Applied Acoustics **22** (1987) 47–69

Measurement of Coupling Loss Factors by Matrix Fitting: An Investigation of Numerical Procedures

C. H. Hodges†

Topexpress Ltd, 13–14 Round Church Street, Cambridge CB5 8AD (UK)

P. Nash

Resultant Ltd, 2 Anderson Court, Cambridge CB3 9EZ (UK)

and

J. Woodhouse*

Engineering Department, University of Cambridge, Cambridge CB2 1PZ (UK)

(Received 1 December 1986; accepted 8 December 1986)

SUMMARY

In an earlier paper,[1] an experimental procedure for determining coupling loss factors for Statistical Energy Analysis was described. To carry this procedure through in certain cases, a measured matrix must be optimally approximated by another matrix of a certain form. Numerical procedures for carrying out that optimisation efficiently are outlined here, and are illustrated with (artificial) examples. The procedures described seem capable of dealing with any case of potential practical significance, and are sufficiently simple that they can be implemented on a small microcomputer.

1 INTRODUCTION

In an earlier paper,[1] some basic aspects of the theory of Statistical Energy Analysis (SEA) modelling of vibration problems were discussed. An

† Deceased.

* To whom all correspondence should be addressed.

Applied Acoustics 0003-682X/87/$03·50

approach was suggested there to the problem of determining whether a SEA model is appropriate to a particular structure, and of evaluating the model parameters, i.e. the coupling loss factors.[2] This suggested approach led to a problem in matrix algebra and optimisation to which no answers were known at that time. That problem has now been examined in some detail, and two simple algorithms have been developed. A computer program has been written which allows those algorithms to be applied to the matrix-fitting problem. Experience with a wide range of simulated data suggests that this program can give acceptable answers to any problem of potential practical utility, and the way now seems open for serious attempts to apply the approach in practice to problems in structural vibration.

We first outline the original problem. Suppose that we have a system composed of a number N of coupled subsystems, as suggested in Fig. 1. We examine the form of a general SEA model for the vibrational behaviour of this system. SEA modelling is based on a thermal analogy, in which the role of temperature is played by the average kinetic energy per mode in a given subsystem. Write E_i for the average energy per mode in subsystem i and P_i for the rate of energy input to that subsystem from external sources. (Note that, for simple subsystems with reasonably uniform geometry (sections of panel, for example), E_i is proportional to the spatially averaged kinetic energy.[2,3])

Three assumptions are made in standard SEA modelling. The first is that the rate of energy dissipation by subsystem i is proportional to the energy E_i. We shall call the proportionality constant S_{ii}; clearly this constant is positive because energy is being dissipated, not created. The second assumption is that the rate of power flow from subsystem i to subsystem j is proportional to the difference of their energies; we take it to equal $S_{ij}(E_i - E_j)$. Again, it is

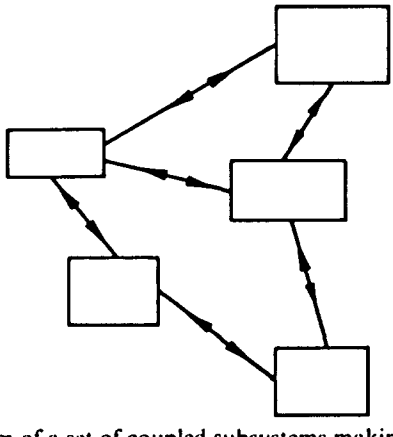

Fig. 1. Schematic diagram of a set of coupled subsystems making up a complex structure, such as might be modelled by SEA.

clear that the constants S_{ij} are all non-negative (some may well be zero, of course, since some pairs of subsystems will have no direct coupling). The third assumption is that the driving forces on the different subsystems are statistically independent, so that we can add energy responses of a given subsystem produced by these different driving forces to obtain the total mean modal energy of that subsystem. We now obtain the SEA equations simply by requiring energy balance for subsystem i:

$$S_{ii}E_i + \sum_{j \neq i} S_{ij}(E_i - E_j) = P_i \tag{1}$$

where the sum over j represents all energy flow away from subsystem i to other subsystems.

The physical meanings and scope of validity of these three assumptions are not discussed here. We have recently presented a substantial discussion of some of the issues raised, and the interested reader is referred to Ref. 3. The primary concern here is with the technology of fitting and using a SEA model when one is appropriate, although we shall see shortly that the method also gives a check on the applicability of SEA.

The main object of finding a SEA model for a given structure would be to predict the change in the pattern of vibrational response when some modification is made to the structure. To do this we need to know the energies E_i in response to given rates of external energy input P_i. At present eqn (1) expresses the P_is in terms of the E_is, so we must invert this relation. We first rearrange eqn (1) by collecting all occurrences of each E_i together, to read

$$P_i = \sum_{j=1}^{N} X_{ij}E_j \tag{2}$$

where

$$X_{ij} = \begin{cases} -S_{ij} & i \neq j \\ \sum_{k=1}^{N} S_{ik} & i = j \end{cases} \tag{3}$$

Inverting the matrix of eqn (2), we obtain the desired form of predictions:

$$E_i = \sum_{j=1}^{N} A_{ij}P_j \tag{4}$$

We have written **A** for the inverse of **X**; we now discuss the physical interpretation of this matrix, following Ref. 1.

Suppose we were to perform a set of measurements on the system, driving one subsystem at a time and measuring the response of all the subsystems. By driving subsystem i in some suitable way with a rate of energy input scaled to unity, and then determining the mean energy per mode in subsystem j (by, for example, a spatial averaging procedure[1-3]), we would have determined the matrix element A_{ij}. In this way the matrix **A** is directly measurable. If now all the subsystems were driven simultaneously with the energy input rates P_i, given the single assumption that the driving forces on the different subsystems are statistically independent, these energy responses could be superposed to give the total mean modal energy of subsystem i in the form of eqn (4).

Thus, we expect the subsystem responses to depend on the inputs to the various subsystems in a simple matrix fashion under very general circumstances: the question of whether a SEA model might now be appropriate hinges on whether this matrix has an inverse of a form which matches eqn (3). From this point of view, the constraints on the applicability of SEA to the system in question lie in the *signs* of the matrix entries. The measured matrix **A** is composed entirely of non-negative terms, while **X** has all its off-diagonal terms non-positive, and all its diagonal terms positive and sufficiently large that the sum along any row is non-negative. Any system whose matrix **A** has an inverse in the form **X** is a candidate for SEA modelling; we shall call any matrix with **X**s pattern of signs and row sums a SEA matrix.

Returning to our coupled system, we enquire a little more closely how we should test our measured matrix **A** to determine whether a SEA model is appropriate, and if so to determine the model parameters. We first evaluate $\mathbf{A}^{-1}$ to find out whether it is in the form **X**. If it is, all is well. If it is not, however, we do not necessarily give up and conclude that SEA is not usable. There will be errors in the measurements of A_{ij}, so we can seek a modification of **A** within the error bands of the measurements which does have an inverse in the correct form. If such an **A** can be found it might well form the basis of a usable SEA model. All the model parameters are determined, since from **X** we can deduce **S**, which is composed of the damping loss factors and coupling loss factors called for in the SEA model. Thus, we are interested in finding the matrix in SEA form whose inverse is closest, in some suitable sense, to the measurements. The closeness of fit which can be achieved will govern whether SEA modelling can be used for the system in question. That matrix-fitting problem is the subject of this paper. The measure of closeness of fit which we shall use is the simple least-squares measure:

$$D = \sum_i \sum_j (X_{ij}^{-1} - A_{ij})^2 \tag{5}$$

where **A** is the matrix of measurements and **X** is the approximating SEA model matrix. This measure can also be written in matrix form:

$$D = \mathrm{tr}\{(\mathbf{X}^{-1} - \mathbf{A})^2\} \tag{6}$$

where 'tr' connotes 'trace'. Note that both **A** and **X** are symmetric matrices; if **A** is not symmetric as measured, it should be symmetrised before any attempt is made to fit a SEA model. The degree of asymmetry in the measured matrix gives a useful indication of the magnitude of the measurement errors.

In practice, there may be additional constraints on this optimisation problem. It will usually be physically apparent that there is no direct coupling between certain pairs of subsystems, and thus we will be interested in the best-fitting SEA matrix with zeros in certain positions. This remark applies equally to the case when the crude inverse of the measurement matrix has the correct sign pattern: it might not have exact zeros in the expected places, and one might again wish to ask whether the data is consistent with zeros in those places, and what the best-fitting model then is. (An issue which we do not pursue here concerns the possibility of 'indirect coupling' terms in the SEA model, coupling between pairs of subsystems which are not directly in contact. We have discussed this possibility recently,[3] where it was pointed out that such indirect coupling can have simple physical significance in certain situations.)

The remainder of this paper is concerned entirely with the matrix-fitting problem, and in particular with algorithms for solving that problem which are sufficiently simple to program that they could reasonably be used in the field, as part of a vibration measurement programme. We do not pursue the acoustical implications here, and we only talk briefly about the experimental methodology of applying the approach in practice. It is intended to take up those issues in the future, when some experimentally determined matrices become available from a test structure.

2 THE ESTIMATION PROBLEM

We shall not go into much analytical detail of the matrix algebra underlying the estimation procedures which we describe. The aim of this paper is rather to show the vibration engineer how the approach could be used in practice with only a minimum of programming effort. We start this discussion simply by listing in summary some useful analytical results relating to SEA matrices and their inverses. Some of these were already known while others seem to be new,[4] and proofs will be published elsewhere. The results are as follows:

(1) The inverse of any SEA matrix is non-negative everywhere. This is reassuring in the light of the physical interpretation in terms of

energy of vibration. We return shortly to the question of whether zero entries are possible.

(2) A SEA inverse must be weakly diagonal dominant; that is to say, the diagonal element must be larger in magnitude than any off-diagonal element in the same row (or column, of course).

(3) Any strictly positive matrix which is 'sufficiently' diagonal-dominant (see Nash[4]) is a SEA inverse. This corresponds to the physical idea that SEA should always apply (in the sense discussed here) when coupling between subsystems is sufficiently weak.

(4) If **A** is a SEA inverse and **E** is a positive diagonal matrix, then **A** + **E** is a SEA inverse. This says that increasing the diagonal dominance is never a bad thing.

(5) If **A** is a SEA inverse and **E** is a positive constant matrix, then **A** + **E** is a SEA inverse. This says that reducing the relative variation among off-diagonal elements cannot destroy SEA-ness.

(6) Any non-negative, weakly diagonal dominant matrix with all its off-diagonal elements equal is a SEA inverse. This combination of results (4) and (5) gives a class of special SEA inverses which has no obvious physical significance, but which is used in one of the estimation algorithms to be described below.

(7) If any of the elements of a SEA inverse matrix are zero, it must be possible to permute rows until the matrix falls in a block-diagonal form, with no zero elements in the blocks. This corresponds to the physical idea that it is only possible to have no response at all on a particular subsystem if it is totally decoupled from the driven subsystem. Thus, the system must divide into a set of totally uncoupled systems. One would not expect this eventuality to arise in practice: if one's system is in several disjoint pieces, one might hope to spot that fact before starting the modelling process!

In the next section we shall describe three algorithms for the estimation problem. One algorithm is based on gradient projection, the second is a very simple procedure based on result (6) above, which we shall term the 'direct method', and the third is an application of Lagrange multipliers to the problem, which is usable once it becomes clear which constraint or constraints are active near the optimum. Of the three methods, the second and third are the main practical tools; the first method is a 'sledgehammer' approach which we use as a yardstick of convergence, and occasionally to help a difficult case out of an awkward spot.

An attempt has been made to convey something of the flavour of the estimation problem and these three approaches to it in a schematic diagram shown in Fig. 2. However, the reader is warned not to place any significance

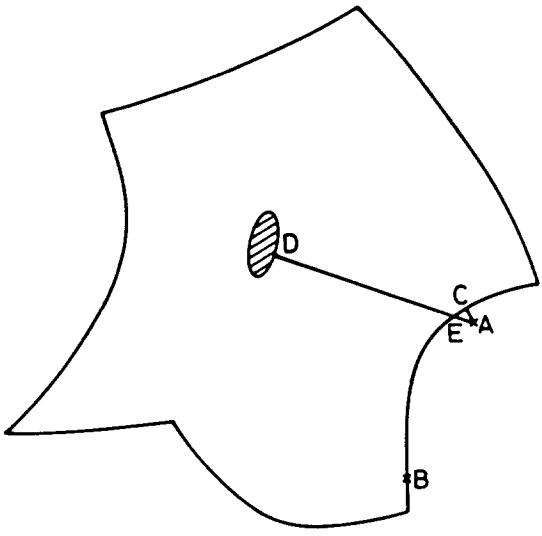

Fig. 2. Schematic diagram to illustrate the processes of matrix fitting. The line encloses the space of SEA-inverse matrices. It is composed of line segments representing the various sign and row-sum constraints. The position of the measured matrix A is marked, just outside this space. The shaded inner region shows the subspace of matrices with constant off-diagonal elements, used in the direct algorithm. The desired closest fit is the point C.

on the details of this figure—it gives the right impression, but it represents an attempt to combine in a two-dimensional picture features from a far higher dimensional space.

From the point of view of general optimisation theory, our most fundamental problem lies in the fact that, while the space of SEA matrices is very well behaved (and in particular is convex), the space of SEA inverses is far from well behaved. This is indicated in Fig. 2 by the non-convex region enclosed by a solid line. The boundary consists of a number of separate constraint surfaces representing our sign and row-sum constraints (we work in this space rather than in the inverse space, since our objective function from eqn (5) is readily visualised here in terms of distance). The interior of the boundary line is called the 'feasible region'. If this region had been convex we would have had some strong uniqueness theorems available, depending on the supporting hyperplane theorem.[5] As it is, there is always the possibility that there might be several local minima of our objective function. It is encouraging to note that, in practice, we have not had trouble with local optima when investigating a variety of artificial matrices (as will be described in Section 4). However, the possibility must always be borne in mind.

To continue describing Fig. 2, the inner region which is shaded indicates the space of the constant off-diagonal matrices mentioned above in result (6): note that this space is really of far lower dimensionality than the space of

SEA inverses, so this aspect of the figure is misleading. However, the figure correctly shows the subspace as convex. This inner region will play an important role in the second of our three algorithms, to be described shortly.

The measured matrix is indicated by the cross labelled A: it is shown here falling a little outside the space of SEA inverses, as one might expect if SEA-ness has only been destroyed by small errors in measurement. As we describe the three algorithms, we shall follow the behaviour of a simple example matrix. We use the 3×3 example discussed in the original paper.[1] We discuss the behaviour of more matrices in Section 4. For our 'measured' matrix **A**, we take

$$\mathbf{A} = \begin{bmatrix} 1{\cdot}0 & 0{\cdot}6 & 0{\cdot}9 \\ 0{\cdot}6 & 1{\cdot}0 & 0{\cdot}5 \\ 0{\cdot}9 & 0{\cdot}5 & 1{\cdot}0 \end{bmatrix} \tag{7}$$

The inverse of this is

$$\begin{bmatrix} 6{\cdot}25 & -1{\cdot}25 & -5{\cdot}00 \\ -1{\cdot}25 & 1{\cdot}58 & 0{\cdot}33 \\ -5{\cdot}00 & 0{\cdot}33 & 5{\cdot}33 \end{bmatrix} \tag{8}$$

The only term which prevents this being in the correct form is the (2, 3) entry 0·33, which is positive. If we simply set this entry to zero as a first guess at a SEA matrix close to the actual inverse, and invert that matrix again to compare with the original matrix (7), we obtain

$$\begin{bmatrix} 1{\cdot}74 & 1{\cdot}38 & 1{\cdot}63 \\ 1{\cdot}38 & 1{\cdot}72 & 1{\cdot}29 \\ 1{\cdot}63 & 1{\cdot}29 & 1{\cdot}72 \end{bmatrix} \tag{9}$$

which is a long way from matrix (7) and is therefore not acceptable. We show this matrix schematically as the point B in Fig. 2. The best-fitting matrix we are aiming for is shown as the point C. In this case it will turn out that, as shown in the figure, B lies on the same constraint surface as C, although a long way along it. However, as will be illustrated in Section 4, this is by no means always the case.

Before we describe the three algorithms for estimating the best fit, it is worthwhile looking in a little more detail at the simple matrix example above to understand how matrix (9) manages to be so different from matrix (7). The point at issue is the sensitivity of a matrix to changes in its inverse. Standard first-order perturbation theory tells us that

$$(\mathbf{X} + \boldsymbol{\delta})^{-1} \simeq \mathbf{X}^{-1} - \mathbf{X}^{-1}\boldsymbol{\delta}\mathbf{X}^{-1} \tag{10}$$

Now recall the expression for a symmetric matrix in terms of its eigenvalues and eigenvectors: if $\phi_i^{(n)}$ is the nth (normalised) eigenvector of a matrix **X**,

with corresponding eigenvalue $\lambda^{(n)}$, then

$$X_{ij} = \sum_n \phi_i^{(n)} \phi_j^{(n)} \lambda^{(n)} \tag{11}$$

and

$$X_{ij}^{-1} = \sum_n \frac{\phi_i^{(n)} \phi_j^{(n)}}{\lambda^{(n)}} \tag{12}$$

In passing, it is worth noting that all eigenvalues of a SEA matrix (and hence of a SEA inverse matrix) are positive. This follows directly from Gerschgorin's theorem,[6] since the diagonal elements of a SEA matrix have to be greater than the sum of moduli of the off-diagonal elements in their row.

Sensitive behaviour of the sort seen in eqns (7)–(9) occurs when the matrices in question are near-singular, in the sense that the ratio between the largest and smallest eigenvalue is large. When that happens, the expansion (11) is dominated by the term involving the largest of the λs, while eqn (12) is dominated by the term involving the smallest of the λs. Since the eigenvectors are normalised, we can use these extreme eigenvalues to get an order-of-magnitude estimate of the terms in eqn (10). The result is that the relative perturbation in the terms of the original matrix is magnified on inversion by a factor roughly equal to the ratio of the largest eigenvalue to the smallest one. The perturbation in the inverse matrix is dominated by the direct product of the eigenvector corresponding to the smallest $\lambda^{(n)}$.

We can see this process operating in the matrices given above, in eqns (7)–(9). The eigenvectors and eigenvalues of matrix (7) are

$$\begin{aligned} &(0{\cdot}623, \quad 0{\cdot}500, \quad 0{\cdot}601), 2{\cdot}35 \\ &(0{\cdot}258, -0{\cdot}857, \quad 0{\cdot}445), 0{\cdot}56 \\ &(0{\cdot}738, -0{\cdot}122, -0{\cdot}664), 0{\cdot}091 \end{aligned} \tag{13}$$

We see that the eigenvalue ratio is around 25. We have perturbed the matrix (8) so that, on inversion, (7) has turned into (9). It will be seen that the difference between these last two matrices is almost a constant matrix, with values around 0·75. This difference matrix will be dominated by the direct product of the eigenvector corresponding to the smallest $\lambda^{(n)}$, or equivalently to the largest eigenvalue of (7). Because any 'measured' matrix like (7) has all terms positive, it is clear from maximising the associated quadratic form that the largest eigenvalue will always correspond to an eigenvector which has all terms of the same sign. (For any vector with mixed signs, the quadratic form

could be increased by setting them all positive.) We can see from (13) that this is indeed the case here, and in fact this eigenvector is almost constant. The direct product of this eigenvector is responsible for the nearly constant difference matrix noted above. Although the matrix here is an invented one, this feature of an approximately constant eigenvector might be expected to arise in practice in 'difficult' cases. Any system with little internal damping will tend to exhibit approximate equipartition of modal energies among all subsystems,[3] so that the matrix **A** will tend to be uniform. As noted above, the matrix **A** itself will be dominated by the direct product of the all-positive eigenvector, although to a lesser degree than the perturbation matrix.

3 THREE SIMPLE ESTIMATION ALGORITHMS

The first of our three algorithms for seeking the optimum matrix at point C (in Fig. 2) is the most obvious one, a version of straightforward gradient projection. The simplest strategy for improving on the rather poor first estimate represented by the point B in the figure and by matrix (9) is to calculate the steepest-descent direction of the objective function, project that on to any currently active constraint surfaces, and take a step in that direction. A crude version of such a method, using hem-stitching rather that projection, was used in the original paper[1] to demonstrate that matrix (9) could be drastically improved on.

In the studies reported here, we have used an improved version of this algorithm for certain stages of the investigation. The simplest version of the method involves calculating the projected gradient direction as described above, then moving in that direction until either (a) we hit the boundary of the feasible region or (b) the objective function reaches a local minimum. Since we prefer (b) to (a), it is wise to adjust the search direction so that it points slightly *into* the feasible region. (Note that in visualising this method, Fig. 2 does not show the right space—gradient projection works in X-space rather than A-space.) The trouble with this method, as is well known, is that it converges very slowly. In particular, it tends to give a zig-zag path when the 'terrain' over which we are travelling exhibits long, narrow valleys—see, for example, Fig. 7.8 of Luenberger.[7] This problem can be alleviated slightly by various minor adjustments to the procedure, but the slow convergence means that the method as a whole is only of subsidiary interest here, and we do not go into any further details.

The 'direct' procedure, the second of our three algorithms, is quite different from any method based on gradients. We first find the closest approximation to our original measurement in the space of constant off-diagonal matrices, the point D. This is very easy: the solution to that least-

squares fitting problem is simply what one might expect, the result of averaging all the off-diagonal elements in the matrix, then increasing any diagonal elements if necessary to preserve weak diagonal dominance.[4] We then move in a straight line towards the original matrix at A until we hit the boundary of the space at E. For our example, this yields the matrix

$$\begin{bmatrix} 1{\cdot}003 & 0{\cdot}612 & 0{\cdot}865 \\ 0{\cdot}612 & 1{\cdot}003 & 0{\cdot}528 \\ 0{\cdot}865 & 0{\cdot}528 & 1{\cdot}003 \end{bmatrix} \tag{14}$$

which is the inverse of the SEA matrix

$$\begin{bmatrix} 4{\cdot}89 & -0{\cdot}97 & -3{\cdot}36 \\ -0{\cdot}97 & 1{\cdot}59 & 0{\cdot}0 \\ -3{\cdot}36 & 0{\cdot}0 & 3{\cdot}90 \end{bmatrix} \tag{15}$$

For later reference, the residual sum of squares here is 0·004 37.

This matrix is a far better fit to the original matrix (7) than was (9), and, surprising as it may seem, this simple process usually seems to yield a good answer immediately in this way. For many purposes it may be a sufficiently good estimate in itself, and in any case it makes a very good starting point for other methods, either a gradient-projection iteration, as already described, or the Lagrange multiplier approach, to which we turn next. In the next section we shall illustrate this with some examples.

The third algorithm for estimating the best-fitting SEA inverse matrix requires us to guess in one way or another which constraint or constraints are active at the optimum. (Of course, at least one constraint is bound to be active when the original matrix lies outside the feasible region.) We shall return later to discuss how this guess should be made. Once one knows which constraints are active, we have a fairly straightforward Lagrange multiplier problem. It turns out to be possible to manipulate the equations arising from this into a form suitable (usually) for a simple iterative solution. When this approach works, it is by far the quickest way of finding the true optimum point C (in Fig. 2). However, as we shall describe, there are sometimes pitfalls in the procedure.

Each of the SEA constraints can be written in the form

$$\mathrm{tr}(\mathbf{B}^{(n)}\mathbf{X}) \geqslant 0 \tag{16}$$

where the matrix $\mathbf{B}^{(n)}$ takes one of two forms. For the (α, β) off-diagonal sign constraint, we take

$$B_{ij} = -(\delta_{\alpha i}\delta_{\beta j} + \delta_{\alpha j}\delta_{\beta i}) \tag{17}$$

while for the αth row-sum constraint we take

$$B_{ij} = \delta_{\alpha i} + \delta_{\beta j} \tag{18}$$

Note that both forms of matrix $\mathbf{B}$ are symmetric, as is $\mathbf{X}$.

At the optimum, the active constraints will satisfy eqn (16) with equality. Thus, introducing Lagrange multipliers $\lambda^{(n)}$, the function

$$\mathrm{tr}\{(\mathbf{X}^{-1} - \mathbf{A})^2\} - \sum_n \lambda^{(n)} \mathrm{tr}(\mathbf{B}^{(n)}\mathbf{X}) \tag{19}$$

must be minimised with respect to the elements of $\mathbf{X}$. The resulting equations can be manipulated quite readily into a form suitable for iterative solution, leading to the system of equations

$$\mathbf{C}^{(n)} = \tfrac{1}{2}(\mathbf{A}^{-1}\mathbf{X}\mathbf{B}^{(n)}\mathbf{X}\mathbf{X} + \mathbf{X}\mathbf{X}\mathbf{B}^{(n)}\mathbf{X}\mathbf{A}^{-1}) \tag{20}$$

$$\mathrm{tr}(\mathbf{B}^{(n)}\mathbf{A}^{-1}) = -\tfrac{1}{2}\sum_m \mathrm{tr}(\mathbf{B}^{(n)}\mathbf{C}^{(m)})\lambda^{(m)} \tag{21}$$

$$\mathbf{X} = \mathbf{A}^{-1} + \tfrac{1}{2}\sum_m \lambda^{(m)}\mathbf{C}^{(m)} \tag{22}$$

The iterative solution proceeds as follows. Starting from the current estimate of $\mathbf{X}$, the $\mathbf{C}^{(n)}$'s are deduced from eqn (20). Then the simultaneous equations (21) are solved for the Lagrange multipliers $\lambda^{(n)}$, and a new estimate of $\mathbf{X}$ is obtained from eqn (22). (Note that the particular form of the matrix $\mathbf{C}^{(n)}$ in eqn (20) is the symmetrised version of the matrix one might first think of. It was found that preserving symmetry at all stages of the iterative scheme by adopting this form improved numerical stability.)

To use this algorithm in practice, we first need a reasonable estimate of $\mathbf{X}$ and a good guess as to which constraints should be active. This can usually be achieved using a combination of the first two algorithms which we have discussed. Also, of course, if we wish to impose a known pattern of zeros, this approach is ideally suited since we just add these zeros to the constraint set. In the case of our 3×3 example (without known zeros), the matrix (15) resulting from the direct algorithm is quite adequate. It has already achieved quite a good fit to the 'data', and it strongly suggests that the (2, 3) off-diagonal sign constraint is active. Iterating eqns (20)–(22) on that basis, just three iterations leads to the SEA model

$$\begin{bmatrix} 5{\cdot}0320 & -0{\cdot}8744 & -4{\cdot}0346 \\ -0{\cdot}8744 & 1{\cdot}5131 & 0{\cdot}0000 \\ -4{\cdot}0346 & 0{\cdot}0000 & 4{\cdot}5960 \end{bmatrix} \tag{23}$$

with inverse

$$\begin{bmatrix} 1{\cdot}0153 & 0{\cdot}5868 & 0{\cdot}8913 \\ 0{\cdot}5868 & 1{\cdot}0000 & 0{\cdot}5151 \\ 0{\cdot}8913 & 0{\cdot}5151 & 1{\cdot}0000 \end{bmatrix} \tag{24}$$

with residual sum of squares 0·001 19. This appears to be the true optimum to the number of significant figures displayed. It is a reassuring check that the gradient-projection algorithm does eventually converge to the same matrices, (23) and (24). However, with this particular case it is very slow, taking several thousand iterations to get there to three-figure accuracy. The advantage of using the direct method and the Lagrange multiplier method is apparent.

Unfortunately, the Lagrange multiplier iterative method does not always work as easily as this. We end this section with a brief discussion of some of the difficulties which can be encountered. The first problem is that the first estimate may not give the correct set of active constraints. What will then tend to happen is that the 'optimum' determined by the algorithm based on this set of constraints will lie outside the space of SEA matrices. Our computer program deals with this problem by a simple, if *ad hoc*, approach. If the new estimate of $\mathbf{X}$ lies outside the feasible region, a line search is carried out along the straight line joining the old and new $\mathbf{X}$s, to find the point where the boundary of the feasible region is reached. One can then look at the resulting $\mathbf{X}$ and $\mathbf{X}^{-1}$ matrices, change the constraint set, and try again.

An analogous problem is that some constraints may become inactive during the iterative process, and thus need to be removed from the set. This will be indicated by a change of sign of the corresponding Lagrange multiplier from positive to negative. (The positive sign for active constraints follows from the '$\geqslant$' inequality in eqn (16). Note that $\lambda^{(n)} \operatorname{tr}(\mathbf{B}^{(n)} \delta \mathbf{X})$ represents the increment in the residual sum of squares when constraint n alone is violated by a small amount $\delta \mathbf{X}$.)

A final problem with the iteration which should be noted is that it may not converge at all in some cases. The method we are using is the matrix analogue of the iteration $x_{n+1} = f(x_n)$ for solving the scalar equation $x = f(x)$. This iteration is well known to converge or diverge depending on whether $f'(x)$ is less than unity or greater than unity at the solution point. In a similar way, our iterative scheme could be linearised around the optimum point in terms of a Jacobian matrix.[8,9] If any of the eigenvalues of that Jacobian matrix happen to be greater than unity, the iteration will diverge, even when given a good starting approximation. In such cases, it is likely that it will converge to a local maximum rather than a local minimum of the objective function! Fortunately, experience with a number of matrices

indicates that this problem may not occur very often with matrices of a kind likely to be of practical interest. Presumably, the iterative scheme could be improved to cope with this problem by an approach analogous to the Newton–Raphson method, if necessary, but we have not attempted this.

4 THE COMPUTER PROGRAM, AND SOME CASE STUDIES

In this section we give a brief description of a computer program which implements the three algorithms discussed in the previous section, and we illustrate its use on some further artificial test matrices. These are all 5×5 matrices, created by taking a SEA matrix, inverting it, and adding some random noise to simulate measurement errors. In order to check whether the simple procedures investigated here really give the best answer, these matrices were also analysed by a far more sophisticated procedure, by the Division of Information Technology and Computing at the National Physical Laboratory.[10] The algorithms used by the NPL group would not be suitable for use on a small computer by vibration engineers, but our concern at present is to validate our simpler procedures, in a problem where it is possible that there might be more than one local optimum. If the NPL routines give the same answers, by a completely different route, then the credibility of the simple procedures is enhanced.

The program which we have written allows a matrix **A** to be read in from keyboard or disk file, and then allows the three estimation algorithms to be used on it, interactively, in any reasonable combination. It also allows 'hands-on' adjustments to be made to any of the four matrices **A**, $\mathbf{A}^{-1}$, **X** and $\mathbf{X}^{-1}$, by changing individual entries, by imposing weak or strong diagonal dominance, by symmetrising, etc. The matrices, residuals, residual sum of squares, eigenvalue ratio and so on can be inspected as the estimation process proceeds. The net result is a program which seems capable of producing a reasonable estimate for any matrix which might be of practical significance.

In this section we apply this program to three matrices of increasing difficulty. As a first case study, we consider the matrix

$$\begin{bmatrix} 0{\cdot}6046 & 0{\cdot}2393 & 0{\cdot}2892 & 0{\cdot}2132 & 0{\cdot}2342 \\ 0{\cdot}2393 & 0{\cdot}6055 & 0{\cdot}2679 & 0{\cdot}3012 & 0{\cdot}2291 \\ 0{\cdot}2892 & 0{\cdot}2679 & 0{\cdot}3373 & 0{\cdot}2221 & 0{\cdot}2086 \\ 0{\cdot}2132 & 0{\cdot}3012 & 0{\cdot}2221 & 0{\cdot}3853 & 0{\cdot}2295 \\ 0{\cdot}2342 & 0{\cdot}2291 & 0{\cdot}2086 & 0{\cdot}2295 & 0{\cdot}2975 \end{bmatrix} \tag{25}$$

with inverse

$$\begin{bmatrix} 2{\cdot}9591 & 0{\cdot}0691 & -2{\cdot}0027 & 0{\cdot}0844 & -1{\cdot}0432 \\ 0{\cdot}0691 & 3{\cdot}0709 & -1{\cdot}3006 & -1{\cdot}4643 & -0{\cdot}3775 \\ -2{\cdot}0027 & -1{\cdot}3006 & 7{\cdot}6475 & -1{\cdot}1550 & -1{\cdot}8943 \\ 0{\cdot}0844 & -1{\cdot}4643 & -1{\cdot}1550 & 6{\cdot}0028 & -2{\cdot}7595 \\ -1{\cdot}0432 & -0{\cdot}3775 & -1{\cdot}8943 & -2{\cdot}7595 & 7{\cdot}9306 \end{bmatrix} \tag{26}$$

Two sign constraints are violated here. Using the direct algorithm, we produce the estimate

$$\begin{bmatrix} 2{\cdot}8239 & 0{\cdot}0000 & -1{\cdot}6705 & -0{\cdot}0178 & -1{\cdot}0127 \\ 0{\cdot}0000 & 2{\cdot}9002 & -1{\cdot}1188 & -1{\cdot}1966 & -0{\cdot}5148 \\ -1{\cdot}6705 & -1{\cdot}1188 & 7{\cdot}5389 & -1{\cdot}2597 & -2{\cdot}3045 \\ -0{\cdot}0178 & -1{\cdot}1966 & -1{\cdot}2597 & 5{\cdot}8970 & -2{\cdot}7465 \\ -1{\cdot}0127 & -0{\cdot}5148 & -2{\cdot}3045 & -2{\cdot}7465 & 8{\cdot}3902 \end{bmatrix} \tag{27}$$

(row sums: 0·1229 0·0700 1·1853 0·6764 1·8117)

with inverse

$$\begin{bmatrix} 0{\cdot}6055 & 0{\cdot}2406 & 0{\cdot}2791 & 0{\cdot}2205 & 0{\cdot}2367 \\ 0{\cdot}2406 & 0{\cdot}6064 & 0{\cdot}2626 & 0{\cdot}2883 & 0{\cdot}2327 \\ 0{\cdot}2791 & 0{\cdot}2626 & 0{\cdot}3378 & 0{\cdot}2274 & 0{\cdot}2170 \\ 0{\cdot}2205 & 0{\cdot}2883 & 0{\cdot}2274 & 0{\cdot}3858 & 0{\cdot}2331 \\ 0{\cdot}2367 & 0{\cdot}2327 & 0{\cdot}2170 & 0{\cdot}2331 & 0{\cdot}2979 \end{bmatrix} \tag{28}$$

eigenvalue ratio 15·62, and residual sum of squares 0·000 968. This is quite a good fit already.

Only one constraint is now active. To see if the other one which was violated in matrix (26) should also be active, we do a few steps of the gradient-projection iteration. Ten steps produces

$$\begin{bmatrix} 2{\cdot}7921 & 0{\cdot}0000 & -1{\cdot}7006 & 0{\cdot}0000 & -0{\cdot}9927 \\ 0{\cdot}0000 & 2{\cdot}8894 & -1{\cdot}1344 & -1{\cdot}2307 & -0{\cdot}4991 \\ -1{\cdot}7006 & -1{\cdot}1344 & 7{\cdot}5287 & -1{\cdot}2497 & -2{\cdot}2830 \\ 0{\cdot}0000 & -1{\cdot}2307 & -1{\cdot}2497 & 5{\cdot}9000 & -2{\cdot}7187 \\ -0{\cdot}9927 & -0{\cdot}4991 & -2{\cdot}2830 & -2{\cdot}7187 & 8{\cdot}4268 \end{bmatrix} \tag{29}$$

(row sums: 0·0989 0·0252 1·1610 0·7009 1·9333)

with inverse

$$\begin{bmatrix} 0{\cdot}6133 & 0{\cdot}2444 & 0{\cdot}2825 & 0{\cdot}2186 & 0{\cdot}2338 \\ 0{\cdot}2444 & 0{\cdot}6146 & 0{\cdot}2663 & 0{\cdot}2912 & 0{\cdot}2313 \\ 0{\cdot}2825 & 0{\cdot}2663 & 0{\cdot}3391 & 0{\cdot}2259 & 0{\cdot}2138 \\ 0{\cdot}2186 & 0{\cdot}2912 & 0{\cdot}2259 & 0{\cdot}3830 & 0{\cdot}2278 \\ 0{\cdot}2338 & 0{\cdot}2313 & 0{\cdot}2138 & 0{\cdot}2278 & 0{\cdot}2913 \end{bmatrix} \tag{30}$$

with residual sum of squares reduced to 0·000 708. The second constraint has indeed become active again. No row-sum constraints are active. We are now in a strong position to use the Lagrange multiplier approach with these two constraints. Three iterations of this yields the final matrix

$$\begin{bmatrix} 2{\cdot}8862 & 0{\cdot}0000 & -1{\cdot}8504 & 0{\cdot}0000 & -0{\cdot}9281 \\ 0{\cdot}0000 & 3{\cdot}0449 & -1{\cdot}1953 & -1{\cdot}5015 & -0{\cdot}3261 \\ -1{\cdot}8504 & -1{\cdot}1953 & 7{\cdot}3678 & -1{\cdot}0209 & -2{\cdot}0779 \\ 0{\cdot}0000 & -1{\cdot}5015 & -1{\cdot}0209 & 5{\cdot}9513 & -2{\cdot}6887 \\ -0{\cdot}9281 & -0{\cdot}3261 & -2{\cdot}0779 & -2{\cdot}6887 & 7{\cdot}8332 \end{bmatrix} \tag{31}$$

(row sums: 0·1078 0·0220 1·2233 0·7403 1·8124)

with inverse

$$\begin{bmatrix} 0{\cdot}6046 & 0{\cdot}2434 & 0{\cdot}2865 & 0{\cdot}2152 & 0{\cdot}2316 \\ 0{\cdot}2434 & 0{\cdot}6055 & 0{\cdot}2653 & 0{\cdot}3012 & 0{\cdot}2278 \\ 0{\cdot}2865 & 0{\cdot}2653 & 0{\cdot}3409 & 0{\cdot}2208 & 0{\cdot}2112 \\ 0{\cdot}2152 & 0{\cdot}3012 & 0{\cdot}2208 & 0{\cdot}3853 & 0{\cdot}2289 \\ 0{\cdot}2316 & 0{\cdot}2278 & 0{\cdot}2112 & 0{\cdot}2289 & 0{\cdot}2992 \end{bmatrix} \tag{32}$$

eigenvalue ratio 14·98, and residual sum of squares 0·000 118. This is the same pair of matrices found by the NPL routines, within the stated accuracy, so on this reasonably straightforward case our program has performed well.

For our second case study we take a harder example, the matrix

$$\begin{bmatrix} 0{\cdot}3486 & 0{\cdot}3546 & 0{\cdot}2740 & 0{\cdot}3632 & 0{\cdot}3040 \\ 0{\cdot}3546 & 0{\cdot}4709 & 0{\cdot}2937 & 0{\cdot}3344 & 0{\cdot}3085 \\ 0{\cdot}2740 & 0{\cdot}2937 & 0{\cdot}7821 & 0{\cdot}2889 & 0{\cdot}2964 \\ 0{\cdot}3632 & 0{\cdot}3344 & 0{\cdot}2889 & 0{\cdot}4507 & 0{\cdot}3607 \\ 0{\cdot}3040 & 0{\cdot}3085 & 0{\cdot}2964 & 0{\cdot}3607 & 0{\cdot}3855 \end{bmatrix} \tag{33}$$

with inverse

$$\begin{bmatrix} 43{\cdot}3274 & -16{\cdot}5055 & -0{\cdot}9758 & -23{\cdot}3554 & 1{\cdot}6412 \\ -16{\cdot}5055 & 11{\cdot}2727 & -0{\cdot}0950 & 6{\cdot}9302 & -2{\cdot}4157 \\ -0{\cdot}9758 & -0{\cdot}0950 & 1{\cdot}8741 & 0{\cdot}5261 & -1{\cdot}0878 \\ -23{\cdot}3554 & 6{\cdot}9302 & 0{\cdot}5261 & 22{\cdot}2181 & -8{\cdot}3194 \\ 1{\cdot}6412 & -2{\cdot}4157 & -1{\cdot}0878 & -8{\cdot}3194 & 11{\cdot}8538 \end{bmatrix} \tag{34}$$

The direct algorithm yields the first estimate

$$\begin{bmatrix} 18{\cdot}9377 & -5{\cdot}1846 & -0{\cdot}8252 & -5{\cdot}6445 & -5{\cdot}5808 \\ -5{\cdot}1846 & 6{\cdot}5598 & -0{\cdot}2034 & 0{\cdot}0000 & -0{\cdot}9807 \\ -0{\cdot}8252 & -0{\cdot}2034 & 1{\cdot}9783 & -0{\cdot}0813 & -0{\cdot}6845 \\ -5{\cdot}6445 & 0{\cdot}0000 & -0{\cdot}0813 & 8{\cdot}4395 & -2{\cdot}6660 \\ -5{\cdot}5808 & -0{\cdot}9807 & -0{\cdot}6845 & -2{\cdot}6660 & 10{\cdot}7763 \end{bmatrix} \tag{35}$$

(row sums: 1·7026 0·1912 0·1840 0·0478 0·8643)

with inverse

$$\begin{bmatrix} 0{\cdot}3491 & 0{\cdot}3321 & 0{\cdot}3019 & 0{\cdot}3353 & 0{\cdot}3131 \\ 0{\cdot}3321 & 0{\cdot}4715 & 0{\cdot}3092 & 0{\cdot}3245 & 0{\cdot}3148 \\ 0{\cdot}3019 & 0{\cdot}3092 & 0{\cdot}7832 & 0{\cdot}3074 & 0{\cdot}3103 \\ 0{\cdot}3353 & 0{\cdot}3245 & 0{\cdot}3074 & 0{\cdot}4513 & 0{\cdot}3343 \\ 0{\cdot}3131 & 0{\cdot}3148 & 0{\cdot}3103 & 0{\cdot}3343 & 0{\cdot}3860 \end{bmatrix} \tag{36}$$

eigenvalue ratio 42·59, and residual sum of squares 0·007 510.

This estimate is perhaps already good enough for most practical purposes. However, we are interested in checking our program against the NPL routine for finding the optimum fit, so we continue with the other algorithms. A blow-by-blow account for this matrix is hardly feasible, because it turns out to be much more awkward than the previous matrix considered. We will thus display the final answer, then explain what makes this a difficult case. The final estimate is

$$\begin{bmatrix} 18{\cdot}9022 & -7{\cdot}2336 & -0{\cdot}2382 & -5{\cdot}5363 & -4{\cdot}4475 \\ -7{\cdot}2336 & 7{\cdot}8054 & -0{\cdot}3787 & 0{\cdot}0000 & -0{\cdot}0002 \\ -0{\cdot}2382 & -0{\cdot}3787 & 1{\cdot}8521 & 0{\cdot}0000 & -0{\cdot}9115 \\ -5{\cdot}5363 & 0{\cdot}0000 & 0{\cdot}0000 & 9{\cdot}7436 & -4{\cdot}2072 \\ -4{\cdot}4475 & -0{\cdot}0002 & -0{\cdot}9115 & -4{\cdot}2072 & 10{\cdot}5431 \end{bmatrix} \tag{37}$$

(row sums: 1·4466 0·1929 0·3237 0·0000 0·9767)

with inverse

$$\begin{bmatrix} 0{\cdot}3681 & 0{\cdot}3545 & 0{\cdot}2760 & 0{\cdot}3461 & 0{\cdot}3172 \\ 0{\cdot}3545 & 0{\cdot}4709 & 0{\cdot}2937 & 0{\cdot}3346 & 0{\cdot}3085 \\ 0{\cdot}2760 & 0{\cdot}2937 & 0{\cdot}7821 & 0{\cdot}2855 & 0{\cdot}2979 \\ 0{\cdot}3461 & 0{\cdot}3346 & 0{\cdot}2855 & 0{\cdot}4507 & 0{\cdot}3505 \\ 0{\cdot}3172 & 0{\cdot}3085 & 0{\cdot}2979 & 0{\cdot}3505 & 0{\cdot}3943 \end{bmatrix} \tag{38}$$

eigenvalue ratio 43·61, and residual sum of squares 0·001 631.

As with the previous matrix studied, this best fit is very close to that found by the NPL routine. However, some patience was needed to coax this answer from our program. The problem lies in choosing the set of active constraints for the Lagrange multiplier algorithm. The first point to note is that in this case the constraints which finally turn out to be active are not the same as those violated by inverting the original matrix. There are also more of them than the result of the direct algorithm suggests. In the end, we have two sign constraints active, one more just inactive, and one row-sum constraint active. However, during the calculation, the combination of these four constraints called for by the program keeps changing for quite a few

iterations. As described in the previous section, one must remove a constraint from the active set if its Lagrange multiplier goes negative, and add to the set any other constraint which becomes active spontaneously. The problem with this example lies in the fact that two of the four constraints here, namely the (2, 5) and (3, 4) sign constraints, are marginally active. They thus prevaricate between having negative (but small) Lagrange multipliers when included in the set, and becoming active again as soon as they are excluded. In this case, gradient projection did not help much either; no clear indication of the correct constraint set was given by that method, except by running it for thousands of iterations so that it came very close to the optimum. The results given above in eqns (37) and (38) seem to be about the best fit, but running the program with different histories seems to produce very slightly different answers, for reasons which are not entirely understood. On the whole, though, our program again performs reasonably well in comparison with the NPL routine.

As a final case study, we consider the hardest matrix from our test set. This is the matrix

$$\begin{bmatrix} 0{\cdot}1850 & 0{\cdot}2008 & 0{\cdot}1762 & 0{\cdot}1689 & 0{\cdot}2013 \\ 0{\cdot}2008 & 0{\cdot}2425 & 0{\cdot}1558 & 0{\cdot}1620 & 0{\cdot}1784 \\ 0{\cdot}1762 & 0{\cdot}1558 & 0{\cdot}1672 & 0{\cdot}1644 & 0{\cdot}1897 \\ 0{\cdot}1689 & 0{\cdot}1620 & 0{\cdot}1644 & 0{\cdot}2128 & 0{\cdot}1921 \\ 0{\cdot}2013 & 0{\cdot}1784 & 0{\cdot}1897 & 0{\cdot}1921 & 0{\cdot}2156 \end{bmatrix} \tag{39}$$

with inverse

$$\begin{bmatrix} -191{\cdot}3529 & 82{\cdot}3251 & 1420{\cdot}6015 & 105{\cdot}3732 & -1233{\cdot}2477 \\ 82{\cdot}3251 & -25{\cdot}1324 & -645{\cdot}0752 & -49{\cdot}8906 & 555{\cdot}9472 \\ 1420{\cdot}6015 & -645{\cdot}0752 & -14675{\cdot}4412 & -1243{\cdot}1944 & 13227{\cdot}2363 \\ 105{\cdot}3732 & -49{\cdot}8906 & -1243{\cdot}1944 & -85{\cdot}4038 & 1112{\cdot}8184 \\ -1233{\cdot}2477 & 555{\cdot}9472 & 13227{\cdot}2363 & 1112{\cdot}8184 & -11933{\cdot}4215 \end{bmatrix} \tag{40}$$

Neither the direct algorithm nor the Lagrange multiplier approach could be applied to this matrix as it stands, as is perhaps not surprising. The matrix (39) is not even weak-diagonal-dominant, and we see that (40) violates all the row-sum constraints, and all but four sign constraints! The simple starting procedure for the direct method yielded a singular matrix in this case, and the true set of active constraints would be impossible to guess from the matrix (40). This is a case where the simpler gradient projection approach comes into its own. To provide a SEA matrix of some kind as a starting point, a unit matrix was typed into the program. The gradient projection

algorithm was then iterated until it ground to a halt, after some 5000 steps, with the matrix

$$\begin{bmatrix} 38{\cdot}7617 & -12{\cdot}5857 & -10{\cdot}8614 & -2{\cdot}2770 & -13{\cdot}0372 \\ -12{\cdot}5856 & 14{\cdot}7364 & -0{\cdot}7734 & -0{\cdot}2962 & -0{\cdot}0260 \\ -10{\cdot}8613 & -0{\cdot}7734 & 30{\cdot}4794 & -8{\cdot}1220 & -7{\cdot}2516 \\ -2{\cdot}2770 & -0{\cdot}2962 & -8{\cdot}1220 & 19{\cdot}3246 & -7{\cdot}5070 \\ -13{\cdot}0371 & -0{\cdot}0260 & -7{\cdot}2516 & -7{\cdot}5070 & 27{\cdot}8217 \end{bmatrix} \tag{41}$$

(row sums: 0·0000 1·0551 3·4711 1·1224 0·0000)

which is the inverse of

$$\begin{bmatrix} 0{\cdot}2118 & 0{\cdot}1938 & 0{\cdot}1726 & 0{\cdot}1749 & 0{\cdot}1916 \\ 0{\cdot}1938 & 0{\cdot}2454 & 0{\cdot}1608 & 0{\cdot}1629 & 0{\cdot}1769 \\ 0{\cdot}1726 & 0{\cdot}1608 & 0{\cdot}1848 & 0{\cdot}1683 & 0{\cdot}1746 \\ 0{\cdot}1749 & 0{\cdot}1629 & 0{\cdot}1683 & 0{\cdot}2173 & 0{\cdot}1846 \\ 0{\cdot}1916 & 0{\cdot}1769 & 0{\cdot}1746 & 0{\cdot}1846 & 0{\cdot}2212 \end{bmatrix} \tag{42}$$

with eigenvalue ratio 48·48, and residual sum of squares 0·002 127. Considering the excesses of matrix (40), this seems quite a reasonable estimate.

For comparison, the NPL best estimate was

$$\begin{bmatrix} 1583{\cdot}0730 & -13{\cdot}4444 & -1546{\cdot}3125 & 0{\cdot}0000 & -23{\cdot}3160 \\ -13{\cdot}4444 & 14{\cdot}0347 & 0{\cdot}0000 & 0{\cdot}0000 & 0{\cdot}0000 \\ -1546{\cdot}3125 & 0{\cdot}0000 & 1560{\cdot}1237 & -10{\cdot}2180 & 0{\cdot}0000 \\ 0{\cdot}0000 & 0{\cdot}0000 & -10{\cdot}2180 & 18{\cdot}9729 & -7{\cdot}4456 \\ -23{\cdot}3160 & 0{\cdot}0000 & 0{\cdot}0000 & -7{\cdot}4456 & 30{\cdot}7617 \end{bmatrix} \tag{43}$$

(row sums: 0·0001 0·5903 3·5932 1·3093 0·0001)

which is the inverse of

$$\begin{bmatrix} 0{\cdot}1867 & 0{\cdot}1789 & 0{\cdot}1862 & 0{\cdot}1722 & 0{\cdot}1832 \\ 0{\cdot}1789 & 0{\cdot}2426 & 0{\cdot}1783 & 0{\cdot}1649 & 0{\cdot}1755 \\ 0{\cdot}1862 & 0{\cdot}1783 & 0{\cdot}1863 & 0{\cdot}1721 & 0{\cdot}1828 \\ 0{\cdot}1722 & 0{\cdot}1649 & 0{\cdot}1721 & 0{\cdot}2172 & 0{\cdot}1831 \\ 0{\cdot}1832 & 0{\cdot}1755 & 0{\cdot}1828 & 0{\cdot}1831 & 0{\cdot}2157 \end{bmatrix} \tag{44}$$

with eigenvalue ratio 2871, and residual sum of squares 0·003 653. In this particularly difficult case, our program has actually done better than the NPL routine by a significant margin. Not only is the residual sum of squares much smaller, but it is apparent on comparing matrices (41) and (43) that our program has gone to a different part of the space. Matrix (43) has many more constraints active, and a vastly higher eigenvalue ratio.

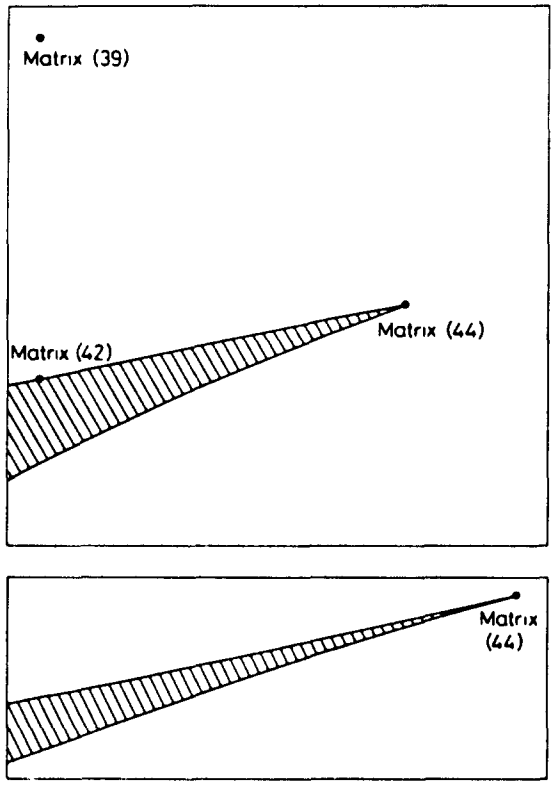

Fig. 3. A cross-section through a portion of the space of response matrices, for the final case studied in the text. The space of SEA-inverses is shaded. The three marked points in the upper diagram represent the original matrix A (39), our best fit (42), and the NPL result (44). The lower diagram is a 20× enlargement of the region near the tip, showing the NPL estimate caught near a cusp associated with a singularity of the matrix.

This last fact is the key to understanding what is going on. While at our point the eigenvalue ratio is quite moderate, the NPL routine has got stuck in the vicinity of a singularity. To explore this further, we have computed a cross-section through the feasible region to show the points in question. Figure 3 shows a region of the plane through the three matrices (39), (42) and (44). At least as seen in this plane, our program is close to the true optimum.The NPL result, however, is at the end of a very sharp spike, shown in a 20× close-up in the lower part of the figure. This spike, representing two or more constraint surfaces meeting more or less tangentially, is a consequence of being near a singularity, as suggested by the very large eigenvalue ratio. While a matrix with such bad behaviour is unlikely to be of much physical significance, it is reassuring that our program has managed to fit it to a fairly sensible SEA model.

5 CONCLUSIONS AND CAUTIONS

The main conclusion of this work is that the matrix-fitting problem proposed previously[1] seems to have been solved to a satisfactory level. A computer program has been described which copes with matrices of the kind likely to be encountered in practice, which is simple enough to run on a small computer, and thus to be of use to the vibration engineer wishing to establish a Statistical Energy Analysis model for a complex structure. The program has reproduced the results given by a much more sophisticated routine used by the NPL group on a set of test matrices.[9] In the previous section we have illustrated some of these matrices. The rest of the set were all handled satisfactorily, usually with no more difficulty than the first matrix discussed above, eqn (25). This includes the matrices with known patterns of zeros: the Lagrange multiplier algorithm in particular is well suited to coping with that case.

In the light of this success with the matrix fitting, it is appropriate to reconsider briefly the whole problem of this approach to SEA modelling of structural vibration. As explained in Section 1, the matrix fitting is simply a stage in the process of measuring a matrix **A** on a structure, testing whether a SEA model is consistent with that matrix within the known error bounds on the measurements, and then using the SEA model to predict the effect of structural modifications. There are many pitfalls in the other stages of this process, to which one should not be blinded by the success with one stage. This paper is not the place for a full discussion: we hope to return to that in due course, when some measurements are available. However, there are some issues directly related to the work reported here which we now discuss briefly.

The main issue concerns the degree of accuracy of SEA models fitted by the kind of procedure described in this paper. We have already noted that the relative errors in the measured matrix **A** tend to be magnified by a factor of the order of the eigenvalue ratio when turned into relative errors in the SEA model parameters. Thus for 'difficult' cases, characterised usually by a large eigenvalue ratio, the accuracy of the fitted model will be substantially lower than that of the measurements. The measurement accuracy will itself need some care to assess. At least in part it will be governed by statistical considerations arising from the number of different points sampled on the subsystems, since the aim is to measure something like a spatial average.[2,3]

As well as this general tendency of errors to be magnified, there are also some systematic biases in the matrix-fitting process which we have described. One might have hoped that some kind of 'averaging' of errors would take place with random noise added to a relatively large number of matrix elements representing a SEA inverse. Perhaps the underlying SEA

model could be recovered to rather higher accuracy than at first appears because of some such process? Unfortunately, this seems not to be the case. A simple and striking illustration of this follows from one of the analytic results listed in Section 2. Result (4) from that list says that we can always add a non-negative diagonal matrix to a matrix **A** without destroying SEA-ness. This immediately means that the diagonal elements will never be underestimated by the optimisation process: if we had a negative residual on the leading diagonal, we could always add a matrix with just that one entry non-zero, and thereby reduce the residual sum of squares. Thus the fitting process can never compensate for negative errors in the diagonal terms.

The matrices given in the two preceding sections illustrate this remark: compare, for example, eqn (24) with eqn (7). This comparison also reveals another interesting fact, although one of less general significance. Notice that not only are all the diagonal elements of eqn (24) greater than or equal to the corresponding elements of eqn (7), but that two of them are fitted exactly. These two, numbers two and three, correspond to the fact that the only active constraint is the (2, 3) sign constraint. By thinking carefully about the form of the matrices involved in the Lagrange multiplier calculation described in Section 2, it can be seen that with just one sign constraint active the two corresponding diagonal elements will always be fitted exactly at the optimum point. With more constraints active, matters are more complicated, and we do not pursue the question any further here since it seems to be of little importance in practice. The only use to which we have put it is as a convergence check on the numerical procedures.

ACKNOWLEDGEMENTS

The authors thank Drs I. Roebuck and G. W. T. White for helpful discussions on the subject matter of this paper. The work was supported by the Procurement Executive, Ministry of Defence.

REFERENCES

1. J. Woodhouse, An introduction to statistical energy analysis of structural vibration, *Appl. Acoust.*, **14** (1981), pp. 455–69.
2. R. H. Lyon, *Statistical energy analysis of dynamical systems*, MIT Press, Cambridge, Mass., 1975.
3. C. H. Hodges and J. Woodhouse, Theories of noise and vibration transmission in complex structures, *Rep. Prog. Phys.*, **49** (1986), pp. 107–70.
4. P. Nash, Estimating SEA response matrices, Internal report TOP88/83/9, Topexpress Ltd, 1983.

5. P. Whittle, *Optimization under constraints*, Wiley-Interscience, London, 1971.
6. J. H. Wilkinson, *The algebraic eigenvalue problem*, Clarendon Press, Oxford, 1965.
7. D. G. Luenberger, *Introduction to linear and nonlinear programming*, Addison-Wesley, Reading, Mass., 1973.
8. P. E. Gill and W. Murray (eds), *Numerical methods for constrained optimisation*, Academic Press, London, 1974.
9. W. Murray (ed.), *Numerical methods for unconstrained optimisation*, Academic Press, London, 1972.
10. E. M. R. Long, *Fitting of SEA matrices*, Report of the Division of Information Technology and Computing, National Physical Laboratory, 1982.