AUTOMATIC ANALYSIS OF FUSELAGES AND PROBLEMS OF CONDITIONING

by

HUSSEIN KAMEL AHMED HUSSEIN KAMEL B. Sc.

A thesis submitted for the degree of
Doctor of Philosophy of the
University of London

1964

SUMMARY

This work deals with the application of the specialised Matrix Force Theory for the analysis of fuselages to an electronic digital computer. The problems associated with the full automation of the procedure are described and solutions suggested. A particular feature is the generalization of simple matrix equations at particular fuselage stations into super-matrix equations involving the whole fuselage. The automation of the cut-out and modifications procedure is fully realised for this type of structure.

Computational problems inevitably associated with the solution of a large structural system are also discussed. The results of the analysis of a fuselage are attached. Suggestions for special functions are made as well to facilitate the programming for a large computer.

ACKNOWLEDGEMENTS

A debt of gratitude is owed to the supervisor of this work, Professor J.H. Argyris, for his constant help and advice, and also to Mr. S. Kelsey, co-author of the volume "Modern Fuselage Analysis and the Elastic Aircraft" for his encouragement. It was a pleasure and a privilege to be associated with them in the research project of which the book, and this thesis are the outcome.

The author would also like to thank Dipl.Math. Th.F. Buck, leader of the "Rechengruppe der Luftfahrt, Institut für Statik und Dynamik der Luft-und Raumfahrtkonstruktionen" of the "Technische Hochschule" in Stuttgart, for the exchange of ideas concerning the super matrix code.

Thanks are also due to Frau Hiltrud Ries for typing the draft, and to my wife Doris for the final manuscript. Messrs. N.K. Benson and M. Sørensen were also of valuable help.

INDEX

(contd)

INDEX

INTRODUCTION

The present thesis forms a natural part in the evolution of the matrix analysis of structure initiated by Argyris in Ref. (1) and especially refined with respect to fuselages in that author's textbook "Modern Fuselage Analysis and the Elastic Aircraft", Ref. (2). Our own contribution to this work involved the important aspects of the verification of the theory, which necessitated an enormous amount of experimentation with electronic computers and the complete solution of many examples including two elaborate fuselage structures both with cut-outs and modifications. As a matter of fact, the matrix theory of fuselages and the programmes upon which the major part of this thesis is based developed inevitably side by side. The results of the programme were used as important illustrations and elucidations to the theory so that by the time the analysis was completed, the first version of the fully automatized programme for the Ferranti-Pegasus Computer was available. This code was capable of analysing an arbitrary single-cell fuselage following the input of the minimum amount of logical and numerical data. The maximum capacity of this programme covered fuselages with up to 3o flanges and 4o ring stations. A further development was an automatized cut-out and modification programme which can handle cut-outs and modifications with up to 5o stations, again with a minimum of input data.

The development of this programme, and that of a further improved version suitable for a larger computer is based upon many other developments. On the one hand there was progress in the Matrix Force Method, particularly in the application to a fuselage type of structure; on the other hand advances in the 'software', involving a more ambitious and refined matrix code as being developed by the "Rechengruppe der Luftfahrt, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen" in Stuttgart.

In between comes the progress in structural software, which is the domain of the "applied programmer", or in this case, the engineer who uses the theory, which must be essentially suitable for programming on an electronic computer, as well as the available software, (in this case, the advanced and refined matrix interpretive scheme), in order to write the actual programme used to solve any one, or a class of structures.

To write a direct programme for solving a particular example using a given theory and an available computer library might be quite an intricate operation, but it would embrace a certain creativity only when full co-operation between theory, applied programmer and pure (or systems) programmer were achieved. In order to understand the relation between the three, we might state that whereas the theoretical part must be written so as to be suitable for programming on a computer, the software programmer prepares a certain library which would be useful for the applied programmer. The applied programmer then has his first function as a bridge between the two parties. Thus he might influence a choice in the theory, or the introduction of new facilities in the computer library. There is, however, another important aspect to the function of the applied programmer, namely: whereas the systems programmer intends his general programmes for the use of another programmer, either pure or applied, the applied programmer intends them for the use of a direct user, in this case, a structural engineer. The connection between the fuselage matrix theory and its automatic programming, as described in this work, is elaborated upon presently. In some of the appendices all the facilities required by the fuselage programme from the software library are listed and their functions specified. The final objective has been to make the programme fully automatic so that the engineer can "address" the machine in the most direct manner and receive back information which can be readily interpreted.

Having outlined the interconnection between the various members of the team involved in the development of theories and programmes suitable for the numerical analysis of complex structures, we discuss now the development of the Matrix Force Method and the programming connected with it.

In the original work of Argyris (Ref. (1)) giving the Matrix Force Method in its basic form, including the cut-out technique - though not yet the modification procedure which is described inter alia in Ref. (3) - we find the basic theory to be of a very general and simple nature; a sign of its intrinsic value and extreme flexibility. In a way, it has been shown that the real problem in analysing a regularized continuous structure can be reduced to the following steps :-

(1) The idealization of the structure.

(2) The formation of the basic matrices b_0 , b_1 and f .

(3) The insertion of these matrices into the very simple equation

$$b = b_0 - b_1 (b_1^t f b_1)^{-1} b_1^t f b_0$$

to obtain the stress distribution in the structure and following that, the the flexibility matrix F from

$$\begin{aligned} F = b_0^t f b &= b_0^t f b_0 + b_0^t f b_1 (b_1^t f b_1)^{-1} b_1^t f b_0 \\ &= F_0 - b_0^t f b_1 (b_1^t f b_1)^{-1} b_1^t f b_0 \end{aligned}$$

The second problem, i.e. that of modifying the stresses to represent the effect of cut-outs and area changes, can be split into two parts,

(1) The forming of the basic matrices required, namely

b_{1h} ,the rows of the b_1 matrix corresponding to the elements affected,

b_{2rh} ,the corresponding rows of the b_{2r} matrix for secondary redundancies,

b_h ,the corresponding rows of the b matrix,

$f_{\Delta h}$,the matrix of the flexibility difference in the modified elements of the structure.

(2) The insertion of the above matrices into the standard equations to obtain the modified stresses

$$b_m = b + b_1 D^{-1} b_{1h}^t \left\{ b_{1h} D^{-1} b_{1h}^t + b_{2rh} D_{2r}^{-1} b_{2rh}^t + \begin{bmatrix} 0 & 0 \\ 0 & f_{\Delta h}^{-1} \end{bmatrix} \right\}^{-1} S_h + \begin{bmatrix} 0 \\ b_{2r} \end{bmatrix} D_{2r}^{-1} b_{2rh}^t \left\{ b_{1h} D^{-1} b_{1h}^t + b_{2rh} D_{2r}^{-1} b_{2rh}^t + \begin{bmatrix} 0 & 0 \\ 0 & f_{\Delta h}^{-1} \end{bmatrix} \right\}^{-1} S_h$$

and

$$F_m = F + b_h^t \left\{ b_{1h} D^{-1} b_{1h}^t + b_{2rh} D_{2r}^{-1} b_{2rh}^t + \begin{bmatrix} 0 & 0 \\ 0 & f_{\Delta h}^{-1} \end{bmatrix} \right\}^{-1} b_h$$

As can be seen the set problem is indeed a simple one thanks to the suitability of the matrix language, and hence the basic theory, for this type of problem. Indeed the whole philosophy of the original work in Ref. (1) was centred on the idea that a matrix orientated analysis and the electronic computers were an ideal and powerful combination.

An important point when programming for a large structure has been already mentioned in Ref.(1), namely the need to partition large matrices. Even with a large computer, such as the UNIVAC 1107, the computing store cannot hold more than 64 K words($K = 2^{10}$). The storage space inside a large computer is arranged in layers, each of which being larger and taking longer to reach than the preceeding one. The fastest of the layers, the computing store, is the only part of the computer where one can store two matrices, perform an operation on them and write the result. All the other layers serve as a "backing store", which is a large memory only used for keeping the information but not for computation. A paper by Hunt (Ref. 6) followed the first work and described some general purpose programmes for the computation of intermediate matrices such as the D , D_o (and F) in a continuous loop of instructions.

So, whereas the restrictions on the size of the matrices to be operated upon is clearly indicated, another great advantage of the partitioning technique is not so obvious, namely the presence of zero sub-matrices which can be excluded from the computation and, naturally enough, not stored. Here again we see one more aspect of the suitability of the theory to the computer. A special theory has been developed, from which one obtains a partitioning of the basic matrices and all relevent intermediate ones according to natural physical considerations, resulting in a pattern of fully populated sub-matrices, the rest being zero.

This point has been fully exploited in the book on fuselage analysis. A special matrix force theory was developed for a certain type of structure resulting in very sparsely populated $\mathbf{b}_1$ and $\mathbf{D}$ matrices, $\mathbf{f}$ being diagonal and treated as such. In this manner the theory is excellently suited to the machine and we are now left with the important aspect of writing a proper programme to make full use of it. Two possible approaches stand before us depending upon the size of the machine. If it is small, like the Pegasus, with a reasonably fast addressable store of 8 K words, one has to develop a special-purpose programme for the analysis of fuselages in which the pattern of the various matrices is embodied. To explain, we must bear in mind that in those 8 K words, a fairly elaborate matrix scheme, the actual programme as well as the data and results have to be stored. A set of addressable magnetic tape units (or in the case of a more modern machine, a faster backing store) is naturally enough indispensable for the computation and might also be used to hold the programme itself. The programme is divided into a large number of independent sub-programmes, each performing one particular specialised operation, so that only a small part is contained in the computing store at a time.

This is the basis of the automatic programme which has been written for the Pegasus. In order to give an idea of its size, we mention that the part of it concerned with the analysis of single-cell continuous fuselages requires about 1500 matrix instructions as well as 10 000 machine orders, in all about 13 000 storage places. The computing store cannot retain all this at one time, whereas the space required by any one sub-programme never exceeds about 800 words. The matrix scheme itself occupies a certain part of the drum, the rest being left for some standard programmes and as a working space; the intermediate and final results being then transferred continuously during the running of the programme to and from tape.

Whilst this approach is suitable for a small computer, a super matrix code is the more recommendable software for a larger one. Thus, as in the one being developed in the Stuttgart Institute, the matrix code has standard mathematical orders dealing with super-matrices, i.e. matrices whose elements are themselves matrices. Such a scheme automatically ignores zero sub-matrices. It does not store them, nor carry out any operation which involves them. Thus the pattern of the matrix is not used to develop the programmes, but the one and same function in the super matrix scheme handles all types and patterns. A large fast machine, with a sufficiently fast large backing store as well as a suitable computing store can retain the super code as well as the programme concerned with the solution of the structure. The question comes now as to how the topology of the structure is to be utilised to simplify, standardise and automatise the programme.

The answer to this lies in the fact which has been already realised namely that the actual formation of the basic matrices $\mathbf{b}_0$, $\mathbf{b}_1$ and $\mathbf{f}$ constitutes a considerable part of the actual solution. As a matter of fact, with the introduction of the super matrix scheme it becomes practically the main part of the programme. For any problem of considerable size and sufficient generality, the preparation of these matrices by hand is a slow, unreliable and almost impossible task. The correct procedure would go even as far as to give the geometry of the structure, if possible, in the simplest manner, making use of any special properties it possesses and to leave the maximum share of the work to the computer. Although one finds already in Ref. (1) standard forms for the flexibilities $\mathbf{f}$, as well as for self-equilibrating systems, the values were given for one particular structural unit typical of the structure concerned, without developing the expressions to show the pattern of the total matrices, or rather their major sub-matrices. This changed with the advent of the book on fuselages, where the patterns of the various matrices, including the $\mathbf{D}$ and $\mathbf{D}_o$ were discussed in detail. Thus there was a good example of the general theory being developed into a special one for application to a certain structural form. The full automatization of the formation of the $\mathbf{b}_o$ and the $\mathbf{b}_1$ through setting up the equilibrium conditions, together with the choice of the $\mathbf{A}_\ell$ matrix which controls the conditioning of the local sets of redundancies, and the development of the orthogonalization technique for the further automatic improvement of the set of equations is an ideal example of the utilization of the topology of the structure in such a special theory. By direct inversion at each cross-section one obtains cover stress-distributions which give a unit loading resultant or a self-equilibrating system. The application of these cover stresses to the boundaries of the ring, using a standard ring matrix analysis gives the internal ring stresses corresponding to these self-equilibrating systems. The supplied geometrical data in the form of co-ordinates , section constants and elastic properties are used for the computation of the $\mathbf{b}_o$ and $\mathbf{b}_1$ matrices, as well as for the matrix $\mathbf{f}$. Data concerned with loading are used to derive a complete set of cover stresses in equilibrium with the applied loading from the cover stress systems corresponding to unit loading resultants. The corresponding stress distribution in the rings is calculated using the same standard ring analysis. Having thus obtained our complete basic system $\mathbf{b}_o$, as well as the $\mathbf{b}_1$ and $\mathbf{f}$, we can set up the equations in the primary redundancies and solve them, obtaining thus the final stress distribution in the fuselage due to the applied loading.

As mentioned before, if only a small computer is available, the best method is to write programmes which perform the required operations one by one, proceeding gradually from one end of the fuselage to the other. The main part of each sub-programme is a set of matrix instructions carrying out a standard operation using the sub-matrices of a certain super-matrix, as a rule stored on tape, and placing the result again on tape. A loop is then set up around this part which considers the pattern of the matrices involved, as well as such parameters as the number of flanges t , number of rings p and number of loading cases ρ , in modifying the input and output addresses of the sub-matrices working from one end of the fuselage to the other, knowing that only non-zero sub-matrices are stored. Having a large fast machine, equipped with a proper matrix scheme changes the picture from the programming point of view considerably, and the resulting programme is the actual one given in this work.

Since the super-code is designed to operate on super-matrices directly, it is obvious that such operations as the formation of the $\mathbf{D}$ and $\mathbf{D}_o$ *, solution of the equations, should be done in a single order from one end of the fuselage to the other, without having to construct a loop.

In forming the stress matrices $\mathbf{b}_o$, $\mathbf{b}_1$, and the flexibility matrix $\mathbf{f}$ of the elements one now has also to adopt this new approach. As a logical development of the original theory we have to assemble the individual equations at each station for one type of structural matrix

---

* $\mathbf{D}$ and $\mathbf{D}_o$ are the matrices of the influence coefficients needed to set up the equations for primary unknowns $\mathbf{Y}$.

into one super-matrix equation to derive the corresponding total matrix for the whole fuselage in one operation. As a matter of fact, the actual computations have still to be broken down into basic logical standardised steps. Whether more than one of these steps can be put together depends upon the degree of refinement of the super matrix scheme. The programme given in this thesis is, in any case, subdivided into the individual operations so that it can be coded immediately into the actual language of the super scheme.

As a result of the co-operation between the systems programmer and the applied programmer the available software will develop in order to suite the problems in question. Thus, whereas with a simple matrix scheme one has to write special purpose loops of varying complexity in contrast to the straightforward matrix orders with a super scheme, one has yet perfect control over the storage space, having allocated it oneself either directly or through a special programme. The super scheme, however, takes this part in hand, thus greatly losing flexibility. If, therefore, only functions of a purely mathematical nature are allowed, the result places an undue restriction on the range of soluble problems. A brief glance at the programme developed in this work shows that by far the main part is devoted to the formation of the basic matrices. In forming these matrices, one needs a considerable amount of freedom. To mention a simple example, one may store a column vector and refer to it later as a diagonal matrix. This may not always be possible with a highly complicated matrix scheme. A certain amount of inflexibility follows invariably as a result of automation. Everything has to be standardised. The solution to this dilemma lies, in our opinion, in the construction of the super code in such a manner that it is always possible to introduce new functions; in this case, a function which 'diagonalises' a vector matrix. These functions need obviously not only be of mathematical, but can be of purely logical nature. In this manner, a user can have a special version of the scheme containing a few extra functions particularly suitable for his purpose. A complete list of the suggested functions, as well as all other standard ones, which are assumed to be contained in the super code is given. In choosing these functions we have to observe that they should be as general and as elementary as possible, so as to be also applicable in other spheres. An excellent example are the suggested Boolean functions, i.e. special functions to deal with Boolean matrices, containing only zeros and ones, which are also separately quoted and described in Appendix (B). These functions have proved to be of enormous value in many problems including ones involving plasticity, use of the displacement method, as well as those associated with cut-outs and modifications (Ref.(4)) and (Ref. (5)). Their immediate applicability to such widely different programmes signifies that they are a rather fortunate choice, and it is almost certain that they will still find further applications in many other problems, not necessarily concerned with structures, and might become a standard part of every matrix scheme.

As we have said before, the most essential data are introduced in a form understandable to an engineer. This is an important part of writing a programme which can be considered as a 'structural software'. Extending now the arrangement to the modification and cut-out techniques we realize that, just as forming the matrices $\mathbf{b}_0$, $\mathbf{b}_1$ and $\mathbf{f}$ is the main difficulty in the analysis of the regularized structure, here the formation of the basic matrices $\mathbf{b}_{1h}$, $\mathbf{b}_{2rh}$, $\mathbf{S}_h$ and $\mathbf{f}_{\Delta h}$, constitutes the real problem. First of all one has to give the structural engineer the facility of specifying his cut-outs and modifications in the form of simple orders giving the positions of the affected elements. These orders have to be comprehensible to an ordinary structural engineer, not necessarily closely aquainted with the computer. The machine then accepts these orders and uses them to form logical matrices which will eventually be used to form the required basic matrices. In general, of course, a problem contains a mixture of cut-outs and modifications.

In the case of the elements to be cut, the machine can prove, from certain logical considerations, whether any cut-outs are superfluous. Having checked this, the machine investigates whether the required modifications of the elements result in changes in the direct flexibility somewhere else. If so, the addition of the extra rows to the $\mathbf{b}_{1h}$, $\mathbf{b}_{2h}$, $\mathbf{S}_h$ and the corresponding extension to $\mathbf{f}_{\Delta h}$ are automatically planned. So, by using the chosen form of orders, and translating them by an interpretative programme, forming the appropriate Boolean matrices and then using them, through direct multiplication to obtain the required basic matrices, the whole cut-out and modification procedure is fully automatized, and greatly simplified. The usual errors arising due to the manual calculation of the addresses and dimensions of the matrices, anyhow an unpractical suggestion with a super matrix scheme, disappear. This method has proved to be extremely reliable and indeed the only errors encountered were in the calculation of the modified cross-section properties of the rings, yet another proof of the advantages achieved by automation.

A chapter devoted to the problems of conditioning is also included. It is best said at this stage that the question is far from being a simple one. It has already been discussed by many authors, who tried to define it and establish criteria to measure it, detect it and try to cure the loss of accuracy. None of the suggested measures is really satisfactory, but they all help to indicate the nature of the problem. Probably the best approach from an engineers point of view is to analyse typical structures with varying grid arrangements in order to study the accuracy obtained, including the effects of some special cases, as with some critical relative ring stiffness (see Ref. (2)), and thus establish empirical or semi-empirical rules depending upon the type of structure, and any peculiarities present. As a typical investigation, we analyse, using the Displacement Method, a one-dimensional chain of flanges fixed at both ends in which all members are of the same length except the first and last elements which have a different one. The effect of the size of the problem as well as the relative length of the end elements and the intermediate ones on the conditioning is then studied, and the various criteria for the detection of ill-conditioning applied to test their reliability. Several incidental mathematical results evolve leading to a simple method of determining the inverse and the eigen-vectors of a certain class of super-matrices. In this manner one can extend the analysis to some other types of matrices whose conditioning can be predetermined, and this should serve also the purpose of testing inversion as well as eigenvalue and eigenvector programmes for large matrices. These results are then also applied to the fuselage problem and some methods are suggested for improving the accuracy of a solution.

In the Appendices (B) and (C) we discuss the aformentioned Boolean orders and other useful functions to be included in the super scheme. There follow the results of calculations carried out on a fuselage with twenty flanges and ten rings, including three different cut-out and modification cases. We conclude with an Appendix connected with the investigations on the conditioning in Chapter V .

CHAPTER I

MATHEMATICAL NOTATIONS AND DEFINITIONS

Throughout our thesis we make full use of the facilities and conventions contained in the so-called super matrix code for our computers (see Appendix C)

In this first chapter we present a listing and definition of the main symbols employed in our work. We start by describing in general terms some characteristic features of the matrix notation.

A uniform super-matrix is a super-matrix whose sub-matrices are all of the same size.

$\mathbf{A}_{[M \times N,\, m \times n]}$ is a general uniform super-matrix of the (super) order $[M \times N]$, the MN sub-matrices of which are all of the order $(m \times n)$. If $M=N$ (or $m=n$) only M (or m) will be written down. If the matrix is a super-diagonal (or has diagonal sub-matrices) this will be denoted by a stroke after the dimension, e.g. $[M/, m \times n]$ or $[M \times N, m/]$, or even $[M/, m/]$.

The designation scalar matrix or scalar super-matrix stands for a diagonal matrix all of whose elements are equal. This is indicated for the matrix or sub-matrix in question by a pair of round brackets, $(\ )$. For example, $\mathbf{A}\,[M \times N, (\ )]$ is a super-matrix of the order $[M \times N]$, the elements of which are scalar matrices whose dimensions are determined by the other matrices involved in the operations. In the matrix $\mathbf{A}[(\ ), m \times n]$, however, the sub-matrix of order $(m \times n)$ is repeated as many times along the diagonal as is required by other matrices with which it is associated.

Since the storage of either matrix is essentially the same, and also similar for $A_{[1, M\times N]}$ or $A_{[M\times N, 1]}$, the same stored information could be called upon in all these different ways. This facilitates certain operations, especially with Boolean matrices.

$A_{i.}$ is the $i^{\underline{th}}$ row of A.

$A_{.j}$ is the $j^{\underline{th}}$ column of A.

e_m or $e_{(m\times 1)}$ is a Boolean column-vector of order $(m\times 1)$ whose elements are all unity. We also call it a summation vector of the order m.

$$e_{(m\times 1)} = \{ 1 \quad 1 \quad 1 \quad 1 \ldots\ldots\ldots\ldots 1 \quad 1 \quad 1 \} \qquad (I,1)$$

e_{m_i} or $e_{(m\times 1)_i}$ is a Boolean column-vector of order $(m\times 1)$ whose elements are all zeros except the $i^{\underline{th}}$ element which is equal to unity. We call it a selection-vector of the order m.

$$e_{(m\times 1)_i} = \{ \underset{1}{0} \quad \underset{2}{0} \quad \underset{3}{0} \ldots\ldots \underset{i-1}{0} \quad \underset{i}{1} \quad \underset{i+1}{0} \ldots\ldots \underset{m-1}{1} \quad \underset{m}{1} \} \qquad (I,1a)$$

$e_{[M\times 1, m\times 1]}$ is a super-summation vector whose sub-matrices are summation vectors of the order $(m\times 1)$.

If it is necessary to distinguish between a Boolean matrix and its equivalent in floating point form we use for the former the suffix B.

Before we proceed with further definitions, it is preferable to state the most elementary matrix rules arising in the use of uniform super-matrices. For example, if two such matrices are equal

$$\mathbf{A}_{[M_a \times N_a\ ,\ m_a \times n_a]} = \mathbf{B}_{[M_b \times N_b\ ,\ m_b \times n_b]} \tag{I,2}$$

then we must have

$$M_a = M_b \quad , \quad N_a = N_b \quad , \quad m_a = m_b \quad , n_a = n_b \tag{I,2a}$$

all corresponding sub-matrices being then also element by element equal.

If it happens that each element of one of the two matrices is equal to the corresponding one in the other, these relations, however, not applying to the sub-matrices themselves, then it is evident that the matrices are merely partitioned in a different manner. We describe such super-matrices $\mathbf{A}$ and $\mathbf{B}$ as equivalent and express this as follows

$$\mathbf{A} \;\sqsubseteq\; \mathbf{B} \tag{I,3}$$

Naturally the relations

$$M_a\ m_a = M_b\ m_b$$

and (I,3a)

$$N_a\ n_a = N_b\ n_b$$

must hold.

If two super-matrices are to be added, they must be of the same order. That is to say, if the operation

$$A_{[M_a \times N_a,\, m_a \times n_a]} \pm B_{[M_b \times N_b,\, m_b \times n_b]} = C \qquad (I,4)$$

is performed, conditions (I,2a) must again be true.

If we proceed with the multiplication of the super-matrices

$$A_{[M_a \times N_a,\, m_a \times n_a]} \; B_{[M_b \times N_b,\, m_b \times n_b]} = C \qquad (I,5)$$

it is necessary to have

$$N_a = M_b$$

and (I,5a)

$$n_a = m_b$$

The uniform super-matrix C resulting from the operation is of the order $[M_a \times N_b,\, m_a \times n_b]$.

We now return to the definitions

$E_{(m)} = E_{(1,m)} = e_m \; e_m^t$ is a square matrix $(m \times m)$ all elements of which are equal to unity.

$E_{(1,m)_{ij}} = e_{m_i} \; e_{m_j}^t$ is a square matrix $(m \times m)$ all elements of which are equal to zero, except the element E_{ij} which is equal to unity.

$E_{[M,m]} = e_{[M \times 1,\, m \times 1]} \; e_{[M \times 1,\, m \times 1]}^t$ is a super-matrix of order $[M \times M]$ all elements of which are equal to $E_{(m)}$.

$\underset{[1,m]}{I_a}$ is called the Boolean rotational or advancing operator of order (m). If I_a is placed as a premultiplying operator to a matrix it insures a single rotation of the rows. In particular the resulting matrix contains the original second row as a first row whilst the original first row becomes last. For example,

$$\underset{[1,m]}{I_a} \underset{[1,m\times n]}{A} = \underset{[1,m]}{I_a} \begin{bmatrix} A_{1.} \\ A_{2.} \\ A_{3.} \\ \vdots \\ A_{(p-1).} \\ A_{p.} \end{bmatrix} = \begin{bmatrix} A_{2.} \\ A_{3.} \\ A_{4.} \\ \vdots \\ A_{p.} \\ A_{1.} \end{bmatrix} \qquad (I,6)$$

It also follows that

$$\underset{[1,m]}{I_a^k} \underset{[1,m\times n]}{A} = \begin{bmatrix} A_{(k+1).} \\ A_{(k+2).} \\ A_{(k+3).} \\ \vdots \\ A_{m.} \\ A_{1.} \\ \vdots \\ A_{k.} \end{bmatrix} \qquad (I,6a)$$

$I_{[1,m)}$ or I_m is a unit-matrix of the order $(m\times m)$, stored best as a scalar, or simply as a title.

$\mathbf{I}_{[M,m]}$ is a unit (Boolean) super-matrix of the super-dimensions $M \times M$, whose elements are sub-matrices of the order $m \times m$, the diagonal ones being $\mathbf{I}_m$ and the rest zeros.

The order of joining two (or more) matrices together will be diagrammatically shown in the form

$$A = \begin{bmatrix} B & C \end{bmatrix} \tag{I,7}$$

and

$$A = \begin{bmatrix} B \\ C \end{bmatrix} \tag{I,7a}$$

or for a diagonal matrix

$$A = \ulcorner\overline{B \quad C}\lrcorner \tag{I,7b}$$

The order to split a matrix into two (or more) parts is given as

$$A = B\,\big)\big(\,C \qquad A = \begin{array}{c} B \\ \smile \\ \frown \\ C \end{array} \tag{I,8}$$

or for diagonal matrices

$$A = \underline{B}\rfloor \; \ulcorner\overline{C} \tag{I,8a}$$

The order

$$A(b)$$

usually denotes a "scalar multiplication" of a super-matrix with a matrix compatible with its sub-matrices. That is, each sub-matrix of A will be postmultiplied by b. It is, of course, preferable to write this operation as Ab and consider b as a diagonal scalar super-matrix of dimensions $[\,(\,)\,,m\,]$. For detailed discussion of all necessary matrix operations Appendix C should be consulted.

<u>Definition of Matrices of Structural Interest</u> (see also Fig. I,1)

The super-matrix $\mathbf{P}$ of flange loads at all frame stations before and after each frame is described as the column matrix

$$\underset{[2(p-1)\times 1,\, t\times 1]}{\mathbf{P}} = \left\{ P_{1+} \; P_{2-} P_{2+} \; P_{3-} P_{3+} \cdots\cdots\cdots P_{(p-1)_-} P_{(p-1)_+} \; P_{p-} \right\} \tag{I,9}$$

where p is the number of frame stations and t the number of flanges. The symbols $+$ and $-$ as suffices denote fore and aft cross-sections at a frame.

The super-matrix $\hat{\mathbf{R}}_\ell$ of the resultant normal forces, and bending moments before and after each frame station entering in the computation of the flange loads is defined as

$$\underset{[2(p-1)\times 1,\, 3\times\rho]}{\hat{\mathbf{R}}_\ell} = \left\{ \hat{R}_{\ell_{1+}} \; \hat{R}_{\ell_{2-}} \hat{R}_{\ell_{2+}} \; \hat{R}_{\ell_{3-}} \hat{R}_{\ell_{3+}} \cdots\cdots\cdots \hat{R}_{\ell_{(p-1)_-}} \hat{R}_{\ell_{(p+1)_+}} \hat{R}_{\ell_{p-}} \right\} \tag{I,10}$$

where ρ is the number of loading cases. The sub-matrices of $\hat{\mathbf{R}}_\ell$ we write as

$$\hat{R}_{\ell_{i-}} = \left\{ N_{i-1,i} \; M_{y,i-} \; M_{x,i-} \right\}$$

and

$$\hat{R}_{\ell_{i+}} = \left\{ N_{i,i+1} \; M_{y,i+} \; M_{x,i+} \right\} \tag{I,10a}$$

Here, N is the normal force in a bay, say $i,i+1$, and M_x and M_y bending moments; (see Eqn. ((II, 15a)))*

* Equation numbers in double brackets refer to Ref. (2)

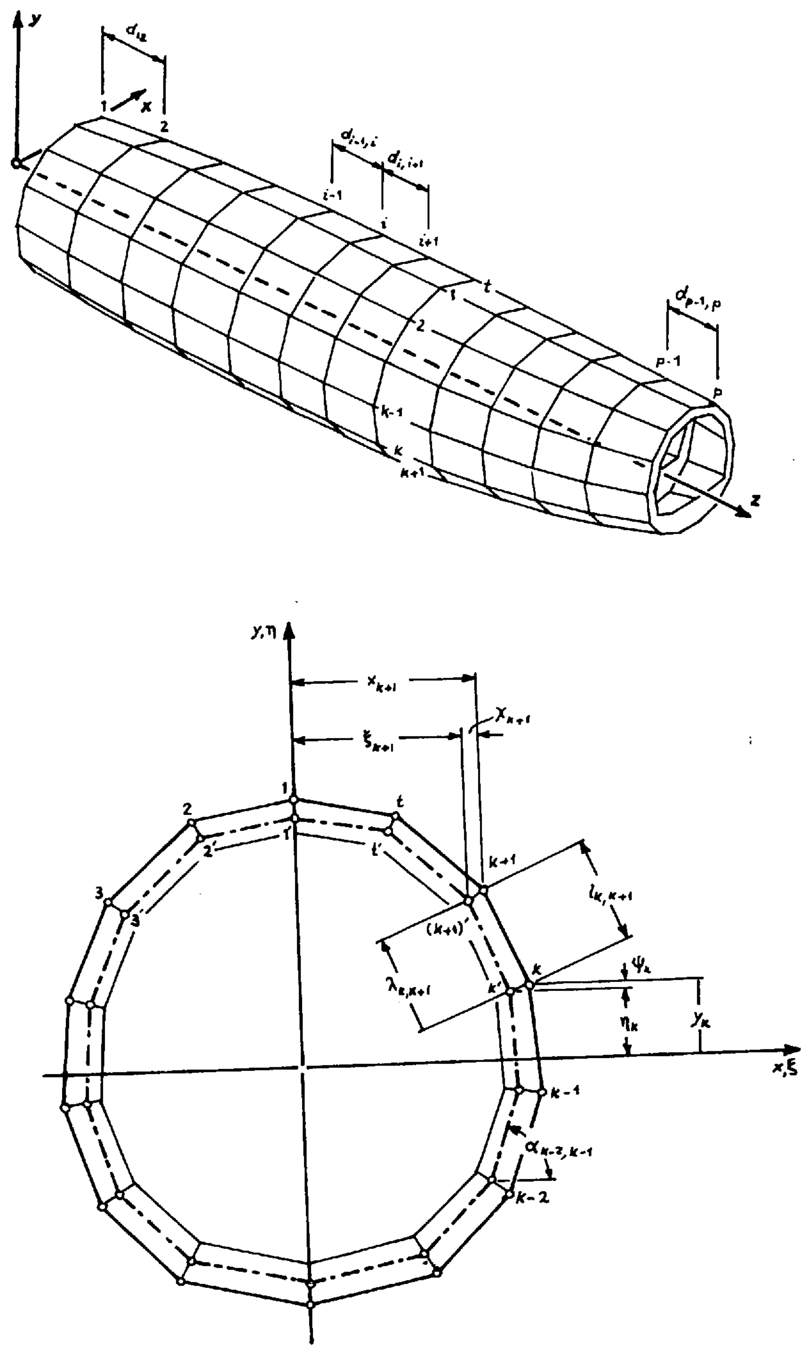


Fig. I,1

Geometry of Regularized Fuselage (see also Ref. (2))

The super-matrix of the field forces

$$\underset{[(p-1)\times 1,\, t\times 1]}{Q} = \left\{ Q_{1,2} \;\; Q_{2,3} \;\; Q_{3,4} \;\cdots\cdots\; Q_{p-1,p} \right\} \tag{I,11}$$

and the associated loading matrix of shear forces and torques

$$\underset{[2(p-1)\times 1,\, 3\times p]}{\hat{R}_q} = \left\{ \hat{R}_{q_{1,2}} \; \hat{R}_{q_{1,2}} \;\; \hat{R}_{q_{2,3}} \; \hat{R}_{q_{2,3}} \;\cdots\cdots\; \hat{R}_{q_{p-1,p}} \; \hat{R}_{q_{p-1,p}} \right\} \tag{I,12}$$

where

$$\hat{R}_{q_{i,i+1}} = \left\{ F_x \quad F_y \quad T \right\}_{i,\, i+1} \tag{I,12a}$$

(see Eqn.(II,26))

and

$$\underset{[(p-1)/,(1)]}{d} = \left\lceil d_{1,2} \;\; d_{2,3} \;\; d_{3,4} \cdots\cdots\; d_{p-1,p} \right\rfloor \tag{I,13}$$

is the diagonal matrix of the bay lengths.

The super-matrices for the x and y co-ordinates of the vertices of the outer polygon.are

$$\underset{[p\times 1,\, t\times 1]}{X} = \left\{ X_1 \;\; X_2 \;\; X_3 \;\cdots\cdots\; X_p \right\} \tag{I,14}$$

and

$$\underset{[p\times 1,\, t\times 1]}{y} = \left\{ y_1 \;\; y_2 \;\; y_3 \;\cdots\cdots\; y_p \right\} \tag{I,14a}$$

where X_i etc. is the $(t\times 1)$ column matrix of the co-ordinates of the outer polygon at frame station i .

Similarly the co-ordinates of the inner polygon or neutral axis of the rings are given by

$$\underset{[p\times 1,\, t\times 1]}{\xi} = \{ \xi_1 \; \xi_2 \; \xi_3 \cdots\cdots\cdots\cdots \xi_p \} \tag{I,15}$$

and

$$\underset{[p\times 1,\, t\times 1]}{\eta} = \{ \eta_1 \; \eta_2 \; \eta_3 \cdots\cdots\cdots \eta_p \} \tag{I,15a}$$

The flange areas, defined at the same stations as the flange loads are continued in the column super-matrix

$$\underset{[2(p-1)\times 1,\, t\times 1]}{B} = \{ B_{1+} \; B_{2-} \; B_{2+} \; B_{3-} \; B_{3+} \cdots\cdots B_{(p-1)_-} \; B_{(p-1)_+} \; B_{p-} \} \tag{I,16}$$

The lengths of the side of the outer polygon

$$\underset{[p\times 1,\, t\times 1]}{l} = \{ l_1 \; l_2 \; l_3 \cdots\cdots\cdots\cdots\cdots l_p \} \tag{I,17}$$

is obtained from the x , y super-matrices in the computer.

Also,

$$\underset{[p\times 1,\, t\times 1]}{\lambda} = \{ \lambda_1 \; \lambda_2 \; \lambda_3 \cdots\cdots\cdots\cdots \lambda_p \} \tag{I,17a}$$

stands for the super column vector of the lengths of the ring elements measured along the centre-line (inner polygon); it is derived from the ξ , η vectors.

The web thicknesses are reproduced as

$$\underset{[(p-1)\times 1, t\times 1]}{\mathbf{t}} = \left\{ t_1 \quad t_2 \quad t_3 \quad \cdots\cdots\cdots \quad t_{(p-1)} \right\} \tag{I,18}$$

The super column vectors of the cross-sectional areas of the rings just before and after the vertices are assembled as

$$\begin{aligned} \underset{[p\times 1, t\times 1]}{A_-} &= \left\{ A_{1-} \quad A_{2-} \quad A_{3-} \quad \cdots\cdots\cdots \quad A_{p-} \right\} \\ \underset{[p\times 1, t\times 1]}{A_+} &= \left\{ A_{1+} \quad A_{2+} \quad A_{3+} \quad \cdots\cdots\cdots \quad A_{p+} \right\} \end{aligned} \tag{I,19}$$

This presentation allows for sudden changes of the ring cross-section of each vertex.

Similarly

$$\begin{aligned} \underset{[p\times 1, t\times 1]}{C_-} &= \left\{ C_{1-} \quad C_{2-} \quad C_{3-} \quad \cdots\cdots\cdots \quad C_{p-} \right\} \\ \underset{[p\times 1, t\times 1]}{C_+} &= \left\{ C_{1+} \quad C_{2+} \quad C_{3+} \quad \cdots\cdots\cdots \quad C_{p+} \right\} \end{aligned} \tag{I,20}$$

are the super column vectors of the areas of the ring cross-sections effective in shear, just before and after the vertices.

And

$$\begin{aligned} \underset{[p\times 1, t\times 1]}{J_-} &= \left\{ J_{1-} \quad J_{2-} \quad J_{3-} \quad \cdots\cdots\cdots \quad J_{p-} \right\} \\ \underset{[p\times 1, t\times 1]}{J_+} &= \left\{ J_{1+} \quad J_{2+} \quad J_{3+} \quad \cdots\cdots\cdots \quad J_{p+} \right\} \end{aligned} \tag{I,21}$$

are the corresponding super column vectors of the ring moments of inertia.

Of importance are, furthermore, the super column matrices

$$\underset{[p\times 1,\ t\times 1]}{S_\alpha} = \left\{ S_{\alpha_1}\ \ S_{\alpha_2}\ \ S_{\alpha_3} \cdots\cdots\cdots S_{\alpha_p} \right\} \tag{I,22}$$

and

$$\underset{[p\times 1,\ t\times 1]}{C_\alpha} = \left\{ C_{\alpha_1}\ \ C_{\alpha_2}\ \ C_{\alpha_3} \cdots\cdots\cdots C_{\alpha_p} \right\} \tag{I,22a}$$

which contain the sines and cosines respectively of the angles made by the sides of the inner polygon with the x-axis.

S is the super-matrix of the final stresses in all elements of the fuselage. It contains as many columns as there are loading cases and is partitioned in the horizontal direction into cover stresses and ring stresses.

$$S = \begin{bmatrix} S_c \\ S_r \end{bmatrix} \tag{I,23}$$

where the suffix c always stands for cover and S for rings.

The cover "stresses" matrix S_c is again composed of sub-matrices for flange loads (suffix ℓ) and panel field forces+ (suffix q)

$$S_c = \begin{bmatrix} S_\ell \\ S_q \end{bmatrix} \tag{I,23a}$$

+ see Ref.(2)

Also the ring stress matrix is divided into sub-matrices for normal forces (suffix N) shear forces (suffix F) and bending moments (suffix M)

$$S_r = \begin{bmatrix} S_N \\ S_F \\ S_M \end{bmatrix} \qquad (I,23b)$$

If we apply the simplified scheme, whereby S_F is directly derivable from S_M , the sub-matrix S_F may be omitted. The same applies to all other super-matrices involving a sub-matrix with a subscript F .

b_o is the corresponding stress matrix due to the loading in the basic system. It is partitioned exactly in the same manner.

$$b_o = \begin{bmatrix} b_{oc} \\ b_{or} \end{bmatrix} \qquad (I,24)$$

where

$$b_{oc} = \begin{bmatrix} b_{o\ell} \\ b_{oq} \end{bmatrix} \qquad (I,24a)$$

and

$$b_{or} = \begin{bmatrix} b_{oN} \\ b_{oF} \\ b_{oM} \end{bmatrix} \qquad (I,24b)$$

b_1 is the stress matrix due to the unit primary redundancies. It has as many linearly independent columns as there are primary unknowns. The horizontal partitioning follows that of S and b_o .

We have

$$b_1 = \begin{bmatrix} b_{1c} \\ b_{1r} \end{bmatrix} \qquad (I,25)$$

where

$$b_{1c} = \begin{bmatrix} b_{1\ell} \\ b_{1q} \end{bmatrix} \qquad (I,25a)$$

and

$$b_{1r} = \begin{bmatrix} b_{1N} \\ b_{1F} \\ b_{1M} \end{bmatrix} \qquad (I,25b)$$

b_2 is the matrix of stresses due to the secondary redundancies in the rings. It is obvious that the latter only affect the rings. Hence

$$b_2 = \begin{bmatrix} 0 \\ b_{2r} \end{bmatrix} \qquad (I,26)$$

where

$$b_{2r} = \begin{bmatrix} b_{2N} \\ b_{2F} \\ b_{2M} \end{bmatrix} \qquad (I,26a)$$

f is the diagonal super-matrix of the flexibility matrices of all independent elements. It follows in its "structure" the same scheme as the stress matrices.

$$f = \lceil f_c \quad f_r \rfloor \tag{I,27}$$

where again

$$f_c = \lceil f_\ell \quad f_q \rfloor \tag{I,27a}$$

and

$$f_r = \lceil f_N \quad f_F \quad f_M \rfloor \tag{I,27b}$$

R is the column matrix of external applied loads.

H is the matrix of initial strains in the elements due to temperature, lack of fit etc. It is assembled in the by now standard form

$$H = \begin{bmatrix} H_c \\ H_r \end{bmatrix} \tag{I,28}$$

where

$$H_c = \begin{bmatrix} H_\ell \\ H_q \end{bmatrix} \tag{I,28a}$$

and

$$H_r = \begin{bmatrix} H_N \\ H_F \\ H_M \end{bmatrix} \tag{I,28b}$$

If initial strains are applied for the specific purpose of simulating cut-outs and/or modifications the matrix $\mathbf{H}$ is rearranged as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_h \\ \mathbf{H}_g \end{bmatrix} = \begin{bmatrix} \mathbf{H}_h \\ 0 \end{bmatrix} \qquad \text{(I,23c)}$$

where h are the elements subject to the specified physical changes and g are the elements which remain unaltered.

$\mathbf{B}$ is the auxiliary matrix from which the $\mathbf{b}_o$ and $\mathbf{b}_1$ are computed. We obtain two distinct matrices

$\mathbf{B}_\ell$ to obtain $\mathbf{b}_{1\ell}$ and $\hat{\mathbf{b}}_{o\ell}$ (flange stress systems due to unit resultant loads)
and $\mathbf{B}_q$ to obtain $\mathbf{b}_{1q}$ and $\hat{\mathbf{b}}_{oq}$ (shear panel stress systems due to unit resultant loads)

$\mathbf{C}$ is the symbol for the special matrix which is inverted to obtain automatically $\mathbf{B}$. It is obvious that we will again have two separate matrices for the cover, $\mathbf{C}_\ell$ for the flanges and $\mathbf{C}_q$ for the shear panels.

Both the self-equilibrating and basic system in the rings are directly computed from the corresponding cover systems.

Ω_ℓ is the auxiliary trigonometrical matrix used in setting up $\mathbf{C}$.

$\mathbf{D}_{\alpha\alpha} = \mathbf{b}_\alpha^t \mathbf{f} \; \mathbf{b}_\alpha$ is the matrix of the influence coefficients d_{ij} of a set of redundancies α.

$\mathbf{D}_{\alpha o} = \mathbf{b}_\alpha^t \mathbf{f} \; \mathbf{b}_o$ is the matrix of the deflections in the basic system in the direction of the set of self-equilibrating systems due to the loading.

$\mathbf{Y}$ is the column matrix of primary (cover) redundancies.

$\mathbf{X}$ is the column vector of secondary ring redundancies.

For further details about the basic theory as well as the special fuselage theory we refer the reader to Argyris, Refs. (1), (2).

CHAPTER II

PROGRAMMING THE CONTINUOUS SINGLE-CELL FUSELAGE USING A SUPER-MATRIX SCHEME

II-a Overall Equilibrium Conditions at a Frame Station

We first refer the reader to Argyris (Ref. 2) for the equilibrium conditions of flanges and shear flows which serve as basis for our further development; see Eqs. ((II,1 to II,28))

In the present case we express all equilibrium relations as the super-matrix equation

$$a_\ell^t \; P = \hat{R}_\ell \qquad \text{(II,1)}$$

where P and $\hat{R}_\ell$ are defined in Eqs. (I,9) , (I,10) and a_ℓ^t is given by

$$\underset{[2(p-1)/,\, 3\times t]}{a_\ell^t} = \left[\begin{array}{c|c|c|c} \begin{matrix} e_1^t \\ x_1^t \\ y_1^t \end{matrix} & 0 & \cdots\cdots & 0 \\ \hline 0 & \begin{matrix} e_2^t \\ x_2^t \\ y_2^t \end{matrix} & \cdots\cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots\cdots & \begin{matrix} e_p^t \\ x_p^t \\ y_p^t \end{matrix} \end{array}\right] \qquad \text{(II,2)}$$

To obtain $\mathbf{a}$ the following programme is suggested:-

a) Build $\mathbf{e}$ which is best achieved by forming a Boolean $\mathbf{e}_B$. Since this vector has subsequently to be joined to other matrices in floating point it is necessary to multiply by a floating point 'one'

b) Using a) and the definitions (I,14, 14a) we form the matrix

$$\underset{[p\times 1,\, t\times 3]}{\bar{\mathbf{a}}^{t}_{\ell v}} = \left[\underset{[p\times 1,\, t\times 1]}{\mathbf{e}} \quad \underset{[p\times 1,\, t\times 1]}{\mathbf{x}} \quad \underset{[p\times 1,\, t\times 1]}{\mathbf{y}}\right] \tag{II,3}$$

c) We next set up the Boolean super-operator $\boldsymbol{\mathcal{E}}_\ell$ which is a uniform super-matrix whose sub-matrices are scalar matrices of arbitrary order and contain either zero or unit elements. Naturally we do not store such matrices in full, but rather by a special convention which is understood by the super-matrix code (see Appendix B)
A simple order is used to form the operator from

$$\underset{[2(p-1)\times p,\,(1)]}{\boldsymbol{\mathcal{E}}_\ell} = \boldsymbol{\mathcal{E}}_{\ell_+} + \boldsymbol{\mathcal{E}}_{\ell_-}$$

$$= \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,\,(1)]_{(2i-1)}}{\mathbf{e}} \; \underset{[p\times 1,\,(1)]_i}{\mathbf{e}^t} + \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,\,(1)]_{2i}}{\mathbf{e}} \; \underset{[p\times 1,\,(1)]_{(i+1)}}{\mathbf{e}^t} \tag{II,4}$$

d) By a simple multiplication we obtain

$$\boldsymbol{\mathcal{E}}_\ell \quad \bar{\mathbf{a}}^{t}_{\ell_v} = \underset{[2(p-1)\times 1,\, t\times 3]}{\mathbf{a}^{*\,t}_{\ell v}} \tag{II,5}$$

e) The super-matrix code provides for the facility of transposing the sub-matrices of this column matrix, without actually transposing the matrix itself,thus

$$\text{TRANSPOSE ELEMENTS of} \quad \underset{[2(p-1)\times 1,\ t\times 3]}{\mathbf{a}_{\ell v}^{*t}}$$

$$\text{to get} \quad \underset{[2(p-1)\times 1,\ 3\times t]}{\mathbf{a}_{\ell v}^{t}}$$

f) By using again the facility available by the super-matrix scheme to re-arrange a column super-matrix as a diagonal one we have

$$\text{REARRANGE} \quad \underset{[2(p-1)\times 1,\ 3\times t]}{\mathbf{a}_{\ell v}^{t}} \quad \text{in diagonal form}$$

$$\text{to give} \quad \underset{[2(p-1)/,\ 3\times t]}{\mathbf{a}_{\ell}^{t}} \quad \text{and the desired matrix of Eqn.(II,2)}$$

It is now possible to write the programme in a concise form as follows

§ 1 Form Boolean matrix $\underset{[p\times 1,\ t\times 1]}{\mathbf{e}_B} = \sum_{i=1}^{pt} \mathbf{e}_{[pt\times 1]_i}$

§ 2 Multiply $\mathbf{e}_B$ with floating point one to get $\underset{[p\times 1,\ t\times 1]}{\mathbf{e}}$

§ 3 Collect $\underset{[p\times 1,\ t\times 3]}{\bar{\mathbf{a}}_{\ell v}^{t}} = \left[\underset{[p\times 1,\ t\times 1]}{\mathbf{e}} \quad \underset{[p\times 1,\ t\times 1]}{\mathbf{x}} \quad \underset{[p\times 1,\ t\times 1]}{\mathbf{y}} \right]$

§ 4 Form (Boolean) $\underset{[2(p-1)\times p,(1)]}{\mathcal{E}_{\ell_+}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,(1)]_{(2i-1)}}{e} \; \underset{[p\times 1,(1)]_i}{e^t}$

§ 5 Form (Boolean) $\underset{[2(p-1)\times p,(1)]}{\mathcal{E}_{\ell_-}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,(1)]_{2i}}{e} \; \underset{[p\times 1,(1)]_{i+1}}{e}$

§ 6 Form (Boolean) $\underset{[2(p-1)\times p,(1)]}{\mathcal{E}_{\ell}} = \mathcal{E}_{\ell_+} + \mathcal{E}_{\ell_-}$

§ 7 Form $\underset{[2(p-1)\times 1, t\times 3]}{a^{*\,t}_{\ell v}} = \underset{[2(p-1)\times p,(1)]}{\mathcal{E}_{\ell}} \; \underset{[p\times 1, t\times 3]}{\bar{a}_{\ell v}}$

§ 8 Transpose Elements of $\underset{[2(p-1)\times 1, t\times 3]}{a^{*\,t}_{\ell v}}$ to give $\underset{[2(p-1)\times 1, 3\times t]}{a^{t}_{\ell v}}$

§ 9 Rearrange $\underset{[2(p-1)\times 1, 3\times t]}{a^{t}_{\ell v}}$ as a diagonal to give $\underset{[2(p-1)/, 3\times t]}{a^{t}_{\ell}}$

§ 1o Stop

This involves in all 9 matrix orders

II-a-2 Equilibrium of Shear Panels

Following Ref. (2) we derive here the super-matrix transformations relating the matrices Q of the field forces (see Eqn.(I,11))with the corresponding loading matrix $\hat{R}_q$ (Eqn.(I,12)). Due to the general nature of the taper, and the idealization of the panels, we can set up distinct equilibrium conditions at either end of a bay, obtaining nevertheless the same load-resultant matrix. This may serve as a further check for the numerical operations in the computer.

Applying the standard notation (+) and (-) for the fore and aft stations at a frame we have the two super-matrix equations

$$a_{q_+}^t \; Q \; + \; a_{T_+}^t \; P \; = \; \bar{\varepsilon}_{\ell_-}^t \; \hat{R}_q \qquad \text{(II,6)}$$

or

$$a_{q_-}^t \; Q \; + \; a_{T_-}^t \; P \; = \; \bar{\varepsilon}_{\ell_+}^t \; \hat{R}_q \qquad \text{(II,6a)}$$

The matrices a_{q_+} , a_{q_-} , a_{T_+} , a_{T_-} are defined below.

Alternatively, we can join the two expressions in the single relation

$$\bar{\varepsilon}_{\ell_-}\left[a_{q_+}^t Q + a_{T_+}^t P \right] + \bar{\varepsilon}_{\ell_+}\left[a_{q_-}^t Q + a_{T_-}^t P \right] = \hat{R}_q \qquad \text{(II,7)}$$

or better still

$$\left[\bar{\varepsilon}_{\ell_-} \; a_{q_+}^t + \bar{\varepsilon}_{\ell_+} \; a_{q_-}^t \right] Q + \left[\bar{\varepsilon}_{\ell_-} \; a_{T_+}^t + \bar{\varepsilon}_{\ell_+} a_{T_-} \right] P \; = \; \hat{R}_q \qquad \text{(II,7a)}$$

The method of obtaining the matrices $\mathbf{a}_{q_+}$, $\mathbf{a}_{q_-}$, $\mathbf{a}_{\tau_+}$ and $\mathbf{a}_{\tau_-}$ will be now described.

A) The Matrix $\mathbf{a}_{q_+}$

A-a) We form the rotational operator $\underset{[(),t]}{I_a}$ defined in (I,6) from which we obtain the operators

$$\underset{[(),t]}{\beta} = \underset{[(),t]}{I} + \underset{[(),t]}{I_a^t} \qquad \text{(II,8)}$$

and

$$\underset{[(),t]}{\alpha} = -\underset{[(),t]}{I} + \underset{[(),t]}{I_a^t} \qquad \text{(II,9)}$$

A-b) We next set up the difference matrix X_Δ (see Eqn.((I,6)) by the operation

$$\begin{aligned} \underset{[p\times 1,t\times 1]}{X_\Delta} &= \left\{ X_{\Delta 1} \quad X_{\Delta 2} \quad \cdots\cdots\cdots \quad X_{\Delta p} \right\} \\ &= -\underset{[(),t]}{\alpha^t}\ \underset{[p\times 1,t\times 1]}{X} \end{aligned} \qquad \text{(II,10)}$$

A-c) In exactly the same manner we form

$$\underset{[p\times 1,t\times 1]}{y_\Delta} = -\underset{[(),t]}{\alpha^t}\ \underset{[p\times 1,t\times 1]}{y} \qquad \text{(II,10a)}$$

A-d) As the super-matrix scheme is capable of splitting a matrix (see Appendix C) into two matrices, we can issue an order to have

$$\underset{[p\times 1,t\times 1]}{X_\Delta} = \cfrac{\underset{[1\times 1,t\times 1]}{X_{\Delta 1}}}{\underset{[(p-1)\times 1,t\times 1]}{X_{\Delta +}}} \qquad \text{(II,11)}$$

A-e) In the same manner we split

$$\underset{[p\times 1\,,\,t\times 1]}{y_{\triangle}} = \begin{array}{c} \underset{[1\times 1\,,\,t\times 1]}{y_{\triangle 1}} \\ \hline \underset{[(p-1)\times 1\,,\,t\times 1]}{y_{\triangle +}} \end{array} \tag{II,11a}$$

On the other hand if no super-matrix scheme is available for the computer we can always apply for this purpose an operator

$$\underset{[(p-1)\times p,()]}{L_{+}} = \sum_{i=1}^{p-1} \underset{[(p-1)\times 1,()]_i}{e} \; \underset{[p\times 1\;,()]_{(i+1)}}{e^{t}} \tag{II,12}$$

which proves also useful in our subsequent developments. We have

$$x_{\triangle +} = L_{+} \; x_{\triangle} \tag{II,11b}$$

and

$$y_{\triangle +} = L_{+} \; y_{\triangle} \tag{II,11c}$$

Similarly we build up the operator

$$\underset{[(p-1)\times p\,,()]}{L_{-}} = \sum_{i=1}^{p-1} \underset{[(p-1)\times 1,()]_i}{e} \; \underset{[p\times 1\,,\,()\,]_i}{e^{t}} \tag{II,12a}$$

A-f) We rearrange the two super-matrices $x_{\triangle +}$ and $y_{\triangle +}$ in diagonal form, i.e.

$$\underset{[(p-1)/\,,\,t/]}{x^{d}_{\triangle +}} \quad \text{and} \quad \underset{[(p-1)/\,,\,t/]}{y^{d}_{\triangle +}}$$

A-g) We form the super-matrices $x_{\Sigma -}$ and $y_{\Sigma -}$ (see Eqs. (I,3a))

$$\underset{[(p-1)\times 1\,,\,t\times 1]}{x_{\Sigma -}} = \left(\tfrac{1}{2}\right) L_{-} \underset{[()\,,\,t\,]}{\beta^{t}} \; x \tag{II,13}$$

and

$$\underset{[(p-1)\times 1,\, t\times 1]}{y_{\Sigma-}} = \left(\tfrac{1}{2}\right) L_{-} \underset{[(\,),\, t\,]}{\beta^{t}} \; y \qquad \text{(II,13a)}$$

A-h) We form the column super-matrix

$$\underset{[(p-1)\times 1,\, t\times 1]}{\Omega_{+}} = y^{d}_{\Delta+}\, x_{\Sigma_-} - x^{d}_{\Delta+}\, y_{\Sigma_-} \qquad \text{(II,14)}$$

A-i) With an order similar to that described under (II,a(b)) we join the three resulting column super-matrices into one matrix

$$\underset{[(p-1)\times 1,\, t\times 3]}{\bar{a}^{t}_{qv+}} = \left[\underset{[(p-1)\times 1,\, t\times 1]}{x_{\Delta+}} \; \underset{[(p-1)\times 1,\, t\times 1]}{y_{\Delta+}} \; \underset{[(p-1)\times 1,\, t\times 1]}{\Omega_{+}} \right] \qquad \text{(II,15)}$$

A-j) Finally, we transpose the elements of the matrix and rearrange them in a diagonal form as described before.

The programme is now reproduced in the concise arrangement

F O R M A T I O N of a_{q+}

§ 1 Form $\underset{[(\,),\, t\,]}{I_a}$

§ 2 Form $\underset{[(\,),\, t\,]}{\alpha} = -\underset{[(\,),\, t\,]}{I^{t}_{a}} + \underset{[(\,),\, t\,]}{I}$

§ 3 Form $\underset{[(\,),\, t\,]}{\beta} = \underset{[(\,),\, t\,]}{I^{t}_{a}} + \underset{[(\,),\, t\,]}{I}$

§ 4 Form $$x_{\Delta} = - \underset{[(),\, t]}{\alpha^{t}} \; \underset{[p\times 1,\, t\times 1]}{x}$$

§ 5 Form $$y_{\Delta} = - \underset{[(),\, t]}{\alpha} \; \underset{[p\times 1,\, t\times 1]}{y}$$

§ 6 Form $$L_{+} = \sum_{i=1}^{(p-1)} e_{[(p-1)\times 1,\, ()]_{i}} \; e^{t}_{[p\times 1,\, ()]_{(i+1)}}$$

§ 7 Form $$L_{-} = \sum_{i=1}^{(p-1)} e_{[(p-1)\times 1,\, ()]_{i}} \; e^{t}_{[p\times 1,\, ()]_{i}}$$

§ 8 Form $$x_{\Delta +} = L_{+} \; x_{\Delta}$$

§ 9 Form $$y_{\Delta +} = L_{+} \; y_{\Delta}$$

§ 1o Rearrange $\underset{[(p-1)\times 1,\, t\times 1]}{x_{\Delta +}}$ in a diagonal to give $\underset{[(p-1)/,\, t/]}{x^{d}_{\Delta +}}$

§ 11 Rearrange $\underset{[(p-1)\times 1,\, t\times 1]}{y_{\Delta +}}$ in a diagonal to give $\underset{[(p-1)/,\, t/]}{y^{d}_{\Delta +}}$

§ 12 Form $$x_{\Sigma -} = \left(\tfrac{1}{2}\right) L_{-} \underset{[(),\, t]}{\beta^{t}} \; x$$

§ 13 Form $$y_{\Sigma -} = (\tfrac{1}{2})\, L_{-} \underset{[1,t]}{\beta^{t}}\, y$$

§ 14 Form $$\Omega_{+} = y_{\Delta +}^{d}\, x_{\Sigma -} - x_{\Delta +}^{d}\, y_{\Sigma -}$$

§ 15 Form by joining $$\underset{[(p-1)\times 1,\, t\times 3]}{a^{*t}_{qv+}} = \left[\underset{[(p-1)\times 1,\, t\times 1]}{x_{\Delta +}} \quad \underset{[(p-1)\times 1,\, t\times 1]}{y_{\Delta +}} \quad \underset{[(p-1)\times 1,\, t\times 1]}{\Omega_{+}} \right]$$

§ 16 Transpose elements of $\underset{[(p-1)\times 1,\, t\times 3]}{a^{*t}_{qv+}}$ to give $\underset{[(p-1)\times 1,\, 3\times t]}{a^{t}_{qv+}}$

§ 17 Rearrange $\underset{[(p-1)\times 1,\, 3\times t]}{a^{t}_{qv+}}$ in a diagonal form to give $\underset{[(p-1)/,\, 3\times t]}{a^{t}_{q+}}$

§ 18 Stop

We notice that the more complex operations can be split up into simpler operations, or vice versa,that some steps might be joined together to form one step. This is best left to the ingenuity of the coder who is best acquainted with the machine he is using and the supermatrix code in question.

B) The Matrix $\mathbf{a}_{T+}$

Since the programme for $\mathbf{a}_{T+}$ is similar to that used for the previous matrices, it is sufficient to confine our presentation to the final tabular arrangement.

FORMATION of $\mathbf{a}_{T+}$

§ 1 Form $\quad x_{LD} = \left[\underset{[p,()]}{I_a} \quad \underset{[p,()]}{-I} \right] x$

§ 2 Form $\quad y_{LD} = \left[\underset{[p,()]}{I_a} \quad \underset{[p,()]}{-I} \right] y$

§ 3 Form $\quad \underset{[px1,tx1]}{x_{[2]}} = \underset{[p,()]}{I_a} \; x$

§ 4 Form $\quad \underset{[px1,tx1]}{y_{[2]}} = \underset{[p,()]}{I_a} \; y$

§ 5 Rearrange $\quad \underset{[px1,tx1]}{x_{[2]}}$ as a diagonal to get $\underset{[p/,t/]}{x^{d}_{[2]}}$

§ 6 Rearrange $\quad \underset{[px1,tx1]}{y_{[2]}}$ as a diagonal to get $\underset{[p/,t/]}{y^{d}_{[2]}}$

§ 7 Form $\underset{[p\times 1,\, t\times 1]}{\Omega_{T+}} = y_{[2]}^{d}\, x - x_{[2]}^{d}\, y$

§ 8 Form by joining $\bar{a}_{TV}^{*t} = \underset{[(p-1)\times p,(1)]}{\bar{L}} \left[\underset{[p\times 1,\, t\times 1]}{x_{LD}} \quad \underset{[p\times 1,\, t\times 1]}{y_{LD}} \quad \underset{[p\times 1,\, t\times 1]}{\Omega_{T+}} \right]$

§ 9 Transpose elements of $\underset{[(p-1)\times 1,\, t\times 3]}{\bar{a}_{TV}^{*t}}$ to give $\underset{[(p-1)\times 1,\, 3\times t]}{\bar{a}_{TV}^{t}}$

§ 1o Rearrange $\underset{[(p-1)\times 1,\, 3\times t]}{\bar{a}_{TV*}^{t}}$ as diagonal to give $\bar{a}_{T+}^{t}$

§ 11 Form operator $\underset{[(p-1)\times 2(p-1),(1)]}{\mathcal{E}_{T+}} = \sum_{i=1}^{p-1} \underset{[(p-1)\times 1,(1)]_i}{e} \; \underset{[1\times 2(p-1),(1)]_{(2i-1)}}{e}$

§ 12 Form $\underset{[(p-1)\times 2(p-1),\, 3\times t]}{a_{T+}^{t}} = \bar{a}_{T+}^{t} \quad \mathcal{E}_{T+}$

§ 13 Stop

Now the equilibrium at the ends of the bays may be checked by the simple relation

$$a_{q+}^{t}\, Q + a_{T+}^{t}\, P = \hat{R}_{q}$$

In order to check the equilibrium at the (-) stations of the frames we have to form the remaining two super-matrices $\mathbf{a}_{q-}$ and $\mathbf{a}_{T-}$. Again only the summarized programmes are given. The reader's attention is drawn to the fact that some of the intermediate results are common to the previously described programmes. This may shorten the setting up of $\mathbf{a}_{q-}$ and $\mathbf{a}_{T-}$ but is best left to the coder.

C) The Matrix $\mathbf{a}_{q-}$

FORMATION of $\mathbf{a}_{q-}$

§ 1 Form $\quad x_{\Delta} = - \underset{[0,t]}{\alpha^{t}} \; x$

§ 2 Form $\quad y_{\Delta} = - \underset{[0,t]}{\alpha^{t}} \; y$

§ 3 Form $\quad x_{\Delta -} = L_{-} \; x_{\Delta}$

§ 4 Form $\quad y_{\Delta -} = L_{-} \; y_{\Delta}$

§ 5 Rearrange $x_{\Delta -}$ as diagonal to give $\underset{[(p-1)/,\, t/]}{x^{d}_{\Delta -}}$

§ 6 Rearrange $y_{\Delta -}$ as diagonal to give $\underset{[(p-1)/ ,\ t/]}{y^{d}_{\Delta -}}$

§ 7 Form $x_{\Sigma +} = (\tfrac{1}{2})\ L_{+} \underset{[(), t]}{\beta^{t}}\ x$

§ 8 Form $x_{\Sigma +} = (\tfrac{1}{2})\ L_{+} \underset{[(), t]}{\beta^{t}}\ y$

§ 9 Form $\Omega_{-} = y^{d}_{\Delta -}\ x_{\Sigma +} - x^{d}_{\Delta -}\ y_{\Sigma +}$

§ 1o Form by joining $a^{*t}_{qv-} = \left[\ x_{\Delta -} \quad y_{\Delta -} \quad \Omega_{-}\ \right]$

§ 11 Transpose elements of a^{*t}_{qv-} to give $\underset{[(p-1)\times 1,\ 3\times t]}{a^{t}_{qv-}}$

§ 12 Rearrange a^{t}_{qv-} as diagonal to give $\underset{[(p-1)/ ,\ 3\times t]}{a^{t}_{q-}}$

§ 13 Stop

D) <u>The Matrix a_{T-}</u>

FORMATION of a_{T-}

The programme is exactly similar to that for a_{T+} except for the operator formed in § 11 and used in § 12. This should be now

$$\varepsilon_{T-} = \sum_{i=1}^{p-1} e_{[(p-1)\times 1,\,(\,)]_{i}} \; e^{t}_{[2(p-1)\times 1,\,(\,)]_{2i}} \qquad \text{(II,1[illegible])}$$

The complete equilibrium conditions at the (-) ends are

$$a^{t}_{q-} \; Q + a^{t}_{T-} \; P = \hat{R}_{q}$$

II-b The Automatic Formation of the Self-Equilibrating Stress Systems in the Cover

Again as in the previous section, the formation of the basic and self-equilibrating stress systems is based on the further development of the ideas laid down in Ref. (2). The readers attention is particularly drawn to Eqs. ((IV,1 to IV,52))

II-b 1 Flange Loads

a) The first step in setting up the sequence of operation is the choice of the so-called conditioning matrix A_{ℓ} . One may select for instance the orthogonal matrices for the cylindrical fuselage, i.e. the Ω_ℓ , which may be generated by the fully automatic sub-programme described in Appendix D. As in the original publication one may also use any other standard matrix for which a special sub-programme has to be written. It is doubtful, however, that any simple matrix may yield a better conditioning than Ω_ℓ . A third possibility is to introduce certain computed matrices, based upon the orthogonalisation technique described in Ref. (2). The exact manner in which this can be automatised is set out further below. We only note here that in the most general case, this matrix may vary from one frame station to another. The symbol used for the typical conditioning matrix of a station i is

$$A_{\ell_i}$$

If the ends of the fuselage with p frame stations are taken to be subjected to known forces, the determination for the self-equilibrating stress systems is restricted to the $(p-2)$ intermediate stations. Correspondingly the appropriate super-matrix of the A_{ℓ_i} matrices is strictly,

$$\underset{[(p-2)/\,,\;t\times(t-3)]}{A_{\ell m}} = \lceil A_{\ell_2} \quad A_{\ell_3} \cdots\cdots A_{\ell_i} \cdots\cdots A_{\ell_{(p-1)}} \rfloor \qquad \text{(II,17)}$$

However, for reasons connected with the formation of the basic system in the cover, we must also introduce fictitious systems at frame stations 1 and p . To this purpose we use two extra sub-matrices, which we simply obtain by repeating the first and last sub-matrices. Hence, the final A_ℓ super-matrix takes the form

$$\underset{[p/\,,\;t\times(t-3)]}{A_\ell} = \lceil A_{\ell_2} \quad A_{\ell_2} \quad A_{\ell_3} \cdots A_{\ell_i} \cdots A_{\ell_{(p-2)}} \quad A_{\ell_{(p-1)}} \quad A_{\ell_{(p-1)}} \rfloor$$

(II,18)

b) We partition a unit matrix $\underset{(1,t)}{I}$ in two parts, of which the first contains 3 columns and the second the last $(t-3)$ columns.
Thus we represent the formation of these Boolean operators as follows:-

$$I_{(t)} = \begin{bmatrix} I_{(3)} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ I_{(t-3)} \end{bmatrix} = \underset{(t\times 3)}{u_a} \Big] \Big[\underset{(t\times(t-3))}{u_A} \qquad (II,19)$$

c) We set up $$\underset{[p/,t]}{C_{\ell a}} = \underset{[1,t\times 3]}{u_a} \quad a_\ell^t \qquad (II,20)$$

d) We transpose the elements of the super-matrix A_ℓ to obtain

$$\underset{[p/,(t-3)\times t]}{A_\ell^t}$$

e) We similarly form

$$\underset{[p/,t]}{C_{\ell A}} = \underset{[1,t\times(t-3)]}{u_A} \quad A_\ell^t \qquad (II,21)$$

We then add the two matrices (II,20,21) to find the super diagonal matrix

$$\underset{[p/,t]}{C_\ell} = C_{\ell a} + C_{\ell A} \qquad (II,22)$$

f) Inverting Eqn.(II,22), we determine the basic matrix from which we derive the flange load distribution for the basic and self-equilibrating systems; (see Eqn. ((IV,22))).

$$\underset{[p/,t]}{B_\ell} = C_\ell^{-1} \qquad (II,23)$$

g) In order to get the self-equilibrating system we construct the operator

$$\underset{[(p-2)\times p,\,(\,)]}{\mathcal{E}_m} = \sum_{i=1}^{p-2} \underset{[(p-2)\times 1,\,(\,)]_i}{\mathbf{e}} \quad \underset{[p\times 1,\,(\,)]_{(i+1)}}{\mathbf{e}^t} \qquad \text{(II,24)}$$

h) We now get an extended super-matrix of the self-equilibrating stress systems, namely

$$\underset{[p/,\,t\times(t-3)]}{\mathbf{b}^*_{1\ell e}} = \mathbf{B}_\ell \quad \underset{[(\,),\,t\times(t-3)]}{\mathbf{u}_A} \qquad \text{(II,25)}$$

Applying next the operator $\mathcal{E}_m$, we find

$$\underset{[p\times(p-2),\,t\times(t-3)]}{\mathbf{b}^*_{1\ell}} = \mathbf{b}^*_{1\ell e} \quad \mathcal{E}^t_m \qquad \text{(II,26)}$$

i) Using the originally developed Boolean operator $\mathcal{E}_\ell$ (see Eqn.(II,4)) we obtain the desired repeat pattern before and after each station. Thus

$$\underset{[2(p-1)\times(p-2),\,t\times(t-3)]}{\mathbf{b}_{1\ell}} = \underset{[2(p-1)\times p,\,t]}{\mathcal{E}_\ell} \quad \underset{[p\times(p-2),\,t\times(t-3)]}{\mathbf{b}^*_{1\ell}} \qquad \text{(II,27)}$$

which completes the determination of the self-equilibrating flange loads.

j) To obtain the basic system, i.e. $\mathbf{b}_{o\ell}$, we proceed as follows

$$\underset{[p/,\,t\times 3]}{\hat{\mathbf{b}}^*_{o\ell}} = \mathbf{B}_\ell \quad \underset{[(\,),\,t\times 3]}{\mathbf{u}_a} \qquad \text{(II,28)}$$

which is rearranged as the super vector

$$\underset{[p\times 1,\,t\times 3]}{\hat{\mathbf{b}}^*_{o\ell v}}$$

Hence we form

$$\underset{[2(p-1)\times 1,\, t\times 3]}{\hat{b}_{o\ell v}} = \varepsilon_{\ell} \quad \hat{b}^{*}_{o\ell v} \qquad \text{(II,29)}$$

This is diagonalized to become

$$\underset{[2(p-1)/,\, t\times 3]}{\hat{b}_{o\ell}}$$

We now summarize the formation of $b_{i\ell}$ and $\hat{b}_{o\ell}$

FORMATION of $b_{i\ell}$ and $\hat{b}_{o\ell}$

§ 1 Form $\underset{[(p-2)/,\, t\times(t-3)]}{A_{\ell m}}$ (see Appendix A)

§ 2 Split

$$\underset{[(p-2)/,\, t\times(t-3)]}{A_{\ell m}} = \underline{\underset{[1/,\, t\times(t-3)]}{A_{\ell 2}}}\rfloor \; \lceil \overline{\underline{\underset{[(p-4)/,\, t\times(t-3)]}{A_{\ell mm}}}} \rfloor \; \lceil \overline{\underset{[1/,\, t\times(t-3)]}{A_{\ell(p-1)}}}$$

§ 3 Join

$$\underset{[p/,\, t\times(t-3)]}{A_{\ell}} = \lceil \overline{\underset{[1/,\, t\times(t-3)]}{A_{\ell 2}} \quad \underset{[(p-2)/,\, t\times(t-3)]}{A_{\ell m}} \quad \underline{\underset{[1/,\, t\times(t-3)]}{A_{\ell(p-1)}}}}\rfloor$$

§ 4 Form two Boolean operators $\underset{[1,\, t\times 3]}{u_a}$ and $\underset{[1,\, t\times(t-3)]}{u_A}$

§ 5 $$\underset{[p/,\, t]}{C_{\ell a}} = \underset{[(1),\, t\times 3]}{u_a} \quad a_\ell^t$$

§ 6 Transpose elements of A_ℓ giving $\underset{[p/,\,(t-3)\times t]}{A_\ell^t}$

§ 7 $$\underset{[p/,\, t]}{C_{\ell A}} = \underset{[(1),\, t\times(t-3)]}{u_A} \quad A_\ell^t$$

§ 8 $$\underset{[p/,\, t]}{C_\ell} = C_{\ell a} + C_{\ell A}$$

§ 9 $$\underset{[p/,\, t]}{B_\ell} = C_\ell^{-1}$$

§ 10 $$\underset{[(p-2)\times p,\,(1)]}{\varepsilon_m} = \sum_{i=1}^{(p-2)} \underset{[(p-2)\times 1,\,(1)]_i}{e} \quad \underset{[p\times 1,\,(1)]_{(i+1)}}{e^t}$$

§ 11 $$\underset{[p/,\, t\times(t-3)]}{b^*_{1\ell e}} = B_\ell \quad \underset{[(1),\, t\times(t-3)]}{u_A}$$

§ 12 $$\underset{[p\times(p-2),\, t\times(t-3)]}{b^*_{1\ell}} = b^*_{1\ell e} \quad \varepsilon_m^t$$

§ 13 $$\underset{[2(p-1)\times(p-2),\, t\times(t-3)]}{b_{1\ell}} = \varepsilon_\ell \quad b^*_{1\ell}$$

§ 14 $$\underset{[p/,\ t\times 3]}{\hat{b}_{o\ell}^{*}} = B_\ell \quad u_a$$

§ 15 Rearrange $\hat{b}_{o\ell}^{*}$ as column $\underset{[p\times 1,\ t\times 3]}{\hat{b}_{o\ell v}^{*}}$

§ 16 $$\underset{[2(p-1)\times 1,\ t\times 3]}{\hat{b}_{o\ell v}} = \mathcal{E}_\ell \quad \hat{b}_{o\ell v}^{*}$$

§ 17 Rearrange $\hat{b}_{o\ell v}$ as diagonal $\underset{[2(p-1)/,\ t\times 3]}{\hat{b}_{o\ell}}$

§ 18 Stop

II-b 2 Field Forces

The formation of the basic and self-equilibrating field forces proceeds analogously to that of the flange loads.

a) We initially form the two matrices

$$\underset{[(p-1)/,\ t\times(t-3)]}{A_{\ell +}} = \left\lceil \underset{[(p-2)/,\ t\times(t-3)]}{A_{\ell m}} \quad \underset{[1/,\ t\times(t-3)]}{A_{\ell(p-1)}} \right\rfloor \tag{II,30}$$

and

$$\underset{[(p-1)/,\ t\times(t-3)]}{A_{\ell -}} = \left\lceil \underset{[1/,\ t\times(t-3)]}{A_{\ell 2}} \quad \underset{[(p-2)/,\ t\times(t-3)]}{A_{\ell m}} \right\rfloor \tag{II,30a}$$

b) We derive hence by scalar multiplications

$$A_{q+} = \underset{[(1),t]}{\alpha^{t}} \; A_{\ell+} \tag{II,31}$$

and

$$A_{q-} = \underset{[(1),t]}{\alpha^{t}} \; A_{\ell-} \tag{II,31a}$$

$\underset{[(1),t]}{\alpha}$ having been defined in (II,9)

c) As before we form

$$\underset{[(p-1)/,t]}{C_{qa+}} = \underset{[(1),t\times 3]}{u_a} \; a^{t}_{q+} \tag{II,32}$$

and

$$\underset{[(p-1)/,t]}{C_{qA+}} = \underset{[(1),t\times(t-3)]}{u_A} \; A^{t}_{q+} \tag{II,32a}$$

d) Adding Eqs. (II,32 and 32a) and inverting

$$\underset{[(p-1)/,t]}{C^{-1}_{q+}} = \left[C_{qa+} + C_{qA+} \right]^{-1} \tag{II,33}$$

e) Form by joining

$$\underset{[(p-1)\times p,\,t]}{B_{q+}} = \left[\underset{[(p-1)\times 1,\,t]}{0} \quad \underset{[(p-1)/,\,t]}{C^{-1}_{q+}} \right] \tag{II,34}$$

f) Form by a "scalar" multiplication

$$\underset{[(p-1)/,t]}{C_{qa-}} = \underset{[(1),\,t\times 3]}{u_a} \; a^{t}_{q-} \tag{II,35}$$

and

$$\underset{[(p-1)/,t]}{C_{qA-}} = \underset{[(1),\,t\times(t-3)]}{u_A} \; A^{t}_{q-} \tag{II,35a}$$

g) Once more adding and inverting

$$\underset{[(p-1)^2,\, t]}{C_{q-}^{-1}} = \left[C_{qa-} + C_{qA-} \right]^{-1} \qquad (II,36)$$

h) Form by joining

$$\underset{[(p-1)\times p,\, t]}{B_{q-}} = \left[\underset{[(p-1)^2,\, t]}{C_{q-}^{-1}} \quad \underset{[(p-1)\times 1,\, t]}{0} \right] \qquad (II,37)$$

i) Form by addition

$$\underset{[(p-1)\times p,\, t]}{B_{q}} = B_{q-} + B_{q+} \qquad (II,38)$$

j) We can now form the basic system by two scalar multiplications, followed by rearrangements using the operators $\mathcal{E}_{\ell+}$ and $\mathcal{E}_{\ell-}$

Thus

$$\underset{[(p-1)\times p,\, t\times 3]}{\hat{b}_{oq-}^{*}} = \underset{[(p-1)\times p,\, t]}{B_{q-}} \quad \underset{[(1,\, t\times 3]}{u_a} \qquad (II,39)$$

and similarly $\hat{b}_{oq+}^{*}$. Applying an assembly order

$$\underset{[(p-1)\times 2(p-1),\, t\times 3]}{\hat{b}_{oq}} = \underset{[(p-1)\times p,\, t\times 3]}{\hat{b}_{oq-}^{*}} \; \underset{[2(p-1)\times p,\, 1]}{\mathcal{E}_{\ell+}^{t}} + \underset{[(p-1)\times p,\, t\times 3]}{b_{oq+}^{*}} \; \underset{[2(p-1)\times p,\, 1]}{\mathcal{E}_{\ell-}^{t}} \qquad (II,40)$$

k) In order to form the b_{1q} matrix, we have to proceed the scalar multiplication by a splitting operation of the matrix B_q

$$\underset{[(p-1)\times p,\, t]}{B_{q}} = \underset{[(p-1)\times 1,\, t]}{B_{qa}} \Big) \Big(\underset{[(p-1)\times (p-1),\, t]}{B_{qm}} \Big) \Big(\underset{[(p-1)\times 1,\, t]}{B_{qz}} \qquad (II,41)$$

1) Then we form $$\underset{[(p-1)\times(p-2),\, t\times(t-3)]}{b_{1q}} = \underset{[(p-1)\times(p-2),\, t]}{B_{qm}} \quad \underset{[(),\, t\times(t-3)]}{u_A} \qquad \text{(II, 42)}$$

The complete programme may now be reproduced concisely.

FORMATION of B_q

§ 1 Form by joining

$$\underset{[(p-1)/,\, t\times(t-3)]}{A_{\ell+}} = \left[\underset{[(p-2)/,\, t\times(t-3)]}{A_{\ell m}} \quad \underset{[1/,\, t\times(t-3)]}{A_{\ell(p-1)}} \right]$$

§ 2 Form by joining

$$\underset{[(p-1)/,\, t\times(t-3)]}{A_{\ell-}} = \left[\underset{[1/,\, t\times(t-3)]}{A_{\ell_1}} \quad \underset{[(p-2)/,\, t\times(t-3)]}{A_{\ell_m}} \right]$$

§ 3 Form

$$\underset{[(),\, t]}{\alpha} = \underset{[(),\, t]}{I} - \underset{[(),\, t]}{I_a^t}$$

§ 4 Form $$A_{q+}^t = A_{\ell+}^t \underset{[(),\, t]}{\alpha}$$

§ 5 Form $$A_{q-}^t = A_{\ell-}^t \underset{[(),\, t]}{\alpha}$$

§ 6 Form $$C_{qa+} = u_a \quad a_{q+}^t$$

§ 7 Form $$C_{qA+} = u_A \quad A^t_{q+}$$ (diagonal)

§ 8 Form $$C_{q+} = C_{qa+} + C_{qA+}$$ (diagonal)

§ 9 Form $$C^{-1}_{q+}$$ (diagonal)

§ 1o Form by joining

$$\underset{[(p-1)\times p\ ,t\ ]}{B_{q+}} = \begin{bmatrix} \underset{[(p-1)\times 1,t]}{0} & \underset{[(p-1)\times(p-1),t]}{C} \end{bmatrix}$$

§ 11 Form $$C_{qa-} = u_a \quad a^t_{q-}$$

§ 12 Form $$C_{qA-} = u_A \quad A^t_{q-}$$

§ 13 Form $$C_{q-} = C_{qa-} + C_{qA-}$$ (diagonal)

§ 14 Form $$C^{-1}_{q-}$$ (diagonal)

§ 15 Form by joining

$$\underset{[\ldots]}{B_{q-}} = \begin{bmatrix} \underset{[(p-1)\times(p-1),t]}{C^{-1}_{q-}} & \underset{[(p-1)\times 1,t]}{0} \end{bmatrix}$$

§ 16 Form $\quad B_q = B_{q_-} + B_{q_+}$

§ 17 Stop

FORMATION of $\hat{b}_{oq}$

§ 18
$$\underset{[(p-1)\times p,\, t\times 3]}{\hat{b}^*_{oq_-}} = \underset{[(p-1)\times p,\, t]}{B_{q_-}} \; \underset{[(\,),\, t\times 3]}{u_a}$$

§ 19
$$\underset{[(p-1)\times p,\, t\times 3]}{\hat{b}^*_{oq_+}} = \underset{[(p-1)\times p,\, t]}{B_{q_+}} \; \underset{[(\,),\, t\times 3]}{u_a}$$

§ 20
$$\underset{[(p-1)\times 2(p-1),\, t\times 3]}{\hat{b}_{oq}} = \hat{b}^*_{oq_-} \, \varepsilon^t_{\ell_+} + \hat{b}^*_{oq_+} \, \varepsilon^t_{\ell_-}$$

§ 21 Stop

FORMATION of b_{1q}

§ 22
$$\underset{[(p-1)\times p,\, t]}{B_q} = \underset{[(p-1)\times 1,\, t]}{B_{qa}} \Big) \Big(\underset{[(p-1)\times(p-2),\, t]}{B_{qm}} \Big) \Big(\underset{[(p-1)\times 1,\, t]}{B_{qz}}$$

§ 23 $$b_{1q} = B_{qm} \quad u_A$$

§ 24 Stop

II-c Flexibilities

For the foundations of our present exposé of flexibility matrices the reader is referred to Argyris (Ref. (1) and (2).

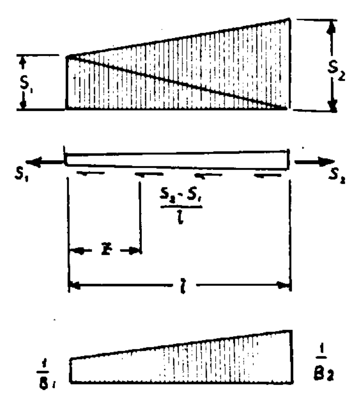
II-c 1 Flange Flexibilities

In order to allow for a reasonably wide variation of flange flexibility whilst avoiding cumbersome mathematical expressions Ref. (2) suggests the application of a linear longitudinal variation to the flexibility itself instead of the area. Considering Fig. II,1 we may then write for the flexibility per unit length at the station z

$$\phi = \frac{1}{E} \left[\frac{1}{B_1} + \left(\frac{1}{B_2} - \frac{1}{B_1} \right) \frac{z}{l} \right]$$

where $\frac{1}{B_1}$ and $\frac{1}{B_2}$ are the values at the two end stations.

Fig. II,1
Loading and flexibility distributions in a flange element



Carrying out the integrating procedure of Eqn. ((III,3))
we obtain for a single flange element;

$$f_l = \frac{l}{6 E B_1 B_2} \begin{bmatrix} B_1 + 3 B_2 & B_1 + B_2 \\ B_1 + B_2 & 3B_1 + B_2 \end{bmatrix} \qquad (II,12)$$

For a uniform flange $B_1 = B_2 = B$ and expression (II,43) reduces to the standard form

$$f_l = \frac{l}{6EB} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad \text{(II,43a)}$$

In order to set up the complete flexibility super-matrix for the whole fuselage, we use the following programme

a) We first form the operator

$$\underset{[2(p-1),()]}{\mathcal{E}_{fl}} = \sum_{i=1}^{(p-1)} \underset{[2(p-1)\times 1,()]_{(2i-1)}}{e} \; \underset{[2(p-1)\times 1,()]_{2i}}{e^{t}} + \sum_{i=1}^{(p-1)} \underset{[2(p-1)\times 1,()]_{2i}}{e} \; \underset{[2(p-1)\times 1,()]_{(2i-1)}}{e^{t}} \qquad \text{(II,44)}$$

b) Applying this operator we find

$$\underset{[2(p-1)\times 1, t\times 1]}{\overline{B}} = \underset{[2(p-1),()]}{\mathcal{E}_{fl}} \; \underset{[2(p-1)\times 1, t\times 1]}{B} \qquad \text{(II,45)}$$

c) Rearranging the two columns B and $\overline{B}$ in a diagonal form we have

$$\underset{[2(p-1)/, t/]}{B^{d}} \quad \text{and} \quad \underset{[2(p-1)/, t/]}{\overline{B}^{d}}$$

d) We now construct the two operators

$$\underset{[2(p-1)\times(p-1),(1)]}{\bar{\mathcal{E}}_{\ell+}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,(1)]_{(2i-1)}}{\mathbf{e}} \quad \underset{[(p-1)\times 1,(1)]_{i}}{\mathbf{e}}^{t} \qquad \text{(II,46)}$$

and

$$\underset{[2(p-1)\times(p-1),(1)]}{\bar{\mathcal{E}}_{\ell-}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,(1)]_{2i}}{\mathbf{e}} \quad \underset{[(p-1)\times 1,(1)]_{i}}{\mathbf{e}}^{t} \qquad \text{(II,46a)}$$

as well as their sum

$$\underset{[2(p-1)\times(p-1),(1)]}{\bar{\mathcal{E}}_{\ell}} = \bar{\mathcal{E}}_{\ell+} + \bar{\mathcal{E}}_{\ell-} \qquad \text{(II,46b)}$$

e) We form the extended matrix $\mathbf{d}_e$ of the lengths of the bays. Although the matrix is diagonal, we may equally consider it as a general square matrix since the super-matrix scheme does not operate on the zero sub-matrices.

$$\underset{[2(p-1),(1)]}{\mathbf{d}_e} = \bar{\mathcal{E}}_{\ell} \quad \mathbf{d} \quad \bar{\mathcal{E}}_{\ell}^{\,t} \qquad \text{(II,47)}$$

f) We can now calculate the first part of the flange flexibility matrix according to the simple relation

$$\underset{[2(p-1)/,t]}{\mathbf{f}_{\ell(a)}} = \left(\frac{1}{12}\right)\left(\frac{1}{E}\right) \mathbf{d}_e \quad \left[\mathbf{B}^{\alpha}\ \bar{\mathbf{B}}^{\alpha}\right]^{-1} \left[3\bar{\mathbf{B}}^{\alpha} + \mathbf{B}^{\bar{\alpha}}\right] \qquad \text{(II,48)}$$

g) We next set up the two column matrices

$$\underset{[(p-1)\times 1,\, t\times 1]}{B_+} = \bar{\varepsilon}^{\,t}_{\ell_+} \; B \tag{II,49}$$

and

$$\underset{[(p-1)\times 1,\, t\times 1]}{B_-} = \bar{\varepsilon}^{\,t}_{\ell_-} \; B \tag{II,49a}$$

h) Rearranging Eqs. (II,49,49a) in a diagonal we have

B_+^d and B_-^d each of order $[(p-1)/ \;,\; t/]$

i) We construct the matrix

$$\underset{[(p-1)\times 1,\, t\times 1]}{f^{*}_{\ell(b)v}} = \left(\tfrac{1}{12}\right)\left(\tfrac{1}{E}\right) d \; \left[B_+^d \;\; B_-^d\right]^{-1} \left[B_+ + B_-\right] \tag{II,50}$$

j) Expanding the last matrix we find

$$\underset{[2(p-1)\times 1,\, t\times 1]}{f_{\ell(b)v}} = \bar{\varepsilon}_{\ell_+} \; f^{*}_{\ell(b)v} \tag{II,51}$$

k) We read (II,51) as a diagonal matrix

$$\underset{[2(p-1)/ \,,\, t/]}{f_{\ell(b)}}$$

1) Using the operator of Eqn. (I,6) we now form the total flexibility matrix

$$\underset{[2(p-1),t/]}{f_{\ell}} = f_{\ell(a)} + \underset{[2(p-1),()]}{I_a} \; f_{\ell(b)} + f_{\ell(b)} \; \underset{[2(p-1),()]}{I_a^{t}} \qquad (II,52)$$

We give as usual the full programme

FORMATION of f_{ℓ}

§ 1 $$\underset{[2(p-1),()]}{\varepsilon_{fl}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,()]_{(2i-1)}}{e} \; \underset{[2(p-1)\times 1,()]_{2i}}{e^{t}} + \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,()]_{2i}}{e} \; \underset{[2(p-1)\times 1,()]_{(2i-1)}}{e^{t}}$$

§ 2 $$\underset{[2(p-1)\times 1,t\times 1]}{\bar{B}} = \varepsilon_{fl} \quad B$$

§ 3 Rearrange B and $\bar{B}$ as diagonals to give

$$\underset{[2(p-1)/,t/]}{B^{d}}$$

and

§ 4 $$\underset{[2(p-1)/,t/]}{\bar{B}^{d}}$$

§ 5 $$\underset{[2(p-1)\times(p-1),(1)]}{\bar{\varepsilon}_{\ell+}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,(1)]_{(2i-1)}}{e} \; \underset{[(p-1)\times 1,(1)]_{i}}{e^{t}}$$

§ 6 $$\underset{[2(p-1)\times(p-1),(1)]}{\bar{\varepsilon}_{\ell-}} = \sum_{i=1}^{p-1} \underset{[2(p-1)\times 1,(1)]_{2i}}{e} \; \underset{[(p-1)\times 1,(1)]_{i}}{e^{t}}$$

§ 7 $$\bar{\varepsilon}_{\ell} = \bar{\varepsilon}_{\ell+} + \bar{\varepsilon}_{\ell-}$$

§ 8 $$\underset{[2(p-1),(1)]}{d_e} = \bar{\varepsilon}_{\ell} \; d \; \bar{\varepsilon}_{\ell}^{t}$$

§ 9 $$\underset{[2(p-1)/,t/]}{f_{\ell(a)}} = \left(\frac{1}{12}\right)\left(\frac{1}{E}\right) d_e \left[B^{d} \bar{B}^{d}\right]^{-1} \left[3\bar{B}^{d} + B^{d}\right]$$

§ 10 $$\underset{[(p-1)\times 1,\, t\times 1]}{B_{+}} = \bar{\varepsilon}_{\ell+}^{t} \; B$$

and

§ 11 $$\underset{[(p-1)\times 1,\, t\times 1]}{B_{-}} = \bar{\varepsilon}_{\ell-}^{t} \; B$$

§ 12 Rearrange B_+ as diagonal $\underset{[(p-1)/,t/]}{B_+^d}$

§ 13 Rearrange B_- as diagonal $\underset{[(p-1)/,t/]}{B_-^d}$

§ 14 $$\underset{[(p-1)\times 1,t\times 1]}{f^*_{l(b)v}} = \left(\tfrac{1}{12}\right)\left(\tfrac{1}{E}\right) d \left[B_+^d \; B_-^d\right]^{-1} \left[B_+ + B_-\right]$$

§ 15 $$\underset{[2(p-1)\times 1,t\times 1]}{f_{l(b)v}} = \bar{\mathcal{E}}_{l+} \; f^*_{l(b)v}$$

§ 16 Rearrange $f_{l(b)v}$ as diagonal $\underset{[2(p-1)/,t/]}{f_{l(b)}}$

§ 17 $$\underset{[2(p-1),t/]}{f_l} = f_{l(a)} + \underset{[2(p-1),(1)]}{I_a} f_{l(b)} + f_{l(b)} \underset{[2(p-1),(1)]}{I_a^t}$$

§ 18 Stop

II-c 2. Shear Panel Flexibility

Since the stress distribution in the panels is defined by field forces rather than stress flows or actual stresses, the flexibility is calculated in accordance with Eqn. ((III,1)).

The expansion of this expression to cover the whole of the fuselage proceeds as follows

a) Using the operators $\mathbf{L}_+$ and $\mathbf{L}_-$ of section (II-a2) we form

$$\underset{[(p-1)\times 1,\, t\times 1]}{\mathbf{l}_m} = \left(\tfrac{1}{2}\right) \underset{[(p-1)\times p,\, (1)]}{\left[\mathbf{L}_+ + \mathbf{L}_-\right]} \quad \underset{[p\times 1,\, t\times 1]}{\mathbf{l}} \qquad (II,53)$$

b) Rearranging the thickness matrix $\mathbf{t}$ of Eqn. (I,18) in a diagonal form we have

$$\underset{[(p-1)/\,,\, t/]}{\mathbf{t}^{d}}$$

c) We now derive the flexibility as

$$\mathbf{f}_{qv} = \left(\tfrac{1}{G}\right)\left[\mathbf{t}^{d}\right]^{-1} \mathbf{d}^{-1} \quad \mathbf{l}_m \qquad (II,54)$$

and rearrange it in diagonal form to read

$$\underset{[(p-1)/\,,\, t/]}{\mathbf{f}_q}$$

The programme follows as

FORMATION of $\mathbf{f}_q$

§ 1 $$\underset{[(p-1)\times 1,\, t\times 1]}{\mathbf{l}_m} = \left(\tfrac{1}{2}\right)\left[\mathbf{L}_+ + \mathbf{L}_-\right] \quad \mathbf{l}$$

§ 2 Rearrange $\mathbf{t}$ as diagonal $\underset{[(p-1)/\,,\, t/]}{\mathbf{t}^{d}}$

§ 3 $$\underset{[(p-1)\times 1,\, t\times 1]}{\mathbf{f}_{qv}} = \left(\tfrac{1}{G}\right)\left[\mathbf{t}^{d}\right]^{-1} \mathbf{d}^{-1} \quad \mathbf{l}_m$$

§ 4 Rearrange f_{qv} as diagonal $f_{q\,[(p-1)/,\,t/]}$

§ 5 Stop

If it proves advantageous to scale down the fundamental matrices of our programme e.g D , in order to reduce the danger of overflow and underflow, then E should be omitted in all the flexibility programmes, and G replaced by G/E . Only when calculating deflections it is subsequently necessary to multiply the results by E .

II-c 3. Ring Flexibilities

Following Ref. (2) the flexibility of the rings is formed as two super-matrices. The first one accounts for the deformability due to the normal forces N and the second f_{M+S} corresponds to the combined effects of bending moments M and shear forces S in the rings. (see Eqn. ((III,36))). The corresponding force pattern is completely described by $b_{N_i\,(t\,x_i)}$ and $b_{M_i\,(t\,x_i)}$ since the shear forces are taken in this procedure to be solely determined by the bending moments. A criticism of this technique is given in Ref.(2) where it is suggested that for large computers it may be computationally advantageous not to merge f_M and f_S . However, in our current work we restrict ourselves to the case of f_N and the combined f_{M+S} .

When the transverse stiffening of the fuselage is achieved with diaphragms instead of rings the definition of stresses and flexibility is best carried out in a completely different manner. This technique is described in Chapter IV.

The programme detailed below is limited to the case of single-cell fuselages with singly connected rings. The flexibility of the ring element may vary linearly between adjacent nodal points.

In order to set up $\mathbf{f}_N$, the first component super-matrix of the flexibility, we proceed as follows :-

a) We determine the super vector of the mean areas from

$$\underset{[p\times 1,\, t\times 1]}{\mathbf{A}_m} = \left(\tfrac{1}{2}\right)\left[\underset{[p\times 1,\, t\times 1]}{\mathbf{A}_+} + \underset{[\cdot,\, t]}{\mathbf{I}_a}\ \underset{[p\times 1,\, t\times 1]}{\mathbf{A}_-}\right] \qquad \text{(II,55)}$$

b) We then rearrange the matrices $\mathbf{A}_m$ and λ (see Eqn.(I,17a)), in diagonal form to yield

$$\underset{[p/,\, t/]}{\mathbf{A}_m^d}$$

and

$$\underset{[p/,\, t/]}{\lambda^d}$$

c) We hence obtain the final normal force flexibility from

$$\underset{[p/,\, t/]}{\mathbf{f}_N} = \left(\tfrac{1}{E}\right)\left[\mathbf{A}_m^d\right]^{-1}\ \lambda^d \qquad \text{(II,56)}$$

We next derive the combined flexibility for bending and shear

d) We form the three matrices

$$\underset{[p\times 1,\, t\times 1]}{\mathcal{J}_2} = \underset{[\cdot,\, t]}{\mathbf{I}_a}\ \mathcal{J}_- \qquad \text{(II,57)}$$

and

$$\underset{[p\times 1,\, t\times 1]}{\mathcal{J}_t} = \underset{[\cdot,\, t]}{\mathbf{I}_a^t}\ \mathcal{J}_+ \qquad \text{(II,57a)}$$

and

$$\underset{[p\times 1,\, t\times 1]}{\lambda_t} = \underset{[1,\, t]}{I_a^t} \quad \lambda \qquad \text{(II,58)}$$

e) We rearrange $\mathcal{J}_+ , \mathcal{J}_- , \mathcal{J}_2 , \mathcal{J}_t , \lambda$ and λ_t in a diagonal form to give

$$\underset{[p/,\, t/]}{\mathcal{J}_+^d} \qquad \underset{[p/,\, t/]}{\mathcal{J}_-^d}$$

$$\underset{[p/,\, t/]}{\mathcal{J}_2^d} \qquad \underset{[p/,\, t/]}{\mathcal{J}_t^d}$$

and

$$\underset{[p/,\, t/]}{\lambda^d} \qquad \underset{[p/,\, t/]}{\lambda_t^d}$$

f) The bending flexibility f_M is then determined from

$$\underset{[p/,\, t]}{f_M} = \left(\tfrac{1}{12}\right)\left(\tfrac{1}{E}\right)\Big\{ \left[\mathcal{J}_+^d \mathcal{J}_2^d\right]^{-1}\left[\mathcal{J}_+^d + 3\mathcal{J}_2^d\right]\lambda^d + \left[\mathcal{J}_-^d \mathcal{J}_t^d\right]^{-1}\left[\mathcal{J}_-^d + 3\,\mathcal{J}_t^d\right]\lambda_t^d$$
$$+ \left[\mathcal{J}_+^d \mathcal{J}_2^d\right]^{-1}\left[\mathcal{J}_+^d + \mathcal{J}_2^d\right]\lambda^d \underset{[1,\, t]}{I_a} + \underset{[1,\, t]}{I_a^t}\left[\mathcal{J}_+^d \mathcal{J}_2^d\right]^{-1}\left[\mathcal{J}_+^d + \mathcal{J}_2^d\right]\lambda^d \Big\} \qquad \text{(II,59)}$$

To calculate f_s, the contribution of the shear, we first form the two matrices

$$\underset{[p\times 1,\, t\times 1]}{C_{m2}} = \left(\tfrac{1}{2}\right)\left[C_+ + \underset{[1,\, t]}{I_a}\; C_-\right] \qquad \text{(II,60)}$$

and

$$\underset{[p\times 1,\, t\times 1]}{C_{mt}} = \left(\tfrac{1}{2}\right)\left[C_- + \underset{[1,\, t]}{I_a^t}\; C_+\right] \qquad \text{(II,60a)}$$

which are then read as the diagonal matrices,

$$\underset{[p/\,,\,t/]}{C^{d}_{m2}}$$

and

$$\underset{[p/\,,\,t/]}{C^{d}_{mt}}$$

g) The shear flexibility based on the bending moment vector b_M is now

$$\underset{[p/\,,\,t]}{f_S} = \left(\tfrac{1}{G}\right)\left\{\left[C^{d}_{m2}\right]^{-1}\left[\lambda^{d}\right]^{-1} + \left[C^{d}_{mt}\right]^{-1}\left[\lambda^{d}_{t}\right]^{-1} - \left[C^{d}_{m2}\right]^{-1}\left[\lambda^{d}\right]^{-1}\underset{[(),\,t]}{I_a} - \underset{[(),\,t]}{I^{t}_{a}}\left[\lambda^{d}\right]^{-1}\left[C^{d}_{m2}\right]^{-1}\right\} \qquad (II,61)$$

h) The total bending- cum - shear flexibility super-matrix is thus

$$\underset{[p/\,,\,t]}{f_{M+S}} = f_M + f_S \qquad (II,62)$$

i) We join the two parts of the flexibility diagonally to obtain the total ring flexibility.

$$\underset{[2p/\,,\,t]}{f_r} = \begin{bmatrix} f_N & \\ & f_{M+S} \end{bmatrix} \qquad (II,63)$$

We rewrite the programme in the usual form

FORMATION of f_r

§ 1 $$\underset{[p\times 1,\, t\times 1]}{A_m} = \left(\tfrac{1}{2}\right)\left[A_+ + \underset{[(),\, t]}{I_a} \; A_- \right]$$

§ 2 Rearrange A_m as diagonal $\underset{[p/,\, t/]}{A_m^{\,d}}$

§ 3 Rearrange λ as diagonal $\underset{[p/,\, t/]}{\lambda^{d}}$

§ 4 $$\underset{[p/,\, t/]}{f_N} = \left(\tfrac{1}{E}\right)\left[A_m^d\right]^{-1} \lambda^d$$

§ 5 $$\underset{[p\times 1,\, t\times 1]}{\mathfrak{J}_2} = \underset{[(),\, t]}{I_a} \; \mathfrak{J}_-$$

§ 6 $$\underset{[p\times 1,\, t\times 1]}{\mathfrak{J}_t} = \underset{[(),\, t]}{I_a^t} \; \mathfrak{J}_+$$

§ 7 $$\underset{[p\times 1,\, t\times 1]}{\lambda_t} = \underset{[(),\, t]}{I_a^{\,t}} \; \lambda$$

§ 8 Rearrange $\mathcal{J}_+$ as diagonal $\underset{[p/,t/]}{\mathcal{J}_+^d}$

§ 9 Rearrange $\mathcal{J}_-$ as diagonal $\underset{[p/,t/]}{\mathcal{J}_-^d}$

§ 10 Rearrange $\mathcal{J}_2$ as diagonal $\underset{[p/,t/]}{\mathcal{J}_2^d}$

§ 11 Rearrange $\mathcal{J}_t$ as diagonal $\underset{[p/,t/]}{\mathcal{J}_t^d}$

§ 12 Rearrange λ_t as diagonal $\underset{[p/,t/]}{\lambda_t^d}$

§ 13

$$\underset{[p/,t]}{f_M} = \left(\tfrac{1}{12}\right)\left(\tfrac{1}{E}\right)\Big\{\left[\mathcal{J}_+^d \mathcal{J}_2^d\right]^{-1}\left[\mathcal{J}_+^d + 3\,\mathcal{J}_2^d\right]\lambda^d + \left[\mathcal{J}_-^d \mathcal{J}_t^d\right]\ \left[\mathcal{J}_-^d + 3\,\mathcal{J}_t^d\right]\lambda_t^d$$
$$+ \left[\mathcal{J}_+^d \mathcal{J}_2^d\right]\ \left[\mathcal{J}_+^d + \mathcal{J}_2^d\right]\lambda^d \underset{[0,t]}{I_a} + \underset{[0,t]}{I_a^t}\left[\mathcal{J}_+^d \mathcal{J}_2^d\right]^{-1}\left[\mathcal{J}_+^d + \mathcal{J}_2^d\right]\lambda^d\Big\}$$

§ 14

$$\underset{[p\times 1,\,t\times 1]}{C_{m2}} = \left(\tfrac{1}{2}\right)\Big[\ C_+ \ + \underset{[0,t]}{I_a}\ C_-\Big]$$

§ 15

$$\underset{[p\times 1,\,t\times 1]}{C_{mt}} = \left(\tfrac{1}{2}\right)\Big[\ C_- \ + \underset{[0,t]}{I_a^t}\ C_+\Big]$$

§ 16 Rearrange C_{m2} as diagonal $\underset{[p/,t/]}{C_{m2}^d}$

§ 17 Rearrange C_{mt} as diagonal $\underset{[p/,t/]}{C_{mt}^d}$

§ 18

$$\underset{[p/,t]}{f_s} = \left(\frac{1}{G}\right)\left\{ \left[C^d_{m2}\right]^{-1}\left[\lambda^d\right]^{-1} + \left[C^d_{mt}\right]^{-1}\left[\lambda^d_t\right]^{-1} - \left[C^d_{m2}\right]^{-1}\left[\lambda^d\right]^{-1} \underset{[(),t]}{I_a} - \underset{[(),t]}{I^t_a}\left[\lambda^d\right]^{-1}\left[C^d_{m2}\right]^{-1} \right\}$$

§ 19 $$\underset{[p/,t]}{f_{M+S}} = f_M + f_S$$

§ 20 $$\underset{[2p/,t]}{f_r} = \left\lceil \underset{[p/,t/]}{f_N} \qquad \underset{[p/,t]}{f_{M+S}} \right\rfloor$$

§ 21 Stop

II-d Self-Equilibrating Stress Systems in the Rings

We list below the steps involved in setting up the self-equilibrating stress systems in the rings due to the primary redundancies in the cover . As a preparatory step we have to derive some information necessary for the closure of the rings. (see also Eqs. ((V, 43 to V,62))

a) We form the zero super-column $\underset{[p\times 1,\ t\times 1]}{0}$

b) We set S_{α} of Eqn. (I,22), negative

$$\underset{[p\times 1,\ t\times 1]}{S_{\alpha n}} = (-1)\ S_{\alpha} \qquad (II,64)$$

c) We construct by joining

$$\underset{[p\times 1,\ t\times 3]}{N_{2rv}} = \left[\underset{[p\times 1,\ t\times 1]}{0}\ \underset{[p\times 1,\ t\times 1]}{S_{\alpha n}}\ \underset{[p\times 1,\ t\times 1]}{C_{\alpha}}\right] \qquad (II,65)$$

d) and similarly

$$\underset{[p\times 1,\ t\times 3]}{M_{2rv}} = \left[\underset{[p\times 1,\ t\times 1]}{e}\ \underset{[p\times 1,\ t\times 1]}{\xi}\ \underset{[p\times 1,\ t\times 1]}{\eta}\right] \qquad (II,66)$$

e) We rearrange N_{2rv} as a diagonal matrix to give

$$\underset{[p/\ ,\ t\times 3]}{N_{2r}}$$

f) and similarly

$$\underset{[p/\ ,\ t\times 3]}{M_{2r}}$$

g) We now join the two diagonal matrices to form

$$\underset{[2p\times p\,,\,t\times 3]}{b_{2r}} = \begin{bmatrix} \underset{[p/\,,\,t\times 3]}{N_{2r}} \\ \underset{[p/\,,\,t\times 3]}{M_{2r}} \end{bmatrix} \qquad \text{(II,67)}$$

and thus we have our final matrix for the self-equilibrating stress systems due to the secondary redundancies.

h) We next obtain by a very simple operation

$$\underset{[p,\,3]}{D_{2r}} = b_{2r}^{t} \quad f_r \quad b_{2r} \qquad \text{(II,68)}$$

This matrix is obviously a diagonal one. However, since we assume that the super-matrix code does not form or store zero sub-matrices, and does not perform any actual computations involving such sub-matrices, we may equally consider D_{2r} as a full matrix. The loss of time involved in this procedure is insignificant.

i) We form the matrix $\underset{[(),\,t]}{\sum}$ described in Appendix (D).

j) Using the matrices X_Δ and y_Δ of Eqs. (II,10, 10a) we set up the diagonal matrices

$$\underset{[p/\,,\,t/]}{X_\Delta^{d}} \quad \text{and} \quad \underset{[p/\,,\,t/]}{y_\Delta^{d}}$$

k) We next assemble

$$\underset{[(p+2)\times p\,,\,t]}{\overline{X}_{\Delta 1}} = \begin{bmatrix} \underset{[p/\,,\,t/]}{X_\Delta^{d}} \\ \underset{[2\times p\,,\,t]}{0} \end{bmatrix} \qquad \text{(II,69)}$$

l) Also,

$$\underset{[(p+2)\times(p+1),t]}{\overline{X}_{\Delta 1}^{*}} = \left[\underset{[(p+2)\times p, t]}{\overline{X}_{\Delta 1}} \quad \underset{[(p+2)\times 1, t]}{0} \right] \tag{II,70}$$

m) and similarly,

$$\underset{[(p+2)\times p, t']}{\overline{X}_{\Delta 2}} = \begin{bmatrix} \underset{[2\times p, t']}{0} \\ \\ \underset{[p', t']}{X_{\Delta}^{d}} \end{bmatrix} \tag{II,71}$$

n) Then we form

$$\underset{[(p+2)\times(p+1),t']}{\overline{X}_{\Delta 2}^{*}} = \left[\underset{[(p+2)\times 1, t']}{0} \quad \underset{[(p+2)\times p, t']}{\overline{X}_{\Delta 2}} \right] \tag{II,72}$$

o) By subtraction we obtain

$$\underset{[(p+2)\times(p+1),t']}{\overline{X}_{\Delta}^{*}} = \overline{X}_{\Delta 1}^{*} - \overline{X}_{\Delta 2}^{*} \tag{II,73}$$

p) Splitting the matrix of Eqn. (II,73) horizontally

$$\underset{[(p+2)\times(p+1),t']}{\overline{X}_{\Delta}^{*}} = \begin{array}{c} \overline{X}_{\Delta\alpha}^{*} \\ {\scriptstyle [1\times(p+1), t']} \\ \hline \overline{X}_{\Delta} \\ {\scriptstyle [p\times(p+1), t']} \\ \hline \overline{X}_{\Delta\beta} \\ {\scriptstyle [1\times(p+1), t']} \end{array} \tag{II,74}$$

q) We next split the matrix $\bar{X}_{\Delta}$ vertically

$$\underset{[p\times(p+1),\, t/]}{\bar{X}_{\Delta}} = \underset{[p\times 1,\, t/]}{\bar{X}_{\Delta\Gamma a}} \Bigg)\Bigg(\underset{[p\times(p-1),\, t/]}{\bar{X}_{\Delta M}} \Bigg)\Bigg(\underset{[p\times 1,\, t/]}{\bar{X}_{\Delta\Gamma z}} \qquad \text{(II,75)}$$

r) Using $\bar{X}_{\Delta M}$ we determine the transformation matrix

$$\underset{[p\times(p-1),\, t/]}{T_{G}} = \underset{[p\times(p-1),\, t/]}{\bar{X}_{\Delta M}} \; \underset{[(p-1)/,\, ()]}{d} \qquad \text{(II,76)}$$

s) Using y_{Δ} instead of X_{Δ} we form the analogous matrix

$$\underset{[p\times(p-1),\, t/]}{T_{J}} = \underset{[p\times(p-1),\, t/]}{\bar{y}_{\Delta M}} \; \underset{[(p-1)/,\, ()]}{d} \qquad \text{(II,76a)}$$

t) We next find the components of the forces due to the field forces

$$\underset{[p\times(p-2),\, t\times(t-3)]}{G_1} = \underset{[p\times(p-1),\, t/]}{T_{G}} \; \underset{[(p-1)\times(p-2),\, t\times(t-3)]}{b_{1q}} \qquad \text{(II,77)}$$

and

$$\underset{[p\times(p-2),\, t\times(t-3)]}{J_1} = \underset{[p\times(p-1),\, t/]}{T_{J}} \; \underset{[(p-1)\times(p-2),\, t\times(t-3)]}{b_{1q}} \qquad \text{(II,77a)}$$

u) The next step is to construct the super vector

$$\underset{[p\times 1,\, t\times 1]}{x_1} = \Big[\underset{[p,\, ()]}{I_a} \;\; - \underset{[p,\, ()]}{I} \Big] \; X \qquad \text{(II,78)}$$

v) We then assemble by joining

$$\underset{[p/\ ,1/]}{d^{*}} = \left[\underset{[(p-1)/,\ 1/]}{d} \qquad \underset{[1/,1/]}{(+1)} \right] \qquad (II,79)$$

w) Hence we now form

$$d^{*\,-1} \quad X, \quad = \underset{[p\times 1\ ,\ t\times 1]}{V_{x+}} \qquad (II,80)$$

and

$$\underset{[p,\ 1/]}{I_{a}^{t}} \quad V_{x+} \quad = \underset{[p\times 1\ ,\ t\times 1]}{V_{x-}} \qquad (II,81)$$

x) Putting the two super-vectors in diagonal form (V_{x+}^{d} and V_{x-}^{d}) and using the operators developed in Section II-a we find the transformation matrix

$$\underset{[p\times 2(p-1),\ t/]}{T_{px}} = \varepsilon_{\ell+}^{t} \quad V_{x+}^{d} \quad - \quad \varepsilon_{\ell-}^{t} \quad V_{x-}^{d} \qquad (II,82)$$

Similarly introducing y instead of X we obtain the corresponding matrix for the y direction,

$$\underset{[p\times 2(p-1),t/]}{T_{py}} = \varepsilon_{\ell+}^{t} \quad V_{y+}^{d} \quad - \quad \varepsilon_{\ell-}^{t} \quad V_{y-}^{d} \qquad (II,82a)$$

y) We are now in a position to determine the matrices K and L, which represent the nodal loads on the rings in the x and y directions (see Eqs. ((II,32,33)))

$$\underset{[p\times(p-2),\ t\times(t-3)]}{K_{1}} = \frac{1}{2} \underset{[1/,t]}{\beta} \underset{[p\times(p-2),t\times(t-3)]}{G_{1}} + \underset{[p\times 2(p-1),\ t/]}{T_{px}} \underset{[2(p-1)\times(p-2),t\times(t-3)]}{b_{1\ell}}$$

and

(II,83)

$$\underset{[p\times(p-2),\,t\times(t-3)]}{L_1} = \left(\tfrac{1}{2}\right) \underset{[(),t]}{\beta} \; J_1 + T_{py} \; b_{1\ell} \qquad (II,83a)$$

z) Applying a straightforward scalar multiplication we find

$$\underset{[p\times(p-2),\,t\times(t-3)]}{U_1} = \underset{[(),t]}{\Sigma} \; K_1 \qquad (II,84)$$

and

$$\underset{[p\times(p-2),\,t\times(t-3)]}{V_1} = \underset{[(),t]}{\Sigma} \; L_1 \qquad (II,84a)$$

aa) Rearranging S_γ and C_γ into the diagonal forms S_γ^d and C_γ^d we obtain the normal force distribution in the cut rings due to the primary redundancies

$$\underset{[p\times(p-2),\,t\times(t-3)]}{N_{10}} = (-1)\left[C_\gamma^d \; U + S_\gamma^d \; V \right] \qquad (II,85)$$

ab) Using next a "scalar and diagonal super multiplication" we find

$$\underset{[p\times(p-2),\,t\times(t-3)]}{\overline{M}_{10}} = \left[\underset{[(),t]}{\Sigma} - \underset{[(),t]}{I} \right] \left[- y_\Delta^d \, U + x_\Delta^d \; V \right] \qquad (II,86)$$

ac) Rearranging χ and ψ in the diagonal forms

$$\chi^d \quad \text{and} \quad \psi^d$$

ad) we establish the moments on the open or cut rings due to the primary redundancies (see Eqs.((II,48)))

$$\underset{[p\times(p-2),\,t\times(t-3)]}{M_{10}} = \overline{M}_{10} - \left(\tfrac{1}{2}\right)\chi^d \left[\underset{[(),t]}{\beta} \, V - \left(\tfrac{1}{2}\right) \underset{[(),t]}{\alpha} \; J \right] + \left(\tfrac{1}{2}\right)\psi^d \left[\underset{[(),t]}{\beta} \, U - \left(\tfrac{1}{2}\right) \underset{[(),t]}{\alpha} \; G \right] \qquad (II,87)$$

ae) To determine the final stress distribution in the actual rings we apply the usual closure procedure (see Eqs.((V,43 to 62))). First we write,

$$\underset{[2p\times(p-2),\,t\times(t-3)]}{b_{10r}} = \begin{bmatrix} \underset{[p\times(p-2),\,t\times(t-3)]}{N_{10}} \\ \\ \underset{[p\times(p-2),\,t\times(t-3)]}{M_{10}} \end{bmatrix} \qquad \text{(II,88)}$$

af) We then form

$$\underset{[p\times(p-2),\,3\times(t-3)]}{D_{2ro}} = b_{2r}^{t} \quad f_r \quad b_{10r} \qquad \text{(II,89)}$$

and hence

$$\underset{[2p\times(p-2),\,t\times(t-3)]}{b_{1r}} = b_{10r} - b_{2r}\, D_{2r}^{-1}\, D_{2ro} \qquad \text{(II,90)}$$

Thus giving the final stress distribution in the rings.

F O R M A T I O N of b_{1r}

§ 1 Form $\underset{[p\times1,\,t\times1]}{0}$

§ 2 $\underset{[p\times1,\,t\times1]}{S_{\alpha n}} = (-1)\; S_{\alpha}$

§ 3 Form by joining $\underset{[p\times1,\,t\times3]}{N_{2rv}} = \Big[\underset{[p\times1,\,t\times1]}{0} \quad \underset{[p\times1,\,t\times1]}{S_{\alpha n}} \quad \underset{[p\times1,\,t\times1]}{C_{\alpha}}\Big]$

§ 4 Form by joining $\underset{[p\times 1,\ t\times 3]}{M_{2rv}} = \left[\underset{[p\times 1,\ t\times 1]}{e} \quad \underset{[p\times 1,\ t\times 1]}{\xi} \quad \underset{[p\times 1,\ t\times 1]}{\eta} \right]$

§ 5 Rearrange $\underset{[p\times 1,\ t\times 3]}{N_{2rv}}$ as diagonal $\underset{[p/,\ t\times 3]}{N_{2r}}$

§ 6 Rearrange $\underset{[p\times 1,\ t\times 3]}{M_{2rv}}$ as diagonal $\underset{[p/,\ t\times 3]}{M_{2r}}$

§ 7 Form by joining

$$\underset{[2p\times p,\ t\times 3]}{b_{2r}} = \begin{bmatrix} \underset{[p/,\ t\times 3]}{N_{2r}} \\ \underset{[p/,\ t\times 3]}{M_{2r}} \end{bmatrix}$$

§ 8 $\underset{[p\ ,\ 3]}{D_{2r}} = b_{2r}^{\ t} \quad f_r \quad b_{2r}$

§ 9 Form $\underset{[\cdot,\ t]}{\Sigma}$ (see Appendix D)

§ 1o Rearrange x_Δ as diagonal $\underset{[p/,\ t/]}{x_\Delta^d}$

§ 11 Rearrange y_Δ as diagonal $\underset{[p/,\ t/]}{y_\Delta^d}$

§ 12 Form by joining

$$\underset{[(p+2)\times p\,,\,t/]}{\overline{X}_{\Delta 1}} = \begin{bmatrix} \underset{[p/\,,\,t/]}{X_{\Delta}} \\ \underset{[2\times p\,,\,t/]}{0} \end{bmatrix}$$

§ 13 Form by joining

$$\underset{[(p+2)\times(p+1),t/]}{\overline{X}^{*}_{\Delta 1}} = \begin{bmatrix} \underset{[(p+2)\times 2,t/]}{\overline{X}_{\Delta 1}} & \underset{[(p+2)\times 1,t/]}{0} \end{bmatrix}$$

§ 14 Join

$$\underset{[(p+2)\times p,t/]}{\overline{X}_{\Delta 2}} = \begin{bmatrix} \underset{[2\times p\,,\,t/]}{0} \\ \underset{[p/\,,\,t/]}{X^{d}_{\Delta}} \end{bmatrix}$$

§ 15 Join

$$\underset{[(p+2)\times(p+1),t/]}{\overline{X}^{*}_{\Delta 2}} = \begin{bmatrix} \underset{[(p+2)\times 1,t/]}{0} & \underset{[(p+2)\times p,\,t/]}{\overline{X}_{\Delta 2}} \end{bmatrix}$$

§ 16 $$\underset{[(p+2)\times(p+1),t/]}{\overline{X}^{*}_{\Delta}} = \overline{X}^{*}_{\Delta 1} - \overline{X}^{*}_{\Delta 2}$$

§ 17 Split

$$\underset{[(p+2)\times(p+1),\,t/]}{\bar{X}^{*}_{\Delta}} = \begin{array}{c} \underset{[1\times(p+1)\;,\;t/]}{\bar{X}^{*}_{\Delta\alpha}} \\ \hline \hline \underset{[p\times(p+1)\,,\,t/]}{\bar{X}_{\Delta}} \\ \hline \hline \underset{[1\times(p+1)\;,\;t/]}{\bar{X}_{\Delta z}} \end{array}$$

§ 18 Split

$$\underset{[p\times(p+1),t/]}{\bar{X}_{\Delta}} = \underset{[p\times1,\;t/]}{\bar{X}_{\Delta r\alpha}} \;\Big)\Big(\; \underset{[p\times(p-1),\,t/]}{\bar{X}_{\Delta M}} \;\Big)\Big(\; \underset{[p\times1\,,\,t/]}{X_{\Delta r z}}$$

§ 19 $$\underset{[p\times(p-1)\,,\,t/]}{T_{G}} = \underset{[p\times(p-1)),\,t/]}{\bar{X}_{\Delta M}} \;\; \underset{[(p-1)/,1]}{d}$$

§ 2o Join

$$\underset{[(p+2)\times p,t/]}{\bar{y}_{\Delta 1}} = \begin{bmatrix} \underset{[p/\,,\,t/]}{y_{\Delta}} \\ \underset{[2\times p,\,t/]}{0} \end{bmatrix}$$

§ 21 Join

$$\underset{[(p+2)\times(p+1),t/]}{\bar{y}^{*}_{\Delta 1}} = \left[\underset{[(p+2)\times 2,t/]}{\bar{y}_{\Delta 1}} \quad \underset{[(p+2)\times 1,t/]}{0} \right]$$

§ 22 Join

$$\underset{[(p+2)\times p,t/]}{\bar{y}_{\Delta 2}} = \begin{bmatrix} \underset{[2\times p,\ t/]}{0} \\ \\ \underset{[p/,\ t/]}{y^{d}_{\Delta}} \end{bmatrix}$$

§ 23 Join

$$\underset{[(p+2)\times(p+1),t/]}{\bar{y}^{*}_{\Delta 2}} = \left[\underset{[(p+2)\times 1,t/]}{0} \quad \underset{[(p+2)\times p,t/]}{\bar{y}_{\Delta 2}} \right]$$

§ 24 $$\underset{[(p+2)\times(p+1),t/]}{\bar{y}^{*}_{\Delta}} = \bar{y}^{*}_{\Delta 1} - \bar{y}^{*}_{\Delta 2}$$

§ 25 Split

$$\underset{[(p+2)\times(p+1),t/]}{\bar{y}^{*}_{\Delta}} = \begin{array}{c} \underset{[1\times(p+1),\ t/]}{\bar{y}^{*}_{\Delta a}} \\ \hline \underset{[p\times(p+1),\ t/]}{\bar{y}_{\Delta}} \\ \hline \underset{[1\times(p+1),\ t/]}{\bar{y}_{\Delta z}} \end{array}$$

§ 26 Split

$$\underset{[p\times(p+1),\,t/]}{\bar{y}_{\Delta}} = \underset{[p\times 1,\,t/]}{\bar{y}_{\Delta ra}} \Big)\Big(\underset{[p\times(p-1),\,t/]}{\bar{y}_{\Delta M}} \Big)\Big(\underset{[p\times 1,\,t/]}{y_{\Delta rz}}$$

§ 27
$$\underset{[p\times(p-1),\,t/]}{T_J} = \underset{[p\times(p-1),\,t/]}{\bar{y}_{\Delta M}} \quad \underset{[(p-1)/,\,()]}{d}$$

§ 28
$$\underset{[p\times(p-2),\,t\times(t-3)]}{G_1} = T_G \quad b_{1q}$$

§ 29
$$\underset{[p\times(p-2),\,t\times(t-3)]}{J_1} = T_J \quad b_{1q}$$

§ 30
$$\underset{[p\times 1,\,t\times 1]}{x,} = \Big[\underset{[p,\,()]}{I_a} - \underset{[p,\,t]}{I} \Big] \; x$$

§ 31 Join
$$\underset{[p/,\,()]}{d^*} = \Big\lceil \underset{[(p-1)/,\,()]}{d} \quad \underset{[1/,\,()]}{(+1)} \Big\rfloor$$

§ 32
$$\underset{[p\times 1,\,t\times 1]}{v_{x+}} = d^{*\,-1} \; x,$$

§ 33 $\underset{[p\times 1,\ t\times 1]}{V_{x-}} = \underset{[p,()]}{I_a^t}\ V_{x+}$

§ 34 Rearrange V_{x+} as diagonal $\underset{[p/,\ t/]}{V_{x+}^d}$

§ 35 Rearrange V_{x-} as diagonal $\underset{[p/,\ t/]}{V_{x-}^d}$

§ 36 $\underset{[p\times 2(p-1),\ t/]}{T_{px}} = \left[\varepsilon_{\ell+}\ V_{x+}^d - \varepsilon_{\ell-}\ V_{x-}^d \right]^t$

§ 37 $\underset{[p\times 1,\ t\times 1]}{y,} = \left[\underset{[p,()]}{I_a} - \underset{[p,t]}{I} \right] y$

§ 38 $\underset{[p\times 1,\ t\times 1]}{V_{y+}} = d^{*-1}\ y,$

§ 39 $\underset{[p\times 1,\ t\times 1]}{V_{y-}} = \underset{[p,()]}{I_a^t}\ V_{y+}$

§ 40 Rearrange V_{y+} as diagonal $\underset{[p/,\ t/]}{V_{y+}^d}$

§ 41 Rearrange V_{y-} as diagonal $\underset{[p/,\ t/]}{V_{y-}^d}$

§ 42 $$\underset{[p\times 2(p-1),\ t/]}{T_{Py}} = \left[\varepsilon_{\ell_+} \ V_{Y_+}^{\alpha} - \varepsilon_{\ell_-} \ V_{Y_-}^{\alpha} \right]^t$$

§ 43 $$\underset{[p\times(p-2),\ t\times(t-3)]}{K_1} = \left(\tfrac{1}{2}\right) \underset{[1,\ t]}{\beta} \ G_1 + T_{Px} \ b_{1\ell}$$

§ 44 $$\underset{[p\times(p-2),\ t\times(t-3)]}{L_1} = \left(\tfrac{1}{2}\right) \underset{[1,\ t]}{\beta} \ J_1 + T_{Py} \ b_{1\ell}$$

§ 45 $$\underset{[p\times(p-2),\ t\times(t-3)]}{U_1} = \sum_{[1,\ t]} K_1$$

§ 46 $$\underset{[p\times(p-2),\ t\times(t-3)]}{V_1} = \sum_{[1,\ t]} L_1$$

§ 47 Rearrange S_{α} as diagonal $\underset{[p/,\ t/]}{S_{\alpha}^{d}}$

§ 48 Rearrange C_{α} as diagonal $\underset{[p/,\ t/]}{C_{\alpha}^{d}}$

§ 49 $$\underset{[p\times(p-2),\ t\times(t-3)]}{N_{10}} = (-1) \left[C_{\alpha}^{d} \ U_1 + S_{\alpha}^{d} \ V_1 \right]$$

§ 50 $$\underset{[p\times(p-2),\ t\times(t-3)]}{\overline{M}_{10}} = \left[\sum_{[1,\ t]} - \underset{[1,\ t]}{I} \right] \left[- y_{\Delta}^{\alpha} \ U_1 + x_{\Delta}^{\alpha} \ V_1 \right]$$

§ 51 Rearrange χ as diagonal $\underset{[p1,\ t1]}{\chi^{d}}$

§ 52 Rearrange ψ as diagonal $\underset{[p1,\ t1]}{\psi^{d}}$

§ 53

$$\underset{[p\times(p-2),\,t\times(t-3)]}{M_{10}} = \bar{M}_{10} - \left(\tfrac{1}{2}\right)\chi^{d}\left[\underset{[(),t]}{\beta} V_1 - \left(\tfrac{1}{2}\right)\underset{[(),t]}{\alpha} J_1\right] + \left(\tfrac{1}{2}\right)\psi^{d}\left[\underset{[(),t]}{\beta} U_1 - \left(\tfrac{1}{2}\right)\underset{[(),t]}{\alpha} G_1\right]$$

§ 54 Join

$$\underset{[2p\times(p-2),\,t\times(t-3)]}{b_{10r}} = \begin{bmatrix} \underset{[p\times(p-2),\,t\times(t-3)]}{N_{10}} \\ \underset{[p\times(p-2),\,t\times(t-3)]}{M} \end{bmatrix}$$

§ 55

$$\underset{[p\times(p-2),\,3\times(t-3)]}{D_{20r}} = b_{2r}^{t}\ f_r\ b_{10r}$$

§ 56

$$\underset{[2p\times(p-2),\,t\times(t-3)]}{b_{1r}} = b_{10r} - b_{2r}\ D_{2r}^{-1}\ D_{20r}$$

§ 57 Stop

II-e The Basic System in the Cover

A) For the basic theory we refer the reader to Eqs. ((IV,56 to 76)) If the fuselage is only loaded by transverse loads, the flange loads and field forces can be immediately derived from

$$\underset{[2(p-1)\times 1,\, t\times \rho]}{b_{o\ell}} = \underset{[2(p-1)/,\, t\times 3]}{\hat{b}_{o\ell}} \; \underset{[2(p-1)\times 1,\, 3\times \rho]}{\hat{R}_{\ell}} \qquad (II,91)$$

and

$$\underset{[(p-1)\times 1,\, t\times \rho]}{b_{oq}} = \underset{[(p-1)\times 2(p-1),\, t\times 3]}{\hat{b}_{oq}} \; \underset{[2(p-1)\times 1,\, 3\times \rho]}{\hat{R}_{q}} \qquad (II,91a)$$

B) If, on the other hand prescribed flange loads are applied at either or both ends, it is necessary (a) to allow for these distributions in the matrices for the basic flange loads and (b) to modify the field force distributions in the first and last bays in order to transfer from the prescribed flange loads at station 1 and p to the automatically calculated ones at station 2 and $(p-1)$. This is achieved by the use of, and explains the formation of,the extreme self-equilibrating stress systems that have been included in the automatic calculation of the self-equilibrating stress systems.

We denote in accordance with our standard procedure the matrix for the basic systems obtained in (II,91, 91a) by b_o^* , the self-equilibrating set of flange loads at the frame stations by

$$\underset{(t\times(t-3))}{b_{1\ell_1}}$$

and the associated self-equilibrating shear flow by $\underset{(t\times(t-3))}{b_{1q_{1,2}}}$

If, in order to transform the sub-matrix $b^*_{o\ell_1}$ into the prescribed P_{ℓ_1} matrix we add a set of redundancies $\overline{Y}_1$ at the first frame station this may be expressed by the condition

$$P_{\ell_1} = b^*_{o\ell_1} + b_{1\ell_1} \quad \bar{Y}_1 \qquad \text{(II,92)}$$

We now proceed essentially as in Ref. (2) using though, whenever possible, the super-matrices relating to the whole structure. Premultiplying by $A^t_{\ell_2}$ we obtain

$$A^t_{\ell_2} \left[P_{\ell_1} - b^*_{o\ell_1} \right] = \bar{Y}_1 \qquad \text{(II,93)}$$

which determines the values of the required self-equilibrating flange loads.

The associated field forces can be obtained from

$$b_{oq_{1,2}} = b^*_{oq_{1,2}} + b_{1q_{1,2}} \quad \bar{Y}_1 \qquad \text{(II,94)}$$

giving immediately the modified field forces.

Similarly, we may derive at the station p ,

$$\bar{Y}_p = A^t_{\ell(p-1)} \left[P_{\ell_p} - b^*_{o\ell_p} \right] \qquad \text{(II,93a)}$$

and in the bay $(p-1)\,,\,p$

$$b_{oq_{(p-1),p}} = b^*_{oq_{(p-1),p}} + b_{1q_{p,(p-1)}} \quad \bar{Y}_p \qquad \text{(II,94a)}$$

In order to obtain the sub-matrices $b^*_{o\ell_1}$, $b^*_{o\ell_p}$, $b^*_{oq_{1,2}}$ and $b_{oq_{(p-1),p}}$ as well as $b_{1q_{1,2}}$ and $b_{1q_{(p-1),p}}$, all required in the above procedure, we introduce the following operations

$$\mathbf{b}_{o\ell_1}^{\;*} = \underset{[\,2(p-1)\times 1\,,\,()\,]_1}{\mathbf{e}^{\,t}} \quad \underset{[\,2(p-1)\times 1\,,\,t\times\rho\,]}{\mathbf{b}_{o\ell}^{\;*}} \qquad \text{(II,95)}$$

$$\mathbf{b}_{o\ell_p}^{\;*} = \underset{[\,2(p-1)\times 1\,,\,()\,]_p}{\mathbf{e}^{\,t}} \quad \underset{[\,2(p-1)\times 1\,,\,t\times\rho\,]}{\mathbf{b}_{o\ell}^{\;*}} \qquad \text{(II,95a)}$$

$$\mathbf{b}_{oq_{1,2}}^{\;*} = \underset{[\,(p-1)\times 1\,,\,()\,]_1}{\mathbf{e}^{\,t}} \quad \underset{[\,(p-1)\times 1\,,\,t\times\rho\,]}{\mathbf{b}_{oq}^{\;*}} \qquad \text{(II,95b)}$$

$$\mathbf{b}_{oq_{(p-1),p}}^{\;*} = \underset{[\,(p-1)\times 1\,,\,()\,]_{(p-1)}}{\mathbf{e}^{\,t}} \quad \underset{[\,(p-1)\times 1\,,\,t\times\rho\,]}{\mathbf{b}_{oq}^{\;*}} \qquad \text{(II,95c)}$$

$$\mathbf{b}_{1q_{1,2}} = \underset{[\,(p-1)\times 1\,,\,()\,]_1}{\mathbf{e}^{\,t}} \quad \underset{[\,(p-1)\times p\,,\,t\,]}{\mathbf{B}_q} \quad \underset{[\,p\times 1\,,\,()\,]_1}{\mathbf{e}} \quad \underset{[\,()\,,\,t\times(t-3)\,]}{\mathbf{u}_A} \qquad \text{(II,95d)}$$

$$\mathbf{b}_{1q_{(p-1),p}} = \underset{[\,(p-1)\times 1\,,\,()\,]_{(p-1)}}{\mathbf{e}^{\,t}} \quad \underset{[\,(p-1)\times p\,,\,t\,]}{\mathbf{B}_q} \quad \underset{[\,p\times 1\,,\,()\,]_p}{\mathbf{e}} \quad \mathbf{u}_A \qquad \text{(II,95e)}$$

C) The most general case arises, however, when sets of prescribed longitudinal forces are applied at each frame station. The complete group of these prescribed forces in the direction z is described by the super-matrix $\mathbf{P}_\ell \; [\,p\times 1\,,\; t\times\rho\,]$. If we now consider the equilibrium of the flange loads at one particular frame station i ,

$$\mathbf{b}_{o\ell_{i-}} - \mathbf{b}_{o\ell_{i+}} = \mathbf{P}_{\ell_i} \qquad \text{(II,96)}$$

As in the case B, Eqn. (II,96) is, in general, not satisfied by the $\mathbf{b}_{o\ell}^{*}$ matrix. The difference may again be expressed by a self-equilibrating flange load system

$$\mathbf{b}_{o\ell_{i-}}^{*} - \mathbf{b}_{o\ell_{i+}}^{*} = \mathbf{P}_{\ell_i} + \mathbf{b}_{1\ell_i}\,\bar{\mathbf{Y}}_{i} \qquad \text{(II,97)}$$

Proceeding as before we find

$$\bar{\mathbf{Y}}_{i} = \mathbf{A}_{\ell_i}^{t}\left[\mathbf{b}_{o\ell_{i-}}^{*} - \mathbf{b}_{o\ell_{i+}}^{*} - \mathbf{P}_{\ell_i}\right] \qquad \text{(II,98)}$$

We may now modify $\mathbf{b}_{o\ell_{i+}}^{*}$ to read

$$\mathbf{b}_{o\ell_{i+}} = \mathbf{b}_{o\ell_{i+}}^{*} + \mathbf{b}_{1\ell_i}\,\mathbf{A}_{\ell_i}^{t}\left[\mathbf{b}_{o\ell_i}^{*} - \mathbf{b}_{o\ell_{i+}}^{*} - \mathbf{P}_{\ell_i}\right] \qquad \text{(II,99)}$$

We do not change on the other hand $\mathbf{b}_{o\ell_{i-}}$, except at the last station p when we write

$$\mathbf{b}_{o\ell_{p-}} = \mathbf{b}_{o\ell_{p-}}^{*} - \mathbf{b}_{1\ell p}\,\mathbf{A}_{\ell(p-1)}^{t}\left[\mathbf{b}_{o\ell_{p-}}^{*} - \mathbf{P}_{\ell_p}\right] \qquad \text{(II,100)}$$

In order to achieve this generalized correction we proceed as follows:-

a) We form the operator

$$\underset{[2(p-1)\times p,(\,)]}{\boldsymbol{\varepsilon}_{\ell_\Delta}} = \boldsymbol{\varepsilon}_{\ell_-} - \boldsymbol{\varepsilon}_{\ell_+} \qquad \text{(II,101)}$$

b) We then obtain

$$\underset{[p\times 1,\ (t-3)\times \rho]}{\bar{Y}} = \underset{[p/,\ (t-3)\times t]}{A_{\ell}^{t}} \left[\underset{[p\times 2(p-1),(1)]}{\varepsilon_{\ell_{\Delta}}^{t}} \underset{[2(p-1)\times 1,\ t\times \rho]}{b_{o\ell_{i}}^{*}} - \underset{[p\times 1,\ t\times \rho]}{P_{\ell}} \right] \quad (II,102)$$

c) We write down the modified (and final) super-matrix for the basic flange load distribution as

$$\underset{[2(p-1)\times 1,\ t\times \rho]}{b_{o\ell}} = \underset{[2(p-1)\times 1,t\times \rho]}{b_{o\ell}^{*}} + \left[\underset{[2(p-1)\times p,(1)]}{\varepsilon_{\ell+}} - \underset{[2(p-1)\times 1,(1)]}{e} \underset{[p\times 1,(1)]_{p}^{2(p-1)}}{e^{t}} \right] \underset{[p/,t\times(t-3)]}{b_{i\ell e}^{*}} \underset{[p\times 1,(t-3)\times \rho]}{\bar{Y}}$$

(II,103)

d) We now derive the corresponding field force modification through a similar procedure. Thus

$$\underset{[(p-1)\times 1,t\times \rho]}{b_{oq}} = \underset{[(p-1)\times 1,t\times \rho]}{b_{oq}^{*}} + \underset{[(p-1)\times p,t]}{B_{q-}} \underset{[(1),t\times(t-3)]}{u_{A}} \underset{[p\times 1,\ (t-3)\times \rho]}{\bar{Y}}$$

$$- \underset{[(p-1)\times p,t]}{B_{q+}} \underset{[(1),t\times(t-3)]}{u_{A}} \underset{[p\times 1,(1)]_{p}}{e} \underset{[p\times 1,(1)]_{p}}{e^{t}} \underset{[p\times 1,(t-3)\times \rho]}{\bar{Y}}$$

(II,104)

Since the super-matrix scheme is assumed not to carry out any operations with zero sub-matrices it is recommended that the most general case is programmed. Thus

MODIFICATIONS of BASIC SYSTEM in COVER

§ 1 $$\underset{[2(p-1)\times p,(1)]}{\varepsilon_{\ell_{\Delta}}} = \varepsilon_{\ell-} - \varepsilon_{\ell+}$$

§ 2 $$\underset{[p\times1,(t-3)\times\rho]}{\overline{Y}} = A_\ell^t \left[\varepsilon_{\ell_\Delta}^t b_{o\ell}^* - P_\ell \right]$$

§ 3 $$\underset{[2(p-1)\times1,t\times\rho]}{b_{o\ell}} = b_{o\ell}^* + \left[\varepsilon_{\ell_+} - \underset{2(p-1)}{\underset{[2(p-1)\times1,(1)]}{e}} \; \underset{[p\times1,(1)]_p}{e^t} \right] \overline{Y}$$

§ 4 $$\underset{[(p-1)\times1,t\times\rho]}{b_{oq}} = b_{oq}^* + \left[B_{q_-} \underset{[(1),t\times(t-1)]}{u_A} - B_{q_+} \underset{[(1),t\times(t-3)]}{u_A} \; \underset{[p\times1,(1)]_p}{e} \; \underset{[p\times1,(1)]_p}{e^t} \right] \overline{Y}$$

§ 5 Stop

II-f The Basic System in the Rings

Having obtained the basic system in the cover, we establish the corresponding stress distribution in the rings by a programme similar to that described under II-d. However, much of the work involved in establishing b_{1r} is of immediate application here and need not be repeated. We only summarize below the programme

F O R M A T I O N of b_{or}

§ 1 $$\underset{[p\times1,t\times\rho]}{G_o} = T_G \; b_{oq}$$

§ 2 $$\underset{[p\times1,t\times\rho]}{J_o} = T_J \; b_{oq}$$

§ 3 $$\underset{[p\times 1,\, t\times \rho]}{K_o} = \left(\tfrac{1}{2}\right) \underset{[(),\, t]}{\beta} \quad G_o + T_{Px} \quad b_{o\rho}$$

§ 4 $$\underset{[p\times 1,\, t\times \rho]}{L_o} = \left(\tfrac{1}{2}\right) \underset{[(),\, t]}{\beta} \quad J_o + T_{Py} \quad b_{o\rho}$$

§ 5 $$\underset{[p\times 1,\, t\times \rho]}{U_o} = \underset{[(),\, t]}{\Sigma} \quad K_o$$

§ 6 $$\underset{[p\times 1,\, t\times \rho]}{V_o} = \underset{[(),\, t]}{\Sigma} \quad L_o$$

§ 7 $$\underset{[p\times 1,\, t\times \rho]}{N_{oo}} = (-1) \left[C_{\alpha}^{d} \; U_o + S_{\alpha}^{d} \; V_o \right]$$

§ 8 $$\underset{[p\times 1,\, t\times \rho]}{\overline{M}_{oo}} = \left[\underset{[(),\, t]}{\Sigma} - \underset{[(),\, t]}{I} \right] \left[- y_{\Delta}^{d} \, U_o + x_{\Delta}^{d} \; V_o \right]$$

§ 9 $$\underset{[p\times 1,\, t\times \rho]}{M_{oo}} = \overline{M}_{oo} - \left(\tfrac{1}{2}\right) \chi^{d} \left[\underset{[(),\, t]}{\beta} \; V_o - \left(\tfrac{1}{2}\right) \underset{[(),\, t]}{\alpha} \; J_o \right] + \psi \left[\underset{[(),\, t]}{\beta} \; U_o - \left(\tfrac{1}{2}\right) \underset{[(),\, t]}{\alpha} \; G_o \right]$$

§ 1o Join $$\underset{[2p\times 1,\, t\times \rho]}{b_{oor}} = \begin{bmatrix} \underset{[p\times 1,\, t\times \rho]}{N_{oo}} \\ \\ \underset{[p\times 1,\, t\times \rho]}{M_{oo}} \end{bmatrix}$$

§ 11 $$\underset{[p\times 1,\ 3\times \rho]}{D_{200r}} = b_{2r}^{t} \quad f_r \quad b_{00r}$$

§ 12 $$\underset{[2p\times 1,\ t\times \rho]}{b_{0r}} = b_{00r} - b_{2r} \quad D_{2r}^{-1} \quad D_{200r}$$

§ 13 Stop

II-g The Final Solution of the Problem

Having all the necessary basic matrices, we proceed to the final solution. The reader is referred to Eqs.((I,16to 39)) of Ref. (2) for the basic theory.

BUILDING UP and SOLUTION of EQUATIONS

§ 1 $$\underset{[(p-2),(t-3)]}{D_{11}} = b_{1\ell}^{t}\ f_\ell \quad b_{1\ell} + b_{1q}^{t} \quad f_q \quad b_{1q} + b_{1r}^{t} \quad f \quad b_{1r}$$

§ 2 $$\underset{[(p-2)\times 1,(t-3)\times \rho]}{D_{10}} = b_{1\ell}^{t}\ f_\ell \quad b_{0\ell} + b_{1q}^{t} \quad f_q \quad b_{0q} + b_{1r}^{t} \quad f_r \quad b_{0r}$$

§ 3 $$\underset{[(p-2)\times 1,\,(t-3)\times p]}{Y} = (-1)\, D_{11}^{-1}\, D_{10}$$

§ 4 $$b_{\ell} = b_{0\ell} + b_{1\ell}\, Y$$

§ 5 $$b_{q} = b_{0q} + b_{1q}\, Y$$

§ 6 $$b_{r} = b_{0r} + b_{1r}\, Y$$

§ 7 $$F = b_{0\ell}^{t}\, f_{\ell}\, b_{\ell} + b_{0q}^{t}\, f_{q}\, b_{q} + b_{0r}^{t}\, f_{r}\, b_{r}$$

§ 8 Stop

II-h Input of Data and Preparation for the Programme

It is all too apparent from the developments of the present dissertation and those of the parent-work (see Refs. (1) ,(2))that the approach to the automatic analysis of a large structure using an electronic computer is basically different from the classical ideas associated with the analysis of a smaller size problem on a desk machine. This new attitude inevitably also influences the way the data is given to the computer. As a fundamental premise we should always keep in mind that the input of unnecessary data should always be avoided. In other words, what can be calculated by the machine, should never be done by hand. Data should always be given to the machine using the full accuracy of which the machine is capable. Unnecessarily prepared data only increase the possibility of error. Here we state the minimum amount of data that are initially required

A) Geometrical Data

No. of flanges t , no. of rings p , no. of loading cases ρ

$$\underset{[p\times 1\,,\,t\times 1]}{\mathbf{x}} \qquad \underset{[p\times 1\,,\,t\times 1]}{\mathbf{y}}$$

$$\underset{[p\times 1\,,\,t\times 1]}{\boldsymbol{\xi}} \qquad \underset{[p\times 1\,,\,t\times 1]}{\boldsymbol{\eta}}$$

$$\underset{[p\times 1\,,\,t\times 1]}{\mathbf{A}_-} \qquad \underset{[p\times 1\,,\,t\times 1]}{\mathbf{A}_+}$$

$$\underset{[p\times 1\,,\,t\times 1]}{\mathbf{C}_-} \qquad \underset{[p\times 1\,,\,t\times 1]}{\mathbf{C}_+}$$

$$\underset{[p\times 1\,,\,t\times 1]}{\mathbf{J}_-} \qquad \underset{[p\times 1\,,\,t\times 1]}{\mathbf{J}_+}$$

$$\underset{[2(p-1)\times 1,\ t\times 1]}{\mathbf{B}} \qquad \underset{[(p-1)\times 1,\ t\times 1]}{\mathbf{t}}$$

$$\underset{[(p-1)/,\ (\,)]}{\mathbf{d}}$$

E Modulus of elasticity

G Shear modulus

A generalization can easily be made to take care of the case when the elements of the fuselage are made of different materials. Thus instead of E we have the diagonal matrices

$\underset{[p/,\ t/]}{\mathbf{E}_r}$ of the E modulii for ring segments.

$\underset{[(p-1)/,\ t/]}{\mathbf{E}_\ell}$ of the E modulii for the flange segments.

Similarly

$\underset{[p/,\ t/]}{\mathbf{G}_r}$ of the shear modulii of the ring segments.

and

$\underset{[(p-1)/,\ t/]}{\mathbf{G}_q}$ of the shear modulii for the shear panels.

We must, however, not forget the principle mentioned before, namely that no data should be prepared unnecessarily by hand. It is possible for instance, when the fuselage panelling is uniform, to prepare the matrix $\mathbf{t}$ by a single trivial programme, for instance :-

$$\underset{[(p-1)\times 1,\, t\times 1]}{\mathbf{t}} = (t) \underset{[(p-1)\times 1,\, t\times 1]}{\mathbf{e}}$$

The formation of the $\mathbf{B}$ matrix, taking account of the skin contributions and adding it to the actual flanges that are present($\mathbf{B}_o$) can be achieved by using the following simple programme:-

ADDITION of the SKIN CONTRIBUTIONS to FLANGES

§ 1 $$\underset{[2(p-1)\times 1,\, t\times 1]}{\mathbf{t}_{ev}} = \left[\varepsilon_{T+} + \varepsilon_{T-} \right]^t \underset{[(p-1)\times 1,\, t\times 1]}{\mathbf{t}}$$

§ 2 Rearrange $\mathbf{t}_{ev}$ as diagonal $\underset{[2(p-1)/\,,\, t/]}{\mathbf{t}_e}$

§ 3 $$\underset{[2(p-1)\times 1,\, t\times 1]}{\mathbf{B}} = \mathbf{B}_o + \mathbf{t}_e\, \varepsilon_\ell \left(\tfrac{1}{2}\right) \underset{[(),\, t]}{\boldsymbol{\beta}^t} \quad \mathbf{l}$$

§ 4 Stop

Of course, this contribution can still be weighted by a diagonal matrix, thus for instance reducing the contribution at the tip or even bringing it down to zero.

The same method could, of course, also be applied to the rings. A certain freedom, however, must be left in such a case for the stressman to interfere with some parts of this programme. This can be done by a set of weighting diagonal matrices which will mostly consist of a scalar, or a group of scalars determined by the stressman to control his idealisation. The programme is not described in detail here. It is similar in principle to that of the flange areas, only more complicated by the fact that the co-ordinates of the minor polygon have to be modified too.

We must not forget also that, if we input a matrix $\underset{[p\times 1,\, t\times 1]}{\mathbf{h}}$ of the depths of the rings at the vertices, that the $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ might also be possibly calculated from the properties of the ring cross-section. This is in fact to be recommended in order to avoid errors. The same applies to $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{J}$.

Having obtained either by input or computation, the necessary data, we can proceed to the rest of the necessary geometrical matrices. Naturally enough we can only for instance add the contribution of the skin to the flanges after having calculated the operators

$$\boldsymbol{\varepsilon}_{T+}\ ,\ \boldsymbol{\varepsilon}_{T-}\ ,\ \boldsymbol{\varepsilon}_{\ell}\quad \text{and}\quad \mathbf{l}$$

In what follows we reproduce various programmes for calculation of the various geometrical data still required.

<u>FORMATION of $\mathbf{l}$</u>

§ 1 $$\underset{[p\times 1,\, t\times 1]}{\mathbf{x}_{\Delta}} = -\underset{[c,t]}{\boldsymbol{\alpha}^{t}}\ \mathbf{x}$$

§ 2 $$\underset{[p\times 1,\, t\times 1]}{\mathbf{y}_{\Delta}} = -\underset{[c,t]}{\boldsymbol{\alpha}^{t}}\ \mathbf{y}$$

§ 3 Rearrange $\underset{[p\times 1,\ t\times 1]}{x_\Delta}$ as diagonal $\underset{[p/,\ t/]}{x_\Delta^d}$

§ 4 Rearrange $\underset{[p\times 1,\ t\times 1]}{y_\Delta}$ as diagonal $\underset{[p/,\ t/]}{y_\Delta^d}$

§ 5 Form $\underset{[p\times 1,\ t\times 1]}{l^{(2)}} = x_\Delta^d \quad x_\Delta + y_\Delta^d \quad y$

§ 6 Elements of $\underset{[p\times 1,\ t\times 1]}{l}$ = Square Root of Elements of $l^{(2)}$

§ 8 Stop

λ follows immediately in the same manner, thus

FORMATION of λ

§ 1 $\underset{[p\times 1,\ t\times 1]}{\xi_\Delta} = - \underset{[1,\ t]}{\alpha^t} \quad \xi$

§ 2 $\underset{[p\times 1,\ t\times 1]}{\eta_\Delta} = - \underset{[1,\ t]}{\alpha^t} \quad \eta$

§ 3 Rearrange ξ_Δ as diagonal $\underset{[p/,\ t/]}{\xi_\Delta^d}$

§ 4 Rearrange η_Δ as diagonal $\underset{[p/,\ t/]}{\eta_\Delta^d}$

§ 5 Form $\lambda^{(2)} = \xi_\Delta^d \quad \xi_\Delta + \eta_\Delta^d \quad \eta_\Delta$

§ 6 Elements of $\underset{[p\times 1,\ t\times 1]}{\lambda}$ = Square Root of Elements of $\lambda^{(2)}$

§ 7 Stop

The computation of the two column super-matrices S_α and C_α follows again immediately

FORMATION of S_α and C_α

§ 1 Rearrange λ as diagonal $\underset{[p/,\ t/]}{\lambda^d}$

§ 2 $\underset{[p\times 1,\ t\times 1]}{S_\alpha} = \left[\lambda^d\right]^{-1} \eta_\Delta$

§ 3 $\underset{[p\times 1,\ t\times 1]}{C_\alpha} = \left[\lambda^d\right]^{-1} \xi_\Delta$

§ 4 Stop

B) The Loading on the Fuselage

Here again we need only input (or form) the matrices

$$\underset{[p\times 1\ ,\ t\times\rho]}{\mathbf{K}_{oR}} \qquad \text{and} \qquad \underset{[p\times 1\ ,\ t\times\rho]}{\mathbf{L}_{oR}}$$

of the loading at the ring vertices as well as the loading along the fuselage

$$\underset{[p\times 1,\ t\times\rho]}{\mathbf{P}_{\ell}}$$

Assuming that the loading on the fuselage consists only of such concentrated forces at the ring vertices in the directions x , y and z , we can calculate the load-resultant matrices $\hat{\mathbf{R}}_{\ell}$ and $\hat{\mathbf{R}}_{q}$.

The matrices $\mathbf{K}_{oR}$, $\mathbf{L}_{oR}$ and $\mathbf{P}_{\ell}$ in themselves mainly consist of

1) concentrated loads, e.g. freight loads, loads transmitted by other aircraft components.
2) aerodynamic loads (idealised)
3) inertia loads (idealised)

These three components are best calculated separately, and then added.

Part 1) will be normally a sparsely populated matrix and is best computed by individual addressing of the force values.

Part 2) is best calculated by an aerodynamic programme using matrix language. Whether a suitable procedure is available is questionable, but at any rate this should be the ideal.

Part 3) is very easily calculated from the matrix of the mass distribution as well as the accelerations of the aircraft. These in turn might also be calculated from the force distribution on the complete body; (see Ref. (2)).

Having obtained these three matrices, we proceed to form the other matrices

FORMATION of $\hat{R}_q$

§ 1 $$\underset{[p\times 1,\ 1\times \rho]}{F_{x\Delta}} = \underset{[(\ ),\ 1\times t]}{e^{t}} \quad K_{oR}$$

§ 2 Join $$\underset{[(p-1)\times p,(\ )]}{\mathcal{E}_N} = \begin{bmatrix} \underset{[(p-1),(\ )]}{\Sigma} & \underset{[(p-1)\times 1,(\ )]}{0} \end{bmatrix}$$

§ 3 $$\underset{[(p-1)\times 1,\ 1\times \rho]}{F_x} = \mathcal{E}_N \quad F_{x\Delta}$$

§ 4 $$\underset{[p\times 1,\ 1\times \rho]}{F_{y\Delta}} = \underset{[(\ ),\ 1\times t]}{e^{t}} \quad L_{oR}$$

§ 5 $$\underset{[(p-1)\times 1,\ 1\times \rho]}{F_y} = \mathcal{E}_N \quad F_{y\Delta}$$

§ 6 Transpose Elements of $\mathbf{X}$ to give $\underset{[p\times 1,\,1\times t]}{\mathbf{X}^{*}}$

§ 7 Transpose Elements of $\mathbf{y}$ to give $\underset{[p\times 1,\,1\times t]}{\mathbf{y}^{*}}$

§ 8 Rearrange $\mathbf{X}^{*}$ in diagonal form $\underset{[p/,\,1\times t]}{\mathbf{X}^{*d}}$

§ 9 Rearrange $\mathbf{y}^{*}$ in diagonal form $\underset{[p/,\,1\times t]}{\mathbf{y}^{*d}}$

§ 10 $$\underset{[p\times 1,\,1\times \rho]}{\mathbf{T}_{\Delta}} = \mathbf{X}^{*d}\,\mathbf{L}_{oR} - \mathbf{y}^{*d}\,\mathbf{K}_{oR}$$

§ 11 $$\underset{[(p-1),\,1\times \rho]}{\mathbf{T}} = \mathcal{E}_{N}\ \mathbf{T}_{\Delta}$$

§ 12 Transpose Elements of $\mathbf{F}_{x}$ to give $\underset{[(p-1)\times 1,\,\rho\times 1]}{\mathbf{F}_{x}^{*}}$

§ 13 Transpose Elements of $\mathbf{F}_{y}$ to give $\underset{[(p-1)\times 1,\,\rho\times 1]}{\mathbf{F}_{y}^{*}}$

§ 14 Transpose Elements of $\mathbf{T}$ to give $\underset{[(p-1)\times 1,\,\rho\times 1]}{\mathbf{T}^{*}}$

§ 15 Join $\underset{[(p-1)\times 1,\, \rho\times 3)}{\hat{R}_q^*} = \begin{bmatrix} F_x^* & F_y^* & T^* \end{bmatrix}$

§ 16 Transpose Elements of $\hat{R}_q^*$ to give $\underset{[(p-1)\times 1,\, 3\times \rho]}{\hat{R}_q'}$

§ 17 Form $\underset{[2(p-1)\times 1,\, 3\times \rho]}{\hat{R}_q} = \underset{[2(p-1)\times (p-1),\, ()]}{\bar{\mathcal{E}}_\ell} \ \underset{[(p-1)\times 1,\, 3\times \rho]}{\hat{R}_q'}$

§ 18 Stop

Then we proceed to $\hat{R}_\ell$

FORMATION of $\hat{R}_\ell$

§ 1 $\underset{[p\times 1,\, 1\times \rho]}{N_\Delta} = \underset{[(),\, 1\times t]}{e^t} \ P_\ell$

§ 2 $\underset{[2(p-1)\times 1,\, 1\times \rho]}{N} = \underset{[2(p-1),\, ()]}{\sum} \ \mathcal{E}_{\ell +} \ N_\Delta$

§ 3 $\underset{[p\times 1,\, 1\times \rho]}{M_{\ell y \Delta}} = X^{*d} \ P_\ell$

§ 4 $\underset{[(p-1)\times 1,\, 1\times \rho]}{M_{Ry\Delta}} = \underset{[(p-1),(\,)]}{\sum} \; d \quad F_x$

§ 5 $\underset{[2(p-1)\times 1,\, 1\times \rho]}{M_y} = \underset{[2(p-1),(\,)]}{\sum} \left[\varepsilon_{\ell+} \; M_{\ell y\Delta} + \varepsilon^{t}_{T-} \; M_{Ry\Delta} \right]$

§ 6 $\underset{[p\times 1,\, 1\times \rho]}{M_{\ell x\Delta}} = y^{*\alpha} \quad P_\ell$

§ 7 $\underset{[(p-1)\times 1,\, 1\times \rho]}{M_{Rx\Delta}} = \underset{[(p-1),(\,)]}{\sum} \; d \quad F_y$

§ 8 $\underset{[2(p-1)\times 1,\, 1\times \rho]}{M_x} = \underset{[2(p-1),(\,)]}{\sum} \left[\varepsilon_{\ell+} \; M_{\ell x\Delta} + \varepsilon^{t}_{T-} \; M_{Rx\Delta} \right]$

§ 9 Transpose Elements of N to give $\underset{[2(p-1)\times 1,\, \rho\times 1]}{N^*}$

§ 10 Transpose Elements of M_y to give $\underset{[2(p-1)\times 1,\, \rho\times 1]}{M_y^*}$

§ 11 Transpose Elements of M_x to give $\underset{[2(p-1)\times 1,\, \rho\times 1]}{M_x^*}$

§ 12 Join $\underset{[2(p-1)\times 1,\, \rho\times 3]}{\hat{R}_\ell^*} = \left[N^* \quad M_y^* \quad M_x^* \right]$

§ 13 Transpose Elements of $\underset{[2(p-1)\times 1,\ \rho\times 3]}{\hat{R}^{*}_{\ell}}$ to give $\underset{[2(p-1)\times 1,\ 3\times \rho]}{\hat{R}_{\ell}}$

§ 14 Stop

Having obtained this, all preparations are ready and one can proceed to solve the problem.

CHAPTER III

MODIFICATIONS AND CUT-OUTS

III-a Summary and Introduction

The first step in the analysis of a fuselage is, as described in Refs. (1) and (2), its regularization so as to yield a topologically continuous structure. This leads to the programme set out in some detail in Chapter II. Subsequently, when dealing with the cut-outs and modifications necessary to simulate the original system one would agree, that extraction of the rows of the $\mathbf{S}$ and $\mathbf{b}_1$ matrices, as well as formation of $\mathbf{f}_{\Delta h}$ present the main difficulty, due to the great possibility of error. In accordance with the main philosophy of this work, this part of the programme should therefore be written so as to reduce this danger to a minimum by full automatisation of the procedure, reducing the human element's contribution to what it should be, viz. the supervision of the programme. A set of instructions having a clear physical meaning indicates which elements are to be modified and which stresses are to be nullified. These instructions will then control the input of new section constants and properties for the modified elements. Since this is logically sufficient for the machine to solve the problem, it would be wrong to give anything further. The input should, as always, be logically sufficient, i.e. should contain all the required data in the form of logical instructions and numerical information, but should never contain redundant information, unless of course it is meant purposely to be a check on the rest.

The purpose of this chapter is to define a technique suitable for such a programme in general terms so as to be applicable to all computers. It should be noted, however, that this one has been developed for the Ferranti Pegasus, used here for all the relevant calculations.

Although the programme has been written for a fuselage, it should also be in principle applicable to all other regularized structures analysed by the Force Method. The programme uses again Boolean matrices which are logical matrices consisting of zeros and ones and which are not always stored in full. This type of matrix is best introduced as a special class of matrices in the super code, and all the arithmetical operations associated with them(either of pure logical or semi-logical nature)defined specially. A preliminary discussion of these matrices may be found in Appendix I.

The problem of the automatisation of the cut-out and modification techniques for any structure in general, and for the fuselage in particular, reduces to three parts: The input and storage of the information in the computer, the construction of the basic matrices required for performing the calculation, and the straightforward matrix programme resulting in the modified stresses. It should be noted, however, that the basic matrices required for the calculation of the cut and modified structure are no longer the $\mathbf{b}_h$, $\mathbf{b}_{1h}$ and $\mathbf{f}_{\Delta h}$ but rather the logical matrices resulting in their formation.

II-b The Input and Storage of Information in the Computer.

The input to the computer in preparation for a cut-out and modifications run consists of two parts: The first comprises the logical data specifying the affected elements and whether they are to be cut-out or modified. The second contains the numerical data needed for the modifications i.e. the new section constants.

In order to present the first part, we have to introduce a certain code, which is clear to the stress analyst as well as acceptable to the computer. Again we stress the fact that the code is primarily intended for the use of the structural engineer and not just for simplification of the input.

The code described here is for the Pegasus programme, and should be regarded as a guide rather than general for all other computers. The following meanings are ascribed to the letters or groups of letters below

CO	cut-out (s)
MN	Modification (s)
F	Flange
S	Shear panel
N	Normal force in ring
M	Bending moment in ring
A	Area of ring cross-section
I	Moment of inertia of ring cross-section.

The shear force in the ring is assumed to be coupled to the bending moment, and therefore any modifications in the moment of inertia $\mathcal{J}$ or in the cross-sectional area effective in shear C will alter the flexibility, due to the fact that the bending flexibility carries also the shear influence. However, a change in a ring cross-section will result in the alteration of all A , C and $\mathcal{J}$ simultaneously.

In view of the explicit criticism of the couple shear and bending flexibility put forward in Ref. (2) it is advisable when working with a large computer to separate these effects and carry distinct "stress" matrices for bending moment and shear force. For the case when the stress matrices contain the shear forces explicitly the technique can be extended easily.

Using the above symbols, we can now give the form of the orders or "macro-instructions", required to define the various operations. It should be noted further that the symbols (+) or (-) after a number mean that a dimensional change takes place on one side only of the nodal point. The explanations follow with each instruction. Thus

(i)	CO * F5 * 1,2,8	= Cut-outs at frame station 5 Flanges no. 1,2 and 8 are affected
(ii)	MN * F5 * 3,7 + , 9,12-	= Modifications at frame station 5. Flanges 3 and 9 change areas on both sides whereas flange 12 is affected only at the "-" side of the ring, i.e. at the end of bay 4,5, and flange 7 only on the opposite side - i.e. the "+" side.
(iii)	CO * S3,4 * 4,8	= Cut-outs at bay 3,4. Panels no. 4 and 8 are removed.
(iv)	MN * S3,4 * 3,5,7,9	= Modifications at bay 3,4. Panels no. 3,5,7 and 9 are affected.
(v)	CO * N7 * 2,5	= Slides introduced in ring 7 at polygon sides 2 and 5.
(vi)	CO * M 7 * 6,7,9	= Hinges introduced in ring 7 at vertices 6,7 and 9.
(vii)	MN * A 6 * 1,2,6,8+,9-,10	= Modifications of ring 6. Areas at vertices no. 1,2,6 and 10 changed on both sides. At vertix 8, only the value on the side 8,9 is affected. At vertix 9, the value on the side (8,9) is also altered.

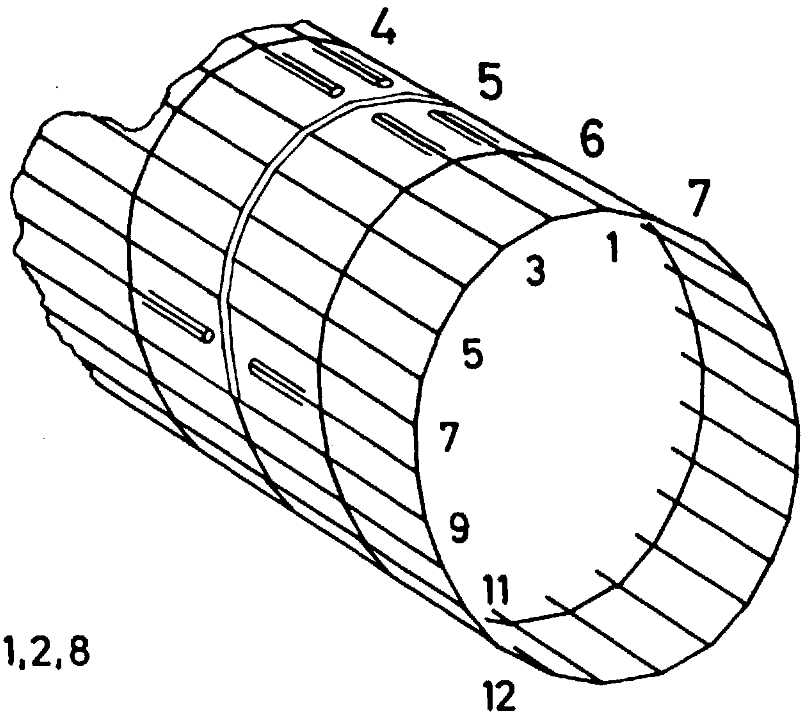


Fig.III-1
CO * F5 * 1,2,8

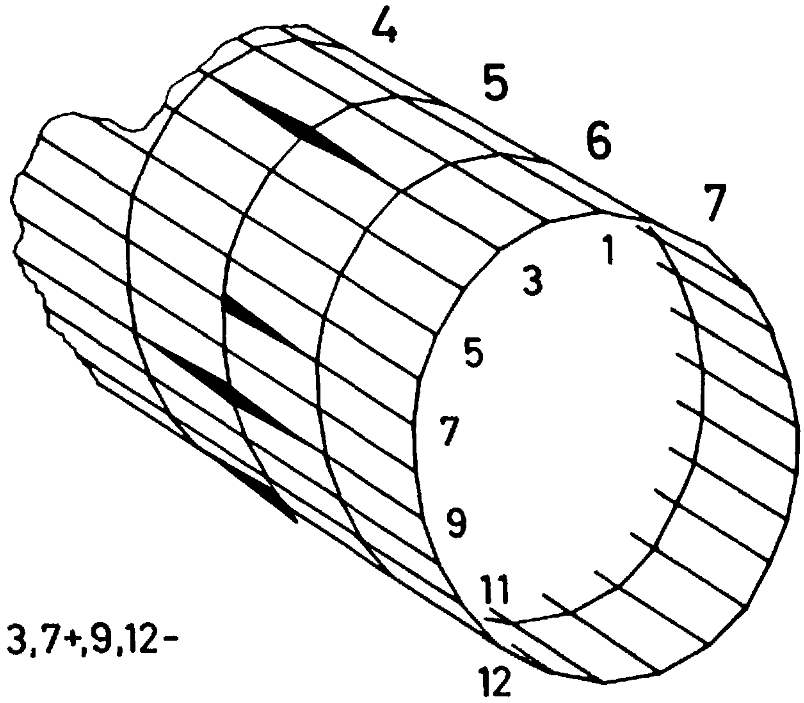


Fig.III-2
MN * F5 * 3,7+,9,12-

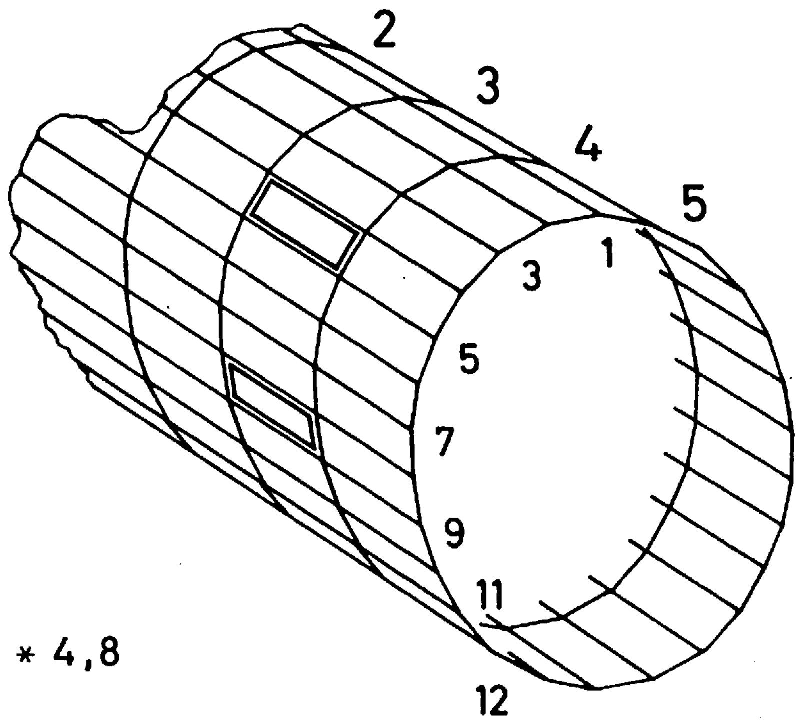


Fig.III-3
CO * S 3,4 * 4,8

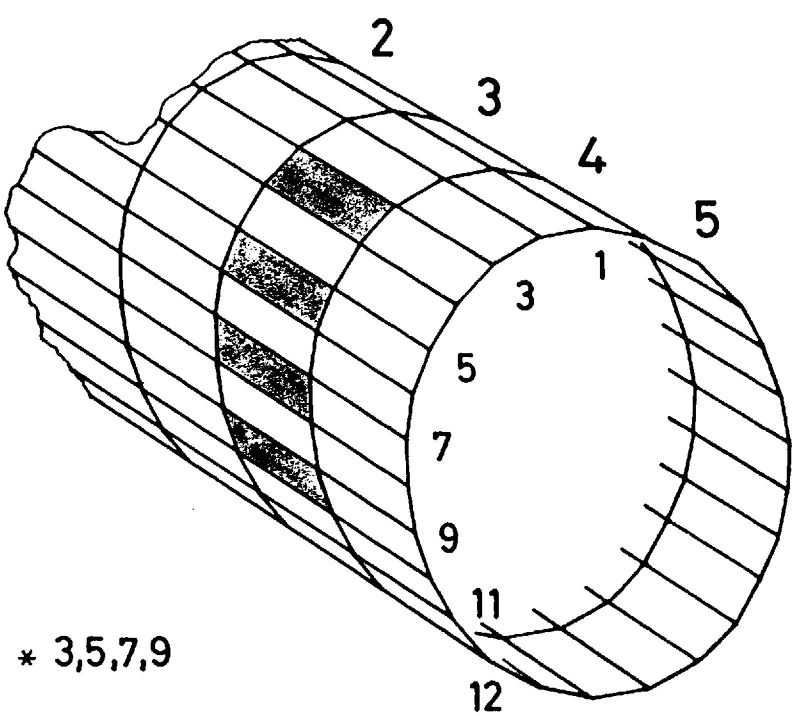


Fig.III-4
MN * S 3,4 * 3,5,7,9

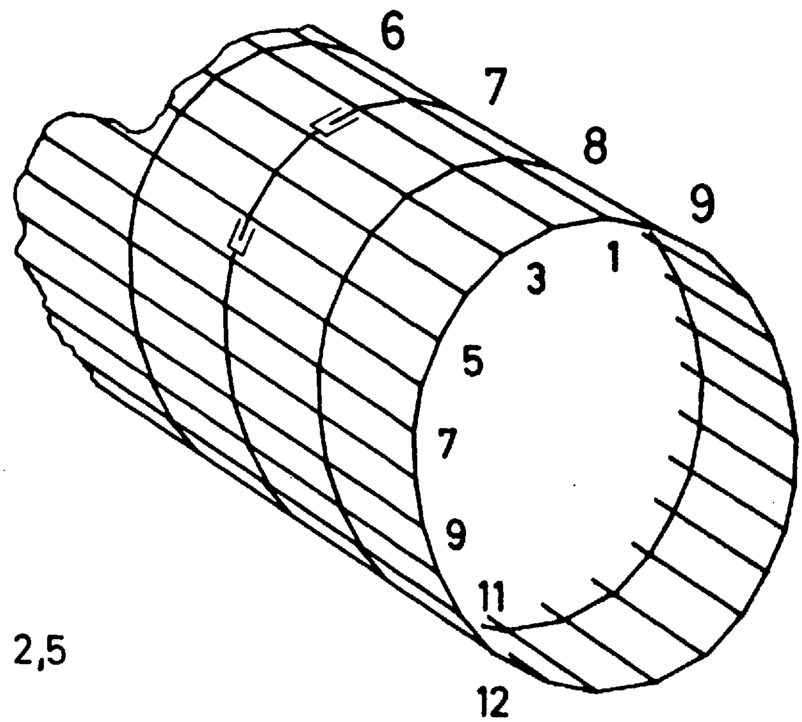


Fig.III - 5
CO * N 7 * 2,5

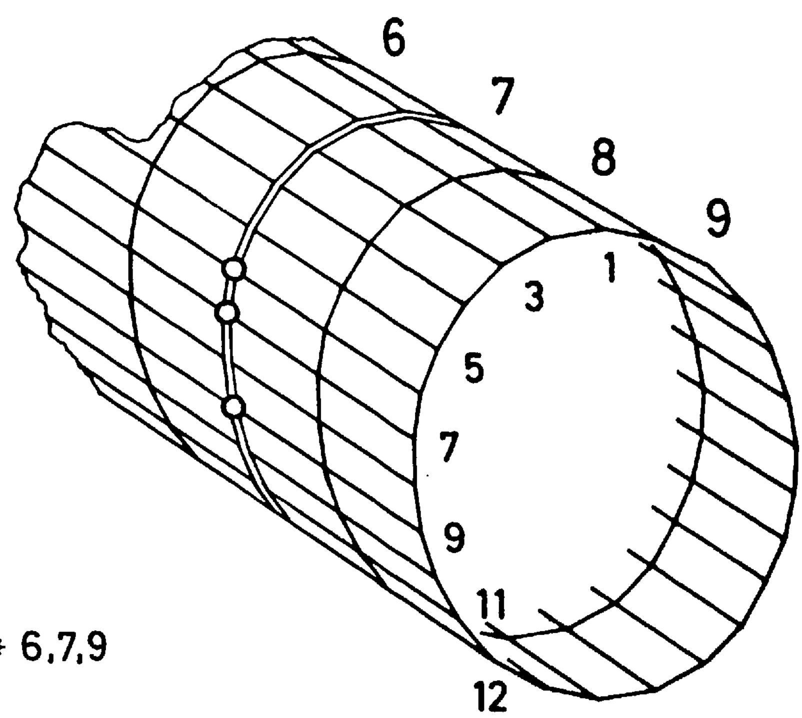


Fig.III - 6
CO * M 7 * 6,7,9

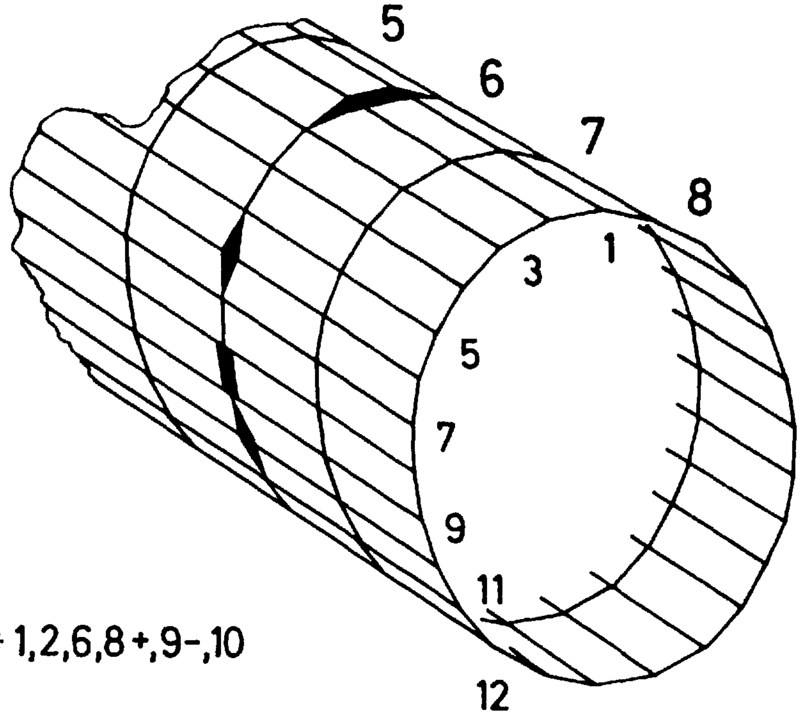


Fig.III -7
MN * A6 * 1,2,6,8+,9-,10

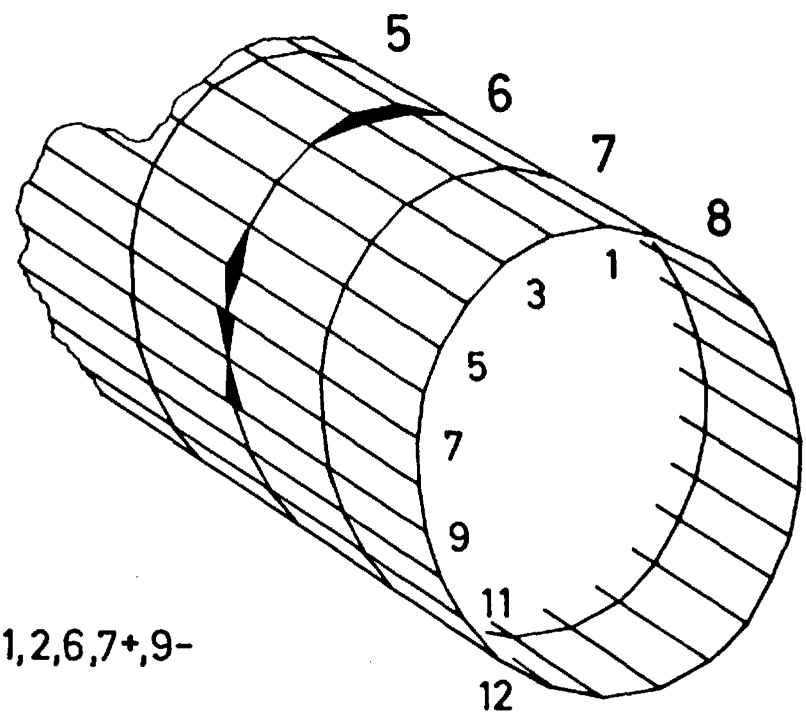


Fig.III - 8
MN * I6 * 1,2,6,7+,9-

(viii) MN * I6 * 1,2,6,7+,9- = Changes of the moment of inertia I (as well as area C) in ring 6. At vertices 1,2 and 6 the changes are on both sides. At vertex 7 only on the (7,8) side and at vertex 9, only on the (8,9) side.

Sketches of the physical meaning of these orders are shown in Figs.III-1 to 8

These macro instructions result first of all in the formation of four word "lists" which we call

$\underset{[2p-1]}{L_{co-c}}$ for the cut-outs in the cover,

$\underset{[2p]}{L_{co-r}}$ for cut-outs in the rings,

$\underset{[3(p-1)]}{L_{MN-c*}}$ a preliminary list of dimensional changes in the cover,

and $\underset{[6p]}{L_{MN-r*}}$ a preliminary list of dimensional changes in the rings.

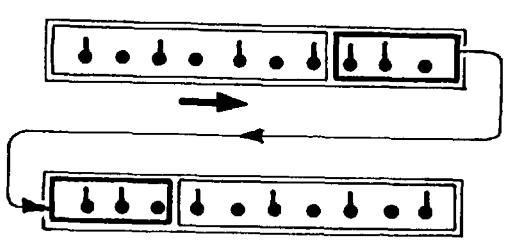
Each of these lists consist of a number of units. Each unit is either one word, or a group of words according to the type of computer, and the size of problems to be expected. The number of bits in each of these units correspondsto the maximum number of stresses along the periphery, i.e. the maximum number of flanges $t_{max.}$ to be expected in practice. It should be noted that the only case in which all bits of the unit are taken up by the word, is when the programme is running to full capacity, i.e. when the number of flanges is the maximum allowed. Otherwise, what we call the "listword" occupies only a part of the unit. This is important when performing "circular listword shifts".*

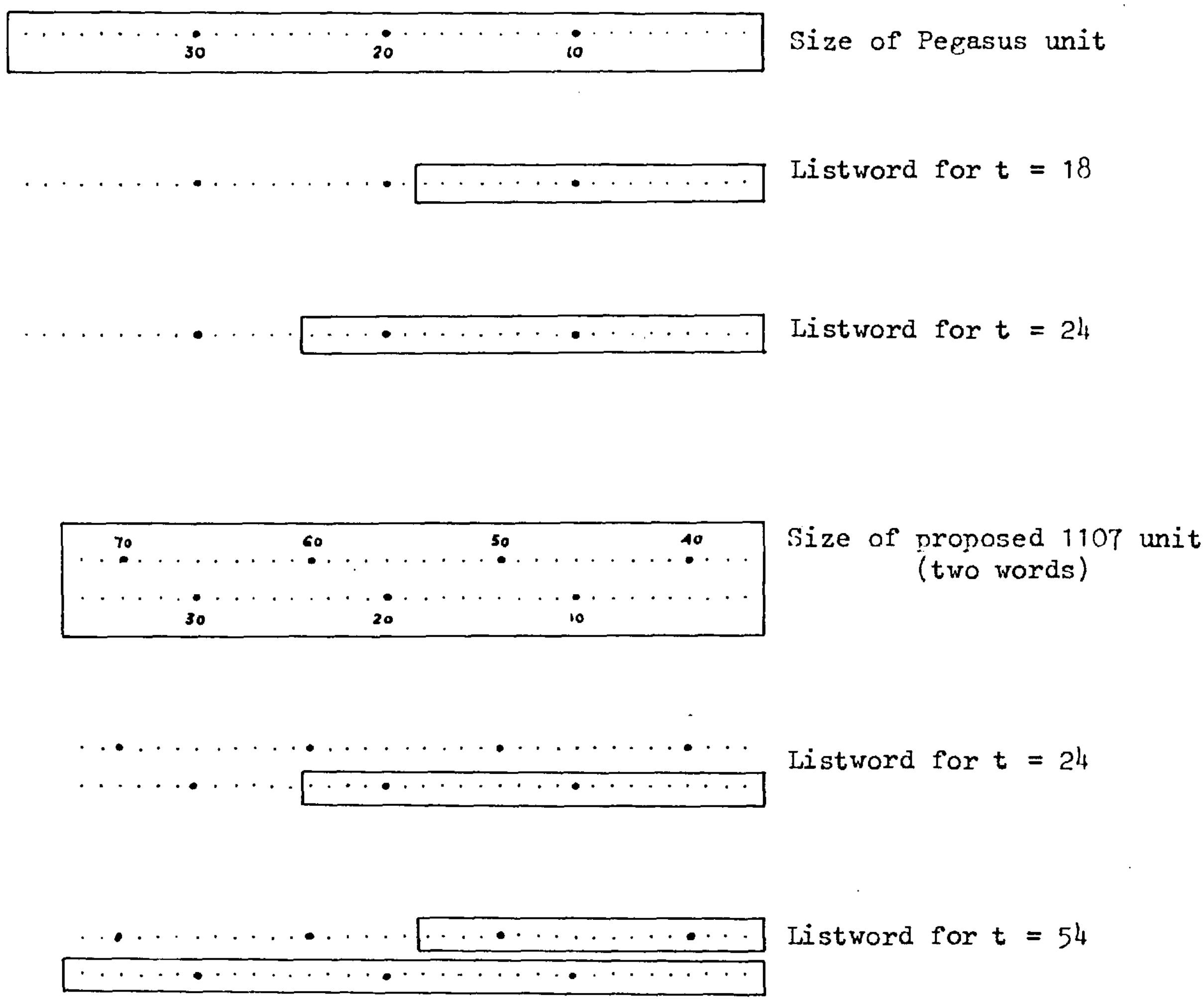
On the Pegasus, for instance, a full word is allowed for each unit so that the maximum number of flanges is 39, a value suitable for the size of this computer. For the UNIVAC 1107, on the other hand, such a value is too low. It would be desirable to allow two words for each unit, thus raising the maximum number of flanges to 72, which is consistent with the size and speed of this computer. A still larger unit might be necessary, if one has to include more than one circuit at each cross-section, for instance in a multi-cell fuselage. However, the choice of the two word units has advantage in the ease with which a double circular shift can be performed on the 1107, as well as in many other computers.

As mentioned before, since the number of items in each unit is generally less than the maximum, only the corresponding number of bits starting from the least significant position is used. Examples are best given in Fig. III-9.

---

* A circular word shift in computer terminology signifies a shift of the bits to the right (down shift), or to the left (up shift) , where the bits to the extreme right (respectively to the extreme left) are fed back again at the other end.
<u>Example:</u> A circular right shift of three places on a ten bit computer word.





(Fig. III-9) Examples of units and list words.

Now we describe the individual lists, which are all similar in nature.

<u>The list</u> $\underset{[2p-1]}{L}_{co-c}$

The list, in all $(2p-1)$ words long, consists in itself of two parts. These can be best expressed as

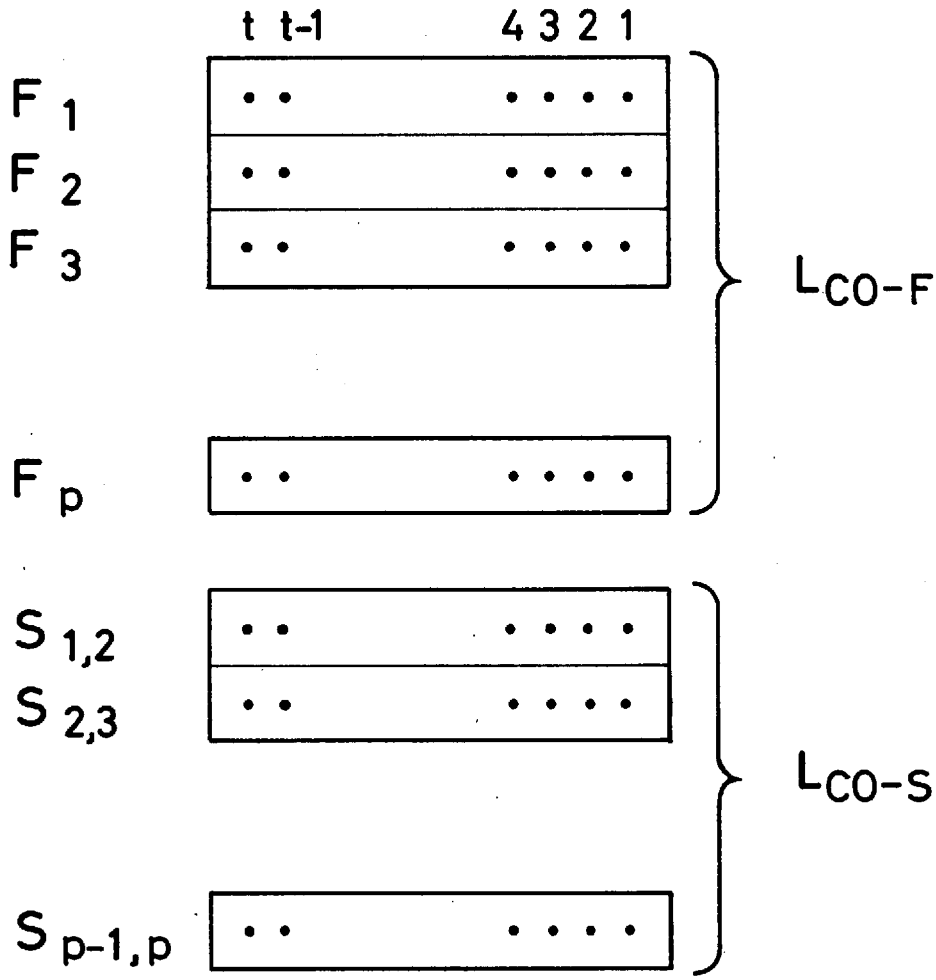
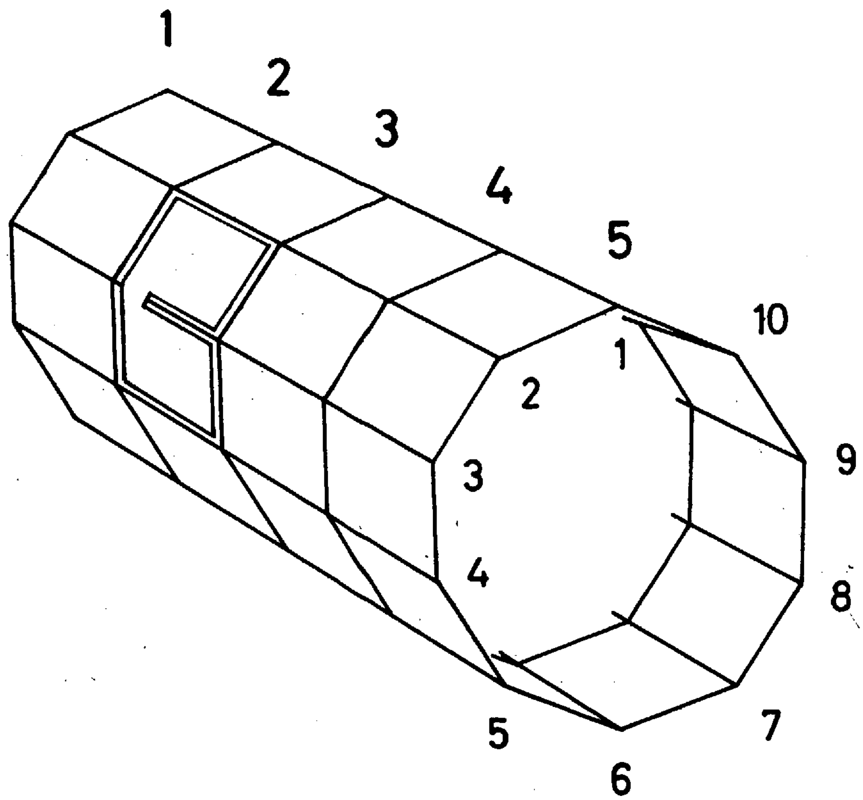


Fig. III-10
A typical L_{CO-C} list



	10	9	8	7	6	5	4	3	2	1
F_1	•	•	•	•	•	•	•	•	•	•
F_2	•	•	•	•	•	•	•	↓	•	•
F_3	•	•	•	•	•	•	•	•	•	•
F_4	•	•	•	•	•	•	•	•	•	•
F_5	•	•	•	•	•	•	•	•	•	•
$S_{1,2}$	•	•	•	•	•	•	•	•	•	•
$S_{2,3}$	•	•	•	•	•	•	•	↓	↓	•
$S_{3,4}$	•	•	•	•	•	•	•	•	•	•
$S_{4,5}$	•	•	•	•	•	•	•	•	•	•

Fig.III-11

Example of L_{CO-C} for a fuselage with cover cut-outs

$$\underset{[2p-1]}{\mathsf{L}_{co-c}} = \begin{bmatrix} \underset{[p]}{\mathsf{L}_{co-F}} \\ \underset{[p-1]}{\mathsf{L}_{co-S}} \end{bmatrix} \qquad \text{(III,1)}$$

It indicates which flanges or shear panels are cut. That is, if a bit in a certain unit is zero, the corresponding stress can take any value. If it is a "one" , the corresponding stress is nullified and therefore will have zero value.

L_{co-F} contains p listwords. The first represents the group of flange stresses at frame station 1 , the next unit those at station 2, and so on.

L_{co-S} is composed of $p-1$ listwords. The first represents the shear panels of bay (1,2), and so on.

We show the string of listwords in Fig.III,10 . An example of an actual fuselage is given in Fig.III,11.

The List L_{MN-C*}

This list is similar but refers to the positions where a design alteration has taken place. Since we allow a jump, or discontinuity of the flange area at a nodal point, we need two listwords instead of one to describe the flange station i (except for $i=1$ or p), the listwords are then referred to as F_{i-} and F_{i+} . In this manner the list consists of $3(p-1)$ listwords or

$$\underset{[3(p-1)]}{\mathsf{L}_{MN-C*}} = \begin{bmatrix} \underset{[2(p-1)]}{\mathsf{L}_{MN-F*}} \\ \underset{[p-1]}{\mathsf{L}_{MN-S*}} \end{bmatrix} \qquad \text{(III,2)}$$

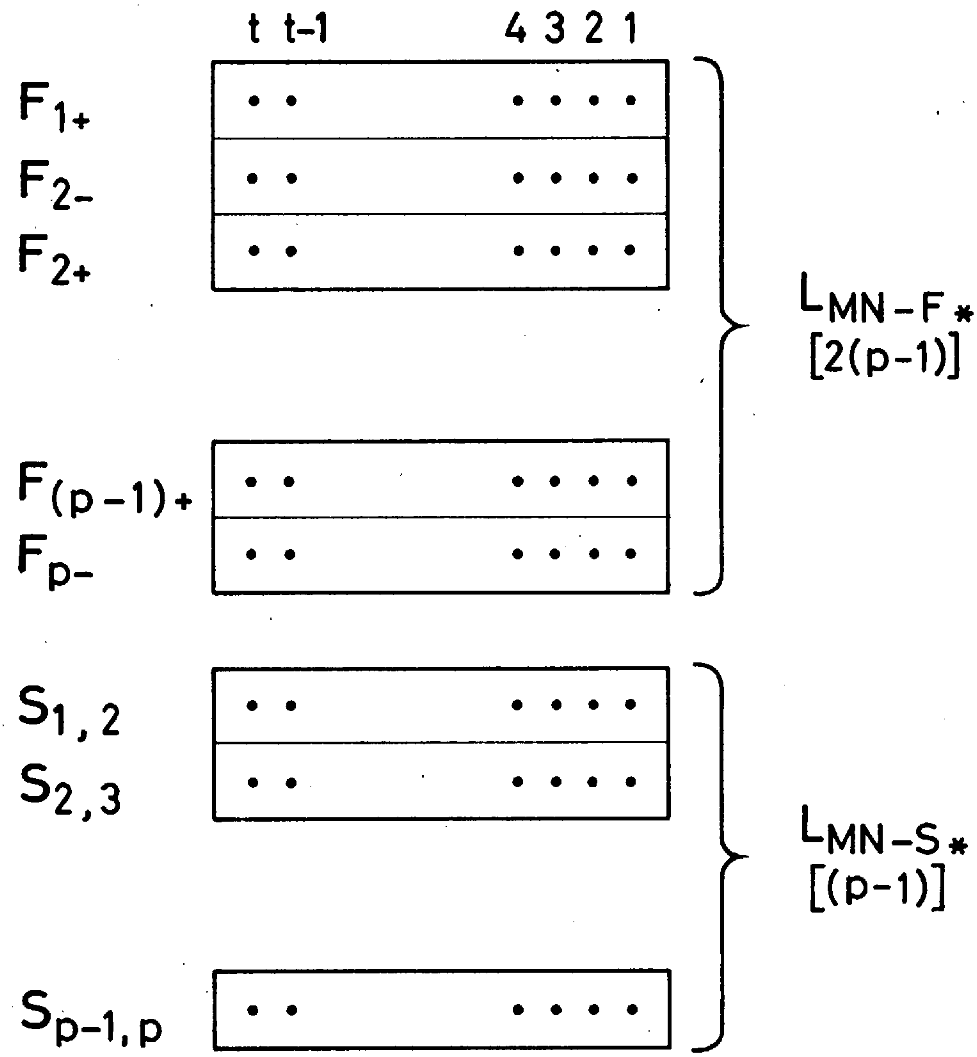
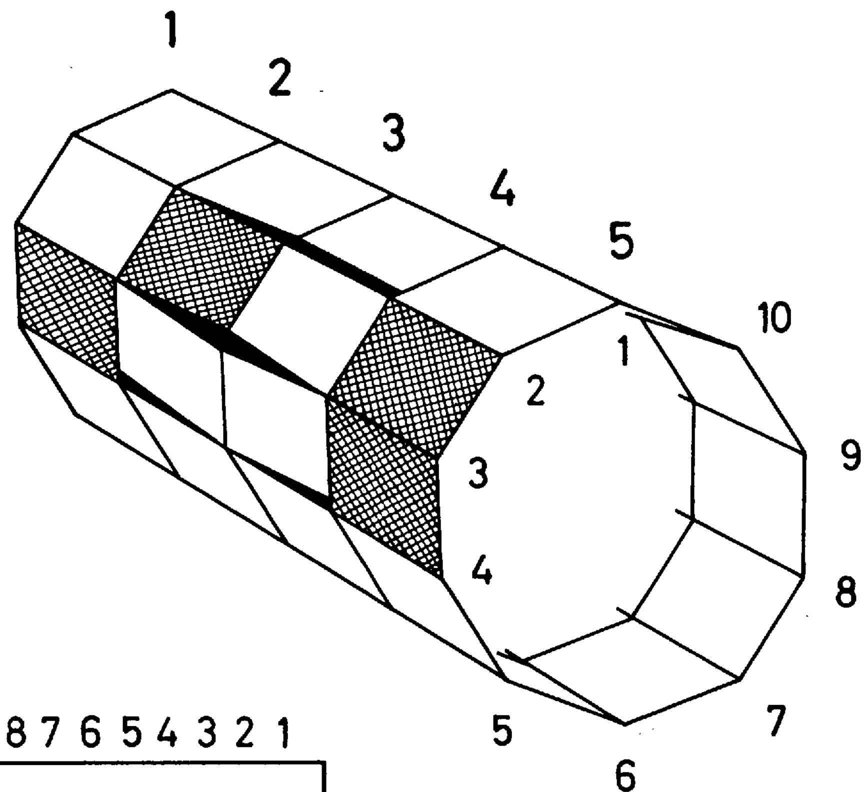


Fig. III-12

A typical L_{MN-C*} list



	10	9	8	7	6	5	4	3	2	1
F_{1+}	•	•	•	•	•	•	•	•	•	•
F_{2-}	•	•	•	•	•	•	•	•	•	•
F_{2+}	•	•	•	•	•	•	↓	•	•	•
F_{3-}	•	•	•	•	•	•	•	↓	↓	•
F_{3+}	•	•	•	•	•	•	•	↓	↓	•
F_{4-}	•	•	•	•	•	•	↓	•	↓	•
F_{4+}	•	•	•	•	•	•	•	•	•	•
F_{5-}	•	•	•	•	•	•	•	•	•	•
$S_{1,2}$	•	•	•	•	•	•	•	↓	•	•
$S_{2,3}$	•	•	•	•	•	•	•	•	↓	•
$S_{3,4}$	•	•	•	•	•	•	•	•	•	•
$S_{4,5}$	•	•	•	•	•	•	•	↓	↓	•

Fig.III-13

Example of primary modifications list L_{MN-C*}

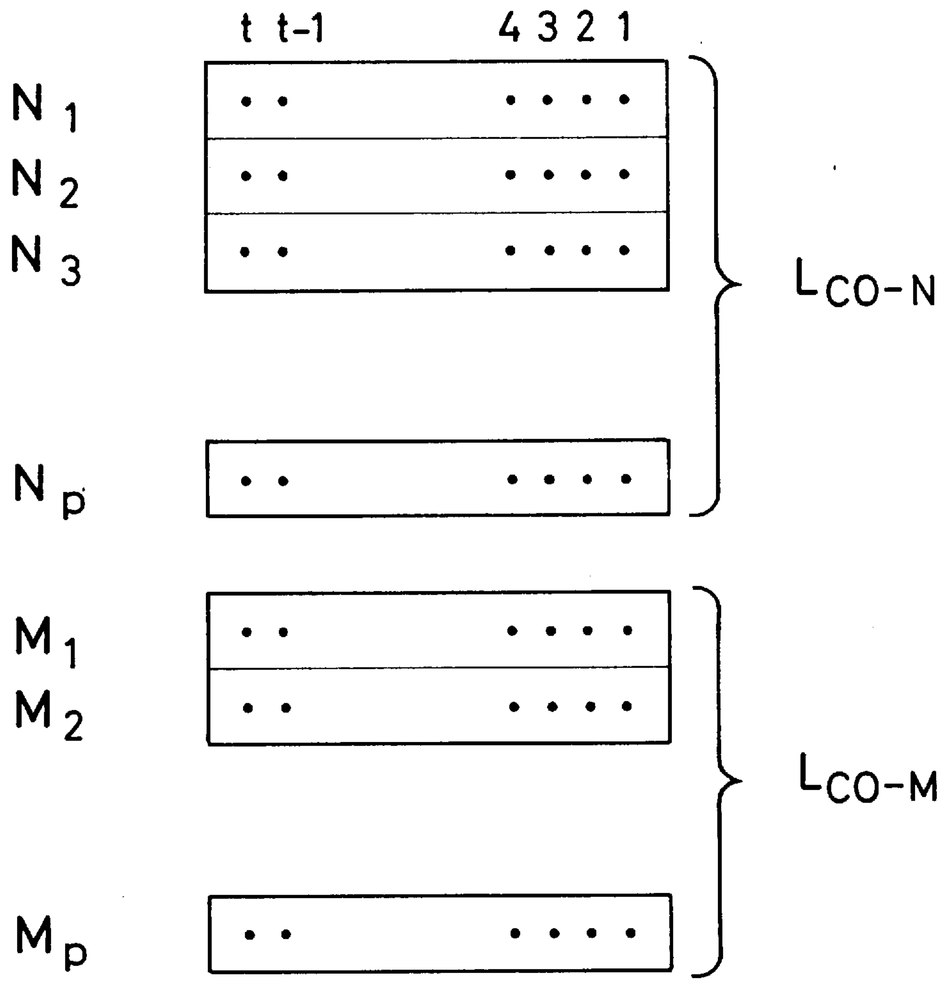
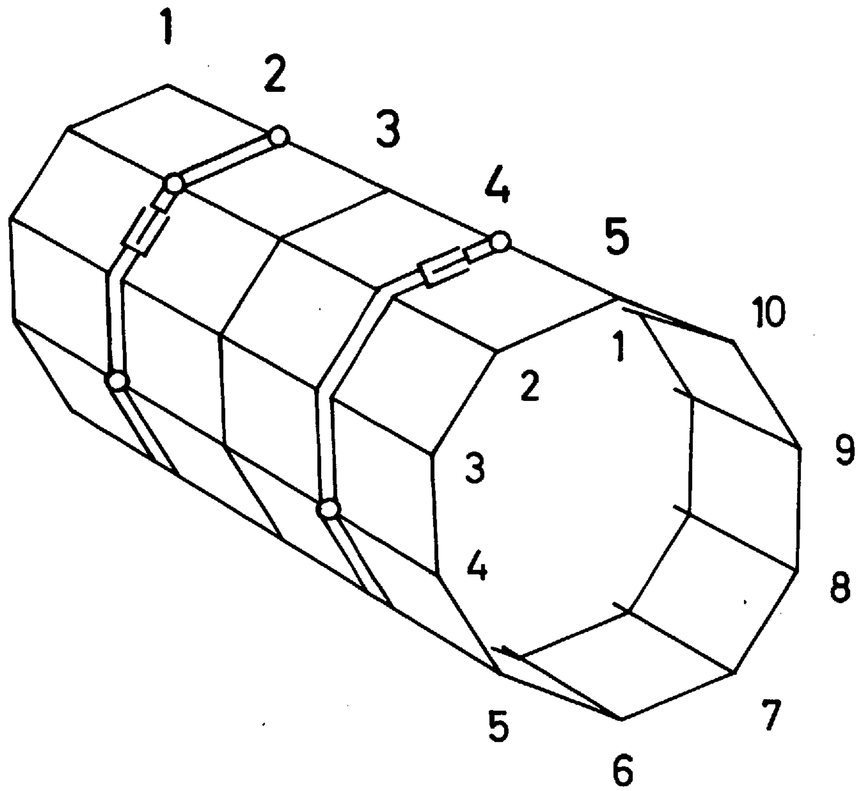


Fig. III-14
A typical L_{CO-r}



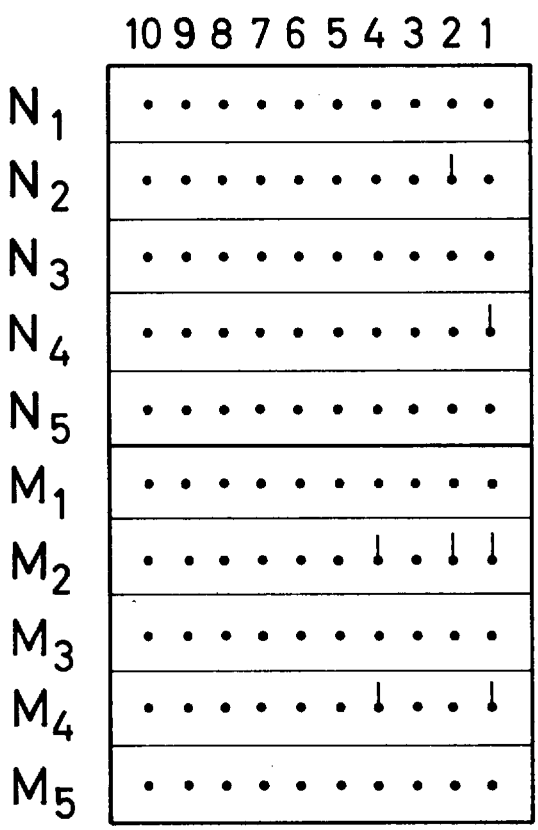


Fig.III-15

Example of $L_{CO\text{-}r}$ indicating the introduction of slides and hinges in the rings

Again the general scheme of the list is illustrated in Fig. III-12 and an example fuselage is shown in Fig. III-13 , the thick lines indicating where the modifications have taken place in the flanges, and the hatched panels are ones which have undergone a thickness alteration. One should keep in mind that the flange flexibilities are always assumed to vary linearly from one frame station to the next.

The list L_{MN-c*} is used only in the control of the input of the new flange areas and sheet thicknesses.

Corresponding to these two lists, we find also two lists for the rings, L_{co-r} and L_{MN-r*}.

<u>The List</u> L_{co-r}

Similar to L_{co-c}, but concerning the rings, this list contains $2p$ "listwords". It is composed of two smaller lists, each of p words, thus

$$\underset{[2p]}{L_{co-r}} = \begin{bmatrix} \underset{[p]}{L_{co-N}} \\ \\ \underset{[p]}{L_{co-M}} \end{bmatrix} \qquad \text{(III,3)}$$

The first list indicates the presence of slides* introduced along the periphery of the rings. Thus if a "one" is present, it denotes a slide, and if the bit is zero, the ring element is still capable of carrying direct load. The second list, L_{co-M} records where hinges are introduced at various vertices. The presence of a "one" at a certain bit means that the corresponding vertex has a hinge, otherwise it is capable of transmitting bending moment. As before we give the lay-out of the list in Fig. III,14 and an example in Fig. III,15

* Slides is in accordance with Ref.(2) the terminology for a cut-out nullifying the normal force.

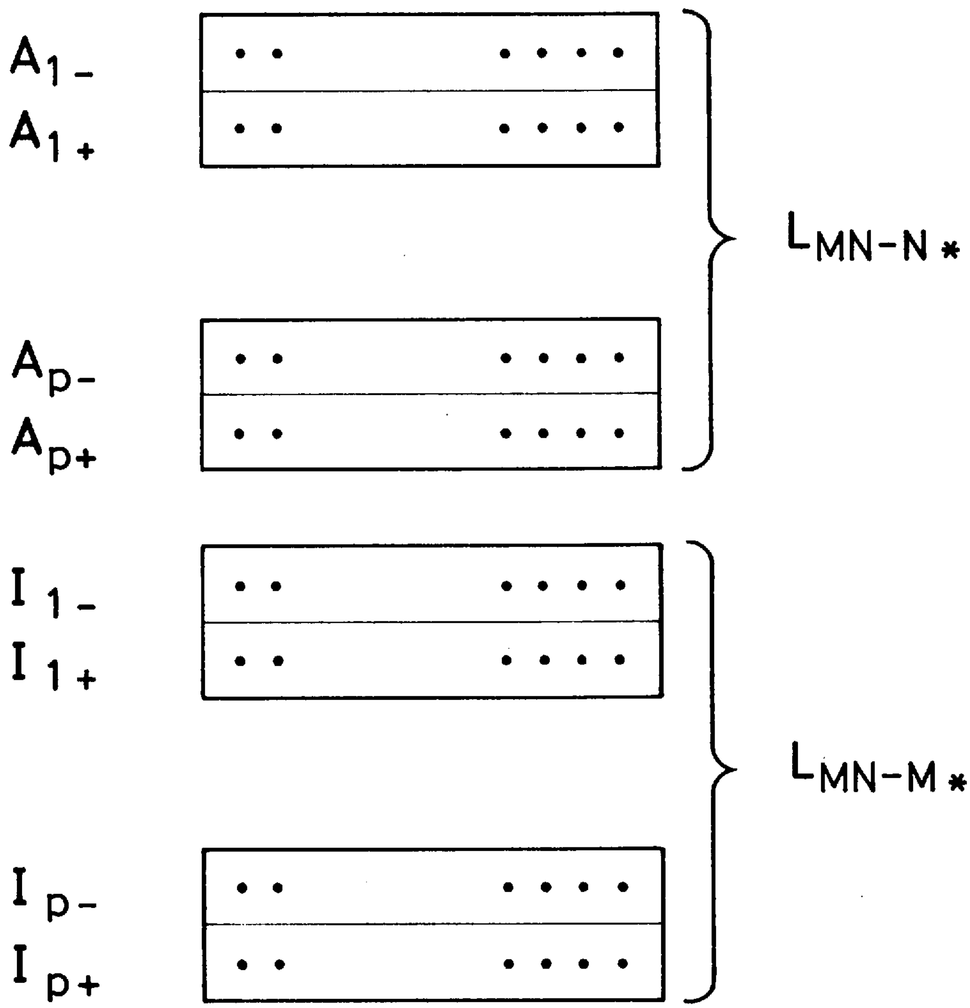
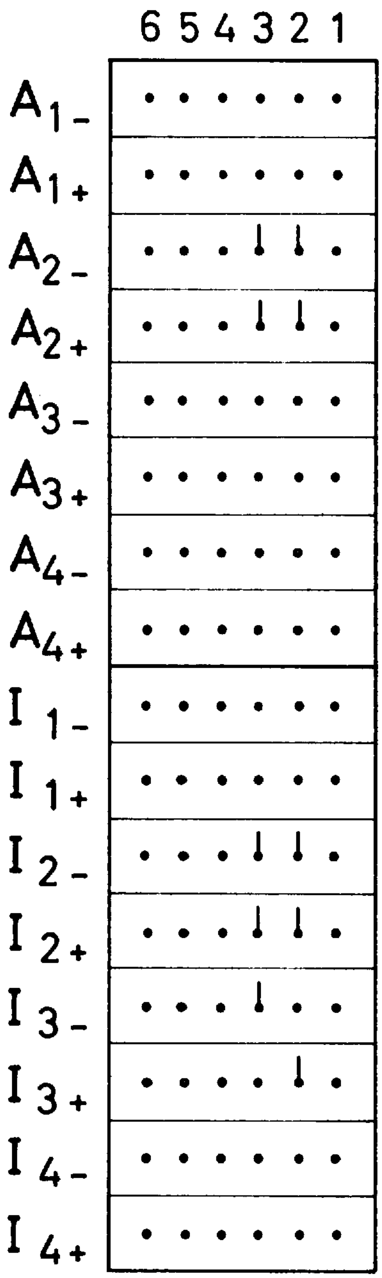


Fig. III - 16
Typical lay-out of L_{MN-r*}



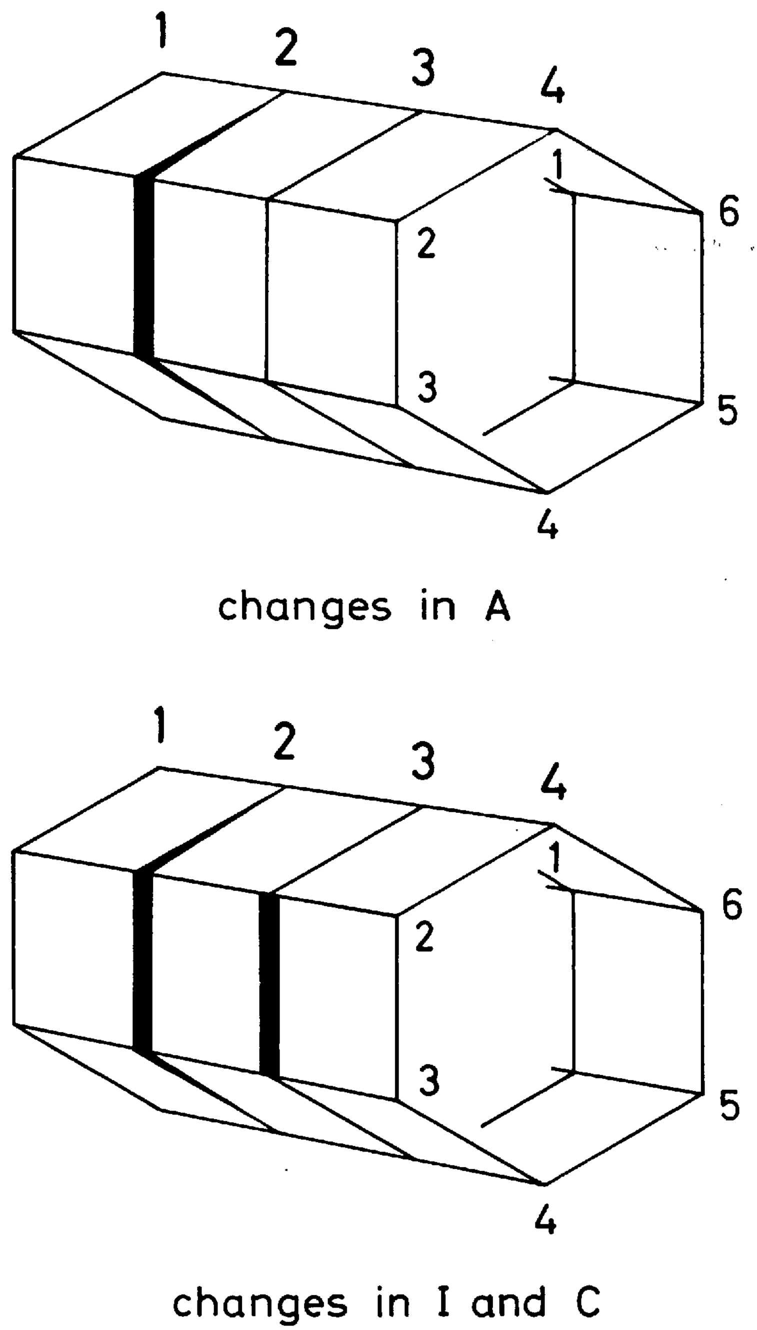


Fig.III - 17

Example of $L_{MN\text{-}r*}$

The List L_{MN-r*}

This list, again analogous to L_{MN-C*}, describes where modifications are introduced in the rings. Due to our admitting jumps in cross-sectional areas, as well as moments of inertia and areas effective in shear, we actually need $4p$ listwords in all. The list again consists of two parts, L_{MN-N*} and L_{MN-M*}. Thus

$$\underset{[4p]}{L_{MN-r*}} = \begin{bmatrix} \underset{[2p]}{L_{MN-N*}} \\ \\ \underset{[2p]}{L_{MN-M*}} \end{bmatrix} \qquad \text{(III,4)}$$

Each pair of words in the list L_{MN-N*} refers to a particular ring. The first listword indicates alterations in A_- at the various vertices, and the second in A_+. Similarly in the list L_{MN-M*} in each pair the first listword indicates alterations in J_- (and C_-), the second indicates alterations in J_+ (and C_+).

This is shown in Figs. (III-16) and (III-17) as a lay-out and typical example.

The Input Programme for the Lists

The first part of the cut-out and modification programme is concerned with reading the orders, or macro-instructions, from perforated paper tape, punched cards, magnetic tape, core or drum and forming the four lists

$$L_{CO-C}, \quad L_{MN-C*}, \quad L_{CO-r} \quad \text{and} \quad L_{MN-r*}$$

The information needed is contained within the instructions, and writing the necessary programme reduces then to a simple exercise. In the case of the Pegasus, the programme has been written to feed in the orders one by

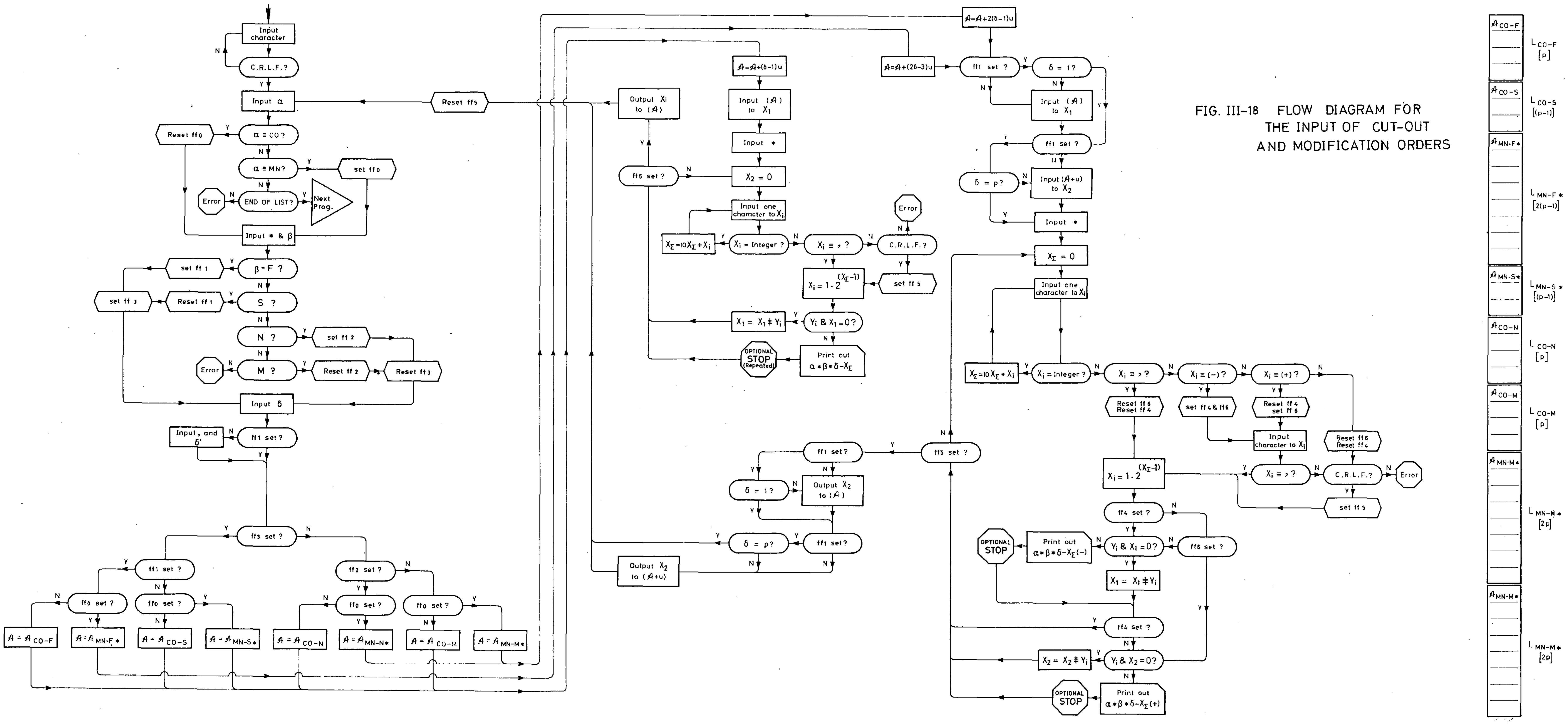


FIG. III-18 FLOW DIAGRAM FOR THE INPUT OF CUT-OUT AND MODIFICATION ORDERS one from paper tape and obey each one immediately upon its input. We can describe here such a programme by a simplified flow diagram (Fig. III,18), which can easily be extended to apply to other computers.

We observe first that the form of the order is generally

$$\alpha * \beta \left\langle \begin{matrix} \delta \\ \delta, \delta' \end{matrix} \right\rangle * \gamma_1 , \gamma_2 , \gamma_3 , \cdots\cdots \gamma_c .$$

At the beginning of the set of instructions as well as between each two consecutive orders there is a certain symbol. It might be simply the fact that the next instruction appears on a new punched card card, or in the case of punched tape, it takes the form of "carriage turn, line feed". In some codes the symbol is punched as two distinct characters, in some others as one. The end of the set instructions can be indicated by a special card, or in the case of paper tape by the presence of a length of free tape - say after a minimum of two blank characters.

The addresses of a certain list L_i is indicated by $\mathcal{A}_i$. Otherwise the flow-diagram is self-explanatory. We have to observe that although the flow diagram is directly applicable to paper-tape input, its extension to punched cards should not be too difficult. A set of logical yes/no mechanisms always gives an indication of the nature of the order, i.e. cut-out or modification, cover or ring, etc. Certain controls are provided for instance that no stress is mentioned twice. Further checks can be incorporated as a separate programme to ensure for example that no excessive number of cut-outs has been specified. (see also Chapter IV). These programmes might also have a built-in automatic correction.

The Secondary Modification Lists L_{MN-c} and L_{MN-r}

Whereas the primary modification lists L_{MN-c*} and L_{MN-r*} are mainly used to control the input of the new values of the areas, thicknesses, etc., these two new lists, which are analogous to L_{co-c} and L_{co-r}, are required for choosing the appropriate rows (and columns) of b_1 , S (and f_Δ) to give b_{1h} , S_h and $f_{\Delta h}$. The four lists L_{co-c} , L_{co-r} , L_{MN-c} and L_{MN-r} are not directly used for this purpose, but rather Boolean matrices based upon them. In order to illustrate the meaning attached to these lists, we now proceed to discuss them, taking into account the fact that these two secondary modification lists are in themselves divided into sub-lists. So,

$$\underset{[3(p-1)]}{L_{MN-c}} = \begin{bmatrix} \underset{[2(p-1)]}{L_{MN-F}} \\ \underset{[p-1]}{L_{MN-S}} \end{bmatrix} \qquad \text{(III,5)}$$

and

$$\underset{[2p]}{L_{MN-r}} = \begin{bmatrix} \underset{[p]}{L_{MN-C}} \\ \underset{[p]}{L_{MN-M}} \end{bmatrix} \qquad \text{(III,6)}$$

The List L_{MN-F}

Due to the assumption of linear variation of the flexibilities between each two stations, as well as the permitted jump in the values of the areas at frame stations, it follows that the modification of an area for example on the "+" side of the frame station i not only results in an alteration of the direct flexibility at that position, but also in the direct flexibility of the same flange at the "-" side of the next station ($i+1$). So, whereas the list L_{MN-F*} gives the actual positions of the flange-area changes, the list L_{MN-F} gives the position of all stresses in which the direct flexibilities are affected by such a modification.

We can easily see that the two units describing $(i)_+$ and $(i+1)_-$ are always the same in this new list. Furthermore we see the connection immediately between this and the logical addition, represented by the symbol $\oplus$. According to this, if two computer words are added, the corresponding binary bits become added "logically" to one another. The result is a $(+1)$ if one or both bits are one, and a zero only if both are zeros. Such a logical function exists on all computers either explicitly or implicitly. The new list L_{MN-F} may therefore be simply derived from the list L_{MN-F*} through the equations

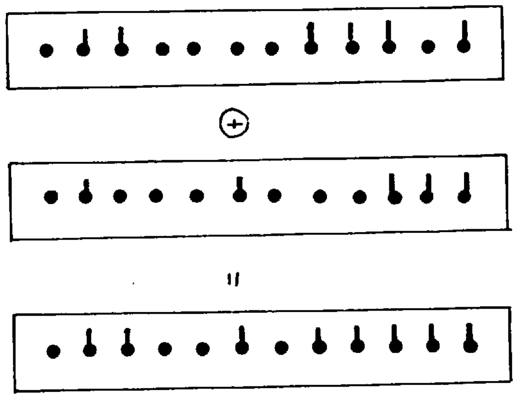
$$\left(L_{MN-F}\right)_{i+} = \left(L_{MN-F*}\right)_{i+} \oplus \left(L_{MN-F*}\right)_{(i+1)-} \qquad \text{(III,7)}$$

and

$$\left(L_{MN-F}\right)_{(i+1)-} = \left(L_{MN-F*}\right)_{i+} \oplus \left(L_{MN-F*}\right)_{(i+1)-} \qquad \text{(III,7a)}$$

where the expressions in the brackets refer obviously to the unit in the appropriate list connected with the frame station and side given by the subscript.

The flow diagram is very simple and need not be given, but the following example may be helpful:



$(L_{MN-F*})_{i+}$

$(L_{MN-F*})_{(i+1)-}$

$= (L_{MN-F})_{i+} = (L_{MN-F})_{(i+1)-}$

(Fig.III,19) Example of logical addition

The List L_{MN-S}

This is identical with L_{MN-S*}, thus

$$L_{MN-S} = L_{MN-S*} \tag{III,8}$$

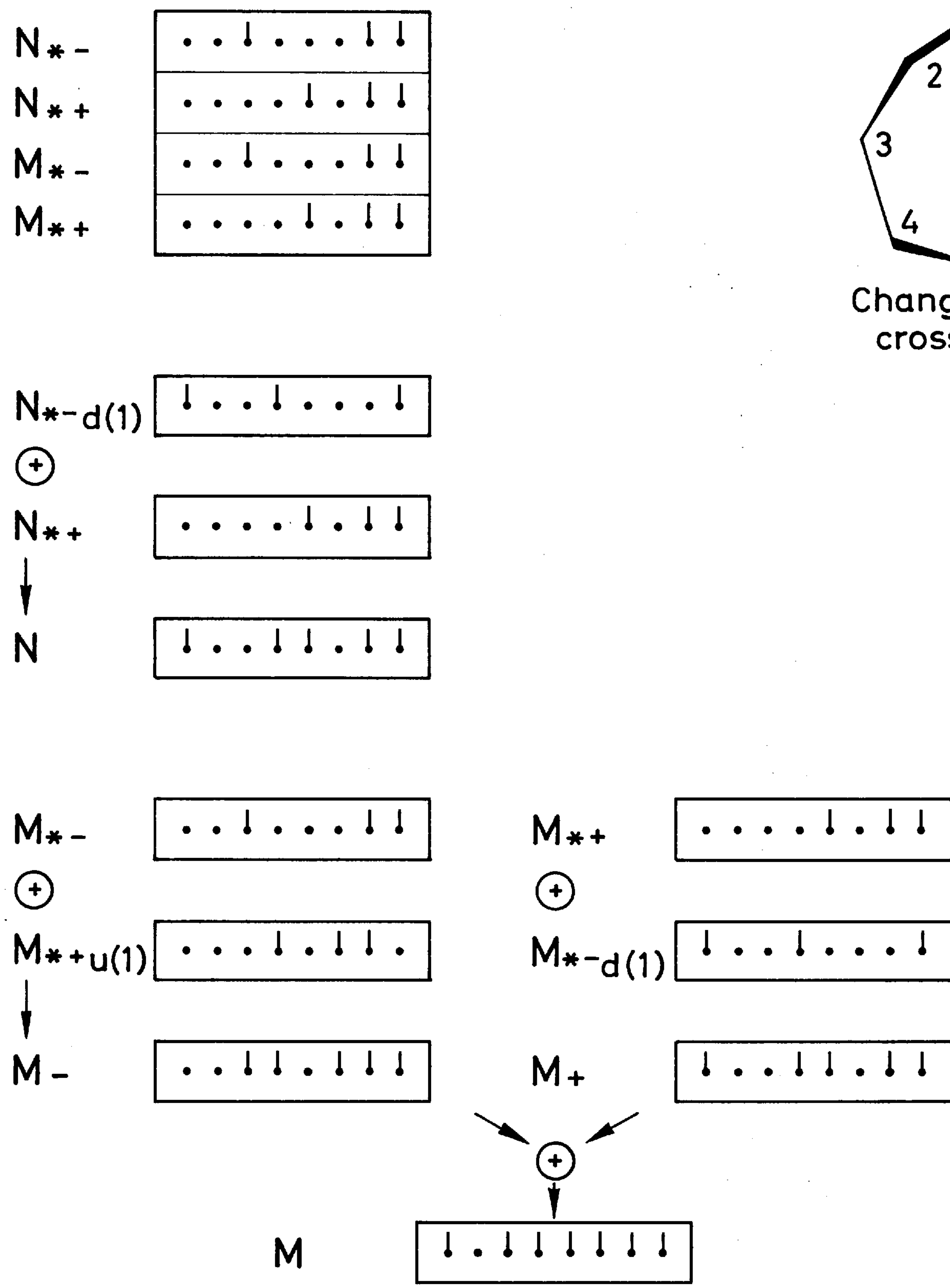
The List L_{MN-N}

Two units corresponding to N_{i-} and N_{i+} are necessary to describe the input of the modified ring areas. In the present list, we need only one unit, since the normal force is specified as constant along each polygon side.

For that purpose we have to introduce the notation if $\mathcal{U}$ is a unit in a list. $\mathcal{U}_{u(y)}$ is the same unit after exercising a round shift in the up (left) direction of y places. Correspondingly $\mathcal{U}_{d(y)}$ is the same unit after exercising a round shift in the down (right) direction of y places. According to this definition we can write

$$\left[L_{MN-N} \right]_i = \left[L_{MN-N*} \right]_{i-\,d(1)} \oplus \left[L_{MN-N*} \right]_{i+} \tag{III,9}$$

An example is given in Fig. (III,20).



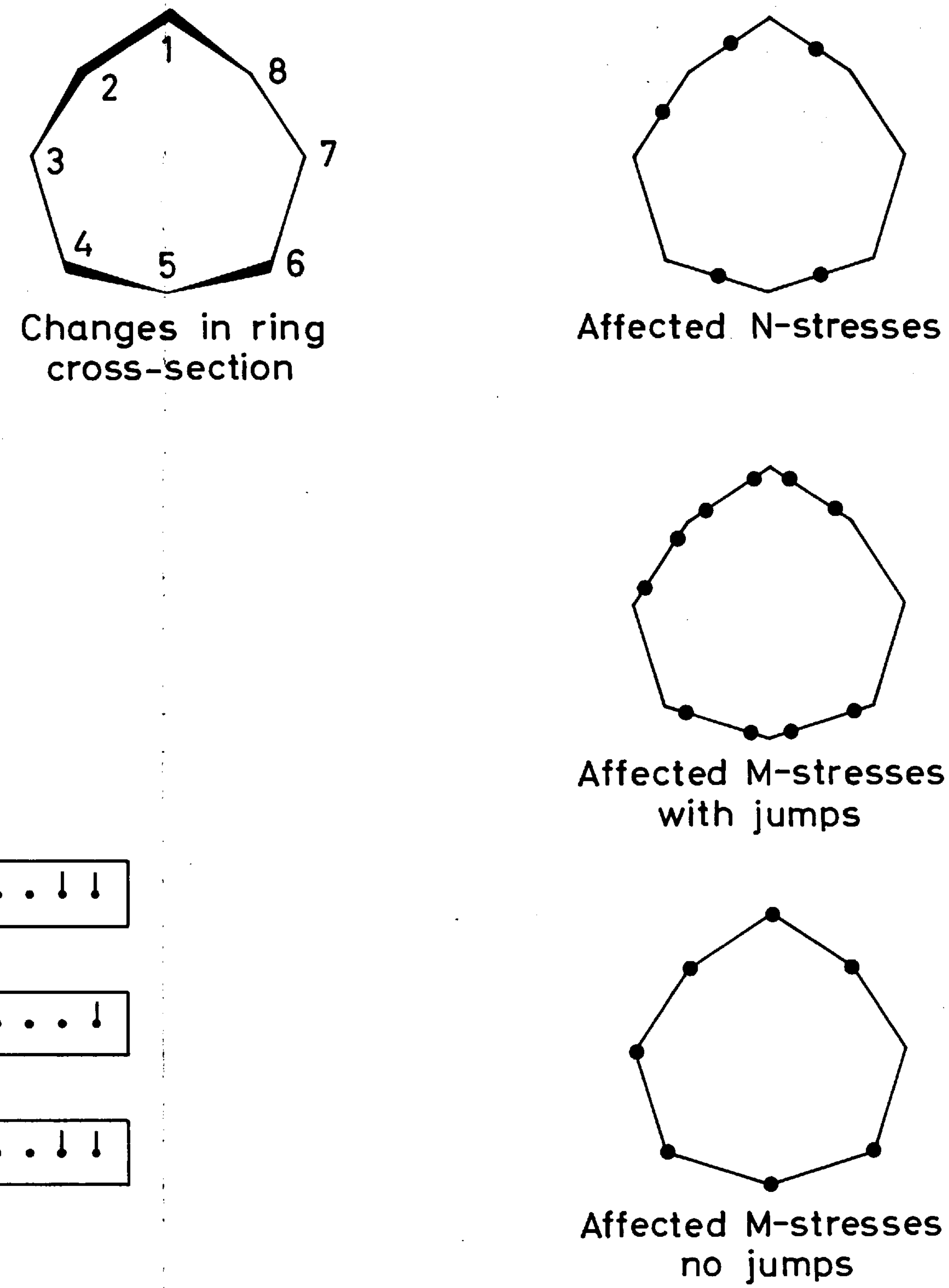


Fig.III-20

Example for the formation of the list L_{MN-r}

The List L_{MN-M}

In our case where we assume no discontinuities in the bending moment, we observe that a change in the cross-section at $(i-1)_+$, $(i)_-$, $(i)_+$ or $(i+1)_-$ will result in a change in the direct flexibility at the vertex i. We need one listword per ring, which we obtain from the relationship

$$\left[L_{MN-M}\right]_i = \left\{\left[L_{MN-M*}\right]_{i_-} \oplus \left[L_{MN-M*}\right]_{i_{+_{u(1)}}}\right\} \oplus \left\{\left[L_{MN-M*}\right]_{i_+} \oplus \left[L_{MN-M*}\right]_{i_{-_{d(1)}}}\right\} \qquad \text{(III,10)}$$

It is, of course, obvious that the actual order in which the four words are taken is immaterial.

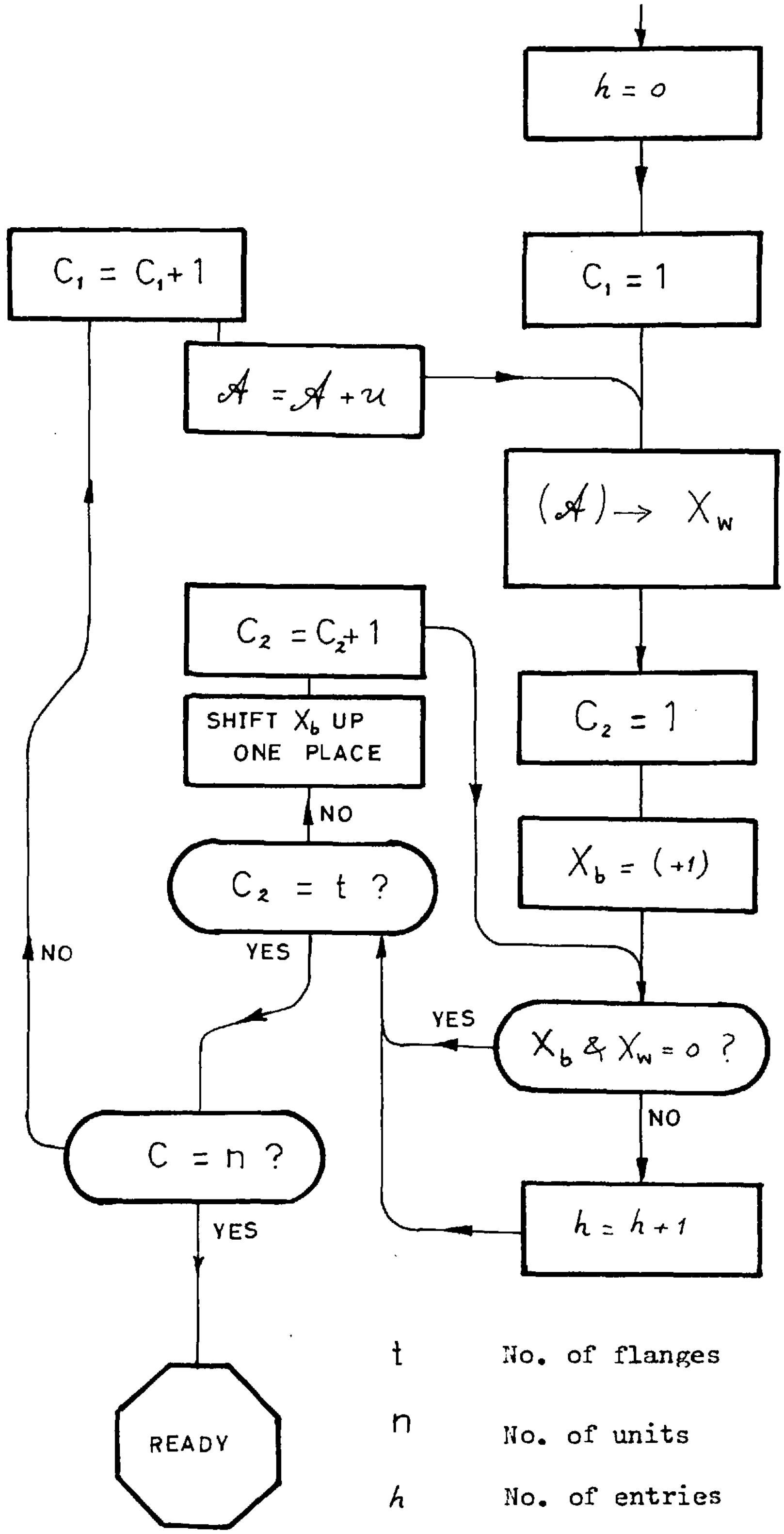
If on the other hand one allows for a discontinuous bending moment, resulting in two rows in the S_M matrix at each station, we then derive the two required words $\left[L_{MN-M}\right]_{i_-}$ and $\left[L_{MN-M}\right]_{i_+}$ from

$$\left[L_{MN-M}\right]_{i_-} = \left[L_{MN-M*}\right]_{i_-} \oplus \left[L_{MN-M*}\right]_{i_{+_{u(1)}}} \qquad \text{(III,11)}$$

$$\left[L_{MN-M}\right]_{i_+} = \left[L_{MN-M*}\right]_{i_+} \oplus \left[L_{MN-M*}\right]_{i_{-_{d(1)}}} \qquad \text{(III,11a)}$$

Counting the Number of Entries

Now, after all lists have been formed, we have to count the number of entries in each list. The basic flow diagram to perform this on a list of units, each consisting of t bits stored at address $\mathcal{A}$, is given in Fig.(III,21). It is straightforward and self-explanatory.



t No. of flanges

n No. of units

h No. of entries

(Fig. III,21)

Flow diagram for counting the number of entries in a list.

Using this diagram, we count the number of entries in all the lists. We denote them by the symbol h , thus

No. of entries in	L_{CO-F}	=	h_{CO-F}
" " " "	L_{CO-S}	=	h_{CO-S}
" " " "	L_{MN-F*}	=	h_{MN-F*}
" " " "	L_{MN-S*}	=	h_{MN-S*}
" " " "	L_{MN-F}	=	h_{MN-F}
" " " "	L_{MN-S}	=	h_{MN-S}
" " " "	L_{CO-N}	=	h_{CO-N}
" " " "	L_{CO-M}	=	h_{CO-M}
" " " "	L_{MN-N*}	=	h_{MN-N*}
" " " "	L_{MN-M*}	=	h_{MN-M*}
" " " "	L_{MN-N}	=	h_{MN-N}
" " " "	L_{MN-M}	=	h_{MN-M}

We denote further

$$
\begin{aligned}
h_{co\text{-}C} &= h_{co\text{-}F} + h_{co\text{-}S} \\
h_{MN\text{-}C*} &= h_{MN\text{-}F*} + h_{MN\text{-}S*} \\
h_{MN\text{-}C} &= h_{MN\text{-}F} + h_{MN\text{-}S} \\
h_{co\text{-}R} &= h_{co\text{-}N} + h_{co\text{-}M} \\
h_{MN\text{-}R*} &= h_{MN\text{-}N*} + h_{MN\text{-}M*} \\
h_{MN\text{-}R} &= h_{MN\text{-}N} + h_{MN\text{-}M}
\end{aligned}
\tag{III,12}
$$

also

$$
\begin{aligned}
h_{co} &= h_{co\text{-}C} + h_{co\text{-}r} \\
h_{MN*} &= h_{MN\text{-}C*} + h_{MN\text{-}r*} \\
h_{MN} &= h_{MN\text{-}C} + h_{MN\text{-}r}
\end{aligned}
\tag{III,13}
$$

and

$$h = h_{co} + h_{MN}$$

<u>The Input of the Data</u>

For this purpose, we have to form a group of Boolean matrices corresponding to the various lists governing the input of data. We name these matrices corresponding to the lists

$$\beta_{MN\text{-}F}\ ,\ \beta_{MN\text{-}S}\ ,\ \beta_{MN\text{-}N}\ ,\ \beta_{MN\text{-}M}$$

and discuss now the information.

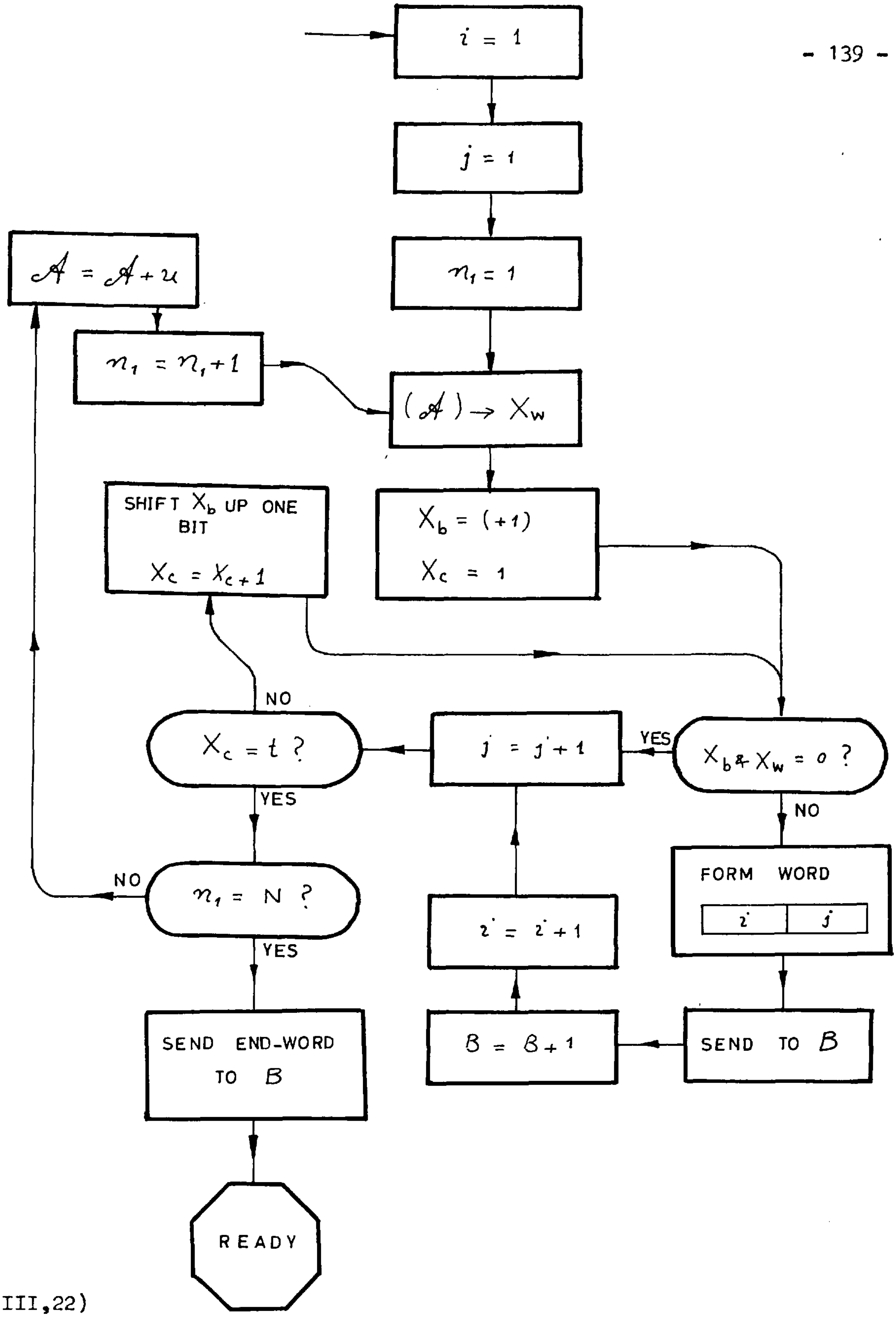


Fig. (III,22)

The formation of a β matrix from a list of N units. The list is stored in $\mathcal{A}$ and the β matrix in $\mathcal{B}$.

The Formation of the Matrix β_{MN-F}

The Boolean matrix β_{MN-F}, of the order $[2p\times 1,\ t\times h_{MN-F*}]$ is derived from the list L_{MN-F*} through the use of the flow diagram shown in Fig.(III-22) followed by repartitioning to suit the matrix B of the flange areas as will be apparent from the following relation. This matrix has the purpose of selecting the new areas and placing them in the right positions, ready to replace the old areas. Thus if all the new areas introduced are fed in in the correct order as the $[h_{MN-F*}\times 1]$ matrix B_{mh}, then we can say that

$$\underset{[2p\times 1,\,t\times 1]}{B_m} = \underset{[2p\times 1,\,t\times 1]}{B} - \underset{[2p\times 1,\,t\times h_{MN-F*}]}{\beta_{MN-F}}\ \beta_{MN-F}^t\ \underset{[2p\times 1,\,t\times 1]}{B} + \beta_{MN-F}\ \underset{[1\times 1,\,h_{MN-F}^*\times 1]}{B_{mh}}$$

$$= \left[I - \beta_{MN-F}\,\beta_{MN-F}^t \right] B + \beta_{MN-F}\, B_{mh} \qquad \text{(III,14)}$$

The new areas of the flanges should be arranged in the matrix B_{mh} so that they all come in the correct order, first of all according to ring stations $1+$ to $p-$ and then within one station from 1 to t.

The Formation of the Matrix β_{MN-S}

This follows directly from the list L_{MN-S*} (or L_{MN-S}) through the use of the same flow diagram (Fig. III,22). The dimensions of this new Boolean matrix are $[(p-1)\times 1\ ,\ t\times h_{MN-S*}]$. Exactly as before, the modified thicknesses are calculated from

$$\underset{[(p-1)\times 1,\, t\times 1]}{t_m} = \Big[\underset{[(p-1),\, t]}{I} - \beta_{MN-S}\,\beta^{t}_{MN-S}\Big] t + \beta_{MN-S}\, t_{mh} \qquad \text{(III,15)}$$

where t_{mh} is the $[h_{MN-S*}\times 1]$ matrix representing the new thicknesses, in the correct order in their occurence in the t .

The Formation of β_{MN-N-} and β_{MN-N+}

The input of the new cross-sectional areas for the normal force is best divided up in conformance with the super-matrices A_- and A_+ , as two matrices A_{mh-} and A_{mh+} . The list L_{MN-N*} is best split into two sub-lists by choosing every second element. Thus starting with the first element we get L_{MN-N*-} and if we begin with the second L_{MN-N*+} . Using those two lists, each consisting of p units, and the flow diagram of Fig.(III,22), we obtain the two Boolean matrices β_{MN-N_-} and β_{MN-N_+} , of the dimensions $[p\times 1\ ,\ t\times h_{MN-N_-}]$ and $[p\times 1\ ,\ t\times h_{MN-N_+}]$ respectively. The number of entries in the sub-lists being of course h_{MN-N_-} and h_{MN-N_+} . In the same manner we get the new ring cross-sectional areas from

$$\underset{[p\times 1,\, t\times 1]}{A_{m-}} = \Big[I - \beta_{MN-N_-}\,\beta^{t}_{MN-N_-}\Big] A_- + \beta_{MN-N_-}\, A_{mh_-} \qquad \text{(III,16)}$$

and

$$\underset{[p\times 1,\, t\times 1]}{A_{m+}} = \Big[I - \beta_{MN-N_+}\,\beta^{t}_{MN-N_+}\Big] A_+ + \beta_{MN-N_+}\, A_{mh_+} \qquad \text{(III,16a)}$$

where again the changes in area are fed in as the two matrices

$$\underset{[h_{MN-N-} \times 1]}{A_{mh-}} \quad \text{and} \quad \underset{[h_{MN-N+} \times 1]}{A_{mh+}}$$

The Formation of β_{MN-M-} and β_{MN-M+}

Following the same logic, the list L_{MN-M*} is split into the sub-lists L_{MN-M*-} and L_{MN-M*+} in which the numbers of entries are given by h_{MN-M*-} and h_{MN-M*+} respectively. It is used to form the matrices β_{MN-M-} and β_{MN-M+} of dimensions $[p \times 1 \;, t \times h_{MN-M-}]$ and $[p \times 1 \;, t \times h_{MN-M+}]$ in accordance with the flow diagram of Fig.(III,22). These two Boolean matrices are not only used to replace the old moments of inertia J, but also the areas effective in shear C, by the new values. This we express as

$$\underset{[p\times 1, t\times 1]}{C_{m-}} = \left[I - \beta_{MN-M-} \beta_{MN-M-}^{t}\right] C_{-} + \beta_{MN-M-} C_{mh-} \qquad \text{(III,17)}$$

$$\underset{[p\times 1, t\times 1]}{C_{m+}} = \left[I - \beta_{MN-M+} \beta_{MN-M+}^{t}\right] C_{+} + \beta_{MN-M+} C_{mh+} \qquad \text{(III,17a)}$$

and

$$\underset{[p\times 1, t\times 1]}{J_{m-}} = \left[I - \beta_{MN-M-} \beta_{MN-M-}^{t}\right] J_{-} + \beta_{MN-M-} J_{mh-} \qquad \text{(III,18)}$$

$$\underset{[p\times 1, t\times 1]}{J_{m+}} = \left[I - \beta_{MN-M+} \beta_{MN-M+}\right] J_{+} + \beta_{MN-M+} J_{mh+} \qquad \text{(III,18a)}$$

The inserted matrices C_{mh-} , C_{mh+} , $\mathfrak{J}_{mh-}$ and $\mathfrak{J}_{mh+}$ have the dimensions

$[h_{MN-M-} \times 1]$, $[h_{MN-M+} \times 1]$, $[h_{MN-M-} \times 1]$

and $[h_{MN-M+} \times 1]$ respectively.

At this stage the new data have been fed in and the new super-matrices of the geometrical properties stored, ready for the calculation of the new flexibilities.

The Final Solutions of the Problem

We establish the super-matrix of the stresses

$$\underset{[\Delta \times \rho]}{S} = \begin{bmatrix} S_c \\ \\ S_R \end{bmatrix} = \begin{bmatrix} S_L \\ S_Q \\ S_N \\ S_M \end{bmatrix} \qquad \text{(III,19)}$$

and the super diagonal matrix of the flexibility

$$\underset{(\Delta \times \Delta)}{f} = \lceil f_L \quad f_Q \quad f_N \quad f_M \rfloor \qquad \text{(III,20)}$$

We also form the new flexibilities

$$\underset{(\Delta \times \Delta)}{f_m} = \lceil f_{Lm} \quad f_{Qm} \quad f_{Nm} \quad f_{Mm} \rfloor \qquad \text{(III,20a)}$$

introducing the new data B_m , t_m , A_{m-} , A_{m+} , C_{m-},

C_{m+}, $\mathfrak{J}_{m-}$ and $\mathfrak{J}_{m+}$.

Then we form the Boolean matrix $\beta^t_{h\,[h \times \Delta]}$, where Δ is the total number of stresses in the fuselage. This matrix can be shown schematically as

$$\underset{[h \times s]}{\beta_h^t} = \begin{array}{c} \left[\begin{array}{c|c} \underset{[h_{co-c} \times 3(p-1)t]}{\beta_{co-c}^t} & \underset{[h_{co-c} \times 2pt]}{0} \\ \hline \underset{[h_{co-r} \times 3(p-1)t]}{0} & \underset{[h_{co-r} \times 2pt]}{\beta_{co-r}^t} \\ \hline \underset{[h_{MN-c} \times 3(p-1)t]}{\beta_{MN-c}^t} & \underset{[h_{MN-c} \times 2pt]}{0} \\ \hline \underset{[h_{MN-r} \times 3(p-1)t]}{0} & \underset{[h_{MN-r} \times 2pt]}{\beta_{MN-r}^t} \end{array}\right] \\ \underbrace{\longleftarrow 3(p-1)t \longrightarrow \longleftarrow 2pt \longrightarrow}_{\longleftarrow\; s \;\longrightarrow} \end{array} \begin{array}{l} h_{co-c} \\ h_{co-r} \\ h_{MN-c} \\ h_{MN-r} \end{array}$$

(III,21)

The formation of the sub-matrices can be accomplished using the flow diagram in Fig.(III,22), thus

one forms β_{co-c} from the list L_{co-c} ,
β_{co-r} from L_{co-r} ,
β_{MN-c} from L_{MN-c}
and β_{MN-r} from L_{MN-r} .

Then, in order to form β_h

$\beta^{(1)}_{co-r}$ is β_{co-r} with the "origin" shifted up through $[3(p-1)t, h_{co-c}]$

$\beta^{(1)}_{MN-c}$ is β_{MN-c} " " " " " " $[0 , h_{co}]$

$\beta^{(1)}_{MN-r}$ is β_{MN-r} " " " " " " $[3(p-1)t, h_{co}+h_{MN-c}]$

Then the matrix

$$\beta_h = \beta_{co-c} + \beta^{(1)}_{co-r} + \beta^{(1)}_{MN-c} + \beta^{(1)}_{MN-r} \qquad \text{(III,22)}$$

is transposed to give the required matrix.
Now we obtain

$$S_h = \beta_h^t \quad S \qquad \text{(III,23)}$$

and

$$b_{1h} = \beta_h^t \quad b_1 \quad ; \qquad \text{(III,24)}$$

two very simple operations.

Similarly if $\beta^{(2)}_{MN-r} = \beta_{MN-r}$ shifted through $[o, h_{co-r}]$,

and
$$\beta_{rh} = \beta_{co-r} + \beta^{(2)}_{MN-r} , \qquad \text{(III,25)}$$

then
$$b_{2rh} = \beta_{rh}^{t} \; b_{2r} . \qquad \text{(III,26)}$$

We also form $\beta^{(3)}_{MN-r}$ as β_{MN-r} shifted through $[3(p-1)t, h_{MN-c}]$,

and hence the matrix

$$\underset{[s \times h_{MN}]}{\beta_{MN}} = \beta_{MN-c} + \beta^{(3)}_{MN-r} \qquad \text{(III,27)}$$

Then we find

$$f_{\Delta h} = \beta_{MN}^{t} \left[f_m - f \right] \beta_{MN} \qquad \text{(III,28)}$$

and can set up the matrix

$$\delta_{III} = \begin{bmatrix} \underset{[h_{co} \times h_{co}]}{0} & \underset{[h_{co} \times h_{MN}]}{0} \\ \underset{[h_{MN} \times h_{co}]}{0} & \underset{[h_{MN} \times h_{MN}]}{f_{\Delta h}^{-1}} \end{bmatrix} \qquad \text{(III,29)}$$

to proceed to the final modified stresses from the basic relations of Ref. (2)

$$S_m = S - \left[b_1 D_{11}^{-1} b_{1h}^{t} + b_2 \; D_{22}^{-1} \; b_{2h}^{t} \right] \left[b_{1h} \; D_{11}^{-1} \; b_{1h} + b_{2h} \; D_{22}^{-1} \; b_{2h}^{t} + \delta_{III} \right]^{-1} S_h$$
(III,30)

or

$$S = \begin{bmatrix} S_{cm} \\ S_{rm} \end{bmatrix} = \begin{bmatrix} S_c \\ S_r \end{bmatrix} - \left[\begin{bmatrix} b_{1c} \\ b_{1r} \end{bmatrix} D_{11}^{-1} \begin{bmatrix} b_{1ch}^t & b_{1rh}^t \end{bmatrix} + \begin{bmatrix} 0_c \\ b_{2r} \end{bmatrix} D_{22r}^{-1} \begin{bmatrix} 0_{ch}^t & b_{2rh}^t \end{bmatrix} \right] \times$$

$$\left[\begin{bmatrix} b_{1ch} \\ b_{1rh} \end{bmatrix} D_{11}^{-1} \begin{bmatrix} b_{1ch}^t & b_{1rh}^t \end{bmatrix} + \begin{bmatrix} 0_{ch} \\ b_{2rh} \end{bmatrix} D_{22r}^{-1} \begin{bmatrix} 0_{ch}^t & b_{2rh}^t \end{bmatrix} + \begin{bmatrix} \delta_{IIIc} & 0 \\ 0 & \delta_{IIIr} \end{bmatrix} \right]^{-1} \begin{bmatrix} S_{ch} \\ S_{rh} \end{bmatrix}$$

(III,31)

For the full automatisation, assuming that we have b_{1rh} from re-partitioning b_{1h}, we recommend the following procedure :-

FORM MODIFIED STRESSES

§ 1 Form $\delta_{I} = b_{1h} \; D_{11}^{-1} \; b_{1h}^t$

§ 2 Form $\delta_{II} = b_{2rh} \; D_{22}^{-1} \; b_{2rh}^t$

§ 3 Form δ_{III} (see previous page)

§ 4 Form $\delta = \delta_{I} + \delta_{II} + \delta_{III}$

§ 5 Form $H = \delta^{-1} \; S_h$

§ 6 Form $D_{oh} = b_{1h}^t \; H$

§ 7 Form $Y_h = - D_{11}^{-1} D_{oh}$

§ 8 Form $S_{cm} = b_c R + b_{1c} Y_h$

§ 9 Form $X_h = - D_{22}^{-1} b_{2rh}^{t} H$

§ 10 Form $S_{rm} = b_r R + b_{1r} Y_h + b_{2r} X_h$

§ 11 Stop

In principle one could add the computed D_{oh} to the original D_o to obtain D_{om} which could then be used instead of D_o to derive the total new primary redundancies. However, for obvious reasons of numerical accuracy, the suggested procedure is considered superior.

We also observe that we could obtain S_h directly, without having to calculate the whole S, by a direct operation of the β matrices on the b_o to get b_{oh} (b_{1h} required in any case) and thus S_h ; hence directly the S_m . However, for checking purposes, it is always recommended to calculate the full S and control compatibility.

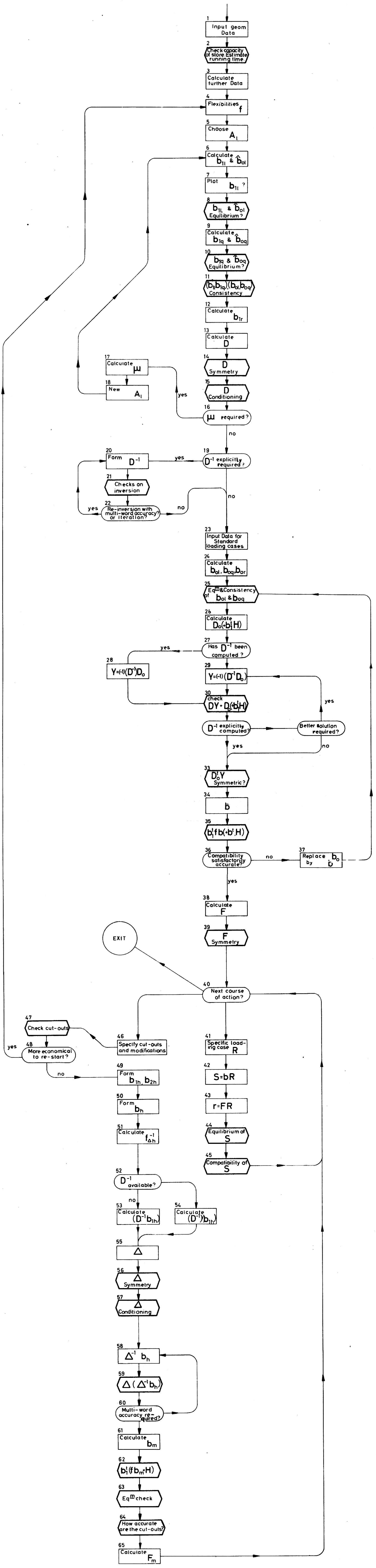


FIG. IV-1. GENERAL LAY OUT OF THE COMPLETE FUSELAGE PROGRAMME

CHAPTER IV

GENERAL LAY OUT OF THE PROGRAMME

INTERMEDIATE AND FINAL CHECKS

Whereas Chapters I and II dealt with the programming of the individual operations involved in the analysis of a fuselage, this chapter deals first of all with the interrelation between the various sub-divisions of the main programme, as well as the necessary checks introduced in the middle, either to ensure the correct functioning of the machine or simply to determine the accuracy to which the computations have been carried out. The sequence of operations is best discussed using the flow diagram (Fig. IV,1), but first we need a preliminary discussion of the checks.

Checks on the Calculation

While running a problem on the computer, one needs to be constantly informed of what is being carried out. This becomes more and more necessary the larger the problem. A programme which runs for several hours must contain checks at intervals, say of half an hour, which result in a certain amount of output. The best ones to construct are those which require a short additional time in comparison with that needed for the main computations. These checks can be either of a structural or mathematical nature, although physical ones are always to be preferred.

Since the sizes of the matrices become larger with the size of the problem, the amount of output must then be so arranged that it can be easily and quickly analysed and understood, so that a certain decision may be made at once as to how the programme is to be directed further. This is achieved either by printing statistical data about the matrices or by extracting the results in a more comprehensible form. e.g. graphically.

The Automatic Programme and Intermediate Checks

The programme starts (1) with the input of the basic data required. The first part of this data is used for organisational purposes, to determine the size of the different matrices, the necessary storage space etc. as well as some data, controlling decisions, which can be made automatically in the programme, for instance, the accuracy considered to be sufficient in the solution of the system of equations and the number of standard loading cases. Other control data are the expected number of modifications and cut-outs(to be specified later more precisely) and so on. A check (2) is then carried out to ensure that the problem is not too large for the machine. One limitation is the amount of storage space available, which the computer can check automatically, and interrupt the calculation if exceeded, or give a warning if the size approaches the critical. The other limitation is the calculation time required. A certain amount of manual control is here allowed to enable the operator to interrupt or continue as desired.

The second part are the basic geometrical data which are also used to compute all secondary dimensions(3) such as polygon side lengths and trigonometrical values of the inclinations. This performed, the machine proceeds to calculate the matrix $\mathbf{f}$, of flexibilities of the individual elements (4).

Having established this, one is ready to proceed with the automatic generation of the self-equilibrating systems. First of all $\mathbf{A}_\ell$ has to be chosen (5). In view of the excellent results obtained by using the Fourier coefficients for a circular fuselage $(\mathbf{\Omega}_\ell)$, it is recommended that this should be programmed automatically. A certain manual control must be provided, however, to substitute a different $\mathbf{A}_\ell$ if necessary; either by entering another sub-programme, or by direct input. The facility for the improvement of $\mathbf{A}_\ell$ through orthogonalisation is provided through (16) to (18) with an iteration back to (6).

Box no. (6) represents the determination of the self-equilibrating stress systems in the flanges $\mathbf{b}_{1\ell}$ and the accompanying statically equivalent basic systems $\hat{\mathbf{b}}_{o\ell}$. After having obtained this, a means is provided in (7) to plot the matrix $\mathbf{b}_{1\ell}$ through the order described in Appendix (C). This gives an idea of the linear independence of the systems. Similar facilities have not been included in the flow diagram for plotting $\mathbf{b}_{1q}$ and $\mathbf{b}_{1r}$ as well, however, these can also be included.

In (8) the resulting $\mathbf{b}_{1\ell}$ and $\hat{\mathbf{b}}_{o\ell}$ are checked for equilibrium. The programme is straightforward, and can be directly given here, using the standard order forms assumed in Appendix (C).

CHECK EQUILIBRIUM OF $\mathbf{b}_{1\ell}$

```
(SALT, 2 (p-1) /, 3 x t) x (B1L, 2 (p-1) x (p-2), t x (t-3))  ——→ WS1
MKMAX1 (SALT, 2(p-1 /, 3 x t) x B1L, 2 (p-1) x(p-2), t x (t-3)) ——→ WS2
DIVEL (WS2, 2(p-1) x (p-2), 3 x (t-3)) (WS1,2(p-1) x (p-2), 3 x (t-3))  ——→ WS3
COSPEC (WS3,2 (p-1) x (p-2), 3 x (t-3)) (10^-20, 1) (10^+1)
```

That is to say, the multiplication $\mathbf{a}_\ell^t\ \mathbf{b}_{1\ell}$ is first carried out. The result should be zero. Since, however, this zero is a relative one, we must compare it with another matrix which we build by the special matrix function "MKMAX" described in Appendix (C). This effectively multiplies the rows of $\mathbf{a}_\ell^t$ with the columns of $\mathbf{b}_{1\ell}$, taking as the result of each such multiplication not the sum of the elemental products, but the maximum product encountered.

This accomplished we divide the elements of this new matrix into those of the other one, giving us directly the accuracy of the equilibrium, by printing out the column spectrum of the result between, say 10^{-20} and 1. If we want even less to be printed out, we can use the following order instead of the last

```
MASPEC (WS3, 2(p-1) x (p-2), 3 x (t-3)) (10^-20,1) (10^+1)
```

Similarly we check the equilibrium of the $\hat{b}_{o\ell}$ matrix

CHECK EQUILIBRIUM OF BoLC

(SALT, 2 (p-1) /, 3 x t) x (BoLC, 2 (p-1) /, t x 3) ⟶ WS1
CL (UNMAT, 2 (p-1) /, 3) ⟶ UNMAT
(UNMAT, 2 (P-1) /, 3) + (ONE) ⟶ UNMAT
(WS1, 2 (p-1) /, 3) - (UNMAT, 2 (p-1) /, 3) ⟶ WS2
MKMAX (SALT, 2 (p-1) /, 3 x t)(BoLC, 2 (p-1) /, t x 3) ⟶ WS3
DIVEL (WS3, 2 (p-1) /, 3) (WS2, 2 (p-1) /, 3) ⟶ WS4
MASPEC (WES4, 2 (p-1) /, 3) (10^{-20}, 1) (10)

Here our error matrix is $\left[a_\ell^t \hat{b}_{o\ell} - I_{[2(p-1),3]} \right]$ and this is again compared with the maximum elements occuring during the multiplication of the first term $a_\ell^t \hat{b}_{o\ell}$.

Having checked the above, the computer then procedes to calculate b_{1q} and $\hat{b}_{oq}$ (9). Afterwards a similar check on equilibrium is provided (10). This is done by using one of the two expressions (say for b_{1q}).

$$a_{q+}^t \quad b_{1q} \quad + \quad a_{T+}^t \quad b_{1\ell} \quad = \quad 0 \qquad \text{(IV,1)}$$

or

$$a_{q-}^t \quad b_{1q} \quad + \quad a_{T-}^t \quad b_{1\ell} \quad = \quad 0 \qquad \text{(IV,1a)}$$

However, in trying to obtain the base line with which we compare the zeros, we need only to consider the left hand term, since the second one is of secondary importance in comparison.

The same applies for the $\hat{b}_{oq}$ matrix, although we have as before to subtract a unit matrix to get the error matrix. Thus we verify the following relations

$$\left[a_{q+}^t \quad \hat{b}_{oq} \quad + \quad a_{T+}^t \quad \hat{b}_{o\ell}(\varepsilon_{\ell q}) \right] \bar{\varepsilon}_{\ell -} - I_{[(p-1),3]} = 0 \qquad \text{(IV,2)}$$

and

$$\left[a_{q-}^t \quad \hat{b}_{oq} \quad + \quad a_{T-}^t \quad \hat{b}_{o\ell}(\varepsilon_{\ell q}) \right] \bar{\varepsilon}_{\ell +} - I_{[(p-1),3]} = 0 \qquad \text{(IV,2a)}$$

where

$$\mathcal{E}_{\ell q} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \text{(IV,2b)}$$

The programmes can then be written.

EQUILIBRIUM OF B1Q and BoQC

```
(AQPT, (p-1) /, 3 x t) x (B1Q, (p-1) x (p-2), t x (t-3))                ──→ WS1
(ATPT, (p-1) x 2 (p-1), 3 x t) x (B1L, 2 (p-1) x (p-2), t x (t-3))      ──→ WS2
(WS1, (p-1) x (p-2), 3 x (t-3)) + (WS2, (p-1) x (p-2), 3 x (t-3))       ──→ WS1
MKMAX (AQPT, (p-1) /, 3 x t) x (B1Q, (p-1) x (p-2), t x (t-3))          ──→ WS2
DIVEL (WS2, (p-1) x (p-2), 3 x (t-3)) (WS1, (p-1) x (p-2), 3 x (t-3))   ──→ WS3
MASPEC (WS3,(p-1) x (p-2), 3 x (t-3)) (10^-20, 1) (10)

(AQMT, (p-1) /, 3 x t) x (B1Q, (p-1) x (p-2), t x (t-3))                ──→ WS1
(ATMT, (p-1) x 2 (p-1), 3 x t) x (B1L,2(p-1) x (p-2), t x (t-3))        ──→ WS2
(WS1, (p-1) x (p-2), 3 x (t-3)) + (WS2, (p-1) x (p-2), 3 x (t-3))       ──→ WS1
MKMAX (AQPT, (p-1) /, 3 x t) x (B1Q, (p-1) x (p-2), t x (t-3))          ──→ WS2
DIVEL(WS2, (p-1) x (p-2), 3 x (t-3))(WS1, (p-1) x (p-2), 3 x (t-3))     ──→ WS3
MASPEC (WS3, (p-1)x (p-2), 3 x (t-3)) (10^-20,1) (10)

Gb (2,1) (1,1) (2)                                                      ──→ ELQ
(bOQC, (p-1) x 2 (p-1), t x 3) x (bEBLM, 2 (p-1) x (p-1), ( ))          ──→ BoQCP
(BoLC, 2 (p-1) /, t x 3) x (bELQ, ( ), 3)                               ──→ BoLCC
(BoLCC, 2 (p-1) /, t x 3) x (bEBLM, 2 (p-1) x p-1), (  ))               ──→ BoLCCP
(AQPT, (p-1) /, 3 x t) x (BoQCP, (p-1), t x 3)                          ──→ WS1
(ATPT, (p-1) x 2 (p-1), 3 x t) x (BoLCCP, 2 (p-1) x (p-1), t x 3)       ──→ WS2
(WS1, (p-1), 3) + (WS2, (p-1), 3)                                       ──→ WS1
(WS1, (p-1), 3) - (ONE)                                                 ──→ WS1
MASPEC (WS1,(p-1), 3) (10^-20, 1) (10)
```

(BoQC, (p-1) x 2 (p-1), t x 3) x (bEBLP, 2 (p-1) x (p-1), ()) $\longrightarrow$ boQCM
(BoLCC, 2 (p-1) /, t x 3) x (bEBLP, 2 (p-1) x (p-1), ( )) $\longrightarrow$ BoLCCM
(AQMT, (p-1) /, 3 x t) x (BoQCM, (p-1), t x 3) $\longrightarrow$ WS1
(ATMT, (p-1) x 2 (p-1), 3 x t) x (BoLCCM, 2 (p-1)x (p-1), t x 3) $\longrightarrow$ WS2
(WS1, (p-1), 3) + (WS2, (p-1), 3) $\longrightarrow$ WS1
(WS1, (p-1), 3) - (ONE) $\longrightarrow$ WS1
MASPEC (WS1, (p-1),, 3) (10^{-20}, 1) (10)

Having checked that, the computer proceeds to control the consistancy of the flange loads and shear flows. This we can do for the self-equilibrating stress system using the identity

$$\underset{[(\,),t]}{\alpha} \quad b_{1q} - \left[\bar{\varepsilon}^{t}_{\ell+} - \bar{\varepsilon}^{t}_{\ell-} \right] b_{1\ell} = 0 \qquad \text{(IV,3)}$$

To compare our zero to a base line, we can take as reference the moduli of the elements of b_{1q} .

CHECK CONSISTANCY OF B1L, B1Q

TR (bEBLP, 2 (p-1) x(p-1), ( )) $\longrightarrow$ EBLPT
TR (bEBLM, 2 (p-1) x (p-1), ( )) $\longrightarrow$ EBLMT
(bEBLPT, (p-1) x 2 (p - 1), ( )) - (bEBLMT, (p-1) x 2 (p-1),( )) $\longrightarrow$ LQOPER
(bLQOPER, (p-1) x 2 (p-1), ( )) x (B1L, 2 (p-1) x (p-2), t x (t-3)) $\longrightarrow$ WS2
(bALFT, ( ), t) x (B1Q, (p-1) x (p -2), t x (t - 3)) $\longrightarrow$ WS1
MODEL (WS1, (p-1) x (p-2), t x (t-3)) $\longrightarrow$ ERMOD
(WS1, (p-1) x (p-2), t x (t-3)) - (WS2, (p-1) x (p-2), t x (t-3)) $\longrightarrow$ LQER
DIVEL (ERMOD, (p-1) x (p-2), t x (t-3)) (LQER, (p-1) x (p-2), t x(t-3)) $\rightarrow$ RELER
MASPEC (RELER, (p-1) x (p-2), t x (t-3))(10^{-20}, 1) (10)

Following the previous argument, we check the consistency of $\hat{b}_{oq}$ and $\hat{b}_{o\ell}$ through

$$\left[\underset{[\;\;,t]}{\alpha} \quad \hat{b}_{oq} \; - \left[\bar{\varepsilon}^{t}_{\ell_+} - \bar{\varepsilon}^{t}_{\ell_-}\right] \hat{b}_{o\ell}\left(\varepsilon_{\ell q}\right)\right] \bar{\varepsilon}_{\ell_-} = 0 \qquad \text{(IV,4)}$$

and

$$\left[\underset{[\;\;,t]}{\alpha} \quad \hat{b}_{oq} \; - \left[\bar{\varepsilon}^{t}_{\ell_+} - \bar{\varepsilon}^{t}_{\ell_-}\right] \hat{b}_{o\ell}\left(\varepsilon_{\ell q}\right)\right] \bar{\varepsilon}_{\ell_+} = 0 \qquad \text{(IV,4a)}$$

<u>CHECK CONSISTANCY OF BoLC, BoQC</u>

(bALFT, ( ),t) x (BoQCP, (p-1), t x 3) ⟶ WS1
MODEL (WS1, (p-1), t x 3) ⟶ MODER
(bOPER, (p-1) x 2 (p - 1), ( )) x (BoLCCP, 2 (p-1) x (p-1), t x 3) ⟶ WS2
(WS1, (p-1), t x 3) -(WS2, (p-1), t x 3) ⟶ LQoER
DIVEL (MODER, (p-1), t x 3) (LQoER, (p-1), t x 3) ⟶ RELER
MASPEG (RELER, (p-1), t x 3) (10^{-20}, 1) (10)

(bALFT, ( ), t) x (BoQCM, (p-1), t x 3) ⟶ WS1
MODEL (WS1, (p-1), t x 3) ⟶ MODER
(bOPER, (p-1) x 2 (p-1), ( )) x (BoLCCM, 2 (p-1) x (p-1), t x 3) ⟶ WS2
(WS1, (p-1), t x 3) - (WS2, (p-1), t x 3) ⟶ LQoER
DIVEL (MODER, (p-1), t x 3) (LQoER, (p-1), t x 3) ⟶ RELER
MASPEC (RELER, (p-1), t x 3) (10^{-20}, 1) (10)

With these controls accomplished, the machine proceeds to calculate the self-equlibrating stress systems in the rings (12). This part of the calculation can also be checked although it has not been explicitly shown in the flow diagram. The equilibrium inside the closed rings, that is the consistancy of b_{IN}, b_{IS} and b_{IM}, as well as the accuracy to which the compatability has been satisfied in their closure, are then examined.

The formation of D (13) is followed immediately by two checks (14) and (15) to analyse the result. The first is on the symmetry of the matrix. This can be done in many ways, varying in speed and simplicity. The quickest way is to form

$$\left[\left[e^t D\right]^t - D e\right] = 0 \qquad \text{(IV,5)}$$

and compare it with the matrix composed of the largest elements in the summations involved in the matrix multiplication. The programme for this is

<u>CHECK SYMMETRY OF D</u>

```
Gb (1,1) (1,0) ((p-2) x (t-3))                                          ——→ BEFD
(ONE) x (bBEFD, (p-2) x 1, (t-3) x 1)                                    ——→ EFD
(D, (p-2), (t-3)) x (EFD, (p-2)x 1, (t-3) x 1)                           ——→ DEL 1
MKMAX (D, (p-2), (t-3)) x (EFD, (p-2) x 1, (t-3) x 1)                    ——→ ZERBAS
TR (EFD, (p-2) x 1, (t-3) x 1)                                           ——→ EFDT
(EFDT, 1 x (p-2), 1 x (t - 3)) x (D, (p-2), (t-3))                       ——→ DEL2T
TR (DEL2T, 1 x (p-2), 1 x (t-3))                                         ——→ DEL2
(DEL2, (p-2) x 1, (t-3) x 1) - (DEL1, (p-2) x 1, (t-3) x 1)              ——→ DEL
DIVEL (ZERBAS, (p-2) x 1, (t-3) x 1) (DEL, (p-2) x 1, (t-3) x 1)         ——→ RELZER
MASPEC (RELZER, (p-2) x 1, (t-3) (10^-20, 1) (10)
```

We might also mention here that a vector other than e can be used if desired.

Another more "precise" method is to compare directly D and D^t. In other words to form

$$\left[D - D^t\right]$$

and compare it with the original D . This method, however, shows little advantage.

The conditioning of the D matrix can be checked by various methods (see Chapter V , but in our experience a very simple method helps enormously to point out trouble at an early stage, i.e. through the use of the order for diagonal normalization described in Appendix (C) followed by taking a matrix spectrum between 1.0 and say 0.05.

RATIO OF ELEMENTS IN D

DIANOR (D, (p-2), (t-3) $\longrightarrow$ DNORM
MASPEC (DNORM, (p-2), (t-3)) (1, 0, 05) (0.9)

According to the results of this test (manual intervention being here also allowed) a decision is made (16) as to whether the conditioning is satisfactory, or whether in (17) and (18) the diagonalization technique should be used.

This decision can already be made at an earlier stage. For instance, only the diagonal sub-matrices of the D_r need first be calculated and examined, and the technique can be used accordingly.

Whenever this orthogonalization technique is required, however, it is to be recommended that it be done right through the fuselage for the sake of simplicity and to secure the best results.

The programme for the steps (17) and (18) can be given as follows

ORTHOGONALIZE DR.

EXDISM (DR, (p-2), (t-3)) ⟶ DRDIAG
EIGV (DRDIAG, (p-2) /, (t-3)) ⟶ DRVAL, CHOW
(ALM, (p-2) /, t x (t - 3)) x (CHOW, (p-2) /, (t-3)) ⟶ ALMB
SD (ALMB, (p-2) /, t x (t-3)) (1, (p-3)) ⟶ ALB2, ALB2C
SD (ALB2C, (p-3) /, t x (t-3)) ((p-4), 1) ⟶ ALBMME,ALBPMO
JD (ALB2, 1/, t x (t-3)) (ALMB, (p-2) /, t x (t-3)) ⟶ ALMMB
JD (ALMMB, (p-1) /, t x (t-3)) (ALBPMO, 1/, t x (t-3)) ⟶ ALB
(ALB,p/, t x (t-3)) ⟶ AL

If the D matrix is finally considered to be satisfactorily conditioned, or if no further improvement can be obtained, the machine proceeds to (19), where it decides whether D^{-1} is explicitely required, or if a direct inversion into D_0 be more advantageous. This decision might be made manually, or by a criterion set up in the machine, based on the size of the problem, the number of loading cases, the number of expected modifications, and the number of times this has to be done. Reasons of accuracy may also be involved.

If the machine decides, or is instructed that the D^{-1} is to be formed, it proceeds to (20), and then to (21) where the inversion is checked. The first check is on symmetry. This is done in exactly the same manner as with the D matrix. The second possible one is whether D^{-1} is the true inverse of D, and if it is, to what accuracy.

Mathematically, it is said to be best to carry out both multiplications $D \; D^{-1}$ and $D^{-1} \; D$ and compare the results with a unit matrix. This, however, is a lengthy operation, and contradicts one of the principle conditions of a check, namely that it should take much less time than the main operations. A satisfactory and quick control is in our opinion the pre-multiplication of a unit vector e with the matrix D and then its inverse D^{-1}, or vice-versa, and to compare the results with the unit matrix vector. Thus in effect forming the two expressions

$$\left[D \; D^{-1} e - e \right] \quad \text{and} \quad \left[D^{-1} \; D \; e - e \right]$$

The programme is simple and can be given immediately

QUICK CHECK ON INVERSION

(DMO, (p-2), (t-3)) x (EFD, (p-2) x 1, (t-3) x 1) $\longrightarrow$ DMOE
(D, (p-2), (t-3)) x (DMOE, (p-2) x 1, (t-3) x 1) $\longrightarrow$ DDMOE
(EFD, (p-2) x 1, (t-3) x 1) - (DDMOE, (p-2) x 1, (t-3) x 1) $\longrightarrow$ ERR
MASPEC (ERR, (p-2) x 1, (t-3) x 1) (10^{-20}, 1) (10)

(D, (p-2), (t-3)) x (EFD, (p-2) x 1, (t-3) x 1) $\longrightarrow$ DE
(DMO, (p-2), (t-3)) x(DE, (p-2) x 1 (t-3) x 1) $\longrightarrow$ DMODE
(EFD, (p-2) x 1, (t-3) x 1) - (DMODE, (p-2) x 1,(t-3 x 1) $\longrightarrow$ ERR
MASPEC (ERR, (p-2) x 1, (t-3) x 1)(10^{-20}, 1) (10)

This done, the machine checks (22), possibly under outside control, whether the results obtained are satisfactory, or whether the inversion has to be repeated with higher precision.

Having decided that the inversion fulfils the desired accuracy, further data describing the standard loading cases are called for and fed in (23). Accordingly the basic systems in the cover ($\mathbf{b}_{o\ell}$ and $\mathbf{b}_{oq}$) are calculated (24) and then the same checks applied to them as on the cover stresses before, namely equilibrium and consistancy. The procedure is similar to the previous one and is based on the following relations: -
Equilibrium of flange loads:

$$\underset{[2(p-1)/,\,3\times t]}{\mathbf{a}_{\ell}^{t}} \quad \underset{[2(p-1)\times 1,\,t\times \rho]}{\mathbf{b}_{o\ell}} = \underset{[2(p-1)\times 1,\,3\times \rho]}{\hat{\mathbf{R}}_{\ell}} \qquad \text{(IV,6)}$$

Equilibrium of shear panels:

$$\underset{[2(p-1)\times(p-1),\,1]}{\bar{\boldsymbol{\varepsilon}}_{\ell-}} \left[\underset{[(p-1)/,\,3\times t]}{\mathbf{a}_{q+}^{t}} \quad \underset{[(p-1)\times 1,\,t\times\rho]}{\mathbf{b}_{oq}} + \underset{[(p-1)\times 2(p-1),\,3\times t]}{\mathbf{a}_{T+}^{t}} \quad \underset{[2(p-1)\times 1,\,t\times\rho]}{\mathbf{b}_{o\ell}} \right]$$
$$+ \underset{[2(p-1)\times(p-1),\,1]}{\bar{\boldsymbol{\varepsilon}}_{\ell+}} \left[\underset{[(p-1)/,\,3\times t]}{\mathbf{a}_{q-}^{t}} \quad \underset{[(p-1)\times 1,\,t\times\rho]}{\mathbf{b}_{oq}} + \underset{[(p-1)\times 2(p-1),\,3\times t]}{\mathbf{a}_{T}^{t}} \quad \underset{[2(p-1)\times 1,\,t\times\rho]}{\mathbf{b}_{o\ell}} \right] = \underset{[2(p-1)\times 1,\,3\times\rho]}{\hat{\mathbf{R}}_{q}}$$

(IV,7)

or better still

$$\left[\bar{\varepsilon}_{\ell_-} \ a_{q_+}^{t} + \ \bar{\varepsilon}_{\ell_+} \ a_{q_-}^{t}\right] b_{oq} + \left[\bar{\varepsilon}_{\ell_-} \ a_{T_+}^{t} + \ \bar{\varepsilon}_{\ell_+} \ a_{T_-}^{t}\right] b_{o\ell} = \hat{R}_q \qquad (IV,7a)$$

and for the consistency of $b_{o\ell}$ and b_{oq}

$$\underset{[(\ ),\ t]}{\alpha} \quad \underset{[(p-1)\times 1,\ t\times\rho]}{b_{oq}} - \underset{[(p-1)\times 2(p-1),\ (\ )]}{\left[\bar{\varepsilon}_{\ell_+}^{t} - \bar{\varepsilon}_{\ell_-}^{t}\right]} \quad \underset{[2(p-1)\times 1,\ t\times\rho]}{b_{o\ell}} = \underset{[(p-1)\times 1,\ t\times\rho]}{0} \qquad (IV,8)$$

The programmes read as follows:

CHECK EQUILIBRIUM OF BoL

(SALT,2(p-1)/, 3 x t) x (BoL,2 (p-1) x 1, t x ρ) ⟶ RESTNT
MKMAX (SALT, 2(p-1)/, 3 x t) x (BoL, 2 (p-1) x 1, t x ρ) ⟶ BASLIN
RESTNT, 2(p-1) x 1, 3 x ρ) - (RLL, 2 (p-1) x 1, 3 x ρ) ⟶ ABZERO
DIVEL (BASLIN,2 (p-1) x 1, 3 x ρ)(ABZERO, 2 (p-1) x 1, 3 x ρ) ⟶ RELZER
MASPEC (RELZER, 2 (p-1) x 1, 3 x ρ) (10^{-20}, 1) (10)

CHECK EQUILIBRIUM OF BoQ

(bEBLM, 2(p-1) x (p-1), ( )) x AQPT, (p-1)/, 3 x t) ⟶ QOP1
(bEBLP, 2 (p-1) x (p-1), ( )) x AQMT, (p-1)/, 3 x t) ⟶ QOP2
(QOP1, 2 (p-1) x (p-1), 3 x t) + (QOP2, 2(p-1) x (p-1), 3 x t) ⟶ QOP
(bEBLM, 2 (p-1) x (p-1), ( )) x (ATPT, (p-1) x 2 (p-1), 3 x t) ⟶ POP1
(bEBLP, 2(p-1) x (p-1), ( )) x (ATMT, (p-1) x 2 (p-1) x 3x t) ⟶ POP2
(POP1,2 (p-1), 3 x t) + (POP2, 2(p-1), 3 x t) ⟶ POP
(QOP, 2(p-1) x (p-1), 3 x t) x (BoQ, (p-1) x 1, t x ρ) ⟶ RSTNT1
(POP, 2 (p-1), 3 x t) x (BoL, 2 (p-1) x 1, t x ρ) ⟶ RSTNT2
(RSTNT1, 2 (p-1) x 1, 3 x ρ) + (RSTNT2, 2 (p-1) x 1, 3 x ρ) ⟶ RSTNT
(RSTNT, 2 (p-1) x 1, 3 x ρ) - (RQL, 2 (p-1) x 1, 3 x ρ) ⟶ ABZERO
MoDEL (RQL, 2 (p-1) x 1, 3 x ρ) ⟶ RQLMOD
DIVEL (RQLMOD, 2 (p-1) x 1, 3 x ρ) (ABZERO, 2 (p-1) x 1, 3 x ρ) ⟶ RELZER
MASPEC (RELZER, 2 (p-1) x 1, 3 x ρ) (10^{-20}, 1) (10)

CHECK CONSISTANCY OF BoL and BoQ

(bALFT, ( ), t) x (BoQ, (p-1) x 1, t x ρ) ⟶ RES1
(bLQOPER, (p-1) x 2 (p-1), ( )) x (BoL, (p-1) x 1, t x ρ) ⟶ RES2
(RES1, (p-1) x 1, t x ρ) - (RES2, (p-1) x 1, t x ρ) ⟶ RES
MODEL (RES2, (p-1) x 1, t x ρ) ⟶ BASLIN
DIVEL (BASLIN, (p-1) x 1, t x ρ) (RES, (p-1) x 1, t x ρ) ⟶ RELZER
MASPEC (RELZER, (p-1) x 1, t x ρ)(10^{-20}, 1) (10)

Following the determination of the strains in the basic system, the total incompatible strains $\mathbf{D}_o (+ \mathbf{b}_1^t \mathbf{H})$ are calculated (26), and then the machine checks if the $\mathbf{D}^{-1}$ has been previously determined, or if a direct solution is necessary. Accordingly it chooses either operation (28) (direct multiplication) or (29) (direct inversion) to obtain the primary redundancies $\mathbf{Y}$. A check is again performed (30) to determine the validity and accuracy of the solution. This is best verified using the identity

$$\mathbf{D}\,\mathbf{Y} + \mathbf{D}_o = 0 \tag{IV,9}$$

The zero here is, naturally enough, compared with $\mathbf{D}_o$. The programme for that is again straightforward

CHECK SOLUTION OF EQUATIONS

(D, (p-2), (t-3)) x (WY, (p-2) x 1, (t-3) x ρ) ⟶ NDN
MODEL (NDN, (p-2) x 1, (t-3) x ρ) ⟶ BASE
(NDN, (p-2) x 1, (t-3) x ρ) + (WY, (p-2) x 1, (t-3) x ρ) ⟶ ABZ
DIVEL (BASE, (p-2) x 1, (t-3) x ρ) (ABZ, (p-2) x 1, (t-3) x ρ) ⟶ RELZ
COSPEC (RELZ, (p-2) x 1, (t-3) x ρ) (10^{-20}, 1) (10)

According to the results of this check a decision is made, perhaps by the operator, as to whether a re-calculation is needed using higher precision (32). This is only carried out if a direct solution has taken place, since if the $\mathbf{D}^{-1}$ has already been computed, measures would already have been taken to improve the accuracy (22).

A further possible check is to form $D_o^t\ Y$ and then to examine its symmetry. This is actually a further proof that the solution of the equation is correct as well as that the D matrix has been symmetrically formed. The programme for that is as follows

CHECK ON Y AND D

TR (Do, (p-2) x 1, (t-3) x ρ)	$\longrightarrow$	DOT
(DoT, 1 x (p-2), ρ x (t-3) x (WY, (p-2) x 1, (t-3) x ρ)	$\longrightarrow$	MTBCHK
Gb (1,1) (1,0) (ρ)	$\longrightarrow$	ERO
(ONE) x (bERO, 1, ρ x 1)	$\longrightarrow$	EROD
(MTBCHK, 1, ρ) x (EROD, 1, ρ x 1)	$\longrightarrow$	CSUM
TR (EROD, 1, ρ x 1)	$\longrightarrow$	ERODT
(ERODT, 1, 1 xρ) x (MTBCHR, 1, ρ)	$\longrightarrow$	RSUMT
TR (RSUMT, 1, 1 x ρ)	$\longrightarrow$	RSUM
(RSUM, 1, ρ x 1) - (CSUM, 1, ρ x 1)	$\longrightarrow$	SUMDIF
MODEL (CSUM, 1, ρ x 1)	$\longrightarrow$	BASEL
DIVEL (BASEL, 1, ρ x 1) (SUMDIF, 1, ρ x 1)	$\longrightarrow$	RELZ
MASPEC (RELZ, 1, ρ x 1) (10^{-20}, 1) (10)		

After this check, the first stress distribution is calculated (34) and followed immediately by the compatibility check (35) which is again printed out as statistical information on the columns in order to keep the different loading cases separate. The machine then decides, (or is instructed, (36)) whether to perform a further iteration, i.e. use the final solution as a basic system (37) and repeat steps (25) - (36) (see Chapter V), or to proceed to calculate the flexibility matrix F (38), and follow it by a check for symmetry similar to that previously mentioned.

Block no. (40) contains information on the purpose of the programme. If cut-outs or modifications are still required, it proceeds to (46), returning again after this has been performed. If a number of specific loading cases is to be derived from linear combinations of the previous unit loadings contained in the $\mathbf{b}$, the computer proceeds to block (41). This part of the programme is usually oriented to the final objective, whether it is a straightforward computation or whether an iteration is to be carried out to achieve an optimum weight, whether the programme is acting as a sub-routine to a master programme, e.g. analysing a whole aircraft.

If a number of specific loading cases is to be dealt with, the computer proceeds to (41) where $\mathbf{R}$ is fed in, then to (42) for the stresses and to (43) for the deflections. Following this, again the external and internal equilibria, and compatibility, are checked (44) and (45); then the computer returns again to (40) for the next decision.

If cut-outs and modifications are to follow, they are either specified, at least partly, by the machine if the programme contains an automatic iteration, or by an input of orders (see Chapter III). In (47) the machine checks whether any cut-outs are redundant, resulting in a singular set of equations. This is best done using again logical machine orders of the type used in Chapter III , and will not be discussed here in detail. The machine might be programmed to exclude automatically any unnecessary cut-out, and print out information to that effect.

In (48) the machine checks further whether it is more economical to proceed with the modifications, or if it be not better to modify the original input data and start anew. If it is found more advantageous to continue, the machine proceeds to (49), (50), (51) to (55).

At (55), the matrix to be inverted

$$\delta = \left[b_{1h} D^{-1} b_{1h}^{t} + b_{2rh} D_{2r}^{-1} b_{2rh}^{t} + \begin{bmatrix} 0 & 0 \\ 0 & f_{\Delta k}^{-1} \end{bmatrix} \right]$$

is calculated, then checked for symmetry (56) and conditioning (57) in the already prescribed manner. Then the inversion into $\mathbf{b}_h$ takes place (58) followed by checking the identity

$$\delta \; (\delta^{-1} \; b_h) \; - \; b_h \; = \; 0 \qquad \text{(IV,10)}$$

in (59). A decision is made (60) whether the inversion should be repeated with higher precision, or to proceed at once to the calculation of the modified stresses $\mathbf{b}_m$ (61). These results are then checked for compatibility (62) and equilibrium (63). The cut-out stresses $\mathbf{b}_{hc}$ are extracted with the aid of the β matrices (see Chapter III) and compared with the stresses on the remaining structure (64) in order to verify that they have vanished, and to determine the accuracy of the elimination. Finally the flexibility matrix $\mathbf{F}_m$ is obtained and a return is made to block (40) again for further decisions.

The introduction of multi-cell cross-sections as well as the associated multiply-connected rings, rigid diaphragms etc. scarcely changes the main line of the programme. One of the advantages of the freedom to idealize and regularize the fuselage enables first the setting up of the basic matrices as described. One can then always extract certain elements and add others instead , using either Boolean super-matrices or special orders. For instance, to extract a certain sub-matrix $\mathbf{A}_{ij}$ of a super-matrix $\mathbf{A}$ and insert another one, $\bar{\mathbf{A}}_{ij}$ of the same dimension in its place, we use the following equation

$$A_{new} \; = \; A \; - E_{ii} \, A \; E_{jj} + E_{ii} \; \underset{[(\,),-]}{\bar{A}_{ij}} \; E_{jj}$$

where all the $\mathbf{E}_{pq}$ matrices are Boolean super-matrices whose o and 1 elements are considered to be sub-matrices for an order such as to be compatible with the other matrices involved.

One can also split and rejoin matrices, changing their order in the meantime, thus allowing the new super-matrix to contain sub-matrices of a different dimension.

The programmes used up till now employ mainly uniform super-matrices. One would not usually specify the order of a matrix, except perhaps during its formation or in some special operation which affects the dimensions such as a re-partitioning. The matrix code should then store the dimensions and addresses, and organize its own storage space.

It would be useful, if before the programme is obeyed, even during its assembly, that the matrix code effects a simulated run in which all logical orders, such as looping, are obeyed, but no actual matrix computations carried out. This run could be used to organize the storage space, inserting orders to reserve room for results, or to over-write intermediate results which will not be used again.

CHAPTER V

CONDITIONING OF STRUCTURAL EQUATIONS

V-a Summary and Introduction

The problems of conditioning matrices have been widely discussed by mathematicians, as well as many of those who had to apply matrices to the solution of physical problems. Probably many years will elapse before one can obtain a satisfactory general theoretical approach for the detection and measurement of the loss of accuracy involved in the solution of a set of simultaneous equations. However, from the practical point of view, it may be useful to study the symptoms which are associated with such a phenomenon in one or two cases of particular importance. For this purpose the general form of the inverse of a finite difference equation of any order has been established with the first and last diagonal elements free to assume any value. In this manner we can study the effect of changing these elements on the accuracy of the others, particularly when the matrix becomes ill-conditioned and finally singular. Although this particular example is not in any way claimed to be characteristic of all possible structural cases which could occur in practice, it may be nevertheless be illustrative to give a generalized exact discussion of such a matrix. The type of matrix in question is, for example, encountered in the $\mathbf{D}_q$ of a cylindrical fuselage (not necessarily circular) or even in the case of fuselages with taper such that certain invariants are preserved. If the fuselage violated this condition, the resulting system of equations, although not following strictly the same pattern, is, however, sufficiently similar to justify the discussions being extended at least qualitatively to these cases. This inverse of the finite difference equations is subsequently used to obtain the inverse of some related matrices, of which again the most important for the purpose of this work is the equation of the second difference which can immediately be derived from the original one. After the effect of the variation of the two elements is discussed, the inverse is used to test various mathematical criteria advanced by various authors as being a measure of the "conditioning" of matrices.

In the course of this chapter we discuss a new technique which might help to overcome problems of accumulation of errors, especially in systems which contain a large number of unknowns. Although intended here primarily for the fuselage analysis, it might possibly be extended to various other types of structure.

Other incidental results also given turn out to be extremely useful, amongst other things, as tests for accuracy in certain computer programmes, such as those for the inversion and determination of the eigenvectors of matrices.

V-b The Inverse of the Finite Difference Matrix ∇ of Order $m \times m$ with Variable End Coefficients

The general form of the matrix is

$$\underset{(m\times m)}{\nabla} = \begin{bmatrix} 1+\alpha & -1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 1+\alpha \end{bmatrix} \qquad (V,1)$$

Using the notation

$$E_{ij} = e_i \quad e_j^t \qquad (V,2)$$

(see also Ref. 7), we can write

$$\underset{(m\times m)}{\nabla} = (1+a)\, E_{1,1} + 2\sum_{i=2}^{m-1} E_{i,i} - \sum_{i=1}^{m-1} E_{i,i+1} - \sum_{i=1}^{m-1} E_{i+1,i} + (1+a)\, E_{mm}$$

(V,1a)

In order to obtain the inverse of a general three band matrix of this form, we study the inverses of the matrices of order (2×2) , $(3,3)$ and (4×4)

$$\underset{(2\times 2)}{\nabla} = \begin{bmatrix} (1+a) & -1 \\ -1 & (1+a) \end{bmatrix}$$

$$\underset{(2\times 2)}{\nabla}^{-1} = \frac{1}{a(2+a)} \begin{bmatrix} (1+a) & +1 \\ +1 & (1+a) \end{bmatrix} \qquad \text{(V,3)}$$

$$\underset{(3\times 3)}{\nabla} = \begin{bmatrix} (1+a) & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & (1+a) \end{bmatrix}$$

$$\underset{(3\times 3)}{\nabla}^{-1} = \frac{1}{a(2+2a)} \begin{bmatrix} (1+2a) & (1+a) & 1 \\ (1+a) & (1+a)^2 & (1+a) \\ 1 & (1+a) & (1+2a) \end{bmatrix} \qquad \text{(V,3a)}$$

$$\underset{(4\times 4)}{\nabla} = \begin{bmatrix} (1+a) & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & (1+a) \end{bmatrix} \qquad \text{(V,3b)}$$

$$\underset{(4\times 4)}{\nabla}^{-1} = \frac{1}{a(2+3a)} \begin{bmatrix} (1+3a) & (1+2a) & (1+a) & 1 \\ (1+2a) & (1+a)(1+2a) & (1+a)^2 & (1+a) \\ (1+a) & (1+a)^2 & (1+a)(1+2a) & (1+2a) \\ 1 & (1+a) & (1+2a) & (1+3a) \end{bmatrix}$$

We observe first that there is always the common denominator

$$a\left[2 + (m-1)a\right]$$

Since the elements left inside the matrix are easily recognised to be the co-factors of the original elements, this denominator is obviously the value of the determinant.

Next we note that the value of the element $\nabla_{1,m}$ within the square brackets of the inverse shown in Eqs.(V,3, 3a, 3b) is unity and that the values increase linearly in both directions towards $\nabla_{1,1}$ and $\nabla_{m,m}$. From this we deduce that

$$\nabla^{-1}_{(m\times m)\,1,j} = \frac{[1+(m-j)a]}{a[2+(m-1)a]} \tag{V,4}$$

and

$$\nabla^{-1}_{(m\times m)\,i,m} = \frac{[1+(i-1)a]}{a[2+(m-1)a]} \tag{V,4a}$$

Examining the general element we find for $i \leqslant j$

$$\nabla^{-1}_{(m\times m)\,i,j} = \nabla^{-1}_{(m\times m)\,i,m} \times \nabla^{-1}_{(m\times m)\,1,j} \times \frac{1}{\text{Determinant}} = \frac{[1+(i-1)a][1+(m-j)a]}{a[2+(m-1)a]}$$

and for $i \geqslant j$ (V,5)

$$\nabla^{-1}_{i,j} = \nabla^{-1}_{j,i} = \frac{[1+(m-i)a][1+(j-1)a]}{a[2+(m-1)a]} \tag{V,5a}$$

Eqs. (V,5) and (V,5a) establish the general inverse of ∇ .

We can prove this to be the correct inverse if we establish that

(i) $\nabla_{1.} \quad \nabla_{.1}^{-1} = +1$

(ii) $\nabla_{1.} \quad \nabla_{.j}^{-1} = 0$

(iii) $\nabla_{i.} \quad \nabla_{.i}^{-1} = +1$

(iv) $\nabla_{i.} \quad \nabla_{.j}^{-1} = +0$

(v) $\nabla_{m.} \quad \nabla_{.m}^{-1} = +1$

(vi) $\nabla_{m.} \quad \nabla_{.j}^{-1} = 0$

Proofs

(i) $$\nabla_{1.} \quad \nabla_{.1}^{-1} = \frac{(1+a).1.[1+(m-1)a]}{a[2+(m-1)a]} - \frac{1.[1+(m-2)a].1}{a[2+(m-1)a]}$$

$$= \frac{(1+a)+(m-1)a(1+a)-1-(m-2)a}{a[2+(m-1)a]}$$

$$= \frac{2a+(m-1).a.(1+a)-(m-1)a}{a[2+(m-1)a]} = \frac{2a+(m-1).a^2}{a[2+(m-1)a]}$$

$$= +1$$

(ii) $$\nabla_{1.} \quad \nabla_{.j}^{-1} = \frac{(1+a)\,.\,1.\,[1+(m-j)\,a]}{a\,[2+(m-1)\,a]} - \frac{1.\,(1+a)\,[1+(m-j)\,a]}{a\,[2+(m-1)\,a]}$$

$$= 0$$

(iii) $$\nabla_{i.} \quad \nabla_{.i}^{-1} = - \frac{1.\,[1+(m-2)a]\,[1+(m-i)a]}{a\,[2+(m-1)\,a]}$$

$$+ \frac{2\,[1+(i-1)a]\,[1+(m-i)a]}{a\,[2+(m-1)\,a]} - \frac{1.\,[1+(m-i-1)a]\,[1+(i-1)\,a]}{a\,[2+(m-1)\,a]}$$

$$= \frac{[1+(m-i)a]\,a + a\,[1+(i-1)a]}{a\,[2+(m-1)\,a]}$$

$$= 1$$

(iv) a) if $(i+1) \leqslant j$

$$\nabla_{i.} \quad \nabla_{.j}^{-1} = - \frac{1.\,[1+(i-2)\,a]\,[1+(m-j)a]}{a\,[2+(m-1)a]}$$

$$+ \frac{2.\,[1+(i-1)a]\,[1+(m-j)a]}{a\,[2+(m-1)a]}$$

$$- \frac{1.\,[1+i\,a]\,[1+(m-j)\,a]}{a\,[2+(m-1)\,a]}$$

$$= 0$$

b) if $(i-1) \geqslant j$

$$\nabla_{i.} \quad \nabla_{.j}^{-1} = - \frac{1.\,[1+(m-i+1)a]\,[1+(j-1)\,a]}{a\,[2+(m-1)\,a]} + \frac{2\,[1+(m-i)a]\,[1+(j-1)a]}{a\,[2+(m-1)a]}$$

$$- \frac{1.\,[1+(m-i-1)a]\,[1+(j-1)a]}{a\,[2+(m-1)a]}$$

$$= 0$$

(v) and (vi) follow in the same manner as (i) and (ii)
Thus we have shown that Eqs (V,5, 5a) are correct general expressions for any element in the inverse of ∇.

V-c Discussion of the Inverse of the Matrix

A characteristic feature we observe about the inverse of the matrix, is that all elements are proportional to

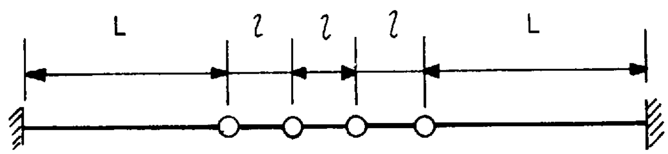
$$\frac{1}{\alpha\,[2+(m-1)\alpha]}$$

Thus the whole matrix is inversely proportional to α. If α be zero, the matrix becomes singular, and the inverse infinite. The nearer α approaches zero, the higher the elements of ∇^{-1} become. Now, as we stated in the introduction, this does not affect the accuracy of the general solution except when one uses a limited number of digits, as when a matrix is inverted in the computer. It is obvious that the error in the inverse is at least as large as the error in α, and if we consider as an example the following ∇ of the order (4×4)

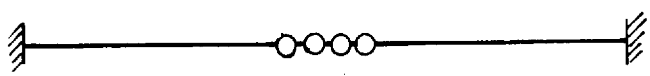
$$\nabla = \begin{bmatrix} 1.000\,052\,3 & -1.000\,000\,0 & 0 & 0 \\ -1.000\,000\,0 & 2.000\,000\,0 & -1.000\,000\,0 & 0 \\ 0 & -1.000\,000\,0 & 2.000\,000\,0 & -1.000\,000\,0 \\ 0 & 0 & -1.000\,000\,0 & 1.000\,523\,0 \end{bmatrix}$$

we can immediately deduce that the inverse will only be accurate to three figures.

In order to draw a more definite picture of what this means physically, we discuss the problem of a chain of 5 bars connected together in line as in Fig. V, 1a. The bars are all of the same cross-section and of the same length l, except the first and last which are of length L.



a) General case



b) First limiting case

c) Second limiting case

d) Third limiting case

Fig. V,1 Chain of five bars

In analysing this system by the Displacement Method, we obtain for the stiffness of a regular element

$$k_i = \frac{EA}{l} \qquad (V,6)$$

and for the first and last elements

$$k = k = \frac{EA}{L} \qquad (V,6a)$$

Our stiffness matrix for the complete system is hence,

$$K = \begin{bmatrix} \frac{EA}{L} + \frac{EA}{l} & -\frac{EA}{l} & 0 & 0 \\ -\frac{EA}{l} & 2\frac{EA}{l} & -\frac{EA}{l} & 0 \\ 0 & -\frac{EA}{l} & 2\frac{EA}{l} & -\frac{EA}{l} \\ 0 & 0 & -\frac{EA}{l} & \frac{EA}{l} + \frac{EA}{L} \end{bmatrix}$$

$$= \frac{EA}{l} \begin{bmatrix} (1+a) & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & (1+a) \end{bmatrix} \qquad (V,7)$$

where $$a = \frac{l}{L} \qquad (V,7a)$$

which, but for the factor $\left(\frac{EA}{l}\right)$, is identical with the (4 x 4) matrix shown in the first of Eqs. (V, 3b). It is now clear, according to our previous argument, that as α diminishes, (which is when l tends to become much smaller than L), the equation becomes ill-conditioned. This corresponds to the structure in Fig. (V,1b), where the first and last elements are much larger than the ordinary member in the chain. It would also appear by physical reasoning that the second idealisation as shown in Fig. (V,1c)will lead to ill-conditioning. Although this is discussed later on, the limiting form of $\mathbf{K}^{-1}$ in the case when α becomes very small is

$$\mathbf{K}^{-1} = \frac{1}{2}\left(\frac{L}{l}\right)\left(\frac{1}{EA}\right)\begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 \end{bmatrix} \qquad (V,3)$$

Although the mathematical significance is discussed further below, it is obvious that this limiting matrix for $\mathbf{K}^{-1}$ is of rank one. A further case of interest arises when α becomes very large.Then the first and last diagonal elements will be of a different order of magnitude. To examine this case we set

$$\alpha \rightarrow \infty$$

and write

$$b = \frac{1}{\alpha}$$

which is a very small quantity, so that b^2 is negligible.

Then

$$\nabla = \frac{b^2}{(2b+3)} \begin{bmatrix} (b+3)/b & (b+2)/b & (b+1)/b & 1 \\ (b+2)/b & \frac{(b+1)(b+2)}{b^2} & (b+1)^2/b^2 & (b+1)/b \\ (b+1)/b & (b+1)^2/b^2 & \frac{(b+1)(b+2)}{b^2} & (b+2)/b \\ 1 & (b+1)/b & (b+2)/b & (b+3)/b \end{bmatrix}$$

$$= \begin{bmatrix} b\,\frac{(b+3)}{(2b+3)} & b\,\frac{(b+2)}{(2b+3)} & b\,\frac{(b+1)}{(2b+3)} & \frac{b^2}{(2b+3)} \\ b\,\frac{(b+2)}{(2b+3)} & \frac{(b+1)(b+2)}{(2b+3)} & \frac{(b+1)^2}{(2b+3)} & b\,\frac{(b+1)}{(2b+3)} \\ b\,\frac{(b+1)}{(2b+3)} & \frac{(b+1)^2}{(2b+3)} & \frac{(b+1)(b+2)}{(2b+3)} & b\,\frac{(b+2)}{(2b+3)} \\ \frac{b^2}{(2b+3)} & b\,\frac{(b+1)}{(2b+3)} & b\,\frac{(b+2)}{(2b+3)} & b\,\frac{(b+3)}{(2b+3)} \end{bmatrix} \tag{V,9}$$

This becomes in the limit,

$$\begin{bmatrix} b & \frac{2b}{3} & \frac{b}{3} & \frac{b^2}{3} \\ \frac{2b}{3} & 2/3 & 1/3 & \frac{b}{3} \\ \frac{b}{3} & 1/3 & 2/3 & \frac{2b}{3} \\ \frac{b^2}{3} & \frac{b}{3} & \frac{2b}{3} & b \end{bmatrix}$$

The two corner elements have not been put to zero since we only neglect b^2 in comparison with b or with unity.

Obviously, this does not result in the matrix becoming singular. Effectively, the matrix is split into two parts, one of which, a (2 x 2) matrix, is independent of b. This matrix is nothing but the inverse of the corresponding part in the original matrix.

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix}$$

It can be proved generally that that part of the inverse obtained by striking out the first and last rows and columns, is merely the inverse of the corresponding finite displacement matrix of order $(m-2) \times (m-2)$ with $\alpha = 1$

If $\alpha = 1$ and $(m-2)$ be substituted for m, Eqs. (V,5) becomes for $i \leq j$

$$\nabla^{-1}_{(m-2)\times(m-2)\, i,j} = \frac{i\,(m-j+1)}{(m-1)} \qquad \text{(V,10)}$$

with a similar expression for $i \geq j$

Taking the original expression, we can rewrite it as follows for $i \leq j$

$$\nabla^{-1}_{(m\times m)\, i,j} = \frac{(i-1)(m-j) + \dfrac{(m-1)-(j-i)}{\alpha} + \dfrac{1}{\alpha^2}}{\left[\dfrac{2}{\alpha} + (m-1)\right]}$$

If we take $\alpha \geq 1$ and write $i+1$ instead of i, and $j+1$ instead of j, we obtain

$$\nabla^{-1}_{((m-2)\times(m-2))\, i,j} = \frac{i\,(m-j-1)}{(m-1)} \qquad \text{q.e.d.}$$

wherein the first and last rows and columns are roughly proportional to b . Thus the matrix remains theoretically non-singular and probably well-conditioned, although, in practice, the latter might depend also on the method of inversion.

What we deduce from these examples is that the choice of the grid in the idealization of a structure should be made carefully and as uniformly as possible.

Returning to the general expression for the element $\nabla^{-1}_{i,j}$ where $i \leqslant j$, given in Eqn. (V,5), we split the expression into three distinct parts. The argument that follows is, due to symmetry, applicable in the same manner to the elements below the diagonal

$$\nabla^{-1}_{i,j} = \frac{[1+(i-1)a][1+(m-j)a]}{a\,[2+(m-1)a]}$$

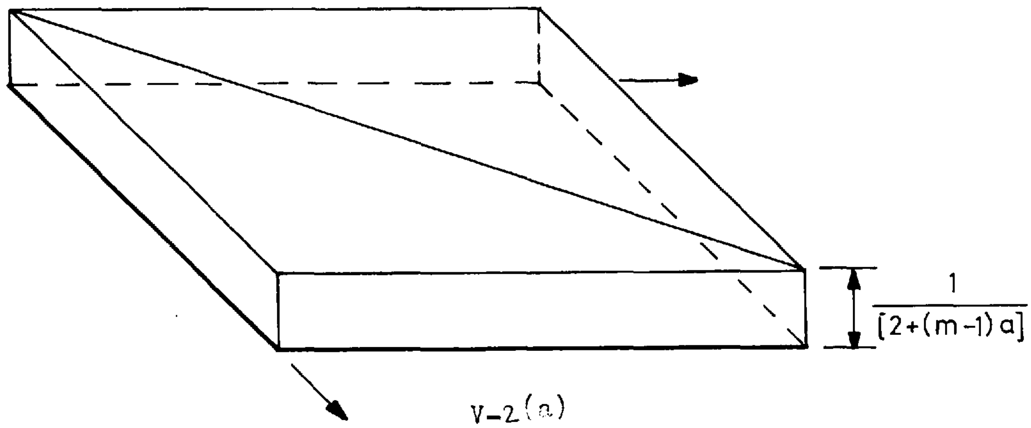
$$= \frac{1}{a[2+(m-1)a]} + \frac{[(i-1)+(m-j)]\,a}{a[2+(m-1)a]} + \frac{(i-1)(m-j)a^2}{a[2+(m-1)a]}$$

$$= \frac{1}{a[2+(m-1)a]} + \frac{(m-1)-(j-i)}{[2+(m-1)a]} + \frac{(i-1)(m-j)\,a}{[2+(m-1)a]} \qquad (V,11)$$

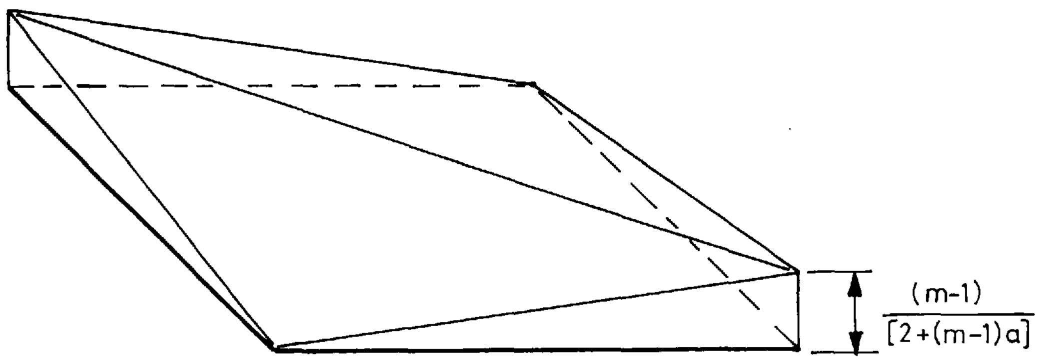
The relative magnitudes of the elements corresponding to these three parts of ∇^{-1} are shown schematically in Figs. (V,1a, b, c). The first term is constant all over the matrix. It is only dependent on a and m . The second term is a maximum on the diagonal, where it is constant and equal to

$$\frac{(m-1)}{a[2+(m-1)a]}$$

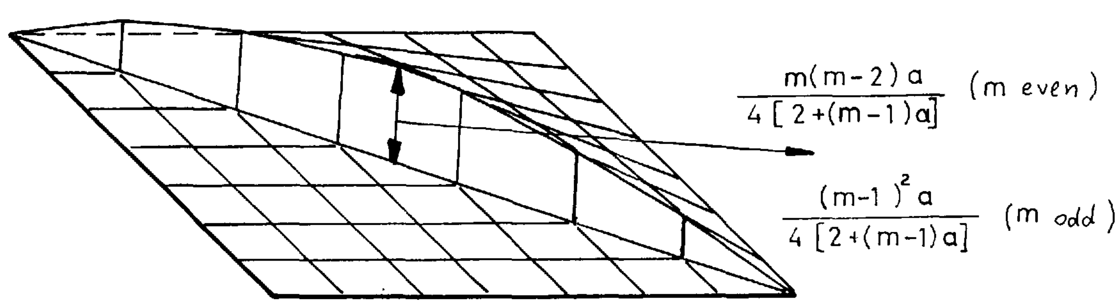
The third term is zero on all sides, but with maximum values on the main diagonal and the one perpendicular to it (extending from $\nabla^{-1}_{1,m}$ to $\nabla^{-1}_{m,1}$)



V-2(a)



V-2(b)



V-2(c)

Fig.V-2, Schematic representation of the basic components in ∇^{-1}.

The maximum value of the element is

$$\frac{m\,(m-2)\,a}{4\,[2+(m-1)\,a]} \qquad \text{for } m \text{ even}$$

and

$$\frac{(m-1)^2\,a}{4\,[2+(m-1)\,a]} \qquad \text{for } m \text{ odd}$$

We study now the ratios between the various contributions starting at the main diagonal

$$\mathrm{I} : \mathrm{II} = \frac{1}{a\,(m-1)} \tag{V,12}$$

i.e. $\mathrm{I} > \mathrm{II}$ if $a\,(m-1) < 1$

or $a < \frac{1}{(m-1)}$ $\left(\approx \frac{1}{m} \quad \text{for large } m \right)$

$$\mathrm{I} : \mathrm{III} \text{ (in the centre)} = \frac{4}{a^2\,(m-1)^2} \tag{V,13}$$

i.e. $\mathrm{I} > \mathrm{III}$

if $a^2(m-1)^2 < 4$

(V,13a)

i.e. if $a\,(m-1) < \pm 2$ or $\left(a \approx < \pm \frac{2}{m} \right)$

Again we show a to be inversely proportional to m. From Eqs. (V,12) and (V,13) it follows that the larger the matrix, the more sensitive it is to disturbances in the first and last elements. This follows from the fact that it is the dominance of the first pattern in Fig. (V,2)which causes inaccuracies in the inverse. So when α approaches the order of magnitude of $1/m$, the accuracy of the diagonal elements in general, and that of the elements in the middle of the matrix in particular, is the same as the accuracy of α in the original matrix.

That of a tending to zero is obviously not the only case in which the elements of ∇^{-1} become very large, and thus ill-conditioned. The other is when $[2 + (m-1)a]$ tends to zero, i.e. if

$$2 + (m-1)\,a = 0$$

or

$$a = -\frac{2}{(m-1)} \approx -\frac{2}{m} \qquad (V,14)$$

This is one of the limiting values of Eqn. (V,13a).

It would be appropriate before concluding the discussion of this particular instructive example to point out that this is a one-dimensional problem. In the case of a two-dimensional grid of elements, we expect the inaccuracies in the solution to be smaller than in the one-dimensional case, since the elements are connected not only in series but also in parallel. If we consider the case of a long chain subjected to a load at a nodal point somewhere in the middle, we see that the difference in the deflected shape due to that and one due to a load applied to the next nodal point is very small. Thus the effect of elasticity of the structure to both sides is a loss of accuracy. Actually in the K^{-1} we would always notice that the maximum inaccuracy occurs in the middle, and increases with size. If the number of the unknowns were distributed in two dimensions, the number of elements in the chain between the supports would be fewer, and a better accuracy be expected, enhanced by the fact that the deflected forms between two adjacent points in the cross-wise direction bear again a certain similarity. Thus we might also say that the condition here is a function of the physical problem as well as the size of the matrix.

V-d Related Matrices; Some Incidental Mathematical Results

V-d 1 Another matrix worthy of consideration , is the similar three band matrix, in which the elements are all positive, i.e.

$$\underset{(m\times m)}{\Delta} = \begin{bmatrix} (1+a) & +1 & 0 & \cdots & 0 & 0 & 0 \\ +1 & +2 & +1 & \cdots & 0 & 0 & 0 \\ 0 & +1 & +2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & +2 & +1 & 0 \\ 0 & 0 & 0 & \cdots & +1 & +2 & +1 \\ 0 & 0 & 0 & \cdots & 0 & +1 & (1+a) \end{bmatrix} \qquad (V,15)$$

Although we can derive the inverse of this matrix from the other by inspection, it might be more interesting to proceed via the matrix $W_{(m)}$ which we define according to the conventions introduced at the beginning by

$$W_{(m)} = \sum_{i=1}^{m} (-1)^{i} \; E_{i,i} \qquad (V,16)$$

One obvious property of this matrix is that

$$W_{(m)}^{2} = \left(\sum_{i=1}^{m} (-1)^{i} E_{i,i} \right) \left(\sum_{p=1}^{m} (-1)^{p} E_{p,p} \right)$$

$$= \sum_{i=1}^{m} \sum_{p=1}^{m} (-1)^{i+p} \; E_{i,i} \; E_{p,p}$$

but according to Ref. (7)

$$E_{i,j} \quad E_{p,k} \quad = \quad \begin{matrix} 0 & j \neq p \\ E_{i,k} & j = p \end{matrix} \tag{V,17}$$

it follows that

$$W_{(m)}^{2} = \sum_{i=1}^{m} E_{i,i} = I \tag{V,18}$$

i.e.

$$W_{(m)}^{-1} = W_{(m)} \tag{V,18a}$$

Applying the notation of Eqn. (V,2) we find from Eqs. (V,1a) and (V,18)

$$\nabla \quad W = \Big[(1+a) \; E_{11} + \sum_{i=2}^{m-1} 2 \cdot E_{i,i} + (1+a) \; E_{m,m} + \sum_{i=1}^{m-1} (-1) \; E_{i,i+1} + \sum_{i=1}^{m-1} E_{i+1,i} \Big] \times \Big[\sum_{p=1}^{m} (-1)^{p} \; E_{p,p} \Big]$$

$$= -(1+a) \; E_{11} + \sum_{i=2}^{m-1} (-1)^{i} \cdot 2 \; E_{i,i} + (-1)^{m} (1+a) \; E_{m,m} + \sum_{i=1}^{m-1} (-1)^{i+2} \; E_{i,i+1} + \sum_{i=1}^{m-1} (-1)^{i+1} E_{i+1,i}$$

Premultiplying now by $W_{(m)}$

$$W_{(m)} \; \nabla_{(m)} \; W_{(m)} = \Big[\sum_{p=1}^{m} (-1)^{p} \; E_{pp} \Big] \Big[(1+a) \; E_{11} + \sum_{i=2} (-1)^{i} \cdot 2 \cdot E_{i,i} + (-1)(1+a) \; E_{m,m} + \sum_{i=1}^{(m-1)} (-1)^{i+2} E_{i,i+1} + \sum_{i=1}^{m-1} (-1)^{i+1} E_{i+1,i} \Big]$$

$$= (-1)^2(1+a)\ E_{1,1} + \sum_{i=2}^{m-1} (-1)^{2i} \cdot 2\ E_{i,i} + (-1)^{2m}(1+a)\ E_{m,m}$$

$$+ \sum_{i=1}^{m-1} (-1)^{2i+2}\ E_{i,i+1} + \sum_{i=1}^{m-1} (-1)^{2i+2}\ E_{i+1,i}$$

$$= (1+a)\ E_{1,1} + \sum_{i=2}^{m-1} 2 \cdot E_{i,i} + (1+a)\ E_{m,m} + \sum_{i=1}^{m-1} E_{i,i+1} + \sum_{i=1}^{m-1} E_{i+1,i}$$

$$= \Delta_{(m)}$$

Therefore

$$\Delta^{-1}_{(m)} = W_{(m)}\ \nabla^{-1}_{(m)}\ W_{(m)} \qquad (V,19)$$

This simply means that each element of the inverse of $\nabla_{(m)}$ has to be multiplied by $(-1)^{i+j}$ to obtain the inverse of $\Delta_{(m)}$. Now we can write directly the typical elements in $\Delta^{-1}_{(m)}$ as follows.

For $i \leqslant j$

$$\Delta^{-1}_{(m)_{i,j}} = \frac{(-1)^{i+j}\left[1+(i-1)a\right]\left[1+(m-j)a\right]}{a\left[2+(m-1)a\right]} \qquad (V,20)$$

and for $i \geqslant j$

$$\Delta^{-1}_{(m)_{i,j}} = \frac{(-1)^{i+j}\left[1+(m-i)a\right]\left[1+(j-1)a\right]}{a\left[2+(m-1)a\right]} \qquad (V,20a)$$

The conditioning is thus not affected by this particular pattern of changing signs.

V-d 2 The ∇ and Δ Matrices for $\alpha = 1$

The ∇ matrix with $\alpha = 1$ corresponds to a chain of the type shown in Fig. V,1 in which all elements are of the same length, and thus, in a way, to a regularly idealized structure. The corresponding matrix follows through the expressions in Eqs. (V,5) and (V,5a) simplifying, for $\alpha = 1$, to

$$\nabla^{-1}_{(m)_{i,j}} = \frac{(-1)^{i+j}\, i\,(m-j+1)}{(m+1)} \qquad (i \leq j) \qquad (V,21)$$

and

$$\nabla^{-1}_{(m)_{i,j}} = \frac{(-1)^{i+j}\,(m-i+1)\, j}{(m+1)} \qquad (i \geq j) \qquad (V,21a)$$

Similarly in the matrix Δ , Eqs. (V,20) and (V,20a) become, for $\alpha = 1$

$$\Delta^{-1}_{(m)_{i,j}} = \frac{(-1)^{i+j}\, i\,(m-j+1)}{(m+1)} \qquad (i \leq j) \qquad (V,22)$$

and

$$\Delta^{-1}_{(m)_{i,j}} = \frac{(-1)^{i+j}\,(m-i+1)\, j}{(m+1)} \qquad (i \geq j) \qquad (V,22a)$$

This matrix is also given in Ref. (7)

V-d 3 The Matrix ∇^2

This matrix, obtained by merely squaring the matrix ∇ , is a five band matrix given by

$$\nabla^2 = \begin{bmatrix} 1+(1+a)^2 & -(3+a) & 1 & \cdots & 0 & 0 & 0 \\ -(3+a) & 6 & -4 & \cdots & 0 & 0 & 0 \\ 1 & -4 & 6 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 6 & -4 & 1 \\ 0 & 0 & 0 & \cdots & -4 & 6 & -(3+a) \\ 0 & 0 & 0 & \cdots & 1 & -(3+a) & 1+(1+a)^2 \end{bmatrix} \qquad (V,23)$$

Eqn. (V,23) may be considered typical for the D_r of a four-boom fuselage in which all rings are similar and all bays except the first and last equal in length. (As in Fig. V,3).
Again

$$\alpha = \frac{d}{D} \qquad (V,24)$$

The restriction on the number of flanges is later removed.

Although one can derive an explicit expression for the element of ∇^{-2} it is most elegantly obtained by a direct multiplication of the ∇^{-1} with itself.

V-d 4 The Finite Difference, and Double Difference Super-Matrices

What has applied before to the matrices with scalar elements, applies also to super-matrices with sub-matrices which are related to one another by a scalar factor. Thus, defining

$$\underset{[M,m]}{\nabla} = \begin{bmatrix} (1+a)A & -A & 0 & \cdots & 0 & 0 \\ -A & 2A & -A & \cdots & 0 & 0 \\ 0 & -A & 2A & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2A & -A \\ 0 & 0 & 0 & \cdots & -A & (1+a)A \end{bmatrix} \qquad (V,25)$$

we can immediately derive the general term in the inverse matrix from that of Eqn. (V,5), (V, 5a) by simply multiplying by $R = A^{-1}$

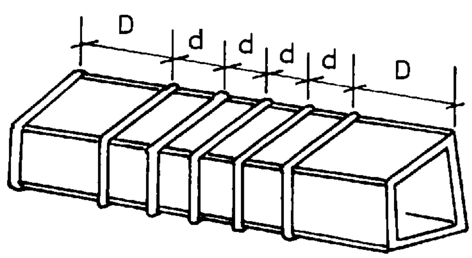


Fig. V-3,
Four boom fuselage

We find for $i \leqslant j$

$$\underset{[M,m]_{i,j}}{\nabla^{-1}} = \frac{[1+(i-1)a][1+(M-j)a]}{a[(M-1)a+2]} \; \mathbf{R} \qquad \text{(V,26)}$$

and for $i \geqslant j$

$$\underset{[M,m]_{i,j}}{\nabla^{-1}} = \frac{[1+(M-i)a][1+(j-1)a]}{a[(M-1)a+2]} \; \mathbf{R} \qquad \text{(V,26a)}$$

The same argument applies to super-matrices of the form $\mathbf{W}\nabla\mathbf{W}$ as well as ∇

V-d 5 Inversion of General Super-Matrices with Directly Proportional Sub-Matrices

An interesting result follows immediately from the last section, when the sub-matrices of a super-matrix are all of the same form, and differ only by a scalar factor. Thus we define a matrix $\mathfrak{D}$ which consists of ($M \times M$) sub-matrices, each of dimensions ($m \times m$)

$$\underset{[M,m]}{\mathfrak{D}} = \begin{bmatrix} d_{11}\mathbf{D} & d_{12}\mathbf{D} & \cdots\cdots & d_{1j}\mathbf{D} & \cdots\cdots & d_{1M}\mathbf{D} \\ d_{21}\mathbf{D} & d_{22}\mathbf{D} & \cdots\cdots & d_{2j}\mathbf{D} & \cdots\cdots & d_{2M}\mathbf{D} \\ \vdots & \vdots & & \vdots & & \vdots \\ d_{i1}\mathbf{D} & d_{i2}\mathbf{D} & \cdots\cdots & d_{ij}\mathbf{D} & \cdots\cdots & d_{iM}\mathbf{D} \\ \vdots & \vdots & & \vdots & & \vdots \\ d_{M1}\mathbf{D} & d_{M2}\mathbf{D} & \cdots\cdots & d_{Mj}\mathbf{D} & \cdots\cdots & d_{MM}\mathbf{D} \end{bmatrix} \qquad \text{(V,27)}$$

To obtain the inverse of this matrix, $\mathfrak{S}^{-1}$ it is obvious that we only need to invert two matrices, one of the order ($m \times m$) and the other of the order ($M \times M$). The first is the ($m \times m$) matrix $\mathbf{D}$ and the second is the matrix of the coefficients d_{ij} given by

$$\mathbf{d} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \cdots & d_{1M} \\ d_{21} & d_{22} & \cdots & d_{2j} & \cdots & d_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ d_{i1} & d_{i2} & \cdots & d_{ij} & \cdots & d_{iM} \\ \vdots & \vdots & & \vdots & & \vdots \\ d_{M1} & d_{M2} & \cdots & d_{Mj} & \cdots & d_{MM} \end{bmatrix} \qquad (V,28)$$

We call $\mathbf{D}$ the basic sub-matrix and $\mathbf{d}$ the pattern matrix. We assume now that we know the inverse of this pattern matrix, and that it is

$$\mathbf{g} = \mathbf{d}^{-1} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1j} & \cdots & g_{1M} \\ g_{21} & g_{22} & \cdots & g_{2j} & \cdots & g_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ g_{i1} & g_{i2} & \cdots & g_{ij} & \cdots & g_{iM} \\ \vdots & \vdots & & \vdots & & \vdots \\ g_{M1} & g_{M2} & \cdots & g_{Mj} & \cdots & g_{MM} \end{bmatrix} \qquad (V,29)$$

Thus the following relations hold

$$\mathbf{g}\,\mathbf{d} = \mathbf{d}\,\mathbf{g} = \mathrm{I}_M \qquad (V,30)$$

$$\begin{aligned} \mathbf{g}_{i\cdot}\,\mathbf{d}_{\cdot j} = \mathbf{d}_{i\cdot}\,\mathbf{g}_{\cdot j} &= 1 \qquad i = j \\ &= 0 \qquad i \neq j \end{aligned} \qquad (V,30a)$$

The inverse of $\mathfrak{D}^{-1}_{[M,m]}$ can be written immediately

$$\mathfrak{D}^{-1}_{[M,m]} = \begin{bmatrix} g_{11}D^{-1} & g_{12}D^{-1} & \cdots & g_{1j}D^{-1} & \cdots & g_{1M}D^{-1} \\ g_{21}D^{-1} & g_{22}D^{-1} & \cdots & g_{2j}D^{-1} & \cdots & g_{2M}D^{-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ g_{i1}D^{-1} & g_{i2}D^{-1} & \cdots & g_{ij}D^{-1} & \cdots & g_{iM}D^{-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ g_{M1}D^{-1} & g_{M2}D^{-1} & \cdots & g_{Mj}D^{-1} & \cdots & g_{MM}D^{-1} \end{bmatrix} \qquad (V,31)$$

This we can easily prove if we consider the product of the i^{th} super row of $\mathfrak{D}_{[M,m]}$ with the j^{th} super-column of $\mathfrak{D}^{-1}_{[M,m]}$

$$\mathfrak{D}_{[M,m]_{i\cdot}} \; \mathfrak{D}^{-1}_{[M,m]_{\cdot j}} = \sum_{p=1}^{M} d_{ip} \, D \, g_{pj} \, D^{-1}$$

$$= \left(\sum_{p=1}^{M} d_{ip} \, g_{pj} \right) I_m$$

$$= (\mathbf{d}_{i\cdot} \, \mathbf{g}_{\cdot j}) \; I_m = \underset{(m \times m)}{0} \qquad i \neq j$$

$$= \underset{(m \times m)}{I} \qquad i = j \qquad (V,32)$$

In this manner we have proved the correctness of the inverse. The importance of the result appears later when discussing its application to improve the accuracy of the solution of a fuselage structure, with possible extension to other similar problems, and also as a valuable test for an inversion programme. Following our above method one can construct very large matrices of arbitrary conditioning, the inverse of which can be accurately determined beforehand.

V-d 6 Eigenvectors of a General Super-Matrix with Directly Proportional Sub- Matrices

Here another incidental mathematical result is described, which surprisingly enough, also has applications in the fuselage, as well as in the problems of a similar nature and as a valuable test for eigenvalue programmes. Thus we consider again our matrix $\mathbf{S}$ given in Eqn. (V,27) and the corresponding pattern matrix $\mathbf{d}$ of Eqn. (V,28) and assume that the eigenvectors of $\mathbf{D}$ are given by

$$\mathbf{V} = \left[V_1 \quad V_2 \quad V_3 \cdots\cdots V_i \cdots\cdots V_m \right] \qquad \text{(V,33)}$$

with the corresponding eigenvalues

$$\boldsymbol{\lambda} = \left\lceil \lambda_1 \quad \lambda_2 \quad \lambda_3 \cdots\cdots \lambda_i \cdots\cdots \lambda_m \right\rfloor \qquad \text{(V,34)}$$

We know that

$$\mathbf{D} \quad V_i = \lambda_i \quad V_i \qquad \text{(V,35)}$$

and

$$\mathbf{D} \quad \mathbf{V} = \mathbf{V} \quad \boldsymbol{\lambda} \qquad \text{(V,35a)}$$

We also assume the latent vectors of $\mathbf{d}$ to be

$$\mathbf{U} = \left[\mathbf{U}_1 \ \mathbf{U}_2 \ \mathbf{U}_3 \cdots\cdots \mathbf{U}_j \cdots\cdots \mathbf{U}_M \right] \qquad (V,36)$$

with corresponding eigenvalues

$$\boldsymbol{\omega} = \left[\omega_1 \ \omega_2 \ \omega_3 \cdots\cdots \omega_j \cdots\cdots \omega_M \right] \qquad (V,37)$$

so that

$$\mathbf{d} \ \mathbf{U}_j = \omega_j \ \mathbf{U}_j \qquad (V,38)$$

and

$$\mathbf{d} \ \mathbf{U} = \mathbf{U} \ \boldsymbol{\omega} \qquad (V,38a)$$

Writing again the vector $\mathbf{U}_j$ in more detail

$$\mathbf{U}_j = \{ u_{1j} \ u_{2j} \cdots\cdots u_{ij} \cdots\cdots u_{Mj} \} \qquad (V,39)$$

and using it to examine (V, 38a) we obtain by multiplying the ith row of $\mathbf{d}$ with the jth column of $\mathbf{U}$,

$$\mathbf{d}_{i\cdot} \ \mathbf{U}_{\cdot j} = d_{i1} \ u_{1j} + d_{i2} \ u_{2j} \cdots\cdots + d_{ij} \ u_{ij} + \cdots + d_{iM} \ u_{Mj}$$

$$= \omega_j \ u_{ij} \qquad (V,38b)$$

We will prove now that the (Mm x 1) vector $\mathbf{V}_{(ji)}$ given by

$$\mathbf{V}_{(ji)} = \{ u_{1j}\mathbf{V}_i \quad u_{2j}\mathbf{V}_i \text{ ------ } u_{ij}\mathbf{V}_i \text{ ------ } u_{Mj}\mathbf{V}_i \} \tag{V,39}$$

is also an eigenvector of the matrix $\mathfrak{S}$. For this purpose we premultiply this matrix with the p^{th} super-row of the matrix $\mathfrak{S}$ and use Eqn.(V,38b)

$$\underset{[M,m]_{p.}}{\mathfrak{S}} \quad \mathbf{V}_{(ji)} = d_{p1}\,\mathbf{D}\;u_{1j}\mathbf{V}_i + d_{p2}\,\mathbf{D}\;u_{2j}\mathbf{V}_i + \cdots\cdots + d_{pi}\,\mathbf{D}\;u_{ij}\mathbf{V}_i + \cdots\cdots + d_{pM}\,\mathbf{D}\;u_{Mj}\mathbf{V}_i$$

$$= \omega_j u_{pj}\,\mathbf{D}\quad\mathbf{V}_i = (\omega_j\lambda_i)u_{pj}\mathbf{V}_i \tag{V,40}$$

proving further that the eigenvalue associated with the eigenvector of Eqn. (V,39) is

$$\Lambda_{(ji)} = (\omega_j\lambda_i) \tag{V,41}$$

The complete matrix of all Mm eigenvectors of the matrix $\mathfrak{S}$ can be set up by allotting i and j in Eqn. (V,39) the values 1 to m and 1 to M respectively. The corresponding eigenvalues are obtained through Eqn. (V,41) by again giving i and j the same values.

V-d 7 The Determination of the Determinant of a Super-Matrix with Directly Proportional Sub-Matrices

Considering again the matrix $\mathfrak{D}$ of Eqn. (V,27), we can rewrite it as a product of a matrix $\mathbf{d}_e$ (again of order ($M \times M$), ($m \times m$), whose super-element $\mathbf{d}_{e[M,m]i,j}$ is equal to $d_{ij}\, \mathbf{I}_m$) and a super-diagonal (super-scalar) matrix $\mathbf{D}_e$ whose elements are all equal to $\mathbf{D}$.

Thus,

$$\underset{[M,m]}{\mathfrak{D}} = \underset{[M,m]}{\mathbf{d}_e} \; \underset{[M,m]}{\mathbf{D}_e} \tag{V,42}$$

where

$$\underset{[M,m]}{\mathbf{d}_e} = \begin{bmatrix} d_{11}\mathbf{I}_m & d_{12}\mathbf{I}_m & d_{13}\mathbf{I}_m & \cdots & d_{1j}\mathbf{I}_m & \cdots & d_{1M}\mathbf{I}_m \\ d_{21}\mathbf{I}_m & d_{22}\mathbf{I}_m & d_{23}\mathbf{I}_m & \cdots & d_{2j}\mathbf{I}_m & \cdots & d_{2M}\mathbf{I}_m \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ d_{i1}\mathbf{I}_m & d_{i2}\mathbf{I}_m & d_{i3}\mathbf{I}_m & \cdots & d_{ij}\mathbf{I}_m & \cdots & d_{iM}\mathbf{I}_m \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ d_{M1}\mathbf{I}_m & d_{M2}\mathbf{I}_m & d_{M3}\mathbf{I}_m & \cdots & d_{Mj}\mathbf{I}_m & \cdots & d_{MM}\mathbf{I}_m \end{bmatrix} \tag{V,43}$$

and

$$\mathbf{D}_e = \left\lceil \mathbf{D} \quad \mathbf{D} \quad \mathbf{D} \cdots\cdots\cdots \mathbf{D} \right\rfloor \qquad (V,44)$$

The determinant of $\mathbf{D}_e$ is now obtained as the product of the determinants of the two matrices. Thus

$$det. \underset{[M,m]}{\mathfrak{D}} = det. \underset{[M,m]}{\mathbf{d}_e} \; det. \mathbf{D}_e \qquad (V,45)$$

The determinant of $\mathbf{D}_e$ can be found, once we know the determinant of $\mathbf{D}$

$$det. \mathbf{D}_e = (det. \mathbf{D})^M \qquad (V,46)$$

We can find the determinant of $\mathbf{d}_e$ if we consider the matrix

$$\underset{[M,m]}{\mathbf{E}} = \begin{bmatrix} E_{m_{1,1}} & E_{m_{2,1}} & \cdots & E_{m_{j,1}} & \cdots & E_{m_{M,1}} \\ E_{m_{1,2}} & E_{m_{2,2}} & \cdots & E_{m_{j,2}} & \cdots & E_{m_{M,2}} \\ \vdots & \vdots & & \vdots & & \vdots \\ E_{m_{1,i}} & E_{m_{2,i}} & \cdots & E_{m_{j,i}} & \cdots & E_{m_{M,i}} \\ \vdots & \vdots & & \vdots & & \vdots \\ E_{m_{1,M}} & E_{m_{2,M}} & \cdots & E_{m_{j,M}} & \cdots & E_{m_{M,M}} \end{bmatrix} \qquad (V,47)$$

with the properties

$$\underset{[M,m]}{\mathsf{E}}^{2} = \underset{[M,m]}{\mathsf{I}} \qquad (V,48)$$

and therefore also

$$\underset{[M,m]}{\mathsf{E}}^{-1} = \underset{[M,m]}{\mathsf{E}} \qquad (V,48a)$$

and

$$det\ \underset{[M,m]}{\mathsf{E}} = 1 \qquad (V,49)$$

It is easily seen that

$$\underset{[M,m]}{\mathsf{E}}\ \underset{[M,m]}{\mathsf{d}_e}\ \underset{[M,m]}{\mathsf{E}} = \lceil\, \mathsf{d}\ \ \mathsf{d}\ \ \mathsf{d} \cdots\cdots\cdots \mathsf{d}\,\rfloor \qquad (V,50)$$

Therefore

$$det.\ \underset{[M,m]}{\mathsf{d}_e} = (\,det.\ \mathsf{d}\,)^{M} \qquad (V,51)$$

and from Eqs. (V,45), (V,46) and (V,51)

$$det.\ \underset{[M,m]}{\mathfrak{D}} = (det.\ \mathsf{D})^{M}\ (det.\ \mathsf{d})^{M}$$

$$= (det.\ \mathsf{d}\quad det.\mathsf{D}\,)^{M} \qquad (V,52)$$

We obtain the value of the determinant of the complete super-matrix through knowledge of the two basic determinants. This is also of use to us.

V-d 8 Generalization of V-d5 and V-d 7

Although of less importance, it is probably justifiable to point out that the operations on the special super-matrices which have been described under (V-d 5) and (V-d 7) can be directly extended to such matrices as

$$\underset{[M,m]}{\bar{\mathfrak{D}}} = \underset{[M,m]}{d_e} \quad \underset{[M/,m]}{\bar{D}_e} \tag{V,53}$$

and

$$\underset{[M,m]}{\mathfrak{D}^*} = \underset{[M/,m]}{\bar{D}_e} \quad \underset{[M,m]}{d_e} \tag{V,53a}$$

where

$$\bar{D}_e = \lceil D_1 \quad D_2 \quad D_3 \cdots\cdots\cdots\cdots D_M \rfloor \tag{V,54}$$

i.e. where D_1 to D_M are all different square matrices of order $(m \times m)$. For instance, in obtaining the inverse of $\underset{[M,m]}{\bar{\mathfrak{D}}}$ as in Eqn (V,31) and if Eqn. (V,53) holds, we get

$$\underset{[M,m]}{\bar{\mathfrak{D}}^{-1}} = \begin{bmatrix} g_{11} D_1^{-1} & g_{12} D_1^{-1} & \cdots & g_{1j} D_1^{-1} & \cdots & g_{1M} D_1^{-1} \\ g_{21} D_2^{-1} & g_{22} D_2^{-1} & \cdots & g_{2j} D_2^{-1} & \cdots & g_{2M} D_2^{-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ g_{i1} D_i^{-1} & g_{i2} D_i^{-1} & \cdots & g_{ij} D_i^{-1} & \cdots & g_{iM} D_i^{-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ g_{M1} D_M^{-1} & g_{M2} D_M^{-1} & \cdots & g_{Mj} D_M^{-1} & \cdots & g_{MM} D_M^{-1} \end{bmatrix} \tag{V,55}$$

which follows at once from

$$\underset{[M,m]}{\bar{\mathfrak{D}}^{-1}} = \bar{D}_e^{-1} \quad d_e^{-1} \qquad (V,55a)$$

Similarly, if Eqn. (V,53a) holds, the columns are proportional to the appropriate D_i or

$$\underset{[M,m]}{\mathfrak{D}^{*-1}} = d_e^{-1} \quad \bar{D}_e^{-1} \qquad (V,55b)$$

In (IV-d 7) Eqn. (V,52) becomes for both cases

$$det. \begin{matrix} \bar{\mathfrak{D}} \\ \mathfrak{D}^* \end{matrix} = (det.\ d)^M (det.\ D_1)(det.\ D_2) \cdots\cdots\cdots (det.\ D_M)$$

We must observe that the two kinds of matrices given by Eqs.(V,53) and (V,53a) are no longer symmetrical as far as the basic sub-matrix is concerned, and therefore such a generalization can not be made so simply.

At the end of this section of rather mathematical nature, it seems logical that the best way to handle a super-matrix is such that the properties of its sub-matrices are also taken into account. In many iteration procedures, such as inversion by iteration or obtaining the eigenvectors and eigenvalues of a matrix, an approximation to the results is needed right at the beginning. The 'idealization' of the super-matrix into a basic sub-matrix and a pattern matrix, if at all feasible, would provide an excellent starting point for such a procedure.

V-e Testing for Ill-Conditioning in the Fuselage(and similar Problems)

V-e 1 The various measures of ill-conditioning

In Ref. (7) several criteria are suggested with the aid of which the ill-conditioning may be measured. Most of these criteria, to use the words of Bodewig (Ref. (7)) 'are circumstantial in practice and mostly applicable only in theory as they require mostly the knowledge of the inverse which is just the difficulty for such matrices'. The various measures are given as

a) If $$\underset{(n \times n)}{N(\mathbf{A})} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |A_{ij}|^2} \qquad (V,56)$$

the N-number of $\mathbf{A}$

$$= Nb(\mathbf{A}) = \frac{1}{n} \; N(\mathbf{A}) \; N(\mathbf{A}^{-1}) \qquad (V,56a)$$

b) If $$\underset{(n \times n)}{m(\mathbf{A})} = max. \left| A_{ij} \right| \qquad (V,57)$$

the m-number of $\mathbf{A}$

$$mb(\mathbf{A}) = n \; m(\mathbf{A}) \; m(\mathbf{A}^{-1}) \qquad (V,57a)$$

c) <u>The Goldstine and von Neumann measure</u>

$|\lambda \Lambda|$, where λ and Λ are the dominant eigenvalues of $\mathbf{A}$ and $\mathbf{A}^{-1}$ respectively.

d) The μ factor

$$\mu(\mathbf{A}) = \frac{A_{11} \; A_{22} \; \ldots\ldots \; A_{ii} \; \ldots\ldots \; A_{nn}}{det. \; \mathbf{A}} \qquad (V,58)$$

We can also add to these two further criteria.

e) The sensitivity of the inverse $\mathbf{A}^{-1}$ to a small charge in an element of the matrix $\mathbf{A}$ represented as $d\mathbf{A}$ is given by the total differential.

$$d\,(\mathbf{A}^{-1}) = -\,(\mathbf{A}^{-1})(d\mathbf{A})(\mathbf{A}^{-1})$$

f) A rough measure of linear independence of the rows and columns in the matrix $\mathbf{A}$ is given by the normalised matrix $\bar{\mathbf{A}}$ whose typical element is

$$\bar{A}_{ij} = \frac{A_{ij}}{\sqrt{A_{ii}\ A_{jj}}}$$

with 'ones' along the diagonal.

In this matrix one should perhaps base the judgement on the pattern, just as much as the magnitude of these ratios.

V-e 2 Estimation of the $\mathbf{D}^{-1}$ for a Fuselage

All of the proposed measures of ill-conditioning in matrices require knowledge either of the inverse directly, or of the eigenvectors and eigenvalues connected with it. However, it is necessary to have a measure of such a property before the inversion actually takes place. One needs to know, for instance, if one should use single accuracy or double accuracy or whether a certain procedure, such as an orthogonalisation of the diagonal sub-matrices, should be carried out at more intermediate stations (see Ref.(2) Chapter VII).

A quick guess at the inverse of the matrix before the actual inversion operation is therefore of great value. We suggest here a certain approximate procedure which may be useful in a typical fuselage and is capable of extension to any problem of a similar nature. The first thing we observe is that the fuselage is an actual physical problem which has been set up in a physically-consistent manner and of which we can say ab initio that an extreme sensitivity of the matrix to changes in a certain element is highly unlikely. We would like only to have an approximate picture of the shape of the inverse to help us to choose the inversion technique, or decide if we should apply to process of sub-matrix orthogonalisation.

Since we know that the $\mathbf{D}$ of a fuselage is highly dominated by the $\mathbf{D}_r$, we might assume that an inverse of $\mathbf{D}_r$ would give a rough idea of the form of $\mathbf{D}$. If the fuselage is cylindrical and the rings are either the same or possess such properties as to satisfy the general form of Eqn. (V,43), an estimate of the $\mathbf{D}_r^{-1}$ would only require two inversions, one of the order $(t-3) \times (t-3)$ and one in order of $(p-2) \times (p-2)$.

If the fuselage does not satisfy this condition exactly because some rings, for example, violate slightly the form of the basic sub-matrix, it is always possible to replace them by 'Ersatz' rings which have a stiffness of the same order of magnitude, but conforming to the standard sub-matrix. In this procedure engineering sense plays the main part.

V-e 3 The N-Number

After using the idealization of the matrix described under V-e 1, this number is in our case, and for a single cell fuselage

$$N_b(\mathbf{D}) \simeq \frac{1}{(p-2)(t-3)} \, N(\mathbf{D}) \quad N(\bar{\mathbf{D}}_r^{-1}) \qquad (V,59)$$

or

$$N_b(\mathbf{D}) \simeq \frac{1}{(p-2)(t-3)} \, N(\bar{\mathbf{D}}_r) \, N(\bar{\mathbf{D}}_r^{-1}) \qquad (V,59a)$$

where $\overline{D}_r$ is the 'Ersatz' D_r .
It is probably more sensible to use (IV 59a) since it has a more consistent physical meaning, i.e., we find the N-number for a certain matrix of which we know the exact inverse, and then say that this number would give an idea of the N-number for the actual D .
For the $\nabla_{(m)}$ matrix we find

$$\begin{aligned} N^2(\nabla_{(m)}) &= m.(2)^2 + 2.(m-1)(-1)^2 \\ &= 6m-2 \end{aligned} \tag{V,60}$$

and

$$\begin{aligned} N^2(\nabla_{(m)}^{-1}) &= 2 \sum_{j=1}^{M} \sum_{i=1}^{j} \frac{[1+(i-1)a][1+(m-j)a]}{a^2[2+(m-1)a]^2} \\ &\quad - \sum_{i=1}^{m} \frac{[1+(i-1)a][1+(m-i)a]}{a^2[2+(m-1)a]^2} \\ &= \frac{m[a^2(m^3-2m^2+11m+2)+6am(3m-1)+12]}{12a^2[2+(m-1)a]^2} \end{aligned} \tag{V,61}$$

Thus the number will be

$$\begin{aligned} Nb(\nabla_{(m)}) &= \frac{1}{m} \frac{m[a^2(m^3-2m^2+11m+2)+6am(3m-1)+12](6m-2)}{12a^2[2+(m-1)a]^2} \\ &= \frac{(3m-1)[a^2(m^3-2m^2+11m+2)+6am(3m-1)+12]}{6a^2[2+(m-1)a]^2} \end{aligned} \tag{V,62}$$

becoming infinite when α is zero, or when $a = -\frac{2}{(m-1)}$. This seems to be a measure which can be applied with ease, and which does point to any undesirable magnification in the inverse. The programming involved, once the approximate inverse is obtained, is trivial. This measure is therefore to be recommended.

V-e 4 Again for a Fuselage we write

$$m\bar{b}(\mathbf{D}) \simeq m\bar{b}(\mathbf{D}_r) \simeq m\bar{b}(\bar{\mathbf{D}}_r)$$
$$= (p-2)(t-3)\; m(\bar{\mathbf{D}}_r) \quad m(\bar{\mathbf{D}}_r^{-1}) \tag{V,63}$$

The programming work necessary to obtain the maximum value in a matrix is again standard.

In order to apply this second measure to the ∇_m matrix, we need the value of the maximum elements in ∇_m and ∇_m^{-1}

$$m(\nabla_m) = 2 \qquad a < 1$$
$$= (1+\alpha) \qquad a > 1 \tag{V,64}$$

The maximum element of ∇_m^{-1} is $\nabla^{-1}_{\frac{m}{2},\frac{m}{2}}$ for m even and $\nabla^{-1}_{\frac{m+1}{2},\frac{m+1}{2}}$ for m odd.

Thus for m even

$$m(\nabla_m^{-1}) = \frac{[1+(\frac{m}{2}-1)a][1+(m-\frac{m}{2})a]}{\alpha[2+(m-1)a]}$$
$$= \frac{[2+(m-2)a][2+ma]}{4a[2+(m-1)a]} \tag{V,65}$$

and for m odd

$$m(\nabla_m^{-1}) = \frac{\left[1+\left(\frac{m+1}{2}-1\right)a\right]\left[1+\left(m-\frac{m+1}{2}\right)a\right]}{a\left[2+(m-1)a\right]}$$

$$= \frac{\left[2+(m-1)a\right]\left[2+(m-1)a\right]}{4a\left[2+(m-1)a\right]}$$

$$= \frac{\left[2+(m-1)a\right]}{4a} \qquad (V,65a)$$

From these we obtain the following table for $m(\nabla)$

$m(\nabla_m)$	$a \leqslant 1$	$a \geqslant 1$
m even	$\frac{m\left[2+(m-2)a\right]\left[2+ma\right]}{2a\left[2+(m-1)a\right]}$	$\frac{(1+a)m\left[2+(m-2)a\right]\left[2+ma\right]}{4a\left[2+(m-1)a\right]}$
m odd	$\frac{m\left[2+(m-1)a\right]}{2a}$	$\frac{(1+a)m\left[2+(m-1)a\right]}{4a}$

(V,66)

So whilst we see that when a approaches zero, the m-number becomes infinite, it is not the case when a is equal to $-\frac{2}{m-2}$ but becomes, on the contrary, very small, even equal to zero for m odd. Despite the fact that this result is of little use to us physically, it shows a weakness of the method, namely that it does not always detect a singularity. It is therefore not to be recommended.

V-e 5 The Goldstine - von Neumann Measure

This measure is nothing really but the ratio of the highest to lowest eigenvalues of the matrix. As described under (V-d 6) a guess can be taken at the dominant eigenvector and eigenvalue of the $\mathbf{D}$ of a fuselage by considering the actual or Ersatz $\mathbf{D}_r$. Although this may lead to an adequate guess for the eigenvector corresponding to λ, the reliability of a guess for Λ is very doubtful and has to be verified. It might be, however, reasonable to assume that it will be mainly determined by $\mathbf{D}_L + \mathbf{D}_q$, so the application of our procedure to obtain an appropriate inverse is not to be recommended here. On the other hand, it is perhaps justified to discuss this proposed measure. It is, of course, true that this ratio is significant. If the eigenvectors and eigenvalues of a matrix $\mathbf{A}$ are respectively

$$\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \cdots\cdots \quad \mathbf{V}_i \quad \cdots\cdots \mathbf{V}_m$$

and

$$\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \cdots\cdots \quad \lambda_i \quad \cdots\cdots \quad \lambda_m$$

then the eigenvectors of the inverse $\mathbf{A}^{-1}$ will be the same, but the eigenvalues are, however, the reciprocals of those of the original matrix. We know that we can then write

$$\mathbf{A} = \sum_{i=1}^{m} \lambda_i \mathbf{V}_i \mathbf{V}_i^t \qquad (V,67)$$

and

$$\mathbf{A}^{-1} = \sum_{i=1}^{m} \frac{1}{\lambda_i} \mathbf{V}_i \mathbf{V}_i^t \qquad (V,68)$$

In Eqn. (V,67) the term $\lambda_1 \mathbf{V}_1 \mathbf{V}_1^t$ is the most dominant and that of $\lambda_m \mathbf{V}_m \mathbf{V}_m^t$ the least significant, and probably appears only in the last few figures from the right. In the inverse, the situation is reversed, and the smaller λ_m is in the first place, the more significant its reciprocal becomes in $\mathbf{A}^{-1}$.

What is more, this eigenvalue and its associated eigenvector are the most sensitive to any change in the elements of the matrix. This means of course, a greater change in $\mathbf{A}^{-1}$. The uncertainty with this method lies in the sensitivity of this last eigenvalue. However, if we assume that we have the largest and smallest eigenvalues we can also safely assume that we have the corresponding eigenvectors. We can then form

$$(\mathbf{A})_m = \lambda_m \mathbf{V}_m \mathbf{V}_m^t \tag{V,69}$$

and

$$(\mathbf{A}^{-1})_m = \frac{1}{\lambda_m} \mathbf{V}_m \mathbf{V}_m^t \tag{V,70}$$

If we also have an approximate inverse we can form the matrix in which the typical element

$$G_{ij} = \text{smaller of} \begin{cases} f - \log_{10} A_{ij} + \log_{10} (A)_{m\,ij} \\ f - \log_{10} (A^{-1})_{ij} + \log_{10} (A^{-1})_{m\,ij} \end{cases} \tag{V,71}$$

where f number of figures used in the computation.

This matrix gives the maximum accuracy one can obtain in the inverse in decimal figures if we consider λ_m as the only determining factor. However, for some elements the determining factor is probably some other eigenvalue. This method can be further extended if we know a few eigenvalues from each end of the scale and the corresponding eigenvectors.

V-e 6 The Factor μ

This is, of course, a very accurate measure. Applying it to the ∇_m we find

$$\mu (\nabla_m) = \frac{2^{(m-2)} \; (1+a)^2}{a [2 + (m-1) a]} \qquad (V,72)$$

The expression becomes infinite for the two critical values of a. However, it is surprising that it gets bigger also very quickly with m due to the presence of an exponent of m in the numerator. So while the highest element in the determinant is more or less proportional to 2^m, the determinant itself is only proportional to m. The increase of value here of the factor μ with increasing size of the matrix is therefore a true indication of the inevitable sensitivity to size which is, as it seems, a property of finite difference equations. This criterion is undoubtedly very accurate and takes care of all cases. The difficulty lies in the time required to compute a determinant, which is approximately equal to that required to carry out an inversion with one right hand side. It nevertheless is always possible to compute an approximate value for the determinant using (V-d 7), provided that due care is taken in the interpretation of the result. Ill-conditioning results due to the presence of a very small eigenvalue, and this appears as a factor in the determinant, and is not accurately determined by an approximate method. However, a certain indication of the diminishing value of the determinant as well as the influence of size may still be present, so that it may still be worth while to compute an approximate estimate of the determinant. It may be of interest here to mention that none of the other measures detects this sensitivity of the finite difference matrix to the inverse of size in quite the same way as this criterion does, with the exception of the Goldstine - von Neumann measure which cannot be explicitly tested.

V-e 7 The Improvement of the Basic System in the Fuselage

The analysis of a simple framework with seven unknowns has been worked out as an example to demonstrate the value of the following procedure.

One starts with an ordinary $\mathbf{b}_0$ and $\mathbf{b}_1$ based on the conventional self-equilibrating stress systems. A first solution is obtained, and the compatibility checked. Afterwards the $\mathbf{b}_1$ is successively transformed by the repetitive use of a transformation matrix Π so that

$$\mathbf{b}_1^{(i+1)} = \mathbf{b}_1^{(i)} \, \Pi \tag{V,73}$$

where Π is the (7×7) matrix

$$\underset{(7\times 7)}{\Pi} = \begin{bmatrix} 1 & 0.9 & 0 & 0 & 0 & 0 & 0 \\ 0.9 & 1 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 1 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 1 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 1 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 1 & 0.9 \\ 0 & 0 & 0 & 0 & 0 & 0.9 & 1 \end{bmatrix} \tag{V,74}$$

which serves the sole purpose of mixing the self-equilibrating systems so much together that they become more and more linearly dependent. At one particular stage the process suddenly results in a deterioration of the conditioning of the $\mathbf{D}$ combined with an inferior compatibility check. As soon as this appears the inferior results are fed back in again, i.e. $\mathbf{b}_0$ is replaced by the calculated $\mathbf{b}$ and the process repeated.

The method works, giving an excellent convergence to the final correct results. However, if the ill-conditioning goes beyond a certain point, the results can no longer be improved by the procedure. This is due perhaps to the fact, that the representation of the physical system becomes incorrect. However, from our point of view the procedure is only interesting as far as it is a proof that a better ab initio approximation to the solution helps greatly in obtaining a more accurate solution. This fact which is by no means new, and which has been repeatedly mentioned by various authers derives from the insignificance of the modification required to obtain the final solution in comparison with the starting value. Thus if in

$$\mathbf{b} = \mathbf{b}_0 + \mathbf{b}_1 \mathbf{X}$$

the $\mathbf{b}_0$ matrix is correct to seven figures, and the $\mathbf{b}_1\mathbf{X}$ only to three, the final results will only be correct to three figures if the second term were of the same order of magnitude as the first. If it is 10^{-4} times smaller, then the original accuracy is maintained.

The procedure proposed for the fuselage and problems of similar nature, rests on the fact that $\mathbf{D}_r$ is the dominant part of the total $\mathbf{D}$, as well as that $\mathbf{D}_{or}$ is the dominant part of the total $\mathbf{D}_o$. After $\mathbf{b}_o$, $\mathbf{b}_1$, $\mathbf{f}$, $\mathbf{D}_{or}$ and $\mathbf{D}_r$ have been calculated, a certain $\overline{\mathbf{D}}_r$, a matrix of the type described before, is set up. Its inversion only requires two inversions of the order ($M \times M$) and ($m \times m$) respectively, as well as M^2 multiplications of a matrix ($m \times m$) with a scalar. This $\bar{D}_r^{-1}$ is used now in order to produce an approximation to the b and hence a better guess at the final solution. Hence we calculate

$$\bar{Y} = - \bar{D}_r^{-1} D_{or} \tag{V,75}$$

Note that D_{or} and not D_o total is used. The new approximation to b , which we call b_o^*, is now obtained from

$$b_o^* = b_o - b_1 \bar{D}_r^{-1} D_{or} \tag{V,76}$$

and this is used as a basis for another calculation of D_o and then one proceeds with the calculation as usual.

It should be noted that this method is not an iteration method. If the results are still not satisfactory, a feeding back of the very final results and a complete repetition of the solution is required. If that again does not succeed one has probably to operate with double or treble accuracy. However, it must be stressed that the mere repetition of the inversion using double accuracy is of limited significance. If double accuracy is required one should preferably use it right from the beginning, i.e. already in the formation of the b_o , b_1 and f from initial data, which are themselves also represented in double-length numbers. One of the main principles of the whole philosophy of the fuselage analysis given in Ref. (2) as well as in matrix analysis in general is that the mere use of the machine in order to do the inversion of the matrix is not correct. The machine has to set up everything consistently from the beginning to the end. This is not only dictated by the desirability of automation for the whole calculation but also extends to such questions as the accuracy required in the computation.

V-e 8 Accumulation of Errors

In this chapter up till now the 'inherent' conditioning of a matrix, that is to say the maximum possible accuracy to be obtained in the inversion of a matrix of a certain type using a limited number of figures has been discussed. This maximum accuracy is rarely attainable. The accumulation of errors is a very important additional factor here, which has to be considered. The problem is of a rather statistical nature, and is, therefore, only to be touched upon. We can here study a standard procedure for improving the inverse of a matrix. If instead of A^{-1} we have a matrix B such that

$$A \; B = I + \Delta \tag{V,77}$$

and

$$A = B + B_{\Delta} \tag{V,78}$$

Δ is an error matrix. It follows from Eqn. (V,77)that

$$A \, (B + B_{\Delta}) = I$$

and the standard relations for the iterative improvement of the inverse are

$$B_{\Delta} = - B \; \Delta \tag{V,79}$$

and

$$A^{-1} = B \, (I - \Delta) \tag{V,80}$$

Eqn. (V,80) derives from the simplification that

$$A^{-1} \cong B$$

It would be interesting to investigate a limiting case where the iteration does not converge, namely when

$$A \quad B \, (I - \Delta) = I + \Delta$$

Thus

$$\Delta^2 = \Delta \qquad (V,81)$$

If one does not interpret the equation literally it means that if the square of the error matrix is a matrix of the same order, then another error is created which is equally large. This is again quite logical and obvious, and in this case would result if all the elements of Δ were constant and equal to ε where

$$\varepsilon = \frac{1}{m} \qquad (V,82)$$

and m is the size of the matrix Δ. Again this discussion is very qualitative and serves merely to point out the existence of this side to the problem. What Eqn. (V,82) says in effect is that the bigger the matrix the more accuracy is required, again a well-known fact.

APPENDIX A

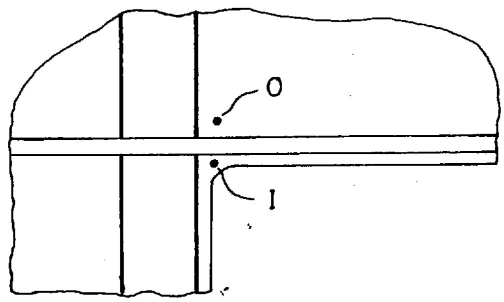
RESULTS OF FUSELAGE COMPUTATIONS INCLUDING COMPARISONS WITH EXPERIMENTAL RESULTS

In this appendix we present the results of the analysis of a single-cell fuselage including the effects of cut-outs and modifications together with a comparison with experimental results measured on a full scale model at Imperial College of Science and Technology.

The preliminary Fig. (A-i) and the two tables (A-i) and (A-ii) give all necessary geometrical data required for the regularized fuselage. Four different cut-out cases, termed A, B, C and D are detailed in Figs.(A-ii) to (A-v). Since the removal of a fuselage panel results in the reduction of the skin contribution to the neighbouring flanges and rings, the necessary modifications are given in tabular form beside each sketch.

Following that, the actual results (Figs. (A-1) to (A-13)) are plotted and the experimental points introduced. These include normal stresses in the flanges, shear stresses in the panels and ring bending stresses. The drawings comprise examples of the regularized fuselage as well as of the various cut-out cases. Five different loading patterns are considered, and sketches are made beside all the drawings giving the nature of the applied loads.

On the whole, we observe excellent agreement between the computed and measured results. However, one should perhaps note that the flange normal stresses are experimentally determined at the edge of a cut-out at two different points which we term "I" and "O" (see sketch on next page). The value at "O" always shows remarkable accuracy, while that at "I" is somewhat high. This can be explained by the presence of a stress concentration at the corner of the cut-out. Another source of slight disagreement is the non-consideration of the effects of the ring lateral bending.



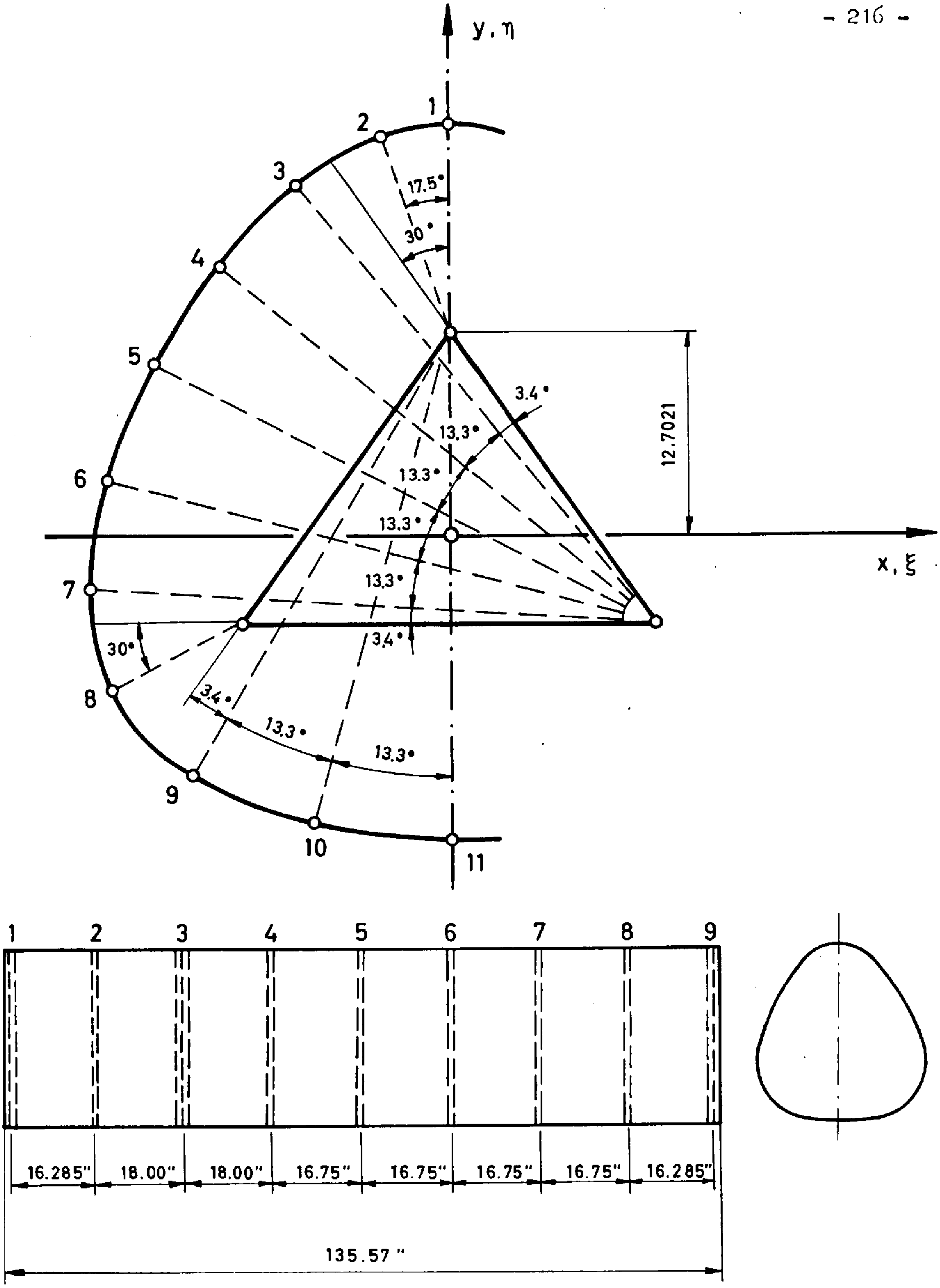


Fig.(A-i) Geometry of the analysed fuselage

Table A-i Effective Flange Area (B)

Stringer \ Station	1	2 - 8	9
1	0,2295	0.2715	0.2295
2	0.0481	0.0962	0.0481
3	0.2624	0.3373	0.2624
4	0.3630	0.3785	0.3630
5	0.3063	0.4017	0.3063
6	0.2831	0.3786	0.2831
7	0.3806	0.4755	0.3806
8	0.2817	0.3758	0.2817
9	0.3613	0.3771	0.3613
10	0.2831	0.3786	0.2831
11	0.3064	0.4019	0.3064
-	-	-	-
20	0.0481	0.0962	0.0481

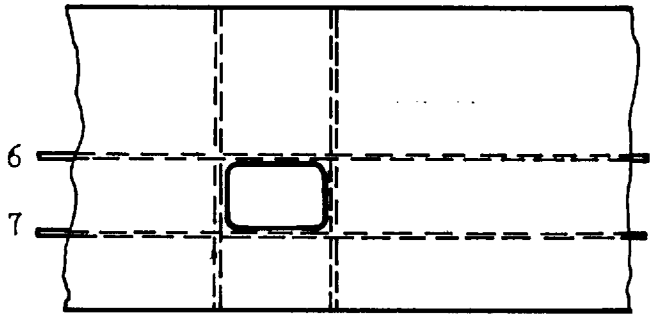
Wall-thickness constant and equal to 0.0248"

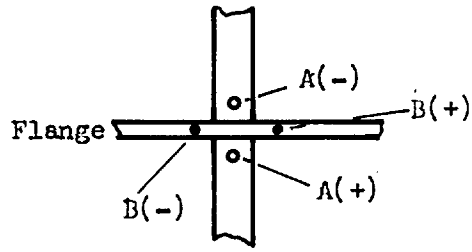
	All Rings		Light Rings No. 2, 4, 5, 6, 7 and 8					Heavy Rings at ends No. 1 and 9					Heavy Ring No. 3				
Stringer No.	x	y	ξ	η	A	C	I	ξ	η	A	C	I	ξ	η	A	C	I
1	0	23.320	0	23.331	0.402	0.169	0.144	0	23.094	1.359	0.484	0.894	0	23.133	1.435	0.484	0.933
2	-3.343	23.305	-3.196	22.839	"	"	"	-3.125	22.613	1.359	0.484	0.894	-3.137	22.650	1.435	0.484	0.933
3	-7.231	21.297	-6.961	20.839	"	"	"	-6.831	20.691	1.359	0.484	0.894	-6.852	20.724	1.435	0.484	0.933
4	-13,102	16.362	-12.746	16.026	"	"	"	-12.574	15.864	1.359	0.484	0.894	-12.602	15.890	1.435	0.484	0.933
5	-17.681	10.208	-17.257	9.963	"	"	"	-17.052	9.845	1.359	0.484	0.894	-17.086	9.864	1.435	0.484	0.933
6	-20.721	3.166	-20.252	3.025	"	"	"	-20.026	2.957	1.359	0.484	0.894	-20.063	2.968	1.435	0.484	1.502
7	-22.059	-4.387	-21.571	-4.416	"	"	"	-21.114	-4.360	1.448	0.572	1.439	-21.175	-4.360	1.524	0.572	5.343
8	-20.628	-11.910	-20.205	-11.665	"	"	"	-19.196	-10.887	1.793	0.918	5.116	-19.254	-10.929	1.869	0.918	8.128
9	-14.829	-16.910	-14.610	-16.473	"	"	"	-13.865	-14.984	1.954	1.080	7.788	-13.901	-15.057	2.031	1.079	8.128
10	-7.619	-19.527	-7.506	-19.052	"	"	"	-7.123	-17.432	1.954	1.080	7.788	-7.142	-17.510	2.031	1.079	8.128
11	0	-20.415	0	-19.927	"	"	"	0	-18.262	1.954	1.080	7.788	0	-18.343	2.031	1.079	8.128
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	3.343	23.305	3.196	22.839	0.402	0.169	0.144	3.125	22.613	1.359	0.484	0.894	3.137	22.650	1.435	0.484	0.933

Table [illegible], Co-ordinates and ring section properties

Fig. A-ii Cut-out Case A

Station: 4 5

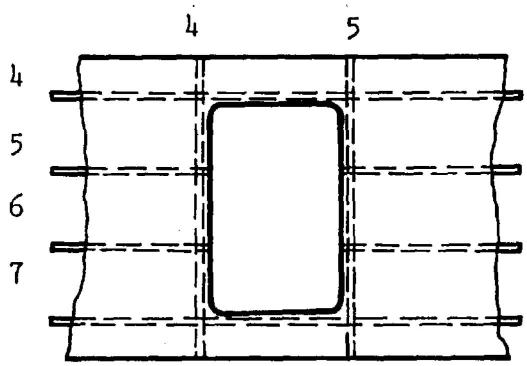




Flange:

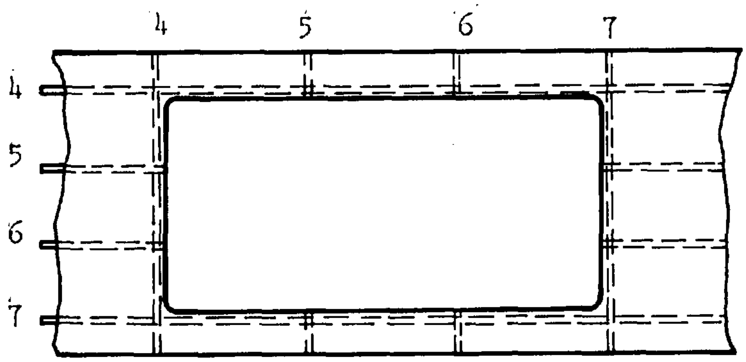
	Flange Stations			Ring Stations 4	Ring Stations 5
A		6	−	0.4023	0.4023
			+	0.3552	0.3552
		7	−	0.3552	0.3552
			+	0.4023	0.4023
C		6	−	0.1692	0.1692
			+	0.1692	0.1692
		7	−	0.1692	0.1692
			+	0.1692	0.1692
I		6	−	0.1441	0.1441
			+	0.1308	0.1308
		7	−	0.1308	0.1308
			+	0.1441	0.1441
B		6	−	0.3786	0.3041
			+	0.3041	0.3786
		7	−	0.4755	0.4010
			+	0.4010	0.4755

Fig. A-iii Cut-out Case B



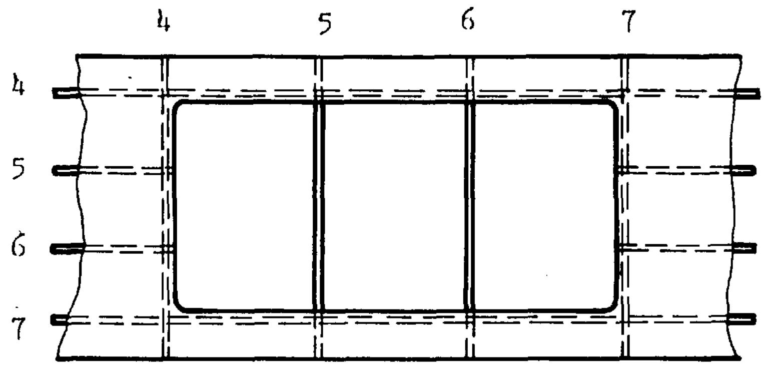
	Flange Stations			Ring Stations 4	Ring Stations 5
A		4	–	0.4023	0.4023
			+	0.3552	0.3552
		5	–	"	"
			+	"	"
		6	–	"	"
			+	"	"
		7	–	"	"
			+	0.4023	0.4023
C		4	–	0.1692	0.1692
			+	"	"
		5	–	"	"
			+	"	"
		6	–	"	"
			+	"	"
		7	–	"	"
			+	0.1692	0.1692
I		4	–	0.1441	0.1441
			+	0.1308	0.1308
		5	–	"	"
			+	"	"
		6	–	"	"
			+	"	"
		7	–	"	"
			+	0.1441	0.1441
B		4	–	0.3785	0.3042
			+	0.3042	0.3785
		7	–	0.4755	0.4010
			+	0.4010	0.4755

Fig. A-iv   Cut-out Case C



				Ring Stations 4	5	6	7
A	Flange Stations	4	−	0.4023			0.4023
			+	0.3552			0.3552
		5	−	"			"
			+	"			"
		6	−	"			"
			+	"			"
		7	−	"			"
			+	0.4023			0.4023
C		4	−	0.1692			0.1692
			+	"			"
		5	−	"			"
			+	"			"
		6	−	"			"
			+	"			"
		7	−	"			"
			+	0.1692			0.1692
I		4	−	0.1441			0.1441
			+	0.1308			0.1308
		5	−	"			"
			+	"			"
		6	−	"			"
			+	"			"
		7	−	"			"
			+	0.1441			0.1441
B		4	−	0.3785	0.3042	0.3042	0.3042
			+	0.3042	0.3042	0.3042	0.3782
		7	−	0.4755	0.4010	0.4010	0.4010
			+	0.4010	0.4010	0.4010	0.4755

Fig. A-v Cut-out Case D



Flange Stations			Ring Stations 4	5	6	7
A	4	−	0.4023	0.4023	0.4023	0.4023
		+	0.3552	0.3081	0.3081	0.3552
	5	−	"	"	"	"
		+	"	"	"	"
	6	−	"	"	"	"
		+	"	"	"	"
	7	−	"	"	"	"
		+	0.4023	0.4023	0.4023	0.4023
C	4	−	0.1692	0.1692	0.1692	0.1692
		+	"	"	"	"
	5	−	"	"	"	"
		+	"	"	"	"
	6	−	"	"	"	"
		+	"	"	"	"
	7	−	"	"	"	"
		+	0.1692	0.1692	0.1692	0.1692
I	4	−	0.1441	0.1441	0.1441	0.1441
		+	0.1308	0.1136	0.1136	0.1308
	5	−	"	"	"	"
		+	"	"	"	"
	6	−	"	"	"	"
		+	"	"	"	"
	7	−	"	"	"	"
		+	0.1441	0.1441	0.1441	0.1441
B	4	−	0.3785	0.3042	0.3042	0.3042
		+	0.3042	0.3042	0.3042	0.3782
	7	−	0.4755	0.4010	0.4010	0.4010
		+	0.4010	0.4010	0.4010	0.4755

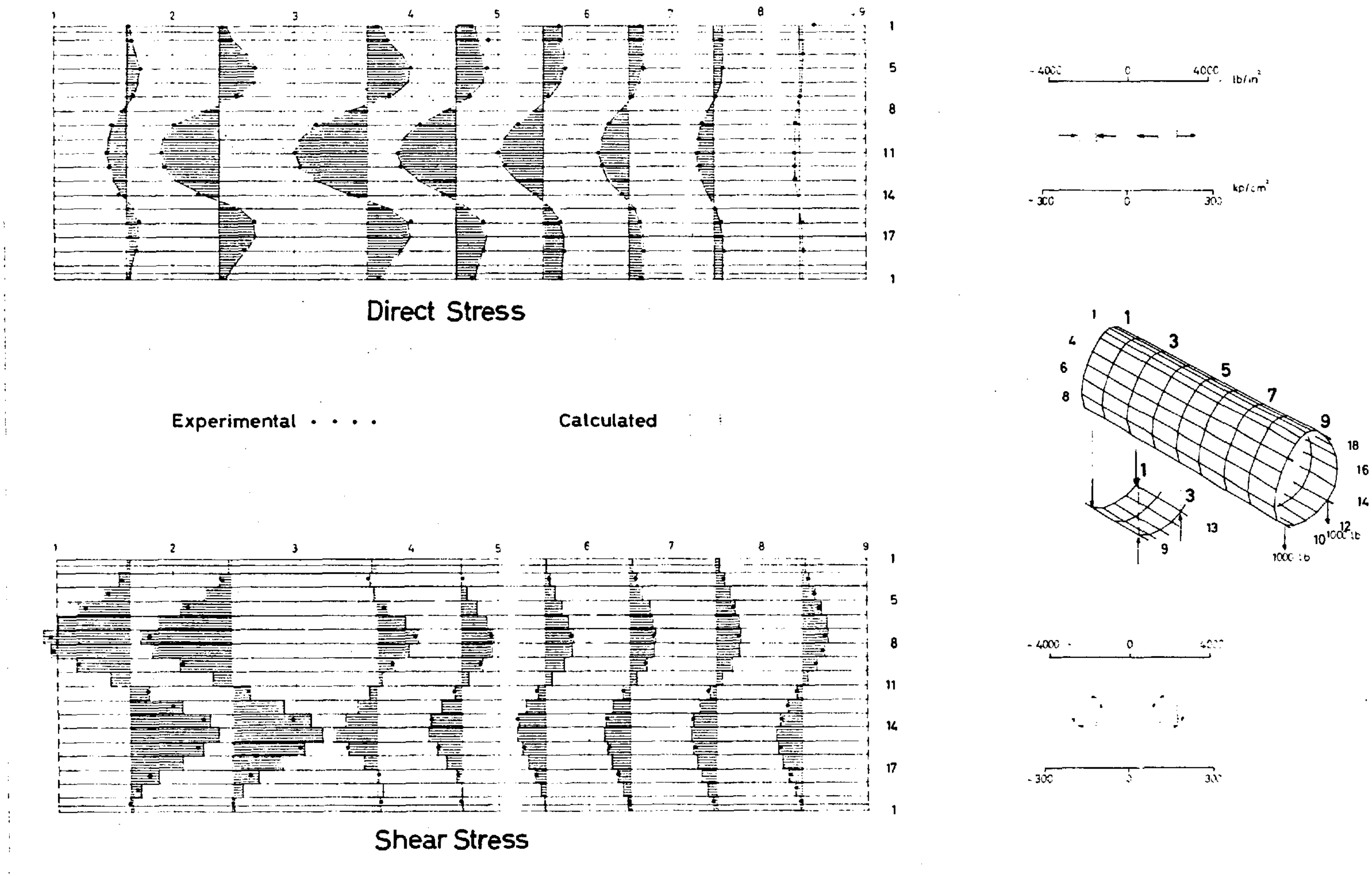


Fig. A-1. Stress distribution in cover. Loading case I.

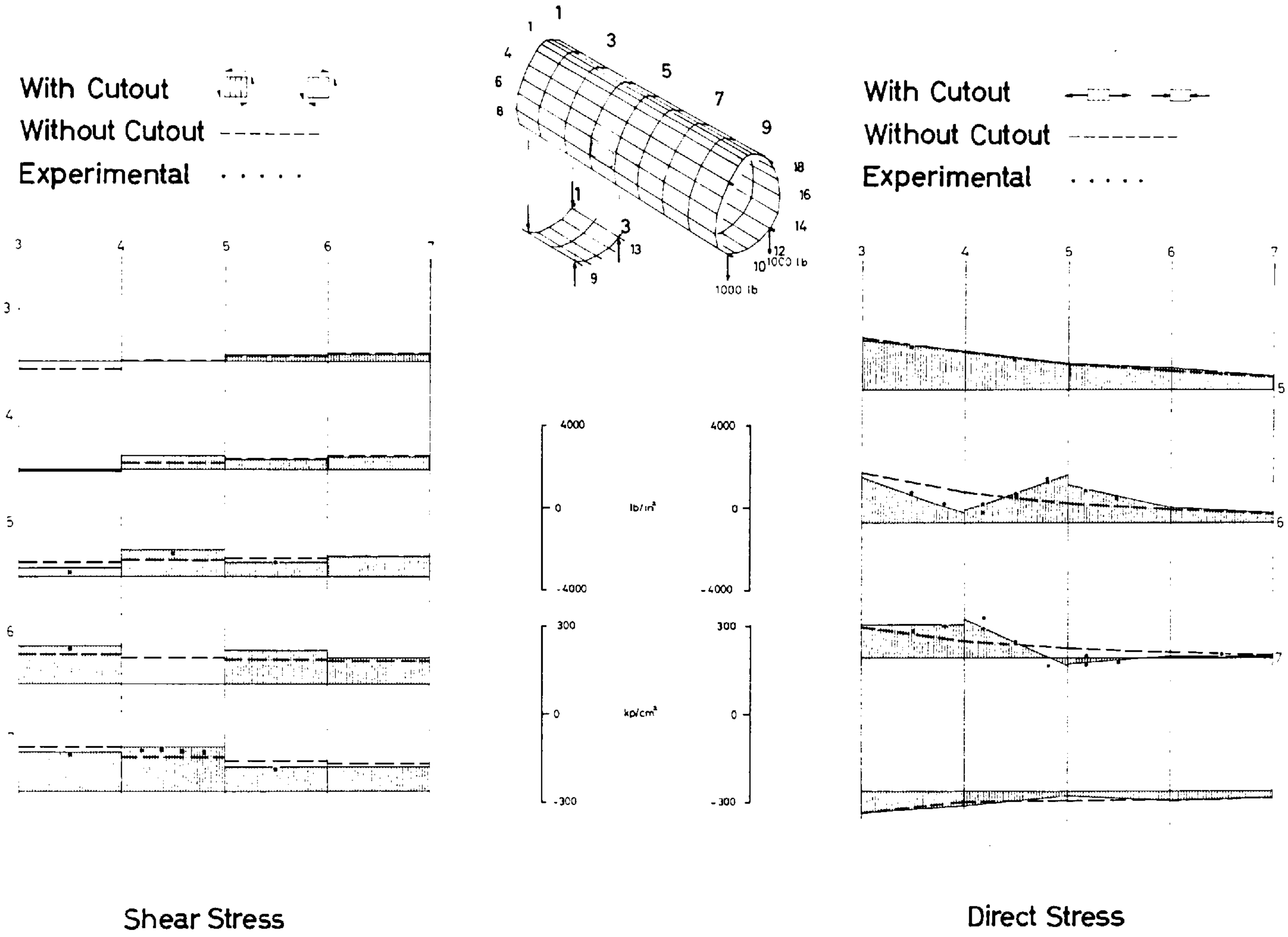


Fig A-2. Stresses in cover around cut-out A. Loading case I

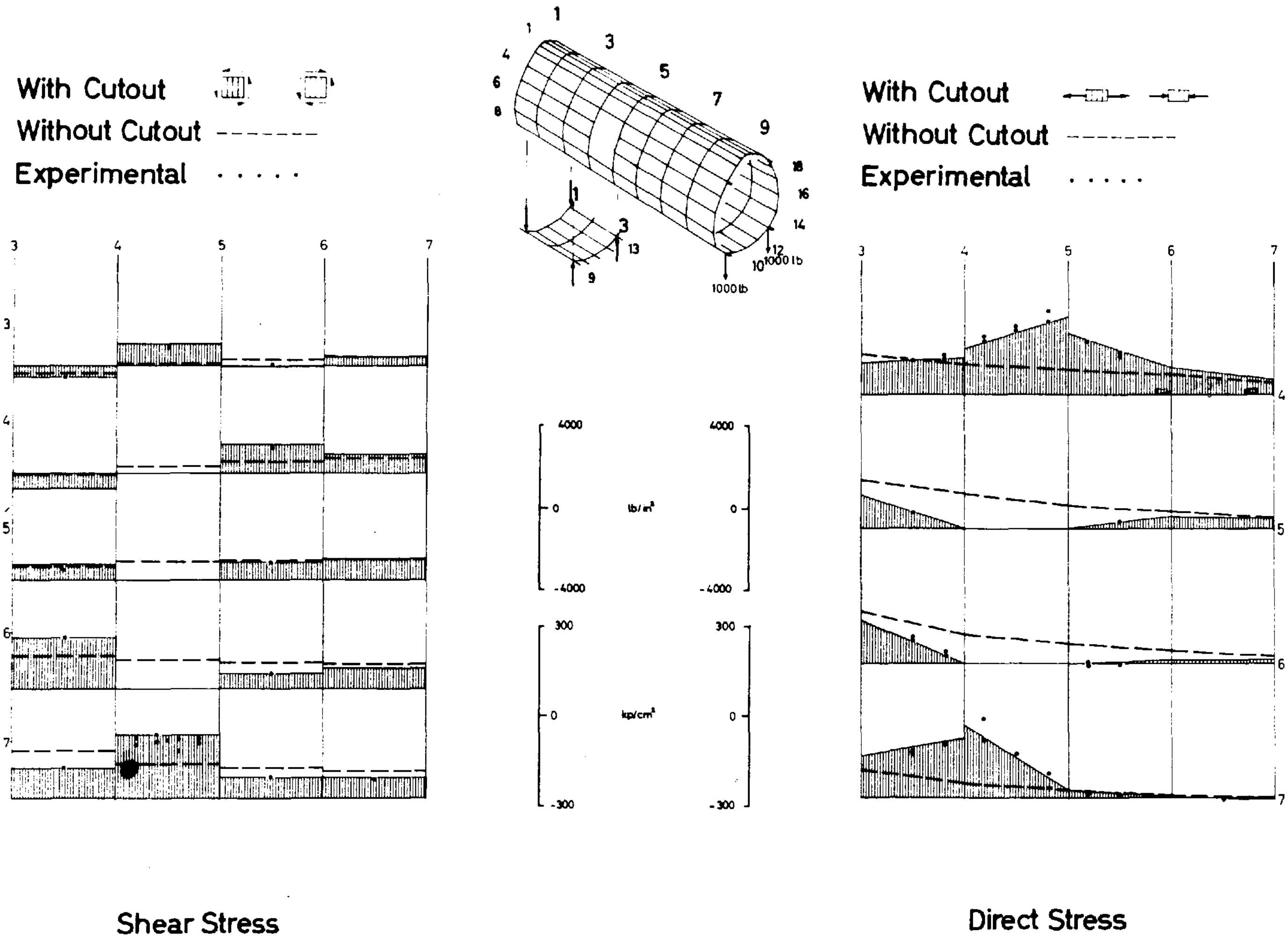


Fig. A-3. Stresses in cover around cut-out B. Loading case I.

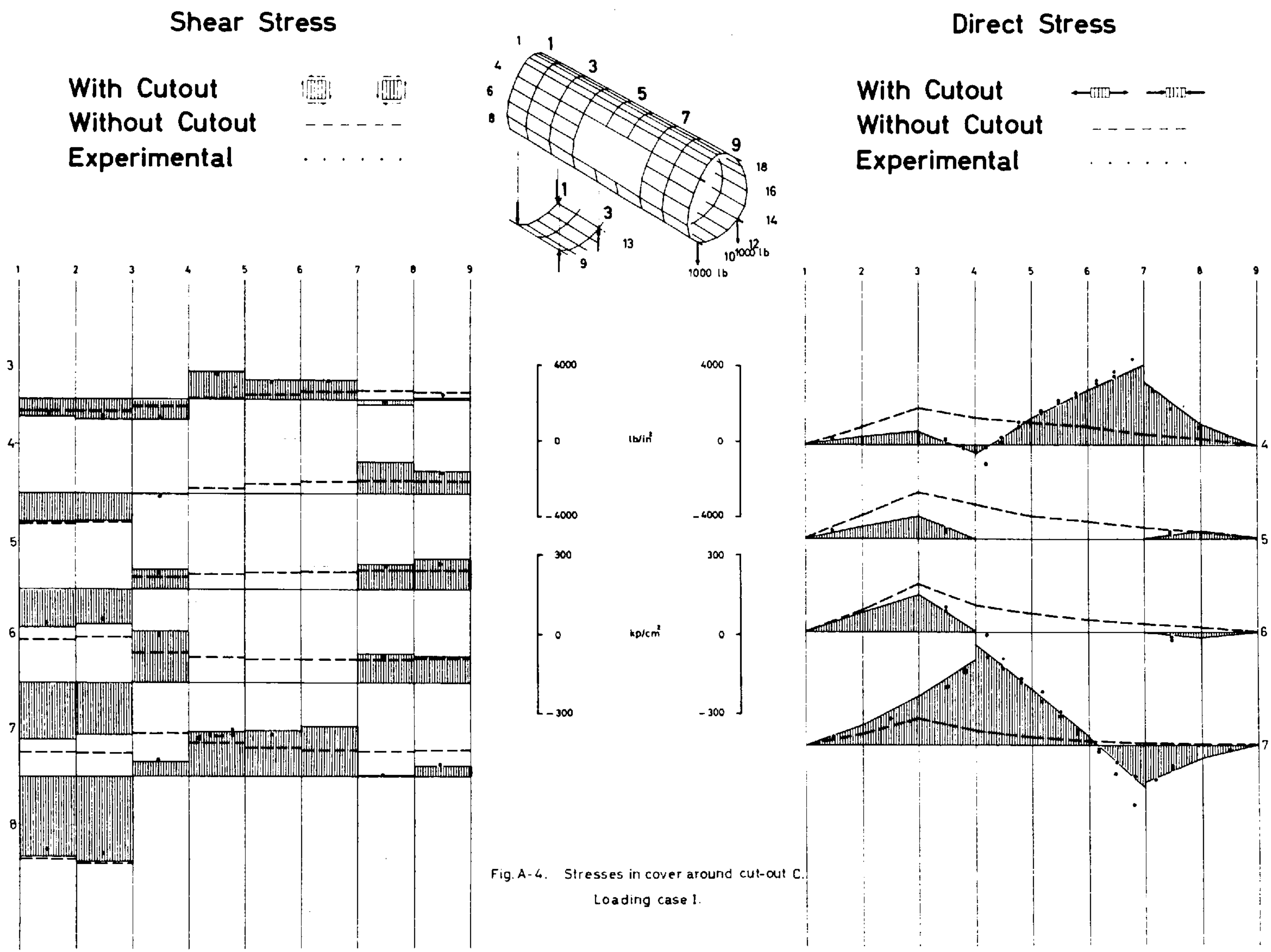


Fig. A-4. Stresses in cover around cut-out C.
Loading case I.

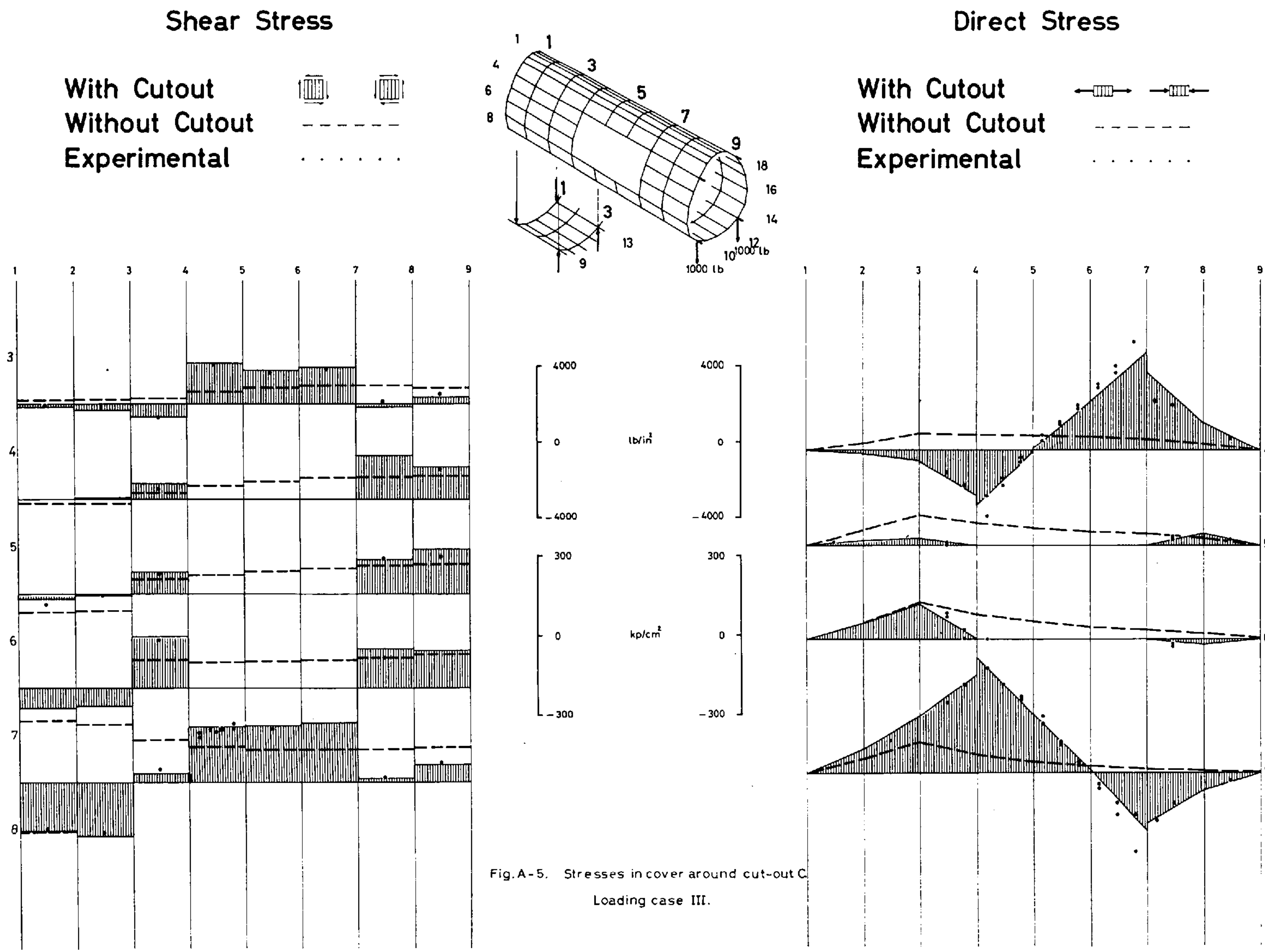


Fig. A-5. Stresses in cover around cut-out C

Loading case III.

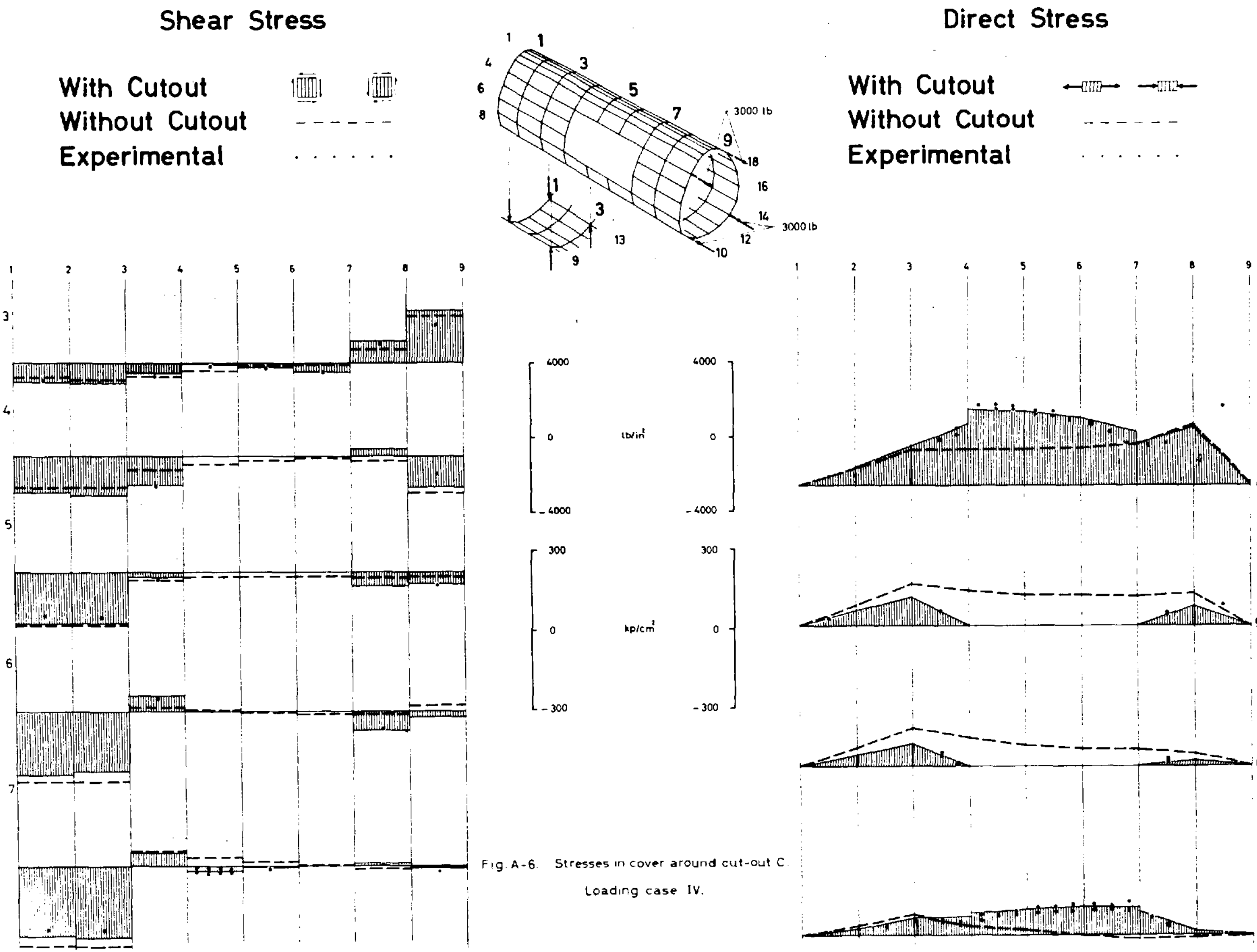


Fig. A-6. Stresses in cover around cut-out C.

Loading case IV.

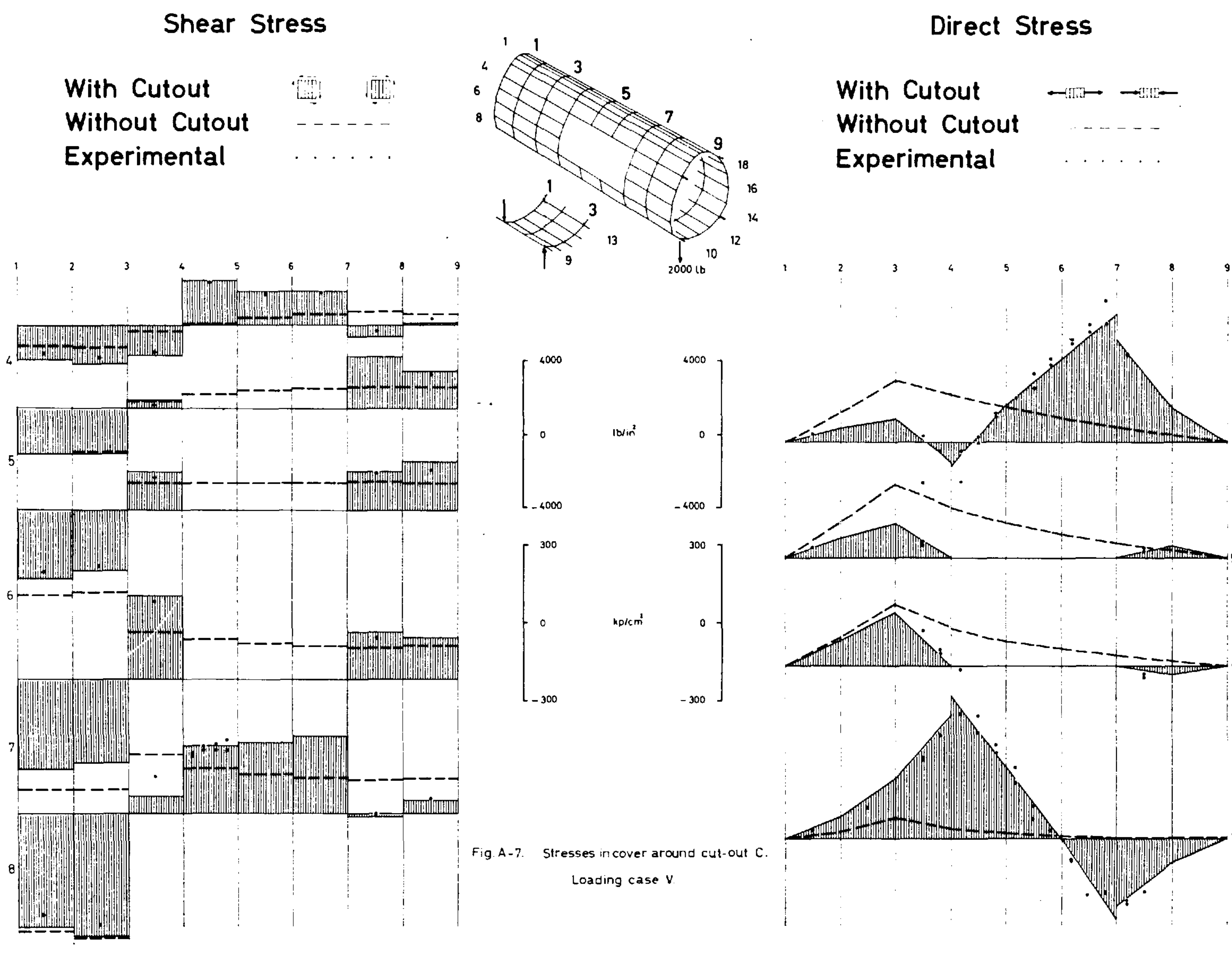


Fig. A-7. Stresses in cover around cut-out C.

Loading case V.

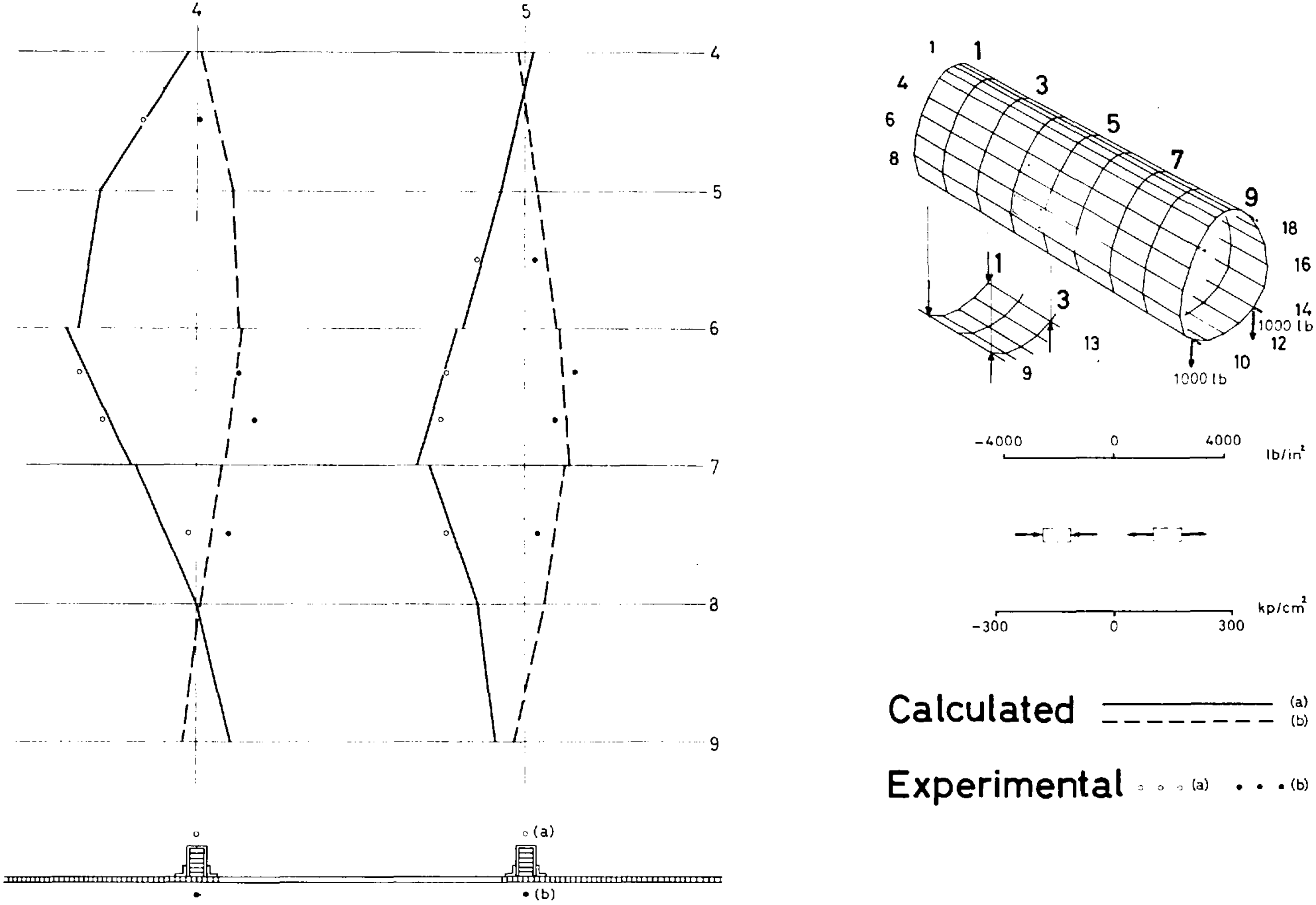


Fig. A-8. Direct stresses in rings at cut-out A. Loading case I.

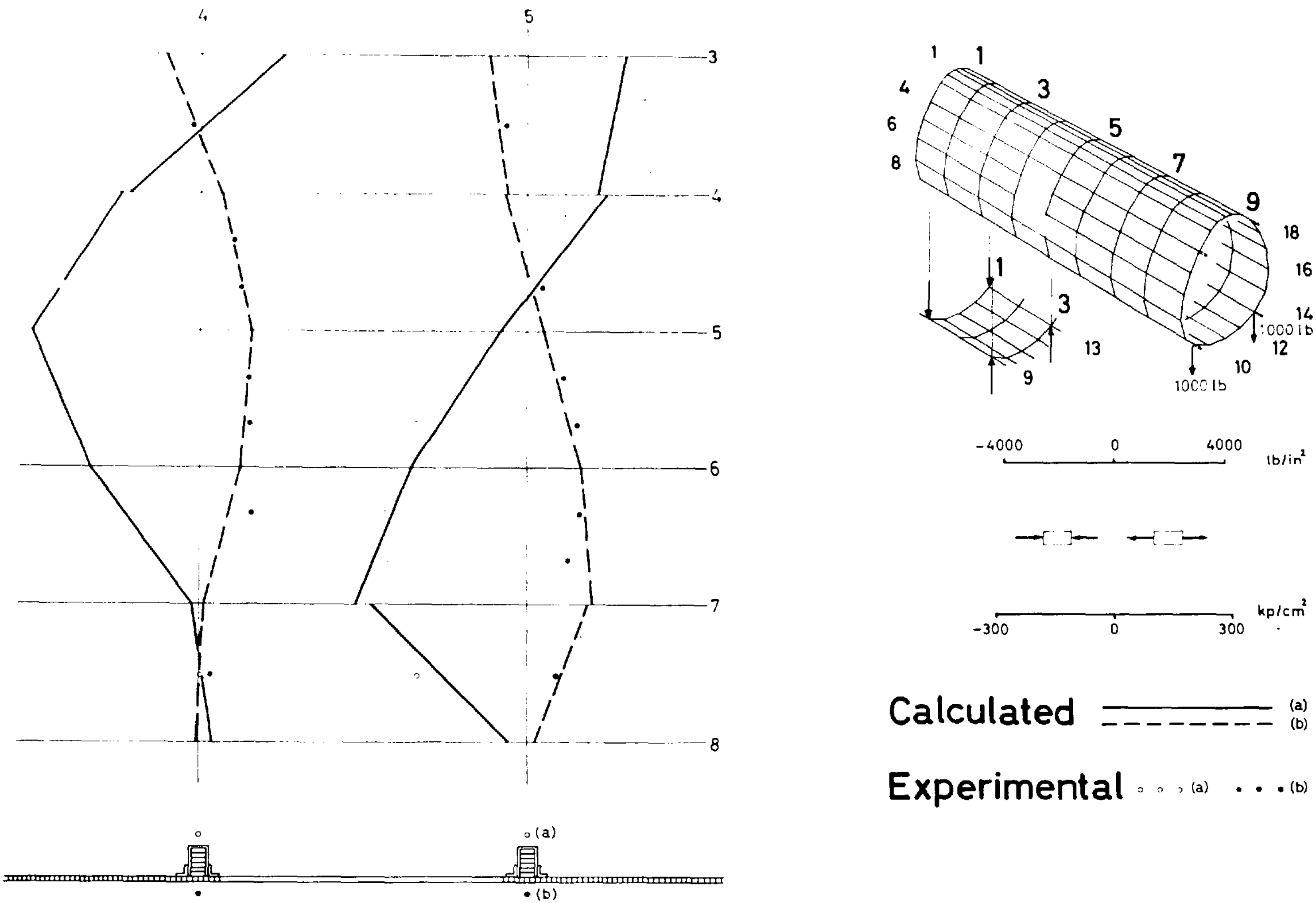


Fig. A-9. Direct stresses in rings at cut-out B. Loading case I.

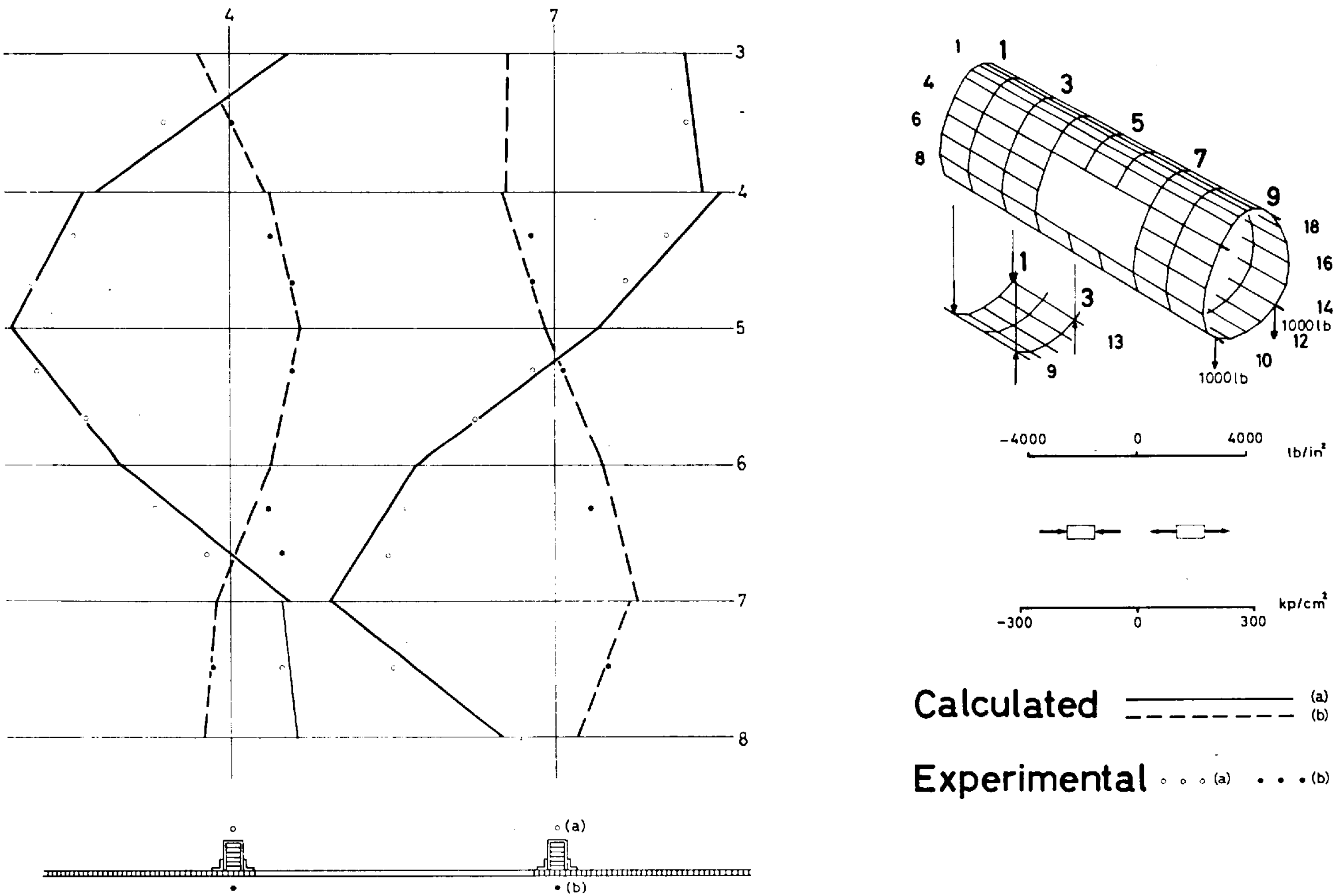


Fig. A-10. Direct stresses in rings around cut-out C. Loading case I.

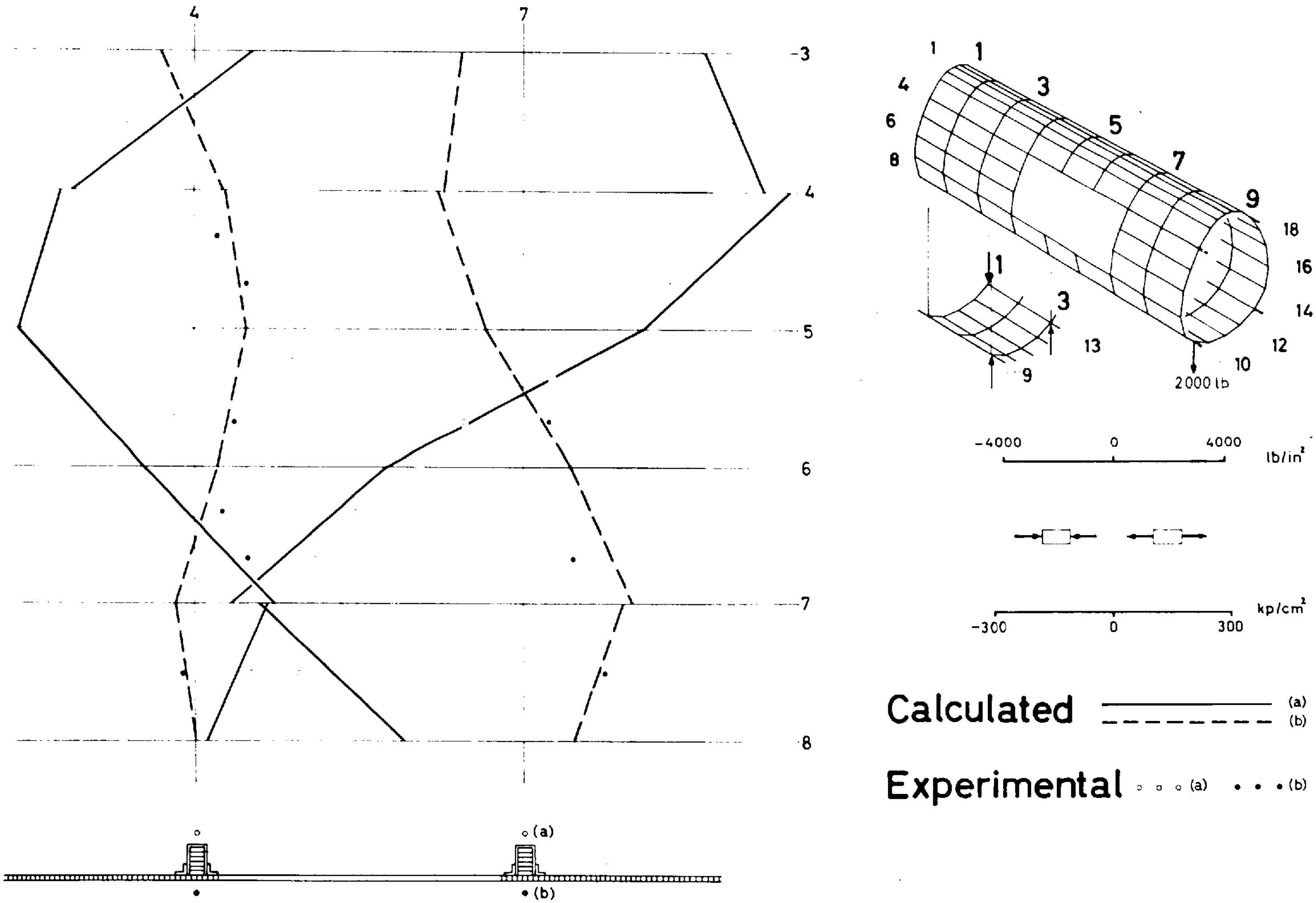


Fig. A-11. Direct stresses in rings at cut-out C. Loading case III.

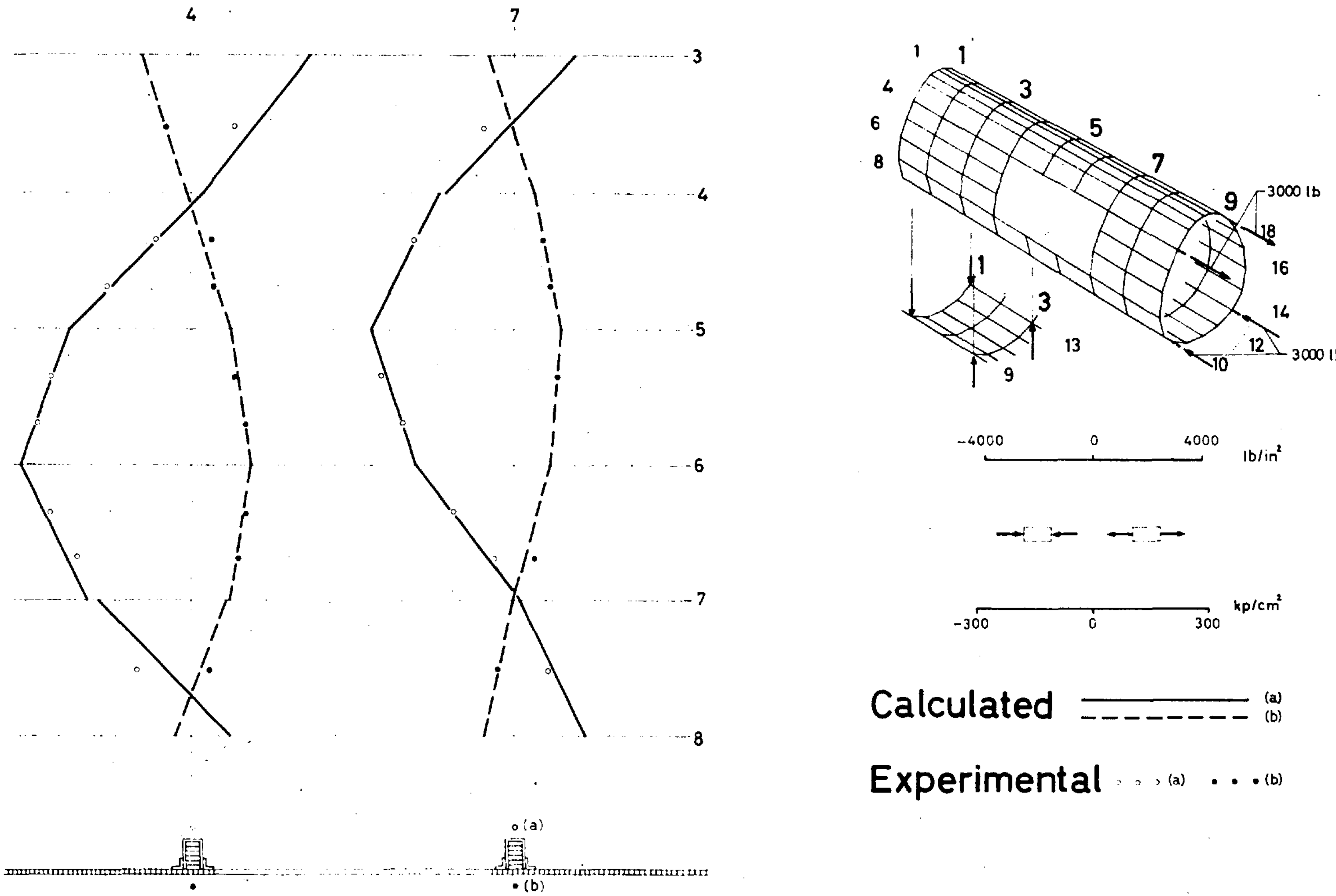


Fig.A-12. Direct stresses in rings at cut-out C. Loading case IV.

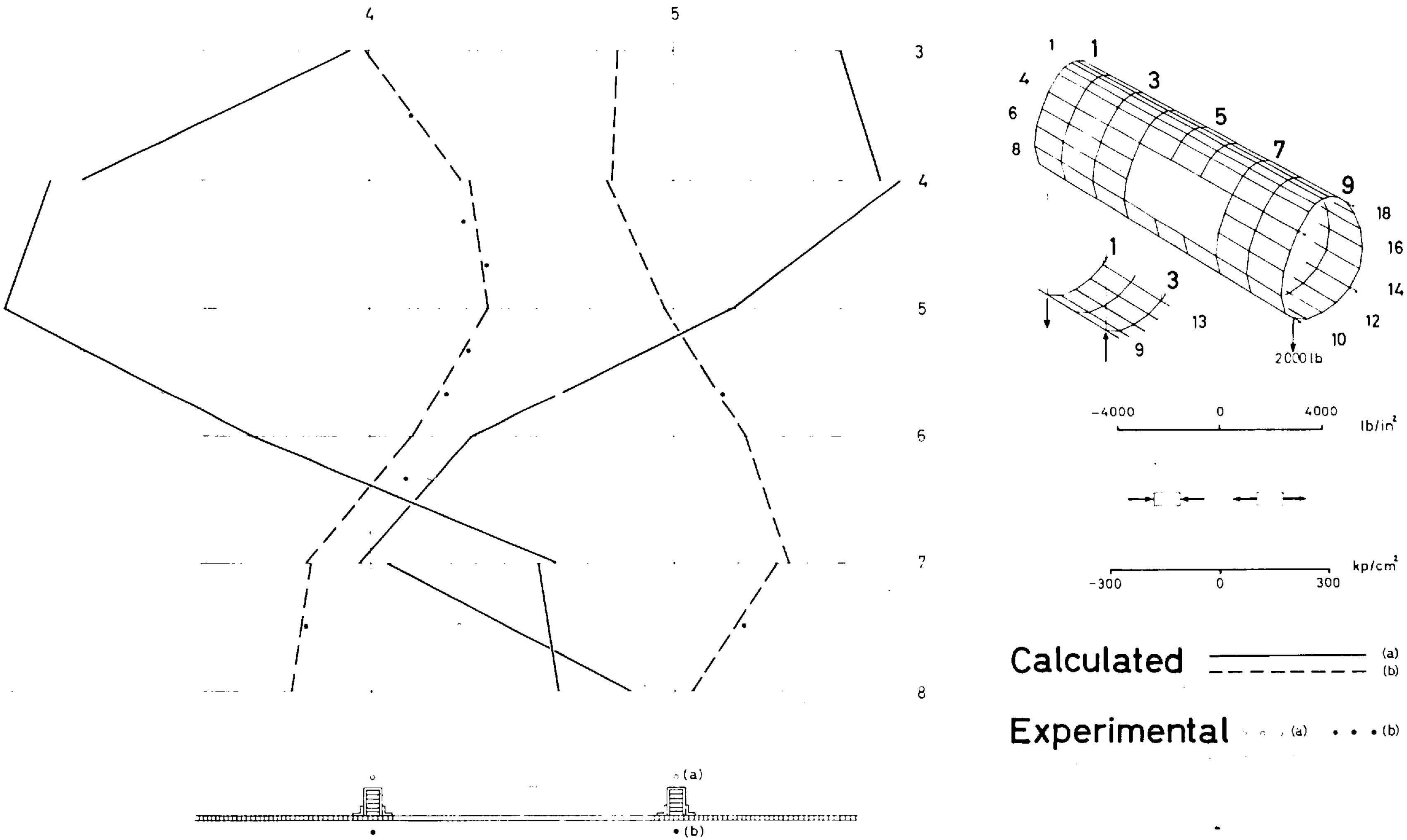


Fig A-13. Direct stresses in rings at cut-out C. Loading case V.

APPENDIX B

SUGGESTIONS FOR A NEW CLASS OF FUNCTIONS

The following Boolean functions are suggested as a start. The exact form of the order is left open, the idea, however, should be the same. For some preliminary applications of the functions the reader may consult Refs. (4) and (5).

Function 1

$$Gb\ (i, j)\ (i_\Delta, j_\Delta)\ (n) \longrightarrow A$$

This function forms a matrix of (+1)'s and zeros whose first element has the co-ordinates (i,j). The matrix contains n non-zero elements whose co-ordinates derive from those of the first by addition of the increments (i_Δ, j_Δ). This is repeated $(n-1)$ times in all.

Example:

i \ j	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	1	0

The order $Gb(2,1)\ (1,2)\ (4) \longrightarrow A$

results in the following matrix being stored in A :

(2,1) (3,3) (4,5) (5,7) (End)

The exact form of the end word is left free, although the word (777 777 777 777) has already been used for similar purposes.

Function 2

Repeat

$$Rb\ (A)\ (i\Delta\Delta,\ j\Delta\Delta)\ (n) \longrightarrow B$$

This order causes a certain Boolean matrix stored at A to be used to form another matrix at the address B by repeating the matrix A n times, each time increasing all co-ordinates by $(i\Delta\Delta, j\Delta\Delta)$

Example:

At A (2,1) (3,3) (4,5) (5,7) (END)

Order $Gb(A)(5,2)$ (3) $\longrightarrow$ B

results in the following Boolean matrix being stored at B .

At B (2,1) (3,3) (4,5) (5,7) (7,3) (8,5) (9,7) (10,9)
(12,5) (13,7) (14,9) (15,11) (END)

Although the result of this particular operation is accidentally sorted, an automatic sorting is recommended after such a formation order. This yields a simplification of programming for subsequent functions.

Dimensions and Numbers of Entries

We observe at this stage that the dimensions of the Boolean matrices have not been mentioned in the orders. On the other hand we may state the dimensions of any formed matrix, which are then stored in a heading preceeding the matrix. This heading may also include the number of 'ones'. This would, of course, be calculated automatically by the code as it forms the Boolean matrices and be used to allocate storage space.

These preliminary ideas on Boolean matrices are to be expanded in a subsequent report.

APPENDIX C

STANDARDISED FORMS FOR ORDERS AND SOME FURTHER REQUIRED NEW FUNCTIONS

Introduction

As a basis this work assumes the existence of a super-matrix scheme which operates with super-matrices. Such a matrix scheme is being developed at present by the "Rechengruppe of the Institut für Statik und Dynamik der Luft-und Raumfahrtkonstruktionen" in Stuttgart.

However, since this work should be presented in a general logical form, which can always be easily translated into such a scheme, whatever final shape the orders assume, certain standard forms for matrix instructions are given in which the programmes may be written. (See Chapter IV).

On the other hand, since a successful computer language is the result of close co-operation between the systems and applied programmers of whom the latter will eventually use the library for the solution of his special problems, it is always one of the results of programming a task using the computer, that certain suggestions are made which result in alterations to the computer language so as to render it more powerful for dealing with the problems in hand. So, in the last chapter, we find some suggestions for new functions dealing with Boolean matrices. This type of matrix first arose during the analysis of a wing by the displacement method (see Refs. 4,5). However, at that time, the function of these matrices and thei: use to perform certain operations was not handled in a sufficiently elegant manner, due to the still experimental nature of the programmes. These matrices proved further to be of great help in a variety of problems hardly related to one another, for example, for modifications and cut-outs in a fuselage, and in plasticity problems. This type of matrices is also used as identification matrices for example in the displacement method. The suggested functions simplify greatly the formation and handling of such matrices, and reduce the operations to their basic mathematical equivalents. The description of regular patterns is easily and logically accomplished, the number of orders required to form a pattern depending upon its 'dimension'. Thus a one-dimensional pattern is formed in one order, a two-dimensional one in two orders and an n -dimensional one in n orders. Different patterns can be superimposed through a simple addition.

Apart from those standard Boolean operations, some new functions which have been found necessary during the calculation, are introduced and described where suitable , although their effect is obvious from their use in the various programmes. Some of them do have a definite mathematical significance, others have none and are only there in order to overcome the unavoidable inflexibility accompanying the automatic nature of the super code. We do believe it possible to extend such a scheme to include functions which enable us to deal with special classes of problems. It is, of course, also obvious that these functions do not occur in the part of the programme where purely mathematical computations are being carried out, but rather in parts where the basic matrices are being formed. In the case of a problem like the fuselage analysis we must repeat that this part is complicated to programme and requires more subtlety than the straightforward computations once the basic matrices are there. This can easily be verified by inspection of the programme described in the main part of this work.

It must be mentioned also that the suggested form of the orders is mainly based on a generalization of those of the Pegasus matrix scheme, since they are in our opinion clear and unambigious.

Representation of the Matrices

The most general way of presenting a matrix of the order mA x nA stored at address A is

(A, mA x nA).

A uniform super-matrix of the super order MA x NA composed of sub-matrices all of the size mA x nA is represented by

(A, MA x NA, mA x nA).

A special case occurs when a matrix is square. Then only one of the dimensions need be mentioned, e.g.

(A, mA)	is of order mA x mA.
(A, MA, mA x nA)	is a uniform square super-matrix with rectangular sub-matrices.
(A,MA x NA, mA)	is a uniform rectangular super-matrix with square sub-matrices.
(A,MA, mA)	is a square super-matrix with square sub-matrices.

A diagonal matrix will be denoted by an inclined stroke after a single dimension. Thus the following matrices or uniform super-matrices are fully or partly diagonal

(A, mA/)
(A, MA/, mA x nA)
(A, MA x NA, mA/)
(A, MA/, mA/).

With a diagonal matrix it is usual to store only the diagonal elements. However, this is a matter of internal organisation of the code.

Another special case is the scalar matrix. Here only one element is stored, let us say, in address A. The matrix is then denoted by

(A). or (A,( ))

The exact dimensions are always interpreted according to the operation and the other matrices involved.

Applying the same for super-matrices, the uniform super-matrix

(A, MA x NA, ( ))

is a rectangular super-matrix whose elements are scalar matrices. So, whereas the super-dimensions are fixed, the dimensions of the sub-matrices are interpreted so as to suit the operation and other matrices involved in it.

The matrix

(A, ( ), mA x nA)

is a scalar super-matrix with an ordinary rectangular matrix as an element.

The last special type of matrices are the Boolean matrices. In this work they are always denoted by placing a " b " before the address, e.g.

(bA, mA x nA)

or (bA, MA x NA, mA x nA)

or (bA, MA x NA, ( ))

or (bA, ( ), mA x nA)

The Functions

We now represent a complete list of the functions assumed to be present in the matrix scheme. The general form of the order is always given and details of operation only when necessary. Special classes of matrices, e.g. diagonal,scalar, will only be used whereever it is necessary to indicate a special use. As stated before, some of these functions derive from the Pegasus scheme, some are based on the idea of a super-matrix code as being developed in the Rechengruppe and a few of these are again a result of a co-operation with systems programmers. Other functions derive, however, mainly from the fuselage problem, but can possibly be used for other problems as well.

1) Transfer matrix to another place

$(A, MA \times NA, mA \times nA) \rightarrow C$

2) Add and Subtract

$(A, MA \times NA, mA \times nA) \pm (B, MB \times NB, mB \times nB) \rightarrow C$

3) Multiply

$(A, MA \times NA, mA \times nA) \times (B, MB \times NB, mB \times nB) \rightarrow C$

4) Divide

$(A, MA \times NA, mA \times nA)^{-1} (B, MB \times NB, mB \times nB) \rightarrow C$

5) Transpose

$TR\,(A, MA \times NA, mA \times nA) \rightarrow C$

6) Clear a matrix (form a zero matrix)

$CL\,(A, MA \times NA, mA \times nA) \rightarrow C$

Conditions which must be satisfied in the various operations are obvious and need only to be mentioned when necessary. For example, we give operations in which an actual matrix is used as a "scalar"

$$(A, (\quad), mA \times nA) \times (B, MB \times NB, mB \times nB) \longrightarrow C$$

where $nA = mB$, and the matrix A is interpreted as $(A, MB \times MB, mA \times mB)$, and so on.

We need still the following functions in order to facilitate the handling of individual elements.

7) Extract Element

EXEL (i,j) $(B, MB \times NB, mB \times nB) \longrightarrow C$

This will cause the sub-matrix $(B_{ij}, mB \times nB)$ to be extracted and stored at C. Also

EXEL (i,j) $(B, mB \times nB) \longrightarrow C$

will cause the element B_{ij} to be stored at C.

8) Extract Diagonal Sub-matrices

EXDISM $(A, MA \times NA, mA \times nA) \longrightarrow C$

where $MA = NA$.

A diagonal super matrix composed of the diagonal elements of A will be stored at C.

9) Extract Diagonal Elements

EXDIEL $(A, MA \times NA, mA \times nA) \longrightarrow C$

where $MA = NA$, $mA = nA$.

A diagonal super-matrix (with diagonal sub-matrices) will be formed from the diagonal elements of the matrix A.

10) Decompose into Elements

DEC (A, MA x NA, mA x nA) ⟶ C

The matrix will be decomposed into its sub-matrices which will be stores at C in a prescribed manner. The super-code might note their addresses, and give them names e.g. A1,1 , A1,2 ....AMA,NA if they are to be stored by rows.

11) Recompose from Elements

REC (A, MA x NA, mA x nA) ⟶ C

This is the opposite of(8). According to the specified dimensions of the uniform super.matrix, the computer will extract the sub-matrices A_{ij} starting from the address A, and store them as a uniform super-matrix in address C.

Then there are functions to be carried out only on the sub-matrices, as for instance

12) Transpose Elements

TREL (A, MA x NA, mA x nA) ⟶ C

This will result in a matrix

(C, MA x NA, nA x mA)

i.e. the elements of which are those of A , but transposed.

13) Sine of Elements

The order SEL (A, mA x nA) ⟶ C

will result in a matrix of the order (mA x nA) stored at C, whose elements are the sines of the elements of A.

The order SEL (A, MA x NA, mA x nA) $\longrightarrow$ C can be interpreted in different ways. Either the sines of the individual elements are calculated, or the sines of the sub-matrices. In the latter case, it would also be logical to have the order

13a) Sine of Matrix

SIN (A, MA x NA, mA x nA) $\longrightarrow$ C

which opens the door to a completely new class of functions which can be used in the solution of more complicated problems than linear systems of equations.

Similarly we have

14) Cosine of Elements

CEL (A, mA x nA) $\longrightarrow$ C

and CEL (A, MA x NA, mA x nA) $\longrightarrow$ C

also

14a) Cosine of Matrix

COS (A,MA x NA, mA x nA) $\longrightarrow$ C

15) Square Root of Elements

SQEL (A, mA x nA) $\longrightarrow$ C

and SQEL (A, MA x NA, mA x nA) $\longrightarrow$ C

These are to be interpreted as (13) and (14). Correspondingly, 15a) Square Root of Matrix

SQRT (A, MA x NA, mA x nA) ⟶ C

Naturally in(13a), (14a) and (15a) the matrices must satisfy certain mathematical conditions, e.g. be square. In (15a) they must also be positive definite.

After this set of functions comes another one which aims at facilitating the manipulation of matrices as well.

16) Modulus of Elements

MODEL (A,MA x NA, mA x nA) ⟶ C

The matrix C will have as elements the moduli of the elements of A .

17) Divide Elements

DIVEL (A,MA x NA, mA x nA) (B,MB x NB, mB x nB) ⟶ C

where MA = MB
NA = NB
mA = mB
nA = nB

This is an element by element division of the matrix A into B

18) Divide Non-Zero Elements

DINZEL (A,MA x NA, mA x nA) (B,MB x NB, mB x nB) ⟶ C

Similar to (17), only that zero elements are not divided, instead the resulting element is made zero.

19) Join Horizontally

JH (A, MA x NA, mA x nA)(B, MB x NB, mB x nB) ⟶ C

This can be represented by

$$C = \begin{bmatrix} A & B \end{bmatrix}$$

Naturally we must have that MA = MB and mA = mB (also nA = nB for a uniform super-matrix).

20) Join Vertically

JV (A,MA x NA, mA x nA) (B, MB x NB, mB x nB) ⟶ C

i.e.

$$C = \begin{bmatrix} A \\ B \end{bmatrix}$$

Again we must have NA = NB and nA = nB (also mA = mB)

21) Join Diagonally

JD (A,MA x NA, mA x nA) (B, MB x NB, mB x nB) ⟶ C

viz.

$$C = \begin{bmatrix} A & \\ & B \end{bmatrix}$$

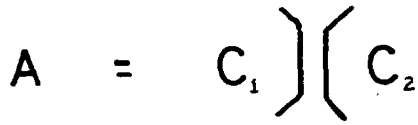
The only conditions necessary, if we want a uniform super-matrix again, are that mA = mB and nA = nB.

The last three functions serve to merge two matrices together and the opposite to these are functions which split a uniform super-matrix into two separate (super)- matrices.

22) Split Horizontally

SH (A,MA x NA, mA x nA) (NA1, NA2) ⟶ C1, C2

Diagrammatically



where naturally enough NA1 + NA2 = NA.

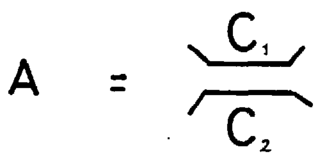
23) Split Vertically

SV (A, MA xNA, mA x nA) (MA1, MA2) $\longrightarrow$ C1, C2

where

MA1 + MA2 = MA

i.e.



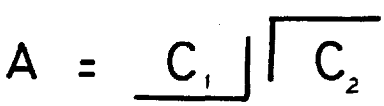
24) Split Diagonally - only applicable for diagonal matrices

SD (A, MA/, mA x nA) (MA1, MA2) $\longrightarrow$ C1, C2

where

MA1 + MA2 = MA.

viz.



Then we need a few orders to re-partition a matrix so as to render an operation involving another matrix possible.

25) Re-partition A as B

REP (A,MA x NA, mA x nA) (B, MB x NB, mB x nB) $\longrightarrow$ C

we need to have

MA x mA = MB x mB

and

NA x nA = NB x nB

The matrix in A will then be re-partitioned in the same manner as B.

26) Re-partition Columnwise

REPCW (A,MA x NA, mA x nA) (B,MB x NB, mB x nB) $\longrightarrow$ C

If(NA) (nA) = (MB) (mB), the matrix in A will be re-partitioned columnwise so that one can pre-multiply the matrix in B by it. Matrix B need not actually exist.

27) Re-partition Row-wise

REPRW (A, MA x NA, mA x nA) (B, MB x NB, mB x nB) $\longrightarrow$ C

Again if (NA)(nA) = (MB) (mB), the matrix B will be re-partitioned row-wise so that it can be pre-multiplied by the matrix in A.

Then we have some orders to convert super-vectors into diagonal super-matrices and vice-versa, thus

28) Diagonalize

DZ (A, MA x 1, mA x nA) $\longrightarrow$ C

or DZ (A, 1 x NA, mA x nA) $\longrightarrow$ C result in the matrices

(C, MA/, mA x nA) or (C, NA/, mA x nA) respectively

29) Diagonal to Column Vector

DCV (A, MA /, mA x nA) ——► C

gives

(C, MA x 1, mA x nA)

30) Diagonal to Row Vector

DRV (A,NA /, mA x nA) ——► C

yields

(C, 1 x NA, mA x nA)

And then similar functions to diagonalize <u>the elements</u> of a super-matrix, or vice versa if these element sub-matrices are vectors.

31) Diagonalize Elements of Row or Column super-matrix

DZEL (A,MA x NA, mA x 1) ——► C

or DZEL (A,MA x NA, 1 x nA) ——► C

result in the matrices

(C, MA x NA, mA/) or (C, MA x NA, nA/)

respectively.

32) Diagonal Elements to Column Vectors

DELCV (A,MA x NA, mA/) ——► C

gives

(C, MA x NA, mA x 1)

33) Diagonal Elements to Row Vectors

DELRV (A, MA x NA, nA/) ⟶ C

i.e. (C, MA x NA, 1 x nA)

34) Obtain Eigenvalues and Eigenvectors

EIG (A,MA x NA, mA x nA) ⟶ VAL, VEC

where

(MA) (mA) = (NA) (nA)

The full matrix of the eigenvectors is formed and stored at VEC. It is of the order, say (MA x NA, mA x nA). The corresponding matrix of eigenvalues of order, say (MA/,mA/) is placed at VAL.

35) Multiply and Keep Maximum

MKMAX (A,MA x NA, mA x nA) (B, MB x NB, mB x nB) ⟶ C

This is similar to an ordinary matrix multiplication. However, not the sum of the products of the elements of the rows of A with those of the columns of B is stored in C, but rather only the numerically largest element occuring in each summation.

36) Diagonal Normalisation

DIANOR (A, MA x NA, mA x nA) ⟶ C

where

MA = NA and mA = nA

This is best described by giving the typical element in C, viz.

$$C_{ij} = \frac{A_{ij}}{\sqrt{A_{ii}\ A_{jj}}}$$

where $C_{ii} = +1$

Referring to Appendix (B), which includes the two typical orders for the formation of Boolean matrices, we now define the corresponding orders in form. Any other operation involving them - say multiplication with an ordinary matrix or of two Boolean matrices - is written in the normal way. To distinguish, however, between a Boolean matrix and an ordinary one, we preceed the address of the former always by a "b". The orders for the formation of the Boolean matrices are

37) Generate Boolean Matrix

Gb (i,j) (i Δ , jΔ) (n) $\longrightarrow$ C

38) Repeat Boolean

Rb (bA) (i,j) (n) $\longrightarrow$ C

39) Shift Origin of Boolean Matrix

SOb (bA) (i,j) $\longrightarrow$ C

In addition to these we have a group of orders aiming at facilitating the inspection of a matrix through a minimum of output.

40) Matrix Spectrum

MASPEC (A,MA x NA, mA x nA) (A,B) (Δ +)

or MASPEC (A,MA x NA, mA x nA) (A,B) (Δ x)

In both cases the machine prints out statistical information about the order of magnitude of the elements of the matrix between the limits A and B at intervals Δ . To demonstrate this, we give the expected output in both cases.

In the first case

MASPEC (A) +

Limits	No. of elements
$A < Aij \leqslant A +$	a1
$A + \Delta < Aij \leqslant A + 2$	a2
$A + 2\Delta < Aij \leqslant A + 3$	a3
$\vdots$	$\vdots$
$B - \Delta < Aij \leqslant B$	ap

Total number of elements examined = a

In the second case

MASPEC (A) x

Limits	No. of elements
$A < Aij \leqslant A\Delta$	a1
$A\Delta < Aij \leqslant A\Delta^2$	a2
$A\Delta^2 < Aij \leqslant A\Delta^3$	a3
$\vdots$	$\vdots$
$B\Delta^{-1} < Aij \leqslant B$	ap

Total number of elements examined = a

4.1) Column Spectrum of Matrix

This is the same as before, only that the information is given in detail for each column. Thus

COSPEC (A,MA x NA, mA x nA) (A,B) (Δ +)

results in the output

Limits \ No. of col.	1	2	3	4		$(N_A n_A)$
$A < A_{ij} \leqslant A+\Delta$	a_1	b_1	c_1	d_1	....	z_1
$A+\Delta < A_{ij} \leqslant A+2\Delta$	a_2	b_2	c_2	d_2	....	z_2
$A+2\Delta < A_{ij} \leqslant A+3\Delta$	a_3	b_3	c_3	d_3	....	z_3
	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$B-\Delta < A_{ij} \leqslant B$	a_p	b_p	c_p	d_p	....	z_p

The output of the order

COSPEC (A,MA x NA, mA x nA) (A,B) (Δ x)

can be written down correspondingly.

42) Row Spectrum of a Matrix

ROSPEC (A,MA x NA,mA x nA) (A,B) (Δ +)

or ROSPEC (A,MA x NA, mA x nA) (A,B,) (Δ x)

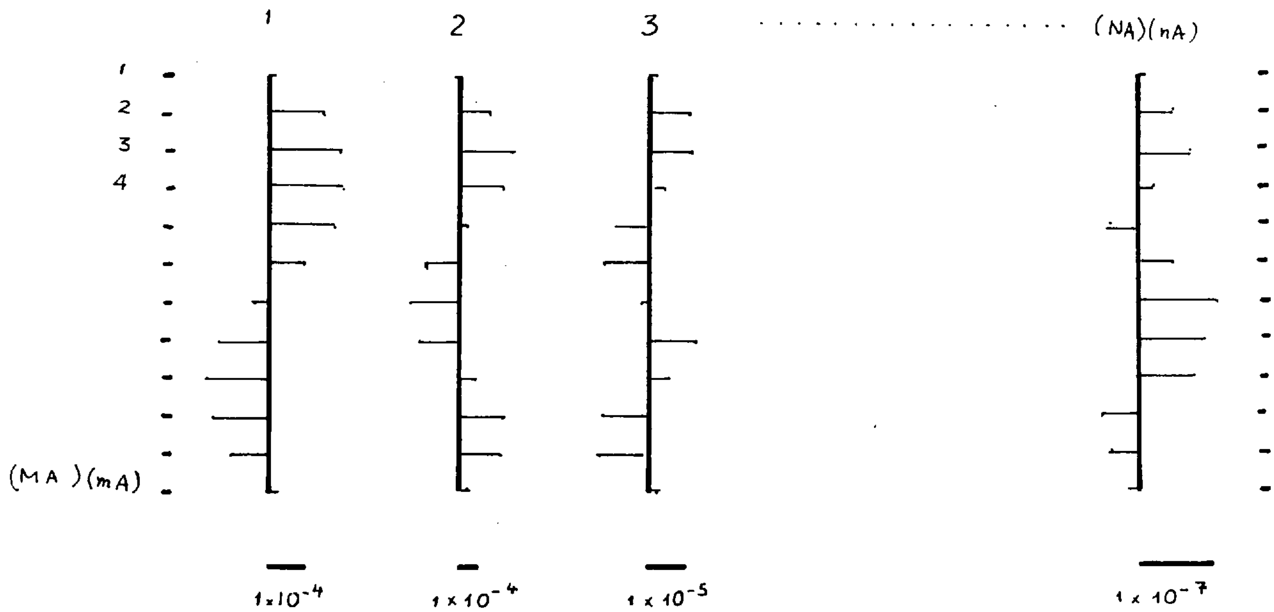
Exactly as in 41), only information about rows is printed out.

43) Plot Columns of Matrix

PLOCOL (A,MA x NA, mA x nA) (D,MA x 1, mA x 1)

The element of the matrix of A will be plotted as co-ordinates measured from bases lying at distances specified by the vector matrix D. The scale can be chosen by the machine itself, and always given beside the plots.

For example



A P P E N D I X D

THE FORMATION OF SPECIAL MATRICES

In this appendix various methods will be discussed by which some of the important standard matrices can be generated on the computer. In the described algorithms, the standard form of instructions described in Appendix (C) will be used. Thus we proceed to the formation of the various matrices.

The Formation of $\mathcal{E}_{\ell+}$, $\mathcal{E}_{\ell-}$ and $\mathcal{E}_{\ell}$

Gb (1,1) (2,1) (p-1) ⟶ ELP

Gb (2,2) (2,1) (p-1) ⟶ ELM

(bEL+ , 2 (p-1) x p) + (bEL- , 2 (p-1) x p) ⟶ EL

It will be assumed that Boolean matrices can either be used as simple matrices, or as super-matrices with sub-matrices which are either $\mathbf{O}$ or $\mathbf{I}$, and have a size automatically adaptable to the sub-matrices of the other matrix involved in the operation.

The Formation of L_+ and L_-

In an exactly similar fashion, the formation of L_+ is carried out by a single order

Gb (1,2) (1,1) (p-1) $\longrightarrow$ LP

similarly for

Gb (1,1) (1,1) (p-1) $\longrightarrow$ LM

The Formation of the Matrices $I_{a(n,1)}$ and $I_{a(n,t)}$

Gb (1,2) (1,1) (n-1) $\longrightarrow$ IA *

Gb (n,1) (-,-) (1) $\longrightarrow$ IA **

(bIA * , n x n, ( )) + (bIA ** , n x n, ( )) $\longrightarrow$ IA(n,1)

It must be mentioned that the matrix $I_{a(n,t)}$ is exactly the same, since the elements of the resulting matrix are assumed to be scalar matrices adaptable to the size of the matrices otherwise involved.

The Formation of ε_{T-} and ε_{T+}

Each is formed as a result of a single order. Thus

Gb (1,2) (1,2) (p-1) $\longrightarrow$ ETM

and Gb (1,1) (1,2) (p-1) $\longrightarrow$ ETP

The Formation of ε_m

Gb (1,2) (1,1) (p-2) $\longrightarrow$ EM

The Formation of $\alpha_{(t)}$ and $\beta_{(t)}$

These again are straightforward

```
Gb  (1,2) (1,1) (t-1) ——► C (t)*
Gb  (t,1) (-,-) (1)   ——► C (t)**
(bC (t)* , t, ( )) + (bC (t)** ,t, (  )) ——► C (t)
TR (bC (t), t, (  ))  ——► C (t) t
Gb (1,1) (1,1) (t)    ——► I (t)
(bI (t), t, (  )) - (bC (t) t, t, (  )) ——► ALF (t)
(bI (t), t, (  )) + (bC (t) t, t, (  )) ——► BET (t)
```

The Formation of $\mathcal{E}_{fl}$

```
Gb (1,2) (2,2) (p-1) ——► Efl1
Gb (2,1) (2,2) (p-1) ——► Efl2
(bEfl1, 2 (p-1), (  )) + ( bEfl2 ,2 (p-1),( ))——►Efl
```

The Formation of $\bar{\mathcal{E}}_{l-}$, $\bar{\mathcal{E}}_{l+}$ and $\bar{\mathcal{E}}_{l}$

```
Gb (2,1) (2,1) (p-1) ——► EBlN
Gb (1,1) (2,1) (p-1) ——► EBlP
(bEBlN, 2(p-1) x (p-1), (  )) + (bEBlP , 2 (p-1) x (p-1), (  )) ——►EBl
```

The Formation of $\sum_{(t)}$

It is evident here that the matrix can only be formed by a standard loop. Thus we have to use symbolic orders for counting which can be easily translated with any other machine.

$$
\sum_{(t)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ \vdots \\ \cdot \\ t \end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & \cdots & & & t \end{array} \\
\left[\begin{array}{cccccccc}
1 & 0 & 0 & 0 & \cdots & & & 0 \\
1 & 1 & 0 & 0 & \cdots & & & 0 \\
1 & 1 & 1 & 0 & \cdots & & & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & & & \vdots \\
1 & \cdots & & 1 & 1 & 0 & 0 \\
1 & \cdots & & 1 & 1 & 1 & 0 \\
1 & \cdots & & 1 & 1 & 1 & 1
\end{array}\right]
\end{array} .
$$

```
     CL (b SIG, t, (   )) ——→ SIG
     A1 = 1
     n1 = t

a)   Gb (A1,1) (1,0) (n1) ——→ SIGDEL
     (bSIG, t, (   )) + (bSIGDEL, t, (   )) ——→ SIG
     n1  = n1  - 1
     A1  = A1  + 1
     Jump to (a) if n1 ≠ 0
     Stop                        ( Σ  ready)
```

The Ω_l Matrix

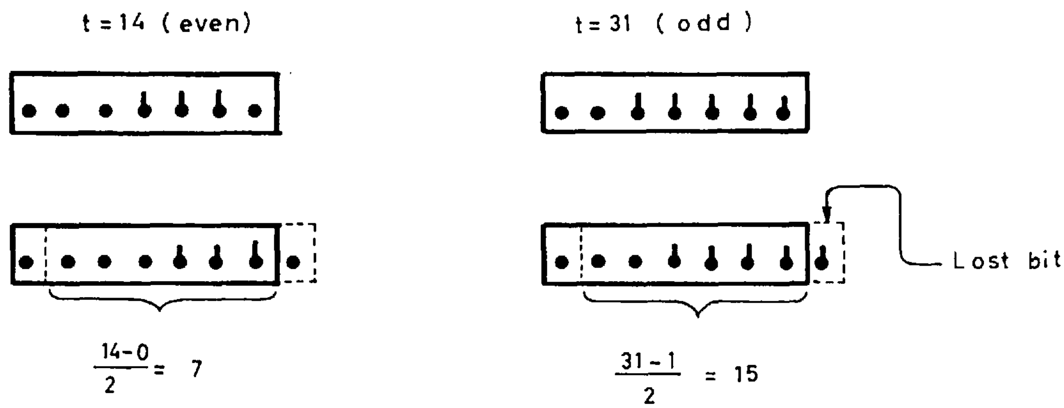
In order to form this matrix automatically we have to revert to a certain stratagem, which we can best understand after considering the previous steps in detail.

First of all we observe that the matrix Ω_l of order $[t \times (t - 3)]$ can be split horizontally in two separate sub-matrices Ω_{ls} and Ω_{lc} (see Ref. (2)) of order $[t \times (s-1)]$ and $[t \times (c-1)]$ respectively.
s and c are given by

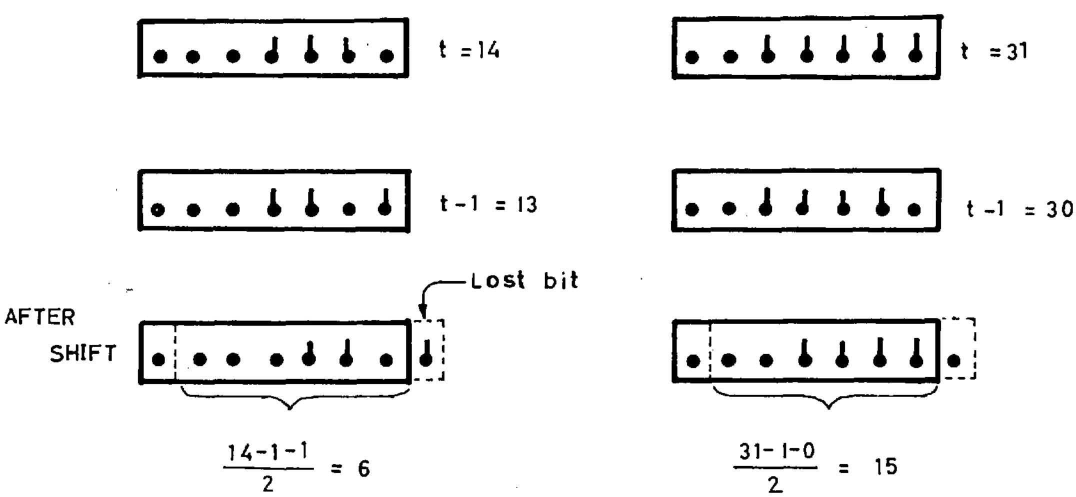
	t even	t odd
c	$t/2$	$(t-1)/2$
s	$(t-2)/2$	$(t-1)/2$

when a flange lies on the y-axis. There is an alternative case when no flange lies on the y-axis, but apart from being different only in detail, it is recommended always to place the first flange on the y-axis, since this can always be done, for the sake of simplicity of the programme.

In order to avoid having two alternatives in the calculation of c, the register containing the flange number t as a binary integer is simply shifted logically one place to the right. If t is odd, the least significant bit simply disappears and the result is the same as subtracting one and then dividing by two. This is best seen in the following example



In the case of s, we subtract one first from the register to obtain $(t-1)$, and then shift it once to the right; thus, for the same number of flanges



Having established c and s for this case, we give it also briefly for the case when no flange lies on the y-axis. Thus

	t even	t odd
c	$(t-2)/2$	$(t-1)/2$
s	$t/2$	$(t-1)/2$

So c is formed as s in the previous case and vice versa. And now, in order to establish Ω_l

$$\Omega_l = \left[\Omega_{ls} \quad \Omega_{lc} \right]$$

we have to form the two component sub-matrices.

The procedure is similar for the two matrices, and therefore it is enough to describe the formation of Ω_{ls} in detail. The matrix Ω_{ls} can be written as

$$\Omega_{ls} = \left[S_2 \; S_3 \cdots\cdots S_r \cdots\cdots S_s \right]$$

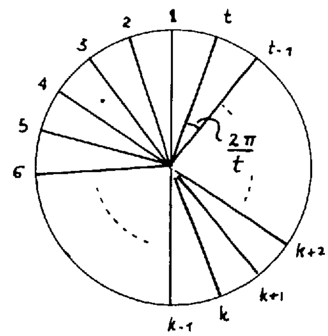
where

$$S_r = \left\{ \sin (k-1) \left(\frac{2\pi r}{t} \right) \right\} \qquad k = 1 \rightarrow t$$

The typical element of Ω_{ls} will then be

$$\Omega_{ls_{i,j}} = \sin \frac{2\pi (i-1)(j+1)}{t}$$

In order to form the j^{th} column therefore, and considering the diagram showing a cyclicly symmetrical fuselage with the circumference divided into t



intervals, we find that we have to take the 'sines' of every $(j+1)^{th}$ angle going around the fuselage in a continuous cyclic manner. For example

$$\Omega_{1s_{\cdot j}} = S_{j+1} = \Big\{ \sin 0 \quad \sin \frac{2\pi}{t} \times (j+1) \quad \sin \frac{2\pi}{t} 2(j+1) \quad \ldots\ldots\ldots\ldots \quad \sin \frac{2\pi}{t} (t-1)(j+1) \Big\}$$

As we see, we have to go around the circle many times. Since it is obvious, however, that all the values that could possibly be chosen must coincide with one of the basic values

$$\sin 0 \;,\quad \sin \left(\frac{2\pi}{t}\right) \;,\quad \sin 2\times\left(\frac{2\pi}{t}\right) \quad \ldots\ldots \quad \sin (t-1)\times\frac{2\pi}{t} \;,$$

we have only to form these and simulate the cyclic symmetry by repeating them as many times as necessary to cover the "swept region" and then choosing every $(j+1)^{th}$ element i.e. t elements in all.

Having thus stated our procedure in general, we describe it now in the form of precise logical instructions.

(a) We form the matrix

$$\underset{(t\times 1)}{\mathbf{p}} = \{ \quad 0 \quad +1 \quad +2 \quad +3 \quad \cdots\cdots\cdots + (t-1)\}$$

Since we have already described the formation of the(Boolean) matrix $\Sigma_{(t)}$ we immediately find that

$$\mathbf{p} = \left[\Sigma_{(t)} - I_{(t)}\right] \mathbf{e}_{(t)}$$

b) Multiplying $\mathbf{p}$ with the scalar $2\pi/t$ which has to be stored beforehand, and then taking the sines of the elements we obtain the column matrix

$$\mathbf{S}_1 = \text{SINEL}\left\{\left(\tfrac{2\pi}{t}\right)\mathbf{p}\right\}$$

c) In order to obtain the column j of the matrix Ω_{ls} we first repeat the column $\mathbf{S}_1$ $(j+1)$ times by a simple post-multiplication through a unit row of the order $(1 \times (j+1))$

$$\underset{(t\times(j+1))}{\mathbf{S}_{1e}} = \underset{(t\times 1)}{\mathbf{S}_1} \quad \underset{(1\times(j+1))}{\mathbf{e}^t}$$

d) Choosing every $(j+1)^{th}$ value is done as follows:- First of all the inflexibility of the representation of a matrix by the automatic matrix scheme has to be overcome by using the special function mentioned before to decompose the matrix in its individual elements, that is to re-consider it as a string of numbers $t \times (j+1)$ long. Assuming that the matrix is stored column-wise, it is then 're-composed' again, without changing the order of the numbers, into a $(j+1) \times t$. In effect this means the string of numbers representing the sines of the angles taken around the circular cross-section $(j+1)$ times, is divided into t groups each $(j+1)$ long. In other words, the first word in each group is one of the sought values and the first row of the new matrix is our required column. To obtain it explicitly, we have then to transpose the new matrix, and then extract the first column of the transpose. Moreover we have to add the column in the proper place in the Ω_{ls} . This is best done by post-multiplying by a Boolean matrix of the order $[(j+1) \times s]$ whose elements are all zero except for the element $(1, j)$.

e) Starting by clearing the matrix Ω_{ls} and then repeating the operation described under (d) s times in all, adding the result each time to Ω_{ls} , we finally obtain our matrix.

f) For Ω_{lc} , exactly the same procedure is used, only using c instead of s and taking cosines of the elements instead of sines in step (b).

The programme can now be written symbolically as

<u>Formation of Ω_{ls}</u>

Input

$(+1) \longrightarrow N$

$(2\pi/t) \longrightarrow P$

$\sum_{(t)} \longrightarrow SIG$

$I_{(t)} \longrightarrow I$

Programme

Gb (1,1) (1,o) (t) ⟶ Be

(N) x (bBe, t x 1) ⟶ e

(bSIG, t) - (bI,t) ⟶ D

(bD,t) x (e,t x 1) ⟶ SP

(P) x (sP, t x 1) ⟶ Q

SEL (Q,t x 1) ⟶ S1

j = 1

CL (OMLS, t x(s - 1)) ⟶ OMLS

```
a)  Gb (1,1) (0,1) (j+1)        ——→ ejt
    (S1, t x 1) x (bejt, 1 x (j+ 1)) ——→ s1e
    DEC (s1e, tx(j+1) ——→ s1ed
    REC (s1ed,(j+1) x t) ——→ s1ed *
    TR (s1ed *  , (j+1) x t) ——→ st
    Gb (1,j) (-,-) (1) ——→ C
    (st, t x (j+1)) x (bC, j+1 x (s-1)) ——→ OMSDEL
    (OMLS, txs - 1) + (OMSDEL, t x s - 1) ——→ OMLS
    j = j + 1
    Jump to a) if j ≠ s
    E n d
```

This is then also performed for Ω_{lc} , and the two sub-matrices are joined to give Ω_l .

APPENDIX E

APPROXIMATION TO THE INVERSE OF D

In this appendix we represent the results of a simplified computation to establish the accuracy of the statement that the inverse of a dominant D_r gives a reasonable approximation to the inverse of the total D. For this purpose we consider a uniform cylindrical four boom fuselage with twelve equally spaced uniform rings. The total $\underset{(10 \times 10)}{D}$ is given by:

$$D_{total} = D_r + D_q + D_l$$

$$\propto \underset{(10\times10)}{\Delta^2} + \lambda_q \underset{(10\times10)}{\Delta} + c\, D_l$$

If we, just for the sake of obtaining a qualitative answer, neglect the D_l, and vary the λ_q, comparing the values of the diagonal elements in the inverse of Δ^2 and of $\Delta^2 + \lambda_q \Delta$, we obtain the following table:

Ratio / λ_q	$\dfrac{D^{-1}_{r11}}{D^{-1}_{r11} + D^{-1}_{q11}}$	$\dfrac{D^{-1}_{r55}}{D^{-1}_{r55} + D^{-1}_{q55}}$
0.90	1.11	1.25
0.081	1.012	1.029
0.0073	1.0011	1.0027
0.00066	1.00010	1.00024

We observe that when the ratio of the elements of the D_r to those of the $D_r + D_q$ is about 100:1, the error becomes very small. This is usually the case in a fuselage. The argument can also be extended to include D_l.

REFERENCES

1. J.H. Argyris and S. Kelsey
"Energy Theorems and Structural Analysis"
(published originally in Aircraft Engineering, October-November 1954; February, March, April, May 1955)
Butterworths 1960.

2. J.H. Argyris
"The Elastic Aircraft and Modern Fuselage Analysis" (book)
Butterworths October 1962.

3. J.H. Argyris and S. Kelsey
"The Matrix Force Method of Structural Analysis and some new Applications"
Brit. Aeron. Research Council, R & M 3034, February, 1956.

4. J.H. Argyris, S. Kelsey and H. Kamel
"Matrix Methods of Structural Analysis"
A precis of recent developments.
Paper presented to the 14th meeting of the Structures and Materials Panel; Structures Group, AGARD,Paris, July,1962.

5. J.H. Argyris
"Recent Advances in Matrix Methods of Structural Analysis"
Progress in Aeronautical Sciences, Volume 4,
Pergamon Press, 1964.

6. P.M. Hunt
"The Electronic Digital Computer in Aircraft Structural Analysis"
A reprint from Aircraft Engineering, March-April-May, 1956.
Bunhill Publications Ltd.

7. E. Bodewig
"Matrix Calculus"
Second edition, 1959
North Holland Publishing Co.

8. Entwurf zu einem allgemeinen Matrizencode
Technische Hochschule Stuttgart
- Institut für Statik und Dynamik der Flugkonstruktionen -
Rechengruppe der Luftfahrt, 1963.