}^{(k)}, \dots, \mathbf{v}_{n_k}^{(k)})$.

a3) Call `cal_JB_of_GE`($F_{n_{k+1}}^{(k+1)}, \varepsilon_{\max}, \varepsilon_{\min}, \mathbf{S}(\mu)$) with $n_{k+1} = r'_k$.

Otherwise, if $r'_k = n_k$, then perform b1)–b4).

b1) Set $t = k - 1$.

Table 1. Experimental environment.

CPU	Intel Core(TM)2 Duo 2.66GHz
Memory	1.99GB
OS	Windows XP SP3
Compiler	cygwin gcc version 3.4.4
LAPACK	version 3.0.8

b2) Set $q_t = n_t - n_{t+1}$. Define

$$S(F_{n_t}^{(t)}) \equiv \{(\mathbf{v}_{n_{t+1}+1}^{(t)}, \dots, \mathbf{v}_{n_t}^{(t)})\}.$$

Set $p = q_t$.

b3) For $k = t - 1, \dots, 1$, repeat i)–iii).

i) For each $s'_j = (\mathbf{x}'_{j;1}, \dots, \mathbf{x}'_{j;l_j}) \in S(F_{n_{k+1}}^{(k+1)})$ ($j = 1, \dots, p$), set $\text{ext}(s'_j) = (\mathbf{x}_{j;1}, \dots, \mathbf{x}_{j;l_j}, \mathbf{x}_{j;l_j+1})$ with

$$\mathbf{x}_{j;z} = U_{n_{k+1}}^{(k)} \mathbf{x}'_{j;z} \quad (z = 1, \dots, l_j),$$

$$\mathbf{x}_{j;l_j+1} = V_{n_{k+1}}^{(k)} D_{n_{k+1}}^{(k)-1} \mathbf{x}'_{j;l_j}.$$

Define $S_1(F_{n_k}^{(k)}) \equiv \{\text{ext}(s'_j) \mid s'_j \in S(F_{n_{k+1}}^{(k+1)}), j = 1, \dots, p\}$.

ii) Set $q_k = n_k - n_{k+1} - p$. Extend a basis $\{\mathbf{x}_{j;1} \mid (\mathbf{x}_{j;1}, \dots) \in S_1(F_{n_k}^{(k)}), j = 1, \dots, p\}$ of $N(F_{n_k}^{(k)}) \cap R(F_{n_k}^{(k)})$ to a basis of $N(F_{n_k}^{(k)})$ by appending $\mathbf{x}_1, \dots, \mathbf{x}_{q_k}$. Define

$$S_2(F_{n_k}^{(k)}) \equiv \{(\mathbf{x}_j) \mid j = 1, \dots, q_k\}.$$

iii) Define $S(F_{n_k}^{(k)}) \equiv S_1(F_{n_k}^{(k)}) \cup S_2(F_{n_k}^{(k)})$. Set $p = p + q_k$.

b4) $\mathbf{S}(\mu) = \mathbf{S}(\mu) \cup \{S(F_{n_1}^{(1)})\}$.

Note that the multiplicity of the input eigenvalue μ is not an input of the algorithm. For each μ , (in general more than one) generalized eigenspaces $G(\mu)$ determined from the JB's in the output $\mathbf{S}(\mu)$ might have different dimensions. Therefore one should select such combinations of JB's for $G(\mu_i)$ from $\mathbf{S}(\mu_i)$ ($i = 1, \dots, m$) that $\sum_{i=1}^m \dim G(\mu_i) = n$.

4. Numerical experiment

Numerical environment is summarized in Table 1. We use ZGESVD routine in LAPACK for SVD. Let $J_l(\mu)$ be the Jordan cell of size l associated with the eigenvalue μ . Numerical test is performed by using 100 matrices with a form

$$F = PJP^{-1}, \quad (9)$$

where

$$J = \bigoplus_{i=1}^3 \left(\bigoplus_{j_i=1}^{n_i} J_{l_{j_i}}(\mu_i) \right) \quad (10)$$

is a JCF with the eigenvalues $(\mu_1, \mu_2, \mu_3) = (1, 1 + \alpha, 10)$ and P is an invertible matrix with uniform random numbers in the range $[-1, 1]$ for elements. In (10), $n_i \in [1, 3]$ ($i = 1, \dots, 3$), $l_{j_i} \in [1, 3]$ ($j_i = 1, \dots, n_i$; $i = 1, \dots, 3$) are random integers. The JCF as well as a JB for F in (9) is obvious by construction, and a comparison with numerical results is easily made.

We first compute the eigenvalues of F in (9) by using ZGEEV routine in LAPACK. This provides n distinct eigenvalues μ_j ($j = 1, \dots, n$) in general, where n is matrix size of F . To recover the Jordan structure of F , we group neighboring eigenvalues together as follows:

- 1) Set $\Lambda = \{\mu_1, \dots, \mu_n\}$. Set $i = 1$.
- 2) Repeat i)–iii) until Λ becomes the empty set.
 - i) Let μ be the eigenvalue with the maximal absolute value in Λ . Define

$$\Lambda_i \equiv \left\{ \mu_j \in \Lambda \mid \left| \frac{\mu_j - \mu}{\mu} \right| < 10^{-4} \right\}.$$

- ii) Define μ'_i by the average of the eigenvalues in Λ_i .

- iii) Set $\Lambda = \Lambda - \Lambda_i$. Set $i = i + 1$.

We use the output μ'_1, μ'_2, \dots as the input eigenvalues for the JB algorithm.

Let J' and P' be the JCF and a JB, numerically determined from the output of the JB algorithm. Let $W_{1;2}$ be the direct sum of the generalized eigenspaces associated with the eigenvalues $\mu_1 = 1$ and $\mu_2 = 1 + \alpha$, while let W_3 be the generalized eigenspace associated with the eigenvalue $\mu_3 = 10$. Both of $W_{1;2}$ and W_3 are determined from P in (9). Numerical counterparts of $W_{1;2}$ and W_3 are denoted by $W'_{1;2}$ and W'_3 respectively, that are determined from the output of the JB algorithm. We estimate two kinds of numerical errors:

$$E_1 = \frac{\|FP' - P'J'\|_\infty}{\|FP'\|_\infty},$$

$$E_2 = \max\{\sin \theta_{W_{1;2}, W'_{1;2}}, \sin \theta_{W_3, W'_3}\},$$

where θ_{V_1, V_2} is the largest canonical angle between the subspaces V_1 and V_2 in general [10]. E_1 measures whether the numerical JSs indeed satisfy the relation in (1) or not, while E_2 measures whether the numerical JB indeed spans the input generalized eigenspaces or not.

Fig. 1 shows the results for a well-conditioned case of $\alpha = 1$. In this case, we set $\varepsilon_{\min} = \varepsilon_{\max} = 10^{-11}$. The bar and broken line graphs show E_1 and E_2 , respectively, for 100 examples. Since $\varepsilon_{\min} = \varepsilon_{\max}$, the JB algorithm outputs a single JB for each example. From Fig. 1, we confirm that, even with the unsophisticated cut-off parameters, the generalized eigenspaces as well as their JBs are computed with high numerical accuracy.

Fig. 2 shows the results for an ill-conditioned case of $\alpha = 10^{-8}$. Here we set $\varepsilon_{\min} = \varepsilon_{\max} = 10^{-5}$. In this case, owing to the lazy selection of the cut-off parameters, the JB algorithm fails to reproduce the input generalized eigenspaces. Indeed, we observe $E_2 > 10^{-2}$ in 43 examples among all. (Note that $E_2 = 10^{-2}$ means that the largest canonical angle is 0.573 degree.) To remedy this, we set $\varepsilon_{\min} = 10^{-18}$ and $\varepsilon_{\max} = 10^{-6}$ in the JB algorithm. Fig. 3 shows the results for this case. In this case, the JB algorithm outputs a number of JBs for each example in general, and in Fig. 3, we show the result with the JB which minimizes E_2 for each. We confirm from Fig. 3 that the overlap between the input and output generalized eigenspaces is drastically improved. Indeed, we attain $E_2 < 10^{-2}$ in 98 examples of all.

In several examples in Fig. 3, we find more than one JBs that satisfy $E_1 \leq 10^{-12}$ as well as $E_2 \leq 10^{-3}$. This

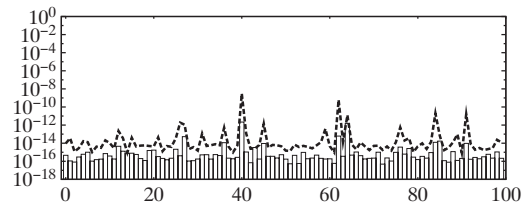


Fig. 1. E_1 (bar graph) and E_2 (broken line) for 100 examples in case of $\alpha = 1$. Here we set $\varepsilon_{\min} = \varepsilon_{\max} = 10^{-11}$.

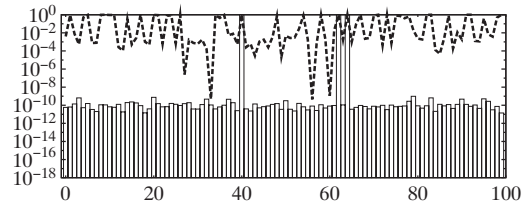


Fig. 2. E_1 (bar graph) and E_2 (broken line) for 100 examples in case of $\alpha = 10^{-8}$. Here we set $\varepsilon_{\min} = \varepsilon_{\max} = 10^{-5}$.

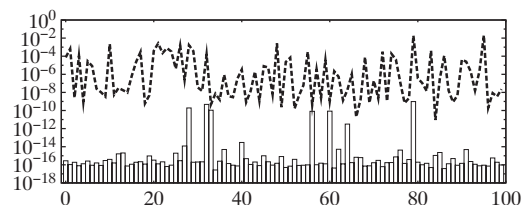


Fig. 3. E_1 (bar graph) and E_2 (broken line) for 100 examples in case of $\alpha = 10^{-8}$. Here we set $\varepsilon_{\min} = 10^{-18}$ and $\varepsilon_{\max} = 10^{-6}$.

indicates that there exist a number of Jordan structures close to the input matrix F , and the JB algorithm succeeds in reproducing them.

Acknowledgments

We are grateful to the anonymous referee for helpful comments, which served to improve the quality of this paper. This work was partially supported by Grant-in-Aid for Scientific Research (C) No.19560058.

References

- [1] V. N. Kublanovskaya, On a method of solving the complete eigenvalue problem for a degenerate matrix, USSR Comput. Math. Math. Phys., **6** (1966) 1–14.
- [2] G. H. Golub and J. H. Wilkinson, Ill-conditioned eigensystems and the computation of the Jordan canonical form, SIAM Rev., **18** (1976) 578–619.
- [3] B. Kågström and A. Ruhe, An algorithm for numerical computation of the Jordan normal form of a complex matrix, ACM Trans. Math. Software, **6** (1980) 398–419.
- [4] D. S. Watkins, The Matrix Eigenvalue Problem, SIAM, Philadelphia, 2007.
- [5] G. Ohtake, M. Koga and M. Sampei, A method for numerical computation of Jordan canonical form of matrix (in Japanese), Trans. ISCIE, **15** (2002) 320–326.
- [6] T. Suzuki and T. Suzuki, An eigenvalue problem for derogatory matrices, J. Comput. Appl. Math., **199** (2007) 245–250.
- [7] T. Suzuki and T. Suzuki, Computing the Jordan canonical form in the finite precision arithmetic, in: Proc. of SCAN 2006, p.39, 2006.
- [8] J. I. Hall, Another elementary approach to the Jordan form, Amer. Math. Monthly, **98** (1991) 336–340.
- [9] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.
- [10] G. W. Stewart, Matrix Algorithms, Vol. II: Eigensystems, SIAM, Philadelphia, 2001.