ON NUMERICAL METHODS FOR ROBUST POLE ASSIGNMENT IN CONTROL SYSTEM DESIGN*1)

SUN JI-GUANG (孙继广)

(Computing Center, Academia Sinica, Beijing, China)

Abstract

It is known[3-5] that selection of a well-conditioned set of vectors from given subspaces is the key step for solving the robust pole assignment problem. In this paper we suggest two numerical methods for selecting such set of vectors. The numerical methods, Method (I) and Method (II), are described, and some numerical results are presented.

§1. An Inverse Eigenvalue Problem

The robust pole assignment problem in control system design may be formulated as follows (see [2]—[5]):

Problem RPA. Given a real $n\times n$ matrix A, a real full rank $n\times m$ matrix B $(m<n)$ and a set $\mathscr{L}$ of n complex numbers $\lambda_1, \lambda_2, \cdots, \lambda_n$, closed under complex conjugation, find a real $m\times n$ matrix F and a non-singular $n\times n$ matrix X satisfying

$$(A+BF)X=X\Lambda, \tag{1.1}$$

where $\Lambda=\operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, such that the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ of $A+BF$ are as insensitive to perturbations in the matrix $A+BF$ as possible.

Clearly, problem RPA is an inverse algebraic eigenvalue problem.

It has been established by Wonham (1967) that there exists a matrix F such that the eigenvalues of $A+BF$ are $\lambda_1, \lambda_2, \cdots, \lambda_n$ if and only if the pair (A, B) is controllable. Hence, we assume in this paper that the pair (A, B) is controllable, i.e., for every complex number μ the only vector x satisfying

$$x^TA=\mu x^T, \quad x^TB=0$$

is the zero vector.

J. Kautsky, N. K. Nichols, P. Van Dooren and L. Fletcher[3-5] have described algorithms for computing solutions to problem RPA. The procedures all consist of three basic steps:

Step A. Compute the decomposition

$$B=(U_0^{(B)}, U_1^{(B)})\begin{pmatrix} Z \\ 0 \end{pmatrix}, \tag{1.2}$$

where $(U_0^{(B)}, U_1^{(B)})$ is a real orthogonal matrix and Z non-singular; construct an orthogonal basis, comprised by the columns of matrix S_j for the space

$$\mathscr{S}_j\equiv\mathscr{N}(U_1^{(B)T}(A-\lambda_jI))$$

* Received February 25, 1986.

1) Projects supported by the Science Fund of the Chinese Academy of Sciences.

for λ_j, $j=1, 2, \cdots, n$, where $\mathcal{N}(\cdot)$ denotes the null space.

Step X. Select vectors $x_j = S_j w_j \in \mathcal{S}_j$, $j=1, 2, \cdots, n$ such that $X=(x_1, x_2, \cdots, x_n)$ is well-conditioned.

Step F. Find the matrix $M=A+BF$ by solving $MX=X\Lambda$ and compute F explicitly from $F=Z^{-1}U_0^{(B)T}(M-A)$.

The key step is Step X. In [3] four methods for accomplishing Step X are described. The methods are all iterative and all aim to minimize a different measure of the conditioning of matrix X.

The aim of this paper is to suggest two methods for accomplishing Step X. In the next section we investigate measures of robustness of the eigenproblem (1.1). In Section 3 and Section 4 we describe two methods for selecting a well-conditioned set of vectors from given subspaces. Numerical results are given in Section 5.

For simplicity we consider in this paper only the case where the eigenvectors are required to be real.

Notation. The symbol $\mathbb{R}^{m\times n}$ denotes the set of real $m\times n$ matrices and $\mathbb{R}^n=\mathbb{R}^{n\times 1}$. $I^{(n)}$ is the $n\times n$ identity matrix, and O is the null matrix. For a real symmetric matrix A, $A>0$ $(A\geqslant 0)$ denotes that A is positive definite (positive semi-definite). The superscript T is for transpose. $A^{\dagger}$ stands for the Moore-Penrose generalized inverse of a matrix A. $\mathcal{R}(A)$ is the column space of A. $\|\ \|_2$ denotes the usual Euclidean vector norm and spectral norm, and $\|\ \|_F$ denotes the Frobenius matrix norm.

§ 2. Measures of Robustness

Let

$$X=(x_1, x_2, \cdots, x_n), \quad Y=X^{-T}=(y_1, y_2, \cdots, y_n). \tag{2.1}$$

It is well known (see [10]) that the sensitivity of the eigenvalues λ_j of $A+BF$ to perturbations in the components of $A+BF$ depends upon the magnitude of the condition numbers c_j, where

$$c_j=\|x_j\|_2\|y_j\|_2\geqslant 1 \tag{2.2}$$

(In the case of multiple eigenvalues, a particular choice of eigenvectors is assumed). Hence, every reasonable measure of the magnitude of the vector $c=(c_1, c_2, \cdots, c_n)^T$ is a reflection of the robustness of the eigenproblem (1.1).

Remark 2.1. If λ_1 is a multiple eigenvalue of $A+BF$ and

$$\lambda_1=\lambda_2=\cdots=\lambda_r, \quad \lambda_1\neq\lambda_j \text{ for } j=r+1, r+2, \cdots, n,$$

then the sensitivity of the eigenvalue λ_1 to perturbations of $A+BF$ depends upon the magnitude of the number

$$\tilde{c}_1=\max\{c_1, c_2, \cdots, c_r\}.$$

Hence, in this case we may refer to $r\tilde{c}_1$ as the condition number of the eigenvalue λ_1 (see [11, 75—77]).

A number of different measures ν of the robustness of the eigenproblem (1.1) are considered in [3]—[5], e.g.,

$$\nu_1=\|c\|_\infty,$$

$$\nu_2=\kappa_2(X)\equiv\|X\|_2\|X^{-1}\|_2,$$

$$\nu_3 = \|X\|_F \|X^{-1}\|_F / n,$$
$$\nu_3(D) = \|XD^{-1}\|_F \|DX^{-1}\|_F / \|D\|_F \|D^{-1}\|_F,$$

where

$$D = \mathrm{diag}(d_1, d_2, \cdots, d_n), \quad d_j > 0 \ \forall j. \tag{2.3}$$

In this paper we consider the following measures ν:

$$\nu_c = \|c\|_2 / n^{1/2}, \tag{2.4}$$
$$\nu_c(D) = \|Dc\|_2 / \|D\|_F, \tag{2.5}$$
$$\nu_h = \|X^T X - I\|_F / \sqrt{n(n-1)}, \tag{2.6}$$
$$\nu_h(d) = [\sum_{1 \le i < j \le n} d_{ij}^2 (x_i^T x_j)^2 / \sum_{1 \le i < j \le n} d_{ij}^2]^{1/2}, \tag{2.7}$$

where the matrix D is as in (2.3), and

$$d = (d_{12}, d_{13}, \cdots, d_{1n}, d_{23}, d_{24}, \cdots, d_{2n}, \cdots, d_{n-1,n})^T, \quad d_{ij} > 0 \quad \forall i, j. \tag{2.8}$$

For ν_h and $\nu_h(d)$ we assume that $\|x_j\|_2 = 1$, $\forall j$.

Obviously, if we restrict that the right eigenvectors x_j within being normalized such that $\|x_j\|_2 = 1$, $\forall j$, then $\nu_3 = \nu_c$ and $\nu_3(D) = \nu_c(D)$. But in this paper we remove the restriction when we use the measures ν_c and $\nu_c(D)$.

J. Kautsky et al.[4] have pointed out that "in essence, the aim of the robust pole placement problem is to select eigenvectors $x_j \in \mathscr{S}_j$, such that $\|x_j\|_2 = 1$ and the vectors x_j are as orthogonal as possible to each other." Therefore it is natural to take ν_h and $\nu_h(d)$ as measures of the conditioning of the matrix X.

Assume that X is an invertible matrix. Let

$$H = X^T X - I. \tag{2.9}$$

It follows from

$$X^T X = I + H, \quad X^{-1} X^{-T} = (I + H)^{-1}$$

that if $\|H\|_2 < 1$, then

$$\|X\|_2^2 \le 1 + \|H\|_2, \quad \|X^{-1}\|_2 \le (1 - \|H\|_2)^{-1},$$

and thus we have

$$\kappa_2(X) \le \left(\frac{1 + \|H\|_2}{1 - \|H\|_2}\right)^{1/2}. \tag{2.10}$$

Assume that $\|x_j\|_2 = 1$, $\forall j$. By (2.2) we get

$$c_j^2 = \|y_j\|_2^2 \le \lambda_{\max}(Y^T Y) = 1/\lambda_{\min}(X^T X). \tag{2.11}$$

Here $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximal and the minimal eigenvalues of a matrix, respectively. Utilizing the Gerschgorin theorem we know that there is an index i_0 $(1 \le i_0 \le n)$ such that

$$|\lambda_{\min}(X^T X) - x_{i_0}^T x_{i_0}| \le \sum_{\substack{k=1 \\ k \ne i_0}}^{n} |x_{i_0}^T x_k|,$$

and thus

$$1 - \sum_{\substack{k=1 \\ k \ne i_0}}^{n} |x_{i_0}^T x_k| \le \lambda_{\min}(X^T X). \tag{2.12}$$

Consequently, if

$$1 - \sum_{\substack{k=1 \\ k \ne i}}^{n} |x_i^T x_k| > 0, \quad \forall i,$$

then from (2.11) and (2.12) we have

$$c_j \leqslant 1 \Big/ \Big(1 - \max_{1\leqslant i \leqslant n} \sum_{\substack{k=1\\k\neq i}}^{n} |x_i^T x_k|\Big)^{1/2}. \tag{2.13}$$

Inequalities (2.10) and (2.13) show that if ν_h is sufficiently small then X is well-conditioned.

The following inequalities for ν_1, ν_2, ν_3 and ν_c have been proved in [9]:

$$\nu_c \leqslant \nu_1, \quad \nu_1 + \sqrt{\nu_1^2 - 1} \leqslant \nu_2, \tag{2.14}$$

$$\max\{\nu_3 + \sqrt{\nu_3^2 - 1},\ \nu_c + \sqrt{\nu_c^2 - 1}\} \leqslant \nu_2 \leqslant \frac{n\nu_3 - (n-2) + \sqrt{[n\nu_3 - (n-2)]^2 - 4}}{2}, \tag{2.15}$$

$$\sqrt{1 + \min_{1\leqslant k\leqslant n} \sum_{\substack{j=1\\j\neq k}}^{n} \Big(\frac{l_j}{l_k}\Big)^2}\ \nu_c \Big/ \sqrt{n} \leqslant \nu_3 \leqslant \sqrt{1 + \max_{1\leqslant k\leqslant n} \sum_{\substack{j=1\\j\neq k}}^{n} \Big(\frac{l_j}{l_k}\Big)^2}\ \nu_c \Big/ \sqrt{n}, \tag{2.16}$$

where

$$l_j = \|x_j\|_2, \quad j = 1, 2, \cdots, n.$$

The inequalities (2.14)—(2.16) show that the measures ν_1, ν_2, ν_3 and ν_c are mathematically equivalent, and the four measures take their minimal values simultaneously when X is real orthogonal.

In the next two sections we suggest two numerical methods for iteratively constructing a well-conditioned set of eigenvectors from given subspaces. Both procedures aim, at each step of the iteration, to reduce the value of one of the measures $\nu_c(D)$ and $\nu_h(d)$.

§ 3. Method (I)

The objective here is to choose vectors $x_j \in \mathscr{S}_j$, $j=1, 2, \cdots, n$, so as to minimize the measure of conditioning $\nu_c(D)$ discussed in § 2 (see (2.5)).

Assume that

$$\mathscr{S}_j = \mathscr{R}(S_j),\ S_j \in \mathbb{R}^{n\times m}, \quad S_j^T S_j = I,\ j = 1, \cdots, n. \tag{3.1}$$

Let

$$X = (x_1, x_2, \cdots, x_n), \quad Y = X^{-T} = (y_1, y_2, \cdots, y_n), \tag{3.2}$$

$$X_j = (x_1, \cdots, x_{j-1}, x_{j+1}, \cdots, x_n), \quad P_j = \begin{pmatrix} 0 & I^{(j-1)} & 0 \\ 1 & 0 & 0 \\ 0 & 0 & I^{(n-j)} \end{pmatrix} \tag{3.3}$$

and

$$x_j = S_j w_j,\ w_j \in \mathbb{R}^m, \quad \hat{w}_j = (w_1^T, \cdots, w_{j-1}^T, w_{j+1}^T, \cdots, w_n^T)^T. \tag{3.4}$$

Then

$$XP_j = (x_j, X_j).$$

Using the singular value decomposition (SVD) we obtain

$$X_j = (U_j, u_j)\begin{pmatrix}\Sigma_j \\ 0\end{pmatrix} V_j^T, \quad u_j \in \mathbb{R}^n,\ \Sigma_j = \mathrm{diag}(\sigma_{j1}, \cdots, \sigma_{j,n-1}), \tag{3.5}$$

where (U_j, u_j) and V_j are orthogonal matrices, and

$$X_j = X_j(\hat{w}_j),\ U_j = U_j(\hat{w}_j),\ u_j = u_j(\hat{w}_j),\ \Sigma_j = \Sigma_j(\hat{w}_j),\ V_j = V_j(\hat{w}_j).$$

Therefore

$$Y^T = P_j(x_j, X_j)^{-1} = P_j\left[(U_j, u_j)\begin{pmatrix} U_j^T x_j & \Sigma_j V_j^T \\ u_j^T x_j & 0 \end{pmatrix}\right]^{-1}$$

$$= P_j\begin{pmatrix} 0 & \frac{1}{u_j^T x_j} \\ V_j \Sigma_j^{-1} & * \end{pmatrix}\begin{pmatrix} U_j^T \\ u_j^T \end{pmatrix} = P_j\begin{pmatrix} \frac{u_j^T}{u_j^T x_j} \\ * \end{pmatrix},$$

and thus $y_j = u_j/u_j^T x_j$, or more explicitly,

$$y_j = u_j(\hat{w}_j)/u_j(\hat{w}_j)^T S_j w_j, \quad j=1, 2, \cdots, n.$$

Hence for $D = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ with $d_j > 0$, $\forall j$ we have

$$[\nu_c(D)]^2 = \frac{1}{\sum_{j=1}^n d_j^2} \sum_{j=1}^n d_j^2 c_j^2 = \sum_{j=1}^n \delta_j \|x_j\|_2^2 \|y_j\|_2^2 = \sum_{j=1}^n \frac{\delta_j w_j^T w_j}{[u_j(\hat{w}_j)^T S_j w_j]^2} \equiv f(w), \tag{3.6}$$

where

$$w = (w_1^T, w_2^T, \cdots, w_n^T)^T, \; w_j = (w_{1j}, w_{2j}, \cdots, w_{mj})^T \in \mathbb{R}^m, \quad j=1, 2, \cdots, n \tag{3.7}$$

and

$$\delta_j = \frac{d_j^2}{\sum_{j=1}^n d_j^2} > 0, \quad \forall j, \quad \sum_{j=1}^n \delta_j = 1. \tag{3.8}$$

Thus, we must solve an unconstrained optimization problem

$$\text{minimize } f(w), \tag{3.9}$$

where $f(w)$ and w are defined by (3.6) and (3.7), $w \in \mathbb{R}^N$ and $N = mn$.

Let

$$A_j(\hat{w}_j) = X_j(\hat{w}_j) X_j(\hat{w}_j)^T. \tag{3.10}$$

Utilizing (3.5) we have

$$A_j(\hat{w}_j) = (U_j(\hat{w}_j), u_j(\hat{w}_j))\begin{pmatrix} \Sigma_j(\hat{w}_j)^2 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} U_j(\hat{w}_j)^T \\ u_j(\hat{w}_j)^T \end{pmatrix}. \tag{3.11}$$

Assume that the matrix $X_j(\hat{w}_j)$ is of full column rank. Then the diagonal matrix $\Sigma_j(\hat{w}_j)$ is non-singular, and $u_j(\hat{w}_j)$ is a unit eigenvector of $A_j(\hat{w}_j)$ corresponding to the simple eigenvalue zero. By Theorem 2.4 of [7] we know that the eigenvector $u_j(\hat{w}_j)$ may be defined to be a real analytic function in some neighbourhood of $\hat{w}_j$ and the partial derivatives of $u_j(\hat{w}_j)$ with respect to w_{il} can be represented as follows:

$$\frac{\partial u_j(\hat{w}_j)}{\partial w_{il}} = -U_j(\hat{w}_j)\Sigma_j(\hat{w}_j)^{-2} U_j(\hat{w}_j)^T \frac{\partial A_j(\hat{w}_j)}{\partial w_{il}} u_j(\hat{w}_j), \; l \neq j, \; i=1, 2, \cdots, m. \tag{3.12}$$

Since

$$A_j(\hat{w}_j) = \sum_{\substack{k=1 \\ k\neq j}}^n S_k w_k w_k^T S_k^T,$$

we get

$$\frac{\partial A_j(\hat{w}_j)}{\partial w_{il}} = S_l(w_l e_i^T + e_i w_l^T) S_l^T, \quad l \neq j, \tag{3.13}$$

where e_i is the i-th column vector of $I^{(m)}$. Substituting (3.13) into (3.12) we obtain

$$\frac{\partial (u_j(\hat{w}_j)^T)}{\partial w_{il}} = -u_j(\hat{w}_j)^T S_l (w_l e_i^T + e_i w_l^T) S_l^T U_j(\hat{w}_j) \Sigma_j(\hat{w}_j)^{-2} U_j(\hat{w}_j)^T$$

and

$$\frac{\partial(u_j(\hat{w}_j)^T)}{\partial w_l}=\left(\frac{\partial(u_j(\hat{w}_j)^T)}{\partial w_{1l}},\ \cdots,\ \frac{\partial(u_j(\hat{w}_j)^T)}{\partial w_{m_l,l}}\right)^T$$

$$=-\left[u_j(\hat{w}_j)^T S_l w_l\begin{pmatrix}e_1^T S_l^T\\ \vdots\\ e_m^T S_l^T\end{pmatrix}+\begin{pmatrix}e_1^T S_l^T\\ \vdots\\ e_m^T S_l^T\end{pmatrix}u_j(\hat{w}_j)w_l^T S_l^T\right]U_j(\hat{w}_j)\Sigma_j(\hat{w}_j)^{-2}U_j(\hat{w}_j)^T$$

$$=-\left[u_j(\hat{w}_j)^T S_l w_l\cdot I^{(m)}+S_l^T u_j(\hat{w}_j)w_l^T\right]S_l^T U_j(\hat{w}_j)\Sigma_j(\hat{w}_j)^{-2}U_j(\hat{w}_j)^T\in\mathbb{R}^{m\times n}. \tag{3.14}$$

Let $\nabla f(w)$ be the gradient vector and let

$$\frac{\partial f(w)}{\partial w_l}=\left(\frac{\partial f(w)}{\partial w_{1l}},\ \frac{\partial f(w)}{\partial w_{2l}},\ \cdots,\ \frac{\partial f(w)}{\partial w_{ml}}\right)^T,\quad l=1,\ 2,\ \cdots,\ n, \tag{3.15}$$

then

$$\nabla f(w)=\left(\left(\frac{\partial f(w)}{\partial w_1}\right)^T,\ \left(\frac{\partial f(w)}{\partial w_2}\right)^T,\ \cdots,\ \left(\frac{\partial f(w)}{\partial w_n}\right)^T\right)^T. \tag{3.16}$$

From (3.6) and (3.15) it follows that

$$\frac{\partial f(w)}{\partial w_l}=\frac{2\delta_l w_l}{[u_l(\hat{w}_l)^T S_l w_l]^2}-\frac{2\delta_l w_l^T w_l S_l^T u_l(\hat{w}_l)}{[u_l(\hat{w}_l)^T S_l w_l]^3}$$
$$-2\sum_{\substack{j=1\\ j\neq l}}^{n}\frac{\delta_j w_j^T w_j}{[u_j(\hat{w}_j)^T S_j w_j]^3}\cdot\frac{\partial(u_j(\hat{w}_j)^T)}{\partial w_l}\cdot S_j w_j,\quad l=1,\ 2,\ \cdots,\ n. \tag{3.17}$$

Substituting (3.14) into (3.17) we obtain

$$\frac{\partial f(w)}{\partial w_l}=2\delta_l\left[w_l-\frac{w_l^T w_l S_l^T u_l(\hat{w}_l)}{u_l(\hat{w}_l)^T S_l w_l}\right]\Big/[u_l(\hat{w}_l)^T S_l w_l]^2$$
$$+2\sum_{\substack{j=1\\ j\neq l}}^{n}\frac{\delta_j w_j^T w_j}{u_j(\hat{w}_j)^T S_j w_j}Y_{j,l}(w)S_l^T Z_j(\hat{w}_j)S_j w_j/[u_j(\hat{w}_j)^T S_j w_j]^2,\quad l=1,\ 2,\ \cdots,\ n, \tag{3.18}$$

where

$$Y_{j,l}(w)=u_j(\hat{w}_j)^T S_l w_l\cdot I^{(m_l)}+S_l^T u_j(\hat{w}_j)w_l^T \tag{3.19}$$

and

$$Z_j(\hat{w}_j)=U_j(\hat{w}_j)\Sigma_j(\hat{w}_j)^{-2}U_j(\hat{w}_j)^T. \tag{3.20}$$

This proves the following theorem.

Theorem 3.1. *Suppose that $X_j(\hat{w}_j)$, $U_j(\hat{w}_j)$, $u_j(\hat{w}_j)$, $\Sigma_j(\hat{w}_j)$ and $f(w)$ are defined as in (3.1)—(3.6). Assume that the matrix $X_j(\hat{w}_j)$ for $j=1,\ 2,\ \cdots,\ n$ are of full column rank. Then the formulas (3.16) and (3.18)—(3.20) give the expression of the gradient vector $\nabla f(w)$.*

The Davidon–Fletcher–Powell method is now applied to solving the optimization problem (3.9):

Initialization Step. Let $\varepsilon>0$ be the termination scalar. Choose an initial vector $w^{(0)}=(w_1^{(0)T},\ w_2^{(0)T},\ \cdots,\ w_n^{(0)T})^T\in\mathbb{R}^N$ with $w_j^{(0)}\in\mathbb{R}^m$, $\forall j$ and an $N\times N$ positive definite matrix H_0 (e.g., $H_0=I^{(N)}$).

Main Step.

(1) Let $k:=0$.

(2) Compute $g_k=\nabla f(w^{(k)})$, $p_k=-H_k g_k$.

(3) Determine $w^{(k+1)}=w^{(k)}+\lambda_k p_k$ by means of an approximate minimization

$$f(w^{(k)}+\lambda_k p_k)\approx\min_{\lambda\geq 0}f(w^{(k)}+\lambda p_k).$$

(4) Set $\Delta w^{(k)}=w^{(k+1)}-w^{(k)}$. If $\|\Delta w^{(k)}\|_2\leq\varepsilon$, then $w^*=w^{(k+1)}$ is an approximate optimal solution; if $\|\Delta w^{(k)}\|_2>\varepsilon$, go to Step (5).

(5) Compute $g_{k+1}=\nabla f(w^{(k+1)})$, $g_{k+1}-g_k=h_k$,

$$H_{k+1}=H_k+\frac{\Delta w^{(k)}(\Delta w^{(k)})^T}{(\Delta w^{(k)})^T h_k}-\frac{H_k h_k h_k^T H_k^T}{h_k^T H_k h_k}$$

and

$$p_{k+1}=-H_{k+1}g_{k+1}.$$

Replace k by $k+1$, and repeat Step (3).

It seems that the above mentioned procedure is rather complicated, but it gives good solutions in practice, and it can lead to rapid convergence (see § 5).

Remark 3.1. At the Initialization step, if we choose an initial vector $w^{(0)}$ such that for some j of the indexes 1, 2, ⋯, n the matrix $X_j(\hat{w}_j^{(0)})$ has a very small singular value, then we must choose another initial vector anew.

Remark 3.2. At Step (2) we compute $\nabla f(w^{(k)})$ by formulas (3.16) and (3.18)—(3.20), in which the SVDs of $X_1(\hat{w}_1^{(k)}), \cdots, X_n(\hat{w}_n^{(k)})$ are obtained by using standard routines. But observe that the matrix $X_j(\hat{w}_j^{(k)})$ is obtained by a rank-one update of the matrix $X_{j-1}(\hat{w}_{j-1}^{(k)})$, hence it is necessary to develop techniques for updating SVD(ref. [1]).

Remark 3.3. At Step (3) the line search techniques based on curve fitting procedures, such as cubic fit and quadratic fit, are feasible in practice.

§ 4. Method (II)

The objective here is to choose vectors $x_j\in\mathscr{S}_j$ with $\|x_j\|_2=1$, $\forall j$, so as to minimize the measure of conditioning $\nu_k(d)$ discussed in § 2 (see (2.7)).

Suppose that $\mathscr{S}_j$ and S_j are the same as in (3.1), and $N=mn$. Let

$$\mathbf{S}=\{w=(w_1^T,\cdots,w_n^T)^T\in\mathbf{R}^N:\ w_j\in\mathbf{R}^m,\ \|w_j\|_2=1\quad\forall j\},\tag{4.0.1}$$

$$A_{ij}=\gamma_{ij}S_i^T S_j,\ r_{ij}(w)=w_i^T A_{ij} w_j,\ A_{ij}(w)=r_{ij}(w)A_{ij},\ \forall i\neq j,\tag{4.0.2}$$

and

$$A(w)=\begin{pmatrix} \alpha^2 I & -A_{12}(w) & \cdots & -A_{1n}(w)\\ -A_{12}(w)^T & \alpha^2 I & \ddots & \vdots\\ \vdots & \ddots & \ddots & \vdots\\ \vdots & & \alpha^2 I & -A_{n-1,n}(w)\\ -A_{1n}(w)^T & \cdots & -A_{n-1,n}(w)^T & \alpha^2 I \end{pmatrix},\quad \alpha>\beta,\tag{4.0.3}$$

where

$$\gamma_{ij}=d_{ij}\Big/\sqrt{\sum_{1<i<j<n} d_{ij}^2}>0,\qquad \sum_{1<i<j<n}\gamma_{ij}^2=1\tag{4.0.4}$$

and

$$\beta=\gamma\max_{1<k<n}\|(S_1,\cdots,S_{k-1},S_{k+1},\cdots,S_n)\|_2,\quad \gamma=\max_{i,j}\gamma_{ij}.\tag{4.0.5}$$

Then minimizing the measure $\nu_k(d)$ may be reduced to solving the following nonlinear programming problem with constraints:

$$\begin{aligned}&\text{maximize } w^T A(w) w,\\ &\text{subject to } w\in\mathbf{S}.\end{aligned}\tag{4.0.6}$$

In this section we develop an algorithm, called P-SOR algorithm (i.e., the Power-Successive Overrelaxation, see [8]), for finding a global optimal solution w of problem (4.0.6).

4.1. P-SOR Algorithm.

A summary of the algorithm is given below.

Initialization Step. Let $\varepsilon>0$ be the termination scalar, δ be a positive number near zero and q be a positive number less than the unity. Choose an initial vector $w^{(0)}=(w_1^{(0)\tau}, w_2^{(0)\tau}, \cdots, w_n^{(0)\tau})^T\in\mathbb{S}^N$.

Main Step.

(1) Let $k:=0$.

(2) For $i=1, 2, \cdots, n$, we compute

$$\hat{z}_i^{(k+1)}=\alpha^2 w_i^{(k)}-\sum_{j=1}^{i-1} w_i^{(k)\tau}A_{ij}w_j^{(k+1)}A_{ij}w_j^{(k+1)}-\sum_{j=i+1}^{n} w_i^{(k)\tau}A_{ij}w_j^{(k)}A_{ij}w_j^{(k)}, \tag{4.1.1}$$

$$\tau_i^{(k)}=\|\hat{z}_i^{(k+1)}\|_2 \tag{4.1.2}$$

and

$$z_i^{(k+1)}=\begin{cases}\hat{z}_i^{(k+1)}/\tau_i^{(k)}, & \text{if } \tau_i^{(k)}>0,\\ w_i^{(k)}, & \text{if } \tau_i^{(k)}=0.\end{cases} \tag{4.1.3}$$

If $\tau_i^{(k)}\leqslant\alpha^2-\beta^2$ then choose

$$\omega_i^{(k)}\in\left[\frac{1}{2}+\delta,\ 2\right], \tag{4.1.4}$$

if $\tau_i^{(k)}>\alpha^2-\beta^2$ then choose

$$\omega_i^{(k)}\in\left[\frac{1}{2}+\delta,\ \frac{2\tau_i^{(k)}}{\tau_i^{(k)}-(1-q)(\alpha^2-\beta^2)}\right]. \tag{4.1.5}$$

Compute

$$y_i^{(k+1)}=(1-\omega_i^{(k)})w_i^{(k)}+\omega_i^{(k)}z_i^{(k+1)}, \tag{4.1.6}$$

$$\mu_i^{(k)}=\|y_i^{(k+1)}\|_2 \tag{4.1.7}$$

and

$$w_i^{(k+1)}=y_i^{(k+1)}/\mu_i^{(k)}. \tag{4.1.8}$$

(3) Set $\Delta w^{(k)}=w^{(k+1)}-w^{(k)}$. If $\|\Delta w^{(k)}\|_2\leqslant\varepsilon$ then $w^*=w^{(k+1)}$ is an approximate optimal solution; if $\|\Delta w^{(k)}\|_2>\varepsilon$, then replace k by $k+1$, and repeat Step (2).

By (4.1.1)—(4.1.3) and (4.1.6)—(4.1.8), we have

$$\tau_i^{(k)}\mu_i^{(k)}w_i^{(k+1)}=(1-\omega_i^{(k)})\tau_i^{(k)}w_i^{(k)}$$
$$+\omega_i^{(k)}\Big(\alpha^2 w_i^{(k)}-\sum_{j=1}^{i-1} w_i^{(k)\tau}A_{ij}w_j^{(k+1)}A_{ij}w_j^{(k+1)}-\sum_{j=i+1}^{n} w_i^{(k)\tau}A_{ij}w_j^{(k)}A_{ij}w_j^{(k)}\Big),$$

i.e.,

$$\tau_i^{(k)}\mu_i^{(k)}w_i^{(k+1)}+\omega_i^{(k)}\sum_{j=1}^{i-1} w_i^{(k+1)\tau}A_{ij}w_j^{(k+1)}A_{ij}w_j^{(k+1)}-\omega_i^{(k)}\sum_{j=1}^{i-1}(w_i^{(k+1)}-w_i^{(k)})^T A_{ij}w_j^{(k+1)}A_{ij}w_j^{(k+1)}$$
$$=(1-\omega_i^{(k)})\tau_i^{(k)}w_i^{(k)}+\omega_i^{(k)}\Big(\alpha^2 w_i^{(k)}-\sum_{j=i+1}^{n} w_i^{(k)\tau}A_{ij}w_j^{(k)}A_{ij}w_j^{(k)}\Big),$$
$$i=1, 2, \cdots, n, \quad k=0, 1, 2, \cdots. \tag{4.1.9}$$

We set

$$w^{(k)}=(w_1^{(k)\tau}, w_2^{(k)\tau}, \cdots, w_n^{(k)\tau})^T,$$
$$A_{ij}^{(k)}=r_{ij}(w^{(k)})A_{ij}, \quad i, j=1, 2, \cdots, n, \tag{4.1.10}$$

$$U^{(k)}=\begin{pmatrix} 0 & A_{12}^{(k)} & \cdots & A_{1n}^{(k)} \\ & \ddots & \ddots & \vdots \\ & & \ddots & A_{n-1,n}^{(k)} \\ 0 & & & 0 \end{pmatrix}, \tag{4.1.11}$$

$$A^{(k)}=\alpha^2 I-U^{(k)}-U^{(k)T}, \tag{4.1.12}$$

$$\delta_{ij}^{(k)}=(w_i^{(k+1)}-w_i^{(k)})^T A_{ij} w_j^{(k+1)}, \quad \Delta_{ij}^{(k+1)}=\delta_{ij}^{(k+1)} A_{ij}, \quad 1\leqslant j<i\leqslant n, \tag{4.1.13}$$

$$\Delta^{(k+1)}=\begin{pmatrix} 0 & & & 0 \\ \Delta_{21}^{(k+1)} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \Delta_{n1}^{(k+1)} & \cdots & \Delta_{n,n-1}^{(k+1)} & 0 \end{pmatrix}, \tag{4.1.14}$$

$$T^{(k)}=\mathrm{diag}(\tau_1^{(k)} I^{(m)}, \cdots, \tau_n^{(k)} I^{(m)}), \tag{4.1.15}$$

and

$$M^{(k)}=\mathrm{diag}(\mu_1^{(k)} I^{(m)}, \cdots, \mu_n^{(k)} I^{(m)}) \tag{4.1.16}$$

$$\Omega^{(k)}=\mathrm{diag}(\omega_1^{(k)} I^{(m)}, \cdots, \omega_n^{(k)} I^{(m)}). \tag{4.1.17}$$

Thereupon the iterative formula (4.1.9) may be written as

$$\begin{aligned}(T^{(k)}M^{(k)}+\Omega^{(k)}U^{(k+1)T})w^{(k+1)}-\Omega^{(k)}\Delta^{(k+1)}w^{(k+1)} \\ =[(I-\Omega^{(k)})T^{(k)}+\Omega^{(k)}(\alpha^2 I-U^{(k)})]w^{(k)},\end{aligned} \tag{4.1.18}$$

where $T^{(k)}\geqslant 0$, $M^{(k)}>0$, $\Omega^{(k)}>0$.

4.2. Convergence properties of P-SOR Algorithm.

Let

$$\mathbf{B}=\{w=(w_1^T, w_2^T, \cdots, w_n^T)^T\in\mathbb{R}^N:\ w_j\in\mathbb{R}^m,\ \|w_j\|_2\leqslant 1,\ \forall j\}. \tag{4.2.1}$$

we can prove easily that the programming problem (4.0.6) is equivalent to the following programming problem with inequality constraints:

$$\begin{aligned}&\text{maximize } w^T A(w) w,\\ &\text{subject to } w\in\mathbf{B}.\end{aligned} \tag{4.2.2}$$

Let

$$\mathbf{D}=\{T=\mathrm{diag}(\tau_1 I^{(m)}, \cdots, \tau_n I^{(m)}):\ \tau_i\geqslant 0,\ \forall i\}. \tag{4.2.3}$$

We consider a multiparameter eigenvalue problem

$$A(w)w=Tw, \quad T\in\mathbf{D},\ w\in\mathbf{S}. \tag{4.2.4}$$

Using the Kuhn-Tucker optimality conditions[6] we can prove that every solution to problem (4.2.2) is necessarily a solution to problem (4.2.4).

We define

$$\rho(w)=w^T A(w) w, \quad w\in\mathbf{S}, \tag{4.2.5}$$

where $A(w)$ is defined by (4.0.3). Now we discuss the convergence properties of the P-SOR algorithm.

Theorem 4.1. *Let $w^{(k)}$ and $w^{(k+1)}$ be the k-th and the $(k+1)$-th iterative solutions generated by the* P-SOR *algorithm, respectively. Then*

$$\rho(w^{(k+1)})-\rho(w^{(k)})$$
$$\geq 2(w^{(k+1)}-w^{(k)})^T[(\alpha^2-\beta^2)I+\Omega^{(k)^{-1}}T^{(k)}(M^{(k)}+I-\Omega^{(k)})](w^{(k+1)}-w^{(k)}). \quad (4.2.6)$$

Proof. From (4.2.5), (4.1.12) and (4.1.18) we get

$$\begin{aligned}\rho(w^{(k+1)}) &= w^{(k+1)^T}(\alpha^2 I-U^{(k+1)}-U^{(k+1)^T})w^{(k+1)}\\ &= w^{(k+1)^T}[\alpha^2 I-U^{(k+1)}+\Omega^{(k)^{-1}}T^{(k)}M^{(k)}-\Omega^{(k)^{-1}}(T^{(k)}M^{(k)}+\Omega^{(k)}U^{(k+1)^T})]w^{(k+1)}\\ &= n\alpha^2 I-w^{(k+1)^T}U^{(k+1)}w^{(k+1)}+w^{(k+1)^T}\Omega^{(k)^{-1}}T^{(k)}M^{(k)}w^{(k+1)}\\ &\quad -w^{(k+1)^T}\Delta^{(k+1)}w^{(k+1)}-w^{(k+1)^T}\Omega^{(k)^{-1}}(I-\Omega^{(k)})T^{(k)}w^{(k)}\\ &\quad -\alpha^2 w^{(k+1)^T}w^{(k)}+w^{(k+1)^T}U^{(k)}w^{(k)}\end{aligned}$$

and

$$\begin{aligned}\rho(w^{(k)}) &= w^{(k)^T}(\alpha^2 I-U^{(k)}-U^{(k)^T})w^{(k)}\\ &= w^{(k)^T}\{[(I-\Omega^{(k)})T^{(k)}+\Omega^{(k)}(\alpha^2 I-U^{(k)})]^T\Omega^{(k)^{-1}}-(I-\Omega^{(k)})T^{(k)}\Omega^{(k)^{-1}}-U^{(k)}\}w^{(k)}\\ &= w^{(k+1)^T}T^{(k)}M^{(k)}\Omega^{(k)^{-1}}w^{(k)}+w^{(k+1)^T}U^{(k+1)}w^{(k)}\\ &\quad -w^{(k+1)^T}\Delta^{(k+1)^T}w^{(k)}-w^{(k)^T}(I-\Omega^{(k)})T^{(k)}\Omega^{(k)^{-1}}w^{(k)}-w^{(k)^T}U^{(k)}w^{(k)},\end{aligned}$$

thereby

$$\rho(w^{(k+1)})-\rho(w^{(k)})=r_1^{(k)}+r_2^{(k)}+r_3^{(k)}, \quad (4.2.7)$$

where

$$r_1^{(k)}=w^{(k+1)^T}\Omega^{(k)^{-1}}T^{(k)}M^{(k)}(w^{(k+1)}-w^{(k)})+(w^{(k)}-w^{(k+1)})^T\Omega^{(k)^{-1}}(I-\Omega^{(k)})T^{(k)}w^{(k)}, \quad (4.2.8)$$

$$r_2^{(k)}=n\alpha^2 I-w^{(k+1)^T}U^{(k+1)}w^{(k+1)}-\alpha^2 w^{(k+1)^T}w^{(k)}+w^{(k)^T}U^{(k)}w^{(k)} \quad (4.2.9)$$

and

$$r_3^{(k)}=w^{(k+1)^T}\Delta^{(k+1)^T}(w^{(k)}-w^{(k+1)})+w^{(k+1)^T}U^{(k)}w^{(k)}-w^{(k+1)^T}U^{(k+1)}w^{(k)}. \quad (4.2.10)$$

First, it is easy to see that

$$\begin{aligned}r_1^{(k)} &= \sum_{i=1}^{n}\frac{\tau_i^{(k)}}{\omega_i^{(k)}}(\mu_i^{(k)}+1-\omega_i^{(k)})(1-w_i^{(k+1)^T}w_i^{(k)})\\ &= \frac{1}{2}(w^{(k+1)}-w^{(k)})^T\Omega^{(k)^{-1}}T^{(k)}(M^{(k)}+I-\Omega^{(k)})(w^{(k+1)}-w^{(k)}). \quad (4.2.11)\end{aligned}$$

Then, from

$$\rho(w^{(k+1)})-\rho(w^{(k)})=2(w^{(k)^T}U^{(k)}w^{(k)}-w^{(k+1)^T}U^{(k+1)}w^{(k+1)}) \quad (4.2.12)$$

and

$$(w^{(k+1)}-w^{(k)})^T(w^{(k+1)}-w^{(k)})=2(n-w^{(k+1)^T}w^{(k)})$$

it follows that

$$r_2^{(k)}=\frac{1}{2}[\rho(w^{(k+1)})-\rho(w^{(k)})]+\frac{\alpha^2}{2}(w^{(k+1)}-w^{(k)})^T(w^{(k+1)}-w^{(k)}). \quad (4.2.13)$$

Finally, we consider $r_3^{(k)}$. Let

$$r_{ij}(k,l)=w_i^{(k)^T}A_{ij}w_j^{(l)}, \quad i,j=1,2,\cdots,n,\ k,l=0,1,2,\cdots, \quad (4.2.14)$$

then

$$\begin{aligned}r_3^{(k)} &= -\sum_{1<j<i<n}[(w_i^{(k+1)}-w_i^{(k)})A_{ij}w_j^{(k+1)}]^2\\ &\quad +\sum_{1<i<j<n}w_i^{(k)^T}A_{ij}w_j^{(k)}w_i^{(k+1)^T}A_{ij}w_j^{(k)}-\sum_{1<i<j<n}w_i^{(k+1)^T}A_{ij}w_j^{(k+1)}w_i^{(k+1)^T}A_{ij}w_j^{(k)}\\ &= -\sum_{1<i<j<n}[r_{ij}^2(k+1,k+1)+r_{ij}^2(k+1,k)-r_{ij}(k,k)r_{ij}(k+1,k)\\ &\quad -r_{ij}(k+1,k+1)r_{ij}(k+1,k)]\end{aligned}$$

$$= -\frac{1}{2}\sum_{1<i<j<n}[r_{ij}(k+1,k+1)-r_{ij}(k+1,k)]^2-\frac{1}{2}\sum_{1<i<j<n}[r_{ij}(k+1,k)-r_{ij}(k,k)]^2$$
$$-\frac{1}{2}\sum_{1<i<j<n}[r_{ij}^2(k+1,k+1)-r_{ij}^2(k,k)]$$
$$= -\frac{1}{2}\Big(\sum_{1<i<j<n}[w_i^{(k+1)T}A_{ij}(w_j^{(k+1)}-w_j^{(k)})]^2+\sum_{1<i<j<n}[(w_i^{(k+1)}-w_i^{(k)})^TA_{ij}w_j^{(k)}]^2\Big)$$
$$+\frac{1}{2}(w^{(k)T}U^{(k)}w^{(k)}-w^{(k+1)T}U^{(k+1)}w^{(k+1)}). \tag{4.2.15}$$

Utilizing the Cauchy inequality we have

$$\sum_{1<i<j<n}[w_i^{(k+1)T}A_{ij}(w_j^{(k+1)}-w_j^{(k)})]^2+\sum_{1<i<j<n}[(w_i^{(k+1)}-w_i^{(k)})^TA_{ij}w_j^{(k)}]^2$$
$$\leqslant \sum_{1<i<j<n}(w_j^{(k+1)}-w_j^{(k)})^TA_{ij}^TA_{ij}(w_j^{(k+1)}-w_j^{(k)})$$
$$+\sum_{1<i<j<n}(w_i^{(k+1)}-w_i^{(k)})^TA_{ij}A_{ij}^T(w_i^{(k+1)}-w_i^{(k)})$$
$$=\sum_{\substack{i,j=1\\ i\neq j}}^{n}(w_j^{(k+1)}-w_j^{(k)})^TA_{ij}^TA_{ij}(w_j^{(k+1)}-w_j^{(k)})$$
$$=\sum_{j=1}^{n}\Big[(w_j^{(k+1)}-w_j^{(k)})^T\sum_{\substack{i=1\\ i\neq j}}^{n}A_{ji}A_{ji}^T(w_j^{(k+1)}-w_j^{(k)})\Big]. \tag{4.2.16}$$

But from (4.0.2) and (4.0.5),

$$\sum_{\substack{i=1\\ i\neq j}}^{n}A_{ji}A_{ji}^T=\sum_{\substack{i=1\\ i\neq j}}^{n}\gamma_{ji}^2S_j^TS_iS_i^TS_j\leqslant\gamma^2S_j^T\Big(\sum_{\substack{i=1\\ i\neq j}}^{n}S_iS_i^T\Big)S_j$$
$$\leqslant\gamma^2\|(S_1,\cdots,S_{j-1},S_{j+1},\cdots,S_n)\|_2^2\cdot I=\beta^2I.$$

Consequently,

$$\sum_{j=1}^{n}\Big[(w_j^{(k+1)}-w_j^{(k)})^T\sum_{\substack{i=1\\ i\neq j}}^{n}A_{ji}A_{ji}^T(w_j^{(k+1)}-w_j^{(k)})\Big]\leqslant\beta^2\|w^{(k+1)}-w^{(k)}\|_2^2. \tag{4.2.17}$$

Combining (4.2.15)—(4.2.17) with (4.2.12) we get

$$r_3^{(k)}\geqslant-\frac{1}{2}\beta^2\|w^{(k+1)}-w^{(k)}\|_2^2+\frac{1}{4}[\rho(w^{(k+1)})-\rho(w^{(k)})]. \tag{4.2.18}$$

Substituting (4.2.11), (4.2.13) and (4.2.18) into (4.2.7) we obtain inequality (4.2.6). ∎

Theorem 4.2. *Let $w^{(k)}$ and $w^{(k+1)}$ be the k-th and the $(k+1)$-th iterative solutions generated by the* P-SOR *algorithm, respectively. Then*

$$\rho(w^{(k+1)})-\rho(w^{(k)})\geqslant2q(\alpha^2-\beta^2)\|w^{(k+1)}-w^{(k)}\|_2^2. \tag{4.2.19}$$

Proof. First we rewrite inequality (4.2.6) as

$$\rho(w^{(k+1)})-\rho(w^{(k)})\geqslant\sum_{i=1}^{n}\rho_i, \tag{4.2.20}$$

where

$$\rho_i=2\Big[\alpha^2-\beta^2+\frac{\tau_i^{(k)}}{\omega_i^{(k)}}(\mu_i^{(k)}+1-\omega_i^{(k)})\Big]\|w_i^{(k+1)}-w_i^{(k)}\|_2^2,\quad i=1,2,\cdots,n. \tag{4.2.21}$$

If $\omega_i^{(k)}<1$, then clearly

$$\rho_i\geqslant2\Big(\alpha^2-\beta^2+\frac{\tau_i^{(k)}\mu_i^{(k)}}{\omega_i^{(k)}}\Big)\|w_i^{(k+1)}-w_i^{(k)}\|_2^2\geqslant2q(\alpha^2-\beta^2)\|w_i^{(k+1)}-w_i^{(k)}\|_2^2. \tag{4.2.22}$$

If $\omega_i^{(k)}\geqslant1$, then by (4.1.6) and (4.1.7) we have

$$\mu_i^{(k)^2} = (1-\omega_i^{(k)})^2 + \omega_i^{(k)^2} - 2\omega_i^{(k)}(\omega_i^{(k)}-1) w_i^{(k)T} z_i^{(k+1)}$$
$$\geqslant (1-\omega_i^{(k)})^2 + \omega_i^{(k)^2} - 2\omega_i^{(k)}(\omega_i^{(k)}-1) = 1,$$

i.e., $\mu_i^{(k)} \geqslant 1$. Therefore, from (4.1.4) and (4.1.5) we know that if $\tau_i^{(k)} \leqslant \alpha^2 - \beta^2$ and $1 \leqslant \omega_i^{(k)} \leqslant 2$, then

$$\alpha^2 - \beta^2 + \frac{\tau_i^{(k)}}{\omega_i^{(k)}}(\mu_i^{(k)} + 1 - \omega_i^{(k)}) > q(\alpha^2 - \beta^2),$$

and that if $\tau_i^{(k)} > \alpha^2 - \beta^2$ and $1 \leqslant \omega_i^{(k)} \leqslant 2\tau_i^{(k)} / [\tau_i^{(k)} - (1-q)(\alpha^2 - \beta^2)]$, then

$$\alpha^2 - \beta^2 + \frac{\tau_i^{(k)}}{\omega_i^{(k)}}(\mu_i^{(k)} + 1 - \omega_i^{(k)}) \geqslant \alpha^2 - \beta^2 - (1-q)(\alpha^2 - \beta^2) = q(\alpha^2 - \beta^2).$$

Hence the inequality (4.2.22) is also valid in the case $\omega_i^{(k)} \geqslant 1$. **Substituting** (4.2.20) we obtain the inequality (**4**.2.19). ▮

Theorem 4.2 shows that for any vector sequence $\{w^{(k)}\}$ generated by the P-SOR algorithm the sequence $\{\rho(w^{(k)})\}$ is monotone increasing. But on the other hand we have

$$\rho(w^{(k)}) = n\alpha^2 - \sum_{i,j=1}^{n} (w_i^{(k)T} A_{ij} w_j^{(k)})^2 \leqslant n\alpha^2 \quad \forall k.$$

Consequently we get the following corollary.

Corollary 4.1. If $\{w^{(k)}\}$ is any vector sequence generated by the P-SOR algorithm, then there exists a real number $r \leqslant n\alpha^2$ such that

$$\lim_{k\to\infty} \rho(w^{(k)}) = r.$$

Moreover, by Theorem 4.2 and Corollary 4.1 we can deduce the following corollary easily (ref. [8], Corollary 3.2 and Corollary 3.3).

Corollary 4.2. Let l be any fixed natural number. If $\{w^{(k)}\}$ and $\{T^{(k)}\}$ are sequences generated by the P-SOR algorithm, where

$$T^{(k)} = \mathrm{diag}(\tau_1^{(k)} I^{(m)}, \cdots, \tau_n^{(k)} I^{(m)}),$$

then

$$\lim_{k\to\infty} \| w^{(k+l)} - w^{(k)} \|_2 = 0, \quad \lim_{k\to\infty} \| T^{(k+l)} - T^{(k)} \|_2 = 0.$$

The following two theorem clarify the relation between any vector sequence $\{w^{(k)}\}$ generated by the P-SOR algorithm and the multiparameter eigenvalue problem (4.2.4). The proofs are similar to that in [8] (Theorem 3.3 and Theorem 3.4), and and therefore are omitted.

Theorem 4.3. *Let w^* be any limit point of a sequence $\{w^{(k)}\}$ generated by the* P-SOR *algorithm. Then there exists a corresponding limit point*

$$T^* = \mathrm{diag}\ (\tau_1^* I^{(m)}, \cdots, \tau_n^* I^{(m)})$$

of the sequence $\{T^{(k)}\}$ such that

$$A(w^*) w^* = T^* w^*, \quad T^* \in \mathsf{D},\ w^* \in \mathsf{S}.$$

Theorem 4.4. *Let $\{w^{(k)}\}$ be any sequence generated by the* P-SOR *algorithm and $\{T^{(k)}\}$ be the corresponding sequence. If for some k*

$$\rho(w^{(k+1)}) = \rho(w^{(k)}),$$

then $T^{(k)} = \mathrm{diag}(\tau_1^{(k)} I^{(m)}, \cdots, \tau_n^{(k)} I^{(m)})$ and $w^{(k)}$ satisfy

$$A(w^{(k)}) w^{(k)} = T^{(k)} w^{(k)}, \quad T^{(k)} \in \mathsf{D},\ w^{(k)} \in \mathsf{S}.$$

The author's computational experience shows that the P-SOR algorithm is simple to implement and any sequence $\{w^{(k)}\}$ generated by the P-SOR algorithm always converges to a global optimal solution to problem (4.0.6). Besides, Method (II) described here can be used to produce a reasonable initial solution for the use of Method (I).

Remark 4.1. From (3.1), (4.0.5) and

$$\|(S_1, \cdots, S_{j-1}, S_{j+1}, \cdots, S_n)\|_2^2 \leqslant \sum_{\substack{i=1\\i\neq j}}^{n} \|S_i S_i^T\|_2 = n-1$$

we see that $\beta \leqslant \gamma\sqrt{n-1}$. Computational practice shows that the choice of $\alpha \gtrsim \gamma\sqrt{n-1}$ is better.

Remark 4.2. The simplest method for choosing overrelaxation parameters $\{\omega_i^{(k)}\}$ is clearly to take $\omega_i^{(k)} = \omega$ for all $i = 1, 2, \cdots, n$ and $k = 0, 1, 2, \cdots$. By the author's experience it is suitable to take $\omega \lesssim 3$.

§ 5. Numerical Results

Test Example $n=3$, $m=2$ (see [3], [5]).

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 6 & -11 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

The eigenvalues of A are $\{1.0, 2.0, 3.0\}$. This system is unstable and a feedback matrix F is required to stabilize the system. We assign the eigenvalue set $\mathscr{S} = \{-0.2, -0.2, -10.0\}$, which produces a stable system. Step A described in § 1 gives

$$S_1 = S_2 = \begin{pmatrix} 0.842844 & -0.359245 \\ 0.492834 & 0.216169 \\ 0.216169 & 0.907862 \end{pmatrix}, \quad S_3 = \begin{pmatrix} -0.817060 & 0.557086 \\ 0.418760 & 0.396300 \\ 0.396300 & 0.729796 \end{pmatrix}$$

and $S = (S_1, S_2, S_3)$ with

$$\|S\|_2 = 1.73205, \quad \|S^\dagger\|_2 = 2.10766, \quad \|S\|_2\|S^\dagger\|_2 = 3.65057.$$

In Table 1 some results obtained in [3] and [5] are shown, among which Method 0, Method 1 and Method 2/3 are described in [3]—[5].

Table 1

Method	$(c_1, c_2, c_3)^T$	$\|c\|_2$	$\kappa_2(X)$	$\|F\|_2$	references
0	$\begin{pmatrix}1.43\\1.47\\1.79\end{pmatrix}$	2.7221	3.2732	16.46	[3]
1	$\begin{pmatrix}1.4732\\1.4253\\1.7894\end{pmatrix}$	2.72095	3.27316	16.4617	[5]
2/3	$\begin{pmatrix}1.59\\1.41\\1.79\end{pmatrix}$	2.7785	6.2827	16.54	[3]

We have calculated this example on a L-340 Computer in single precision with Method (I) and Method (II).

In Table 2 some results are shown, together with the number of iterations required for convergence to the termination scalar $s=10^{-3}$. We choose $w^{(0)}=(w_1^{(0)T}, w_2^{(0)T}, w_3^{(0)T})^T$ with $w_1^{(0)}=(1, 0)^T$, $w_2^{(0)}=(0, 1)^T$ and $w_3^{(0)}=\frac{1}{\sqrt{2}}(1, -1)^T$ as an initial vector. The symbols δ_j and γ_{ij} are defined by (3.8) and (4.0.4), respectively.

Table 2

Method		$(c_1, c_2, c_3)^T$	$\|c\|_2$	$\varkappa_2(X)$	$\|F\|_2$	Noumber of iterations
(I)	$\delta_1=1/3$ $\delta_2=1/3$ $\delta_3=1/3$	$\begin{pmatrix}1.39456\\1.39478\\1.78934\end{pmatrix}$	2.66307	3.27932	11.7000	4
(I)	$\delta_1=0.1$ $\delta_2=0.1$ $\delta_3=0.8$	$\begin{pmatrix}1.39470\\1.39464\\1.78934\end{pmatrix}$	2.66307	3.27371	11.8064	4
(II) $\alpha^2=0.7$ $\omega=2.75$	$\gamma_{12}=1/\sqrt{3}$ $\gamma_{13}=1/\sqrt{3}$ $\gamma_{23}=1/\sqrt{3}$	$\begin{pmatrix}1.67682\\1.67716\\1.78934\end{pmatrix}$	2.97091	3.36093	17.5198	15
(II) $\alpha^2=0.9$ $\omega=2.75$	$\gamma_{12}=\sqrt{0.45}$ $\gamma_{13}=\sqrt{0.45}$ $\gamma_{23}=\sqrt{0.1}$	$\begin{pmatrix}1.0\\1.78934\\1.78934\end{pmatrix}$	2.72093	3.27317	12.6006	22

Some computed solutions $X=(x_1, x_2, x_3)$ and F are given as follows.

With Method (I) and with weights $\delta_1=\delta_2=\delta_3=1/3$, after four iterations we obtain

$$X=\begin{pmatrix}0.88322 & -0.50356 & -1.12971\\ 0.46861 & 0.12788 & 0.35651\\ 0.11429 & 0.86170 & 0.22162\end{pmatrix},\quad F=\begin{pmatrix}2.44276 & -6.22520 & 2.24393\\ -3.78554 & 7.44796 & -4.27650\end{pmatrix},$$

where

$$\|x_1\|_2=1.00635,\quad \|x_2\|_2=1.00620,\quad \|x_3\|_2=1.20518.$$

With Method (II) and with weights $\gamma_{12}=\gamma_{13}=\sqrt{0.45}$, $\gamma_{23}=\sqrt{0.1}$, taking $\alpha^2=0.9$ and $\omega=2.75$, after 22 iterations we obtain

$$X=\begin{pmatrix}0.31274 & -0.86113 & -0.93801\\ 0.49437 & -0.21255 & 0.29482\\ 0.81104 & 0.46182 & 0.18228\end{pmatrix},\quad F=\begin{pmatrix}-5.34409 & 10.30390 & -4.82933\\ -0.37223 & 0.27808 & -1.19998\end{pmatrix},$$

where $\|x_1\|_2=\|x_2\|_2=\|x_3\|_2=1$, $\|X^TX-I\|_F=1.1727$ and $\|X^TX-I\|_2=0.8293$.

Computational practice shows that the choice of an appropriate initial vector $w^{(0)}$ is important for Method (I) (see Remark 3.1). It is worth while to point out that one can use Method (II) to produce a reasonable initial vector for the use of Method (I). For example, if we choose $w^{(0)}=(w_1^{(0)T}, w_2^{(0)T}, w_3^{(0)T})^T$ with $w_1^{(0)}=w_2^{(0)}=w_3^{(0)}$

$=\frac{1}{\sqrt{2}}(1, 1)^T$ as the initial vector, then we find that the 3×2 matrix $X_3^{(0)}=(S_1w_1^{(0)}, S_2w_2^{(0)})$ is not of full column rank. In fact, $X_3^{(0)}$ has singular values $\{\sqrt{2}, 0\}$. Consequently, in this case it is impossible to compute the gradient vector $\nabla f(w^{(0)})$ and to use Method (I). But from the above mentioned initial vector $w^{(0)}$, with Method (II) (take $\gamma_{12}=\gamma_{13}=\gamma_{23}=1/\sqrt{3}$, $\alpha^2=0.7$ and $\omega=2.75$) after two iterations we get $w^{(2)}=(w_1^{(2)T}, w_2^{(2)T}, w_3^{(2)T})^T$, in which

$$w_1^{(2)}=\begin{pmatrix} 0.664007\\ -0.747726\end{pmatrix},\quad w_2^{(2)}=\begin{pmatrix}0.900583\\0.434683\end{pmatrix},\quad w_3^{(2)}=\begin{pmatrix}0.912127\\0.409907\end{pmatrix}.$$

Then, taking $w^{(2)}$ as a new initial vector, with Method (I) after seven iterations we obtain a good solution

$$X=\begin{pmatrix} 0.54844 & 1.11890 & -1.28961\\ -0.13971 & 0.59312 & 0.40691\\ -0.93949 & 0.14354 & 0.25292\end{pmatrix},\quad F=\begin{pmatrix}-4.49346 & 8.00917 & -3.83710\\ -3.65376 & 7.73998 & -4.18704\end{pmatrix},$$

where $\|x_1\|_2=1.09679$, $\|x_2\|_2=1.27449$ and $\|x_3\|_2=1.37573$. The corresponding condition numbers $c_1=1.39408, c_2=1.39526$ and $c_3=1.78934$. The computed feedback matrix F has norm $\|F\|_2=13.7656$.

The following Table 3—Table 5 demonstrate how the convergence of the P-SOR algorithm depends upon ω, α^2 and $w^{(0)}$. Where K denotes the number of iterations required for convergence to the termination scalar $s=10^{-3}$. The weights are $\gamma_{12}=\gamma_{13}=\gamma_{23}=1/\sqrt{3}$.

Table 3 $\alpha^2=0.7$, $w_1^{(0)}=\begin{pmatrix}1\\0\end{pmatrix}$, $w_2^{(0)}=\begin{pmatrix}0\\1\end{pmatrix}$, $w_3^{(0)}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$

ω	0.75	1.0	1.5	2.0	2.5	2.75	3.0	3.25
K	67	60	46	34	23	15	20	35

Table 4 $\omega=2.75$, $w_1^{(0)}=\begin{pmatrix}1\\0\end{pmatrix}$, $w_2^{(0)}=\begin{pmatrix}0\\1\end{pmatrix}$, $w_3^{(0)}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$

α^2	0.6	0.7	0.8	1.0	1.2	1.4	1.6	1.8	2.0
K	40	15	26	38	46	52	57	61	63

Table 5 $\omega=2.75$, $\alpha^2=0.7$

$w_1^{(0)T}$	(1, 0)	(1, 0)	$\frac{1}{\sqrt{2}}(1, 1)$	$\frac{1}{\sqrt{2}}(-1, 1)$	(0, 1)	(1, 0)
$w_2^{(0)T}$	(0, 1)	(0, 1)	$\frac{1}{\sqrt{2}}(1, 1)$	$\frac{1}{\sqrt{2}}(1, -1)$	(0, 1)	(1, 0)
$w_3^{(0)T}$	$\frac{1}{\sqrt{2}}(1, -1)$	$\frac{1}{\sqrt{2}}(1, 1)$	$\frac{1}{\sqrt{2}}(1, 1)$	$\frac{1}{\sqrt{2}}(-1, 1)$	(0, 1)	(1, 0)
K	15	19	19	13	23	14

References

[1] J. R. Bunch, C. P. Nielsen, Updating the singular value decomposition, *Numer. Math.*, **31** (1978), 111—129.
[2] L. R. Fletcher, An inverse eigenvalue problem from control theory, Numerical Treatment of Inverse Problems for Differential and Integral Equations (ed. P. Deuflhard, E. Hairer), Birkhauser/Boston, 1983, 161—170.
[3] J. Kautsky, N. K. Nichols, P. Van Dooren, L. Fletcher, Numerical methods for robust eigenstructure assignment in control system design, *ibid.*, 171—178.
[4] J. Kautsky, N. K. Nichols, P. Van Dooren, Robust pole assignment in linear state feedback, *Int. J. Control*, **41** (1985), 1129-1155.
[5] J. Kautsky, N. K. Nichols, Robust multiple eigenvalue assignment by state feedback in linear systems, Numerical Analysis Report 7/84 (1984), Department of Mathematics, University of Reading.
[6] D. G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Company, Inc., 1973.
[7] J. G. Sun, Eigenvalues and eigenvectors of a matrix dependent on several parameters, *J. Comp. Math.*, **3** (1985), 351—364.
[8] J. G. Sun, An algorithm for the solution of multiparameter eigenvalue problems (I), *Math. Numer. Sinica*, **8** (1986), 137—149.
[9] J. G. Sun, On measures of robustness of a control system, *Math. Numer. Sinica*, to apper.
[10] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.