

A NUMERICAL ALGORITHM FOR
THE MOST ECONOMICAL STRUCTURAL SYNTHESIS
OF LINEAR CONTROL SYSTEMS

by

DR. KING-WAH ERIC CHU

NUMERICAL ANALYSIS REPORT NO. 4/84

Department of Mathematics,
University of Reading,
Whiteknights,
READING,
Berkshire, U.K.,
RG6 2AX.

ACKNOWLEDGEMENT

I wish to thank my wife Kimba for her support and encouragement.

I also wish to thank Dr. Paul Van Dooren of the Philips Research Laboratory, Brussels, for several useful comments on the work.

ABSTRACT

Given a time-invariant state matrix A , the Most Economical Structural Synthesis (MESS) calls for the construction of the controller matrix B and observer matrix C such that (I) the matrices B and C are as sparse as possible and (II) the resulting linear time-invariant control systems are completely controllable and completely observable.

A numerically stable algorithm for the MESS problem is suggested. Three numerical examples are given.

Generalisations of MESS are also discussed.

1. INTRODUCTION

Consider the linear time-invariant control system described by the equations

$$\begin{cases} \dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u} & (1.1a) \\ \underline{y} = \underline{C}\underline{x} & (1.1b) \end{cases}$$

where \underline{x} is an n state vector;

\underline{u} an r input vector;

\underline{y} an m output vector;

A an $n \times n$ state matrix;

B an $n \times r$ controller matrix

and C an $m \times n$ observer matrix .

Given an A , the Most Economical Structural Synthesis (MESS) calls for the construction of a B and a C such that

(I) the system (1.1) is completely controllable and completely observable,

and (II) the matrices B and C are as sparse as possible.

Because of duality, one only needs to consider the MESS of B .

The concept of MESS was first introduced by Tu [11], [12]. If one wants to minimise the costs of investment and maintenance for the controller and observer mechanisms, the following indices have to be minimised:

$$\$B = \sum_{i,j=1}^{n,r} \mu_{ij} \cdot \text{sym}(b_{ij}) \rightarrow \min \quad (1.2)$$

$$\text{and } \$C = \sum_{i,j=1}^{m,n} \eta_{ij} \cdot \text{sym}(c_{ij}) \rightarrow \min \quad (1.3)$$

where $B = (b_{ij})$, $C = (c_{ij})$, μ_{ij} and η_{ij} are positive cost coefficients for b_{ij} and c_{ij} respectively, and

$$\text{sym}(x) = \begin{cases} 1, & \text{if } x \neq 0; \\ 0, & \text{if } x = 0. \end{cases}$$

Simplify the problem by assuming that all cost coefficients are equal, i.e.

$$\mu_{ij} = \mu ; \quad \eta_{ij} = \eta ; \quad \forall i,j ;$$

the indices in (1.2) and (1.3) become

$$\$B = \mu \cdot \sum_{i,j} \text{sym}(b_{ij}) = \mu \cdot N_B \quad (1.4)$$

and

$$\$C = \eta \cdot \sum_{i,j} \text{sym}(c_{ij}) = \eta \cdot N_C , \quad (1.5)$$

where N_B and N_C are the numbers of non-zero elements in B and C respectively.

(The case when μ_{ij} and η_{ij} are not constant is called the Generalised MESS problem (GMESS) and will be discussed in §6.)

(1.4) and (1.5) indicate that making B and C as sparse as possible will minimise costs. In other words, MESS means that the number of controller and observer units are kept to the minimum and yet the complete controllability and complete observability properties of the system are preserved. Note also that $\max\{\mu_{ij}\} \cdot N_B \geq \sum \mu_{ij} \cdot \text{sym}(b_{ij})$ and minimising N_B is thus equivalent to minimising an upper bound of the cost functional $\$B$ in (1.2). (A similar observation holds for N_C).

In Tu [12], a method of the MESS was introduced. Another algorithm was suggested by Cheng and Zhang [3]. Both of these algorithms are not suitable for computer implementations. (Chen and Zhang's algorithm requires the transformation of A into Jordan canonical form and the rank-determinations by inspections of determinants. See §3).

The aim of this paper is to suggest a numerically stable algorithm for the MESS problem.

The MESS problem for the descriptor system

$$J\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u}, \quad y = \underline{C}\underline{x} ;$$

where J may be singular, will be discussed in another paper.

In §2, a brief summary of theoretical results concerning the MESS is contained. Chen and Zhang's algorithm is described in §3, with the help of a numerical example. A numerical algorithm for the MESS is suggested in §4 and some numerical results are given in §5. The GMESS of B and C is discussed in §6 and §7 concludes the paper.

2. SUMMARY OF THEORETICAL RESULTS ON MESS

(Most of the results in this section were quoted from Cheng and Zhang [3]). The problem of the MESS of B is equivalent to finding a B which is as sparse as possible and that

$$\text{rank } \langle A/B \rangle = \text{rank } [B, AB, \dots, A^{n-1}B] = n ; \quad (2.1a)$$

or
$$\text{rank } [\lambda I - A, B] = n, \quad \forall \lambda . \quad (2.1b)$$

(See Rosenbrock [9]. Numerical algorithms based on (2.1a) are unstable because of the powering of A; see Paige [7]. The algorithm in §4 is based on (2.1b)).

The existence of such a B is trivial. Consider $B = I_n$ and obviously (2.1) are satisfied. Partition all B's, which satisfy (2.1) and whose number of non-zero elements $N_B \leq n$, into equivalent classes with each class contains only those B's with the same N_B . There are only finite number of such classes (at most (n-1) of them) and the class with the smallest N_B contains the solutions of the MESS problem. So we have the following lemma:

LEMMA 2.1 There exists at least one solution to the problem of MESS of B.

Assume that $B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}^T$, with $N_B = 3$, be a solution to the MESS problem.

(M^T denotes the transpose of M). $(\lambda I - A)$ is not of full-rank for some λ and the column space of B has to make up the deficiency. However, $\tilde{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}^T$ spans the same column space as B and thus \tilde{B} is also a solution. But $N_{\tilde{B}} = 2$ and so B cannot be a solution. Based on these observations, a solution B to the MESS problem has at most one non-zero element on any row. It leads to the concept of a columnwise uni-non-zero matrix solution to the MESS problem.

DEFINITION 2.2 A columnwise uni-non-zero (CUNZ) matrix is a matrix with one and only one non-zero element on each column and at most one non-zero element on each row.

Note that a CUNZ matrix is always full-ranked. Concerning the CUNZ matrix solution to the MESS problem, we have the following theorem:

THEOREM 2.3 [3] There exists at least one CUNZ matrix solution to the MESS problem.

Proof: From Lemma 2.1, there exists at least one solution $B = (b_{ij})$, $i=1, \dots, n$; $j=1, \dots, r$.

$$\begin{aligned} \text{Construct } b_1^{(1)} &= [b_{11}, 0, \dots, 0]^T, & b_1^{(2)} &= [0, b_{21}, 0, \dots, 0]^T, \\ & \dots, & b_1^{(n)} &= [0, \dots, 0, b_{n1}]^T; \\ b_2^{(1)} &= [b_{12}, 0, \dots, 0]^T, & b_2^{(2)} &= [0, b_{22}, 0, \dots, 0]^T, \\ & \dots, & b_2^{(n)} &= [0, \dots, 0, b_{n2}]^T; \\ & \dots & \dots & \\ b_r^{(1)} &= [b_{1r}, 0, \dots, 0]^T, & b_r^{(2)} &= [0, b_{2r}, 0, \dots, 0]^T, \\ & \dots, & b_r^{(n)} &= [0, \dots, 0, b_{nr}]^T; \end{aligned}$$

and
$$\tilde{B} = [b_1^{(1)}, b_1^{(2)}, \dots, b_1^{(n)}; b_2^{(1)}, b_2^{(2)}, \dots, b_2^{(n)}; \dots; b_r^{(1)}, b_r^{(2)}, \dots, b_r^{(n)}].$$

It is clear that $\text{rank} \langle A/B \rangle \leq \text{rank} \langle A/\tilde{B} \rangle$. Since $\text{rank} \langle A/B \rangle = n$, one has $\text{rank} \langle A/\tilde{B} \rangle = n$; i.e. (A, \tilde{B}) is completely controllable. From \tilde{B} , delete zero columns and redundant multiples to form \hat{B} . From the construction, all the columns of \hat{B} are with one and only one non-zero element and no two columns in \hat{B} are the same and so there is at most one non-zero element on each row. Thus \hat{B} is a CUNZ matrix solution to the MESS problem.

Q.E.D.

Let t_1 be the minimum number of non-zero elements in B for the MESS problem. Let α_1 be the maximum number of eigenvectors any eigenvalue λ_1 of A has. It is clear from Theorem 2.3 that r , the number of columns of B , satisfies

$$0 < \alpha_1 \leq r \leq t_1 \tag{2.2}$$

The numerical algorithm suggested in §4 will produce a CUNZ matrix solution B , which has $r=t_1$ columns, with values of the non-zero elements all equal to unity. A corresponding α_1 -columns solution can be obtained from the CUNZ solution easily. (See step (V) in the example in §3).

3. THE CHEN AND ZHANG ALGORITHM [3]

Transform A into Jordan canonical form:

$$\begin{cases} T^{-1}AT = \text{diag} \{J_1, J_2, \dots, J_\sigma\} , & (3.1) \\ T^{-1}B = [\tilde{B}_1^T, \tilde{B}_2^T, \dots, \tilde{B}_\sigma^T]^T ; & (3.2) \end{cases}$$

where the $m_j \times m_j$ matrix J_j has the form

$$J_j = \text{diag} \{J_{j1}, J_{j2}, \dots, J_{j\alpha_j}\} \quad (3.3)$$

and $m_j \times r$ matrix \tilde{B}_j has the form

$$\tilde{B}_j = [\tilde{B}_{j1}^T, \tilde{B}_{j2}^T, \dots, \tilde{B}_{j\alpha_j}^T]^T . \quad (3.4)$$

Assume that A has σ eigenvalues and each eigenvalue λ_j has α_j eigenvectors. Thus J_{jk} in (3.3) can be a Jordan sub-block or just a scalar.

Further assume that the eigenvalues are arranged in such a way that

$$\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_\sigma .$$

It is well known that (1.1) is completely controllable if and only if

$$\text{rank} [\beta_{j1}^T, \beta_{j2}^T, \dots, \beta_{j\alpha_j}^T]^T = \alpha_j, \quad j=1, \dots, \sigma; \quad (3.5)$$

where β_{jk} is the last row of \tilde{B}_{jk} .

Let $T^{-1} = (t_{ij})$, and $\tilde{T}_j, j=1, \dots, \sigma$; be defined by

$$(\tilde{T}_j)_{k\ell} = t_{jk,\ell}, \quad \begin{matrix} k=1, \dots, \alpha_j; \\ \ell=1, \dots, n. \end{matrix}$$

(3.5) $\Leftrightarrow \tilde{T}_j B$ full-ranked for $j=1, \dots, \sigma$; which means there must be at least one non-singular $\alpha_j \times \alpha_j$ minor of \tilde{T}_j , for $j=1, \dots, \sigma$.

If $(v_1^j, v_2^j, \dots, v_{\alpha_j}^j)$ are the indices of the columns of \tilde{T}_j which form the non-singular minor, $e_{v_k^j}$, $k = 1, \dots, \alpha_j$; can be chosen as columns of B.

The steps in Chen and Zhang's algorithm can be illustrated with the following example [3]:

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Step (i). Transforming A into Jordan canonical form:-

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}; \quad T^{-1} = \begin{pmatrix} 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

$$J = T^{-1}AT = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

Step (ii). $\lambda_1 = 1$; $\lambda_2 = 2$; $\sigma = 2$.

$\alpha_1 = 2$; $\alpha_2 = 2$; (both eigenvalues have two eigenvectors).

$$\tilde{T}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}; \quad (2\text{nd \& } 3\text{rd rows of } T^{-1}).$$

$$\tilde{T}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}; \quad (4\text{th \& } 5\text{th rows of } T^{-1}).$$

Step (iii). For \tilde{T}_1 , 2nd and 5th columns, or 3rd and 5th columns form a nonsingular minor.

For \tilde{T}_2 , 1st and 3rd columns form a nonsingular minor.

$$v_1^1 = 2, v_2^1 = 5; \text{ (or } v_1^1 = 3, v_2^1 = 5;)$$

$$v_1^2 = 1, v_2^2 = 3.$$

Step (iv). For the first set of v 's,

$$B = [e_1, e_2, e_3, e_5], N_B = 4.$$

For the second set of v 's,

$$B = [e_1, e_3, e_5], N_B = 3.$$

Obviously, the second set of v 's gives the CUNZ matrix solution to the MESS problem.

The example indicates that one has to choose the v 's in such a way that N_B , the number of distinct elements in $U = \bigcup_{j,k} \{v_k^j\}$, should be as small as possible.

Step (v). Obviously, B can be chosen to be $B = \begin{pmatrix} b_1 & 0 & 0 & 0 & b_2 \\ 0 & 0 & b_3 & 0 & 0 \end{pmatrix}^T$ which is a α_1 - columns MESS solution. For some cases, the b 's should be chosen with care to make sure that (1.1) is controllable.

(Using the language in [4], the system (A,B) is potentially controllable. By choosing the values of b 's carefully, the system will be completely controllable).

(e.g. If $\text{span}(A-2I) = \text{span}[e_1 + e_5, e_2, e_4]$, $B = [e_1 + e_5, e_3]$ will not be satisfactory; but $B = [e_1 - e_5, e_3]$ will be).

In this example, any values for the b 's will do, as

$$\text{span}(A-2I) = \text{span}[e_2, e_4, e_5]$$

$$\text{and } \text{span}(A-I) = \text{span}[e_1, e_2 - e_3, e_4].$$

The above algorithm is not numerically robust enough for the following reasons:

- (1) Transformation of a matrix to Jordan canonical form in step (i) is not numerically stable, especially if A has defective eigen-structures.

- (2) The indices σ and α_j 's in step (ii) may not be determined accurately for ill-conditioned eigenvalues.
- (3) The inspection of determinants in step (iii) may not be adequate for determining whether a certain minor is singular or not. In addition, a non-singular minor with a small eigenvalue may force the resulting system near an uncontrollable one.

Finally, T^{-1} in the above example is quite sparse and consequently step (iii) can be carried out with ease. It will not be the case for a general A .

In the next section, a numerical algorithm for the MESS problem is suggested. Only stable decompositions like the singular value decomposition (SVD) or QR will be used.

4. THE NUMERICAL ALGORITHM

Observing the example in §3, the problem of finding a CUNZ matrix solution to the MESS problem is reduced to finding the indices $\{v_k^j; j=1, \dots, \sigma; k=1, \dots, \alpha_j\}$.

We work from the criterion (2.1b) and observe that (2.1b) is equivalent to

$$(\text{span}(\lambda I - A))^\perp \cap N((\lambda I - A)^H) \subset \text{span}(B), \quad \forall \lambda; \quad (4.1)$$

where $N(\cdot)$ denote the corresponding right null-space and M^H the Hermitian transpose of M .

Note that $N((\lambda I - A)^H) = \{0\}$ unless $\lambda = \lambda_j$, an eigenvalue of A . As a result, we are looking for a B such that

$$N((\lambda_j I - A)^H) \subset \text{span}(B), \quad j=1, \dots, \sigma. \quad (4.2)$$

Note also that dimension of $N((\lambda_j I - A)^H)$ satisfies

$$\dim\{N((\lambda_j I - A)^H)\} = \alpha_j, \quad j=1, \dots, \sigma. \quad (4.3)$$

It is clear that $v_k^j, k=1, \dots, \alpha_j$; for some j , can be chosen such that

$$B = (e_{v_k^j}, j=1, \dots, \sigma; k=1, \dots, \alpha_j) \quad (4.4)$$

satisfies (4.2).

There may be more than one possible set of feasible v 's, as shown in step (iii) in the example in §3. As a result, we try all possible combinations

of v_k^j for some j , store the angles θ_j between $\text{span}(e_{v_k^j})$ and $R(\lambda_j I - A) = N[(\lambda_j I - A)^H]^\perp$ and choose the v 's in such a way that

- (a) $\sin \theta_j$ should be greater than a prescribed positive tolerance ($\sin \theta_{\min}$; see step 1 below), to avoid making (1.1) uncontrollable or near to an uncontrollable system;
 - (b) $\sum_j \sin^2 \theta_j$ is maximized;
- and
- (c) $N_B = N(\cup_{j,k} \{v_k^j\})$ is minimized.

($N(S)$ denotes the number of distinct elements in the set S and $\{\cdot\}^\perp$ the orthogonal complement; $R(\cdot) \triangleq \text{span}(\cdot)$.)

By decomposing $\text{span}(B)$ into the two components $R(\lambda_j I - A)$ and $\{R(\lambda_j I - A)\}^\perp$, observe that $\theta_j > 0 \iff \text{span}(B)$ is not deficient in $\{R(\lambda_j I - A)\}^\perp \iff (4.2)$.

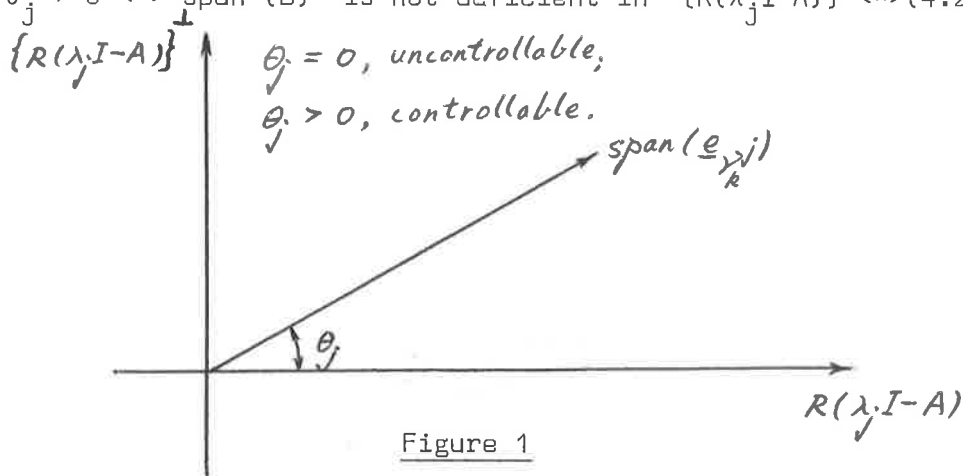


Figure 1

By using the SVD or QR, the range, null-space and angles between various subspaces can be calculated easily and in a numerically stable manner. The SVD can also be used for rank-determinations. ([5]).

The numerical algorithm is summarized as follows:-

Step (1):- Input A , $\sin\theta_{\min}$, $\text{eps } 1$, $\text{eps } 2$.

Step (2):- Find the eigenvalues of A using a numerically stable method.
(e.g. QR; see [13]).

Rearrange the eigenvalues into $\{\lambda_1, \dots, \lambda_\sigma\}$. (Any eigenvalues which are differed in no more than $\text{eps } 1$ may be considered to be multiple and their arithmetic mean can be used to construct the σ_j 's; but see comment (ii) on page 12).
For $j=1, \dots, \sigma$; do steps (3) and (4).

Step (3):- Find the SVD of $(\lambda_j I - A)$, i.e.

$$\lambda_j I - A = (U_{j1}, U_{j2}) \cdot \begin{bmatrix} \Sigma_{n-\alpha_j} & 0 \\ 0 & \Sigma_{\alpha_j} \end{bmatrix} \cdot \begin{bmatrix} V_{j1}^H \\ V_{j2}^H \end{bmatrix} = UDV^H, \quad (4.5)$$

where U and V are $n \times n$ unitary matrices; $D = \text{diag}\{d_1, \dots, d_n\}$ is an $n \times n$ diagonal matrix and Σ_{α_j} and $\Sigma_{n-\alpha_j}$ are $\alpha_j \times \alpha_j$ and $(n-\alpha_j) \times (n-\alpha_j)$ diagonal matrices respectively.

Let the diagonal elements of D be arranged in descending order and

$$(\Sigma_{n-\alpha_j})_{n-\alpha_j} > d_1 \cdot \text{eps}2$$

and

$$(\Sigma_{\alpha_j})_1 \leq d_1 \cdot \text{eps}2. \quad (4.6)$$

(4.6) implies that $(\lambda_j I - A)$ is of $\text{eps}2$ - numerical rank $(n-\alpha_j)$.

Note that

$$\left\{ \begin{array}{l} R(\lambda_j I - A) = \{N((\lambda_j I - A)^H)\}^\perp = \text{span}(U_{j1}); \\ \{R(\lambda_j I - A)\}^\perp = N((\lambda_j I - A)^H) = \text{span}(U_{j2}); \\ N(\lambda_j I - A) = \{R((\lambda_j I - A)^H)\}^\perp = \text{span}(V_{j2}); \\ \{N(\lambda_j I - A)\}^\perp = R((\lambda_j I - A)^H) = \text{span}(V_{j1}). \end{array} \right. \quad (4.7)$$

Step (4):- Let θ_{jk} be the angle between

$N((\lambda_j I - A)^H)^\perp$ and $\text{span} \{e_{k_\ell}, \ell=1, \dots, j\}$;

for each of the $n C_{\alpha_j}$ combinations of $\underline{k} = (k_\ell, \ell=1, \dots, \alpha_j)$,

calculate $M_{jk} = (e_{k_1}, \dots, e_{k_{\alpha_j}})^T U_{j2}$ and $\sin \theta_{jk} = \sigma_{\alpha_j}(M_{jk})$, (4.8)

where $\sigma_{\alpha_j}(M_{jk})$ is the smallest singular value of M_{jk} . $\sin \theta_{jk}$ can be viewed as a distance between the subspaces $\text{span} \{e_{\underline{k}}\}$ and $N((\lambda_j I - A)^H)^\perp$. (See [1]).

Store the \underline{k} 's so that the combinations of indices corresponding to the larger $\sin \theta_{jk}$ appear earlier and discard any \underline{k} when $\sin \theta_{jk} < \sin \theta_{\min}$; i.e. one ignores any \underline{k} which gives rise to $e_{\underline{k}}$, which in turn has span nearly orthogonal to $N((\lambda_j I - A)^H)^\perp$. Note that M_{jk} is just the \underline{k} rows of U_{j2} .

Step (5):- Choose the indices v_k^j . The indices can be chosen by inspecting the indices \underline{k} stored in step (4) such that $N_B = N(U_{j,k} \{v_k^j\})$ is minimised.

The CUNZ matrix solution B can be constructed as

$$B = [e_{v_k^j}, j=1, \dots, \sigma; k=1, \dots, \alpha_j]$$

with repetitions deleted.

A more algorithmic presentation of the above algorithm, written in MATLAB [5], is included as an appendix to this paper.

Some practical details concerning the computer implementation of the algorithms are:-

- (i) θ_{\min} needs only to be non-zero theoretically but it should be reasonably large so that (A, B) will not be close to an uncontrollable system. (e.g. The n -th singular value of $(\lambda I - A, B)$ is non-zero but very small for some λ).

θ_{\min} should not be too large as it will reduce the number of feasible \underline{k} in step (4) and thus increase N_B . Values of $\sin\theta_{\min}$ between 0.1 and 0.3 were used in the numerical examples in §6; experimenting with different values of $\sin\theta_{\min}$ may be necessary in general. (See also §5).

(ii) The determination of σ and α_1 are difficult [5], [10] and should be avoided. Note that only a good estimate of $\sin\theta_{\min}$ is required and a reasonably accurate $N((\lambda_j I - A)^H)$ will provide such an estimate. From [10], it is known that perturbing A slightly will only perturb $N((\lambda_j I - A)^H)$ slightly. Thus it will be safer to perturb A slightly so that all the λ_j 's are simple and the null spaces one dimensional. In this case, steps (3) and (4) have to be carried out for $j=1, \dots, n$; and more work is required in the search in step (5). (See Example 2 in §6).

(iii) eps_2 , the tolerance for numerical rank determinations should be set near to machine accuracy.

Note that some users may prefer to replace (4.6), which takes account of the scaling of $\lambda_j I - A$, by

$$\left(\sum_{n-\alpha_j}^{n-\alpha_j}\right)_{n-\alpha_j} > \text{eps}_2 \text{ and } \left(\sum_{\alpha_j}^{\alpha_j}\right)_1 < \text{eps}_2.$$

(iv) Step (3) can be carried out using either QR or SVD. SVD's have been used in the programs in the appendix because it gives out more information.

In most applications, the extra information is not required and the cheaper QR should be used.

(v) Producing the indices \underline{k} in step 4 for the ${}^n C_{\alpha_j}$ combinations of α_j elements from a total of n can be easily done, using the algorithms in [8]. The Revolving Door algorithm produces a sequence of \underline{k} 's in which

successive indices differ in only one component.

Let \underline{k}_1 and \underline{k}_2 be two successive sets of indices. $M_{j\underline{k}_1}$ and $M_{j\underline{k}_2}$ will differ in only one row and the updating SVD [12] or QR technique may be used to find the smallest singular value of $M_{j\underline{k}_2}$ more efficiently from the available information on $M_{j\underline{k}_1}$.

(vi) There may exist more than one MESS solution and the optimal one is chosen by maximizing $\sum \sin^2 \theta_{j\underline{k}}$. (Recall §4, and see also §5).

(vii) Some particular combinations of rows of U_{j2} may be ill-conditioned in the sense that the rows are nearly linearly dependent. Such combinations may be recorded during step (4) and will not be chosen to compose \underline{k} . Less searching will be needed in step (5).

5. GENERALISED MESS

In (1.4), μ_{ij} are assumed to be constant for all (i,j) and N_B is minimised to yield the MESS solution of B . One can delete this restriction and call the corresponding synthesis of B for a general set of cost coefficients $\{\mu_{ij}\}$ the generalised MESS (GMESS) of B .

It is then possible to replace an expensive controller unit by several cheap ones, in spite of the fact that N_B has been increased.

Instead of N_B in (1.4), one can minimise the functional

$$F_B = \omega_0 \cdot \sum_{i,j=1}^{n,r} \mu_{ij} \cdot \text{sym}(b_{ij}) + \sum_{j=1}^{\sigma} \omega_j \cdot \cos^2 \theta_{j\underline{k}}, \quad (5.1)$$

where $\omega_j, j=0,1,\dots,\sigma$; are non-negative weights and $\theta_{j\underline{k}}$ is the angle between $\text{span}(\underline{e}_{\underline{k}})$ and $\text{span}(U_{j2})$. (Recall (4.5) to (4.8)).

It is assumed that B is a α_1 -column GMESS solution and thus $r = \alpha_1$. The first term in (5.1) minimises the costs. The second term is essential when more than one GMESS solutions with the same cost exist. The second term will

pick the GMESS solution B with its range space lies nearer to that of U_{j2} , the null-space of $(\lambda_j I - A)^H$, and thus force the resulting system further away from an uncontrollable one. This situation arises quite often, especially with $\mu_{ij} = \mu$, and the resulting functional

$$\tilde{F}_B = \omega_0 N_B + \sum_{j=1}^{\sigma} \omega_j \cos^2 \theta_{jk} \quad (5.2)$$

has been used in the MESS programs in the appendix.

If one chooses $\omega_0 = 1$ and $\omega_j = 0, j > 0$, one returns to the original situation in (1.4). If one chooses $\omega_0 = 0$, the costs will be neglected and one concentrates on the conditioning of the system. As costs are the main concern in the MESS or the GMESS, $(\omega_j, j > 0)$ should be chosen to be quite small when compared to $\omega_0 \mu_{ij}$; e.g.

$$\omega_0 \mu_{ij} = O(10) \quad \text{and} \quad \omega_j = O(1), j > 0 \quad . \quad (5.3)$$

In (5.2) for the MESS of B , the values

$$\omega_0 = 1; \quad \omega_j = 1/2\sigma, \quad j > 0; \quad (5.4)$$

can be used. Note that the second term in (5.2) can be at most $\frac{1}{2}$ and thus the first term is always dominant. The situation when N_B is increased by one with a decrease in the value of \tilde{F}_B can never occur.

Recall that the prescribed tolerance $\sin \theta_{\min}$ has been used in §4 to filter out solutions which yield systems which are close to uncontrollable ones. One may choose $\sin \theta_{\min}$ to be very small and $\omega_j \approx \omega_0, j > 0$; to achieve the same goal. General ω_j 's other than those suggested in (5.3) and (5.4) give rise to a different problem and more analysis is required.

F_B in (5.1) can be minimised by searching through all possibilities, as in §4. Note that rearranging the columns of B yields different GMESS solutions and changes the corresponding values of F_B .

For the sake of simple presentation, the programs for the GMESS of B are not included in the appendix.

Finally, it is unlikely that a system is designed based on only the MESS of B. However, it may be helpful to the designer, who has the MESS in mind, to use the suggested algorithm to generate a series of B's by choosing different tolerances or functionals to minimise and choose one using additional information. The algorithm can also be modified easily to cope with constraints, like $b_{ij} = 0$ for some components b_{ij} in B.

6. NUMERICAL EXAMPLES

Example 1. Consider the example in §3 [3]:-

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrix is lower Hessenberg and has eigenvalues $\{1,1,1,2,2\}$, with 2 eigenvectors for $\lambda = 2$, but only 2 eigenvectors for $\lambda = 1$.

The tolerances are chosen to be :

$$\text{eps1} = \text{eps2} = \text{eps}$$

and $\text{minsin} = 0.2$,

where eps is the machine accuracy of the NORD-500 in the University of Reading Computer Centre. ($\text{eps} \approx 5 \times 10^{-16}$; see [5]).

The program in the Appendix grouped the eigenvalues correctly into two multiple eigenvalues of 1 and 2 ($\sigma = 2$) and determined $\alpha_1 = \alpha_2 = 2$ successfully. (Steps (3) and (4) in §4).

The same answer as in §3 was produced:

$$\begin{aligned} v_1^1 &= 3, & v_2^1 &= 5; \\ v_1^2 &= 1, & v_2^2 &= 3; \end{aligned}$$

and $N_B = 3$.

Example 2. Construct a matrix T such that

$$T = \begin{pmatrix} -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1.1 \end{pmatrix}$$

and consider $\tilde{A} = TAT^{-1}$, with A given in Example 1.

The similarity transform T makes \tilde{A} more ill-conditioned than A . In addition, the structure of the null-space $N[(\lambda I - A)^H]$ is relatively simple because A is in lower Hessenberg form. The structure of $N[(\lambda I - \tilde{A})^H]$ will be more complicated.

The following three sets of tolerance are chosen:

$$\begin{aligned} \text{eps1} = \text{eps2} = & \text{(i) } \text{eps} \\ & \text{(ii) } 0.5 \times 10^{-8} \\ & \text{(iii) } 0.5 \times 10^{-6} \end{aligned}$$

and $\text{minsin} = 0.2$.

With (i), the tolerance eps1 was too small and, because of round-off error, the program failed to group the eigenvalues $\{1, 1, 1, 2, 2\}$ into two multiple eigenvalues of 1 and 2. Instead, all eigenvalues are considered to be simple. ($\sigma = 5$ instead of 2).

With (ii), the eigenvalues are grouped into four eigenvalues $\{1, 1, 1, 2\}$ (plus some round-off error). ($\sigma = 4$ instead of 2).

With (i) and (ii), some operations in steps (3) and (4) (§4) were repeated because of the failure to determine the multiplicities of the eigenvalues correctly. The amount of searching involved in step (5) was also increased enormously, because of the wrong value of σ and a small eps2 . Apart from the above defects, (i) and (ii) produced the same answers as (iii):-

With $\text{eps1} = \text{eps2} = 0.5 \times 10^{-6}$, the eigenvalues were grouped correctly, with $\lambda_1 = 1$, $\lambda_2 = 2$ and $\alpha_1 = \alpha_2 = 2$.

There are two feasible sets of v_k^j :

$$\begin{cases} v_1^1 = 3, v_2^1 = 4; \\ v_1^2 = 3, v_2^2 = 4; \end{cases} \quad \text{and} \quad \begin{cases} v_1^1 = 1, v_2^1 = 2; \\ v_1^2 = 1, v_2^2 = 2; \end{cases}$$

with $N_B = 2$. For both sets of v_k^j , $\sum_{j=1}^{\sigma} \cos^2 \theta_{j\underline{k}}$ are the same (= 1.33) and thus both sets are optimal in the sense that (5.2) is minimised.

Example 3. Construct T such that

$$T_{ij} = \begin{cases} i \neq j, & i \neq j \\ 1, & i = j \end{cases}$$

and consider $\tilde{A} = TAT^{-1}$, with A given in Example 1.

The following three sets of tolerance are chosen:-

- mins in = (i) 0.1,
- (ii) 0.2,
- (iii) 0.4;

and $\text{eps1} = \text{eps2} = 0.5 \times 10^{-6}$.

The eigenvalues were grouped correctly ($\sigma = 2, \lambda_1 = 1, \lambda_2 = 2$) and the dimensions of the null-spaces determined successfully ($\alpha_1 = \alpha_2 = 2$).

With (i), 72 possibilities were searched through (in step (5) in §4) and 2 feasible sets of v_k^j were produced:

$$\begin{cases} v_1^1 = 2, v_2^1 = 3; \\ v_1^2 = 2, v_2^2 = 3; \end{cases} \quad \text{and} \quad \begin{cases} v_1^1 = 1, v_2^1 = 3; \\ v_1^2 = 1, v_2^2 = 3; \end{cases}$$

with $N_B = 2$ and $\sum_{j=1}^{\sigma} \cos^2 \theta_{j\underline{k}}$ both equal to 1.58. (There are 5 other sets of v_k^j which yield $N_B = 2$ and $\sum_{j=1}^{\sigma} \cos^2 \theta_{j\underline{k}} = 1.62$).

With (ii), only 35 possibilities had to be searched through because of a smaller mins in.

Two feasible sets of v_k^j were produced:

$$\begin{cases} v_1^1 = 3; & v_2^1 = 5; \\ v_1^2 = 3; & v_2^2 = 5; \end{cases} \quad \text{and} \quad \begin{cases} v_1^1 = 2, & v_2^1 = 5; \\ v_1^2 = 2, & v_2^2 = 5; \end{cases}$$

with $N_B = 2$ and $\sum_{j=1}^{\sigma} \cos^2 \theta_{j\underline{k}}$ both equal to 1.62. (These are the only two sets of v_k^j which yield $N_B = 2$).

With (iii), minsin was too large (0.4) and N_B was pushed to 3 instead of 2. Only 12 possibilities had to be tested.

7. CONCLUSIONS

A numerically stable algorithm for the Most Economical Structural Synthesis of B and C of a linear time-invariant system has been suggested. The tolerance (eps1, eps2 and minsin) have to be chosen carefully for individual systems. Suitable B (or C) can be chosen from a feasible set, possibly based on some additional information concerning the system.

A generalisation of MESS (GMESS in §5) has also been discussed, but more work has to be done.

Finally, for sparse A, the graph theory approach of Frankson, Falster and Evans [4] may be applicable to the MESS problem.

REFERENCES

- [1] Björck, A., and Golub, G.H. "Numerical Methods for Computing Angles Between Linear Subspaces", *Maths. Comp.* 27, 579-594 (1973).
- [2] Bunch, J.R., and Nielsen, C.P. "Updating The Singular Value Decomposition", *Numer. Math.* 31, 111-129 (1978).
- [3] Cheng, Zhaohuan, and Zhang, Rongxiang. "The Algebraical Method of Most Economical Structural Synthesis of Linear Control System", *Acta Automatica Sinica (China)*, Vol. 7, no. 3, July, 1981, pp. 171-178. (In Chinese).
- [4] Franksen, O.I., Falster, P., and Evans, F.J. "Qualitative Aspects of Large Scale Systems", *Lecture Notes in Control and Information Sciences*, Vol. 17, Springer-Verlag, 1979.
- [5] Golub, G.H., and Wilkinson, J.H. "Ill-conditioned Eigensystems and The Computation of The Jordan Canonical Form", *SIAM Review*, Vol. 18, No. 4, October 1976, pp. 578-619.
- [6] Moler, C.B. "MATLAB User's Guide", University of New Mexico, Dept. of Computer Science, 1981.
- [7] Paige, C.C. "Properties of Numerical Algorithms Related to Computing Controllability", *IEEE Trans. Autom. Control*, AC-26, 130-138, (1981).
- [8] Reingold, E.M., Nievergelt, J., and Deo, N. "Combinatorial Algorithms: Theory and Practice", Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
- [9] Rosenbrock, H.H. "State-Space and Multivariable Theory", Thomas Nelson & Sons, London 1970.
- [10] Stewart, G.W. "Error & Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems", *SIAM Review*, Vol. 15, No. 4, October 1973, pp. 727-764.
- [11] Tu, Xuyen. "Controllability, Observability and The Most Economical Structural Synthesis", *The First Chinese Control Theory and Its Applications Conference*, May 1978. The Chinese Scientific Press.
- [12] Tu, Xuyen. "The Problem of The Structural Synthesis of The Most Economical Control Systems", *Acta Automatica Sinica (China)*, Vol. 8, no. 2, April 1982, pp. 103-111. (In Chinese).
- [13] Wilkinson, J.H. "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford, 1965.

APPENDIX : MATLAB PROGRAMS FOR THE MESS PROBLEM

INFMESS-1 : Main-program for Example 2 in §6, with $\text{eps1} = \text{eps2} = \text{eps}$.

```

@ENTER SMS134WAKING,KIS,,10
@5 (SMS125)MLB
EXEC('MESINPUT');
EPS1=EPS;
EPS2=EPS1;
MINSIN=0.2;
FOR I=1:5, FOR J=1:5, TT(I,J)=1;
FOR I=1:5, TT(I,I)=-1;
TT(4,5)=-1; TT(5,5)=1.1;
SVD(A), COND(A)
SVD(TT), COND(TT)
A=TT*A/TT;
SVD(A), COND(A)
TT=<>; I=<>; J=<>;
A
EPS1, EPS2, MINSIN
EXEC('MESEIGEN');
LONG E
EV
SHORT E
EXEC('MESSIGMA');
SIGMA, EV
EXEC('MESLAMDA');
ALFA, MALFA
EXEC('MESINDX2');
VN
IDX2
EXEC('MESINDX3');
EXEC('MESINDEX');
EXEC('MESANSWR');
CLEAR
EXIT

```

MESINPUT : Input datas

```
***
...MESINPUT-MES.
***
...22ND APRIL 1983.
***
SHORT E#
A=< 2  0  0  0  0
    0  1 -1  0  0
    0  0  2  0  0
    1  0 -1  1 -1
    0  0  0  0  1>#
N=5#
EPS1=0.5D-6#EPS2=EPS1#
MINSIN=0.2#
***
...END OF MESINPUT-MES.
...27-7-1983.
```

MESEIGEN : Solve eigenvalue problem for A and sort eigenvalues λ_i into descending order.

```
***
...MESEIGEN-MES.
***
...22ND APRIL 1983.
***
EV=EIG(A)#
I=1#
IFLAG=1#
WHILE IFLAG<>0,...
WHILE I<=N-1,...
IFLAG=0#...
FOR J=N-1:-1:I,...
IF ABS(EV(J))<ABS(EV(J+1)) THEN ...
IFLAG=1#...
EV(<J,J+1>)=EV(<J+1,J>)#...
END#...
END#...
I=I+1#...
END#END#
I=<>#J=<>#IFLAG=<>#
***
...END OF MESEIGEN-MES.
...27-4-1983.
```

MESSIGMA : Calculate σ

```
***
...MESSIGMA-MES.
***
...22ND APRIL 1983.
***
SIGMA=1;
TEMP(1,1)=EV(1);
NTEMP(1)=1;
I=2;
WHILE I<=N,...
IF ABS(TEMP(SIGMA,1)-EV(I))<=EPS1 THEN ...
NTEMP(SIGMA)=NTEMP(SIGMA)+1;...
TEMP(SIGMA,NTEMP(SIGMA))=EV(I);...
ELSE SIGMA=SIGMA+1;...
NTEMP(SIGMA)=1;...
TEMP(SIGMA,1)=EV(I);...
END;...
I=I+1;...
END;
EV=<>;
FOR I=1;SIGMA,...
NNN=NTEMP(I);...
VEC(1;NNN)=TEMP(I,1;NNN);...
EV(I)=SUM(VEC)/NNN;...
VEC=<>;NNN=<>;...
END;
I=<>;TEMP=<>;NTEMP=<>;NNN=<>;VEC=<>;
***
...END OF MESSIGMA-MES.
...27-7-1983.
```

MESLAMDA : Calculate α_j and U_{j2}

```
***
...MESLAMDA-MES.
***
...22ND APRIL 1983.
***
<U,D,V>=SVD(EV(1)*EYE(N)-A);
ALFA(1)=N-RANK(D,(EPS2*D(1,1)));
SP=U;
FOR I=2;SIGMA,...
<U,D,V>=SVD(EV(I)*EYE(N)-A);...
ALFA(I)=N-RANK(D,(EPS2*D(1,1)));...
SP=<SP,U>;...
END;
MALF=NORM(ALFA,'INF');
U2(1;N,1;ALFA(1))=SP(1;N,N-ALFA(1)+1;N);
FOR I=2;SIGMA,...
TEMP(1)=(I-1)*MALF+1;...
TEMP(2)=TEMP(1)+ALFA(I)-1;...
TEMP(4)=I*N;...
TEMP(3)=TEMP(4)-ALFA(I)+1;...
U2(1;N,TEMP(1);TEMP(2))=SP(1;N,TEMP(3);TEMP(4));...
END;
I=<>;U=<>;D=<>;V=<>;
J=<>;TEMP=<>;SP=<>;
***
...END OF MESLAMDA-MES.
...27-7-1983.
```


MESINDX 2 : Form $IDX2$ which contains v_k^j , $\sin^2\theta_j$ and $\cos^2\theta_j$

```
***
...MESINDX2-MES.
***
...22ND APRIL 1983.
***
FOR J=1:SIGMA,...
ALJ=ALFA(J)†...
EXEC('MESINDX1-MES')†...
TEMP(1)=(J-1)*MALF+1†...
TEMP(2)=TEMP(1)+ALJ-1†...
TEMP(3)=(J-1)*(MALF+2)†...
TEMP(4)=TEMP(3)+ALJ+2†...
NVN=0†...
FOR K=1:NO,...
VEC(1:ALJ)=IDX1(K,1:ALJ)†...
MAT=U2(VEC,TEMP(1):TEMP(2))†...
VEC=◊†...
D=SVD(MAT)†...
IF D(ALJ)>=MINSIN THEN NVN=NVN+1†END†...
FOR L=1:ALJ,IDX2(K,TEMP(3)+L)=IDX1(K,L)†END†...
IDX2(K,TEMP(3)+ALJ+1)=D(ALJ)**2†...
IDX2(K,TEMP(3)+ALJ+2)=1.0-...
IDX2(K,TEMP(3)+ALJ+1)†...
END†...
IFLAG=1†...
FOR K=1:NO-1,...
WHILE IFLAG<>0,...
IFLAG=0†...
FOR L=NO-1:-1:K,...
IF IDX2(L,TEMP(3)+ALJ+1)<...
IDX2(L+1,TEMP(3)+ALJ+1) THEN ...
IFLAG=1†...
IDX2(<L,L+1>,TEMP(3)+1:TEMP(4))=...
IDX2(<L+1,L>,TEMP(3)+1:TEMP(4))†...
END†...
END†...
END†...
END†...
VN(J)=NVN†...
END†
VEC=◊;MAT=◊;D=◊;
IFLAG=◊;TEMP=◊;
ALJ=◊;NO=◊;
NVN=◊;
***
...END OF MESINDX2-MES.
...04-08-1983.
```

MESINDEX 1 : The Revolving Door algorithm

```
...MESINDEX1-MES.
...
...22ND APRIL 1983,
...
IDX1=<>
IF ALJ=1,NO=N;FOR I=1:N,IDX1(I,1)=I;END;END;
IF ALJ=1 THEN I=<>;RETURN;
IF ALJ>1 THEN ...
TEMP(1)=-1;NO=0;...
FOR I=1:ALJ,TEMP(I+1)=I;END;...
JJ=1;...
WHILE JJ<>0, ...
NO=NO+1;...
IDX1(NO,1:ALJ)=TEMP(2:ALJ+1)';...
JJ=ALJ;...
WHILE TEMP(JJ+1)=N-ALJ+JJ, ...
JJ=JJ-1;...
END;...
TEMP(JJ+1)=TEMP(JJ+1)+1;...
FOR I=JJ+1:ALJ, ...
TEMP(I+1)=TEMP(I)+1;...
END;...
END;
TEMP=<>;JJ=<>;
...
...END OF MESINDEX1-MES.
...04-08-1983.
```

MESINDEX 3 : Form $IDX3$ which contains a full list of all possible v_k^j

```
...
...MESINDEX3-MES.
...
...22ND APRIL 1983.
...
NIN=0;
FOR I=1:SIGMA,VEC(I)=1;END;
WHILE VEC(SIGMA)<=VN(SIGMA),...
NIN=NIN+1;...
IDX3(NIN,1:SIGMA)=VEC(1:SIGMA)';...
VEC(1)=VEC(1)+1;...
FOR I=1:SIGMA-1, ...
IF VEC(I)>VN(I) THEN ...
VEC(I)=1;VEC(I+1)=VEC(I+1)+1;...
END;...
END;...
END;
I=<>;VEC=<>;VN=<>;
...
...END OF MESINDEX3-MES.
...20-7-1983.
```

MESINDEX : Minimise F_B

```
***
...MESINDEX-MES.
***
...22ND APRIL 1983.
***
FMIN=N#
MINNB=N#
DISP('INDEX3, NB, F, MINNB AND FMIN :')
FOR IN=1#NIN#...
F=0#...
VEC1=IDX3(IN,1#SIGMA)#...
FOR J=1#SIGMA#...
TEMP(1)=(J-1)*(MALF+2)+1#TEMP(2)=TEMP(1)+ALFA(J)+1#...
VEC(J,1#ALFA(J)+2)=IDX2(VEC1(J),TEMP(1)#TEMP(2))#...
END#TEMP=<#...
EXEC('MESCOUNT-MES')#...
IF NB<=MINNB THEN ...
F=0#FOR J=1#SIGMA#F=F+VEC(J,ALFA(J)+2)#END#...
END#...
IF NB<MINNB THEN ...
MINNB=NB#IMIN=IN#FMIN=F#...
END#...
IF NB=MINNB THEN ...
IF F<FMIN THEN ...
MINNB=NB#IMIN=IN#FMIN=F#...
END#...
<VEC1,NB,F,MINNB,FMIN>#...
END#...
END#
IN=<#VEC=<#J=<#F=<#VEC1=<#
NIN=<#NB=<#
***
...END OF MESINDEX-MES.
...28-7-1983.
```

MESCOUNT : Count N_B , given v_k^j

```
***
...MESCOUNT-MES.
***
...22ND APRIL 1983.
***
NB=0;TEMP(1:N)=0.0*EYE(1,N);
FOR I=1:SIGMA,FOR J=1:ALFA(I),...
IND=TEMP(VEC(I,J));...
IF IND=0 THEN ...
TEMP(VEC(I,J))=1;NB=NB+1;...
END;...
END;...
END;
***
...MESCOUNT-MES.
...03-08-1983
```

MESANSWR : Output answers

```
***
...MESANSWR-MES.
***
...22ND APRIL 1983.
***
VEC1=IDX3(IMIN,1:SIGMA);
FOR J=1:SIGMA,TEMP=(J-1)*(MALF+2);...
FOR K=1:MALF,...
IF K<=ALFA(J) THEN ...
OUT(J,K)=IDX2(VEC1(J),TEMP+K);...
END;...
IF K>ALFA(J) THEN ...
OUT(J,K)=0;...
END;...
END;...
END;
EV
SIGMA
ALFA
MALF
FMIN
MINNB
OUT
***
...END MESANSWR-MES.
...21-7-1983.
```