The Analysis of Large Structural Systems

By R. K. Livesley

The paper describes a technique which the author has used for analysing structural frames on EDSAC 2. The method is a general one which could be applied equally well to electrical networks. Its principal feature is a packing technique for the efficient storage of large banded matrices.

Introduction

General computer programs for the analysis of building frames and bridge trusses have been available for several years, and are now being used by many structural engineering designers for their routine analytical work. The methods on which these programs are based may be divided into two general classes.

The first class may be termed methods of nodal analysis. The displacements and rotations of the joints of a structure are taken as the basic variables, and all internal forces and moments expressed in terms of them. The analysis is carried out by setting up and solving the joint equilibrium equations, which relate the joint displacements to the known applied loads. Experiments with a program based on this approach were carried out by J. M. Bennett in 1949 on EDSAC 1 (Bennett, 1953), and a general analysis program was completed for the Manchester University Computer a few years later by the present author (Livesley 1953, 1954). Translations of this program have been made for a number of other machines.

The second class of methods may be called methods of mesh analysis. The basic unknowns are a set of redundant forces and moments, and the displacements of the structure are written in terms of these. The analysis is carried out by solving the equations of displacement compatibility. These methods have been developed and systematized by Argyris (Argyris, 1954), and a general program has been developed for a Pegasus computer (Hunt, 1956).

Whichever type of approach is used, it is natural to employ matrix algebra in describing the analytical procedure. In either case the central problem is the solution of a set of linear algebraic equations built up (preferably by the computer itself) from data describing the physical details and geometry of the structure.

The size of structure which can be analysed by such a program is usually determined by the capacity of the routine for solving the equations. In the earlier programs this routine was usually a standard library one, designed for general use rather than for any specific application. It was gradually realized, however, that the equations of structural frames (like the equations of electrical and finite-difference networks) have certain special properties which can be used to speed up the solution process. The matrices which arise are usually sparse, and in most cases the non-zero elements tend to be grouped about the leading diagonal.

Various techniques have been suggested for the analysis of large systems of equations which possess these characteristics. In the "method of tearing" (Kron, 1955) the structure (or network) is treated as a number of smaller systems connected together. Each of the smaller systems is analysed separately in terms of the boundary values at points which connect it to its neighbours, and these boundary values are obtained subsequently by matching the various solutions. A simple version of this method was incorporated in the Manchester University structural analysis program mentioned earlier.

The "method of tearing" has obvious affinities with matrix partitioning, the main difference being that it is described as a process applied to the system itself rather than to the equations. From a computational point of view the numerical consequences may be very much the same. An example of the use of partitioning for a particular type of "banded" matrix has been given recently by Wilson (Wilson, 1959).

Although valuable, the use of sub-systems leads to complicated organizational programming in both the assembly and the solution of the equations. The present paper describes an alternative method of shortening the solution time for certain types of structural problem which the author has used successfully on EDSAC 2.

Synthesis of the Equations

Although the last-mentioned method was developed in a program for the analysis of rigid-jointed plane frameworks, it will be described here in more general terms. For our purpose a structure is defined as an assembly of nodes, or joints, at which loads can be applied, connected by elements, or members, which behave in a linear manner under load. We shall restrict our attention to structures in which each member only connects two joints—that is to say, to structures in which the members can be represented diagramatically by lines joining the joints. We shall assume that the structure is sufficiently anchored to rigid supports to prevent indeterminate rigid-body displacements.

The displaced form of the structure under load may be defined in terms of the displacements of the joints. Each joint will have a certain number of degrees of freedom—6 for a rigid-jointed space frame, 3 for a rigid-jointed plane frame, etc., and with each degree of freedom there will be associated a certain component of displacement (rotation or translation) and a certain component of external load (moment or force). We use the symbol D to denote the vector formed by all the displacement components of a joint, and term it the "displacement" of the joint. In the same way we use the symbol F to denote the corresponding applied load vector, and term it the "load" at the joint. We imagine that these vectors are defined in a frame of reference which is the same for all the joints of the structure. We assume that the displacement of the end of each member meeting at a given joint is equal to the displacement of the joint, and that all the joints of a particular structure have the same number of degrees of freedom.

We consider now a general structure with N joints. We number the joints $1, 2, \ldots, N$, and consider all points of rigid foundation attachment (where the displacements are all zero) as joint 0. Since in practice there is never more than one member connecting any two joints, we may refer to a member by quoting the joint numbers at its ends. We use the suffices 1 and 2 to refer to the two ends of a member, and introduce the convention that the lesser of the two joint numbers corresponds to end 1, and the greater to end 2. (This allows us to define the "direction" of the member $1 \rightarrow 2$ uniquely, and thus excludes ambiguity in the specification of the member's orientation to the co-ordinate axes of the system.)

For any particular member $pq\,(p < q)$ we may write down an expression for the forces acting on the ends in terms of the displacements of those ends. Since the member is assumed to behave linearly, the expression will be a linear one, which may be written

$$\left.\begin{aligned} F_{pq} &= Y_{11}D_p + Y_{12}D_q, \\ F_{qp} &= Y_{21}D_p + Y_{22}D_q, \end{aligned}\right\} \tag{1}$$

where F_{pq}, F_{qp} are the force vectors (in system co-ordinates) at ends p and q respectively, as shown in Fig. 1, and Y_{11}, Y_{12}, etc., are termed the "stiffness matrices" of the member. It can be shown that equations (1) are always symmetric. Since the F and D vectors are expressed in the overall co-ordinate system of the structure, the stiffness matrices of the member $p\,q$ depend on its orientation with respect to that co-ordinate system, but apart from this they are independent of the position of the member in the structure. Details of these matrices for different types of structural element are available elsewhere (Matheson, 1959).

Equations similar to (1) may be written down for all the other members which meet at joint p. If F_p is the external applied load at this joint, the vector equation of joint equilibrium will be simply

$$F_p = \textstyle\sum_j F_{pj}, \tag{2}$$

where $\sum_j$ denotes summation over all the joints directly connected by members to the joint p. Substituting for the various F_{pj}'s from equations similar to (1), we obtain the vector load-displacement equation for the joint p, and a similar process may be carried out for all the other

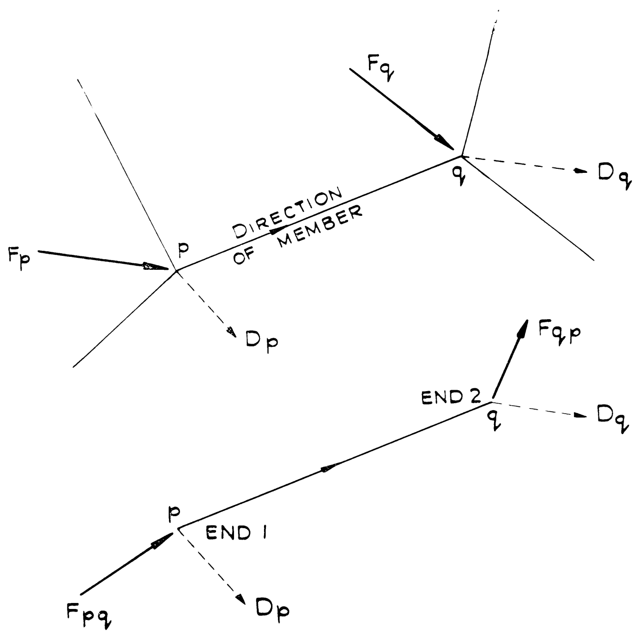

Fig. 1.—Force and displacement vectors for a member of a general structure.

joints of the structure. The complete set of equations may be written

$$\boldsymbol{F} = \boldsymbol{YD}, \tag{3}$$

where $\boldsymbol{F}$ is the column vector of the applied joint loads, $\boldsymbol{D}$ is the column vector of the joint displacements and $\boldsymbol{Y}$ is the stiffness matrix of the complete structure.

The construction of the matrix $\boldsymbol{Y}$ from the matrices of the individual members is very simple. The matrices for the member pq, for instance, will appear in the final matrix as follows:

$$\begin{array}{l} \text{row } p \\ \\ \text{row } q \end{array} \begin{bmatrix} \cdots & Y_{11} & \cdots\cdots & Y_{12} & \cdots \\ & \vdots & & \vdots & \\ \cdots & Y_{21} & \cdots\cdots & Y_{22} & \cdots \end{bmatrix} \tag{4}$$
$$\qquad\qquad\text{column } p \qquad\qquad \text{column } q$$

and the complete matrix will simply be the sum of patterns such as (4). (It should be noted that since $\boldsymbol{Y}$ has no row or column associated with joint 0, members whose end 1 is rigidly fixed only contribute their Y_{22} matrices to $\boldsymbol{Y}$.) Since these patterns are all symmetric, the final matrix $\boldsymbol{Y}$ will also be symmetric, and we need only consider the assembly of the upper triangle. We may state the rules for assembling the p'th row of this triangle as follows.

(1) The leading diagonal element is the sum of the matrices Y_{11} or Y_{22} for all the members meeting

at joint p, where Y_{11} is selected if the member has end 1 at p, and Y_{22} is selected if the member has end 2 at p.

(3) The off-diagonal elements, corresponding to columns $q, r, s \ldots > p$, are the Y_{12} matrices of the members connecting p to the associated joints. If a joint is not directly connected to p, the associated element is 0. It should be noted that in view of symmetry we need only consider here members which have end 1 at joint p, i.e. members leading to higher-numbered joints.

As an example, the frame in Fig. 2 has a $\boldsymbol{Y}$ matrix which may be represented schematically as follows:

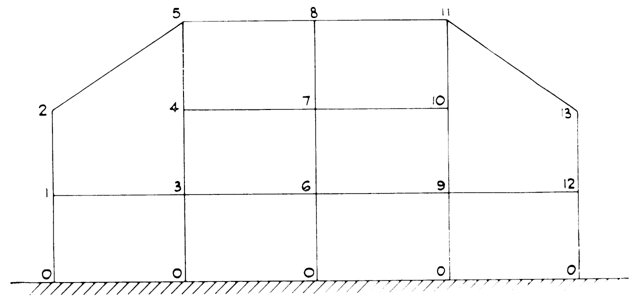

Fig. 2.—A typical frame.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	×	×	×										
2		×	*		×								
3			×	×	*	×							
4				×	×	*	×						
5					×	*	*	×					
6						×	×	*	×				
7							×	×	*	×			
8								×	*	*	×		
9									×	×	*	×	
10										×	×	*	
11											×	*	×
12												×	×
13													×

(5)

Only the part of the matrix above and including the leading diagonal is shown. The ×'s represent non-zero elements, while the significance of the asterisks will be made clear later.

It will be seen that only about a third of the elements in (5) are non-zero. Furthermore, this characteristic of sparseness will increase with the number of joints. Consider, for instance, a space framework of N joints built up on a rectangular lattice pattern. At most of the joints there will be 6 members meeting, so that there will be a total of approximately $3N$ members. There will therefore be approximately $4N$ non-zero elements in the upper triangle of $\boldsymbol{Y}$, in comparison with a possible total number of $N(N + 1)/2$. The proportion of non-zero elements is thus approximately inversely proportional to the number of joints N.

In the majority of practical frameworks joints are only connected to their nearer neighbours. This means that the matrix $\boldsymbol{Y}$ will normally take the form of a band of coefficients about the leading diagonal as in (5), provided that a suitable system of joint numbering is adopted. We shall see later that the band-width of the matrix, which we define here as the maximum value of $(q - p)$, is very important in the solution process, and it is advantageous to choose a system of joint numbering which keeps the band-width as small as possible. Numbering the joints of the structure shown in Fig. 2 in a horizontal sequence, for instance, would increase the band-width of the matrix (5) and make the solution slightly more lengthy. Minimizing the band-width is equivalent to minimizing the greatest difference between the two joint numbers associated with each member, and it would seem that this is a process which could be carried out by the computer itself.

In this connection it is of interest to note that the band-width may sometimes be reduced by the insertion of extra joints.* Consider, for example, the section of a frame shown in Fig. 3(a). The structure is assumed to extend to both the left and the right of the part shown. Insertion of the extra joints shown in Fig. 3(b) has the effect of reducing the minimum band-width considerably.

Storage of the Equations

Any efficient routine for solving sets of simultaneous equations such as (3) must obviously be designed to avoid useless operations with zero elements. It is not enough, however, merely to avoid carrying out such operations. In a computer such as EDSAC 2, with an

* This suggestion was made to the Author by Mr. E. A. Richards of the English Electric Company.

effective high-speed working space of about 700 numbers and an auxiliary store on magnetic tape,* the time spent on magnetic transfers may become a high proportion of the total time if care is not exercised during the programming. It is desirable, therefore, to avoid storing the zero elements entirely.

The simplest way of doing this is to use a method of solution in which the matrix Y is not altered during the solution process. Most such methods are iterative in character, but at least one, the "Method of Conjugate Gradients" (Hestenes and Stiefel, 1952) is theoretically exact, being an iterative process which terminates after N steps. The only use which the method makes of the matrix Y is in multiplying a sequence of vectors.

A program for general plane-frame analysis using the Hestenes–Stiefel method was constructed for EDSAC 2 during the period 1958–59. It has been found, however, that the method has two serious drawbacks. In the first place, the N matrix multiplications require N scannings of the magnetic tape, and this is time-consuming, even without the storage of zero elements. More seriously, rounding errors show a tendency to build up to such an extent that the solution after N steps is often a worse approximation to the correct solution than the starting point. This build-up appears to be due to the fact that some of the displacement components in a structure are often closely coupled, so that the matrix Y is rarely well-conditioned. It could probably be cured by double-length working in certain parts of the program, but this would still further increase the computing time. The method was therefore abandoned in favour of an elimination process.

It is sometimes assumed that elimination requires storage of the complete matrix, since elements which are initially zero may become non-zero during the solution process. Consideration quickly shows, however, that not all the zero elements are affected in this way. If the matrix consists of a number of bands parallel to the leading diagonal, then the parts of the matrix lying outside the extreme bands will not be affected. In the matrix (5), for example, only the zero elements denoted by asterisks are changed during the elimination process. It is possible to anticipate these changes during the assembly of the matrix Y, and leave appropriate spaces in the sequence of stored non-zero elements.

In the EDSAC 2 program, the non-zero sub-matrices of the upper triangular half of Y (with gaps where necessary) are stored consecutively on magnetic tape, reading along the successive rows of Y from the leading diagonal elements. The positions of the elements in the complete matrix are determined by N "tag registers," one for each row. In the tag register for the p'th row of Y, for instance, the first binary digit represents the leading diagonal element, and the $(q - p + 1)$'th digit the element in column q. A 1 in a digit position indicates either a non-zero element or a space necessary for the solution process.

As mentioned earlier, members are identified by the

* For further details, see p. 32 of this issue.

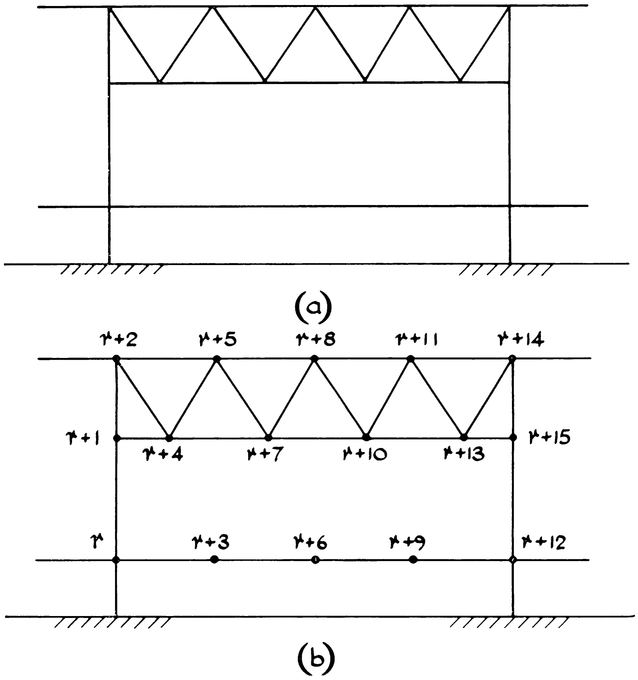

Fig. 3.—Example showing the use of additional nodes.

joint numbers p, q($p < q$) at their ends. The data tape for the structure consists of the sets of physical constants which define the members (i.e. lengths, cross-sectional areas, etc.) arranged in ascending order of p and, for each given value of p, in ascending order of q. Thus the members of the frame shown in Fig. 2 would be taken in the order 0/1, 0/3, 0/6, 0/9, 0/12: 1/2, 1/3: 2/5: 3/4, 3/6, etc.

It will be recalled that the member p/q contributes sub-matrices Y_{11}, Y_{12} to the p'th row of the complete matrix, and a sub-matrix Y_{22} to the leading diagonal of the q'th row (only terms on and above the leading diagonal being stored). At the time when the p'th row is being formed the location of the start of the q'th row will not be known, so that it is necessary to keep a temporary list of Y_{22} matrices during the assembly process.

The first member details to be read are those for which $p = 0$. For each member the matrices Y_{11}, Y_{12}, Y_{22} are formed by a subroutine and stored for subsequent use in the calculation of the internal forces. Since Y has no row corresponding to joint 0, the assembly routine merely adds the Y_{22} matrices for these members into the appropriate places in the Y_{22} list (this being initially clear).

When the number p changes to 1, the program transfers the first sub-matrix from the Y_{22} list to the leading diagonal position in the first row of Y. It also inserts a 1 in the most significant digit position of the first tag-register. For each member which has $p = 1$ it adds the Y_{11} matrix into the leading diagonal position, plants the Y_{12} matrix in the next vacant space, adds the Y_{22} matrix to the temporary Y_{22} list, and plants a 1 in the appropriate position in the tag-register. Applying this process to the structure shown in Fig. 2, the first tag-register would eventually read 11100 . . ., corresponding to the three elements in the first row of (5).

Before the same process can be carried out for the second joint it is necessary to consider the operations which will be carried out with the first row of the matrix during the elimination process. Since $\boldsymbol{Y}$ is symmetric, the presence of off-diagonal elements in the row implies the existence of corresponding elements in the first column, which will be reduced to zero in the elimination process by adding suitable multiples of the first row to the other relevant rows. Thus a 1 in the q'th digit position of the first tag-register indicates that the first row will be added to the q'th row during the solution. Clearly the q'th row must have elements, zero and non-zero, corresponding to all the elements of the first row which will be added during this step. This can be arranged as follows. The content of the first tag-register is shifted one digit left at a time, the most significant digit being tested at each stage. Each time a 1 is found, the content of the first tag-register (in shifted form) is added* to the tag-register corresponding to the original position of that digit. In general the content of the tag-register is shifted $(q - p)$ places left and added into the q'th tag-register if a 1 appears in the most significant digit position. In our example, this process sets the second tag-register equal to 11000 . . . before the assembly routine commences the formation of the second row, the second digit corresponding to the asterisk in the second row of (5).

The second row is now formed in a similar manner to the first, and stored immediately after it. The assembly routine takes note of any 1's transferred to the second register from the first, and leaves appropriate spaces in the row. In our example, the complete second row comprises three sub-matrices, the second one being composed of zeros, and the second tag-register finally reads 1101000 . . . This tag-register is now treated in the same way as the first, and the whole process repeated for the third and subsequent rows. When all the information describing the members of the structure has been read from the data tape and the matrix $\boldsymbol{Y}$ has been completed and stored, the program reads the values of the loads applied to the structure and enters the solution routine.

Solution of the Equations

The solution of the equations by the normal method of successive elimination and back-substitution calls for little comment. The condensed form of the matrix naturally leads to a considerable amount of red-tape programming, but since most of this is concerned with the determination of the addresses of elements, it can be done by fixed-point arithmetic in the modifier registers. These are 11-bit registers, and the instructions associated with them take very little time compared with the floating-point arithmetic used in the main calculation. It must also be remembered that the elements are not single numbers but sub-matrices, so that the address calculations need only be made once for each sub-matrix operation.

For reasons of programming simplicity the elimination process is carried out in a completely systematic manner. No attempt is made to pick out the largest pivot, but experience shows that for the type of matrix considered there is little tendency for round-off errors to be troublesome. This is true even in stability investigations, where one is dealing with a matrix which becomes extremely ill-conditioned as the critical load is approached.

While it is not possible, in general, to hold the whole of the matrix $\boldsymbol{Y}$ in the working store of the machine at once, the banded nature of this matrix implies that each row will only affect its immediate successors in the elimination process. If the last element in the p'th row lies in the q'th column, then row q will be the last to be modified by row p, and it will usually be possible to hold rows $p, p + 1, \ldots, q$ in the working store at the same time. This allows a great economy in magnetic transfers, since each row can be copied into the store for modification, held there until it is used as the pivotal row, and finally written back into the magnetic store.

In the EDSAC 2 program the elements are 3×3 matrices, so that the 500 registers used for the operations on the matrix $\boldsymbol{Y}$ allow for a continuous band-width of about 6 (i.e. $q - p \leqslant 6$). This corresponds to a tall building with 6 joints per floor. The first row of $\boldsymbol{Y}$ is copied from tape into the first of the 500 registers, and the other rows transferred in order when required for modification. When a row has been used as the pivotal row it is written back onto the tape and the remaining rows shifted up, so that the next row comes into the pivotal position. Thus each row is transferred to the next vacant position in the working space, is modified by the other rows, and finally acts as the pivotal row before being written back onto tape. For the final back-substitution the process is reversed, the program working up the successive rows of $\boldsymbol{Y}$ from the bottom. When the displacements of the structure have been found, the internal forces and moments may be found simply by using equation (1).

The part of the program which prints the forces and moments in the members, and the deflections of the joints, also carries out a check on the solution by subtracting the computed forces and moments from the appropriate terms in the external load vector F. Referring to equation (2), it is clear that when all the members have been dealt with, F will have been reduced to the residual vector, and this is printed out with the displacements. In view of the fact that the loads applied to a real structure are only known to within certain limits, any solution with a *residual* vector lying within those limits must be accepted as "correct," whatever the size of the associated *displacement* error vector.

* The instruction used here and elsewhere during the formation of the tag-registers corresponds to logical addition, not ordinary addition.

Conclusions

The EDSAC 2 program described above is in fact part of a larger program (Livesley, 1959) for solving problems of structural instability when plasticity is present. Since such problems are non-linear, they are solved by an iterative approach. Each iteration involves a solution of the linear equations (3), and after each solution the individual member matrices are modified to take account of axial forces and plasticity. The program is therefore not as ambitious, in terms of size of structure, as it might have been if simple linear analysis had been its objective. Its present capacity is a plane frame of 25 joints, giving rise to 75 scalar equations.

As an example of the speed of this program, a frame of 44 members and 25 joints took 4 minutes to analyse. Of this time, $1\frac{1}{2}$ minutes was spent in reading the data tape, 2 minutes in punching out the solution, and only 30 seconds in solving the equations. The frame was more or less "square" in terms of joint arrangement—that is to say it had the maximum possible band-width (6) for a frame of its size. The actual speed of the solution is not very significant. What is important is the fact that for a given band-width the solution time is linear (it is easy to show that when the band-width is small compared with the size of the matrix the solution time varies as the square of the band-width). Thus a tall building frame of 6 joints per storey and 40 storeys high would take about 5 minutes for the solution of the 720 scalar load-displacement equations.

The technique described in this paper is obviously most suitable for systems which can be described as "long and thin." Although described in structural terms it can be used in appropriate cases for dealing with electrical networks and systems of finite difference equations. It can of course be combined with the other devices for reducing solution times mentioned earlier.

Acknowledgement

The author wishes to thank the director, Dr. M. V. Wilkes, for permission to use EDSAC 2 and the other facilities at the Cambridge University Mathematical Laboratory.

References

ARGYRIS, J. H. (1954). "Energy Theorems and Structural Analysis," *Aircraft Engineering*, Vol. 26, p. 347 (October 1954), and subsequent issues.

BENNETT, J. M. (1953). Ph.D. Thesis, Cambridge University.

HESTENES, M. R., and STIEFEL, E. (1952). "Methods of Conjugate Gradients for Solving Linear Systems," *J. Res. Nat. Bur. Standards*, Vol. 49, p. 409.

HUNT, P. M. (1956). "The Electronic Digital Computer in Aircraft Structural Analysis," *Aircraft Engineering*, Vol. 28, p. 70, (March 1956), and subsequent issues.

KRON, G. (1955). "Solving Highly Complex Elastic Structures in Easy Stages," *Journal of Applied Mech.*, Vol. 22, p. 235.

LIVESLEY, R. K. (1953, 1954). "Analysis of Rigid Frames by an Electronic Digital Computer," *Engineering*, Vol. 176, pp. 230 and 277 (August 1953), and Vol. 177, p. 239 (February 1954).

LIVESLEY, R. K. (1959). "The Application of Computers to Problems Involving Plasticity," Conference on Structural Computing, Southampton University (September 1959).

MATHESON, J. A. L. (1959). *Hyperstatic Structures*, Butterworths, London.

WILSON, L. B. (1959). "Solution of Certain Large Sets of Equations on Pegasus Using Matrix Methods," *The Computer Journal*, Vol. 2, p. 130.

Conference Notice

L'Association Française de Calcul is holding its first conference in Grenoble on 14, 15 and 16 September. The following topics will be included:

Logical design.
Programming.
Numerical analysis.
Scientific and industrial applications of computers.
Automatic control.
Use of computers for operational research.
Translation and automatic documentation.

Further information may be obtained from:

Secretariat de l'Association Française de Calcul,
Institut d'Astrophysique,
98 bis, Boulevard Arago,
Paris 14^{e}.