R. & M. No. 3632

MINISTRY OF TECHNOLOGY

AERONAUTICAL RESEARCH COUNCIL
REPORTS AND MEMORANDA

Optimum Design of Pin-Jointed Frameworks

By W. S. HEMP and H. S. Y. CHAN

Department of Engineering Science
University of Oxford

LONDON: HER MAJESTY'S STATIONERY OFFICE
1970
PRICE 14s 0d [70p] NET

Optimum Design of Pin-Jointed Frameworks

By W. S. HEMP and H. S. Y. CHAN

Department of Engineering Science
University of Oxford

Reports and Memoranda No. 3632*

November, 1966

Summary.

A linear programme is formulated which determines the layout and member sizes of frameworks, designed to carry given forces, made from material with a given allowable stress and using a minimum volume of material. The dual program is shown to lead to Michell's criteria for optimum design. A practical method of calculation, based on the dual, is presented and a computer program written in Algol is given in an appendix. Examples include unrestricted cantilevers under tip force and distributed forces, a restricted cantilever under tip force and structures of the type required for machine tools.

LIST OF CONTENTS

*Replaces A.R.C. 30 994.

Illustrations—Figs. 1 to 7

Detachable Abstract Cards

---

1. *Basic Theory.*

1.1. *Structures of Least Weight.*

Consider any system of concentrated forces applied at points in space and forming a system in static equilibrium*. Any pin-jointed framework which balances out these forces must have nodes at the points of application of these forces, but it can have nodes at an indefinite number of other points as well and its members can lie along any of the segments joining nodes. To be definite it is necessary to specify possible nodes and possible members, but these can be assigned with what from a practical point of view can be regarded as complete generality. The region of space, through which structures for the purpose in hand can be allowed to extend, might for example be covered by a rectangular grid of points of close mesh and all segments joining grid points might be allowed as potential members. On the other hand practical restrictions of layout might limit the choice of nodes and segments in some special way. For the present purpose it is only necessary to limit possibilities to a finite class of structural layouts.

Any appropriate structure designed from material which is capable of carrying tension and compression stresses of $\pm\sigma$ will then require a certain volume of material if it is to balance the given forces with safety. The problem to be considered here is the determination of that structure which requires the least volume of material to carry out its function.

Consider then what is for the present purpose the most general structure and let t be the row matrix of end loads in its members. The nodal equilibrium equations can then be written

$$tK = f \tag{1.1.1}$$

where f is a row matrix of components of the given forces, not including reactions at fixed supports and K a matrix which depends on the assumed geometry of the structure. The structure is safe, at least from the point of view of limit design if

$$t \leqslant \sigma a, \qquad -t \leqslant \sigma a \tag{1.1.2}$$

where a is a row matrix of cross sectional areas of members. It will be assumed that a t can be found which satisfies (1.1.1). Equation (1.1.2) then gives an a which will ensure safety. The problem is then to find a t and an a which satisfies

$$\text{Min. } V = al \tag{1.1.3}$$

where l is a column matrix of lengths of members and V the total volume of material in the structure. This linear programming problem certainly has an optimal solution, since V is clearly bounded from below.

Introducing non-negative variables t' and t'' by

$$t = t' - t'', \quad \text{where} \quad t_i'' = 0, \quad \text{if} \quad t_i \geqslant 0 \quad \text{and} \quad t_i' = 0, \quad \text{if} \quad t_i < 0 \tag{1.1.4}$$

---

*Some of the forces may be, perhaps unknown, reactions at fixed supports.

and t_i, t_i', t_i'' are the components of these matrices, enables (1.1.1, 2, 3) to be written

$$\left.\begin{aligned} &(t'-t'')K = f \\ &0 \leqslant t' \leqslant \sigma a, 0 \leqslant t'' \leqslant \sigma a \\ &\text{Min. } V = al \end{aligned}\right\} \tag{1.1.5}$$

or, since l is positive as

$$\left.\begin{aligned} &(t'-t'')K = f \\ &t', t'' \geqslant 0 \\ &\text{Min. } V = \frac{(t'+t'')l}{\sigma} \end{aligned}\right\} \tag{1.1.6}$$

which is a linear programming problem in its standard form with non-negative variables. It is also clear that an optimal solution of (1.1.6) will have either t_i' or t_i'' equal to zero, as is required by (1.1.4). The problem of (1.1.6) can be resolved by the 'simplex method' or by one of its variants.

1.2. *The Criterion of A. G. M. Michell.*

If the first of (1.1.6) be written as a pair of inequalities, the problem of (1.1.6) can be written as

$$\left.\begin{aligned} &[t', t'']\begin{bmatrix} K, & -K \\ -K, & K \end{bmatrix} \geqslant [f, -f] \\ &[t', t''] \geqslant 0 \\ &\text{Min. } V = \frac{1}{\sigma}[t', t'']\begin{bmatrix} l \\ l \end{bmatrix} \end{aligned}\right\} \tag{1.2.1}$$

The dual problem to (1.2.1) is then

$$\left.\begin{aligned} &\begin{bmatrix} K, & -K \\ -K, & K \end{bmatrix}\begin{bmatrix} u' \\ u'' \end{bmatrix} \leqslant \varepsilon\begin{bmatrix} l \\ l \end{bmatrix} \\ &\begin{bmatrix} u' \\ u'' \end{bmatrix} \geqslant 0 \\ &\text{Max. } W = \frac{1}{\sigma\varepsilon}[f, -f]\begin{bmatrix} u' \\ u'' \end{bmatrix} \end{aligned}\right\} \tag{1.2.2}$$

where ε is a positive infinitesimal introduced so as to allow the interpretation of the new variables u', u'' as infinitesimals. These last are of course column matrices. Equations (1.2.2) can be written more concisely as

$$\left.\begin{aligned} &-\varepsilon l \leqslant K(u'-u'') \leqslant \varepsilon l \\ &u', u'' \geqslant 0 \\ &\text{Max. } W = \frac{f(u'-u'')}{\sigma\varepsilon} \end{aligned}\right\} \tag{1.2.3}$$

which on writing

$$u = u' - u'' \tag{1.2.4}$$

is seen to be equivalent to

$$\left.\begin{aligned} -\varepsilon l \leqslant Ku \leqslant \varepsilon l \\ \text{Max. } W = \frac{fu}{\sigma\varepsilon} \end{aligned}\right\} \tag{1.2.5}$$

where u is unrestricted in sign.

The variables of (1.2.3, 5) may be interpreted as follows. The column u can be taken to be a column of nodal virtual displacements corresponding* to the components of force f. The quantities Ku are then the extensions of the lengths of members of the general structural layout. This follows from (1.1.1), which gives $tKu = fu$, which is the principle of virtual work. The problem of (1.2.5) can thus be interpreted as the determination of that virtual displacement of the nodes of the most general structure, which is restricted so that the corresponding strain in all potential members has modulus $\leqslant \varepsilon$ and which gives a maximum value for the work done by the given external forces. The theory of duality in linear programming gives

$$V_{\min} = W_{\max} \tag{1.2.6}$$

and so the problem of determining the minimum structural material for an optimum structure is reduced to solving the problem of (1.2.5)

The solution to (1.2.5) gives in fact complete information about the layout of the optimum structure as well as the values of the end loads in the members for a statically determinate optimal solution. This can be formulated by introducing 'slack variables' in the first of (1.2.5) and writing it as

$$\left.\begin{aligned} \sum_{i=1}^{n} K_{ji}\, u_i + u_{n+j} = \varepsilon l_j \quad (j = 1, 2 \ldots . m) \\ \sum_{i=1}^{n} K_{ji}\, u_i + u'_{n+j} = \varepsilon l_j \quad (j = 1, 2 \ldots . m) \\ u_{n+j}, u'_{n+j} \geqslant 0 \qquad (j = 1, 2 \ldots . m) \end{aligned}\right\} \tag{1.2.7}$$

where

An appropriate numbering of the equations then allows u_{n+j} $(j = 1, 2 - t)$, u'_{n+j} $(j = t+1, -n)$ to be the non-basic variables in a basic optimal solution of (1.2.5). The determination of this solution will then give the relations

$$W = W_{\max} - \frac{1}{\sigma\varepsilon}\left(\sum_{j=1}^{t} z_j\, u_{n+j} + \sum_{j=t+1}^{n} z'_j\, u'_{n+j}\right) \tag{1.2.8}$$

---

*Displacements at fixed supports are not included or alternatively should be taken as zero.

and as is shown in the standard proofs of duality theorems (*see* Ref. 1), an optimal solution to (1.1.6) is given by

$$\begin{aligned} t'_j &= z_j \,(j = 1, 2 \text{---} t), & t'_j &= 0 \,(j = t+1, \text{---} m) \\ t''_j &= z'_j \,(j = t+1, \text{---} n), & t''_j &= 0 \,(j = 1, 2 \text{---} t, n+1, \text{---} m) \end{aligned} \tag{1.2.9}$$

This solution has tensile end loads in members for which $u_{n+j} = 0$ or $\sum_{i=1}^{n} K_{ji} u_i = \varepsilon l_j$ and compressive end loads in members for which $u'_{n+j} = 0$ or $\sum_{i=1}^{n} K_{ji} u_i = -\varepsilon l_j$. The layout of the optimum structure is thus determined by those segments connecting nodes, which have strains $\pm\varepsilon$ in the virtual deformation of (1.2.5), the tensile members lying along segments with strain $(+\varepsilon)$ and the compressive along those which strain $(-\varepsilon)$. Equation (1.2.9) gives the end loads in equilibrium with the given external forces for a structure with this layout. The corresponding cross sectional areas are given by t'_j/σ and t''_j/σ and the total volume of material so determined is given by (1.2.6). This demonstrates the existence of a structure, which satisfies Michell's criterion for an optimum structure:

'A structure is an optimum if it can carry the given external forces with corresponding stresses $\pm\sigma$ in all its members and if further it allows a virtual displacement of its possible nodes which produces strains of $(\pm\varepsilon)$ in its tension and compression members respectively and which further produces no strain of absolute value greater than ε in any segment along which a potential structural member could lie.' (1.2.10)

The sufficiency of the condition (1.2.10) follows from the principle of virtual work, which for a 'Michell structure' gives $fu = \sum_{j=1}^{m} (\sigma a_j)(\varepsilon l_j) = \sigma\varepsilon V$. However $W = fu/\sigma\varepsilon$ by (1.2.5) and so $W = V$. This means, since $W \leqslant W_{\max} = V_{\min} \leqslant V$ that $V = V_{\min}$ and that the Michell structure is an optimum.

An optimal solution to (1.2.5) may be 'degenerate'. This will mean that the layout determined by segments connecting nodes, which have strains of modulus ε, will give a statically indeterminate structure. If (1.1.1) can then be solved, in terms of redundant parameters, to give end loads which are consistent with the sign of the strains in the sense of (1.2.10), then a class of optimum structures will be obtained, which will have the same volume $V_{\min}$ and will include the statically determinate structure of (1.2.9) as a special case.

2. *Method of Calculation.*

2.1. *Equations for Two-Dimensional Frameworks.*

The method of calculation of optimum structures given here is based on (1.2.5). The formulation will, for simplicity, be confined to plane structures, but the method is readily generalised to three dimensions.

Let the region to be occupied by the structure be covered by a grid of points P_i $(i = 1, 2 \text{---} k)$ with co-ordinates (x_{1i}, x_{2i}) referred to rectangular co-ordinate axes x_1, x_2. The points of application of the given forces and the points of fixed support, if any, must be included among P_i. The remaining points should be sufficiently numerous to generate structures of practical generality. Possible members of the structure can be taken to lie along all segments $P_i P_j$, but some segments may be excluded if the requirements of the design demand it.

The virtual displacements at P_i are denoted by (u_{1i}, u_{2i}), where the appropriate displacement components at fixed points are taken as zero. The condition on strain, expressed by the first of (1.2.5), can then be written for a segment $P_i P_j$ as

$$|(x_{1i}-x_{1j})(u_{1i}-u_{1j})+(x_{2i}-x_{2j})(u_{2i}-u_{2j})| \leqslant \varepsilon \{(x_{1i}-x_{1j})^2+(x_{2i}-x_{2j})^2\} \tag{2.1.1}$$

Such relations must be written for all pairs of points, which give independent restrictions on the displacements. For computation it is convenient to replace u_{1i}, u_{2i} $(i = 1, 2 \ldots\ldots k)$ by $\varepsilon\, v_i$ $(i = 1, 2 \text{---} n)$, in other words to return to the formulation of (1.2.5) with the slight change of notation

$$u = \varepsilon\, v \tag{2.1.2}$$

The complete set of relations (2.1.1) can then be written

$$\left|\sum_{i=1}^{n} a_{ji}\, v_i\right| \leqslant b_j \, (j = 1, 2 \text{--} m) \tag{2.1.3}$$

which are equivalent to the first of (1.2.5), but in a more readily computable form.

The problem of (1.2.5) can now be reformulated in the present notation. It is convenient to modify the formulation of (1.2.7) so that the standard 'upper bound' techniques can be applied and to write

$$\left.\begin{aligned}
&\sum_{i=1}^{n} a_{ji}\, v_i + v_{n+j} = b_j \qquad (j = 1, 2 \text{---} m)\\
&v_{n+j} + v'_{n+j} = 2b_j \qquad (j = 1, 2 \text{---} m)\\
&v_{n+j}, v'_{n+j} \geqslant 0 \qquad (j = 1, 2 \text{---} m)\\
&\text{Max. } \sigma\, W = \sum_{i=1}^{n} f_i\, v_i
\end{aligned}\right\} \tag{2.1.4}$$

The first three lines of (2.1.4) are equivalent to (2.1.3), since they confine $\sum_{i=1}^{n} a_{ji}\, v_j$ to the range $(-b_j)$ to $(+b_j)$. The layout of the optimum structure is now determined by $v_{n+j} = 0$ and $v'_{n+j} = 0$, which determine tension and compression members respectively. In the computing program the variable v_{n+j} is denoted by its index $n+j$, while v'_{n+j} is denoted by $-(n+j)$.

2.2. *Data for the Computer Program.*

A computer program which solves the problem of (2.1.4) is given in the Appendix. The data for this program may be given either in the form of the contents of (2.1.3) or more conveniently by the specification of the points P_i. These two cases are distinguished by the values 0 and 1 respectively of a parameter p*.

The data following $p = 0$ is as follows:

$$\left.\begin{aligned}
&o; n; m;\\
&b_1; a_{11}; a_{12}; \text{- - - -} a_{1n};\\
&b_2; a_{21}; a_{22}; \text{- - - -} a_{2n};\\
&\text{- - - - - - - - - - -}\\
&\text{- - - - - - - - - - -}\\
&b_m; a_{m1}; a_{m2}; \text{- - - -} a_{mn};\\
&o; -f_1; -f_2; \text{- - - - -} f_n;
\end{aligned}\right\} \tag{2.2.1}$$

*A third value $p = -1$ is used to terminate the program.

where the first zero gives the value of p, n is the number of variables v_i and m is the number of inequalities (2.1.3) or of slack variables v_{n+j} or v'_{n+j}. The next m lines give the constants of the first of (2.1.4) and the last line is formed using the coefficients f_i from the function to be maximised. The data following $p = 1$ is as follows:

$$
\begin{array}{l}
1; n; m; n; k; \\
x_{11}; x_{21}; r_1; g_1; h_1; s_1; \\
x_{12}; x_{22}; r_2; g_2; h_2; s_2; \\
x_{13}; x_{23}; r_3; g_3; h_3; s_3; \\
------------ \\
------------ \\
x_{1k}; x_{2k}; r_k; g_k; h_k; s_k; \\
c_{11};\ c_{12};\ ---\ c_{1s_1}; \\
c_{21};\ c_{22};\ ---\ c_{2s_2}; \\
------------ \\
------------ \\
c_{k1};\ c_{k2};\ ----\ c_{ks_k}; \\
0;\ -f_1;\ -f_2;------\ -f_n;
\end{array}
\tag{2.2.2}
$$

where the first term gives the value of p, the second and fourth the number of variables v_i and the third an upper limit for the number of inequalities (2.1.3). If the actual number of inequalities generated by the program does not exceed m the calculation will continue. The fifth term gives the number of points P_i and the next $2k$ lines give the co-ordinates (x_{1i}, x_{2i}) of P_i, as well as information about the virtual displacements at P_i and the segments $P_i\,P_j$, which are to be excluded from the constraints (2.1.3), because the corresponding members are not allowed. The program automatically excludes segments, which can be constructed by joining two or more adjacent collinear segments together. These would, if retained, only duplicate the constraints imposed by their component parts.

The variables v_i are allocated to P_i in the strict order of these points with the x_1-components preceding the x_2-components. If P_1 is free its virtual displacement will be (v_1, v_2). If P_2 is free to move in the x_1-direction only, its displacement will be (v_3, o) and if P_3 is free to move in the x_2-direction only, its displacement will by $(0, v_4)$. If P_4 is completely free, then its displacement is (v_5, v_6) and so on. The index r_i then records the lower index of the variable or variables associated with P_i. A fixed point will have no variables associated with it and in this case r_i records the index which would have been used first at P_i, if it had been free. If for example P_5 is fixed, its displacement will be $(0, 0)$ and since v_6 was the last variable for P_4, the value of r_5 is 7. The quantities g_i and h_i record the freedoms assumed for P_i. If $g_i = 1$ then P_i is free to move in the x_1-direction, but if $g_i = 0$ such movement is prevented. Similarly if $h_i = 1$, P_i can move in the x_2-direction, but if $h_i = 0$ movement in this direction is not allowed.

The quantity s_i gives the number of points $P_j (j > i)$ for which members $P_i\,P_j$ are excluded. The actual values of j, if $s_i \neq 0$, are given by $c_{i,1}\,c_{i,2}\,---c_{i,si}$. If $s_i = 0$ no record is to be made in the corresponding row i.e. $c_{i,1}$ etc. are left out of the data.

The last line of (2.2.2) is the same as the last of (2.2.1) and records the values of f_i, where f_i is the external force corresponding to v_i.

Several problems may be included in the same data presentation, if they are headed by $p = 0$ or $p = 1$ as appropriate. Such data should end with $p = -1$, which will terminate the calculations.

2.3. *Example of Data Presentation.*

As an example consider the problem of Fig. 1. Forces F are applied at P_1 and P_2 and the points P_3 and P_4 are fixed supports. The co-ordinates of P_i are given in the figure, as are the segments for possible members of structures designed to transmit the forces F to the supports. The only excluded segment is $P_3 P_4$.

The variables v_i $(i = 1, 2, \text{--}4)$ are assigned to P_i $(i = 1, 2)$ as shown in the figure and the constraints corresponding to (2.1.1) or equivalently by (2.1.3) are

$$\left.\begin{array}{rl} |v_2 - v_4| \leqslant 1 & \text{for } P_1 P_2 \\ |v_1| \leqslant 1 & \text{for } P_1 P_3 \\ |v_1 + v_2| \leqslant 2 & \text{for } P_1 P_4 \\ |v_3 - v_4| \leqslant 2 & \text{for } P_2 P_3 \\ |v_3| \leqslant 1 & \text{for } P_2 P_4 \end{array}\right\} \tag{2.3.1}$$

The data for this present problem can now be written using (2.2.1, 2). The two forms occur in succession in the following presentation:

$$\left.\begin{array}{ll} 0;4;5; & (p, n, m) \\ 1;0{:}1;0;-1; & (|v_2 - v_4| \leqslant 1) \\ 1;1;0;0;0; & (|v_1| \leqslant 1) \\ 2;1;1;0;0; & (|v_1 + v_2| \leqslant 2) \\ 2;0;0;1;-1; & (|v_3 - v_4| \leqslant 2) \\ 1;0;0;1;0; & (|v_3| \leqslant 1) \\ 0;-1;0;0;-1; & (\text{Max. } F(v_1 + v_4)) \\ 1;4;5;4;4; & (p, n, m, n, k) \\ 1;1;1;1;1;0; & (P_1 \text{ is free with displacement } (v_1, v_2)) \\ 1;0;3;1;1;0; & (P_2 \text{ is free with displacement } (v_3, v_4)) \\ 0;1;5;0;0;1; & (P_3 \text{ is fixed; next variable is } v_5;\ P_3 P_4 \text{ excluded}) \\ 0;0;5;0;0;0; & (P_4 \text{ is fixed; next variable is still } v_5) \\ 4; & (P_3 P_4 \text{ is excluded}) \\ 0;-1;0;0;-1; & (\text{Max. } F(v_1 + v_4)) \\ -1;\twoheadrightarrow & (\text{Terminates the program.}) \end{array}\right\} \tag{2.3.2}$$

2.4. *Output of the Computer Program.*

The output from the computer program of the Appendix has the following form:

FAEBDL200KP7

STR 30 (output on line printer)

ITER	LEFT	ENTER	FUNCTION
(number of iterations)	(variable becomes non-basic)	(variable becomes basic)	(value of the objective function σW)

RESULT
(no. of variable) (value of variable)

(2.4.1)

SLACK
(no. of variable) (partial cost)

STRUCTURAL MEMBERS	CROSS SECTIONS	VOLUME
(Number of all potential structural members i.e. number of corresponding slack variable and sign of end load)	(cross sectional area of member)	(area times length of members)

TOTAL VOLUME (total volume of the structure)

In the case when $p = 1$, i.e. when (2.2.2) is used, the inequalities of (2.1.3) are also recorded on paper tape, in a form which can be used in future calculations using (2.2.1).

The program is based on the 'simplex method' of solving linear programming problems. The output given under ITER, RESULT, SLACK records the details of the interations and is useful for checking and for estimating computing times for similar problems.

The numbers recorded under STRUCTURAL MEMBERS list all potential structural members as well as the sign of the end load that they must carry. The requirements of statics i.e. (1.1.1) may well give zero values of the end loads in certain cases and will thus eliminate some of the members. In general the solution obtained using (1.1.1) will be indeterminate. However the present program selects a statically determinate solution, after the manner of (1.2.9), and calculates the corresponding areas of cross section and member volumes. The total volume obtained by summing over all members should agree with the last entry under FUNCTION.

The program terminates after $p = -1$ in the data. If the estimate of m in (2.2.2) is too small the program will print COUNTING ERROR. In the case of errors in formulation, which give data with no optimal solution, the program will terminate and print UNBOUNDED SOLUTION.

2.5. *Example of Output.*

The data of (2.3.2) give the following output:

FAEBDL200KP7

STR 30

ITER	LEFT	ENTER	FUNCTION
1	6	1	$1{\cdot}000000_{10}+0;$
2	-5	4	$2{\cdot}000000_{10}+0;$
3	7	2	$3{\cdot}000000_{10}+0;$

RESULT

4	$2{\cdot}000000_{10}+0;$
1	$1{\cdot}000000_{10}+0;$
2	$1{\cdot}000000_{10}+0;$
−8	$0{\cdot}000000$;
9	$1{\cdot}000000_{10}+0;$

SLACK

6	$0{\cdot}000000$;
7	$1{\cdot}000000_{10}+0;$
3	$0{\cdot}000000$;
−5	$1{\cdot}000000_{10}+0;$

STRUCTURAL MEMBERS	CROSS SECTIONS	VOLUME
−8		
6		
7	$1{\cdot}414214_{10}+0;$	$2{\cdot}000000_{10}+0;$
−5	$1{\cdot}000000_{10}+0;$	$1{\cdot}000000_{10}+0;$

TOTAL VOLUME $3{\cdot}00000000_{10}+0;$

It is seen that three iterations were required and that the optimum value of the objective function is 30, which agrees with the 'total volume' and gives a structural volume of $30F/\sigma$ for the problem of Fig. 1, when the side of the square is of unit length. The program began with the basic variables v_5, ---v_9 and non-basic variables v_1, ---v_4. It ended with basic variables v_1, v_2, v_4, v_8 and v_9 and with non-basic variables v_3, v_5, v_6 and v_7 for the optimal solution. The actual values are recorded under RESULT. The virtual displacements corresponding to the optimal solution are thus $v_1 = 1, v_2 = 1, v_3 = 0$ and $v_4 = 2$; v_3 is non-basic and therefore zero.

The most general structure satisfying Michell's strain criterion of (1.2.10) has members corresponding to v_6 and v_7 in tension and v_5, v_8 in compression. This is shown in Fig. 2a. However the equilibrium of the nodes P_1 and P_2 shows that P_2P_3 and P_1P_3 will have zero end load. The optimum structure is thus that shown in Fig. 2b. This agrees with the specification given under STRUCTURAL MEMBERS.

3. *Examples.*

3.1. *Cantilever under Tip Forces.*

Consider the problem of Fig. 3a. An optimum structure is required to transmit the force F to the two fixed supports P_{11} and P'_{11}. The corresponding virtual displacement system is anti-symmetric about the line $P_{10}P_1$ and is defined approximately by the displacements V_1 to V_{18} of the nodes of the grid of Fig. 3b. For simplicity the possible structural layout is restricted to segments joining nodes of this grid and to their mirror images on the other side of $P_{10}P_1$.*

* Calculations which allow members to cross the line of symmetry give in fact the same volume and layout.

The data for this problem assembled in the manner of (2.2.2) is

$$\left.\begin{array}{l}
1;18;80;18;12;\\
6;0;1;0;1;1;\\
6;1;2;1;1;0;\\
6;2;4;1;1;0;\\
4;0;6;0;1;1;\\
4;1;7;1;1;0;\\
4;2;9;1;1;0;\\
2;0;11;0;1;1;\\
2;1;12;1;1;0;\\
2;2;14;1;1;0;\\
0;0;16;0;1;0;\\
0;1;17;0;0;0;\\
0;2;17;1;1;0;\\
4;7;10;\\
0;-1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
\end{array}\right\} \qquad (3.1.1)$$

$-1; \rightarrow\rightarrow$

The value of $m = 80$ has been taken large enough so that COUNTING ERROR will not arise.* The segements $P_1 P_4$, $P_4 P_7$ and $P_7 P_{10}$ have been excluded since they will have zero strains and cannot form part of the layout of an optimum structure.

The final part of the output for this problem is:

STRUCTURAL MEMBERS	CROSS SECTIONS	VOLUME
43		
−20		
−58		
−19		
33		
−25		
−51	$2{\cdot}000000_{10}+0;$	$4{\cdot}000000_{10}+0;$
−21	$1{\cdot}414214_{10}+0;$	$4{\cdot}000000_{10}+0;$
−46		
26		
39		
56	$1{\cdot}000000_{10}+0;$	$1{\cdot}000000_{10}+0;$
53	$1{\cdot}000000_{10}+0;$	$1{\cdot}000000_{10}+0;$
−54	$4{\cdot}472136_{10}+0;$	$1{\cdot}000000_{10}+1;$
−62		
49	$1{\cdot}414214_{10}+0;$	$4{\cdot}000000_{10}+0;$
−45		
−40		
−61	$2{\cdot}236068_{10}+0;$	$5{\cdot}000000_{10}+0;$
TOTAL VOLUME	$2{\cdot}90000000_{10}+1;$	

(3.1.1)

*The number of constraints cannot be larger than ${}^kC_2$ or 66. The actual number for this problem is 46.

This gives in the first column all the segments along which potential members of the optimum structure may lie, identifying them by their corresponding slack variables and also picking out, with a minus sign, those which will carry compressive loads. This layout of potential members is shown in Fig. 3c. Equilibrium at the nodes shows that many of these members have zero end load and so can be omitted. The resulting structure, which has redundancy of order three, is shown in Fig. 3d. A special case, which is statically determinate and is in fact given by the program, is shown in Fig. 3e. The required cross sectional areas are also shown on this figure. It is to be remarked that the areas given in (3.1.2) are the totals for each member and its mirror image.

The total volume of material required for this design is $V_{min} = 29\ Fd/\sigma$*. This applies to Fig. 3d as well as to Fig. 3e thanks to (1.2.6). It is of interest to compare it with the result given in Ref. 4 in Fig. 25, which solves the same problem using a finer grid and gives $V_{min} = 27{\cdot}6\ Fd/\sigma$. The result for an infinitely small grid as given in Ref. 4, Fig. 16b is $V_{min} = 26\ Fd/\sigma$.

3.2. *Cantilever with Distributed Forces.*

The problem of Fig. 4a can be resolved approximately, using the grid of Fig. 4b, and replacing the distributed forces by the concentrated forces F and $F/2$. The points P_{20}, P'_{20} are taken to be fixed and as in Section 3.1 the members are restricted to those which do not cross the line of symmetry $P_1 P_{19}$. For this problem $n = 33$ and $m = 134$. The virtual displacements corresponding to W_{max} are shown in Fig. 4b and the corresponding structural layout in Fig. 4c. The statically determinate structure given by the program is given in Fig. 4d. All these optimum structures have a volume $V_{min} = 67{\cdot}3\ Fd/\sigma$.

3.3. *Cantilever under Tip Force with Restricted Layout.*

In Fig. 5a the force F is to be reacted at the fixed segment $P_{25} P_{27}$ in such a way that no part of the structure shall lie below $P_1 P_{25}$. The grid used for an approximate analysis is shown in Fig. 5a, which also indicates the $n = 33$ displacement components which define the virtual displacement. This problem has $m = 251$. The general solution is shown in Fig. 5b and the statically determinate solution, given by the program, in Fig. 5c. The required volume of material is $V_{min} = 29\ Fd/\sigma$.

3.4. *Machine Tool Structures.*

The problem of Fig. 6a is basic to many machine tools. The forces F must be balanced by a structure, whose members are not allowed to enter the shaded region of width $2b$ shown in the figure. Fig. 6b shows an identical problem in which the width $2b$ is reduced to zero with the gap, now reduced to a half-line, retained. The grid assumed for this special problem is also shown, as are the displacement components required for the analysis. Symmetry requires that displacement components on $P_1 P_{17}$ normal to the gap should vanish to the left of P_9. They need not vanish to the right of P_9, but must be equal and opposite on either side of the gap. The point P is taken as fixed.†

The solution to this problem is given in Fig. 6c, which has $V_{min} = 13{\cdot}6\ aF/\sigma$. This suggests that the 'exact' solution, with an infinitely small grid, will have the form given in Fig. 6d. This structure is in fact a Michell optimum structure for the region it covers. It consists of two concentrated members in the form of semi-circles joined by a straight tie, together with two 'fans of spokes' or equivalent tapered plates joined with a further tie connecting their centres. The volume of this last structure is

$$V_{min} = 2(2\pi a + 3b)F/\sigma\,, \qquad (3.4.1)$$

which may perhaps be used to assess the efficiency of actual structures.

---

*This corrects an error in Ref. 2 in Fig. 8 and paragraph 3.2. The results given there contradict (1.2.6).

† Rigid body displacements do not affect W.

Some further results for cases where the given forces are inclined at various angles to the line of the cut are given in Fig. 6e. For these problems there is no symmetry and values of $n = 56$ and $m = 188$ were required.* The constraints were generated using the data form of (2.2.2), which permitted the exclusion of the members crossing the gap to be automatically accomplished.

Finally, consider the similar problem of Fig. 7a. Here the excluded region is finite and the problem doubly symmetrical. The assumed form of deformation is shown in Fig. 7b for the ideal case $b = 0$. The corresponding optimum structure, which has a $V_{min} = 6{\cdot}6\, Fa/\sigma$ is given in Fig. 7c. An 'exact' solution for a finite width slot is suggested in Fig. 7d. This has a similar form to that of Fig. 6d and may be interpreted in the same way. Its volume is given by

$$V_{min} = 2(\pi a + b)F/\sigma \qquad (3.4.2)$$

---

*These problems took about 18 minutes of computing time on KDF9.

REFERENCES

No.	*Author(s)*	*Title, etc.*
1	G. Hadley	*Linear Programming.* Addison-Wesley, 1962.
2	W. S. Hemp	Studies in the Theory of Michell Structures. *Proc. 11th Int. Cong. Appl. Mech.*, Munich, 1964.
3	H. S. Y. Chan	Optimum Structural Design and Linear Programming. CoA Rep. Aero 175, 1964. A.R.C. 26 315.
4	W. S. Hemp and H. S. Y. Chan	Optimum Structures. CoA Mem. Aero 70, 1965.
5	W. S. Dorn and H. J. Greenberg	Linear programming and plastic limit analysis of structures. Q. appt. Maths **15**, 155–167, 1957.

APPENDIX

Some Remarks on the Program:

(i) This program is coded by using the original simplex method (Ref. 1 Chapters 4–5). No claim is made that this is the most efficient algorithm. It is chosen because easy modifications can be made to accommodate some special features mentioned below.

To begin with, no phase 1 is necessary because an initial basic feasible solution is given (*see* (2.1.4) with v_{n+j} being basic variables). However, two additional features have to be taken into account. Firstly, the slack variables v_{n+j} are non-negative and are bounded by $2b_j$. The upper bound technique as given in Ref. (1) § 11–7 can be used to take care of this situation without the second line of (2.1.4) appearing explicitly in the simplex formulation. Secondly, the variables v_i $(i = 1, \ldots, n)$ can be either positive or negative. This can be done in actual computation without introducing additional variables (e.g. by writing $v_i = v_i^+ - v_i^-$) by following the procedure of Ref. 5.

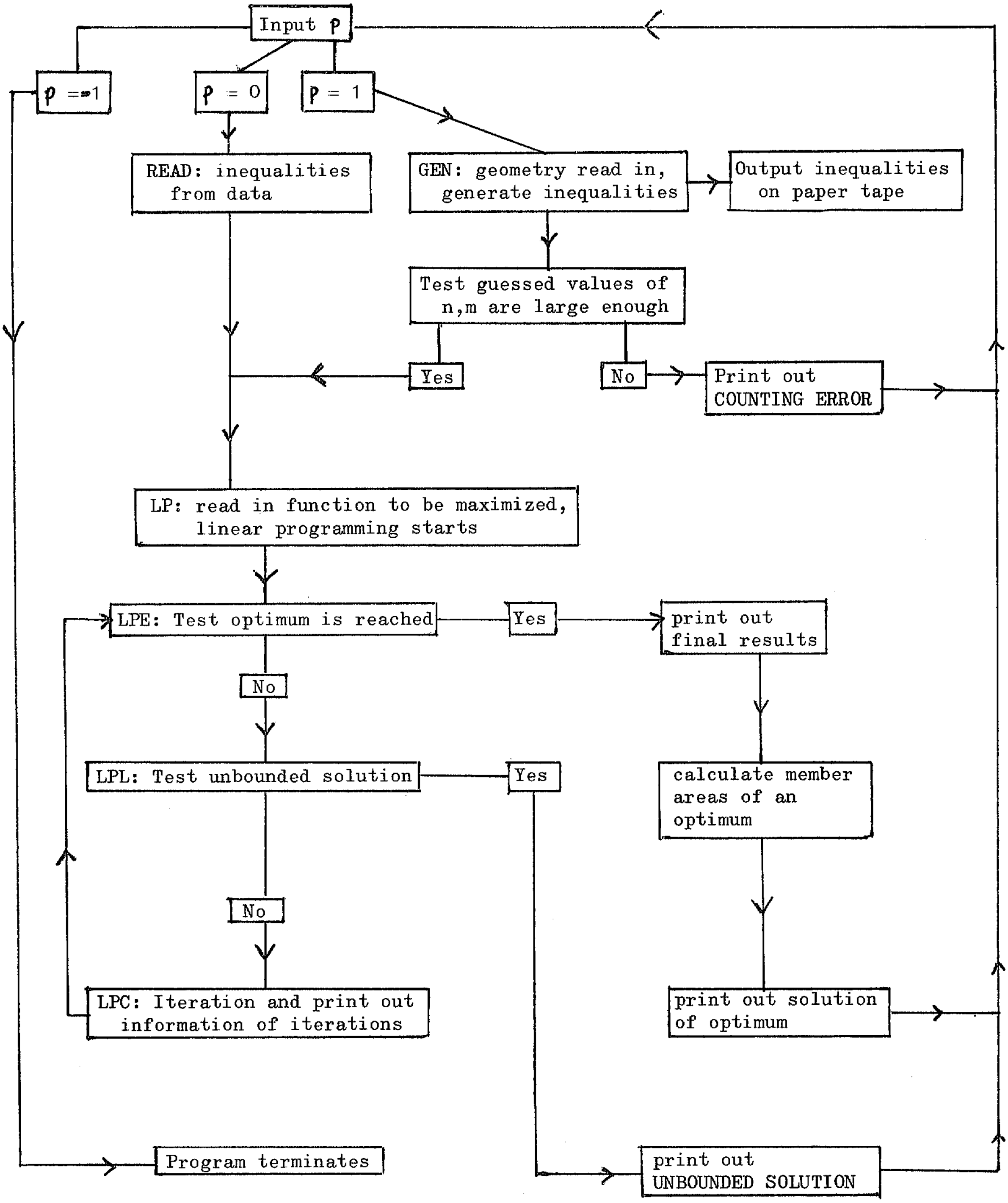
No anti-degeneracy procedure is incorporated in this program.

(ii) The data generating part of the program (i.e. for $p = 1$) works only for plane-frameworks. However, space frames can be calculated if the data is in the form of $p = 0$.

(iii) No special structure of these L.P. problems has yet been observed apart from those mentioned in (i). Although the number of zero coefficients in the constraint matrix are large, it is felt that not much computer time could be gained by taking this into account, nor that computer storage limitation would arise for moderate design problems. However, it is noted that storage problems may cause difficulty in space framework design.

Algol Programme "FAEBDL200KP7" *

Flow Diagram.



* This programme was written by H.S.Y. Chan.

```
begin library A0,A6,A12;
      integer p,n,m;
       open(20);  open(30);
       p:=read(20);
START: n:=read(20);  m:=read(20);
       write text(30,[[10c]]);
 begin array b[0:m,0:n],ub[1:n+m];
       integer i,j;
       integer array x[1:n+m];
       for i:=1 step 1 until n+m do x[i]:=i;
       if p=0 then goto READ else goto GEN;
 READ: for j:=1 step 1 until m do
        begin for i:=0 step 1 until n do b[j,i]:=read(20) end;
       goto LP;
 GEN:  begin integer ng,kg;
             ng:=read(20);  kg:=read(20);
             open(10);   gap(10,240);
        begin array xg,yg,cg,sg[1:kg]; integer array pg,qg,ug,vg[1:kg],
            zg[1:kg×(kg-1)/2];  integer ig,jg,rg,a,bg;
              real w;
         procedure generate(or,oj); integer or,oj;
          begin bg:=0;
            b[ig,0]:=cg[or]↑2+sg[or]↑2;
            if pg[oj]>1 then for a:=1 step 1 until pg[oj]-1 do
               begin b[ig,a]:=0; bg:=bg+1 end;
            if ug[oj]=1 then begin b[ig,pg[oj]]:=cg[or];bg:=bg+1 end;
            if vg[oj]=1 then begin b[ig,bg+1]:=sg[or]; bg:=bg+1 end;
          if bg+1< pg[or] then for a:=bg+1 step 1 until pg[or]-1 do
              begin b[ig,a]:=0; bg:=bg+1 end;
            if ug[or]=1 then begin b[ig,pg[or]]:=-cg[or];bg:=bg+1 end;
            if vg[or]=1 then begin b[ig,bg+1]:=-sg[or]; bg:=bg+1 end;
            if bg<ng then for a:=bg+1 step 1 until ng do
                begin b[ig,a]:=0; bg:=bg+1 end
            end;
        procedure output;
          begin integer mg;
                mg:=0;   gap(10,10);   write text(10,[[c]]);
                for a:=0 step 1 until ng do
              begin mg:=mg+1;
              if mg=5 then begin write text(10,[[c]]);mg:=1 end;
              if b[ig,a]=0 then write(10,format([nd;]),b[ig,a]) else
               write(10,format([-d.ddddddddd₁₀+nd;]),b[ig,a])
               end;  write text(10,[[c]]);  gap(10,10); ig:=ig+1
           end;
   for a:=1 step 1 until kg do begin xg[a]:=read(20);yg[a]:=read(20);
 pg[a]:=read(20);ug[a]:=read(20);vg[a]:=read(20);qg[a]:=read(20) end;
        for a:=1 step 1 until kg do begin if qg[a]≠0 then
        for ig:=1 step 1 until qg[a] do begin
        bg:=(2×kg-a)×(a-1)/2;  zg[ig+bg]:=read(20) end end;
        ig:=1;
        for jg:=1 step 1 until kg-1 do
         begin for rg:=jg+1 step 1 until kg do begin
            cg[rg]:=xg[jg]-xg[rg];sg[rg]:=yg[jg]-yg[rg] end;
          if qg[jg]≠0 then for a:=1 step 1 until qg[jg] do
   begin bg:=(2×kg-jg)×(jg-1)/2;if zg[a+bg]=jg+1 then goto NEXT end;
            generate(jg+1,jg);  output;
```

```
NEXT:        for rg:=jg+2 step 1 until kg do
      begin  if qg[jg]≠0 then for a:=1 step 1 until qg[jg] do
     begin bg:=(2×kg-jg)×(jg-1)/2;if zg[a+bg]=rg then goto SHOT end;
            if cg[rg]=0 then for a:=jg+1 step 1 until rg-1 do
           begin w:=cg[a]; if w=0 then goto SHOT end;
        if sg[rg]=0 then for a:=jg+1 step 1 until rg-1 do
        begin w:=sg[a]; if w=0 then goto SHOT end;
       if cg[rg]≠0 and sg[rg]≠0 then begin
        for a:=jg+1 step 1 until rg-1 do begin
            if cg[a]×sg[a]≠0 then begin w:=cg[a]/cg[rg];
            if w=sg[a]/sg[rg] then goto SHOT end end end;
       generate(rg,jg);  output;
    SHOT:  bg:=0  end end;
    ig:=ig-1;
    gap(10,120);  write(10,format([-nddd;c]),ng);
    write(10,format([-nddd;c]),ig);
    close(10);
    if n≥ng and m>ig then begin n:=ng;m:=ig;goto LP end
     else begin write text(30,[[2c]counting*error[cc]]);goto EXIT
    end end end;

LP:     for i:=0 step 1 until n do b[0,i]:=read(20);
        for j:=1 step 1 until m do ub[j+n]:=b[j,0]+b[j,0];
        for i:=1 step 1 until n do ub[i]:=0;
        write text(30,[Iter[8s]Left[8s]Enter[8s]Function[3c]]);
     begin integer r,s,q,k;
            real ds,dr,v;
           array y[0:n],z[0:m];
            k:=1;
     LPE:   ds:=0;
            for i:=1 step 1 until n do
        begin if b[0,i]>0 and abs(x[i])≤n then begin x[i]:=-x[i];
                 for j:=m step -1 until 0 do b[j,i]:=-b[j,i] end;
            if ds>b[0,i] then begin ds:=b[0,i];s:=i end
           end;  if ds=0 then goto FINE;
     LPL:   dr:=999999;  r:=0;
            for j:=1 step 1 until m do
     begin if b[j,s]<0 and abs(x[j+n])>n then begin x[j+n]:=-x[j+n];
               b[j,0]:=ub[abs(x[j+n])]-b[j,0];
             for i:=1 step 1 until n do b[j,i]:=-b[j,i]
      end;
      if b[j,s]>0 then begin if dr>b[j,0]/b[j,s] then
               begin dr:=b[j,0]/b[j,s];r:=j end end
        end;
      if abs(x[s])>n and ub[abs(x[s])]<dr then begin x[s]:=-x[s];
           for j:=0 step 1 until m do begin b[j,s]:=-b[j,s];
            b[j,0]:=b[j,0]+ub[abs(x[s])]×b[j,s] end;goto LPE end;
      if r=0 then begin write text(30,[[cc]Unbounded*Solution[cc]]);
           goto EXIT end;
     LPC:  for i:=n step -1 until 0 do
           begin y[i]:=b[r,i]; b[r,i]:=0 end;
           for j:=m step -1 until 0 do
           begin z[j]:=b[j,s]; b[j,s]:=0 end;
            v:=y[s];  z[r]:=-1;   y[s]:=1,
           for i:=n step -1 until 0 do
         begin for j:=m step -1 until 0 do
           begin b[j,i]:=b[j,i]-z[j]×y[i]/v;
                 if abs(b[j,i])<⏨-10 then b[j,i]:=0
           end end;
```

```
          q:=x[s];  x[s]:=x[n+r];   x[n+r]:=q;
          write(30,format([nddd]),k);
          write(30,format([7s-nddd]),x[s]);
          write(30,format([8s-nddd]),x[n+r]);
          write(30,format([8s-d.dddddd₁₀+nd;c]),b[0,0]);
          k:=k+1;   goto LPE
     end;
FINE:  write text(30,[[6c]Result[4c]]);
 for j:=1 step 1 until m do begin write(30,format([8s-nddd]),x[n+j]);
     write(30,format([12s-d.dddddd₁₀+nd;c]),b[j,0]) end;
   write text(30,[[4c]Slack[cc]]);
   for i:=1 step 1 until n do begin write(30,format([8s-nddd]),x[i]);
 write(30,format([12s-d.dddddd₁₀+nd;c]),b[0,i]) end;
rite text(30,[[8c]Structural*members[8s]cross*sections[12s]volume[4c]]);
   begin real a,f;integer v;
        f:=0;  for j:=1 step 1 until m do begin
       v:=abs(x[n+j]);
    if v>n and b[j,0]=0 then write(30,format([8s-ndddc]),x[n+j]);
      if v>n and b[j,0]=ub[v] then
         write(30,format([8s-ndddc]),-x[n+j]) end;
       for i:=1 step 1 until n do begin
         if abs(x[i])>n then begin
         write(30,format([8s-nddd]),x[i]);
      if b[0,i]≠0 then begin v:=abs(x[i]);a:=b[0,i]×sqrt(ub[v]/2);
         write(30,format([12s-d.dddddd₁₀+nd;]),a);
      a:=a×sqrt(ub[v]/2);  f:=f+a;
      write(30,format([12s-d.dddddd₁₀+nd;c]),a) end else
         write text(30,[[c]]) end end;
           write text(30,[[4c]total**volume]);
           write(30,format([12s-d.ddddddd₁₀+nd;c]),f)
          end;
:XIT:   write text(30,[[p]]);
       p:=read(20);
       if p>0 then goto START;
       close(20);  close(30)
end
d→
```

Note: To avoid rounding-off errors the programme sets numbers $b[j, i] < 10^{-10}$ equal to zero. This restriction has been generally satisfactory for problems solved on the Oxford University *KDF9* computer, but one rather ill conditioned problem required a coarser limit of 10^{-7}. Users may find it necessary to introduce such a coarser limit on other machines for some or all of their problems.

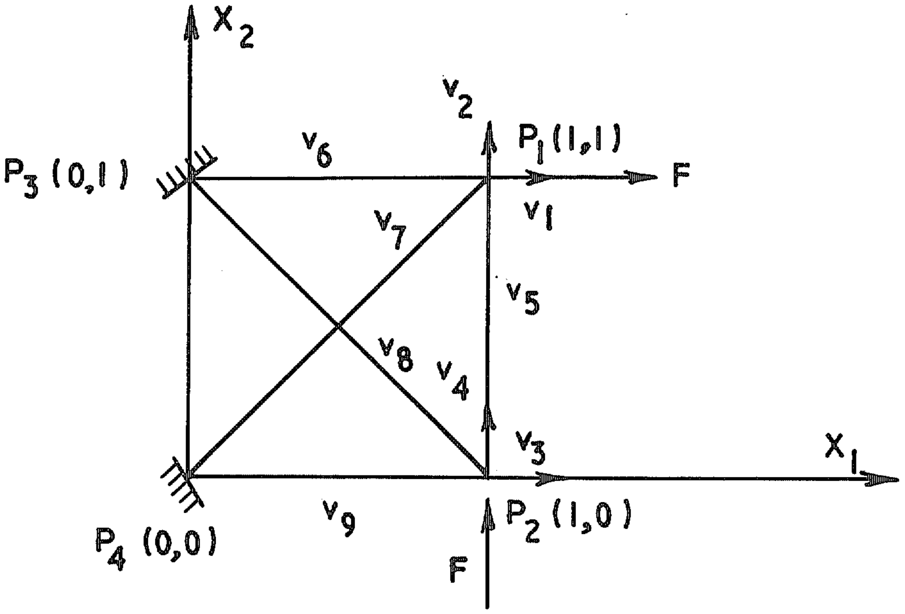

X2
v2
P1 (1,1)
F
P3 (0,1)
v6
v7
v1
v5
v8
v4
v3
X1
v9
P2 (1,0)
P4 (0,0)
F

FIG. 1.

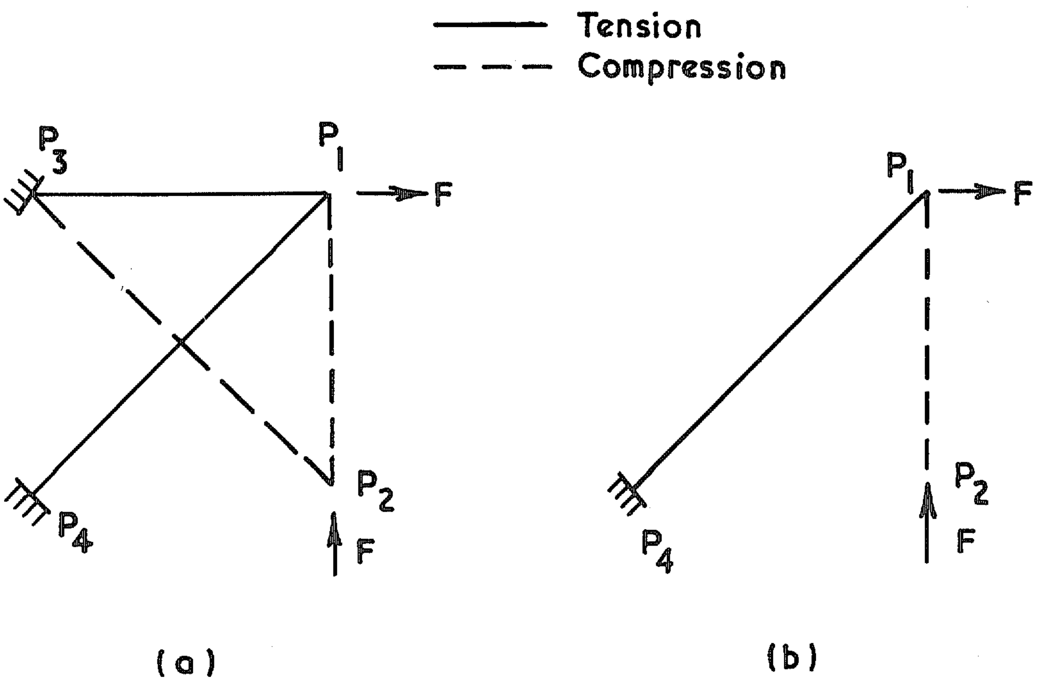
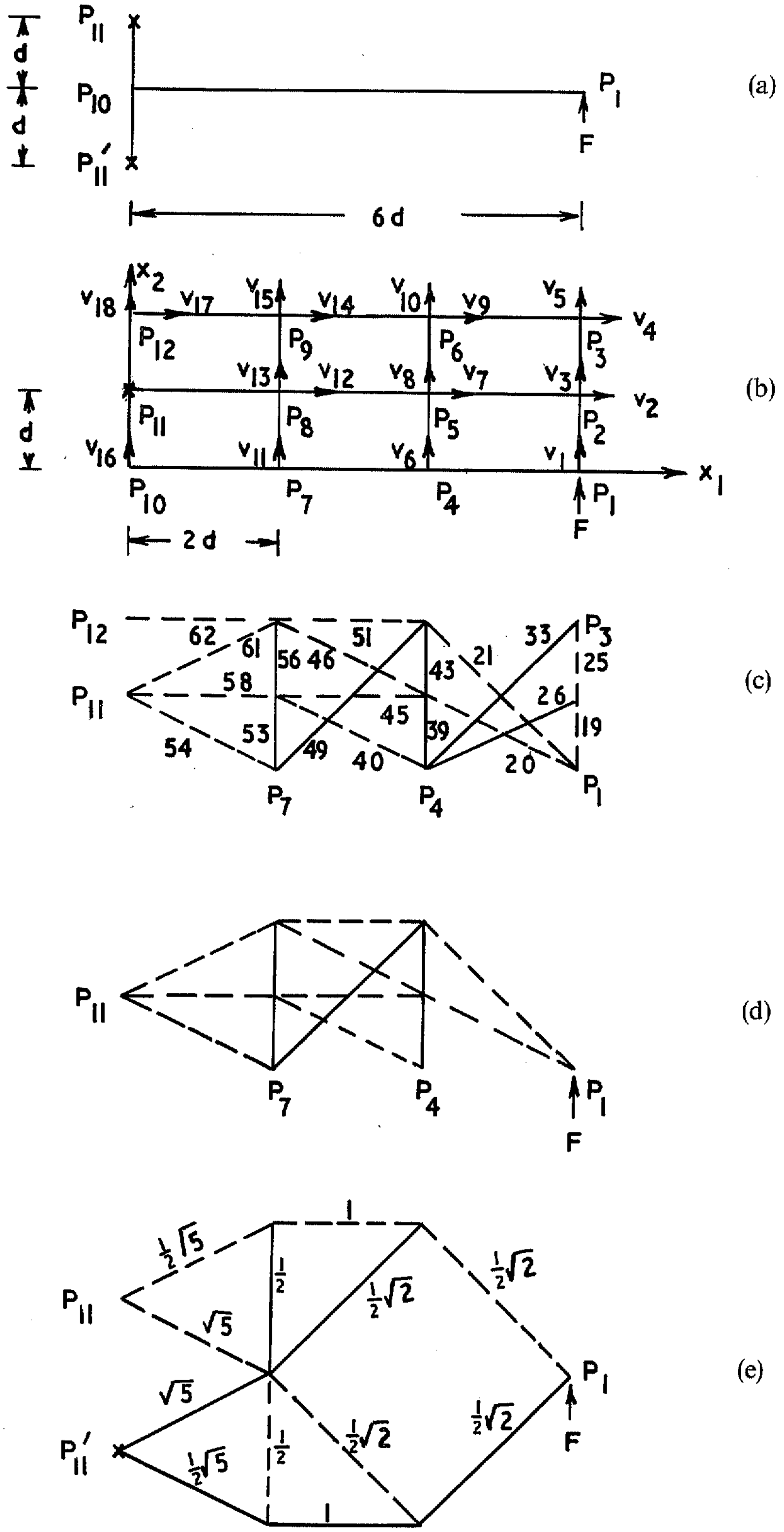
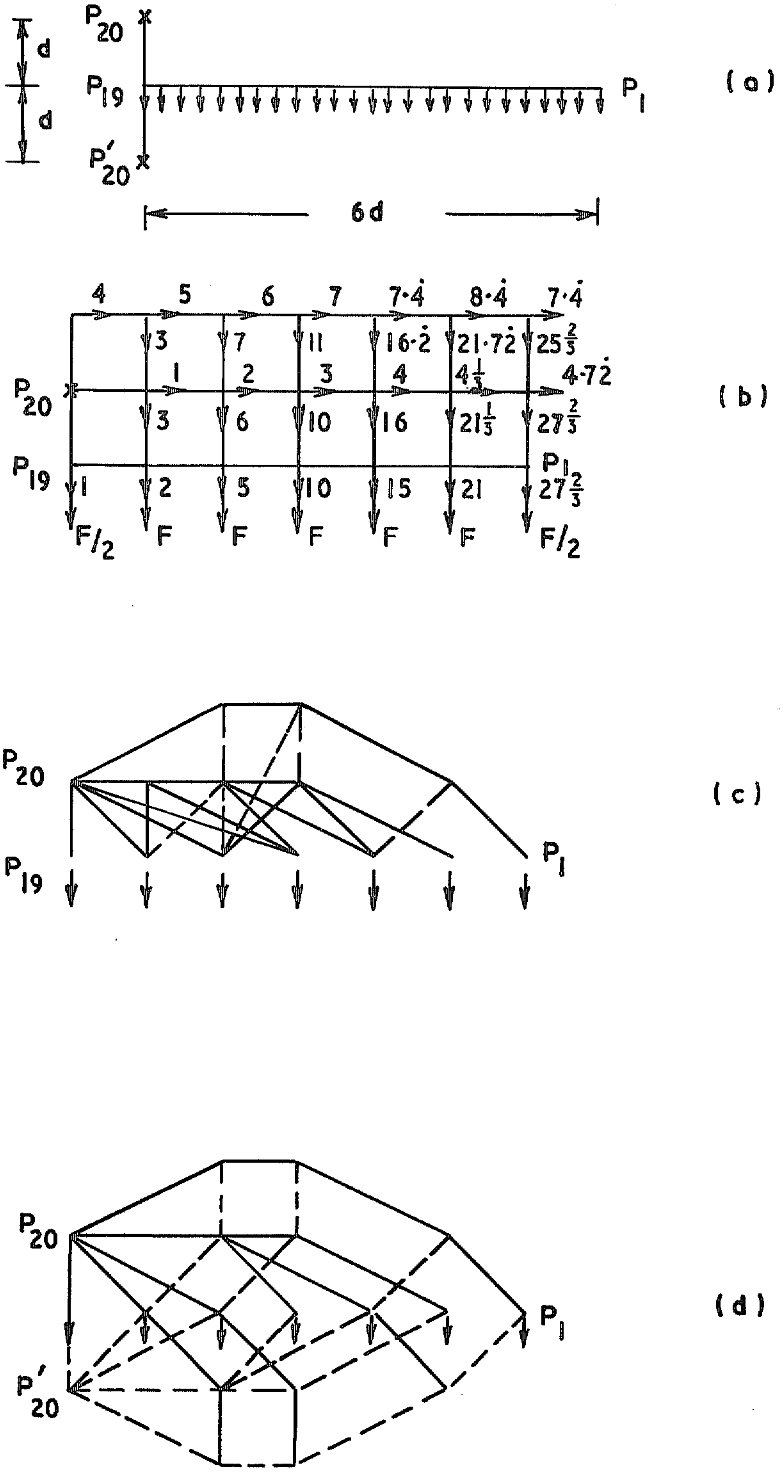

Tension
Compression
P3
P1
F
P2
P4
F
(a)
P1
F
P2
P4
F
(b)

FIG. 2.



(a)
(b)
(c)
(d)
(e)
d
6 d
2 d
F
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P11′
P12
x1
x2
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
v11
v12
v13
v14
v15
v16
v17
v18
62
61
56
46
51
43
21
33
25
58
26
45
39
19
53
54
49
40
20
1
½√5
√5
½
½√2

FIG. 3.



d
d
P20
P19
P'20
P1
6d
(a)
4
5
6
7
7·4̇
8·4̇
7·4̇
3
7
11
16·2̇
21·72̇
25⅔
1
2
3
4
4⅓
4·72̇
P20
3
6
10
16
21⅓
27⅔
P19
P1
1
2
5
10
15
21
27⅔
F/2
F
F
F
F
F
F/2
(b)
P20
P19
P1
(c)
P20
P'20
P1
(d)

FIG. 4.

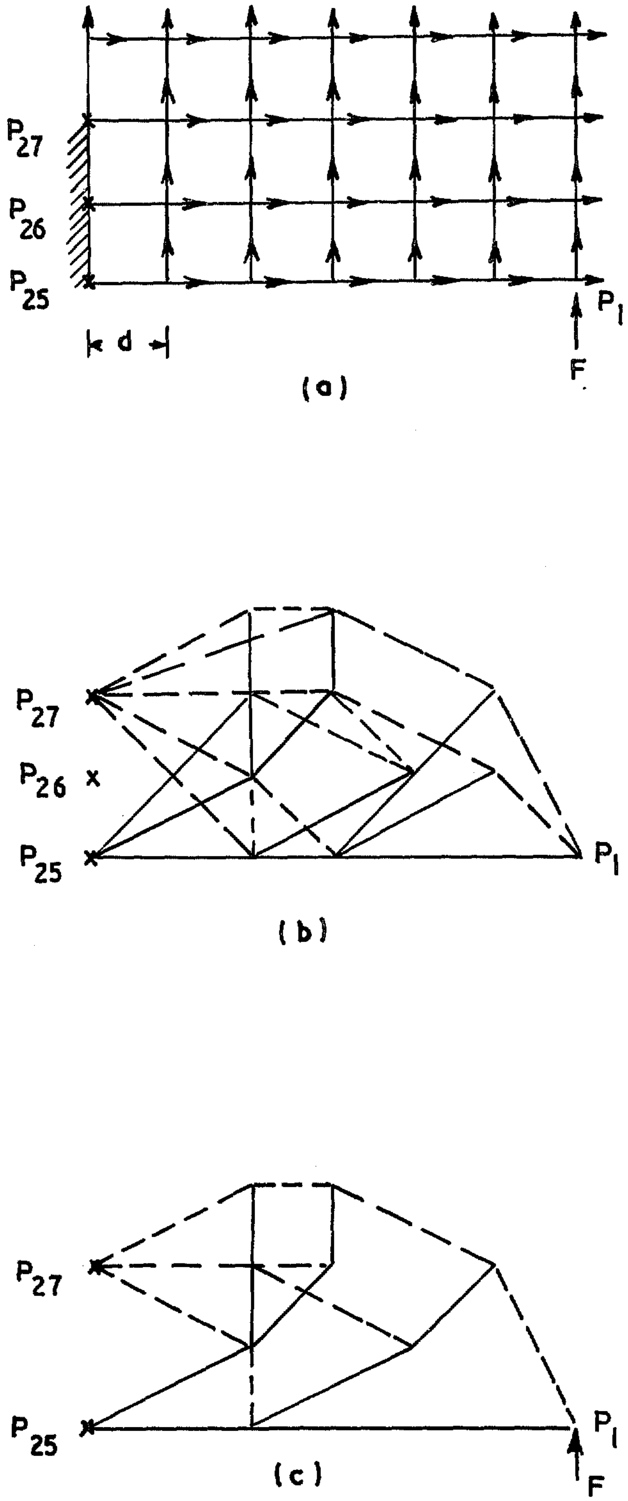


FIG 5.

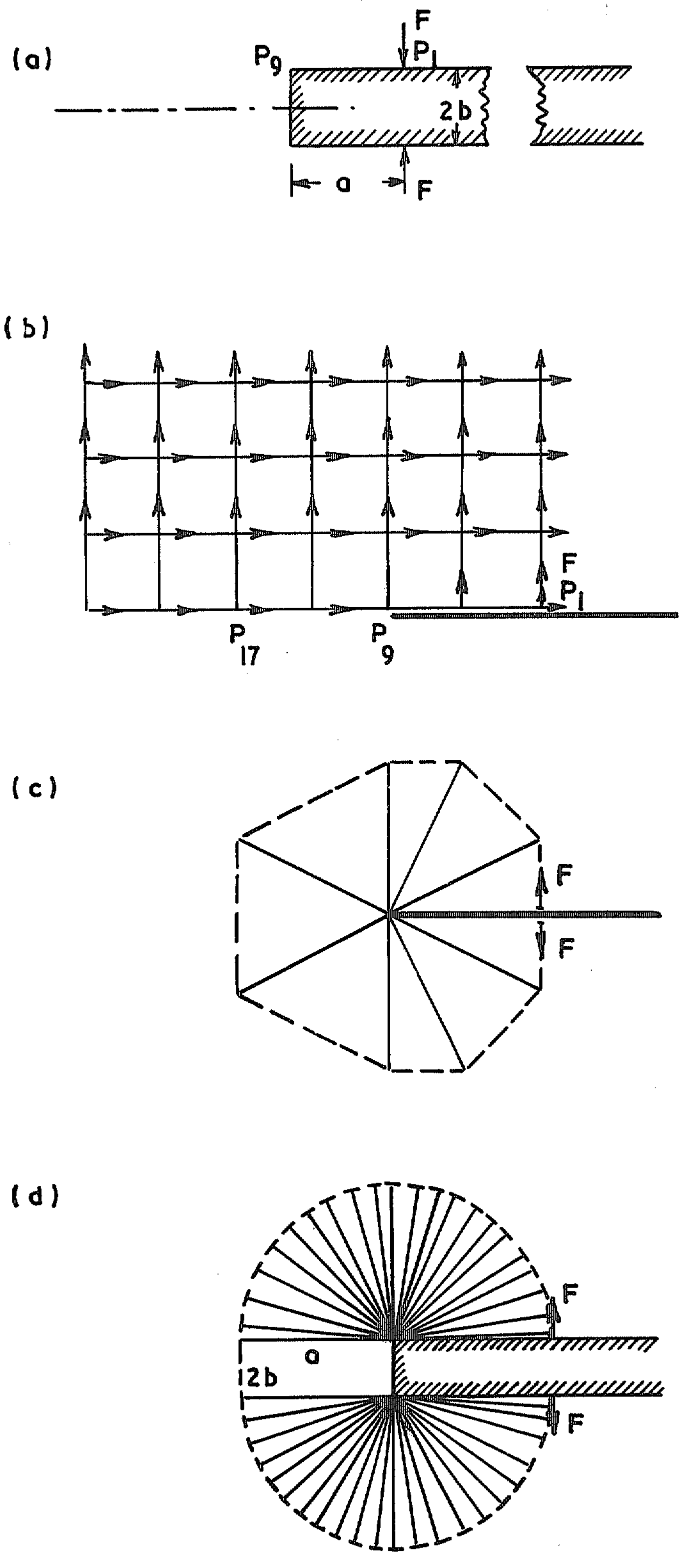


FIG. 6.

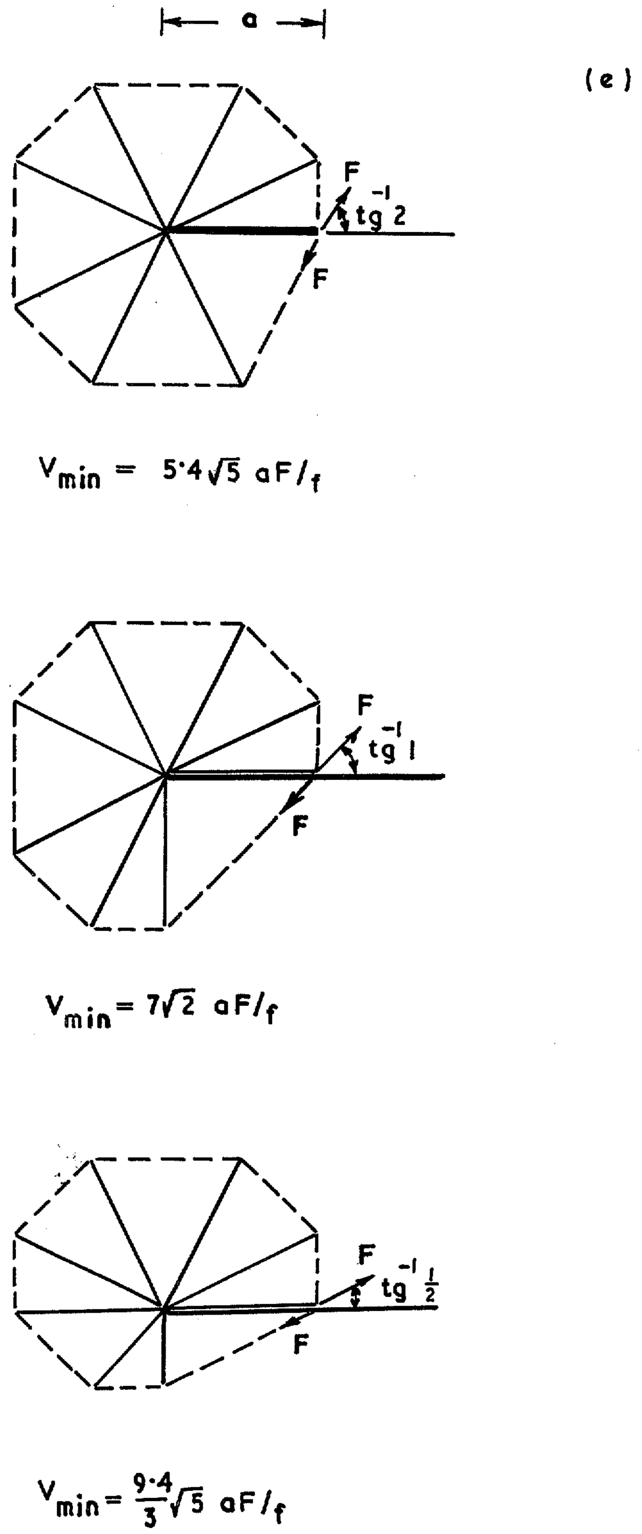

a
(e)
F
tg⁻¹ 2
F
V_min = 5·4√5 aF/f
F
tg⁻¹ 1
F
V_min = 7√2 aF/f
F
tg⁻¹ 1/2
F
V_min = 9·4/3 √5 aF/f

FIG. 6e.

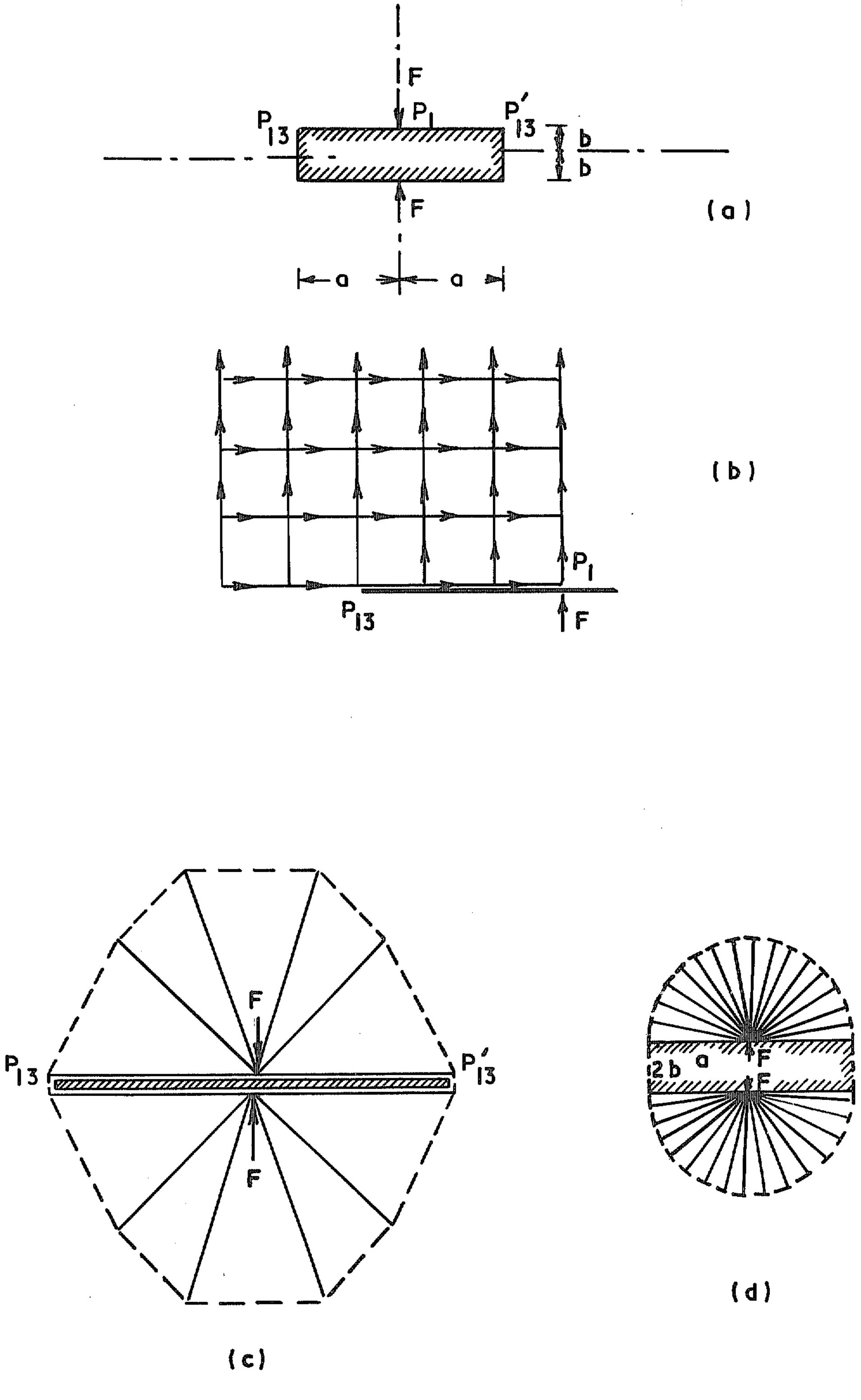


FIG. 7.

Printed in Wales for Her Majesty's Stationery Office by Allens Printers (Wales) Limited

Dd. 501372 K.5

R. & M. No. 3632

Published by
HER MAJESTY'S STATIONERY OFFICE

To be purchased from
49 High Holborn, London WC1
13a Castle Street, Edinburgh EH2 3AR
109 St Mary Street, Cardiff CF1 1JW
Brazennose Street, Manchester M60 8AS
50 Fairfax Street, Bristol BS1 3DE
258 Broad Street, Birmingham 1
7 Linenhall Street, Belfast BT2 8AY
or through any bookseller

R. & M. No. 3632

SBN 11 470332 9