COMPUTER SOLUTION OF TWO - DIMENSIONAL STRUCTURES USING THE FINITE ELEMENT METHOD WITH SUBSTRUCTURE ANALYSIS

ANDREW C. WHYTE

1970

M.Sc.

WHYTE A. C.

UNIVERSITY of STRATHCLYDE

GLASGOW

Department of Mechanical, Civil and Chemical Engineering.

COMPUTER SOLUTION OF TWO-DIMENSIONAL STRUCTURES USING THE FINITE ELEMENT METHOD WITH SUBSTRUCTURE ANALYSIS.

Thesis submitted for the Degree of Master of Science

By

ANDREW C. WHYTE

97 Baldwin Ave.,

Glasgow W.3.

Nov. 1970

CONTENTS

	page
ABSTRACT	1
NOMENCLATURE	2
INTRODUCTION	13
1. THEORY	16
1.1 Stiffness properties of triangular elements with in-plane forces using the direct stiffness method.	16
1.2 Assembly of elements.	25
1.3 Substructure analysis.	31
1.4 Assembly of substructures.	47
2. COMPUTER ANALYSIS	53
2.1 Structure numbering system.	53
2.2 Method of storing the stiffness matrix.	58
2.3 Rearranged order of the substructure stiffness matrix.	63
2.4 Formation of the substructure boundary stiffness matrix.	64
2.5 Determination of the boundary nodal point displacements by GAUSS-SEIDEL iteration.	66
2.6 Introduction of boundary conditions.	69
2.7 Nodal point stresses.	72

page

2.8 Computer check and comparason of results. 75

2.9 Example of scope available. 95

3. LINE BY LINE DESCRIPTION OF COMPUTER PROGRAMS 109

3.1 Program PART A. 111

3.2 Program PART B. 139

3.3 Program PART B with Disc storage. 152

3.4 Program PART C. 158

4. USER GUIDE TO THE COMPUTER PROGRAMS AND INPUT DATA REQUIREMENTS. 167

4.1 Program PART A. 173

4.2 Program PART B. 177

4.3 Program PART B with Disc storage. 182

4.4 Program PART C. 183

4.5 Altering the size of array dimension statements 185

5. PROGRAM LISTING 188

5.1 Program PART A. 188

5.2 Program PART B. 212

5.3 Program PART B with Disc storage. 226

5.4 Program PART C. 248

ACKNOWLEDGEMENTS 260

REFERENCES 261

APPENDIX 1. 262

ABSTRACT

The scope of this thesis covers the following topics

The stiffness properties of triangular elements with in-plane forces is derived, and the assembly of the elements within the assembled stiffness matrix is discussed.

The theory of substructure analysis by the relaxation method is presented.

A computer analysis of the substructure theory is given, together with program listing and line by line description of the programs.

The computer programs have been written in I.C.L. 1900 FORTRAN and are restricted to two-dimensional plane stress or plane strain structures of constant thickness , homogeneous material properties and use triangular elements with in-plane forces. The programs are arranged to suit a computer with 18K core store. This enables about 500 elements to be analysed. If direct access disc storage is available, then this figure may be increased to about 1500 elements.

Although the computer programs listed here are subject to the above restrictions, the basic substructure analysis can be applied to three dimensional structures, to any element configuration and all the advantages of the finite element method may be incorporated in more sophisticated programs.

NOMENCLATURE (theory)

The following is a list of the principal notation used in the theory

ε_x Strain in the X direction.

ε_y Strain in the Y direction.

γ_{xy} Shear strain.

σ_x Normal stress in the X direction.

σ_y Normal stress in the Y direction.

τ_{xy} Shear stress.

U_x Displacement in the X direction.

U_y Displacement in the Y direction.

i, j, k Element, nodal point subscripts.

X_i, X_j, X_k Element, nodal point X co-ordinates.

Y_i, Y_j, Y_k Element, nodal point Y co-ordinates.

$C_1, C_2, C_3, C_4, C_5, C_6$ Constants which define the strain distrubution within an element.

X_{ji}, X_{ki}, X_{jk} Element dimensions, X direction.

Y_{ji}, Y_{ki}, Y_{jk} Element dimensions, Y directions.

A_{ijk} Area of triangular element.

E Young's modulus.

ν Poisson's ratio.

$[U], U$ Nodal point displacement matrix.

$[\varepsilon]$ Strain matrix.

$[A]$ Matrix relating element strain to nodal point displacement.

$[\sigma]$ Stress matrix.

$[B]$ Matrix relating element stress to element strain.

$[W]_{EXT}$ External work done matrix.

$[W]_{INT}$ Internal work done matrix.

$[P], P$ Nodal point external force matrix.

$[U^*]$ Virtual nodal point displacement matrix.

$[\varepsilon^*]$ Virtual strain matrix.

$[\ ]^T$ Matrix transpose.

t Element thickness.

$[K^{(e)}]$, $K^{(e)}$ Element stiffness matrix for the (e) element.

$[K]$, K Assembled stiffness matrix for the complete structure.

K_b Assembled boundary stiffness matrix for the complete structure.

P_b Boundary nodal point external force matrix.

P_i Interior nodal point external force matrix.

U_b Boundary nodal point displacement matrix.

U_i Interior nodal point displacement matrix.

R_b Boundary reaction matrix.

F_b Resultant boundary force matrix.

(r) The (r) substructure.

(e) The (e) element.

$K^{(r)}$ Assembled substructure stiffness matrix for the (r) substructure.

$K_b^{(r)}$ Assembled substructure boundary stiffness matrix.for the (r) substructure.

$P^{(r)}$ (r) substructure, nodal point external force matrix.

$U^{(r)}$ (r) substructure, nodal point displacement matrix.

$P_b^{(r)}$ (r) substructure, boundary nodal point external force matrix.

$P_i^{(r)}$ (r) substructure, interior nodal point external force matrix.

$U_b^{(r)}$ (r) substructure, boundary nodal point displacement matrix.

$U_i^{(r)}$ (r) substructure, interior nodal point displacement matrix.

$R_b^{(r)}$ (r) substructure, boundary reactions.

$K_{xx}, K_{xy}, K_{yx}, K_{yy}$ Terms of the 2x2 submatrix, see pages 23 and 24.

$K_{bb}, K_{bi}, K_{ib}, K_{ii}$ Stiffness matrix for the partitioned formation of stiffness matrix K , see pages 32 to 34.

$K_{bb}^{(r)}, K_{bi}^{(r)}, K_{ib}^{(r)}, K_{ii}^{(r)}$ As above but for the (r) substructure, see page 43.

NOMENCLATURE (computer analysis)

The following is a list of the principal variables used in the computer analysis.

NOTE — One dimensional array, stores the rearranged nodal point numbering order of the substructure stiffness matrix $K^{(r)}$.

NCODE — One dimensional array, stores the global boundary nodal point number order of the (r) substructure.

NUMNP — Number of nodal points in the (r) substucture.

NUMBP — Number of substructure boundary nodal points in the (r) substructure.

NUMGBP — Number of global boundary nodal points in the complete structure.

NUMEL — Number of elements in the (r) substructure.

NUMSUB — Number of substructures.

E — Young's modulus.

XU — Poisson's ratio.

NSUB — The (r) substructure number.

NONZIF — Number of interior nodal points with non-zero external forces within the (r) substructure.

NONZBF — Number of global boundary nodal points with non-zero external forces.

NONZBD — Number of global boundary nodal points with non-zero initial displacements.

NPNUM — One dimensional array, stores the (r) substructure nodal point numbers in consecutive order.

XORD, YORD — One dimensional array, storesthe (r) substructure nodal point X and Y co-ordinates respectively.

NUME — One dimensional array, stores the (r) substructure element numbering order.

NPI, NPJ, NPK — One dimensional arrays, store the three element nodal point numbers i, j, k respectively, for elements within the (r) substructure.

NBP — One dimensioal array, stores the substructure boundary nodal point numbering order for the (r) substructure.

NGBP — One dimensional array, temporary storage of the global boumdary nodal point numbering order of

the (r) substructure.

PIX, PIY — One dimensional arrays, store the (r) substructure interior nodal point external forces in X and Y directions respectively.

AJ, AK — One dimensional arrays, store the element dimensions in the X direction for elements within the (r) substucture.

BJ, BK — One dimensional arrays, store the element dimensions in the Y direction for elements within the (r) substructure.

AREA — Area of the (e) element.

A — Two dimensional array, stores the relationship between element strains and element nodal point displacements.

B — Two dimensional array, stores the relationship between element stress and element strain.

S — Two dimensional array, stores the (e) element 6x6 stiffness matrix.

LM — One dimensional array, stores temporary the element nodal point numbers i, j, k.

NP — Two dimensional array, stores the identity matrix holding the adjacent nodal point numbers for the assembly of stiffness matrix $K^{(r)}$ and K_e .

SXX, SXY, SYX, SYY — Two dimensional arrays, stores the assembled non-zero stiffness terms of matrix $K^{(r)}$ and K_e.

NAP — One dimensional array, stores the number of adjacent nodal points + 1 to any one nodal point.

NUMBER — stores the size of the square matrix $K_{ii}^{(r)}$.

SII — Two dimensional array, stores the square matrix $K_{ii}^{(r)}$ and $\left(K_{ii}^{(r)}\right)^{-1}$

SIB — Two dimensional array, stores two columns of matrix $K_{ib}^{(r)}$

SM — Two dimensional array, stores two columns of matrix $(K_{ii}^{(r)})^{-1}K_{ib}^{(r)}$

SBI — Two dimensional array, stores two rows of matrix K

SB — Two dimensional array, stores a 2×2 submatrix of $K_{bb}^{(r)}$.

UIX, UIY — One dimensional arrays, store the interior nodal point displacements in X and Y directions respectivelY

due to interior nodal point external forces with the boundary fixed.

RBX, RBY — Stores, for a boundary nodal point the boundary reaction forces in X and Y direction respectively.

NUM — Stores the number of adjacent nodal points + 1 for a particular nodal point.

NUMRBP — Number of restrained global boundary nodal points

NRBP — One dimensional array, stores the restrained global boundary nodal point numbering order.

NFIX — One dimensional array, stores the boundary condition code for each restrained global boundary nodal point.

SLOPE — One dimensional array, stores the slope of angle ϕ see page 69 .

FBX, FBY — One dimensional arrays, store the resultant boundary force in X and Y directions respectively.

UBX, UBY — One dimensional arrays, store the global boundary nodal point displacements in X and Y directions respectively.

NCPIN	Cycle print interval.
NOPIN	Output interval of results.
NCYCM	Iteration cycle limit.
TOLER	Iteration tolerance limit.
XFAC	Over-relaxation factor.
SUM	Sum of the force unbalance.
NCYCLE	Iteration cycle number.
UX, UY	One dimensional arrays, store the (r) substructure nodal point displacements in X and Y directions respectively.
X,Y,XY	Stores the element/interior nodal point X and Y normal stress and the shear stress respectively.
SIGXX, SIGYY, SIGXY	One dimensional arrays, store the element X and Y normal stress and shear stress respectively for elements within the (r) substructure.
XMAX, XMIN	Stores the element/interior nodal point maximum and minimum principal stress respectively.
PA	Stores the direction of the max. principal stress from

the X axis.

f_{xx} f_{xy} f_{yx} f_{yy} Terms of the flexibility matrix, see page 67.

f^*_{xx} f^*_{xy} f^*_{yx} f^*_{yy} Terms of the modified flexibility matrix, see page 70.

INTRODUCTION

The finite element method of analysis has now become an established method for the complete analysis of arbitary shaped structures. The usefulness of the method would not be possible were it not for the development of the high speed digital computer.

One severe limitation presented itself; in that the storage of the assembled stiffness matrix required $4n^2$ terms to be stored in the computer. n being the total number of nodal points . To analyse a structure adequately requires several hundred of these nodal points, and as a result only really large computers had the storage capacity necessary to analyse the structures.

Over the years various programming techniques have been developed to take account of the fact that the assembled stiffness matrix is symmetric and contains a very large number of zero terms.

These techniques include

Storing only the upper or lower triangle of the matrix.

Storing only the non-zero terms of the matrix (ref.5).

Storing only the non-zero terms of the upper or lower half band of a tri-diagonal matrix (ref.8).

These techniques did enable larger structures to be analysed on smaller computers; but even so it still required a fairly large computer for proper analysis. For example, if we consider a two-dimensional structure idealized using triangular elements with in-plane forces, and storing only the non-zero terms of the assembled stiffness matrix would require a 32K machine to handle

340 nodal points giving approximately 550 elements.

For users of small computers two possible solutions are available.

1. To store the assembled stiffness matrix on either direct access Disc store or random access Drum store.
2. To arrange the computer program to suit substructure analysis.

Disc or Drum storage can certainly get over the storage problem, but unfortunately the time required to obtain a solution increases and is likely to become prohibitive.

Thisleaves only substructure analysis as a possible means of solution within a reasonable computing time.

Substructure analysis involves dividing the complete structure into a number of substructures. The stiffness properties of the elements within a substructure are determined and assembled into an assembled substructure stiffness matrix, which is then reduced to form an assembled boundary stiffness matrix. The assembled substructure boundary stiffness matrix is formed for each sub-structure, and then assembled to form an assembled boundary stiffness matrix for the complete structure. This matrix is of much smaller size than that of the assembled stiffness matrix without substructure analysis, since it only contains stiffness terms of those nodal points which are specified as "boundary" nodal points.

Using the assembled boundary stiffness matrix and given applied external forces on the structure, and specified boundary conditions,

the boundary nodal point displacements can be obtained. Each substructure can now be analysed separately, and the stresses and nodal point displacements obtained for the elements within the bounds of the substructure.

1. THEORY

1.1 Stiffness properties of triangular elements with In-Plane forces (ref. 1, 2, and 6).

Basic requirements and assumptions

For triangular elements with in-plane forces the basic requirements are.

That compatibility of displacements be maintained on the common boundary of any two adjacent triangular elements. This implies that lines initially straight remain straight in their displaced position.

Equilibrium of nodal point forces must be maintained . This implies that the internal reactions at the nodal points must balance the external applied force at the nodal point.

The above requirements may be satisfied if the following assumptions are made.

The strains $\varepsilon_x, \varepsilon_y$ and γ_{xy} are assumed to be constant within the element. This implies that the stresses within and which act on the edges of each element are also constant.

That the stresses may be replaced by stress resultants (nodal point forces) acting at the corners of the element.

The displacement variation within each element is assumed to be given by the two liner polynomials.

$$U_x = C_1 + C_2 X + C_3 Y \qquad (1)$$

$$U_y = C_4 + C_5 X + C_6 Y \qquad (2)$$

The six coefficients $C_1, C_2, C_3, C_4, C_5, C_6$ may be found from boundary conditions at the three vertices of the triangle, fig. 1.

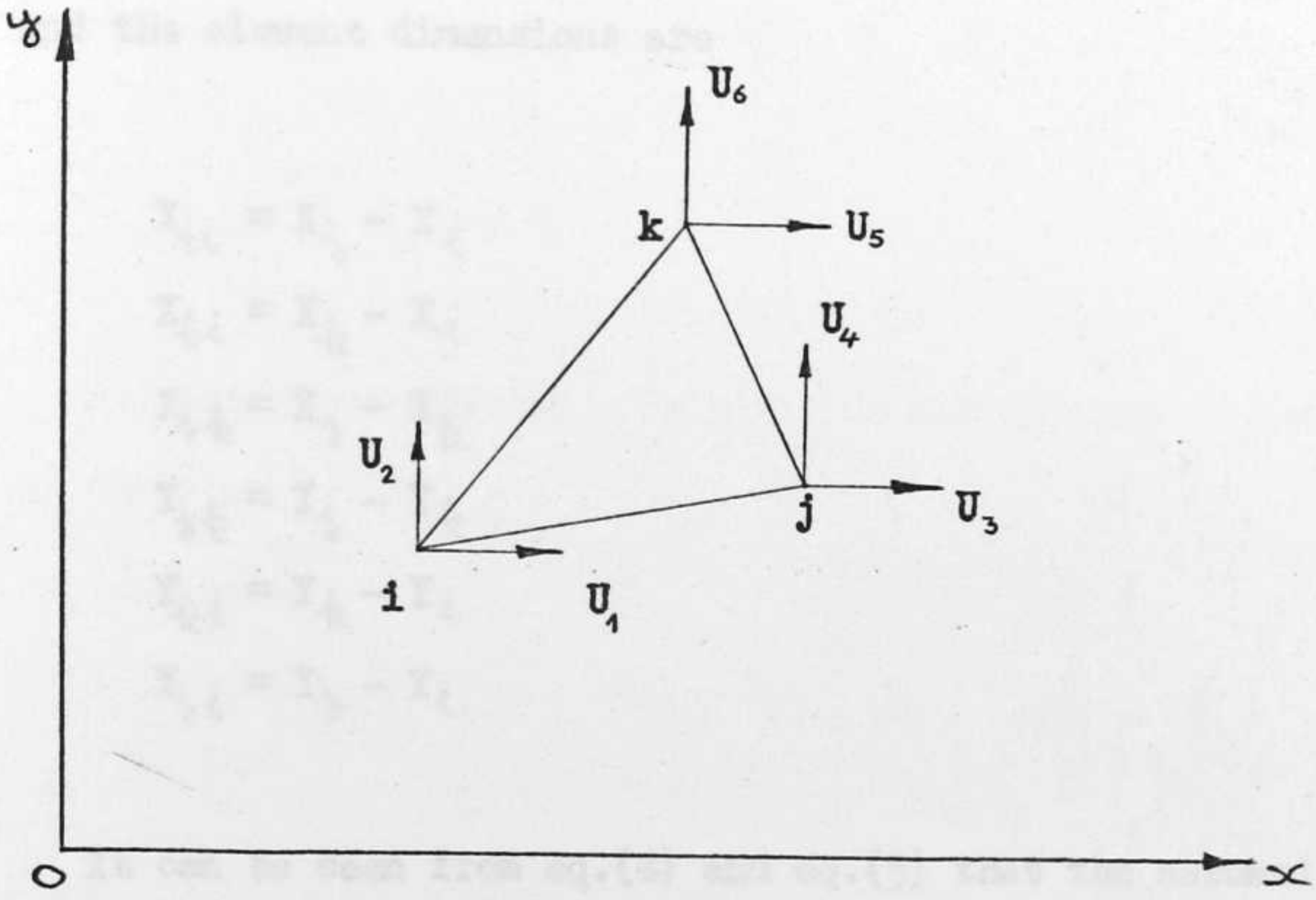


Fig. 1, element co-ordinates.

Boundary conditions are

$$\begin{aligned} U_x &= U_1 \ ; \quad U_y = U_2 \ \text{ at } (X_i, Y_i) \\ U_x &= U_3 \ ; \quad U_y = U_4 \ \text{ at } (X_j, Y_j) \\ U_x &= U_5 \ ; \quad U_y = U_6 \ \text{ at } (X_k, Y_k) \end{aligned} \qquad (3)$$

Substituting boundary conditions (3) in eq.(1) and eq.(2) gives

$$\begin{aligned} U_x = \frac{1}{2A_{ijk}} \Big[& \{Y_{jk}(X-X_k) - X_{jk}(Y-Y_k)\} U_1 \\ & + \{Y_{ki}(X-X_i) - X_{ki}(Y-Y_i)\} U_3 \\ & + \{-Y_{ji}(X-X_j) + X_{ji}(Y-Y_j)\} U_5 \Big] \end{aligned} \qquad (4)$$

$$U_y = \frac{1}{2A_{ijk}}\Big[\{Y_{jk}(X-X_k) - X_{jk}(Y-Y_k)\}U_2 + \{Y_{ki}(X-X_i) - X_{ki}(Y-Y_i)\}U_4 + \{-Y_{ji}(X-X_j) + X_{ji}(Y-Y_j)\}U_6\Big] \quad (5)$$

where

$$A_{ijk} = (X_{ji}Y_{ki} - Y_{ji}X_{ki})/2 \quad (6)$$

= Area of triangle i,j,k.

and the element dimensions are

$$\begin{aligned} X_{ji} &= X_j - X_i \\ X_{ki} &= X_k - X_i \\ X_{jk} &= X_j - X_k \\ Y_{jk} &= Y_j - Y_k \\ Y_{ki} &= Y_k - Y_i \\ Y_{ji} &= Y_j - Y_i \end{aligned} \quad (7)$$

It can be seen from eq.(4) and eq.(5) that the assumed displacements of any points within or on the edge of the triangular element vary linearly and depend only on the co-ordinates of the element and on the displacement of the vertices of the triangle.

For small deformations the strain - displacement relationship is linear and is given by

$$\varepsilon_x = \frac{\partial U_x}{\partial X} \quad (8)$$

$$\varepsilon_y = \frac{\partial U_y}{\partial Y} \quad (9)$$

$$\gamma_{xy} = \frac{\partial U_x}{\partial Y} + \frac{\partial U_y}{\partial X} \quad (10)$$

Differentiating eq.(4) and eq.(5) we obtain the element strains in terms of the nodal point displacements.

In matrix notation

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \frac{1}{2A_{ijk}} \begin{bmatrix} Y_{jk} & 0 & Y_{ki} & 0 & -Y_{ji} & 0 \\ 0 & -X_{jk} & 0 & -X_{ki} & 0 & X_{ji} \\ -X_{jk} & Y_{jk} & -X_{ki} & Y_{ki} & X_{ji} & -Y_{ji} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{bmatrix} \qquad (11a)$$

$$[\varepsilon] = [A][U] \qquad (11b)$$

In general the stress - strain relationship may be considered of the form.

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \qquad (12a)$$

$$[\sigma] = [B][\varepsilon] \qquad (12b)$$

For an isotropic material in the state of plane stress the stress - strain relationship is of the form.

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \qquad (12c)$$

From eq.(11) and eq.(12) the element stresses in terms of the element nodal point displacements may be obtained quite simply by matrix multiplcation.

$$[\sigma] = [B][A][U] \qquad (13)$$

To finally obtain the relationship between applied external nodal point forces and nodal point displacements it is necessary to equate the external work done and the internal work done by the applied external forces and stresses during displacement.

If we now impose an arbitary virtual nodal point displacement to the element, the work done is given by the summation of the product of each nodal point external force and its corresponding displacement.

$$\text{External Work Done} = \sum \text{Nodal Point External Force} \times \text{Nodal Point Displacement.}$$

In matrix notation

$$\left[W\right]_{EXT} = \left[U^*\right]^T \left[P\right] \tag{14}$$

where

$\left[W\right]_{EXT}$ Represents the external work done by the applied nodal point external forces.

$\left[U^*\right]$ Is a column matrix of virtual nodal point displacements $\begin{bmatrix} U_1^* \\ \vdots \\ U_n^* \end{bmatrix}$

$\left[U^*\right]^T$ Is the transpose $\left[U_1^* - - - - - U_n^*\right]$ of the virtual nodal point displacement matrix $\left[U^*\right]$

$\left[P\right]$ Is a column matrix of nodal point external forces $\begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix}$

Now the internal work done per unit volume by the stresses is

$$\left[W\right]_{INT} = \left[\varepsilon^*\right]^T \left[\sigma\right] \tag{15}$$

or

$$\left[W\right]_{INT} = \left[U^*\right]^T \left[A\right]^T \left[\sigma\right] \tag{16}$$

where

$[W]_{INT}$ Represents the internal work done by the stresses

$[\varepsilon^*]$ Is a column matrix of virtual strains $\begin{bmatrix}\varepsilon^*_1\\ \vdots \\ \varepsilon^*_n\end{bmatrix}$

$[\varepsilon^*]^T$ Is the transpose $[\varepsilon^*_1 - - - - \varepsilon^*_n]$ of the virtual strain matrix $[\varepsilon^*]$

$[A]^T$ Is the transpose of matrix $[A]$ from eq.(11)

$[\sigma]$ Is a column matrix of internal stresses. $\begin{bmatrix}\sigma_1\\ \vdots \\ \sigma_n\end{bmatrix}$

Equating the external and internal work done integrating over the volume of the element.

$$[U^*]^T[P] = [U^*]^T \int [A]^T [\sigma] \text{dvol.} \tag{17}$$

$$\therefore [P] = \int [A]^T [\sigma] \text{ dvol.} \tag{18}$$

from eq.(13) $[\sigma] = [B][A][U]$

$$\therefore [P] = \int [A]^T [B][A] \text{ dvol.}[U] \tag{19}$$

where $\int [A]^T[B][A]$ dvol. is the relationship between nodal point external forces and nodal point displacements and is known as the element stiffness matrix for the (e) element.

$$\text{ie} \quad [K^{(e)}] = \int [A]^T [B][A] \text{ dvol.} \tag{20}$$

For a triangular element of constant thickness t the element stiffness matrix becomes

$$[K^{(e)}] = [A]^T [B][A] . t . A_{ijk} \tag{21}$$

and will relate the 6 nodal point forces (3 horizontal, 3 vertical) to the 6 nodal point displacements.

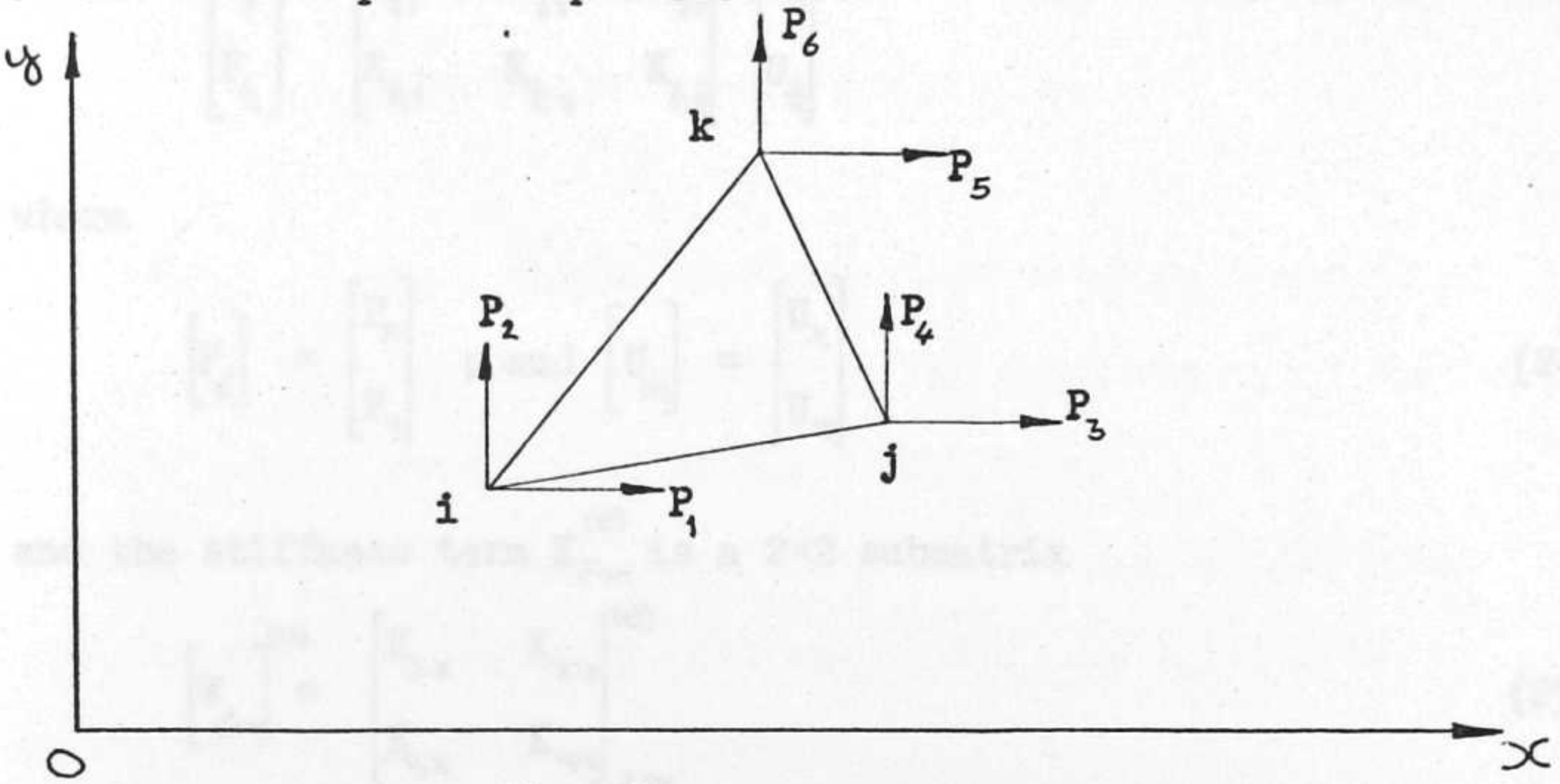


Fig. 2.

The resulting element stiffness matrix $K^{(e)}$ eq.(22), will be fully populated, and will be symmetric; therefore $K^{(e)}$ will equal $K^{(e)^T}$

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{66} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix}^{(e)} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{bmatrix} \qquad (22)$$

The leading diagonal of the stiffness matrix containing the terms $K_{11}, K_{22}, K_{33}, K_{44}, K_{55}, K_{66}$ are known as direct stiffnesses since they relate some force ℓ to its own displacement ℓ . The other stiffness terms are known as cross stiffnesses, and relate some force ℓ to some other displacement m .

It will be convenient to consider eq.(22) in the form

$$\begin{bmatrix} P_i \\ P_j \\ P_k \end{bmatrix} = \begin{bmatrix} K_{ii} & K_{ij} & K_{ik} \\ K_{ji} & K_{jj} & K_{jk} \\ K_{ki} & K_{kj} & K_{kk} \end{bmatrix}^{(e)} \begin{bmatrix} U_i \\ U_j \\ U_k \end{bmatrix} \tag{23}$$

where

$$\begin{bmatrix} P_\ell \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \end{bmatrix} \quad ; \text{ and } \begin{bmatrix} U_m \end{bmatrix} = \begin{bmatrix} U_x \\ U_y \end{bmatrix} \tag{24}$$

and the stiffness term $K^{(e)}_{\ell m}$ is a 2×2 submatrix

$$\begin{bmatrix} K_{\ell m} \end{bmatrix}^{(e)} = \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}^{(e)}_{\ell m} \tag{25}$$

The subscripts x and y refer to the normal X and Y direction respectively.

The significance of the terms of eq.(25) are

$K^{(e)}_{\ell m}$ is a 2×2 submatrix representing the force on the (e) element at nodal point ℓ produced by a unit displacement (the virtual displacement) at nodal.point m .

K_{xx} is a single stiffness term relating the ℓ force in the X direction to the displacement of the m nodal point in the X direction.

K_{xy} is a single stiffness term relating the ℓ force in the X direction to the displacement of the m nodal point in the Y direction.

K_{yx} is a single stiffness term relating the ℓ force in the Y direction to the displacement of the m nodal point in the X direction.

K_{yy} is a single stiffness term relating the ℓ force in the Y direction to the displacement of the m nodal point in the Y direction.

Note that for the case where $m = \ell$ then we are dealing with a direct stiffness relationship, and thus K_{xx} and K_{yy} will be on the leading diagonal of eq.(22).

1.2 Assembly of Elements.

The element stiffness matrix as given by eq.(22) is only for a discrete element. When the elements are connected together to form a complete structure, the element stiffness properties must be assembled into an assembled stiffness matrix for the complete structure.

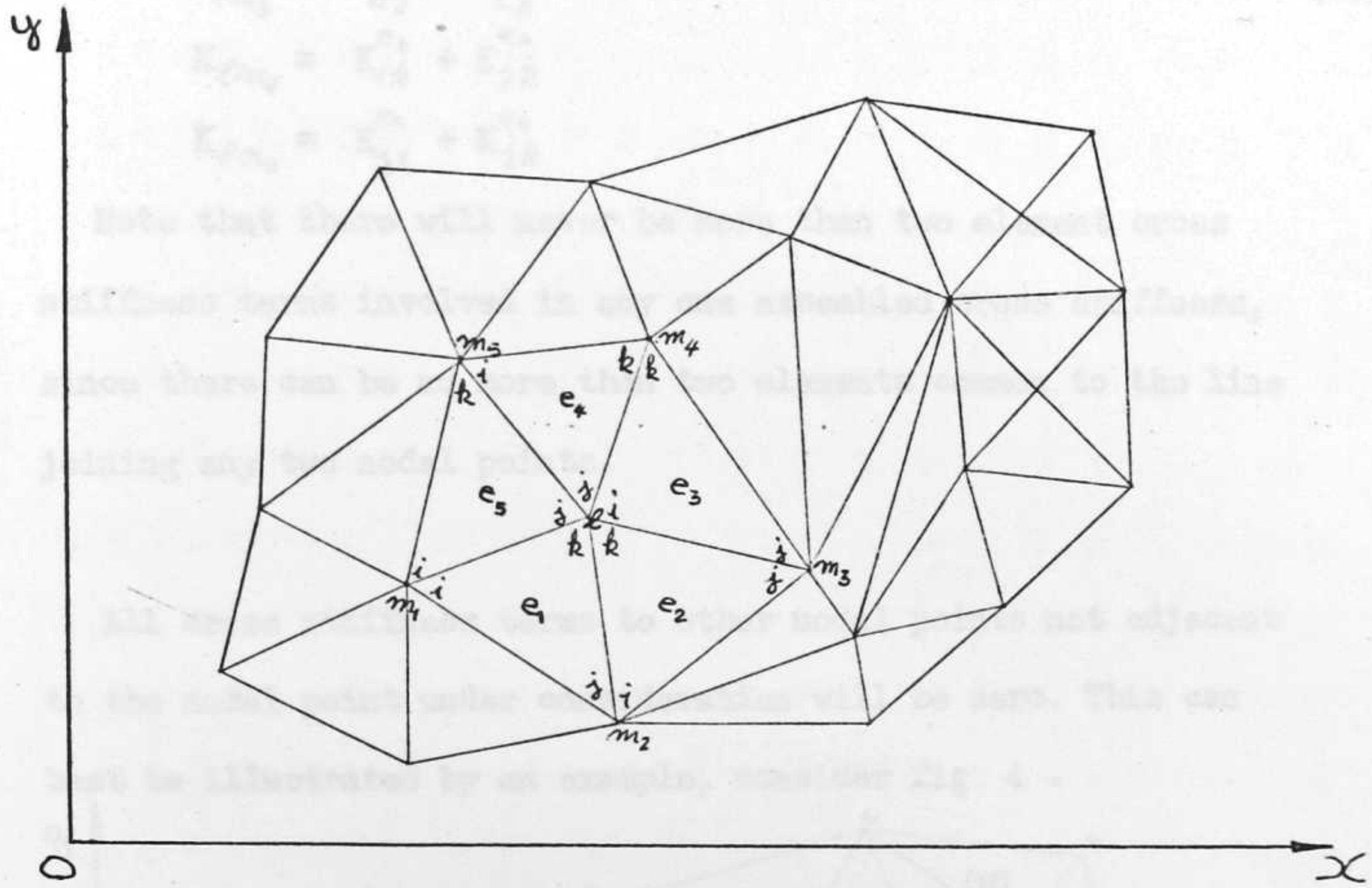


Fig. 3 arbitary structure composed of triangular elements.

Fig. 3 shows a structure of quite arbitary shape. If we consider, say nodal point ℓ which is surrounded by 5 adjacent nodal points m_1 to m_5, and 5 adjacent elements e_1 to e_5. The direct stiffness of nodal point ℓ is obtained by summing up the direct stiffness terms of that vertex of each element which is connected to nodal point ℓ.

$$\text{ie} \quad K_{\ell\ell} = K_{kk}^{(e_1)} + K_{kk}^{(e_2)} + K_{ii}^{(e_3)} + K_{jj}^{(e_4)} + K_{jj}^{(e_5)} \qquad (26)$$

The 5 cross stiffness terms of nodal point ℓ are obtained by summing the element cross stiffness terms to those nodal points adjacent to nodal point ℓ.

ie

$$
\begin{aligned}
K_{\ell m_1} &= K^{e_1}_{ki} + K^{e_5}_{ji} \\
K_{\ell m_2} &= K^{e_1}_{kj} + K^{e_2}_{ki} \\
K_{\ell m_3} &= K^{e_2}_{kj} + K^{e_3}_{ij} \\
K_{\ell m_4} &= K^{e_3}_{ik} + K^{e_4}_{jk} \\
K_{\ell m_5} &= K^{e_4}_{ji} + K^{e_5}_{jk}
\end{aligned}
\qquad (27)
$$

Note that there will never be more than two element cross stiffness terms involved in any one assembled cross stiffness, since there can be no more than two elements common to the line joining any two nodal points.

All cross stiffness terms to other nodal points not adjacent to the nodal point under consideration will be zero. This can best be illustrated by an example, consider fig 4 .

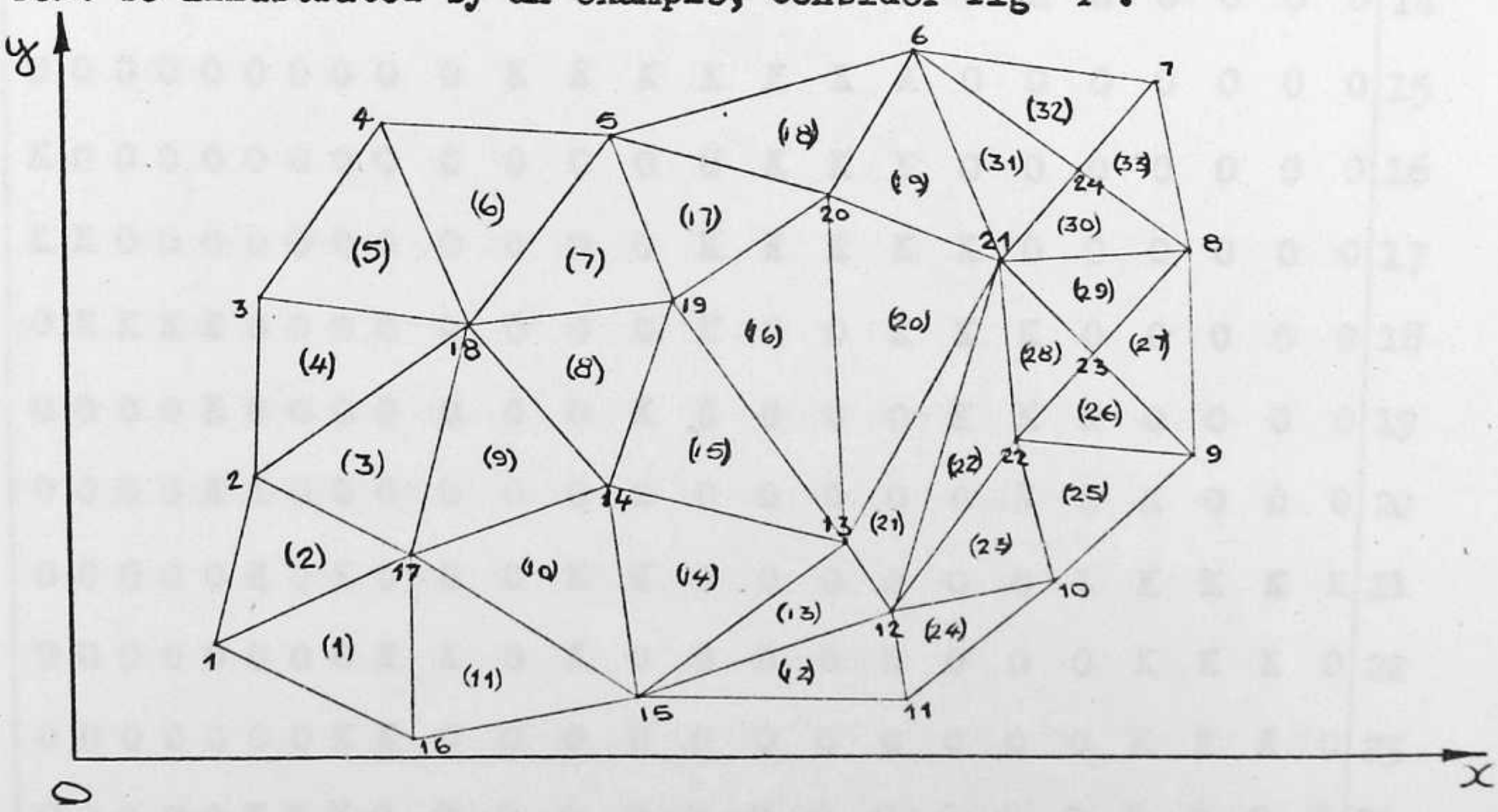


Fig. 4.

The assembled stiffness matrix K for this complete structure will be as indicated in fig 5 .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
K	K	0	0	0	0	0	0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	I
K	K	K	0	0	0	0	0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	2
0	K	K	K	0	0	0	0	0	0	0	0	0	0	0	0	K	0	0	0	0	0	0	0	3
0	0	K	K	K	0	0	0	0	0	0	0	0	0	0	0	K	0	0	0	0	0	0	0	4
0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	5
0	0	0	0	K	K	K	K	0	0	0	0	0	0	0	0	K	K	K	0	0	0	0	K	6
0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	K	K	0	0	0	0	K	7
0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	K	0	K	K	0	K	8
0	0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	K	K	0	0	K	9
0	0	0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	K	0	0	K	10
0	0	0	0	0	0	0	0	0	K	K	K	0	0	K	0	0	0	0	0	0	0	0	0	11
0	0	0	0	0	0	0	0	0	0	K	K	K	K	0	K	0	0	0	0	0	K	K	0	12
0	0	0	0	0	0	0	0	0	0	0	0	K	K	K	K	0	0	0	K	K	K	0	0	13
0	0	0	0	0	0	0	0	0	0	0	0	0	K	K	K	0	K	K	K	0	0	0	0	14
0	0	0	0	0	0	0	0	0	0	0	K	K	K	K	K	K	K	0	0	0	0	0	0	15
K	0	0	0	0	0	0	0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	16
K	K	0	0	0	0	0	0	0	0	0	0	0	K	K	K	K	K	0	0	0	0	0	0	17
0	K	K	K	K	0	0	0	0	0	0	0	0	K	0	0	K	K	K	0	0	0	0	0	18
0	0	0	0	K	0	0	0	0	0	0	0	K	K	0	0	0	K	K	K	0	0	0	0	19
0	0	0	0	K	K	0	0	0	0	0	0	K	0	0	0	0	0	K	K	K	0	0	0	20
0	0	0	0	0	K	0	K	0	0	0	K	K	0	0	0	0	0	0	K	K	K	K	K	21
0	0	0	0	0	0	0	0	K	K	0	K	0	0	0	0	0	0	0	K	K	K	K	0	22
0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	0	0	0	K	K	K	0	23
0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	0	0	0	K	0	0	K	24

Fig 5.

The matrix fig. 5 is symmetric ie. $K_{ij} = K_{ji}^{T}$ and contains a very large number of zero terms.

If we consider, say nodal point number 14, which from fig. 4 is seen to have adjacent nodal points 13, 15, 17, 18, 19. Refering to fig. 5 the stiffness terms associated with nodal point 14 are to be found in row 14.

The only non-zero stiffness terms for nodal point 14 are

(a) The direct stiffness K_{1414} where

$$K_{1414} = K_{1414}^{(8)} + K_{1414}^{(9)} + K_{1414}^{(10)} + K_{1414}^{(14)} + K_{1414}^{(15)}$$

(b) The cross stiffnesses $K_{1413}, K_{1415}, K_{1417}, K_{1418}, K_{1419}$ where

$$K_{1413} = K_{1413}^{(14)} + K_{1413}^{(15)}$$

$$K_{1415} = K_{1415}^{(10)} + K_{1415}^{(14)}$$

$$K_{1417} = K_{1417}^{(9)} + K_{1417}^{(10)}$$

$$K_{1418} = K_{1418}^{(8)} + K_{1418}^{(9)}$$

$$K_{1419} = K_{1419}^{(8)} + K_{1419}^{(15)}$$

All other cross stiffness terms to nodal points 1 tol2, 16, and 20 to 24 are zero since these nodal points are not adjacent to nodal point number 14.

Note that each of the terms in fig. 5 and in eq.(26) and eq.(27) will be a 2×2 submatrix as indicated by eq.(25). Thus for the structure shown in fig. 4 with 24 nodal points, the actual size of the assembled stiffness matrix is 48×48, and would require computer storage capacity for 2304 terms. It is therefore apparent that with structures requiring hundreds of nodal points, the size of the problem that can be analysed is severely limited to the amount of computer storage capacity available.

Finally the relationship between nodal point external forces and nodal point displacements may be written

$$[P] = [K][U] \tag{28}$$

where

P is a column matrix of length 2n and refers to the nodal point external forces.

K is an array of size 2n×2n and refers to the assembled stiffness matrix for the complete structure. The matrix will be symmetric and is likely to contain a large number of zero terms. Computer analysis can be arranged to store only the non-zero terms, but even so, with larger structures the computer storage may not be adequate.

U is a column matrix of length 2n and refers to the nodal point displacements.

n is the total number of nodal points in the complete structure.

The assembled stiffness matrix K as given by eq.(28) will be non-singular, and any arbitary amount of body movement will satisfy eq.(28). To obtain a unique solution, prescribed restraints must be introduced at those nodal points where the displacements

are known. The method of introducing the restraints will be described on page 69.

1.3 Substructure Analysis(ref. 2,3, and 4).

If we consider the two-dimensional idealized structure of arbitary shape and loading as shown in fig. 6 .

The relationship between nodal point external forces and nodal point displacements is given by eq.(28) providing that sufficient restraints are introduced to render matrix K singular.

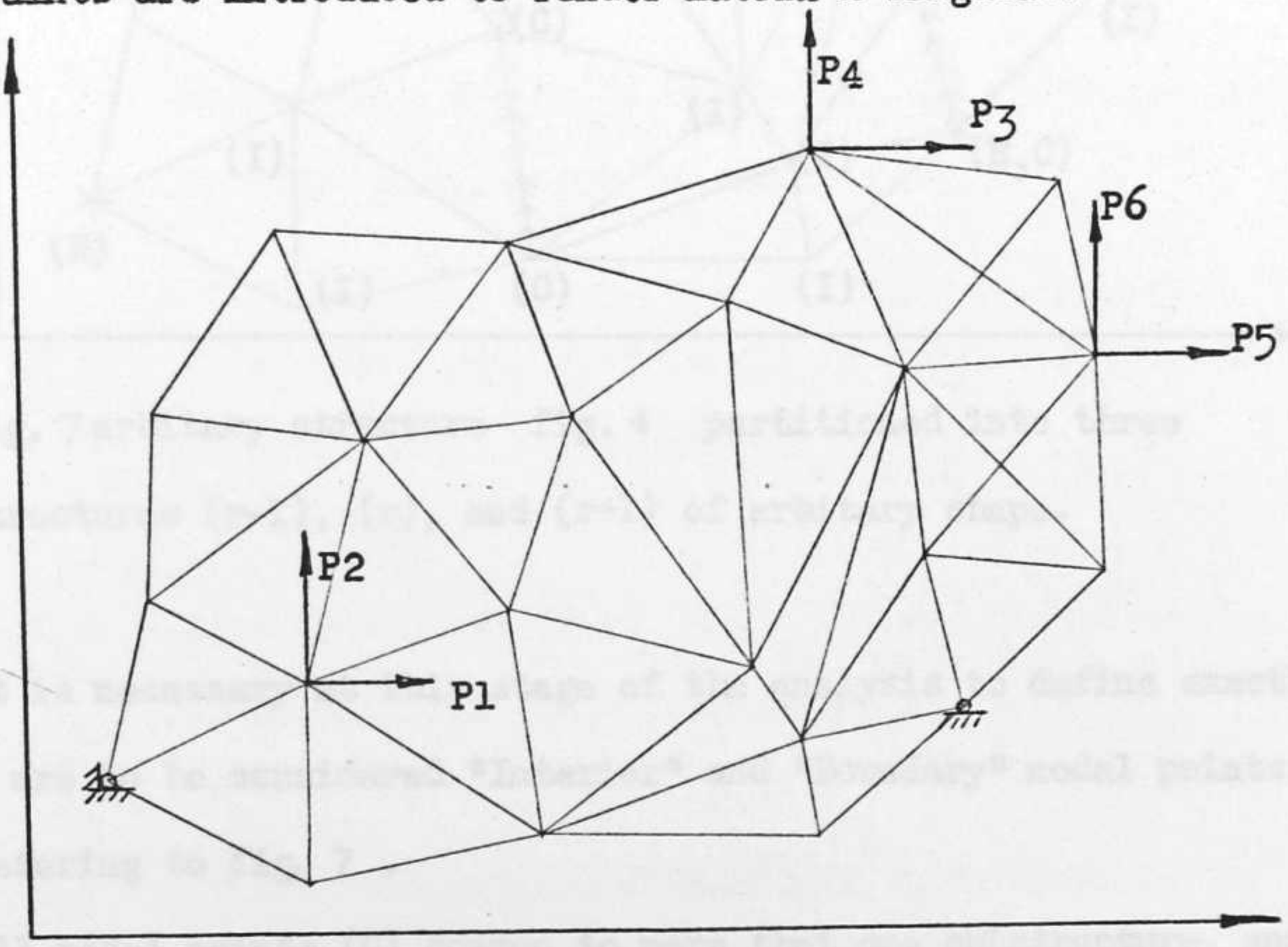


Fig. 6 arbitary structure under the action of applied external non-zero forces P1 to P6.

The complete structure of fig. 6 may be divided into a number of substructures of arbitary shape as shown in fig. 7. Where the substructure boundarys are indicated by the hatched lines. ⤑⤑⤑ ✕

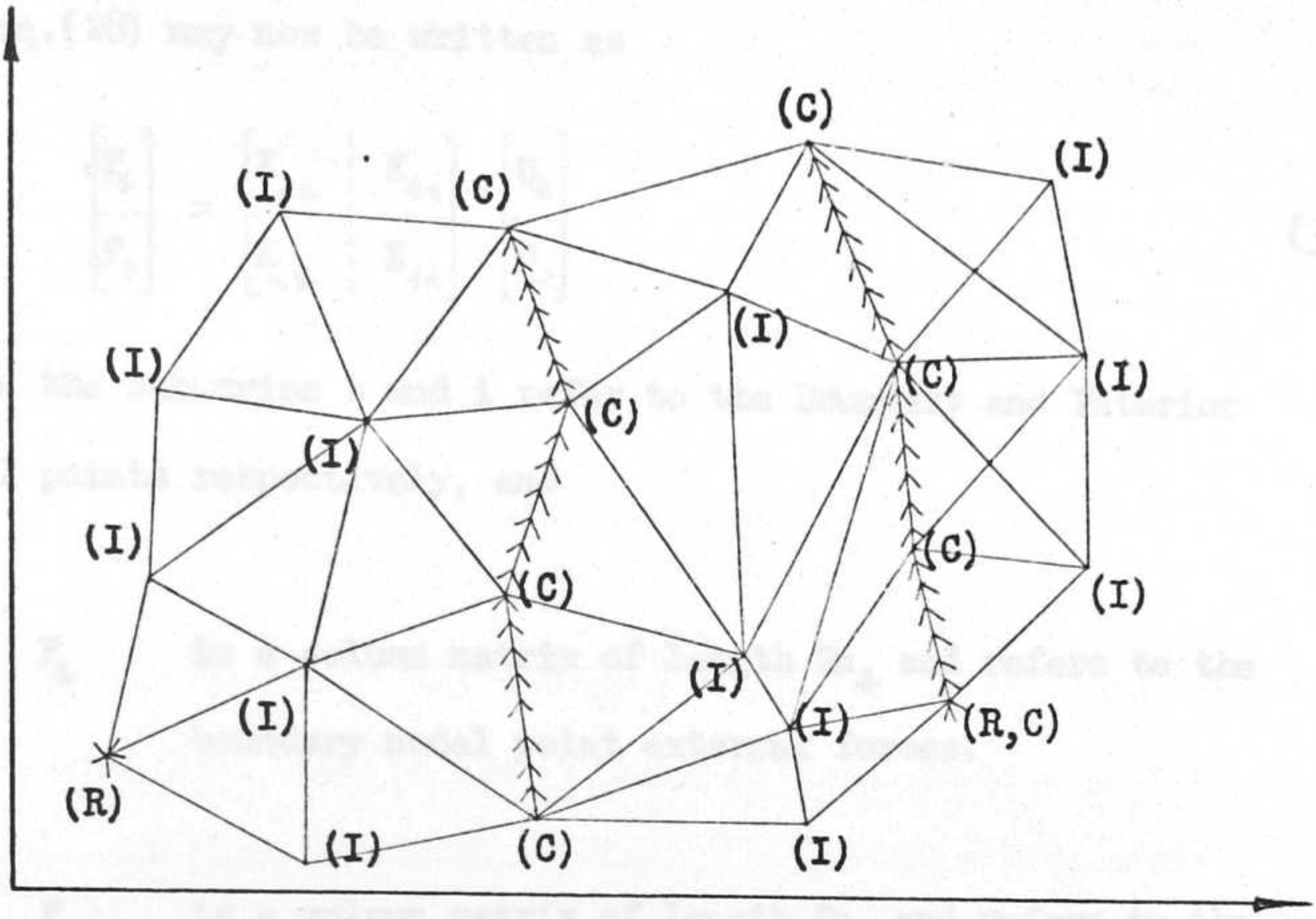


Fig. 7 arbitary structure fig. 4 partitioned into three substructures (r-1), (r), and (r+1) of arbitary shape.

It is necessary at this stage of the analysis to define exactly what are to be considered "Interior" and "Boundary" nodal points.

Refering to fig. 7 .

All nodal points (C) common to more that one substructure, and all restrained nodal points (R) are to be considered as BOUNDARY nodal points. (a nodal point common to only one substructure may be considered as a boundary nodal point if so desired).

All other nodal points are to be considered as INTERIOR nodal points. ie. points(I).

From matrix algebra (ref.2) we may conveniently partition the stiffness matrix K as follows

$$K = \left[\begin{array}{c|c} K_{bb} & K_{bi} \\ \hline K_{ib} & K_{ii} \end{array}\right] \qquad (29)$$

and eq.(28) may now be written as

$$\begin{bmatrix} P_b \\ \hline P_i \end{bmatrix} = \begin{bmatrix} K_{bb} & K_{bi} \\ \hline K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} U_b \\ \hline U_i \end{bmatrix} \tag{30}$$

where the subscrips b and i refer to the Boundary and Interior nodal points respectively, and

P_b is a column matrix of length $2n_b$ and refers to the boundary nodal point external forces.

P_i is a column matrix of length $2n_i$ and refers to the interior nodal point external forces.

U_b is a column matrix of length $2n_b$ and refers to the boundary nodal point displacements.

U_i is a column matrix of length $2n_i$ and refers to the interior nodal point displacements.

K_{bb} is a matrix of size $2n_b \times 2n_b$ and refers to the stiffness relationship between P_b and U_b.

K_{bi} is a matrix of size $2n_b \times 2n_i$ and refers to the stiffness relationship between P_b and U_i.

K_{ib} is a matrix of size $2n_i \times 2n_b$ and refers to the stiffness relationship between P_i and U_b.

K_{ii} is a matrix of size $2n_i \times 2n_i$ and refers to the stiffness relationship between P_i and U_i.

n_b is the total number of boundary nodal points.

n_i is the total number of interior nodal points.

If we now return to the example of fig. 4 the stiffness matrix K when partitioned will now be arranged as shown in fig. 8 , to suit the boundary indicated by fig. 7 . The upper L.H. array represents K_{bb}, the lower R.H. array represents K_{ii}, whilst the upper R.H. and the lower L.H. arrays represent K_{bi} and K_{ib} respectively. The complete matrix is symmetric, and therefore $K_{bi} = K_{ib}^T$.

Now from eq(30)

$$P_b = K_{bb} U_b + K_{bi} U_i \tag{31}$$

$$P_i = K_{ib} U_b + K_{ii} U_i \tag{32}$$

from eq.(32)

$$U_i = K_{ii}^{-1} P_i - K_{ii}^{-1} K_{ib} U_b \tag{33}$$

substituting eq.(33) into eq.(31) gives

$$P_b = K_{bi} K_{ii}^{-1} P_i + (K_{bb} - K_{bi} K_{ii}^{-1} K_{ib}) U_b \tag{34}$$

If we now consider the total displacement U to be the sum of two matrices such that

$$U = U^{(\alpha)} + U^{(\beta)} \tag{35}$$

1	5	6	10	14	15	19	21	22	2	3	4	7	8	9	11	12	13	16	17	18	20	23	24	
K	0	0	0	0	0	0	0	0	K	0	0	0	0	0	0	0	0	K	K	0	0	0	0	1
0	K	K	0	0	0	K	0	0	0	0	K	0	0	0	0	0	0	0	0	K	K	0	0	5
0	K	K	0	0	0	0	K	0	0	0	0	K	0	0	0	0	0	0	0	0	K	0	K	6
0	0	0	K	0	0	0	0	K	0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	10
0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	0	0	K	0	K	K	0	0	0	14
0	0	0	0	K	K	0	0	0	0	0	0	0	0	0	0	K	K	K	K	0	0	0	0	15
0	K	0	0	K	0	K	0	0	0	0	0	0	0	0	0	0	0	K	0	0	K	K	0	19
0	0	K	0	0	0	0	K	K	0	0	0	0	K	0	0	K	K	0	0	0	K	K	K	21
0	0	0	K	0	0	0	K	K	0	0	0	0	0	K	0	K	0	0	0	0	0	K	0	22
K	0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	0	K	K	0	0	0	2
0	0	0	0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	0	0	K	0	0	0	3
0	K	0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	0	K	0	0	0	4
0	0	K	0	0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	0	0	K	7
0	0	0	0	0	0	0	K	0	0	0	0	K	K	K	0	0	0	0	0	0	0	K	K	8
0	0	0	K	0	0	0	0	K	0	0	0	0	K	K	0	0	0	0	0	0	0	K	0	9
0	0	0	K	0	K	0	0	0	0	0	0	0	0	0	K	K	0	0	0	0	0	0	0	11
0	0	0	K	0	K	0	K	K	0	0	0	0	0	0	K	K	K	0	0	0	0	0	0	12
0	0	0	0	K	K	K	K	0	0	0	0	0	0	0	0	K	K	0	0	0	K	0	0	13
K	0	0	0	0	K	0	0	0	0	0	0	0	0	0	0	0	0	K	K	0	0	0	0	16
K	0	0	0	0	K	K	0	0	K	0	0	0	0	0	0	0	0	K	K	K	0	0	0	17
0	K	0	0	K	0	K	0	0	K	K	K	0	0	0	0	0	0	K	K	0	0	0	0	18
0	K	K	0	0	0	K	K	0	0	0	0	0	0	0	0	0	0	K	0	0	0	K	0	20
0	0	0	0	0	0	0	K	K	0	0	0	0	K	K	0	0	0	0	0	0	0	K	0	23
0	0	K	0	0	0	0	K	0	0	0	K	K	0	0	0	0	0	0	0	0	0	0	K	24

Fig. 8 partitioned assembled stiffness matrix K for the structure shown in fig. 4 . Partitioned to suit the boundary indicated by fig. 7 .

where

$U^{(\alpha)}$ is a column matrix of length 2n and refers to the nodal point displacements due to interior nodal point external forces P_i when the boundary nodal points are fixed. ie. $U_b = 0$.

$U^{(\beta)}$ is a column matrix of length 2n and refers to the correction to the displacement $U^{(\alpha)}$ to allow for boundary nodal point displacements U_b (relaxation of the boundary) with the interior nodal point external forces equal to zero. ie. $P_i = 0$.

Eq.(35) may be partitioned as follows

$$U = \begin{bmatrix} U_b \\ \hline U_i \end{bmatrix} = \begin{bmatrix} U_b^{(\alpha)} \\ \hline U_i^{(\alpha)} \end{bmatrix}_{U_b=0} + \begin{bmatrix} U_b^{(\beta)} \\ \hline U_i^{(\beta)} \end{bmatrix}_{P_i=0} \tag{36}$$

$$\therefore \quad \begin{bmatrix} U_b \\ \hline U_i \end{bmatrix} = \begin{bmatrix} 0 \\ \hline U_i \end{bmatrix}^{(\alpha)} + \begin{bmatrix} U_b \\ \hline U_i \end{bmatrix}^{(\beta)}_{P_i=0} \tag{37}$$

In a simmilar manner the external forces P may be separated thus

$$P = P^{(\alpha)} + P^{(\beta)} \tag{38}$$

where

$P^{(\alpha)}$ is a column matrix of length 2n and refers to the nodal point external forces when the boundary nodal points are fixed. ie. $U_b = 0$.

$P^{(\beta)}$ is a column matrix of length 2n and refers to the

external forces acting at the boundary nodal points; the boundary having been relaxed and the interior nodal point external forces are zero. ie. $P_i = 0$.

Eq.(38) may be partitioned as follows

$$P = \begin{bmatrix} P_b \\ P_i \end{bmatrix} = \begin{bmatrix} P_b^{(\alpha)} \\ P_i^{(\alpha)} \end{bmatrix}_{U_b=0} + \begin{bmatrix} P_b^{(\beta)} \\ P_i^{(\beta)} \end{bmatrix}_{P_i=0} \qquad (39)$$

$$\begin{bmatrix} P_b \\ P_i \end{bmatrix} = \begin{bmatrix} P_b \\ P_i \end{bmatrix}^{(\alpha)}_{U_b=0} + \begin{bmatrix} P_b \\ 0 \end{bmatrix}^{(\beta)} \qquad (40)$$

To obtain the terms of eq.(37) and eq.(40) we must consider; first the boundary fixed, then the boundary relaxed condition in turn.

With the boundary fixed; (α) condition.

$$U_b^{(\alpha)} = 0 \; ; \quad \text{and} \quad P_i^{(\alpha)} = P_i$$

and therefore eq(33) and eq.(34) reduce to

$$U_i^{(\alpha)} = K_{ii}^{-1} P_i \qquad (41)$$

and

$$P_b^{(\alpha)} = K_{bi} K_{ii}^{-1} P_i \qquad (42)$$

With the boundary relaxed; (β) condition.

$$U_b^{(\beta)} = U_b \; ; \quad \text{and} \quad P_i^{(\beta)} = 0$$

now eq.(33) and eq.(34) reduce to

$$U_i^{(\beta)} = - K_{ii}^{-1} K_{ib} U_b \qquad (43)$$

and

$$P_b^{(\beta)} = (K_{bb} - K_{bi} K_{ii}^{-1} K_{ib})U_b \tag{44}$$

eq.(37) may now be written

$$\left[\begin{array}{c} U_b \\ \hline U_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline K_{ii}^{-1} P_i \end{array}\right]^{(\alpha)} + \left[\begin{array}{c} U_b \\ \hline -K_{ii}^{-1} K_{ib} U_b \end{array}\right]^{(\beta)} \tag{45}$$

and eq.(40) becomes

$$\left[\begin{array}{c} P_b \\ \hline P_i \end{array}\right] = \left[\begin{array}{c} K_{bi} K_{ii}^{-1} P_i \\ \hline P_i \end{array}\right]^{(\alpha)} + \left[\begin{array}{c} (K_{bb} - K_{bi} K_{ii}^{-1} K_{ib})U_b \\ \hline 0 \end{array}\right]^{(\beta)} \tag{46}$$

To obtain a solution from eq.(45) and eq.(46), consider the top line of eq.(46).

$$P_b = K_{bi} K_{ii}^{-1} P_i^{(\alpha)} + (K_{bb} - K_{bi} K_{ii}^{-1} K_{ib})U_b^{(\beta)} \tag{47}$$

with the boundary fixed.

$$P_b^{(\alpha)} = K_{bi} K_{ii}^{-1} P_i^{(\alpha)} = R_b \tag{48}$$

with the boundary relaxed

$$P_b^{(\beta)} = P_b - K_{bi} K_{ii}^{-1} P_i^{(\alpha)} = P_b - K_{bi} K_{ii}^{-1} P_i \tag{49}$$

$$\therefore P_b^{(\beta)} = P_b - P_b^{(\alpha)} = P_b - R_b$$

rearranging eq.(47), and putting

$$K_b = (K_{bb} - K_{bi} K_{ii}^{-1} K_{ib}) \tag{50}$$

we have

$$P_b - R_b = K_b U_b \tag{51}$$

and we can write

$$F_\ell = K_\ell U_\ell \tag{52}$$

where

$F_\ell = P_\ell - R_\ell$ is a column matrix of length $2n_\ell$ and refers to the resultant boundary force, comprising of the boundary nodal point external forces P_ℓ minus the boundary reaction R_ℓ .

$R_\ell = K_{\ell i} K_{ii}^{-1} P_i$ is a column matrix of length $2n_\ell$ and refers to the boundary reaction necessary to maintain $U_\ell = 0$.

$K_\ell = (K_{\ell\ell} - K_{\ell i} K_{ii}^{-1} K_{i\ell})$ is a matrix of size $2n_\ell \times 2n_\ell$ and refers to the assembled boundary stiffness matrix for the complete structure. The matrix will be symmetric and is likely to contain a large number of zero terms.

n_ℓ and n_i are as defined on page 34.

The boundary nodal point displacements with the boundary relaxed, may now be obtained from eq.(52) by inverting K_ℓ .

$$U_\ell = K_\ell^{-1} F_\ell \tag{53}$$

The significance of eq(53) is that by relaxing all the boundary nodal points simultaneously with the exception of certain specified restrained nodal points to render K_ℓ singular, the resultant

boundary forces F_b will cause the boundary nodal points to be displaced by an amount which will satisfy equlibrium at each boundary nodal point.

To obtain the interior nodal point displacements we now consider the bottom line of eq.(45).

$$U_i = K_{ii}^{-1} P_i^{(\alpha)} - K_{ii}^{-1} K_{ib} U_b^{(\beta)} \tag{54}$$

substituting for U_b we may write

$$U_i = \left(K_{ii}^{-1} P_i \right)_{U_b=0} - K_{ii}^{-1} K_{ib} \left(K_b^{-1} . F_b \right) \tag{55a}$$

$$U_i = \left(K_{ii}^{-1} P_i \right)_{U_b=0} - K_{ii}^{-1} K_{ib} U_b \tag{55b}$$

where

$K_{ii}^{-1} P_i$ is a column matrix of length $2n_i$ and refers to the interior nodal point displacements due to interior nodal point external forces, with the boundary fixed.

K_{ii}^{-1} is a symmetric matrix of size $2n_i \times 2n_i$ and refers to the inverse of K_{ii}, and represents the flexibility relationship between interior nodal point displacements and interior nodal point external forces.

$K_{ii}^{-1} K_{ib} U_b$ is a column matrix of length $2n_i$ and refers to the correction to the interior nodal point displacements brought about by the boundary nodal

point displacements U_ℓ when the boundary is relaxed.

$K_{ii}^{-1} K_{i\ell}$ is a matrix of size $2n_i \times 2n_i$ and refers to the relationship between interior and boundary nodal point displacements.

With the application of eq.(52) and eq.(54), all the required displacements can be obtained. The element stresses may finally be obtained by a direct application of eq.(13).

By inspection of eq.(50) it will be appreciated that K_ℓ will be of much smaller size than K . If we refer again to the structure shown in fig. 4 , K_ℓ for this structure is given by fig. 9 , with boundaries as indicated by fig. 7 . Each term in fig. 9 will be a 2×2 submatrix as indicated by eq.(25). Therefore K_ℓ would require computer storage capacity for only 324 terms as against 2304 terms for K .

1	5	6	10	14	15	19	21	22	
K	K	0	0	K	K	K	0	0	1
K	K	K	K	K	K	K	K	K	5
0	K	K	K	K	K	K	K	K	6
0	K	K	K	K	K	K	K	K	10
K	K	K	K	K	K	K	K	K	14
K	K	K	K	K	K	K	K	K	15
K	K	K	K	K	K	K	K	K	19
0	K	K	K	K	K	K	K	K	21
0	K	K	K	K	K	K	K	K	22

Fig 9 assembled boundary stiffness matrix K_ℓ for the structure

shown in fig. 4 with boundaries as indicated by fig. 7. The matrix is symmetric and in general will contain a large number of zero terms.

If the foregoing analysis is applied directly, we would still require to store at least the non-zero terms of the K matrix, and therefore the basic problem of computer storage still exists. However if instead , we apply the analysis to the small substructures and obtain an assembled substructure boundary stiffness matrix for each substructure then by assembling these substructure boundary stiffness matrix in a simmilar manner as when assembling the individual triangular elements, we may then obtain the assembled boundary stiffness matrix K_b without the need to store the very large assembled stiffness matrix K .

If we now consider say the (r) substructure. The relationship between the substructure applied external forces and nodal point displacements is given by eq.(56).

$$P^{(r)} = K^{(r)} U^{(r)} \tag{56}$$

Where the terms of eq.(56) are identical to those of eq.(28) except that eq.(56) refers only to the (r) substructures and $K^{(r)}$ will contain only those terms associated with the elements within the bounds of the (r) substructure.

For example consider the (r) substructure of fig. 7 , with nodal point numbering as indicated by fig 4. Then the build up of $K^{(r)}$ is given by fig.10.

5	6	10	11	12	13	14	15	19	20	21	22	
K	K	O	O	O	O	O	O	K	K	O	O	5
K	K	O	O	O	O	O	O	O	K	K	O	6
O	O	K	K	K	O	O	O	O	O	O	K	10
O	O	K	K	K	O	O	K	O	O	O	O	11
O	O	K	K	K	K	O	K	O	O	K	K	12
O	O	O	O	K	K	K	K	K	K	K	O	13
O	O	O	O	O	K	K	K	K	O	O	O	14
O	O	O	K	K	K	K	K	O	O	O	O	15
K	O	O	O	O	K	K	O	K	K	O	O	19
K	K	O	O	O	K	O	O	K	K	K	O	20
O	K	O	O	K	K	O	O	O	K	K	K	21
O	O	K	O	K	O	O	O	O	O	K	K	22

Fig.10 assembled substructure stiffness matrix $K^{(r)}$ for the (r) substructure of fig. 7 . The matrix is symmetric and will contain a large number of zero terms. Each term will be a 2×2 submatrix as indicated by eq.(25).

Considering the (r) substructure as a free body, we may partition $K^{(r)}$ into

$$K^{(r)} = \left[\begin{array}{c|c} K_{bb}^{(r)} & K_{bi}^{(r)} \\ \hline K_{ib}^{(r)} & K_{ii}^{(r)} \end{array}\right] \tag{57}$$

and eq.(56) becomes

$$\left[\begin{array}{c} P_b^{(r)} \\ \hline P_i^{(r)} \end{array}\right] = \left[\begin{array}{c|c} K_{bb}^{(r)} & K_{bi}^{(r)} \\ \hline K_{ib}^{(r)} & K_{ii}^{(r)} \end{array}\right] \left[\begin{array}{c} U_b^{(r)} \\ \hline U_i^{(r)} \end{array}\right] \tag{58}$$

Where the terms of eq.(58) are identical to those of eq.(30) except that eq.(58) refers to the nodal points and elements associated only with the (r) substructure.

With $K^{(r)}$ partitioned according to eq.(57) the rearranged order of fig. 10 is indicated by fig. 11 .

	5	6	10	14	15	19	21	22	11	12	13	20	
	K	K	0	0	0	K	0	0	0	0	0	K	5
	K	K	0	0	0	0	K	0	0	0	0	K	6
	0	0	K	0	0	0	0	K	K	K	0	0	10
	0	0	0	K	K	K	0	0	0	0	K	0	14
	0	0	0	K	K	0	0	0	K	K	K	0	15
	K	0	0	K	0	K	0	0	0	0	K	K	19
	0	K	0	0	0	0	K	K	0	K	K	K	21
	0	0	K	0	0	0	K	K	0	K	0	0	22
	0	0	K	0	K	0	0	0	K	K	0	0	11
	0	0	K	0	K	0	K	K	K	K	K	0	12
	0	0	0	K	K	K	K	0	0	K	K	K	13
	K	K	0	0	0	K	K	0	0	0	K	K	20

Fig. 11 substructure assembled stiffness matrix $K^{(r)}$ for the (r) sustructure of fig. 7 , rearranged to suit partitioning according to eq.(57). The matrix is symmetric and each term is a 2×2 submatrix as given by eq.(25).

By expanding eq.(58) we obtain

$$U_i^{(r)} = (K_{ii}^{(r)})^{-1} P_i^{(r)} - (K_{ii}^{(r)})^{-1} K_{ib}^{(r)} U_b^{(r)} \qquad (59)$$

and

$$P_b^{(r)} = K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} P_i^{(r)} + (K_{bb}^{(r)} - K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}) U_b \qquad (60)$$

And by the same reasoning as given by the previous theory, with the boundary fixed; ie. $U_b^{(r)} = 0$, we can obtain the the interior nodal point displacements $U_i^{(r)}$ and boundary reactions $R_b^{(r)}$, due to interior nodal point external forces $P_i^{(r)}$.

Where $P_i^{(r)}$ refers only to those interior nodal point external forces within the bounds of the (r) substructure.

therfore with $U_b^{(r)} = 0$

$$U_i^{(r)} = (K_{ii}^{(r)})^{-1} P_i^{(r)} \qquad (61)$$

and

$$R_b^{(r)} = K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} P_i^{(r)} \qquad (62)$$

by evaluating the expression

$$(K_{bb}^{(r)} - K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}) = K_b^{(r)} \qquad (63)$$

we obtain the assembled substructure boundary stiffness matrix $K_b^{(r)}$ for the (r) substructure. $K_b^{(r)}$ will be symmetric and will usually be fully populated. Zero terms can occure in this matrix, depending on the element configuration chosen, but in general there will be few if any zero terms.

Fig. 12 shows the assembled substructure boundary stiffness matrix $K_b^{(r)}$ for the (r) substructure shown in fig. 7 with nodal point numbers given by fig. 4 .

5	6	10	14	15	19	21	22	
K	K	K	K	K	K	K	K	5
K	K	K	K	K	K	K	K	6
K	K	K	K	K	K	K	K	10
K	K	K	K	K	K	K	K	14
K	K	K	K	K	K	K	K	15
K	K	K	K	K	K	K	K	19
K	K	K	K	K	K	K	K	21
K	K	K	K	K	K	K	K	22

Fig.12 assembled substructure boundary stiffness matrix $K_{\ell}^{(r)}$ for the (r) substructure shown in fig. 7 with nodal point numbering given by fig. 4 . The matrix is symmetric and is fully populated. Each term will be a 2×2 submatrix as indicated by eq.(25).

1.4 Assembly of Substructures (ref. 2, 3, and 4)

Once the assembled substructure boundary stiffness matrix for each substructure has been formed. The assembled boundary stiffness matrix K_ℓ for the complete structure can be obtained by assembling each assembled substructure boundary stiffness matrix in a simmilar manner as when assembling individual triangular elements as described on page 25;ie for boundary nodal points their direct stiffnesses are summed together, and the cross stiffnesses are obtained by summing cross stiffness terms to associated adjacent nodal points.

For example fig.13 indicates the boundary nodal points of fig.7 numbered according to fig. 4 . The assembled boundary stiffness matrix K_ℓ for fig. 13 is given by fig. 9 .

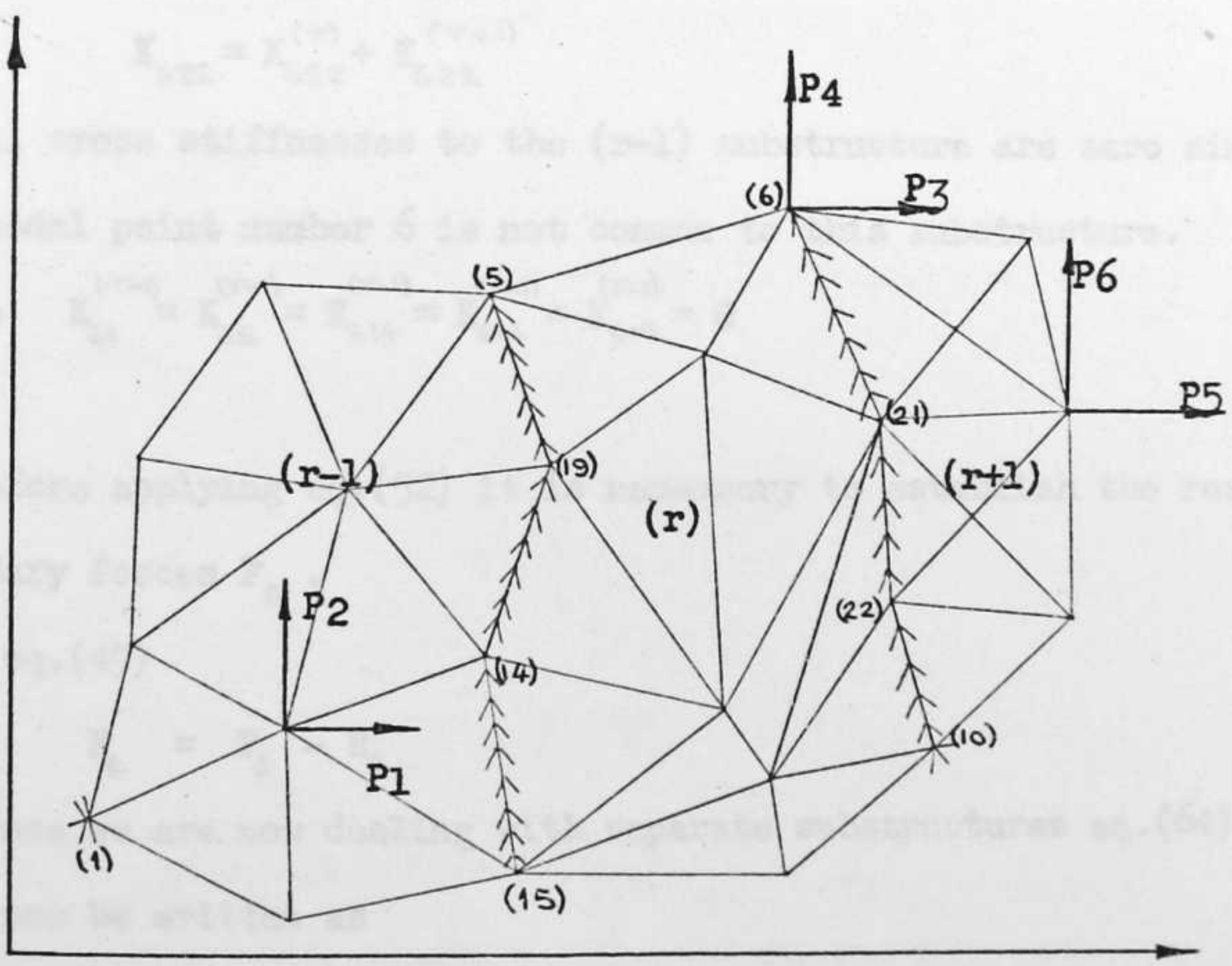


Fig. 13 boundary nodal points of fig. 7 numbered according to fig. 4 .

If we consider say boundary nodal point number 6; which from fig. 13 is seen to be common to substructures (r) and (r+1). Refering to fig. 9 row 3. The only non-zero stiffness terms are

(a) The direct stiffness K_{66}, where

$$K_{66} = K_{66}^{(r)} + K_{66}^{(r+1)}$$

(b) The cross stiffnesses $K_{65}, K_{610}, K_{614}, K_{615}, K_{619}, K_{621}, K_{622}$ to adjacent boundary nodal points 5,10,14,15,19,21, and 22.

where

$$K_{65} = K_{65}^{(r)}$$

$$K_{610} = K_{610}^{(r)} + K_{610}^{(r+1)}$$

$$K_{614} = K_{614}^{(r)}$$

$$K_{615} = K_{615}^{(r)}$$

$$K_{619} = K_{619}^{(r)}$$

$$K_{621} = K_{621}^{(r)} + K_{621}^{(r+1)}$$

$$K_{622} = K_{622}^{(r)} + K_{622}^{(r+1)}$$

All cross stiffnesses to the (r-1) substructure are zero since boundary nodal point number 6 is not common to this substructure.

ie $K_{61}^{(r-1)} = K_{65}^{(r-1)} = K_{614}^{(r-1)} = K_{615}^{(r-1)} = K_{619}^{(r-1)} = 0$

Before applying eq.(52) it is necessary to establish the resultant boundary forces F_ℓ.

from eq.(49)

$$F_\ell = P_\ell - R_\ell \tag{64}$$

Since we are now dealing with separate substructures eq.(64) must now be written as

$$F_\ell = P_\ell - \sum R_\ell^{(r)} \tag{65}$$

Where the summation term is required, as eq.(62) refers only to the (r) substructure.

As an example, consider again boundary nodal point 6 fig. 13 . The resultant boundary forces for this boundary nodal point are

X direction

$$F_{\ell_x} = P3 - R^{(r)}_{\ell_x} - R^{(r+1)}_{\ell_x}$$

Y direction

$$F_{\ell_y} = P4 - R^{(r)}_{\ell_y} - R^{(r+1)}_{\ell_y}$$

There will be no $R^{(r-1)}_{\ell}$ term since boundary nodal point 6 is not common to the (r-1) substructure.

Having formed F_ℓ and K_ℓ we may now relax all the boundary nodal points (except those restrained boundary nodal points which are required to render K singular) by applying the relationship

$$F_\ell = K_\ell \; U_\ell \tag{66}$$

Inverting the matrix K_ℓ we obtain the boundary nodal point displacements. U_ℓ .

$$U_\ell = K_\ell^{-1} \; U_\ell \tag{67}$$

from eq.(59) we may write

$$U_i^{(r)} = U_{i_{U_\ell=0}}^{(r)} - (K_{ii}^{(r)})^{-1} \; K_{i\ell}^{(r)} \; U_\ell^{(r)} \tag{68}$$

where $U_{i_{U_\ell=0}}^{(r)}$ is obtained from eq.(61)

Substituting the respective substructure boundary nodal point displacements $U_\ell^{(r)}$ from eq.(67) into eq.(68) we obtain the interior nodal point displacements $U_i^{(r)}$ with the boundary relaxed.

Applying eq.(68) for each substructure we obtain all the required nodal point displacements, and a direct application of eq.(13) will give the element stresses.

To summerise the foregoing theory, fig. 14 represents the main steps of the analysis, and these basic steps will be used in the computer analysis.

Form element stiffness matrix $K^{(e)}$ for elements within bounds of the (r) substructure.

Form the assembled substructure stiffness matrix $K^{(r)}$.

Invert $K_{ii}^{(r)}$

Obtain interior boundary displacements $U_i^{(r)}$ boundary fixed.

Form $(K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$

Obtain substructure boundary reactions $R_b^{(r)}$, boundary fixed.

Form $K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$

Form the assembled substructure boundary stiffness matrix $K_b^{(r)}$.

$K_b^{(r+1)}$, etc.

$R_b^{(r+1)}$, etc.

Form the assembled boundary stiffness matrix K_b for the complete structure.

Obtain resultant boundary forces F_b.

Introduce boundary conditions

Obtain boundary nodal point displacements U_b boundary relaxed.

P.T.O.

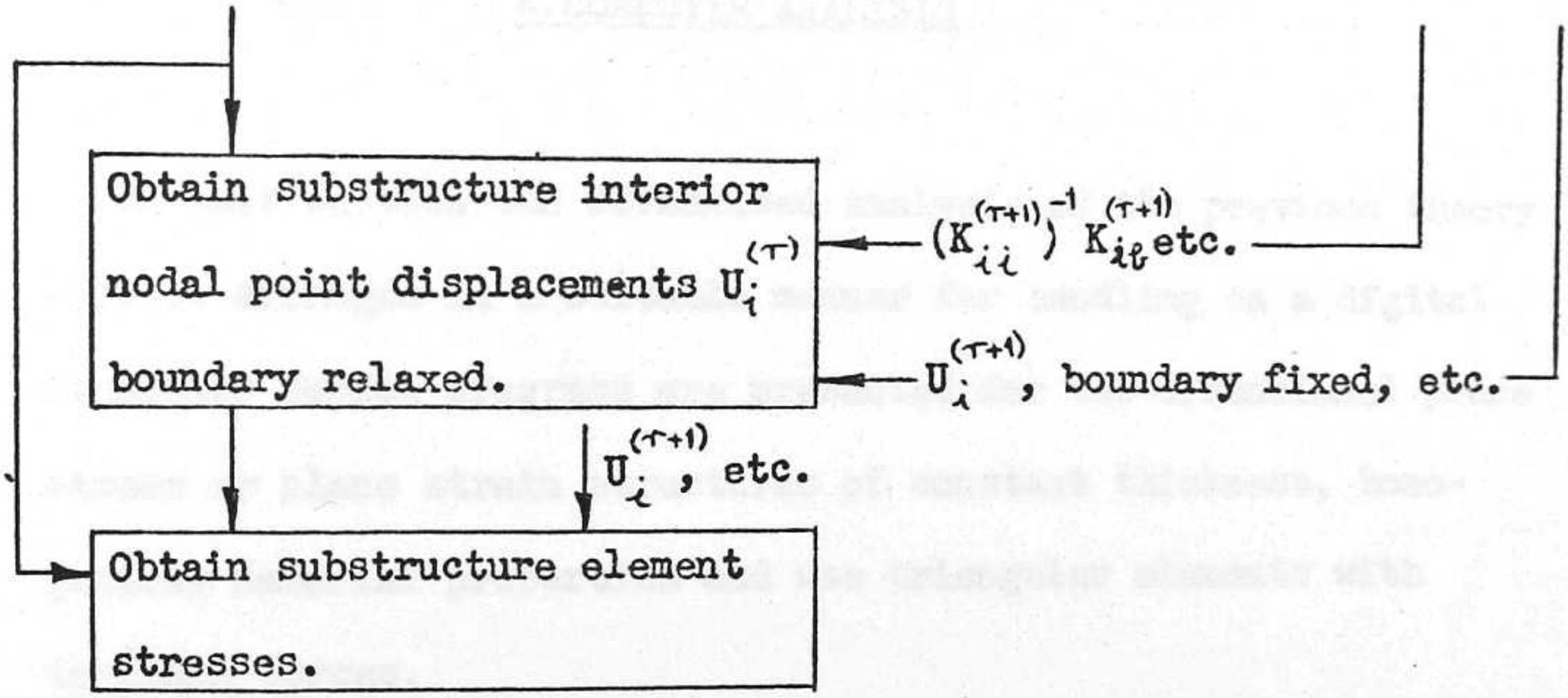


Fig. 14 flow diagram of the analysis.

2. COMPUTER ANALYSIS

In this section the formalised analysis of the previous theory will be arranged in a suitable manner for handling on a digital computer. Sample programs are presented for two-dimensional plane stress or plane strain structures of constant thickness, homogeneous material properties and use triangular elements with in-plane forces.

The sample programs are written in I.C.L. 1900 FORTRAN and are arranged to accomodate a computer having at least 18K core store available. The programs can be rewritten in any programming language, and by altering the dimensions of the arrays can be made to suit all but the very smallest of computers.

The computer analysis differs from the formalised theory in a number of respects, and these will now be discussed in detail.

2.1 Structure Numbering System

Numbering of the structure is carried out in two sets of numbering order. Namely Global numbering and Substructure numbering. The advantage of this numbering system is that small changes in the geometary of the structure may not require a complete re-numbering of the structure.

(a) Global numbering.

With the substructures assembled together the boundary nodal points are numbered consecutively from 1 to the total number of boundary nodal points in the complete structure. Where a boundary nodal point is as defined on page 32 .

For example consider fig. 15 which shows a gravity dam.

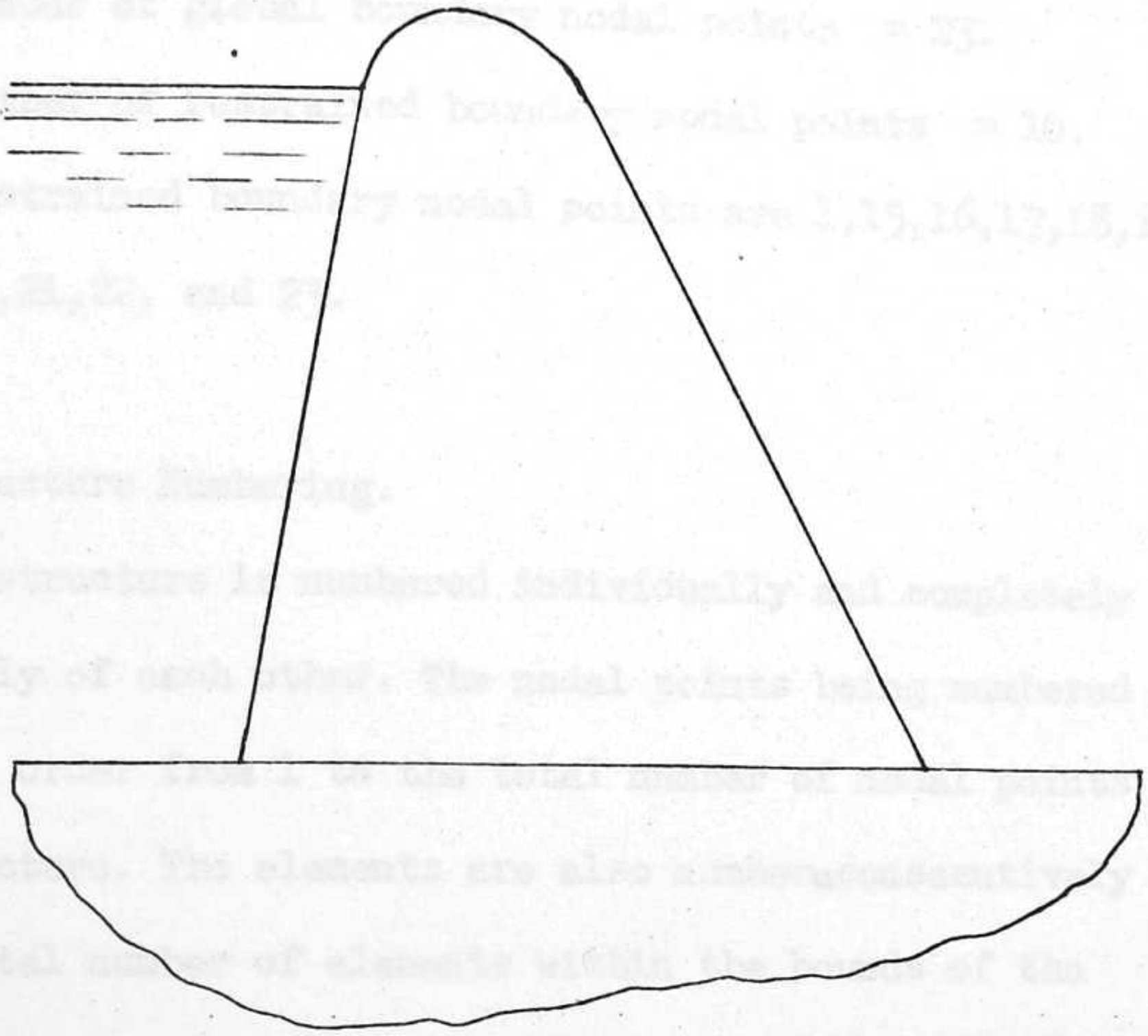


Fig. 15 .

The dam may be divided into a number of substructures as shown by fig. 16 .

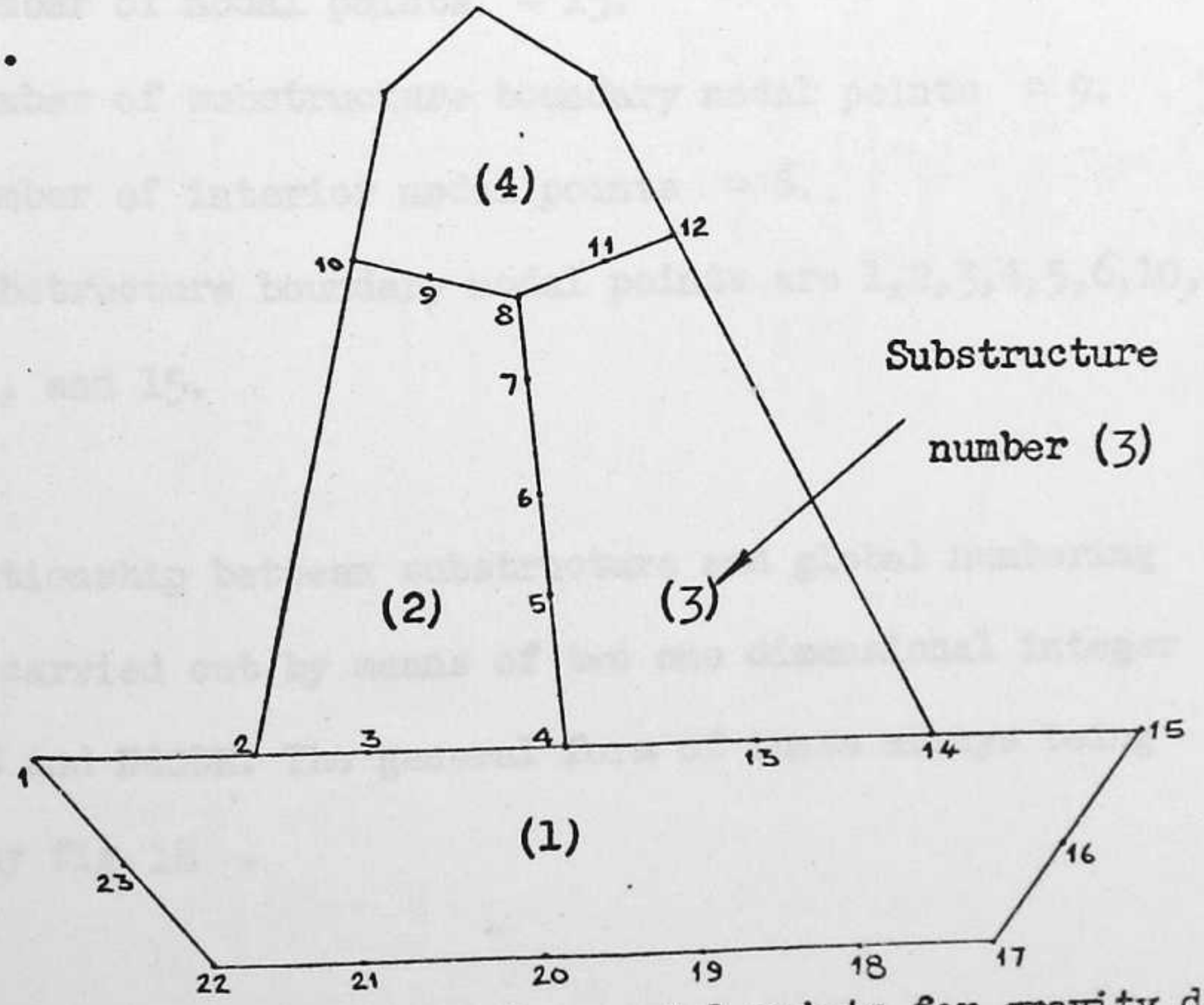


Fig. 16 global numbering of boundary nodal points for gravity dam.

Number of substructures = 4.

Number of global boundary nodal points = 23.

Number of restrained boundary nodal points = 10.

Restrained boundary nodal points are 1,15,16,17,18,19, 20,21,22, and 23.

(b) Substructure Numbering.

Each substructure is numbered individually and completely independently of each other. The nodal points being numbered in consecutive order from 1 to the total number of nodal points in the substructure. The elements are also numbered consecutively from 1 to the total number of elements within the bounds of the substructure.

For example, consider say substructure (3)from fig. 16 . This may be idealized using triangular elements as shown in fig. 17 .

Number of elements = 16.

Number of nodal points = 15.

Number of substructure boundary nodal points = 9.

Number of interior nodal points = 6.

Substructure boundary nodal points are 1,2,3,4,5,6,10, 11, and 15.

The relationship between substructure and global numbering systems is carried out by means of two one dimensional integer arrays NOTE and NCODE. The general form of these arrays being indicated by fig.18 .

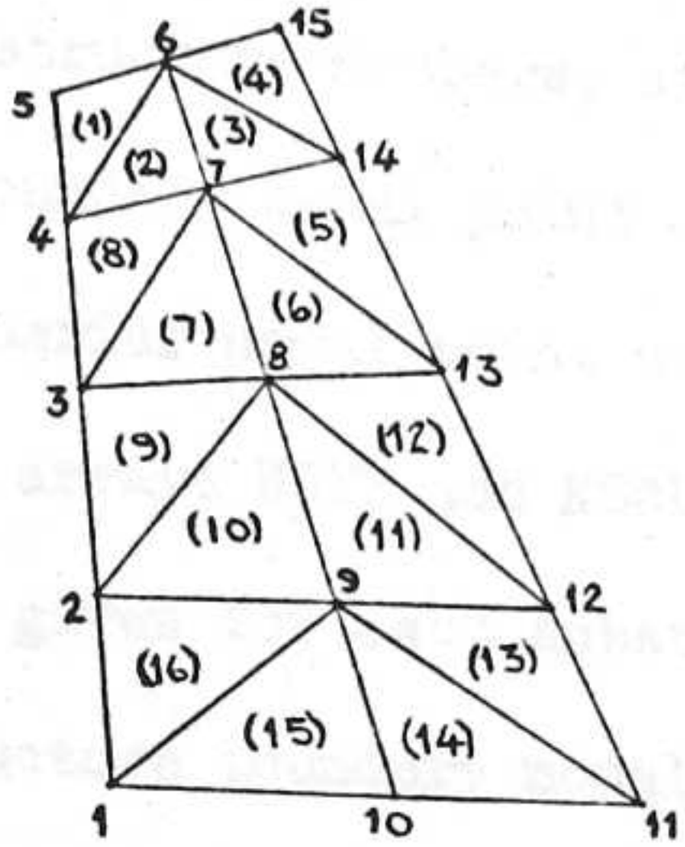


Fig. 17 typical substructure numbering of nodal points and elements for substructure number(3)of the gravity dam fig. 16 .

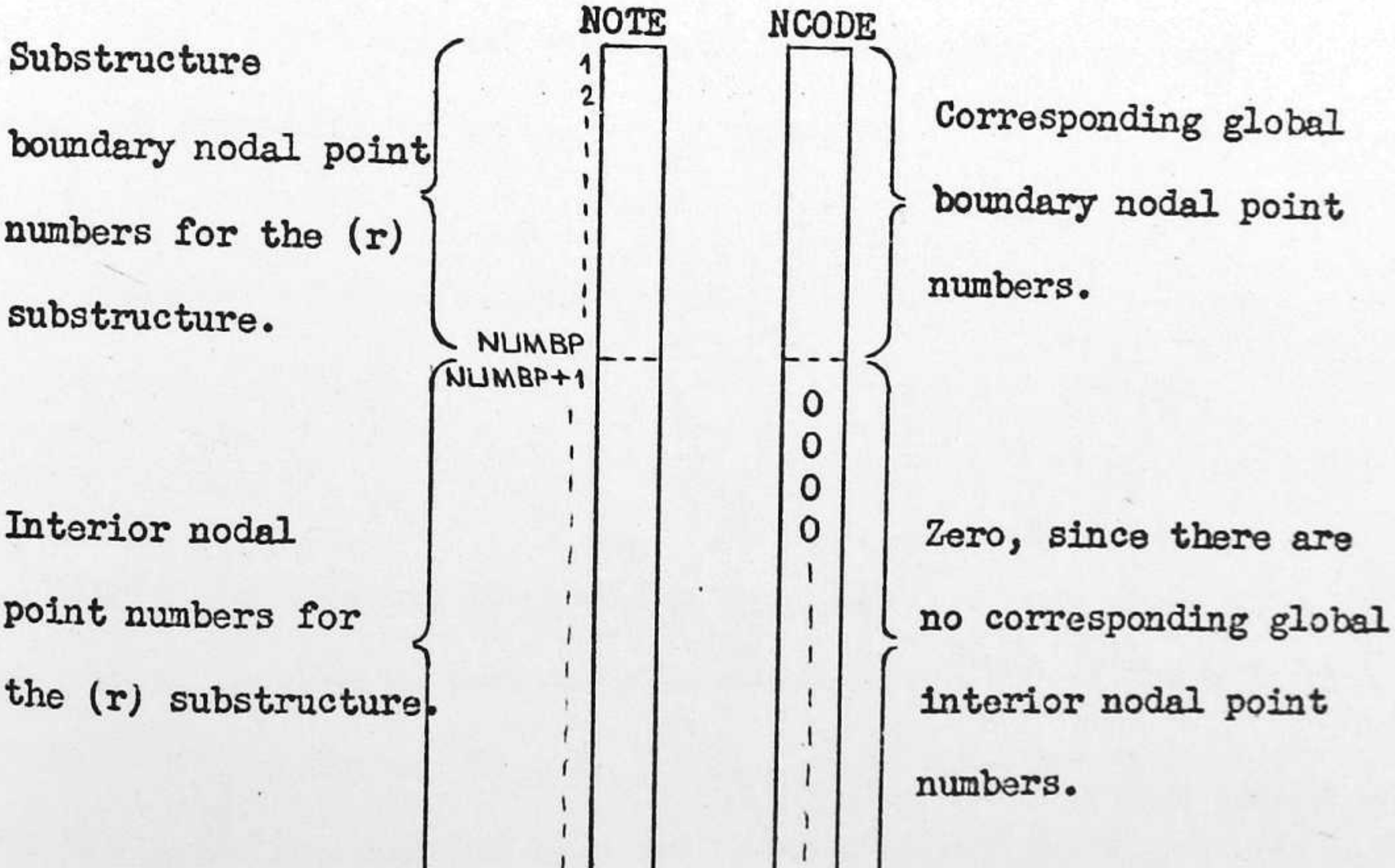


Fig. 18 general form of arrays NOTE and NCODE.

Array NOTE contains substructure numbers, array NCODE contains the corresponding global boundary nodal point numbers. There are no corresponding global interior nodal point numbers.

The actual formation of arrays NOTE and NCODE is preformed automatically by the program , given for each substructure.

The number of substructure boundary nodal points, NUMBP.

The number of nodal points, NUMNP.

The substructure boundary nodal point numbers, together with their corresponding global boundary nodal point numbers.

2.2 Storage of the Non-Zero Stiffness Terms(ref. 5)

If we recall from eq.(23) the 6×6 element stiffness matrix was partitioned into nine 2×2 submatrix.

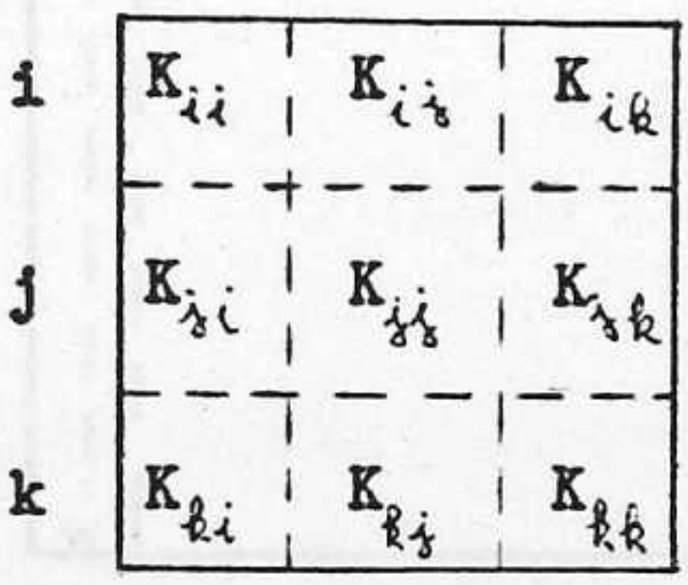


Fig.19 6×6 element stiffness matrix partitioned into nine 2×2 submatrix as indicated on page 23.

The form of each submatrix being

$$K_{\ell m}^{(e)} = \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}_{\ell m}^{(e)} \tag{69}$$

Within the computer program the four terms of each submatrix, are stored in core by four two dimensional real arrays SXX,SXY,SYX, andSYY which refer to K_{xx}, K_{xy}, K_{yx}, and K_{yy} respectively and the terms are related by a two dimensional array NP refered to as an identity matrix.

The general form of this identity matrix is given by fig,(20). The first column of the identity matrix NP is numbered consecutively from 1 to n and refers to the nodal point numbers; n being the total number of nodal points under consideration. Note that the row number equals the nodal point number.

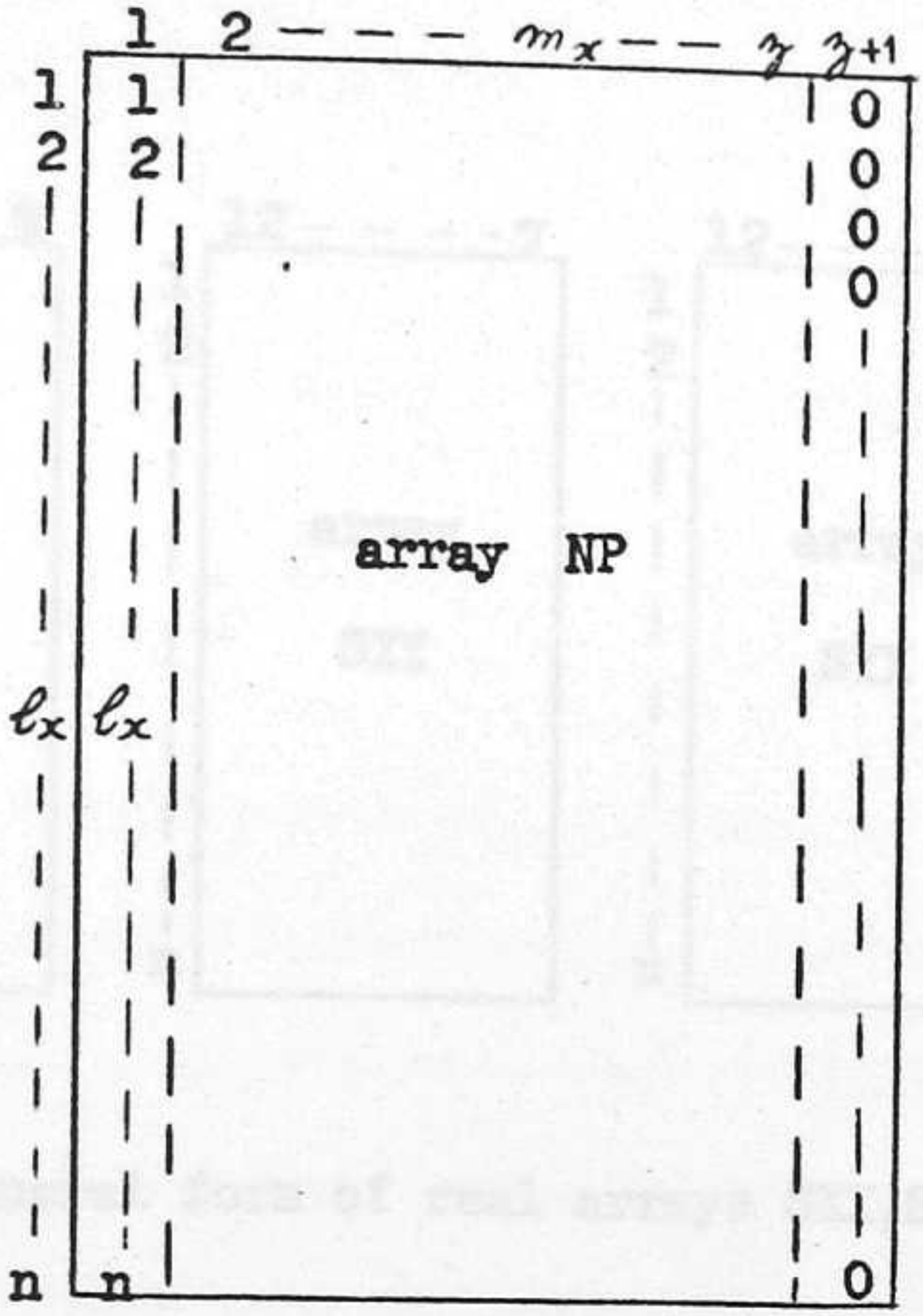


Fig. 20 general form of the identity matrix NP.

If we consider the(l_x) nodal point then the terms of the (l_x) row of array NP from 2 to z contain a note of these nodal points which are adjacent to nodal point (l_x).

The $z+1$ column should contain only zero terms; if not, then this indicates that there are too many nodal points adjacent to the nodal point indicated by column 1. for the computer storage to handle.

Refering now to fig. 21. The firstcolumn of arrays SXX,SXY,SYX, andSYY will contain the direct stiffness terms for the nodal points indicated by the first conumn of array NP. The remaining terms of each row from columns 2 to z contain the cross stiffness terms of those adjacent nodal points indicated by columns 2 to z of array NP.

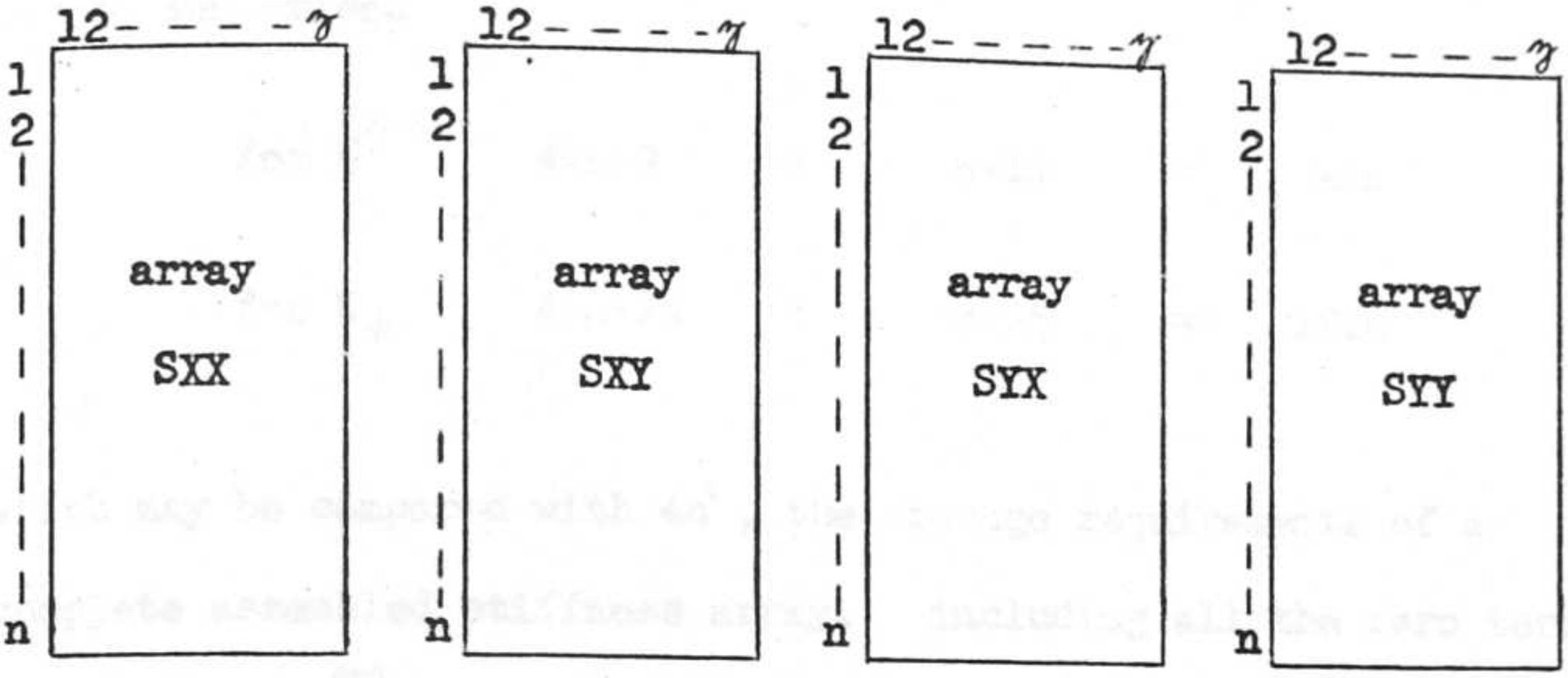


Fig. 21 general form of real arrays SXX,SXY,SYX, and SYY.

The size of arrays SXX,SXY,SYX, and SYY is n $\times y$; where n is the number of nodal points being considered, and $y = x+1$; where x is the maximum number of adjacent nodal points that any nodal point is likely to have. Ideally both n and y would require to be as large as possible to cover all situations. But computer storage limitations, demand that y be limited to some practical value, and then n increased to use up the remainder of the core storage available.

For the storage of the assembled substructure stiffness matrix $K^{(r)}$ a value of $y = 9$, allowing 8 adjacent nodal points to any one nodal point will be found to be sifficient for most purposes. For the assembled boundary stiffness matrix K_b, y has been limited to 24, allowing 23 adjacent global boundary nodal points to any one global boundary nodal point.

The magnitude of y can of course be altered at will, to suit individual problems(often at the expence of n.) providing that sufficient computer storage is available.

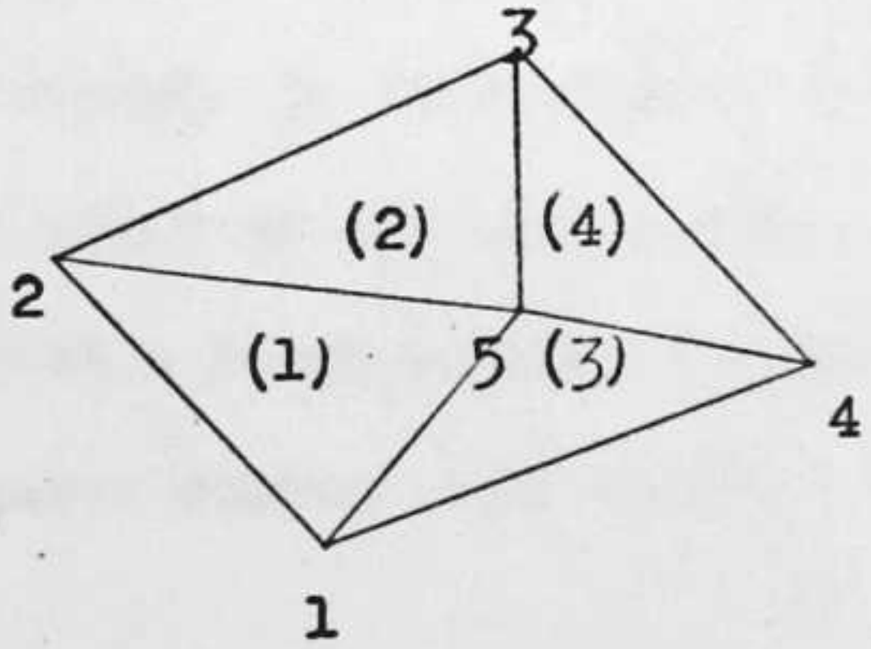
The storage requirements of arrays SXX, SXY, SYX, SYY, and NP may be summerised

for $K^{(r)}$	4×n×9	+	n×10	= 46n
for K_{ℓ}	4×n×24	+	n×25	= 121n

which may be compared with $4n^2$, the storage requirements of a complete assembled stiffness array. including all the zero terms.

The arrays$K^{(r)}$ andK_{ℓ} are assembled stiffness matrix and the computer program has been arranged to automatically assemble the the stiffness terms into SXX, SXY, SYX, and SYY. The assembly process will be described on page119.

As an example of the storage of the non-zero terms, consider the four elements of fig.22 .



Element number.	N.P. number order.	i	j	k
(1)		1	5	2
(2)		2	5	3
(3)		1	4	5
(4)		5	4	3

Fig. 22 arbitary four element structure.

The positioning of the assembled stiffness terms and terms of the identity array for fig. 22 are indicated in figs 23 and 25.

	1	2	3	4	5	6	7	8	9	10
1	1	5	2	4	0	0	0	0	0	0
2	2	1	5	3	0	0	0	0	0	0
3	3	2	5	4	0	0	0	0	0	0
4	4	1	5	3	0	0	0	0	0	0
5	5	1	2	3	4	0	0	0	0	0

Fig 23 terms of array NP for the structure shown in fig. 22.

	1	2	3	4	5	6	7	8	9
1	$K_{11}^{(1,3)}$	$K_{15}^{(1,3)}$	$K_{12}^{(1)}$	$K_{14}^{(3)}$	0	0	0	0	0
2	$K_{22}^{(1,2)}$	$K_{21}^{(1)}$	$K_{25}^{(1,2)}$	$K_{23}^{(2)}$	0	0	0	0	0
3	$K_{33}^{(2,4)}$	$K_{32}^{(2)}$	$K_{35}^{(2,4)}$	$K_{34}^{(4)}$	0	0	0	0	0
4	$K_{44}^{(3,4)}$	$K_{41}^{(3)}$	$K_{45}^{(3,4)}$	$K_{43}^{(4)}$	0	0	0	0	0
5	$K_{55}^{(1,2,3,4)}$	$K_{51}^{(1,3)}$	$K_{52}^{(1,2)}$	$K_{53}^{(2,4)}$	$K_{54}^{(3,4)}$	0	0	0	0

Fig. 24 positioning of terms within arrays SXX, SXY, SYX, and SYY for the structure shown in fig. 22.

The cross stiffness terms K_{13}, K_{24}, K_{31}, K_{42} will be zero and are therefore not stored. In this example there will be no saving of storage space, since we are only dealing with 5 nodal points. In practice, with a large number of nodal points considerable saving in computer storage will result.

2.3 Rearrangement of substructure stiffness matrix $K^{(\tau)}$.

The actual rearrangement of the assembled substructure stiffnes matrix $K^{(\tau)}$ required by partitioned eq.(58), does not in fact take place. ie the arrangement of arrays SXX, SXY, SYX, and SYY are not altered by partitioning. The rearranged order is however held by the array NOTE. The general form of this array being described on page 55. Thus throughout the analysis reference must always be made to array NOTE before preforming any matrix algebra involving matrix $K_{\beta}^{(\tau)}$.

2.4 Formation of the Substructure Boundary Stiffness Matrix $K_{\ell}^{(\tau)}$.

The complete storage of arrays $K_{\ell\ell}^{(\tau)}, K_{i\ell}^{(\tau)}, K_{\ell i}^{(\tau)}, (K_{ii}^{(\tau)})^{-1}$, and $K_{\ell}^{(\tau)}$ used to form $K_{\ell}^{(\tau)}$ does not take place.

Only

$(K_{ii}^{(\tau)})^{-1}$

two columns of $(K_{ii}^{(\tau)})^{-1} K_{i\ell}^{(\tau)}$

two rows of $K_{\ell i}^{(\tau)}$

a 2×2 submatrix of $K_{\ell}^{(\tau)}$

are held in core at any instant of time.

The procedure for forming $K_{\ell}^{(\tau)}$ is as follows.

a/ $(K_{ii}^{(\tau)})^{-1}$ is stored completely, and the first two columns of $K_{i\ell}^{(\tau)}$ as shown in fig. 25.

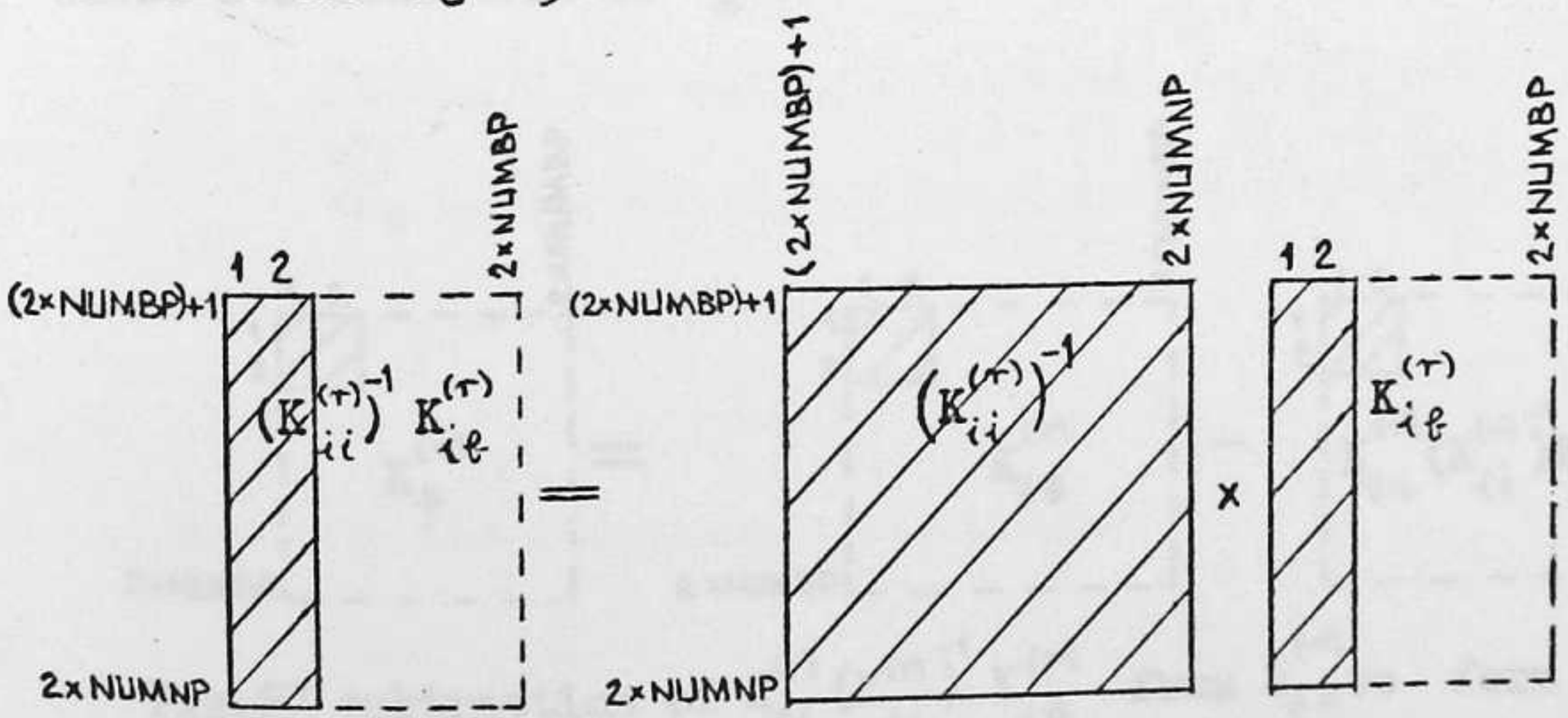


Fig. 25 multiplcation of $(K_{ii}^{(\tau)})^{-1} K_{i\ell}^{(\tau)}$. The shaded areas represent the size of the arrays held in core.

Multiplying $(K_{ii}^{(\tau)})^{-1}$ by the two columns of $K_{i\ell}^{(\tau)}$ gives the first two columns of $(K_{ii}^{(\tau)})^{-1} K_{i\ell}^{(\tau)}$

b/ Storing the first two rows of $K_{bi}^{(r)}$ and multiplying by the first two columns of $(K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$ gives a 2×2 submatrix of $K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$ as shown by fig. 26.

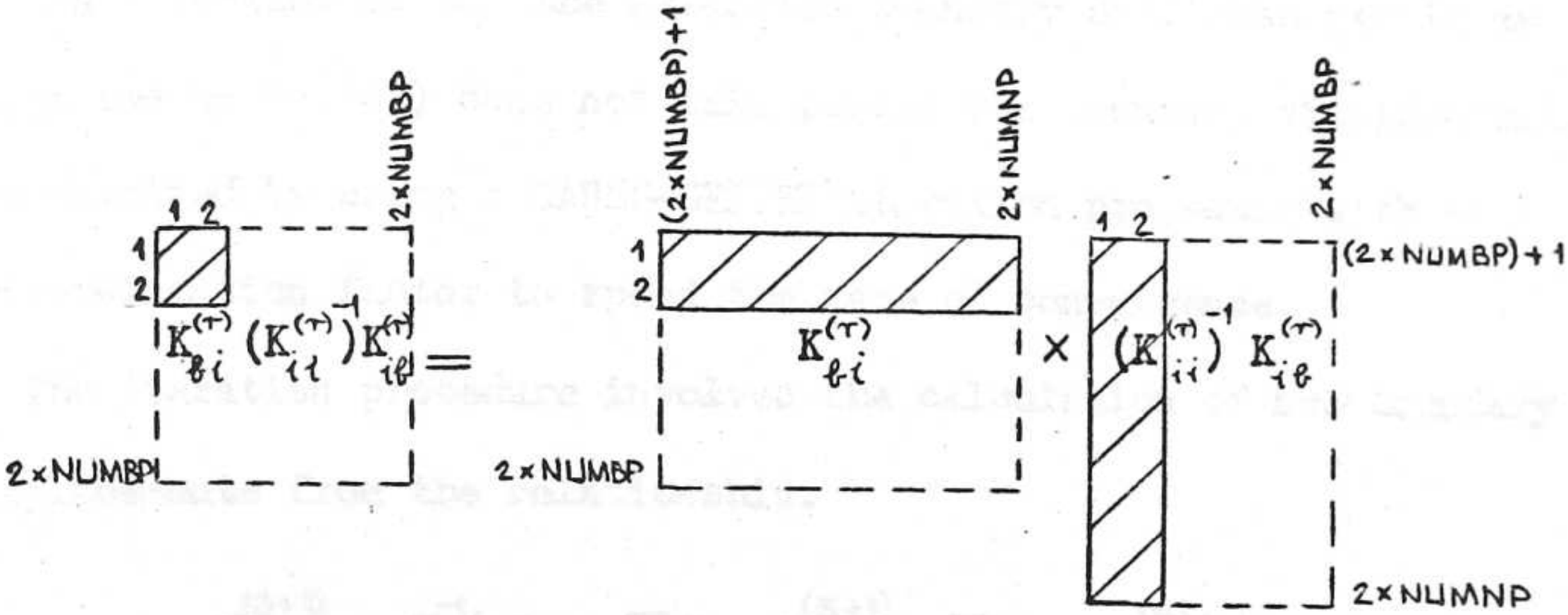


Fig.26 multiplcation of $K_{bi}^{(r)}$ and $(K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$ to form $K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$

c/ Subtracting the first 2×2 submatrix of $K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$ from the corresponding first 2×2 submatrix from $K_{bb}^{(r)}$ gives the first 2×2 submatrix of $K_{b}^{(r)}$.

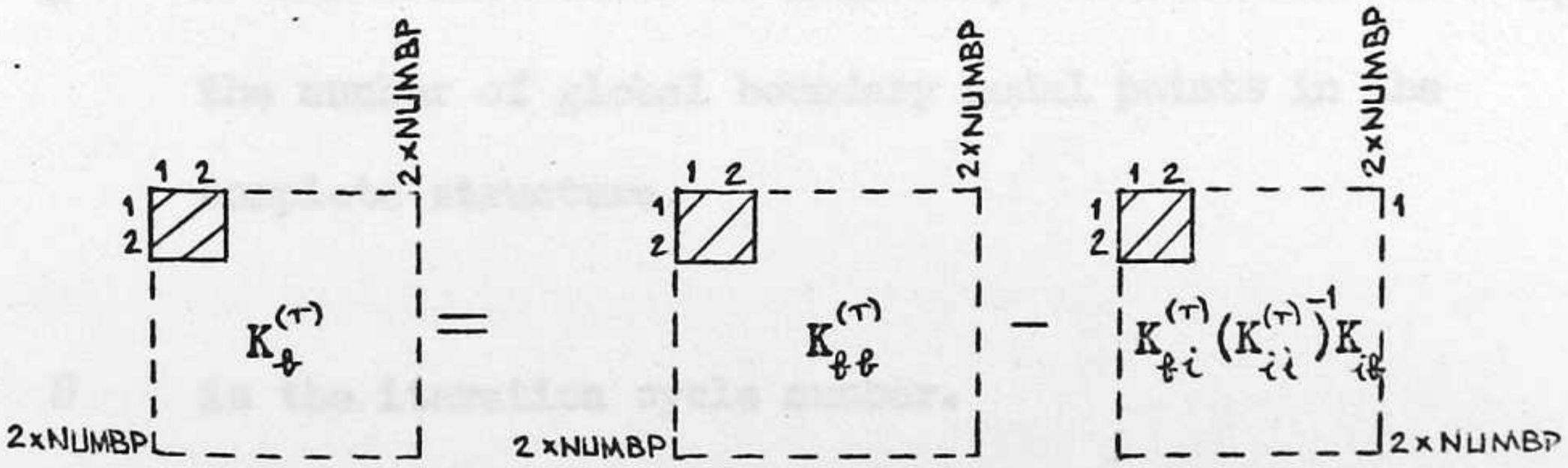


Fig.27 subtraction of $K_{bi}^{(r)} (K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$ from $K_{bb}^{(r)}$ to form $K_{b}^{(r)}$.

By looping back to b/ NUMBP- 1 times, and storing the next two rows of $K_{bi}^{(r)}$. each time the first two columns of $K_{b}^{(r)}$ is obtained.

Looping right back to a/ and repeating the above procedure NUMBP-1 times, each time storing the next two columns of $K_{ib}^{(r)}$.
In this way the complete substructure boundary stiffness matrix $K_{b}^{(r)}$ is formed.

2.5 Determination of Boundary Nodal Point Displacements. (ref.5).

The invertion of K_{ℓ} the assembled boundary stiffness matrix as suggested by eq.(67) does not take place. The boundary displacements are obtained by using a GAUSS-SEIDEL iteration procedure with an over-relaxation factor to speed the rate of convergence.

The iteration procedure involves the calculation of new boundary displacements from the relationship.

$$U_{\ell_n}^{(s+1)} = K_{\ell_{nn}}^{-1}\left(F_{\ell_n} - \sum_{i=1,n-1} K_{\ell_{ni}} U_{\ell_i}^{(s+1)} - \sum_{i=n+1,N} K_{\ell_{ni}} U_{\ell_i}^{(s)} \right) \qquad (70)$$

where

n is the number of the unknown displacement under consideration.

N is the total number of unknowns, which in this case equals the number of global boundary nodal points in the complete structure.

S is the iteration cycle number.

$U_{\ell_n} = \begin{bmatrix} U_{\ell_x} \\ - - - \\ U_{\ell_y} \end{bmatrix}_n$ the X and Y displacements of the n boundary nodal point.

$F_{\ell_n} = \begin{bmatrix} F_{\ell_x} \\ - - - \\ F_{\ell_y} \end{bmatrix}_n$ the X and Y resultant boundary force at global boundary nodal point n.

$K_{\ell_{ni}} = \begin{bmatrix} K_{\ell_{xx}} & K_{\ell_{xy}} \\ K_{\ell_{yx}} & K_{\ell_{yy}} \end{bmatrix}_{ni}$

$K^{-1}_{\ell_{nn}}$ is a 2×2 submatrix and refers to the flexibility of nodal point n, and represents the nodal point displacements due to unit nodal point forces.

therefore we may write

$$K^{-1}_{\ell_{nn}} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}_{nn} \tag{71}$$

The rate of convergence of the Gauss-Seidel iteration process may be greatly increased by the introduction of an over- relaxation factor β, which has the effect of moving the nodal point slightly beyond its equlibrium position before proceeding with the next nodal point. To apply the over-relaxation factor the relationship of eq.(70) must be altered slightly.

Let the change in displacement $\Delta U^{(s,s+1)}_{\ell_n}$ of nodal point n, between cycles (s) and (s+1) be given by

$$\Delta U^{(s,s+1)}_{\ell_n} = U^{(s+1)}_{\ell_n} - U^{(s)}_{\ell_n} \tag{72}$$

$$\therefore U^{(s+1)}_{\ell_n} = U^{(s)}_{\ell_n} + \Delta U^{(s,s+1)}_{\ell_n} \tag{73}$$

substituting eq.(73) in eq.(70) we have for the change of displacement between cycles.

$$\Delta U^{(s,s+1)}_{\ell_n} = K^{-1}_{\ell_{nn}} \left(F_{\ell_n} - \sum_{i=1,n-1} K_{\ell_{ni}} U^{(s+1)}_{\ell_i} - \sum_{i=n+1,N} K_{\ell_{ni}} U^{(s)}_{\ell_i}\right) - U^{(s)}_{\ell_n} \tag{74}$$

Introducing the over-relaxation factor into eq.(73), the new displacement $U^{(s+1)}_{\ell_n}$ becomes

$$U^{(s+1)}_{\ell_n} = U^{(s)}_{\ell_n} + \beta \cdot \Delta U^{(s,s+1)}_{\ell_n} \tag{75}$$

In matrix notation we may write eq.(74) as

$$\begin{bmatrix} \Delta U_{\ell_x} \\ \Delta U_{\ell_y} \end{bmatrix}^{(s,s+1)} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}_{nn} \begin{bmatrix} X \\ Y \end{bmatrix}^{(s,s+1)}_{n} \tag{76}$$

where

where

$$\begin{bmatrix} X \\ Y \end{bmatrix}_n^{(S,S+1)} = \begin{bmatrix} F_{\ell_x} \\ F_{\ell_y} \end{bmatrix}_n - \sum_{i=1,n-1} \begin{bmatrix} K_{\ell_{xx}} & K_{\ell_{xy}} \\ K_{\ell_{yx}} & K_{\ell_{yy}} \end{bmatrix}_{ni} \begin{bmatrix} U_{\ell_x} \\ U_{\ell_y} \end{bmatrix}_i^{(S+1)} - \sum_{i=n,N} \begin{bmatrix} K_{\ell_{xx}} & K_{\ell_{xy}} \\ K_{\ell_{yx}} & K_{\ell_{yy}} \end{bmatrix}_{ni} \begin{bmatrix} U_{\ell_x} \\ U_{\ell_y} \end{bmatrix}_i^{(S)} \tag{77}$$

Which represents the difference between the elastic forces (represented by the summation terms) at nodal point n due to deformation of the elements, and the resultant boundary forces F_{ℓ_n} obtained from eq.(65).

The new displacements written in matrix notation becomes.

$$\begin{bmatrix} U_{\ell_x} \\ U_{\ell_y} \end{bmatrix}_n^{(S+1)} = \begin{bmatrix} U_{\ell_x} \\ U_{\ell_y} \end{bmatrix}_n^{(S)} + \beta \begin{bmatrix} \Delta U_{\ell_x} \\ \Delta U_{\ell_y} \end{bmatrix}_n^{(S,S+1)} \tag{78}$$

To obtain convergence the over-relaxatinn factor β must lie within the range $1 \leqslant \beta < 2$and it will be found that for most two dimensional structures, optimum convergence will result with a value of β between 1.8 and 1.95.

To start of the iteration procedure, the unknown global boundary nodal point displacements must be given initial values (initial displacements). A good estimate of these initial displacements will greatly reduce the computation time.

2.6 Introduction of Boundary Conditions.(ref.5).

The assembled boundary stiffness matrix K_ℓ as used throughout the theory is non-singular and therefore any arbitary amount of displacement will satisfy eq.(66). To render K_ℓ singular, and obtain a unique solution to the particular problem under consideration, a certain minimum number of restraints must be introduced. The restraints are introduced into eq.(73) by modifing the flexibility terms of eq.(71) of each restrained nodal point.

If we consider a typical restrained nodal point fig. 28, restrained to move only along a line at an angle ϕ to the X-axis.

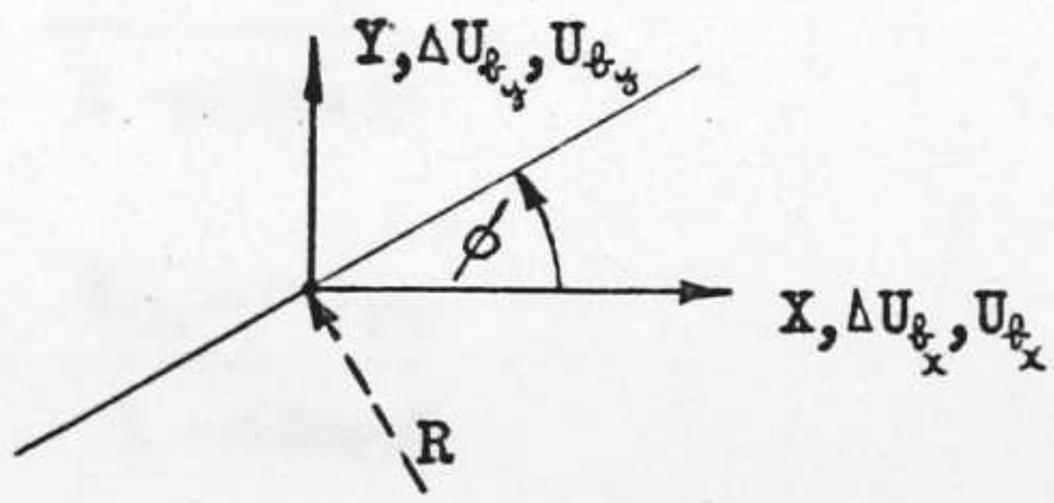


Fig. 28 typical restrained nodal point.

The unknown reaction is represented by force R and the difference in the elastic and applied forces arerepresented by X and Y forces obtained from eq.(77).

From eq.(76) the change in displacement between successive cycles of iteration may be written.

$$\begin{bmatrix} \Delta U_{\ell_x} \\ \Delta U_{\ell_y} \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} \begin{bmatrix} X - R\sin\phi \\ Y + R\cos\phi \end{bmatrix} \tag{79}$$

Expanding out eq.(79) we can eliminate the unknown reaction R.

therefore

$$\Delta U_{\theta_x} = \frac{f_{xx} - \alpha f_{yx}}{1 - \alpha \operatorname{Tan}\phi} X + \frac{f_{xy} - \alpha' f_{yy}}{1 - \alpha' \operatorname{Tan}\phi} Y \qquad (80)$$

where

$$\alpha = \frac{f_{xx} \operatorname{Tan}\phi - f_{xy}}{f_{yx} \operatorname{Tan}\phi - f_{yy}}$$

Now ΔU_{θ_x} and ΔU_{θ_y} are related to each other by the tangent of angle ϕ

$$\therefore \Delta U_{\theta_y} = \Delta U_{\theta_x} \operatorname{Tan}\phi \qquad (81)$$

therefore the modified flexibility terms are

$$f^*_{xx} = \frac{f_{xx} - \alpha f_{yx}}{1 - \alpha \operatorname{Tan}\phi}$$

$$f^*_{xy} = \frac{f_{xy} - \alpha f_{yy}}{1 - \alpha \operatorname{Tan}\phi} \qquad (82)$$

$$f^*_{yx} = (\operatorname{Tan}\phi) f^*_{xx}$$

$$f^*_{yy} = (\operatorname{Tan}\phi) f^*_{xy}$$

The number of restrained nodal points in any structure depends on the boundary conditions pertaining to the particular structure under consideration, but the choice of the restrained nodal points muct be such that the structure is restrained in two directions (for two dimensional structures) and the position of the restrained nodal points must prevent the rotation of the structure. Thus a single restrained nodal point fixed in both the X and Y directions would not be sufficient restraint since this condition will not prevent rotation of the structure. At least one other restrained

nodal point would have to be specified.

2.7 Nodal Point Stresses.(ref. 1 and 5)

The three components of element stress σ_x , σ_y , and τ_{xy} as obtained from eq.(13), do not represent the state of stress at any particular point within the element, although it is usual to consider the three stresses as acting at the centroid of the element.

To obtain more points for evaluating the stress distrubution across a structure, it has been found convenient to estimate nodal point stresses by averageng the element stresses of all those elements which are adjacent to the nodal point. Good results have been obtained for all nodal points except those which are on the physical boundary of the structure (ref.9).

If we consider the element shown in fig. 29.

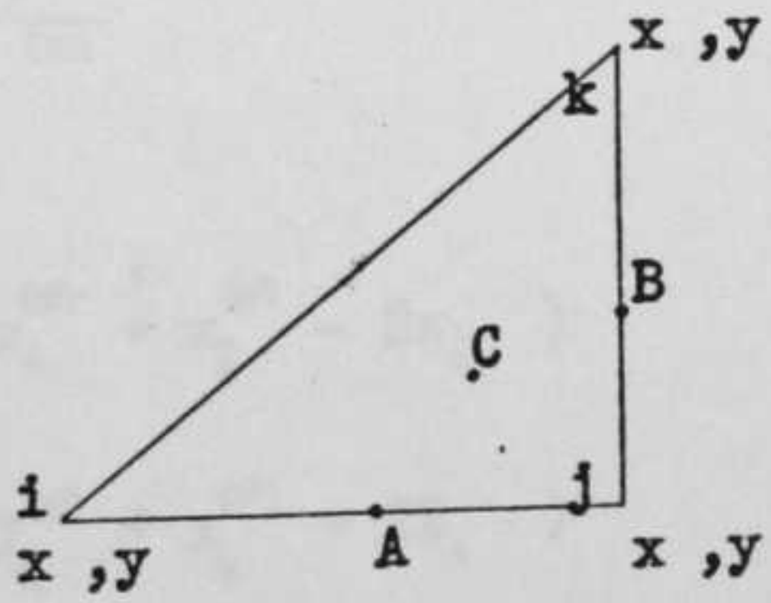


Fig. 29 typical element(e)attached to nodal point i .

The stresses must be consistant with the nodal point displacements.

approximates the horizontal stress of point A,

approximates the vertical stress at point B,

approximates the shearing stress at some point C within the element.

For this case, when determining the stresses at nodal point i, the horizontal stress at A. must be weighted more heavily than the vertical stress at B. Therefore a "weighted average" method of determining the nodal point stresses is desireable.

The three components of stress at nodal point i may be estimated from the following relationships.

$$\sigma_x = \frac{1}{S_x} \sum_{(e)} \frac{a^{(e)}}{a^{(e)} + b^{(e)}} \sigma_x^{(e)} \tag{83}$$

$$\sigma_y = \frac{1}{S_y} \sum_{(e)} \frac{b^{(e)}}{a^{(e)} + b^{(e)}} \sigma_y^{(e)} \tag{84}$$

$$\tau_{xy} = \frac{1}{N} \sum_{(e)} \tau_{xy}^{(e)} \tag{85}$$

where

$$a^{(e)} = (x_k^{(e)} + x_j^{(e)} - 2x_i) \tag{86}$$

$$b^{(e)} = (y_k^{(e)} + y_j^{(e)} - 2Y_i) \tag{87}$$

$$S_x = \sum \frac{a^{(e)}}{a^{(e)} + b^{(e)}} \tag{88}$$

$$S_y = \sum \frac{b^{(e)}}{a^{(e)} + b^{(e)}} \tag{89}$$

The summation $\sum$ is preformed on all N elements adjacent(connected) to nodal point i.

The calculation of element stresses takes place for each substructure in turn. For boundary nodal points all the adjacent

element stresses may not be known, since some of the adjacent elements may be within an adtacent substructure. Therefore only Interior nodal point stresses are calculated.

2.8 Computer Check and Comparason of Results.

The computer program was checked by analysing a uniform stress field.

The nodal point displacements converged exactly with theory.

The element and interior nodal point stress es were within 0.0015% of theory.

The following examples have been analysed by the given programs, and the results plotted.

Thick Cylinder under Internal Pressure of 10,000lb.f/□in.

A segment of the cylinder was idealized with two substructures fig. 30, giving a total of 42 elements and 32 nodal points. The internal pressure being achieved by consentrated forces at the cylinder bore nodal points. The boundary conditions were arranged so that restrained nodal points only displaced in a radial direction.

The results, fig. 31,32, and 33 suggest that the element mesh was adequate for nodal point displacements, but that a finer mesh would be required about the mid-radius of the cylinder for acceptable element stress.

10" Dia. Disc under a Point Load of 2000lb.f.

A quarter segment of a disc was idealized with four substructure fig. 34, giving a total of 234 elements and 136 nodal points.

The results, fig.'s 35 to 38 indicate a good agreement with theory. The largest errors being in the vicinity of the load point.

Cantilever Beam under an end load of 5000lb.f.

A $2\frac{1}{4}$" length of beam was idealized with three substructures fig.'s39, and40 , giving a total of 140 elements and 87 nodal points.

The results, fig.'s41 , 42 , and43 indicate good agreement with the stress distrubution across and along the beam, but considerable discrepancy occures with the nodal point displacements. The fact that the displacement of the beam is large compared to the element deformation, and since the beam was analysed using only 21 global boundary nodal points, this is likely to be the reason for the large error.

It is therefore suggested that the maximum number of global boundary nodal points be used up when analysing a structure.

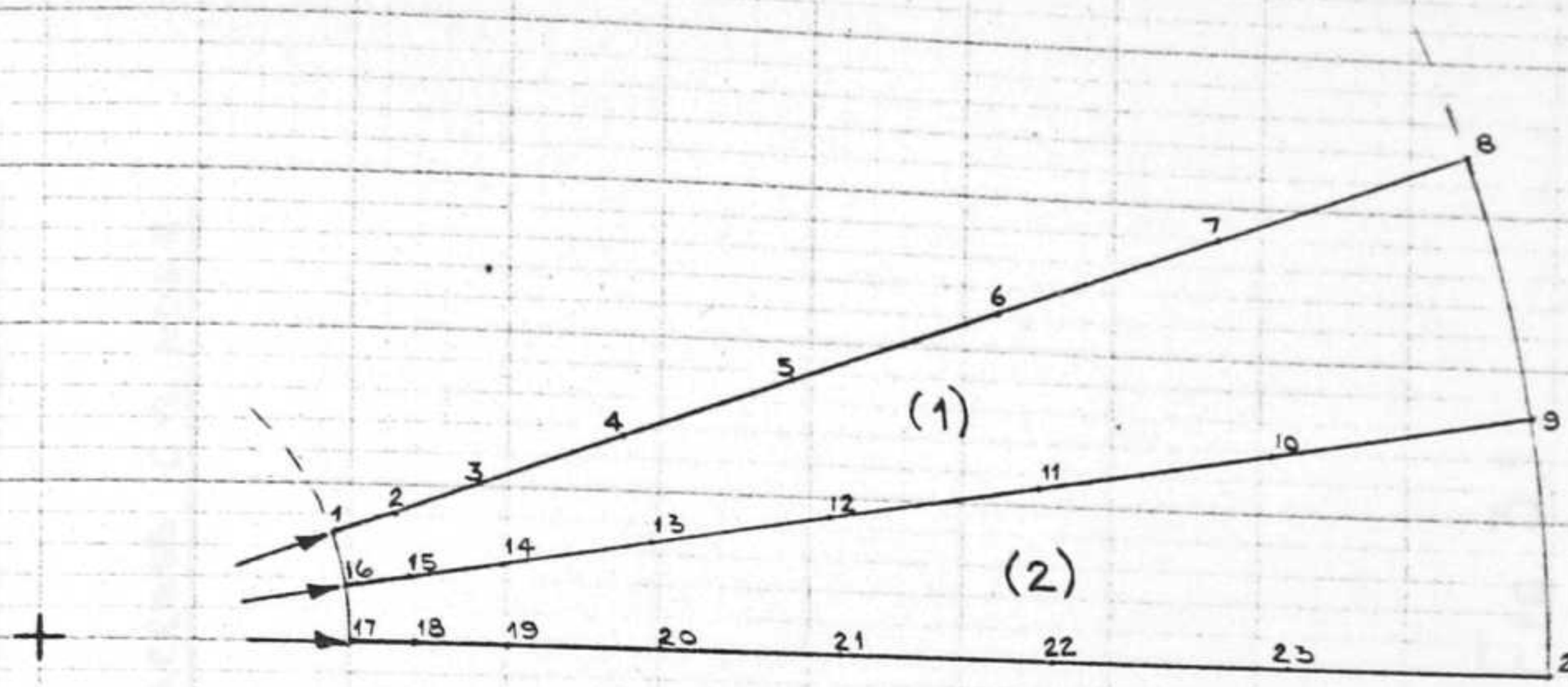


Fig.30a full size segment of thick cylinder.

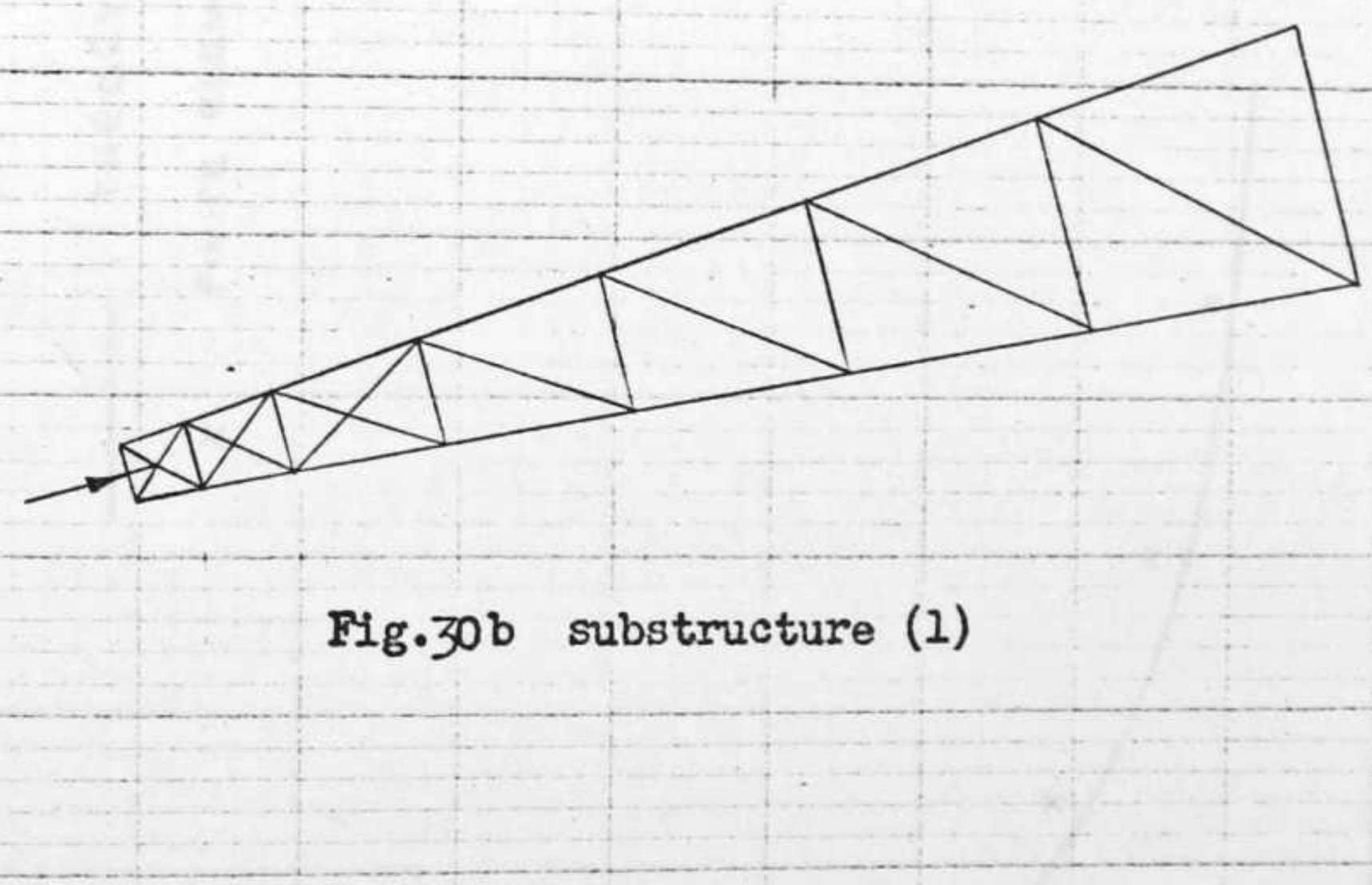


Fig.30b substructure (1)

Fig.30c substructure (2)

THICK CYLINDER UNDER INTERNAL PRESSURE OF 10.000 LBF/□"

HOOP STRESS VARIATION ACROSS CYLINDER

——— THEORY

× . FINITE ELEMENT STRESS

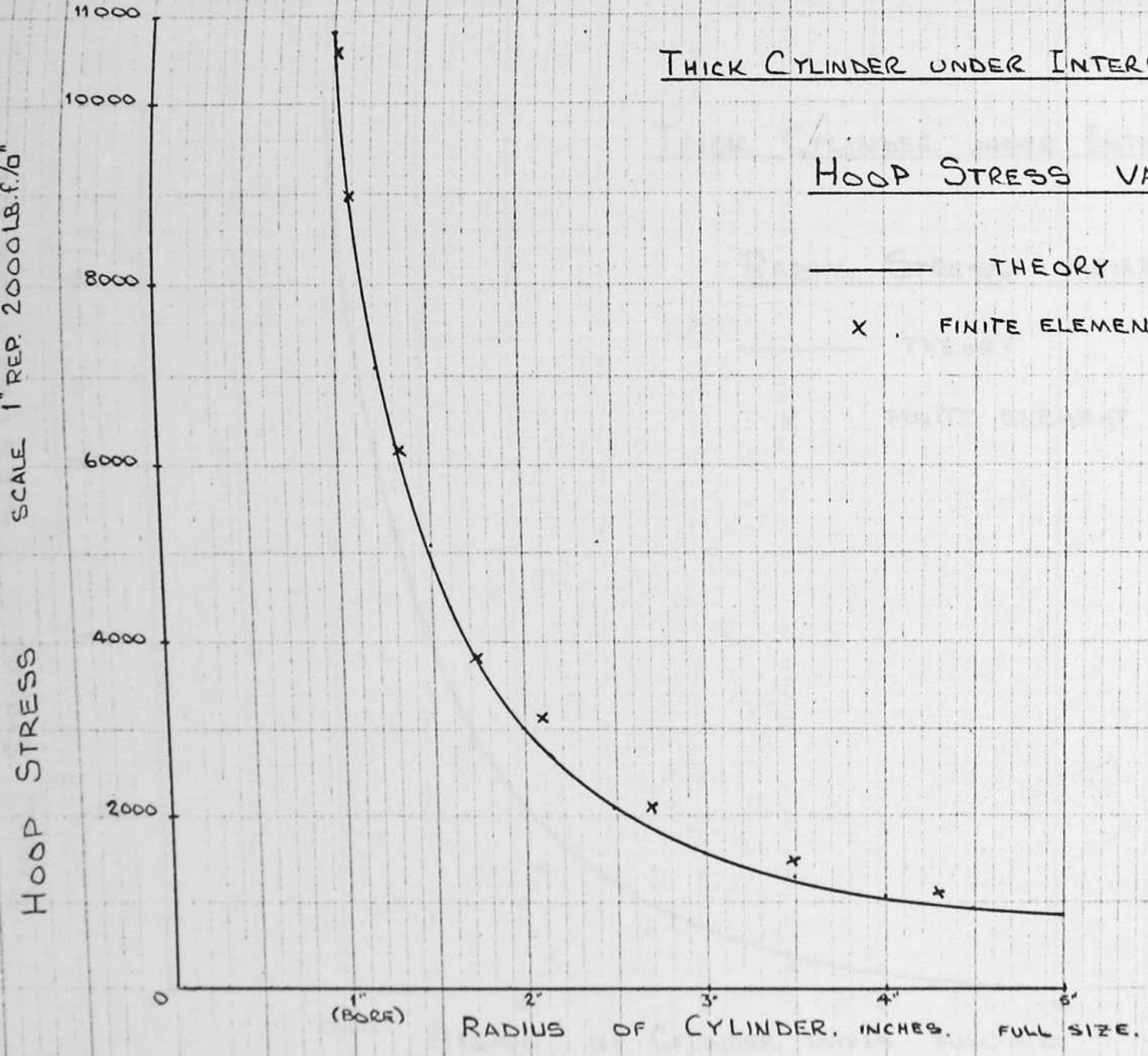


FIG. 31.

THICK CYLINDER UNDER INTERNAL PRESSURE OF 10,000 LBF/□"

RADIAL STRESS VARIATION ACROSS CYLINDER

———— THEORY

× FINITE ELEMENT STRESS

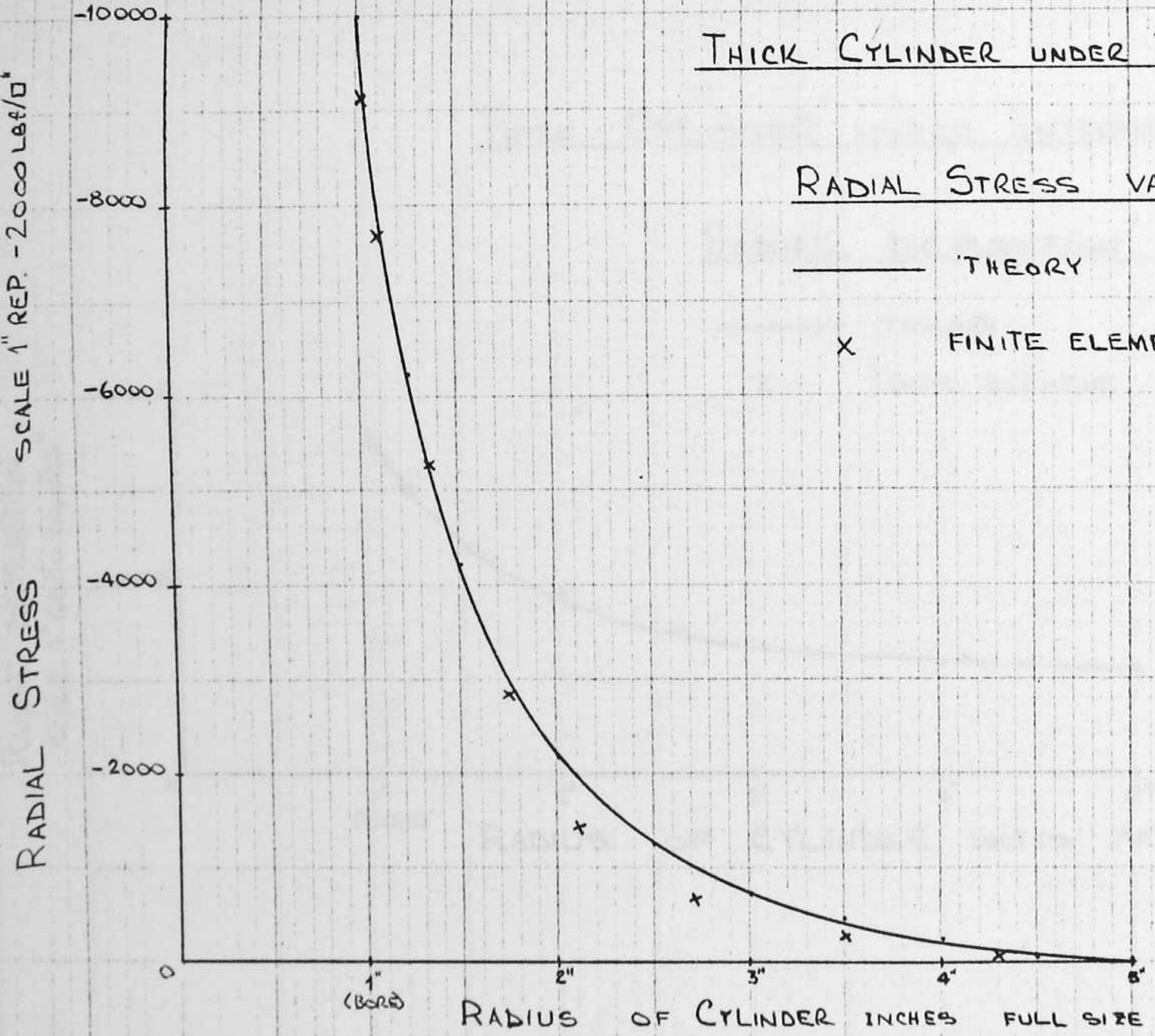


FIG. 32.

THICK CYLINDER UNDER INTERNAL PRESSURE OF 10,000 LBF/□"

RADIAL DEFLECTION ACROSS CYLINDER

——— THEORY

X FINITE ELEMENT, NODAL POINT DISPLACEMENTS.

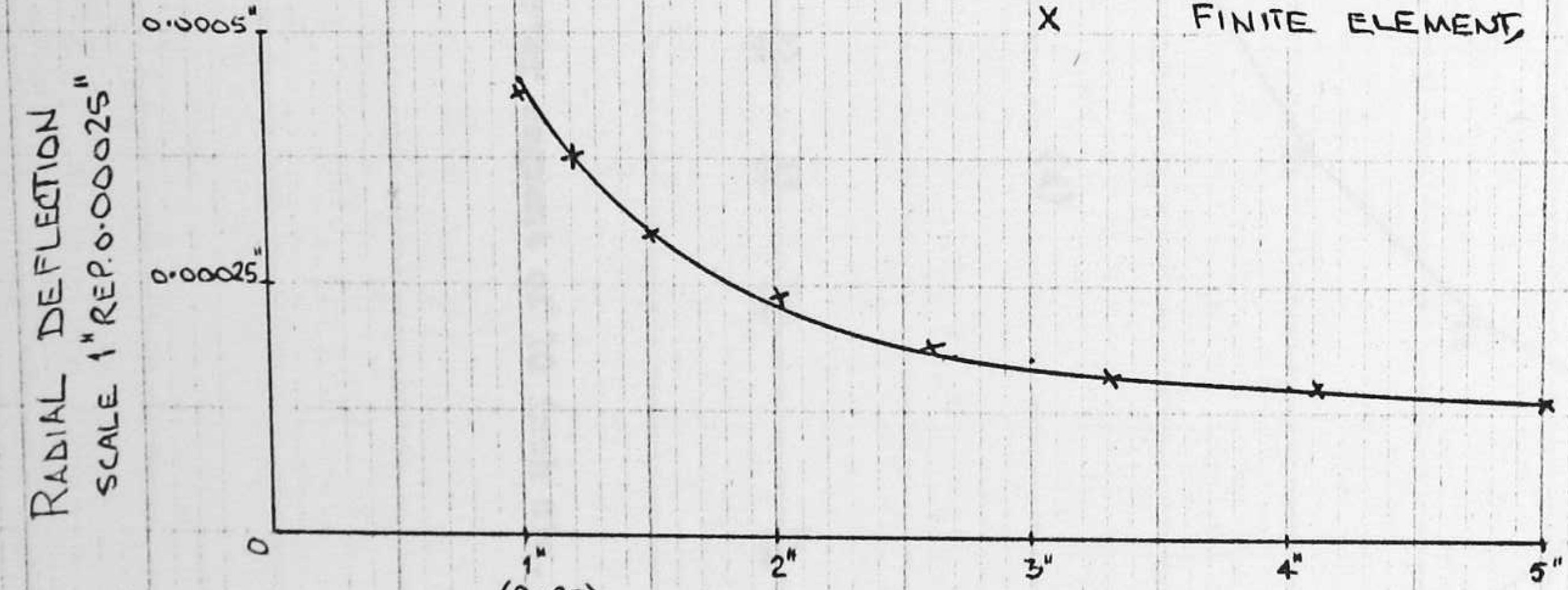


FIG. 33.

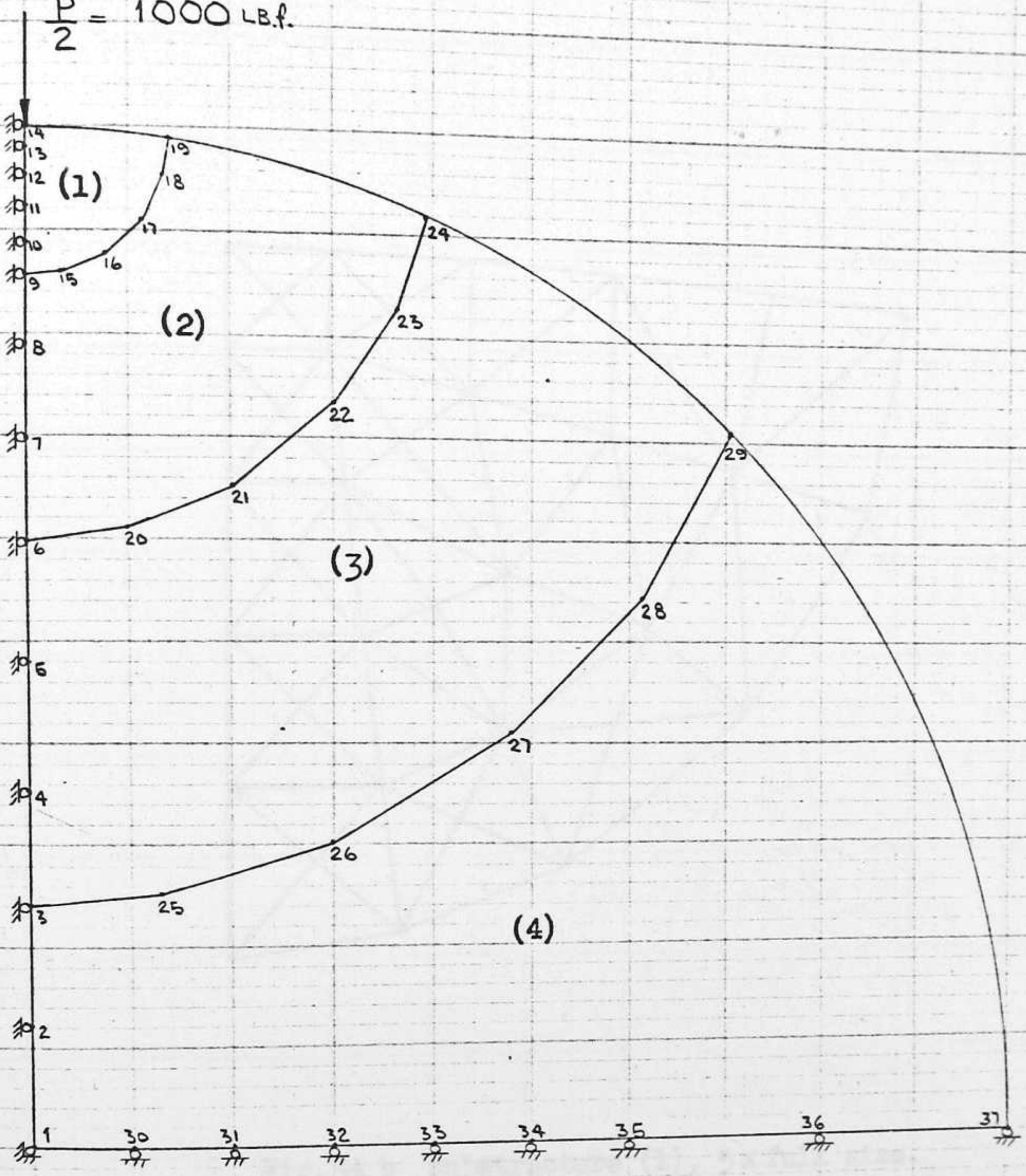


Fig.34 a full size quarter segment of 10 inch dia. Disc.

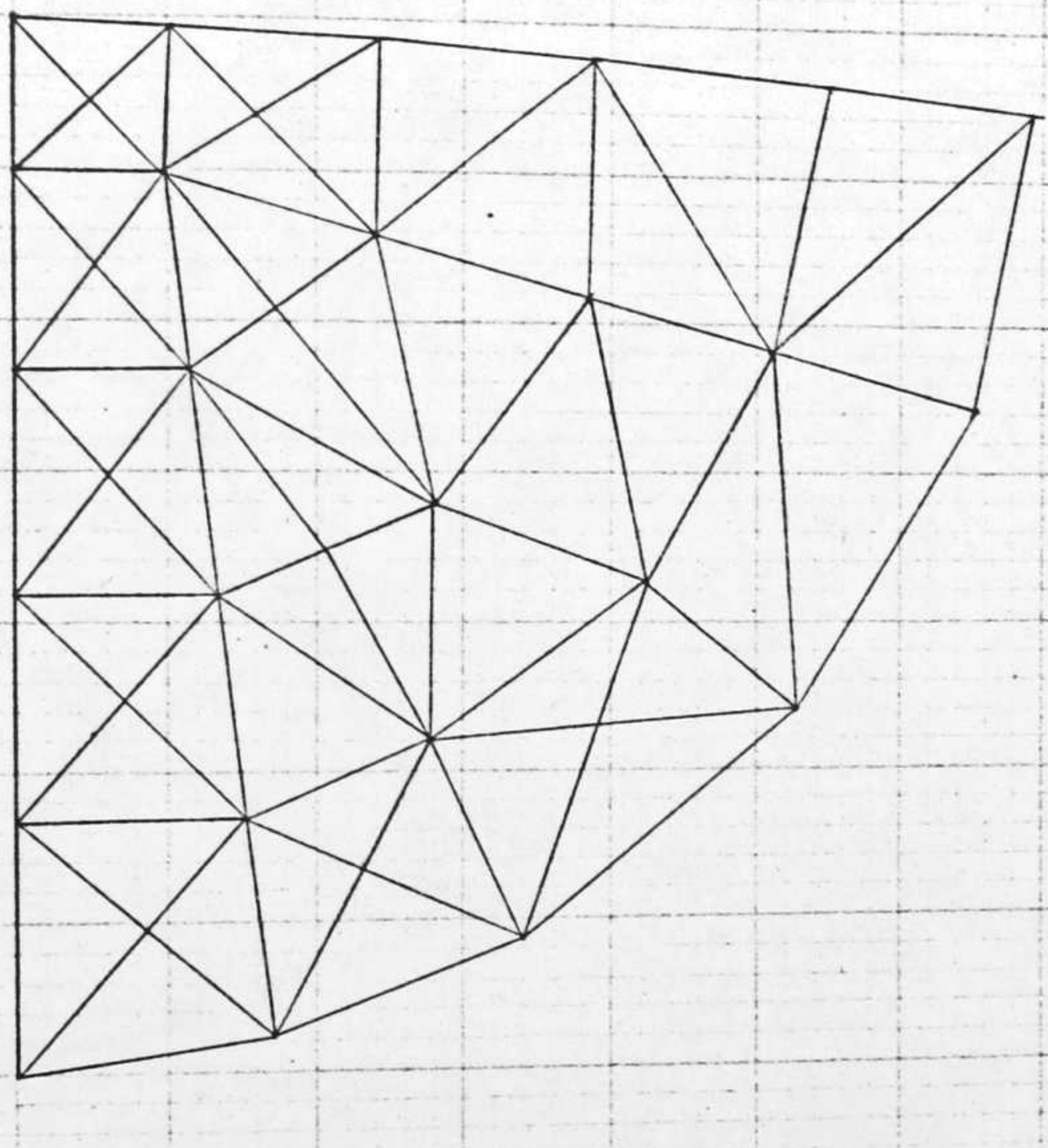


Fig. 34 b substructure (1), 5 × full size.

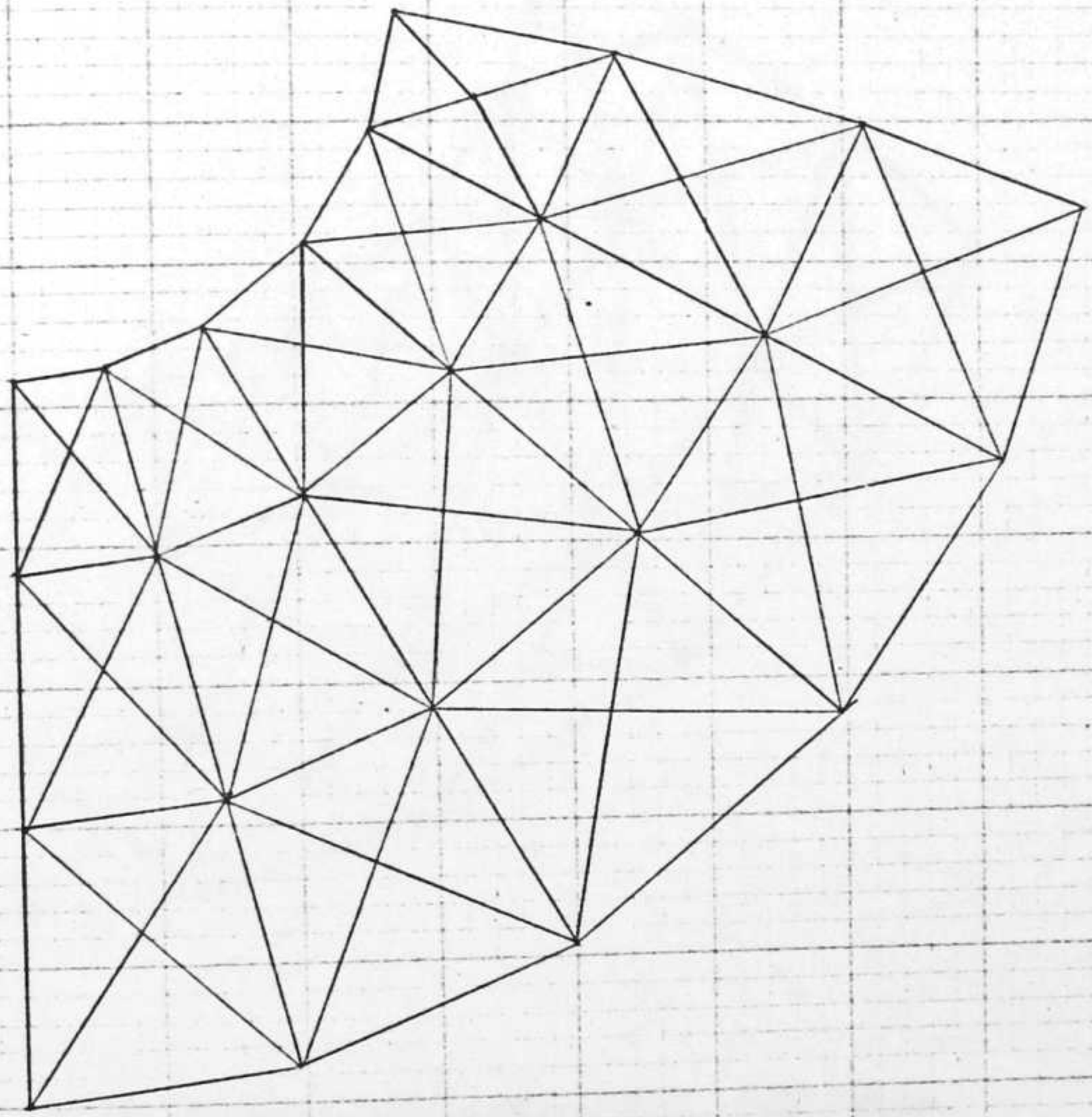


Fig.34 c substructure (2), 2×full size.

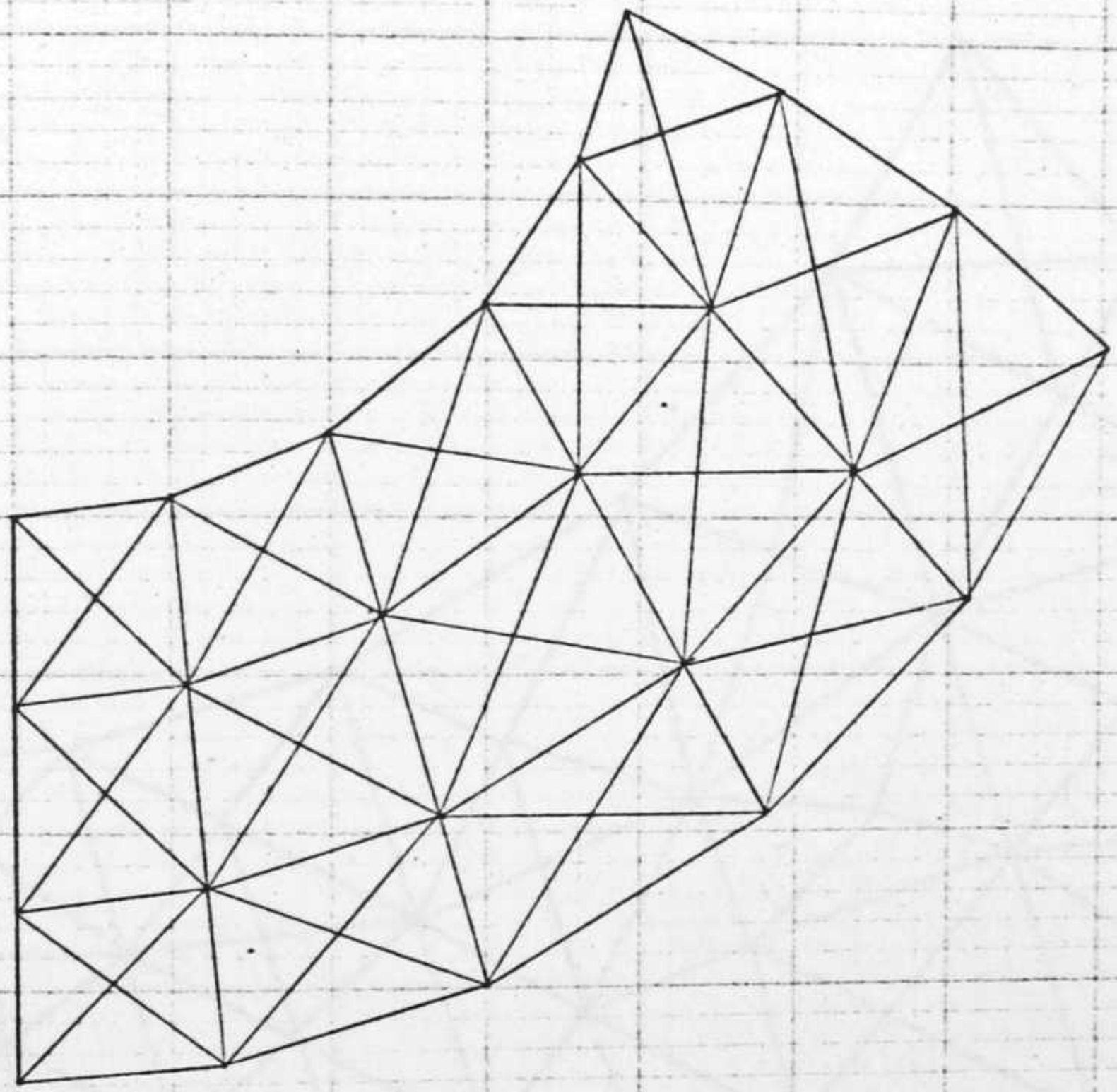


Fig.34 d substructure (3), full size.

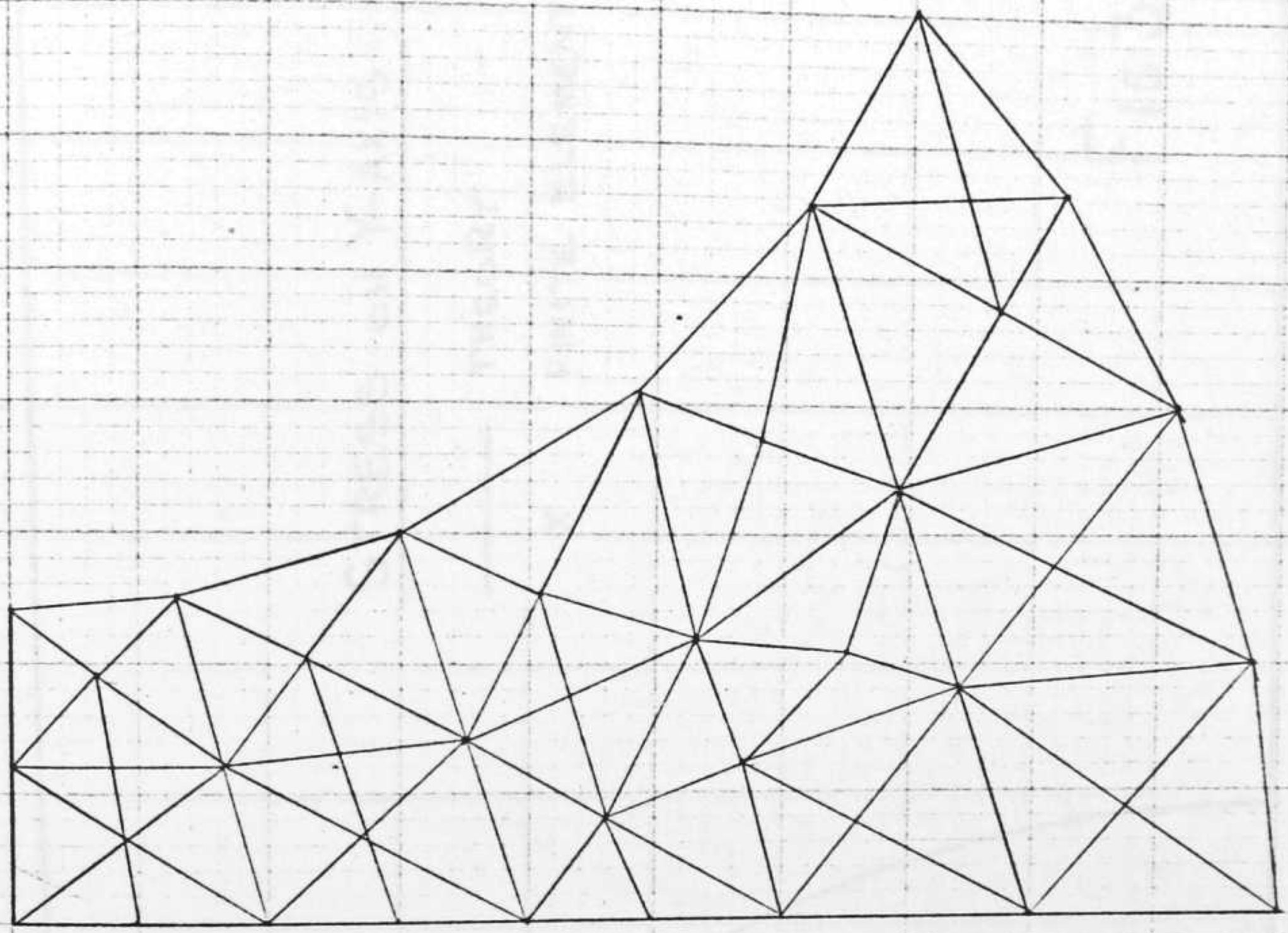
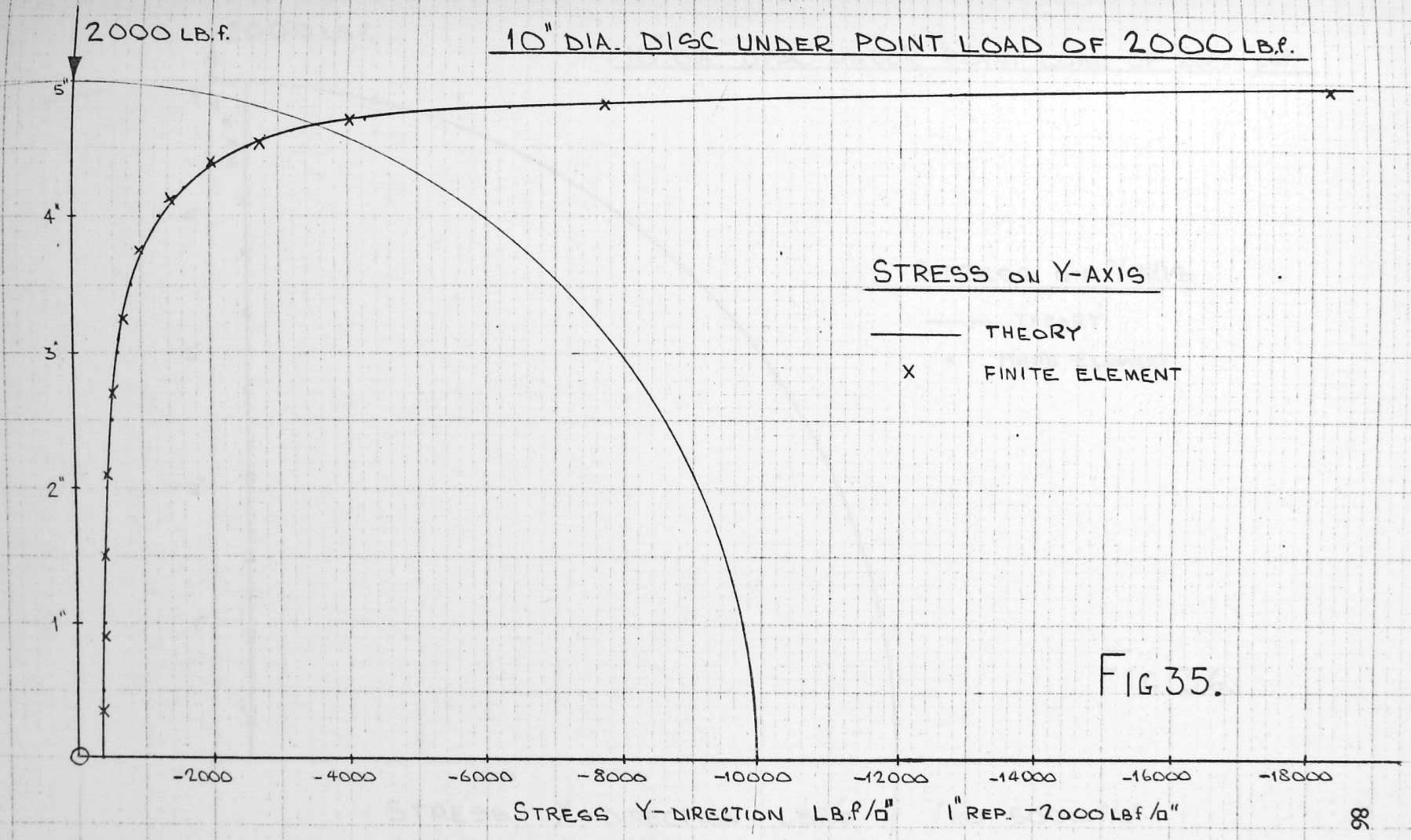


Fig.34 e substructure (4), full size.

10" DIA. DISC UNDER POINT LOAD OF 2000 LB.f.



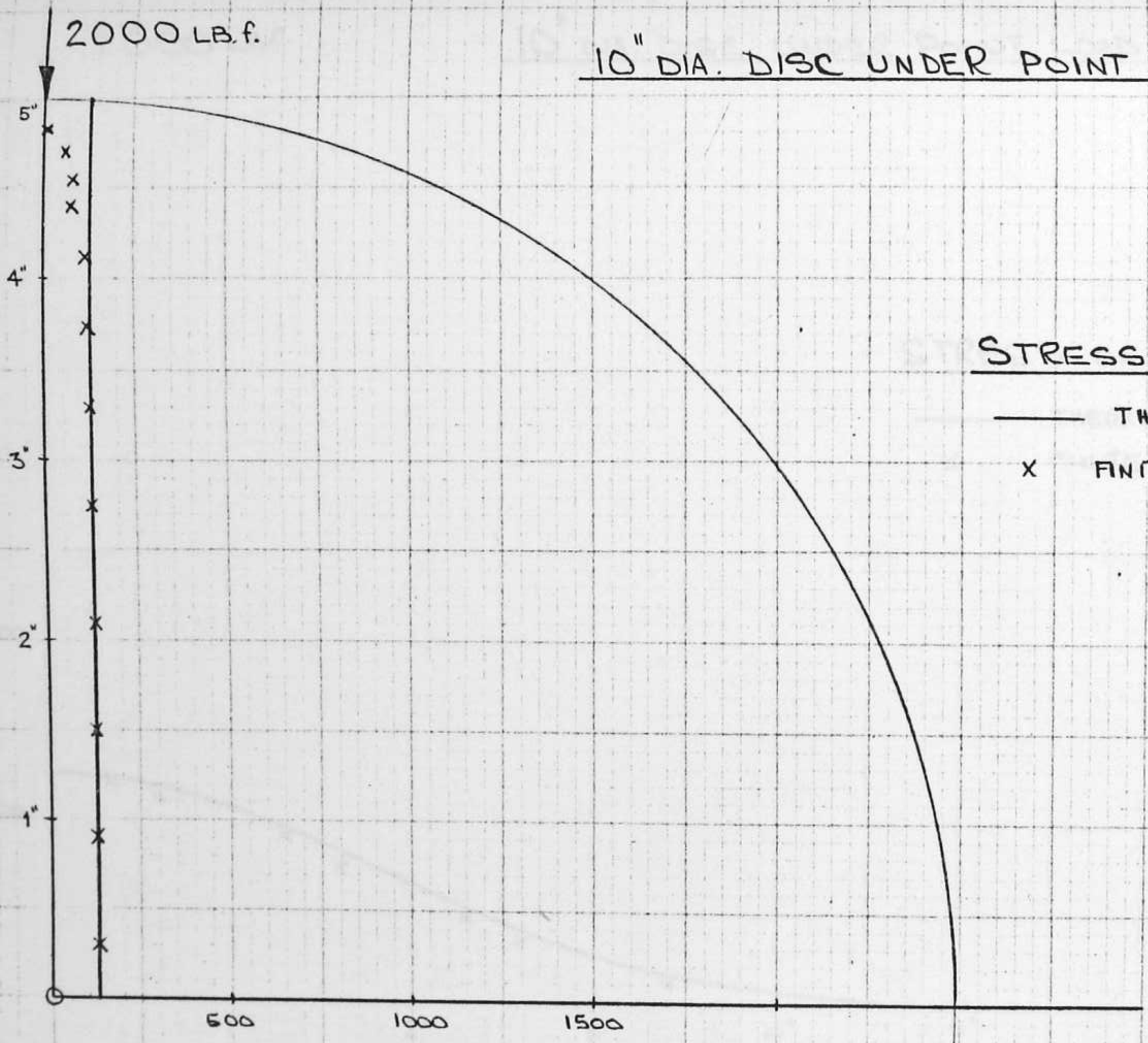
STRESS ON Y-AXIS

——— THEORY

X FINITE ELEMENT

FIG 35.

10" DIA. DISC UNDER POINT LOAD OF 2000 LBf.



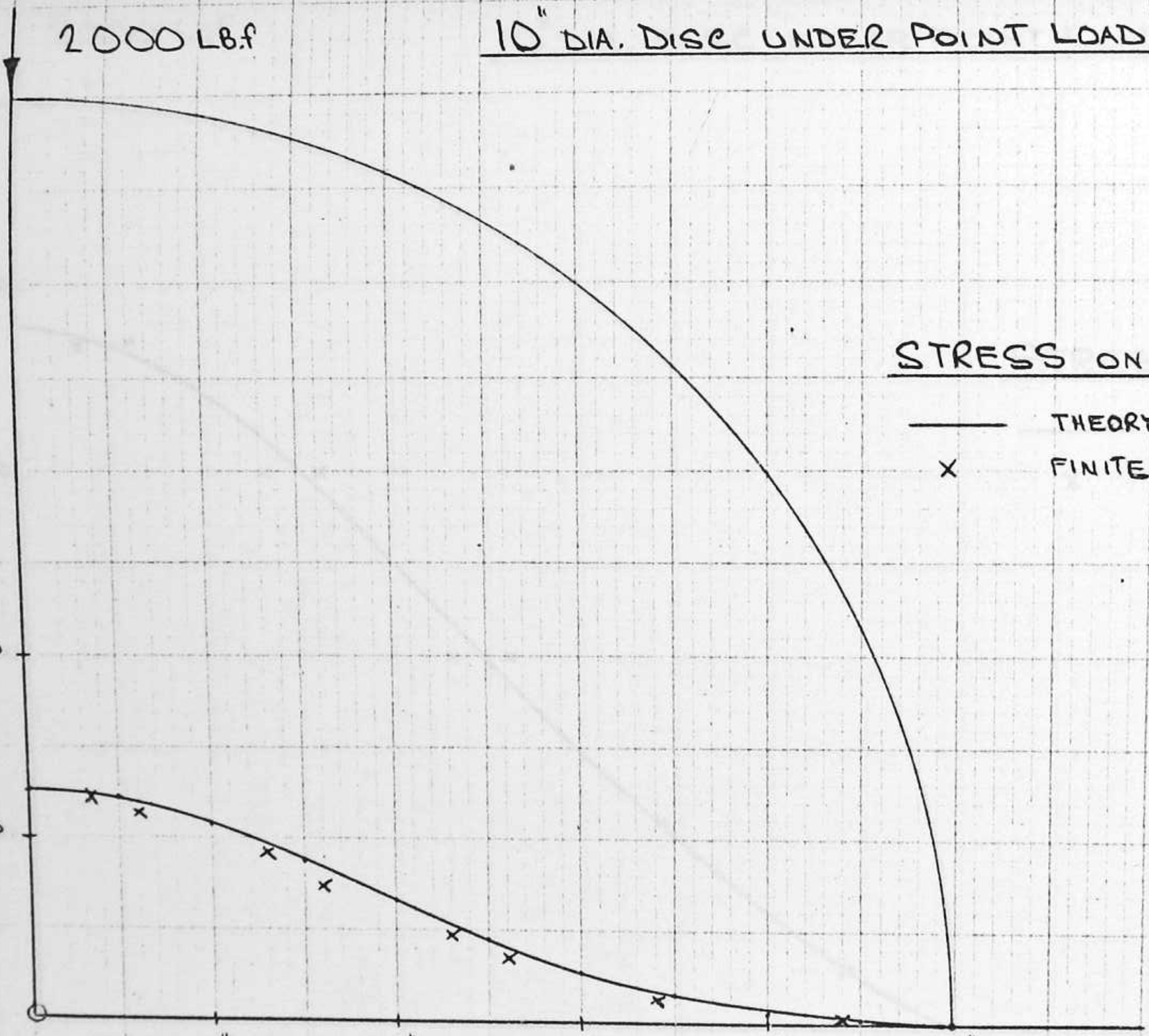
STRESS ON Y-AXIS

—— THEORY

X FINITE ELEMENT

FIG. 36.

10" DIA. DISC UNDER POINT LOAD OF 2000 LBf.



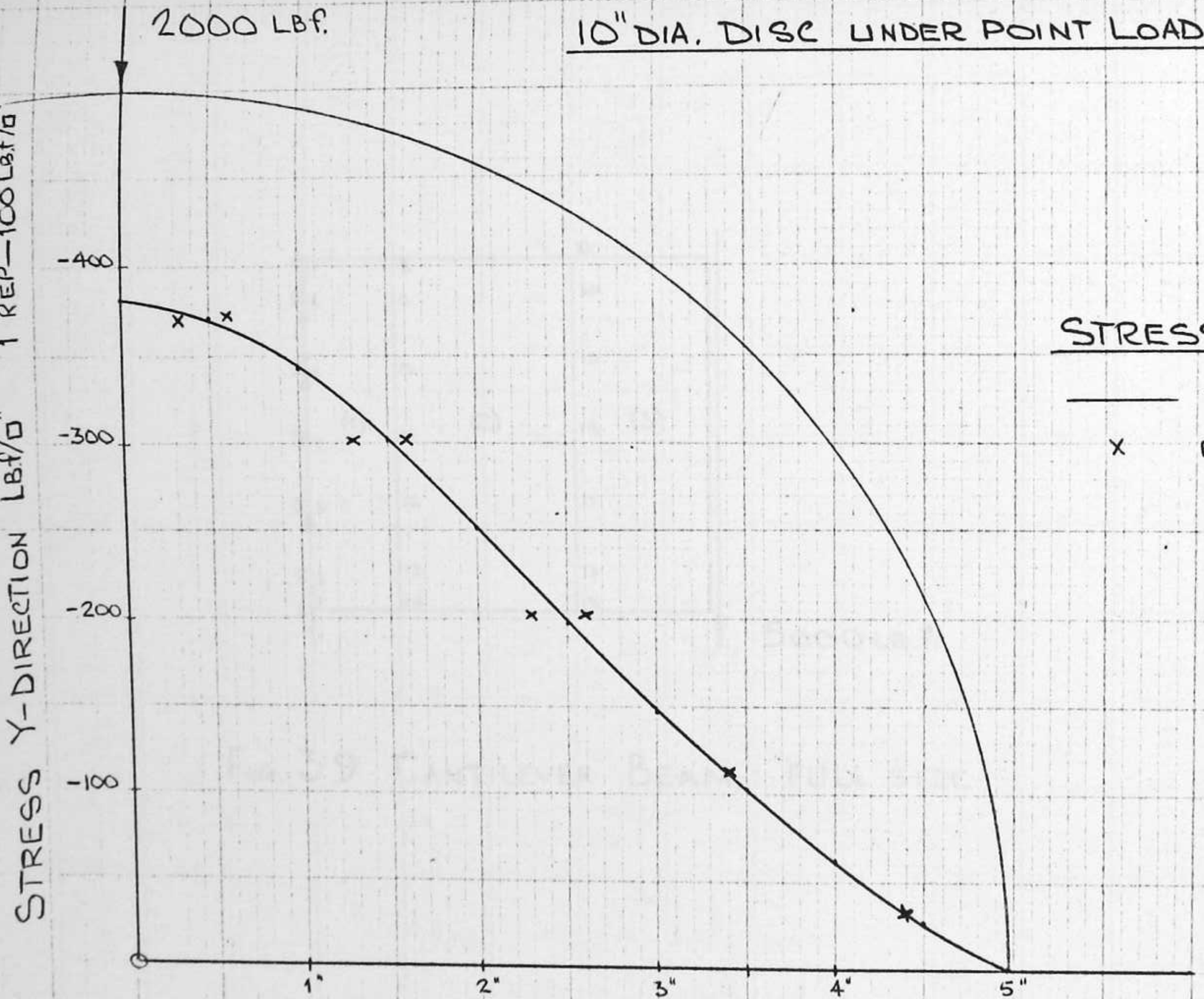
STRESS ON X-AXIS

——— THEORY

X FINITE ELEMENT

FIG. 37.

10" DIA. DISC UNDER POINT LOAD OF 2000 Lbf.



STRESS ON X-AXIS

——— THEORY

X FINITE ELEMENT

FIG. 38.

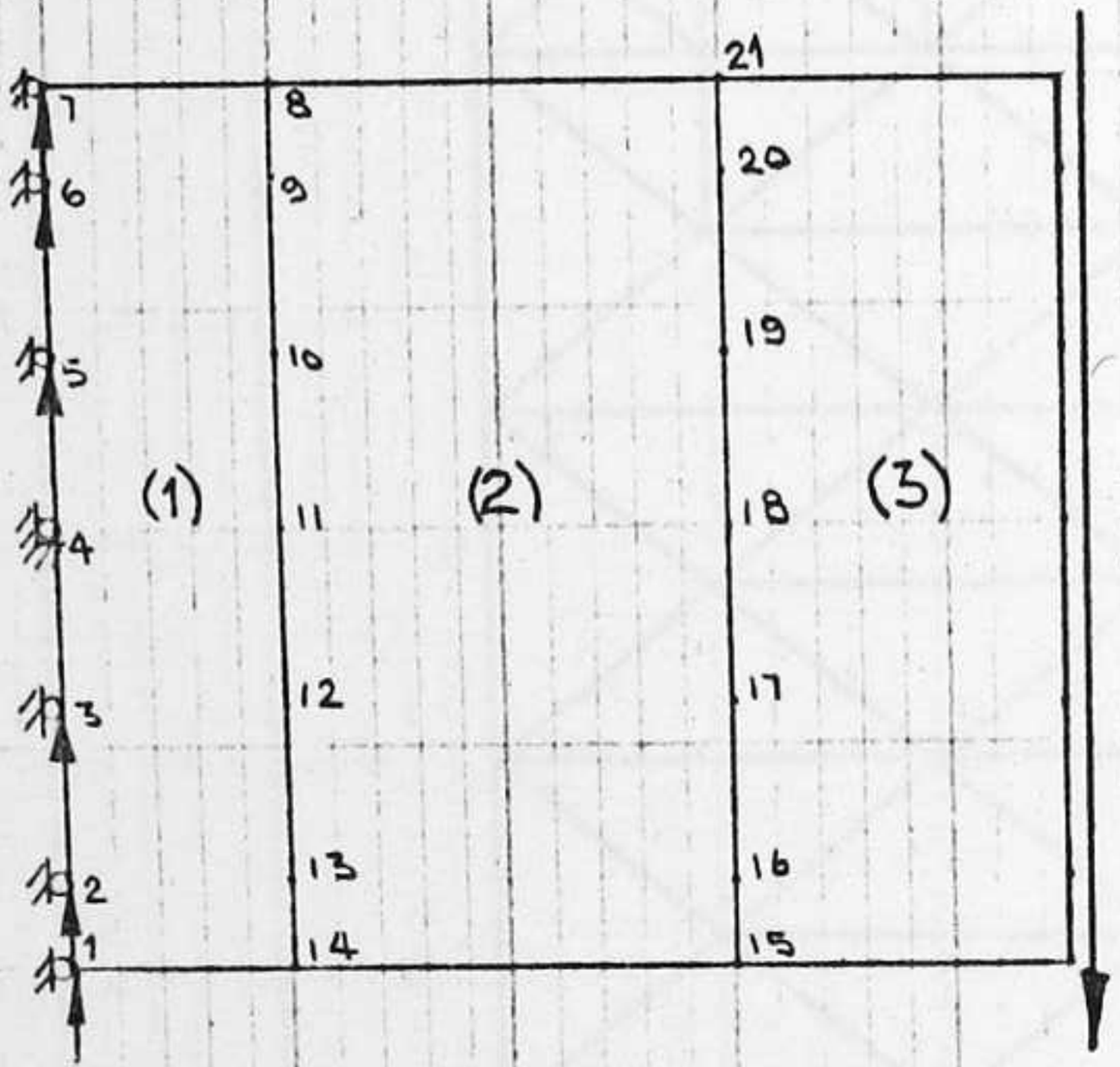
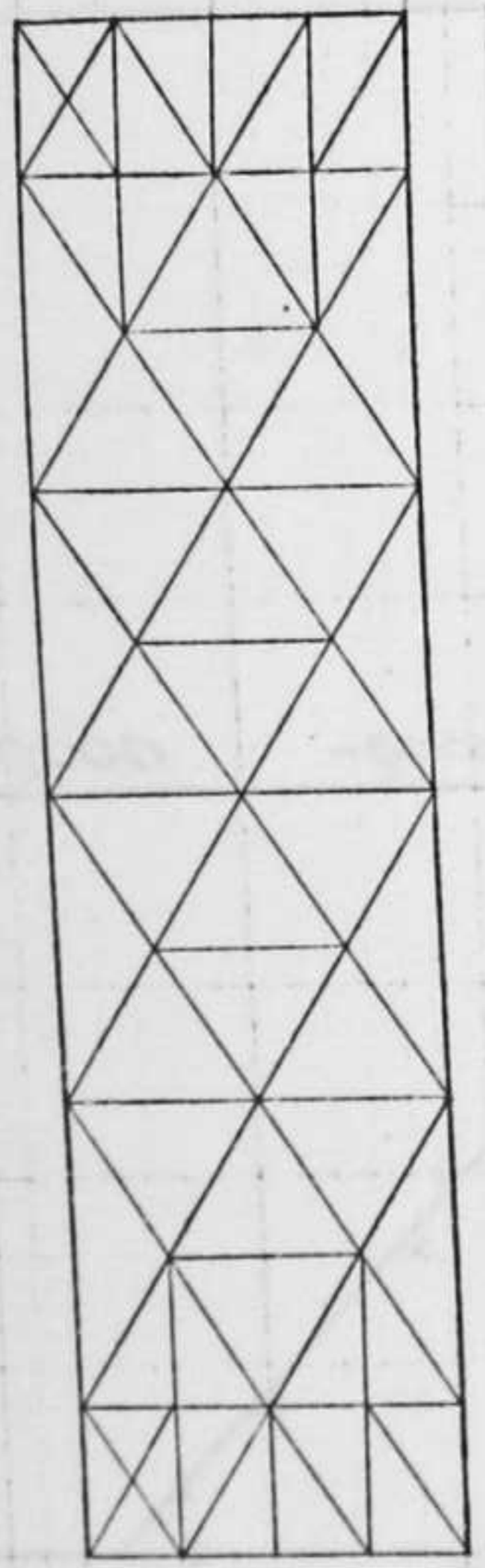
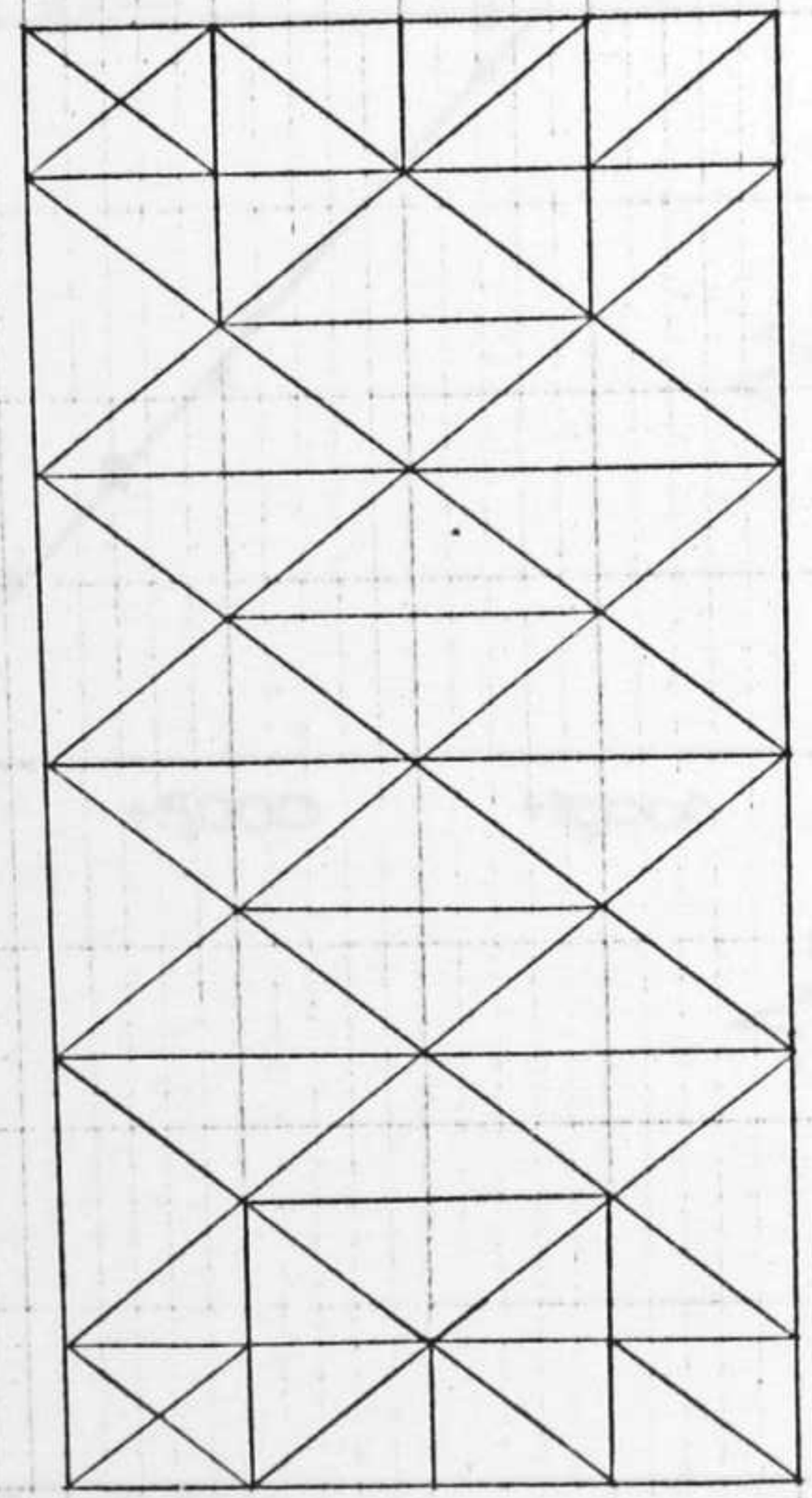


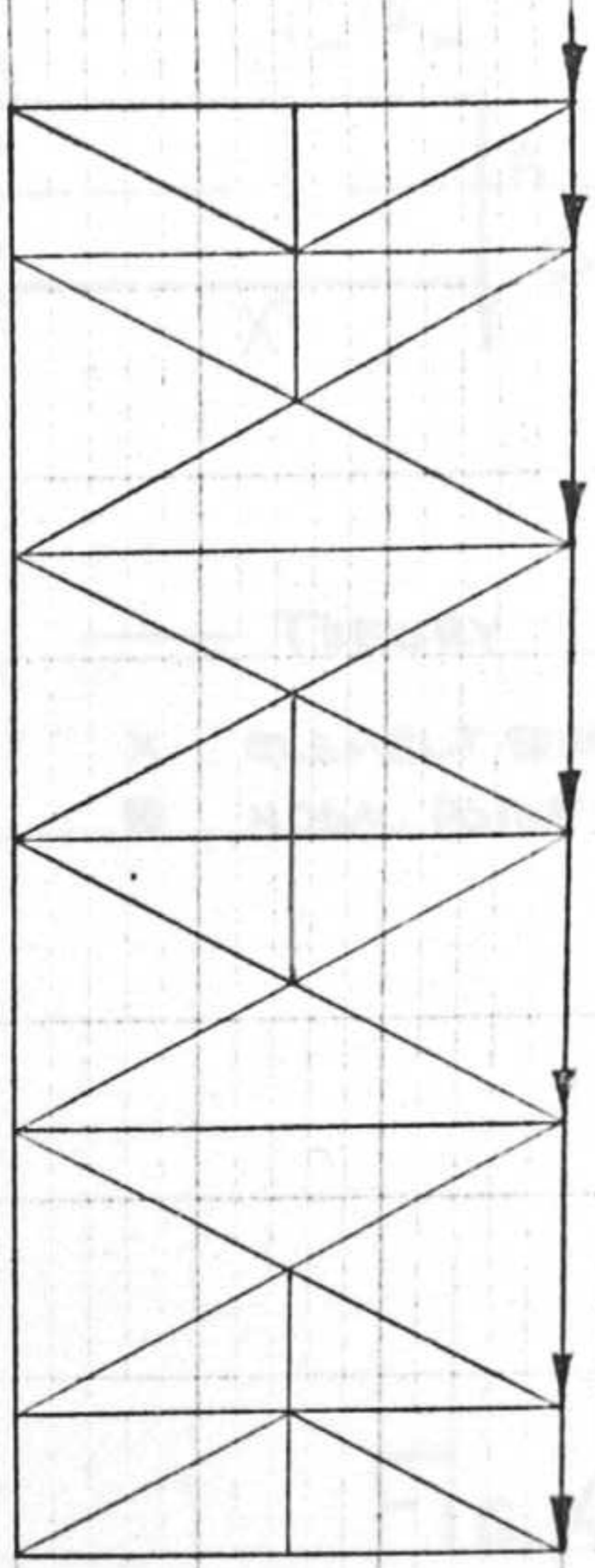
FIG. 39 CANTILEVER BEAM, FULL SIZE.



SUBSTRUCTURE (1)



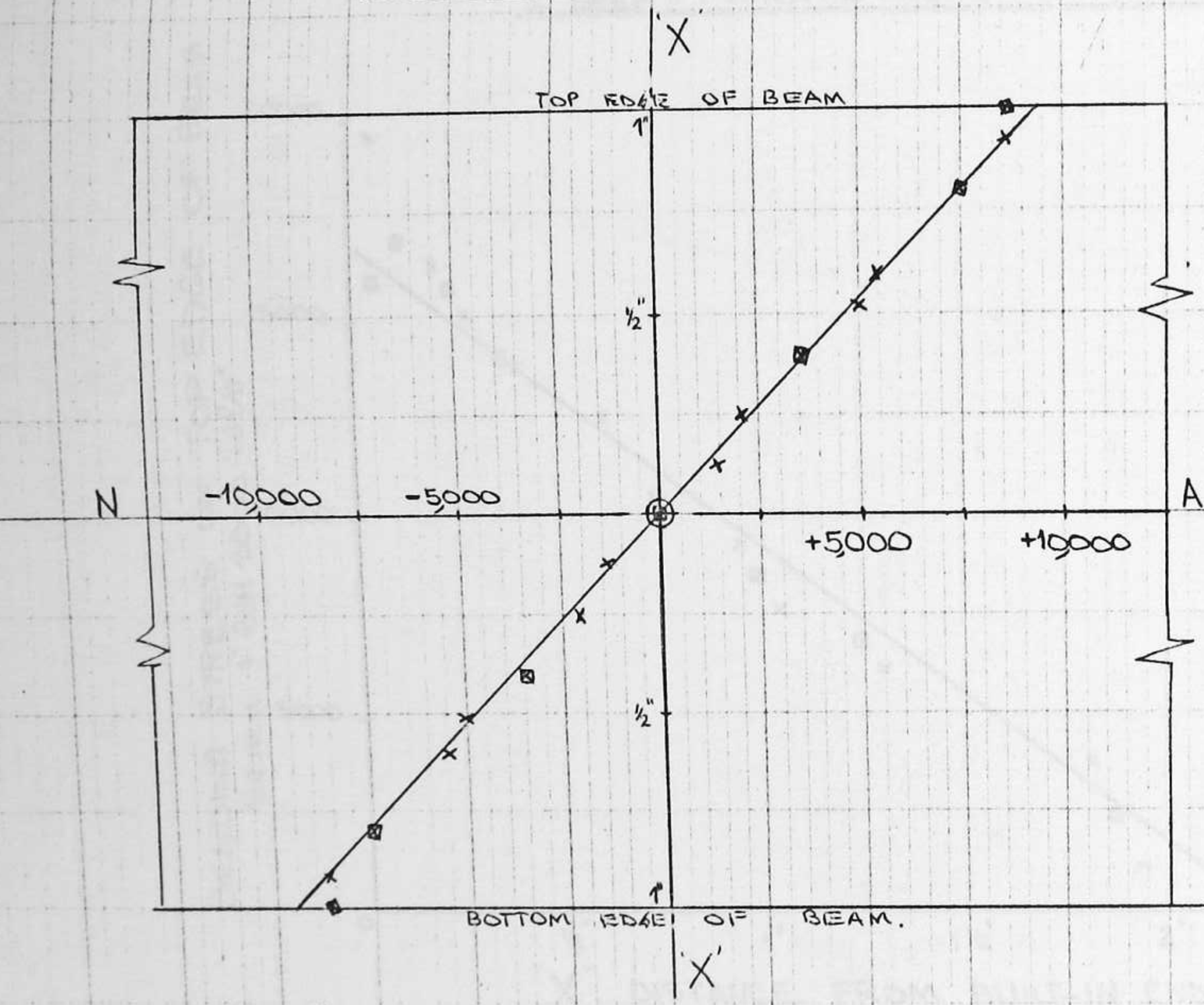
SUBSTRUCTURE (2)

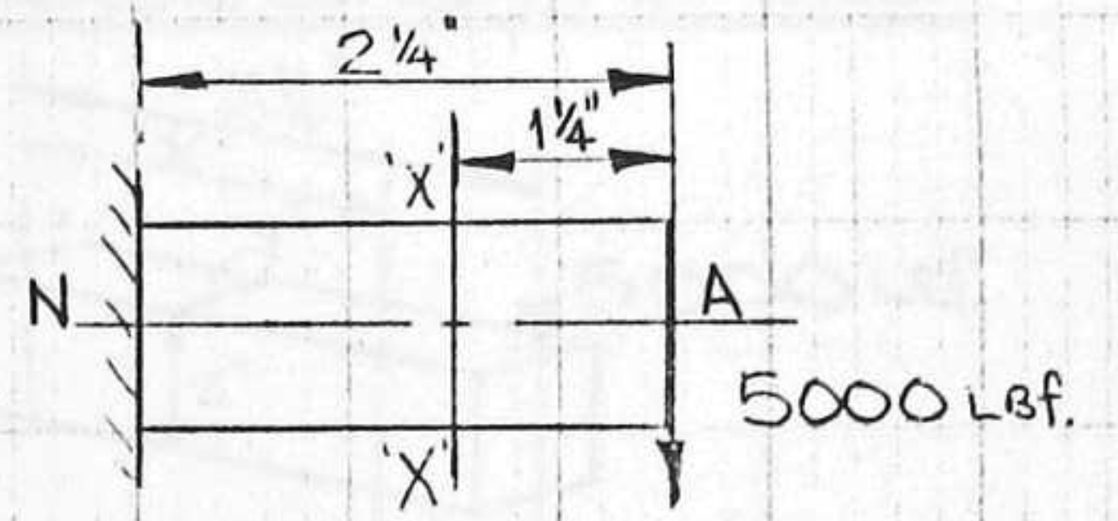


SUBSTRUCTURE (3)

FIG. 40 CANTILEVER BEAM, 2× FULL SIZE.

2" DEEP x 1" THICK x 2¼" LONG CANTILEVER BEAM WITH 5000 LBf. LOAD AT END.





—— THEORY

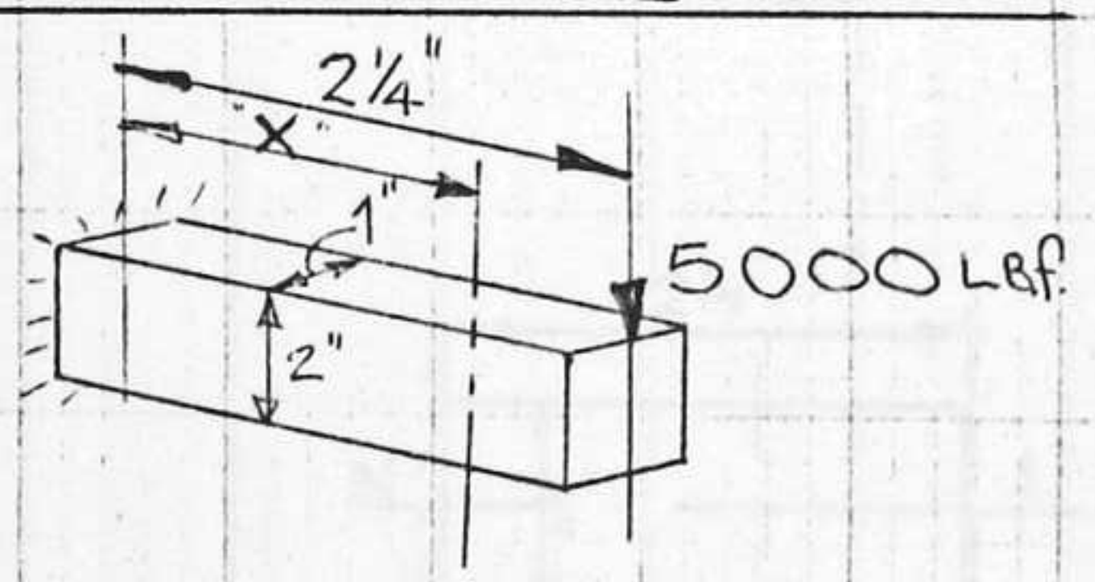
× ELEMENT STRESS } FINITE ELEMENT

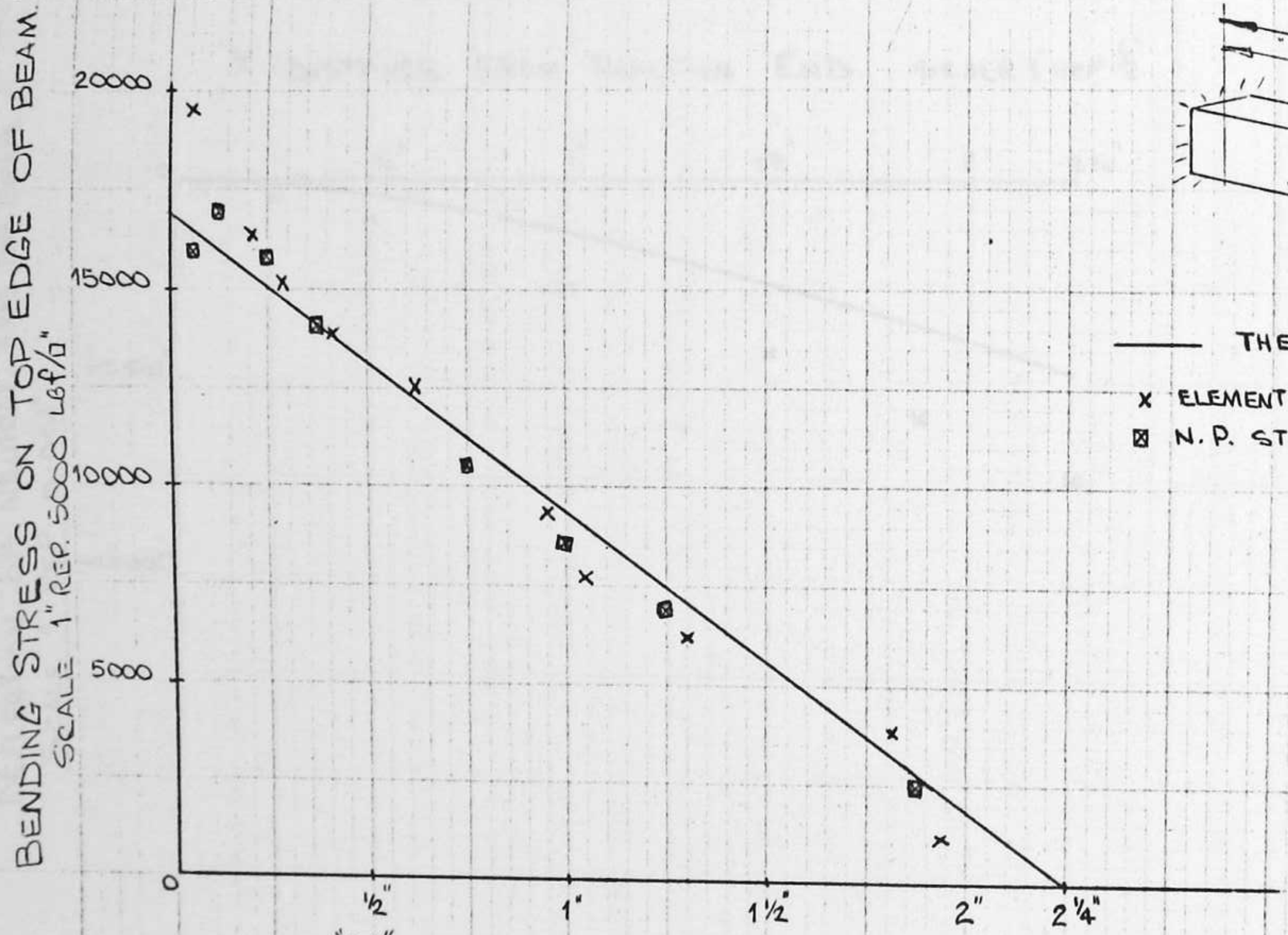
⊠ NODAL POINT STRESS } FINITE ELEMENT

FIG. 41.

BENDING STRESS DISTRUBUTION ACROSS DEPTH OF BEAM. LBf./□" SCALE 1" REP. 5000 LBf/□"

2" DEEP × 1" THICK × 2¼" LONG CANTILEVER WITH 5000 LBf LOAD AT END.





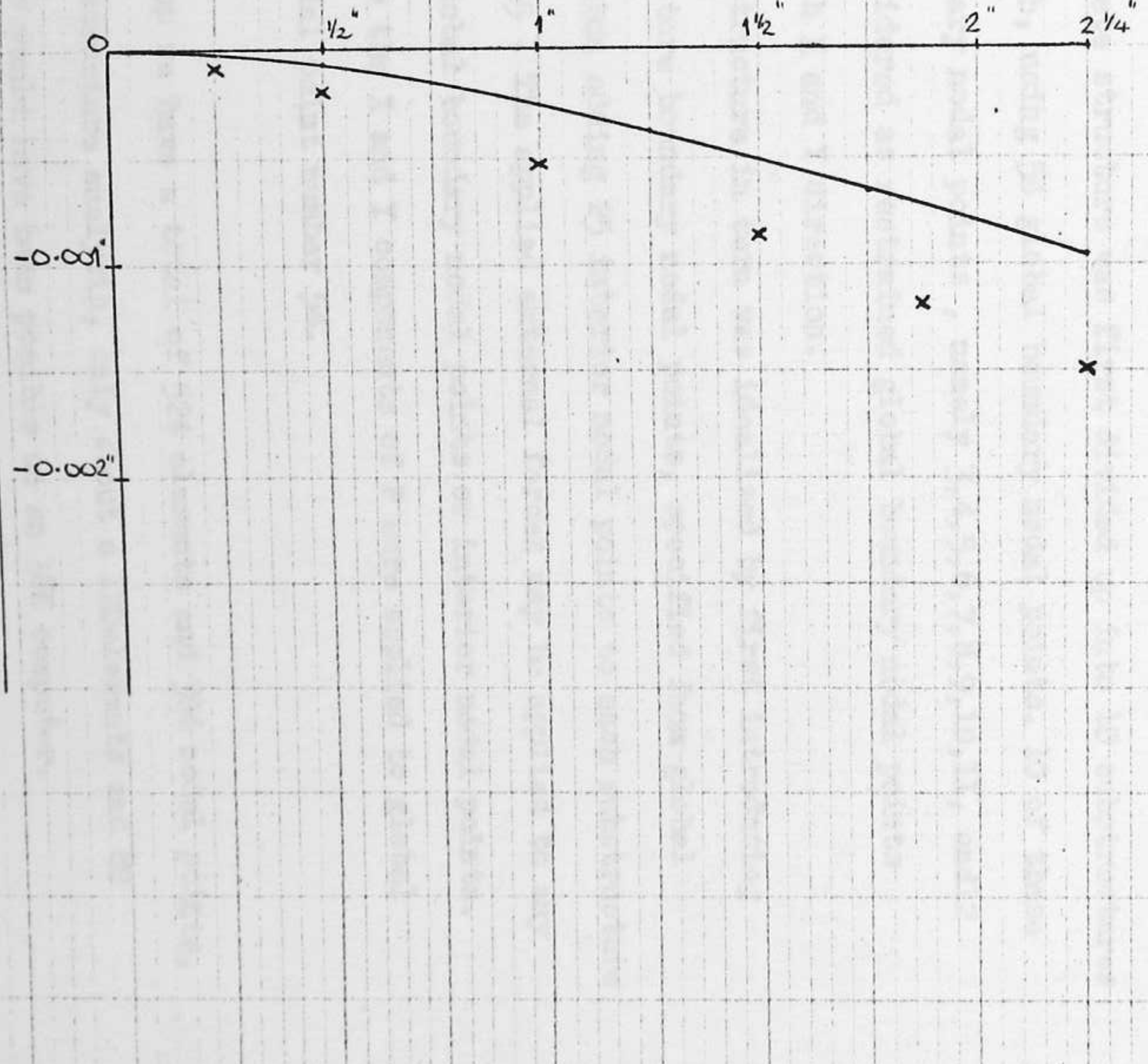
— THEORY

× ELEMENT STRESS
⊠ N. P. STRESS
} FINITE ELEMENT.

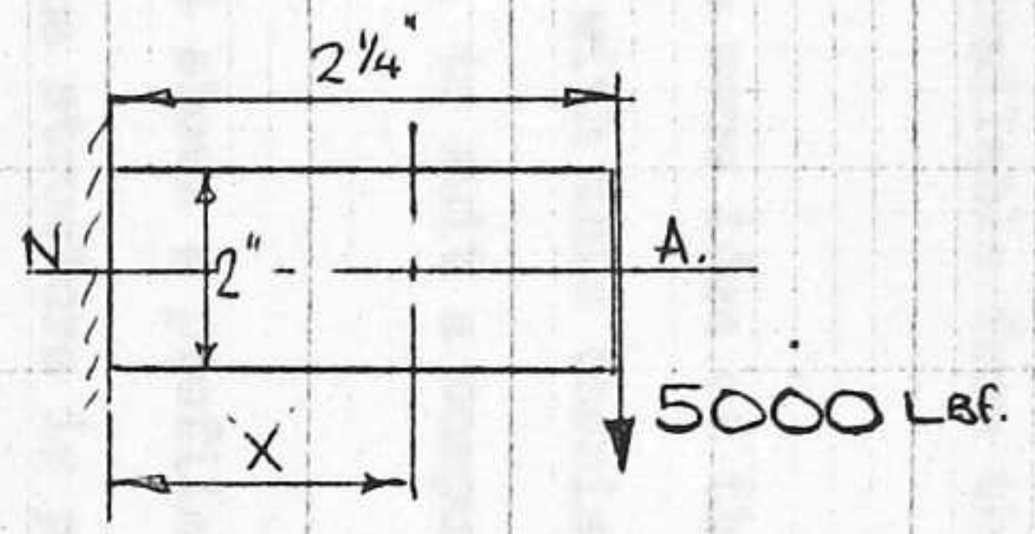
FIG. 42.

2" DEEP × 1" THICK × 2¼" LONG CANTILEVER BEAM WITH 5000 LBf. LOAD AT END.

'X' DISTANCE FROM BUILT-IN END. SCALE 1" REP ½"



DEFLECTION OF NEUTRAL AXIS OF BEAM
SCALE 1" REP 0·001"



—— THEORY

× NODAL POINT DISPLACEMENTS.

FIG. 43.

2.9 Scope Available from the programs presented with this Thesis.

To give some idea of the scope available from the programs presented with this thesis, the following structure has been idealized, to suit a computer having at least 18K storage capacity.

Fig. 44 shows a profile of "NOVIKOV" gear teeth under a point load P. The structure is prevented from rotation by rigidly fixing the hub. It was expected that high stress consentration would occur in the proximity of the tooth root fillets and at the point of load contact.

The complete structure was first divided up into 10 substructures fig.45a,and45b, using 54 global boundary nodal points. 10 of these global boundary nodal points , namely 3,4,5,6,7,8,9,10,11, and12 will be considered as restrained global boundary nodal points fixed in both X and Y direction.

Each substructure in turn was idealized by first introducing the substructure boundary nodal points, specified FROM global numbering, then adding 25 interior nodal points to each substructure fig. 46 to 55 . The applied external forces may be applied to any number of global boundary nodal points or interior nodal points. In this case the X and Y components of P were applied to global boundary nodal point number 54.

Summing up we have a total of 524 elements and 304 nodal points. Without substructure analysis, only about a 120elements and 80 nodal points would have been posible on an 18K computer.

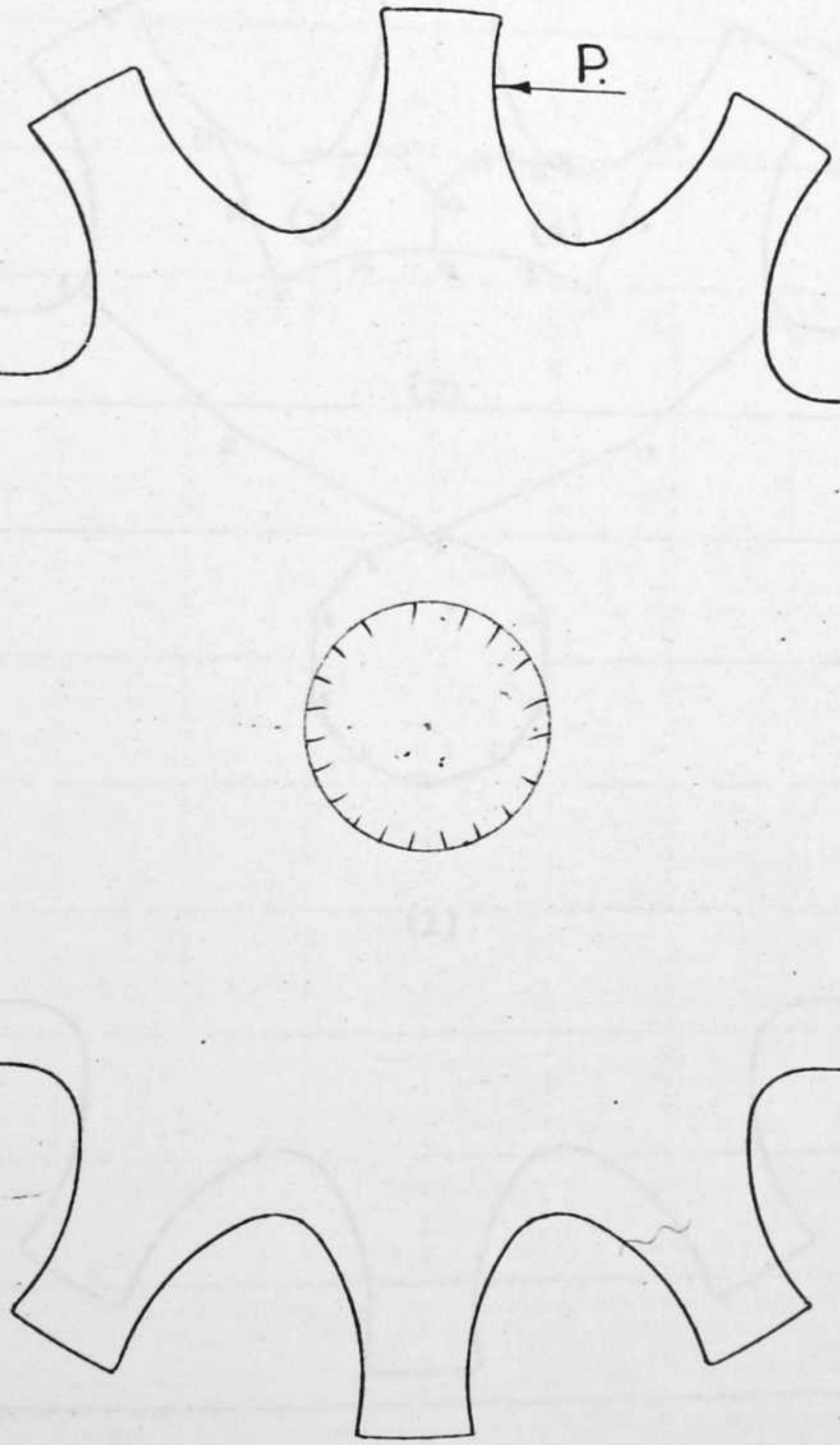


Fig. 44 profile of "NOVIKOV" gear.

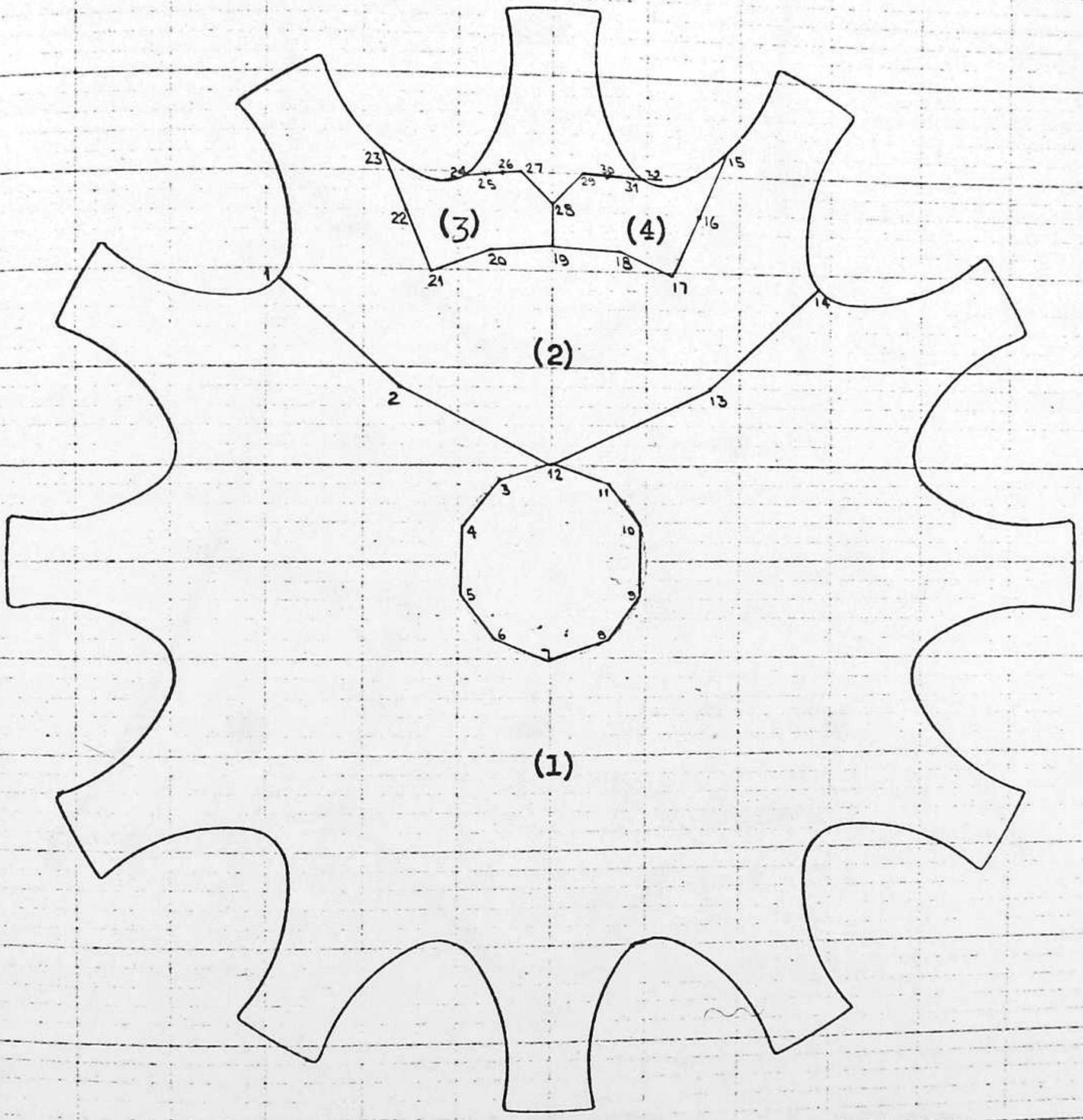


Fig. 45a global numbering of "NOVIKOV" gear; full size.

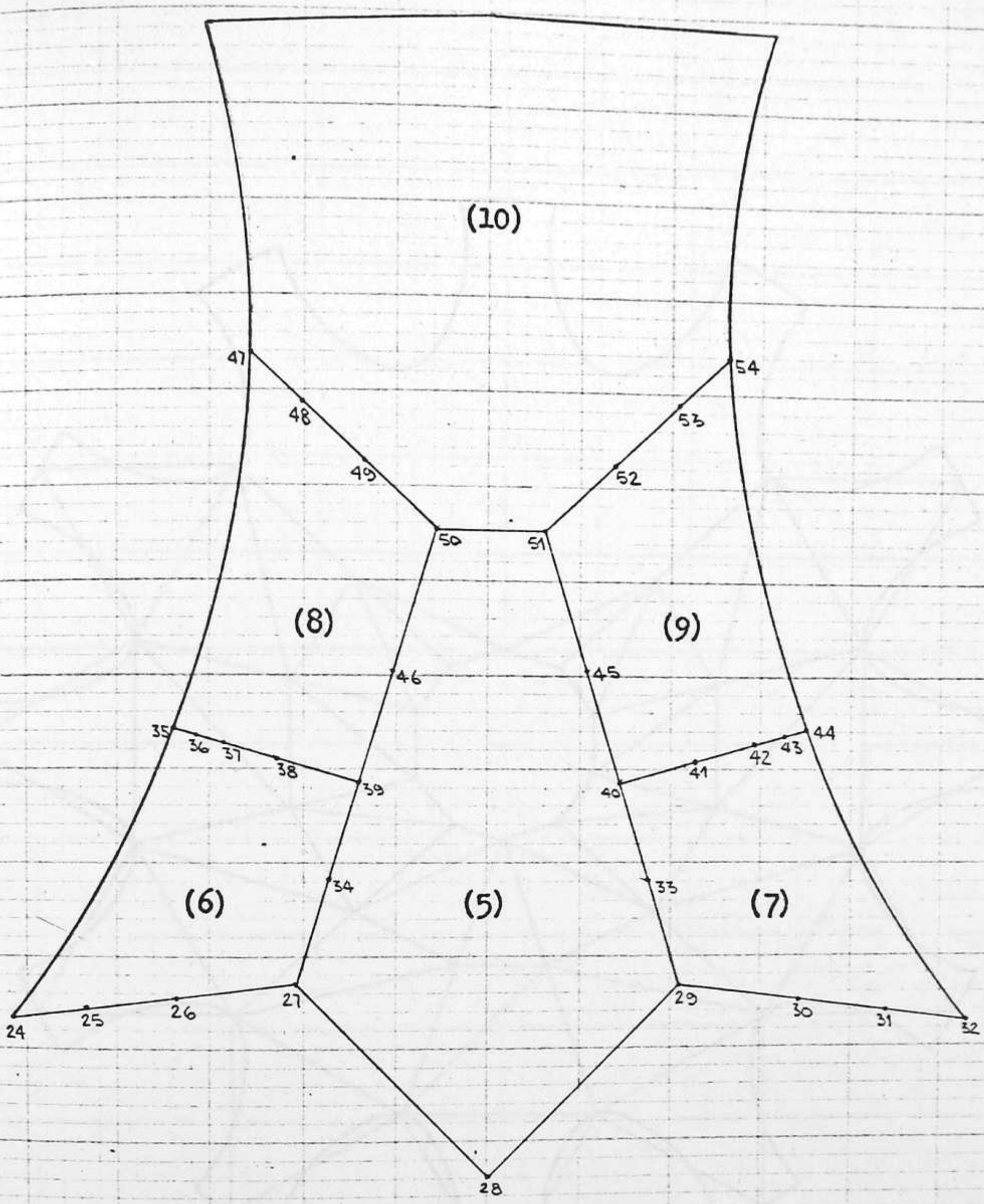


Fig.45 b global numbering of "NOVIKOV" gear; 6 × full size.

Number of substructures = 10.

Number of global boundary nodal points = 54.

Number of restrained boundary nodal points = 10.

Restrained boundary nodal points are 3,4,5,6,7,8,9,10,11,12.

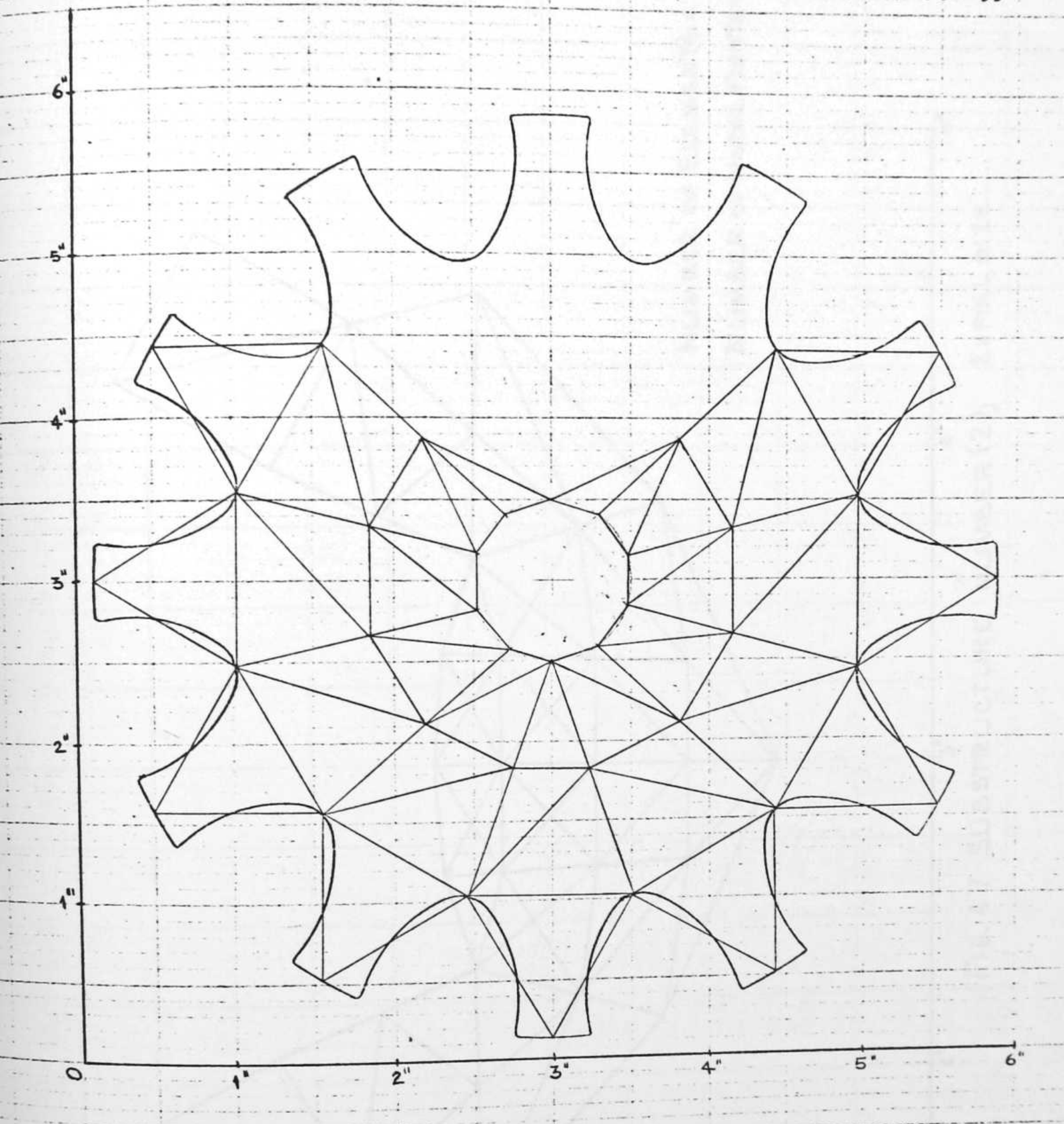
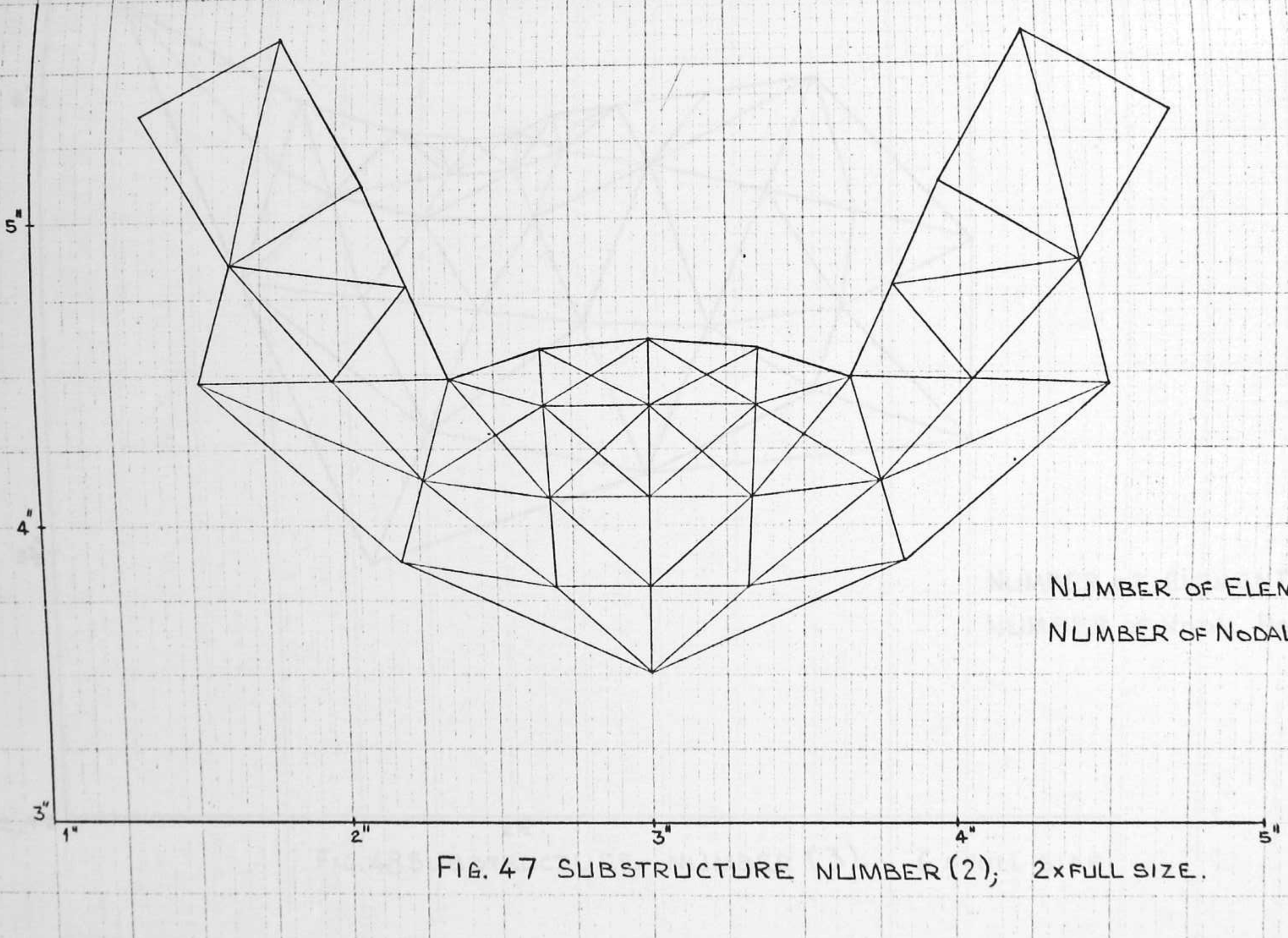


Fig. 46 substructure number (1); full size.

Number of elements = 46.

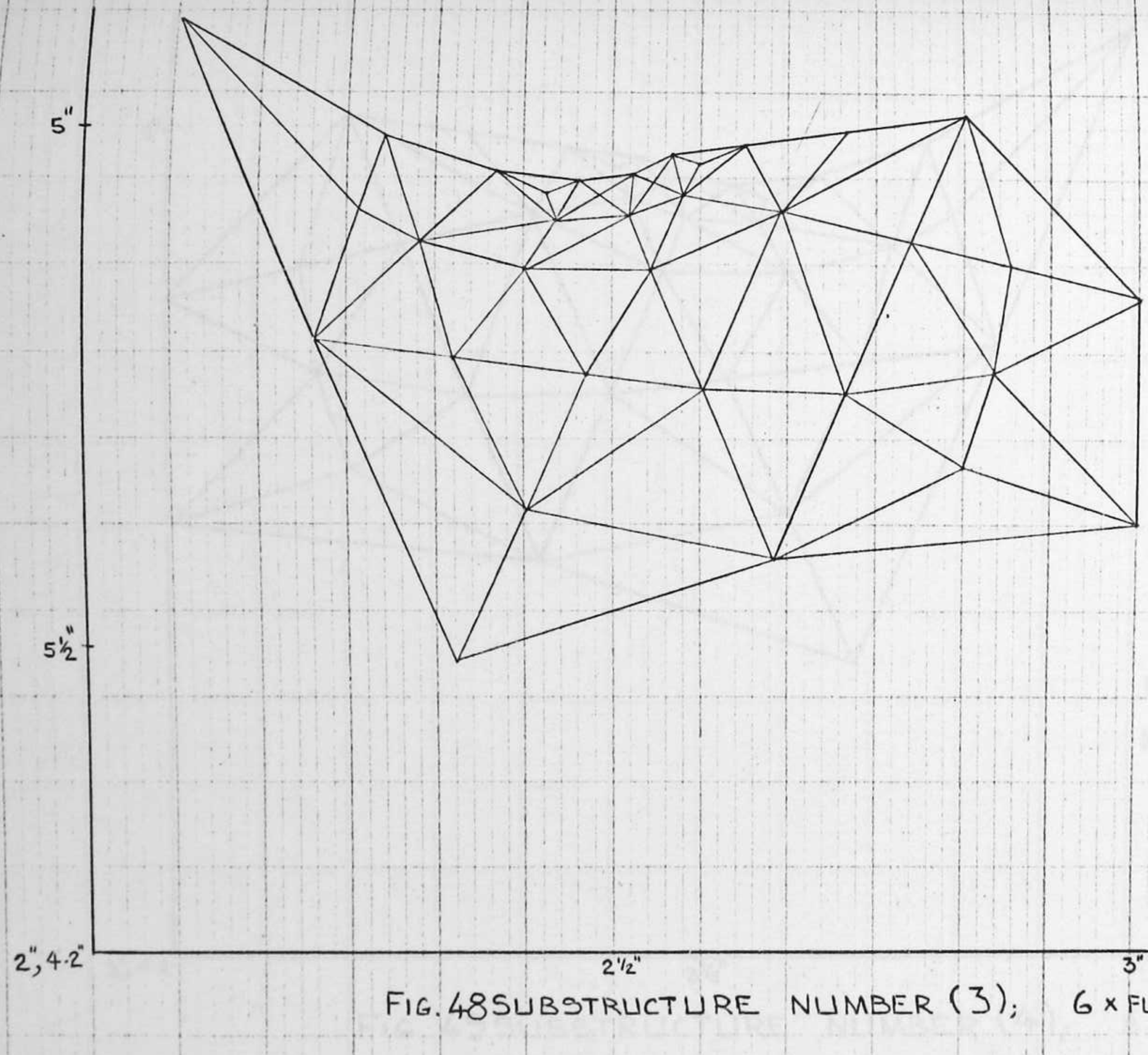
Number of nodal points = 39.



NUMBER OF ELEMENTS = 56.

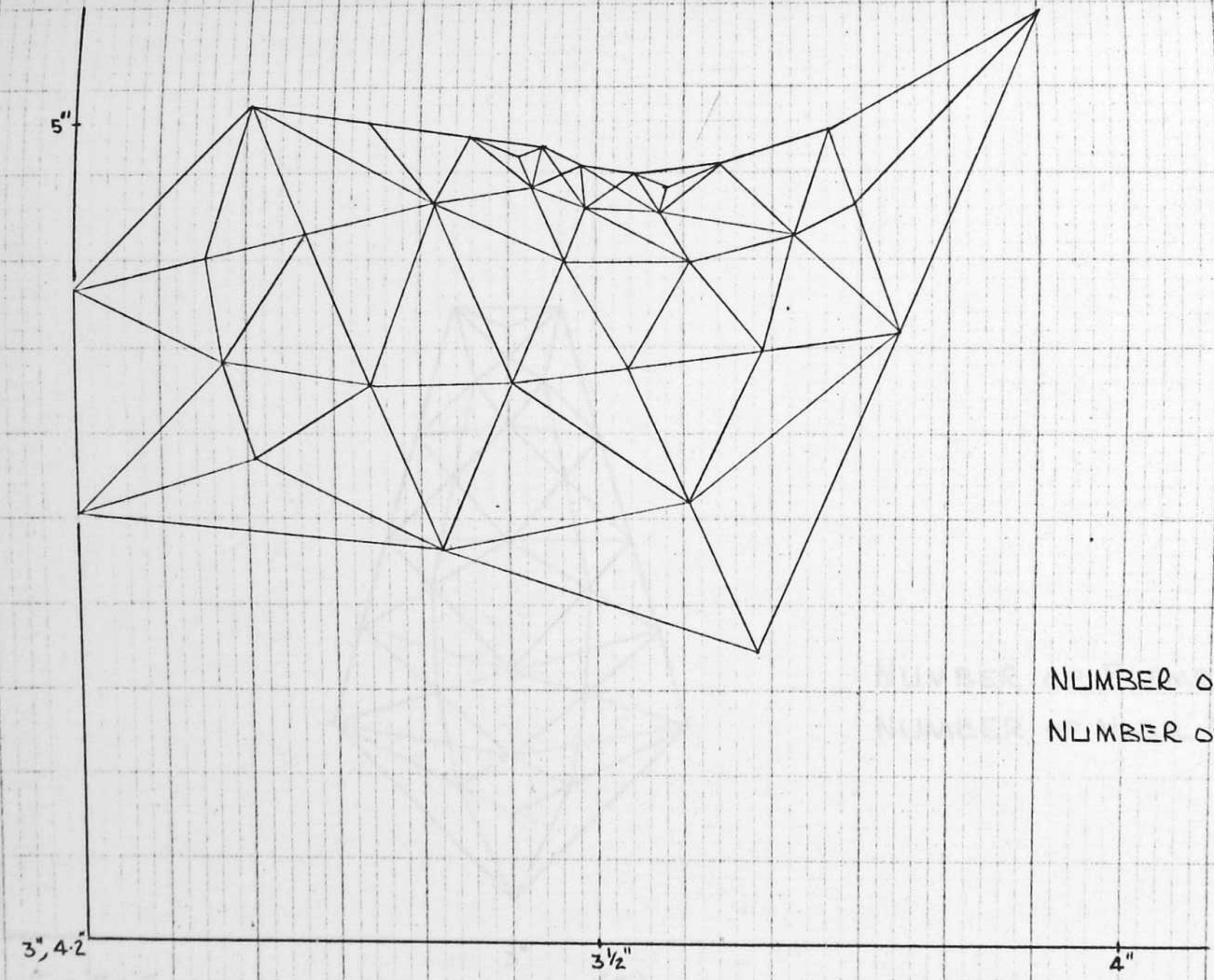
NUMBER OF NODAL POINTS = 39

FIG. 47 SUBSTRUCTURE NUMBER (2); 2 x FULL SIZE.



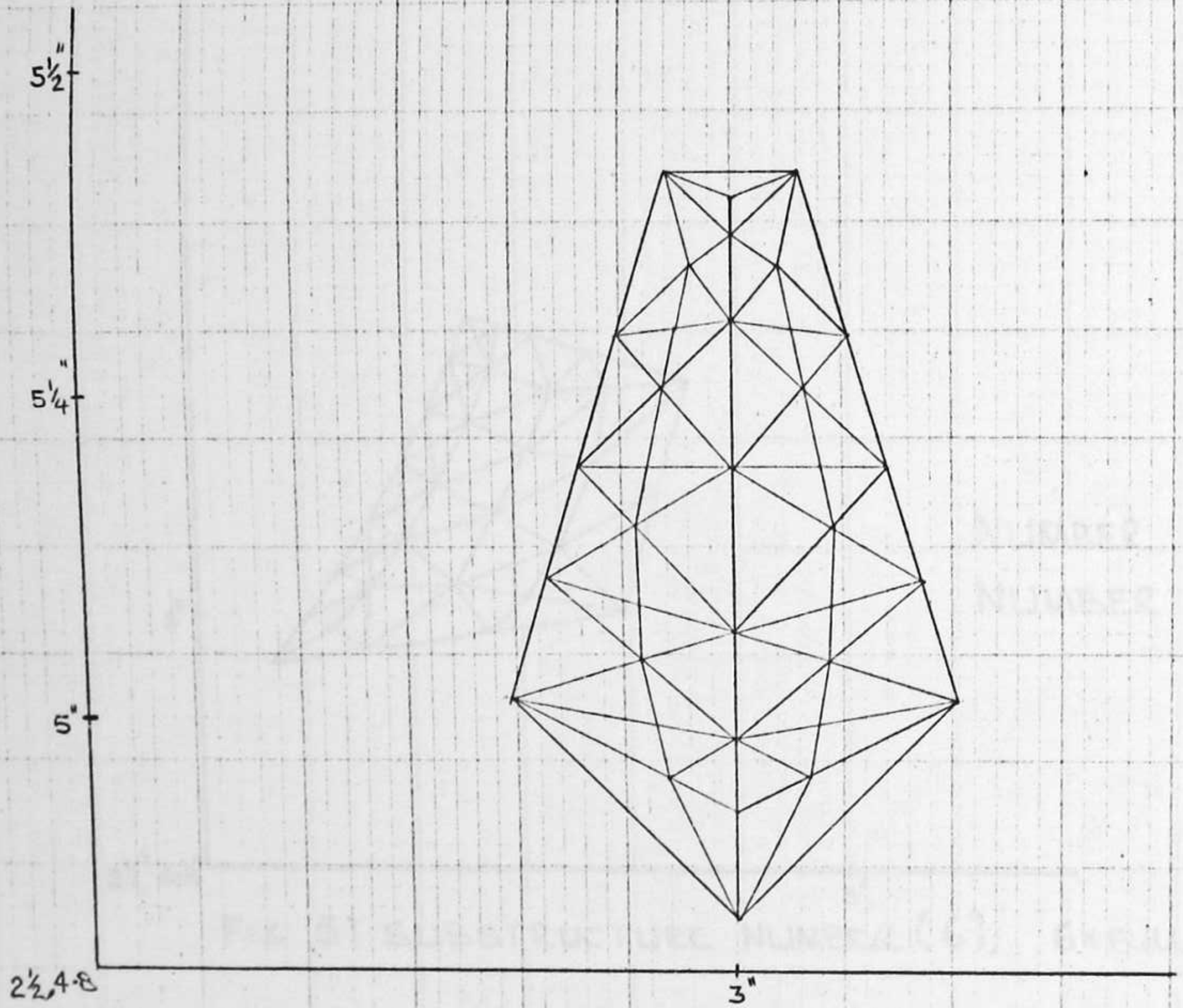
NUMBER OF ELEMENTS = 54.
NUMBER OF NODAL POINTS = 35.

FIG. 48 SUBSTRUCTURE NUMBER (3); 6 × FULL SIZE.



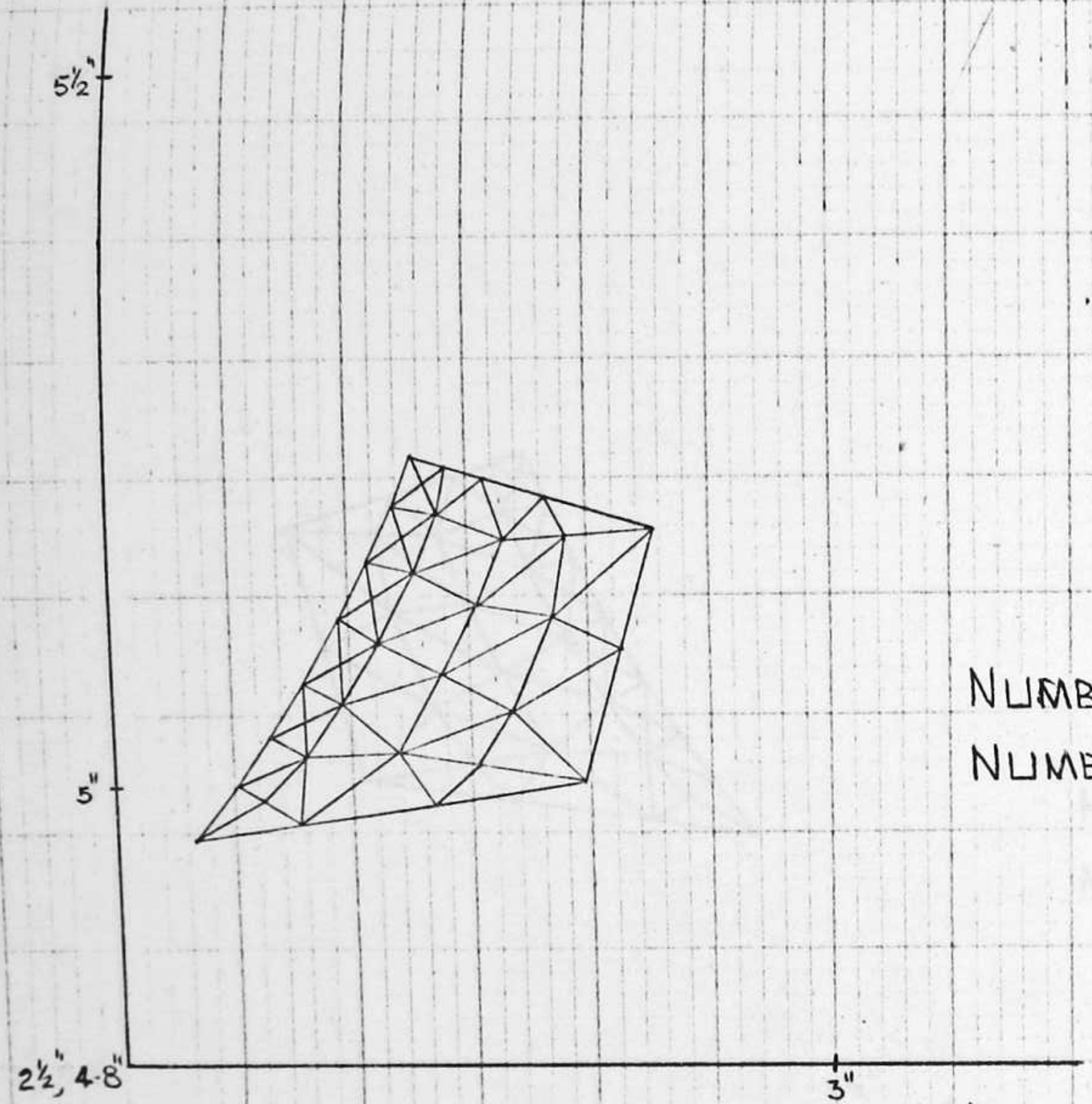
NUMBER OF ELEMENTS = 54.
NUMBER OF NODAL POINTS = 35.

FIG. 49 SUBSTRUCTURE NUMBER (4), 6 × FULL SIZE.



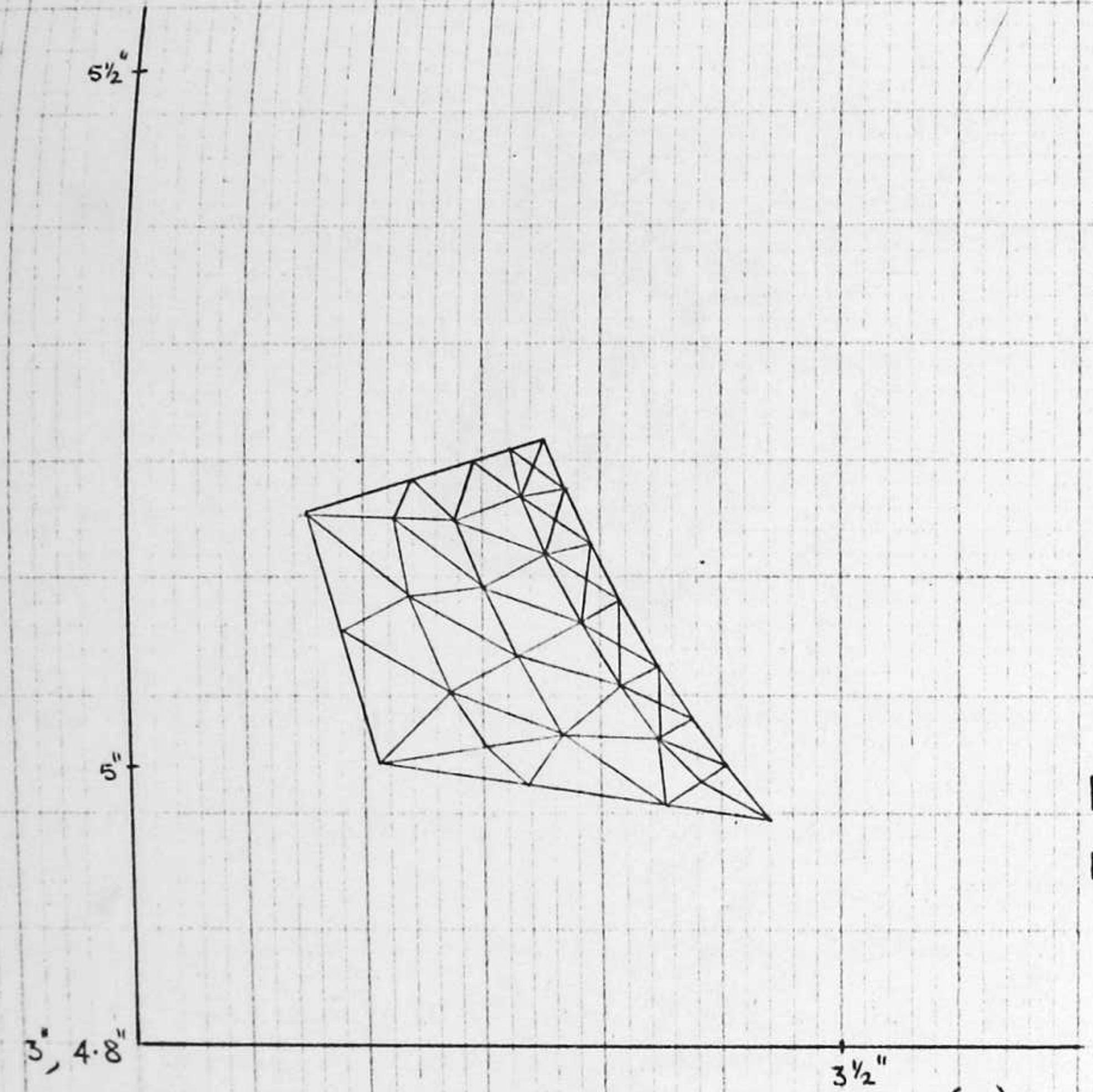
NUMBER OF ELEMENTS = 59.
NUMBER OF NODAL POINTS = 36.

FIG. 50 SUBSTRUCTURE NUMBER (5); 6× FULL SIZE.



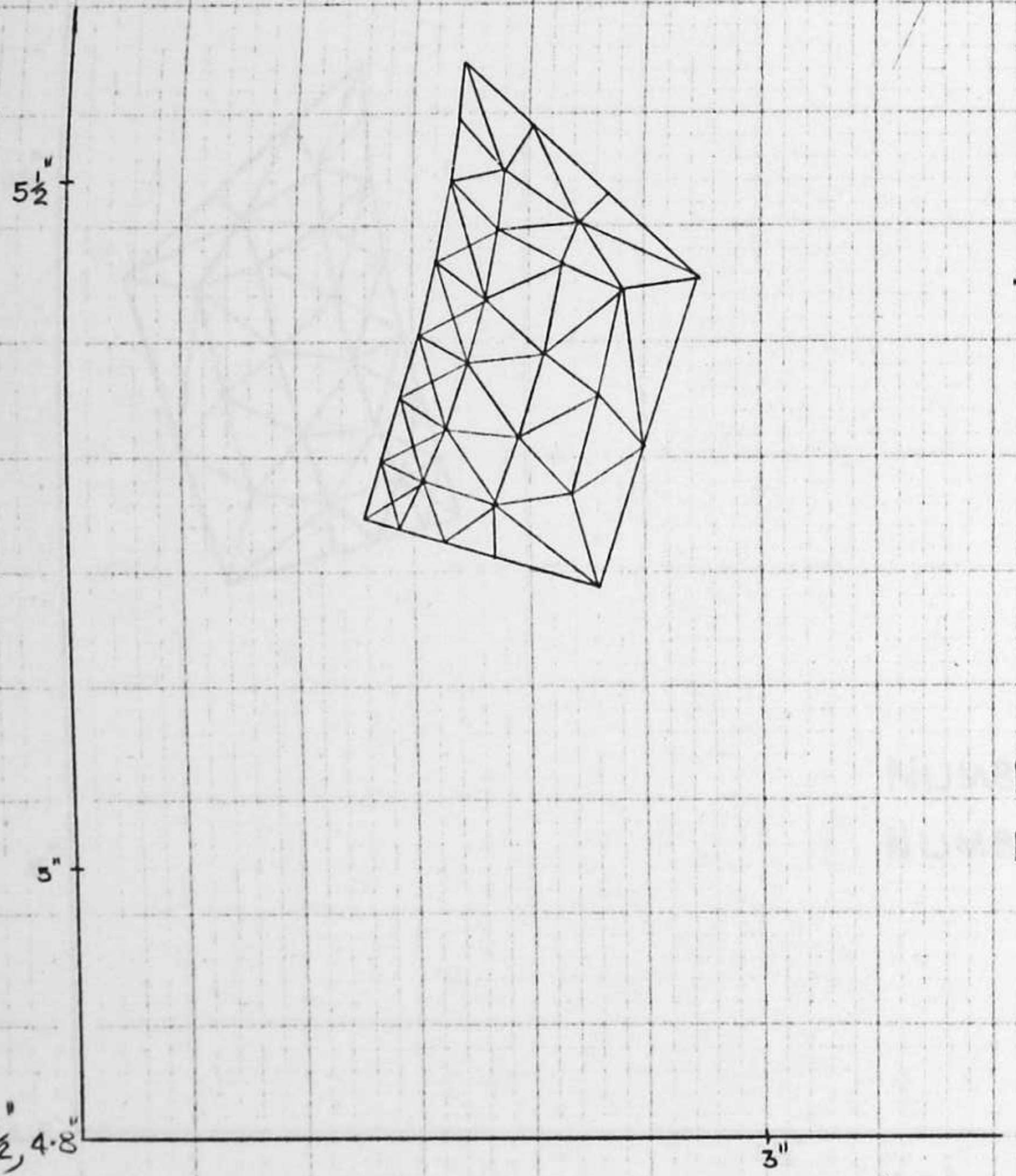
NUMBER OF ELEMENTS = 52.
NUMBER OF NODAL POINTS = 35.

FIG. 51 SUBSTRUCTURE NUMBER (6); 6× FULL SIZE.



NUMBER OF ELEMENTS = 52.
NUMBER OF NODAL POINTS = 35.

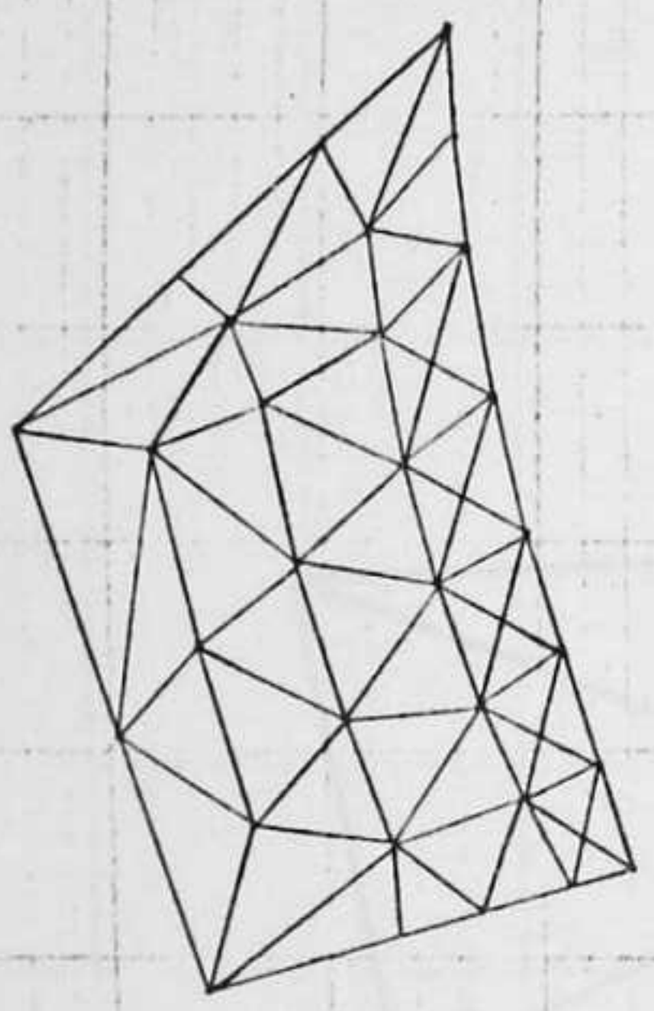
FIG. 52 SUBSTRUCTURE NUMBER (7); 6× FULL SIZE.



NUMBER OF ELEMENTS = 52.

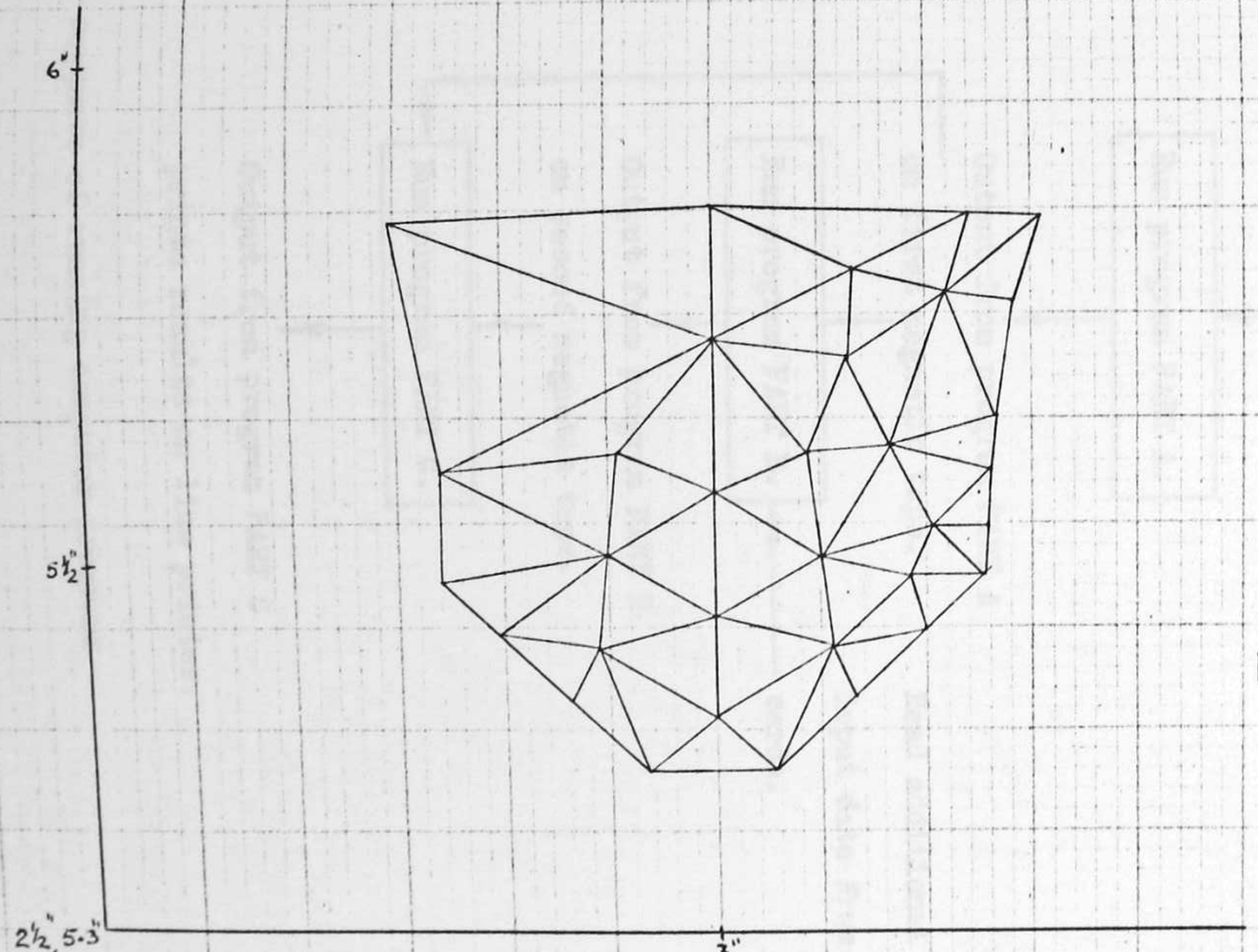
NUMBER OF NODAL POINTS = 35.

FIG. 53 SUBSTRUCTURE NUMBER (8); 6 × FULL SIZE.



NUMBER OF ELEMENTS = 52.
NUMBER OF NODAL POINTS = 35.

FIG. 54 SUBSTRUCTURE NUMBER (9); 6× FULL SIZE.



NUMBER OF ELEMENTS = 47.
NUMBER OF NODAL POINTS = 33.

FIG. 55 SUBSTRUCTURE NUMBER (10); 6 × FULL SIZE.

3. Line by Line Description of Computer Programs.

To enable the complete computer analysis to be handled withih the capasity of an 18K machine, the analysis has been split up into three separate programs. Namely PARTS A, B, and C.

The sequence of running the programs is important, and is shown in fig. 56. The programs must be run in order. PART A. first, PART B. second, and finaly PART C.

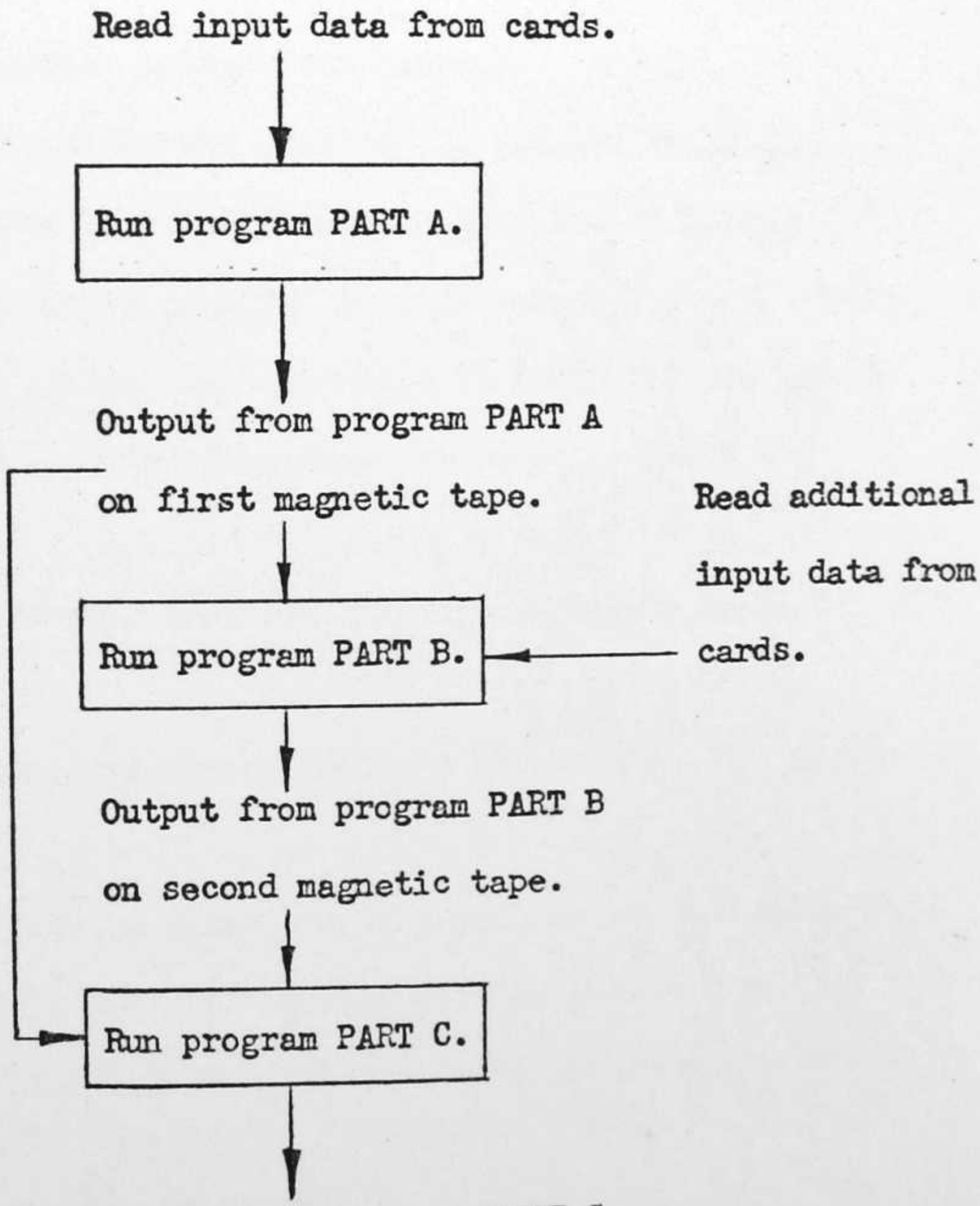


Fig. 56 sequence of running computer programs

Initial card input is to PART A. which produces output on a magnetic tape. This output together with additional input data on cards is read into PART B.

PART B. produces an output onto a second magnetic tape, which together with the output from PART A. is used as input to PART C.

PART C. finaly produces the results on line printer.

Fig. 57 indicates the main steps of PART A. of the computer analysis. The main object of this program is the formation of the boundary stiffness matrix $K_{\beta}^{(\tau)}$ for each substructure.

Refering to pages 188 to 211 for line print out of program PART A.

line no.

0005 Compress integer and logical.
This statement enables all integer and logical terms to be held in core by one word. (instead of two words required by real terms) A useful saving in storage will result with large integer arrays such as the identity array NP.

0006 Indicates that all card input data is onchannel 5.

0007 Indicates output to first magnetic tape is on channel 6,

0008 Indicates output to line printer on channel 7.
The output information on line printer consists of a print out of input data, and data errors if any.

0016 to 0024 Dimension statements arranged to enable the programs to be run within 18K. The actual core storage re-qrirements for PART A. is 16832. The dimensions of the arrays may be altered at will, providing

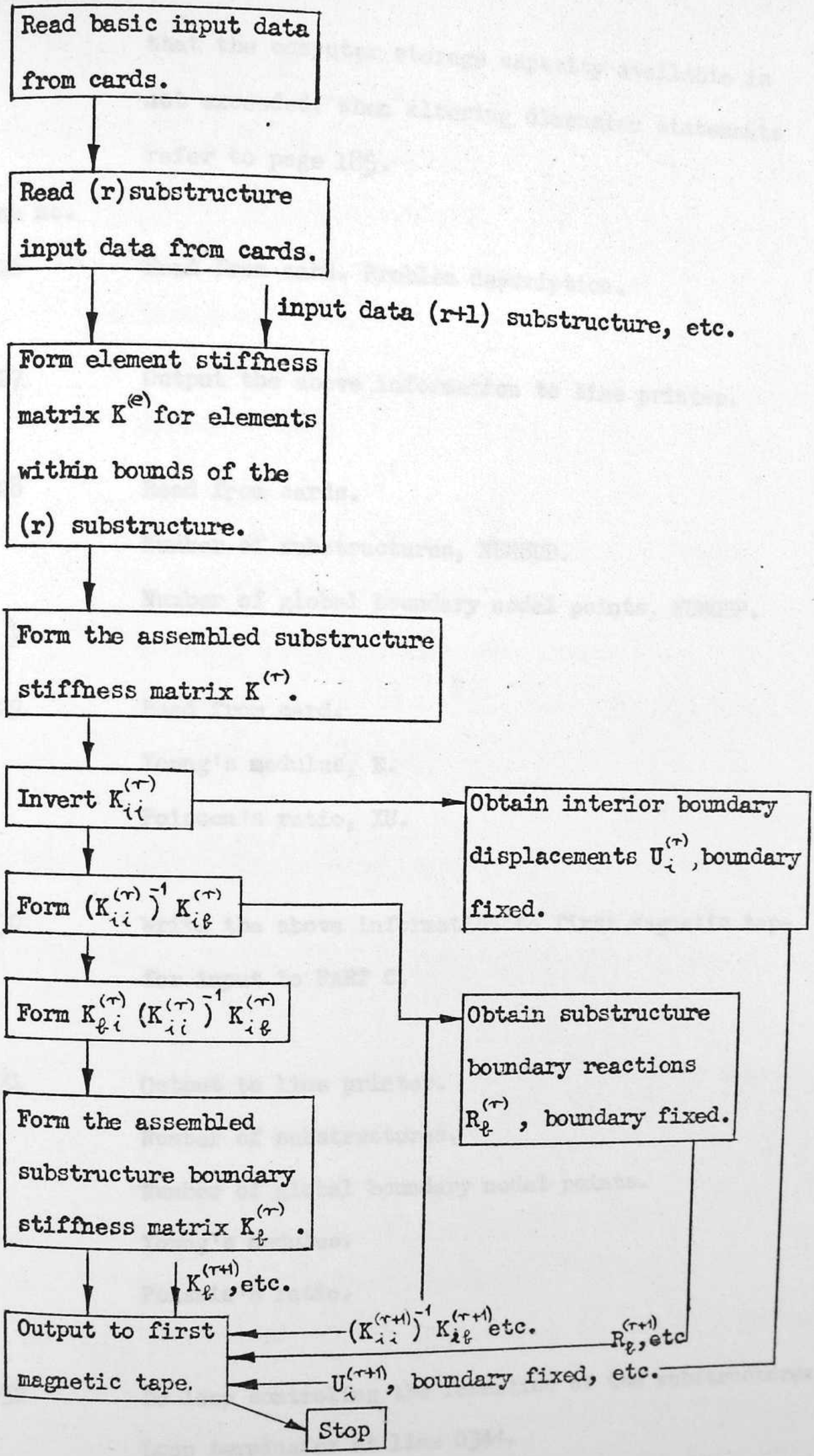


Fig.57 flow diagram for program PART A.

that the computer storage capacity available is not exceeded. When altering dimension statements refer to page 185.

line no.

0026 Read from card. Problem description.

0027 Output the above information to line printer.

0028 Read from cards.

Number of substructures, NUMSUB.

Number of global boundary nodal points, NUMGBP.

0029 Read from card.

Young's modulus, E.

Poisson's ratio, XU.

0030 Write the above information to first magnetic tape for input to PART C.

0031 Output to line printer.

Number of substructures.

Number of global boundary nodal points.

Young's modulus.

Poissin's ratio.

0032 Do loop controling the formation of the substructures.

Loop terminates at line 0344.

0033 Read from cards, for the (r) substructure.

Substructure number, NSUB.

Number of elements, NUMEL.

Number of nodal points, NUMNP.

Number of substructure boundary nodal points, NUMBP.

Number of interior nodal points with non-zero external forces, NONZIF.

0035 Read from cards, for each boundary nodal point in the (r) substructure.

Boundary nodal point numbers, array NBP.

Corresponding global boundary nodal point numbers, array NGBP.

0037 to 0047 Rearrange the arrays NBP, and NGBP in assending order on the substructure boundary nodal point numbers, array NBP.

0048 Read from cards, for each element in the (r) substructure

Element number, arrayNUME.

Nodal point number i, array NPI.

Nodal point number j, arrayNPJ.

Nodal point number k, array NPK.

The three nodal point numbers (i,j,k) must be read ANTI-CLOCKWISE round the element.

0049 Read from cards for each nodal point inthe (r) substructure.

Nodal point number (subsuucture numbering) arrayNPNUM.

X--co-ordinate, array XORD.

Y--co-ordinate, array YORD.

0050 to 0063 Rearrange the arrays NPNUM, XORD, and YORD in assending order of magnitude on the nodal point numbers, array NPNUM. The data now stored in array NPNUM should now read consecutively from 1 to NUMNP.

0064 Write to first magnetic tape, for input to programs PART B. and PART C.

Substructure number.

Number of elements.

Number of nodal points.

Number of substructure boundary nodal points.

0065 Output above information to line printer.

0067 Output to line printer.

Substructure boumdary nodal point numbers, and corresponding global boumdary nodal point numbers.

0068 Write to first magnetic tape for input to program PART C.

Element numbers, with associated nodal point numbers.

0070 Output above information to line printer.

0071 Write to first magnetic tape for input to PART C. Nodal point numbers (substructure numbering), with associated X, and Y co-ordinates.

0073 Output above information to line printer.

0074 to 0077 Initialize the interior nodal point external forces to zero in both X and Y directions, arrays PIX, and PIY.

0078 Control statement, if NONZIF=0; then there are no interior nodal points with non-zero external forces, and control passes to line 0086.

0081 Read from cards, for each interior nodal point with a non-zero force within the (r) substructure. Nodal point number (substructure numbering), NN. External force X--direction, TSX. External force Y--direction, TSY.

0082 Output above information to line printer.

0083 and 0084 Locates non-zero interior nodal point external forces into arrays PIX, and PIY using the nodal point number held by integer variable, NN as the subscript.

line no.

0087 to Initializes to zero arrays, SXX, SXY, SYX, and SYY for storage of non-zero stiffness terms, and array NP, for storage of adjacent nodal point number.

0095 Initializes the first column of array NP, to give a consecutive column of numbers as required by fig.20.

0096 to Do loop to compute difference in element co-ordinates.

0106

$$AJ = X_j - X_i$$

$$AK = X_k - X_i$$

$$BJ = Y_j - Y_i \qquad (90)$$

$$BK = Y_k - Y_i$$

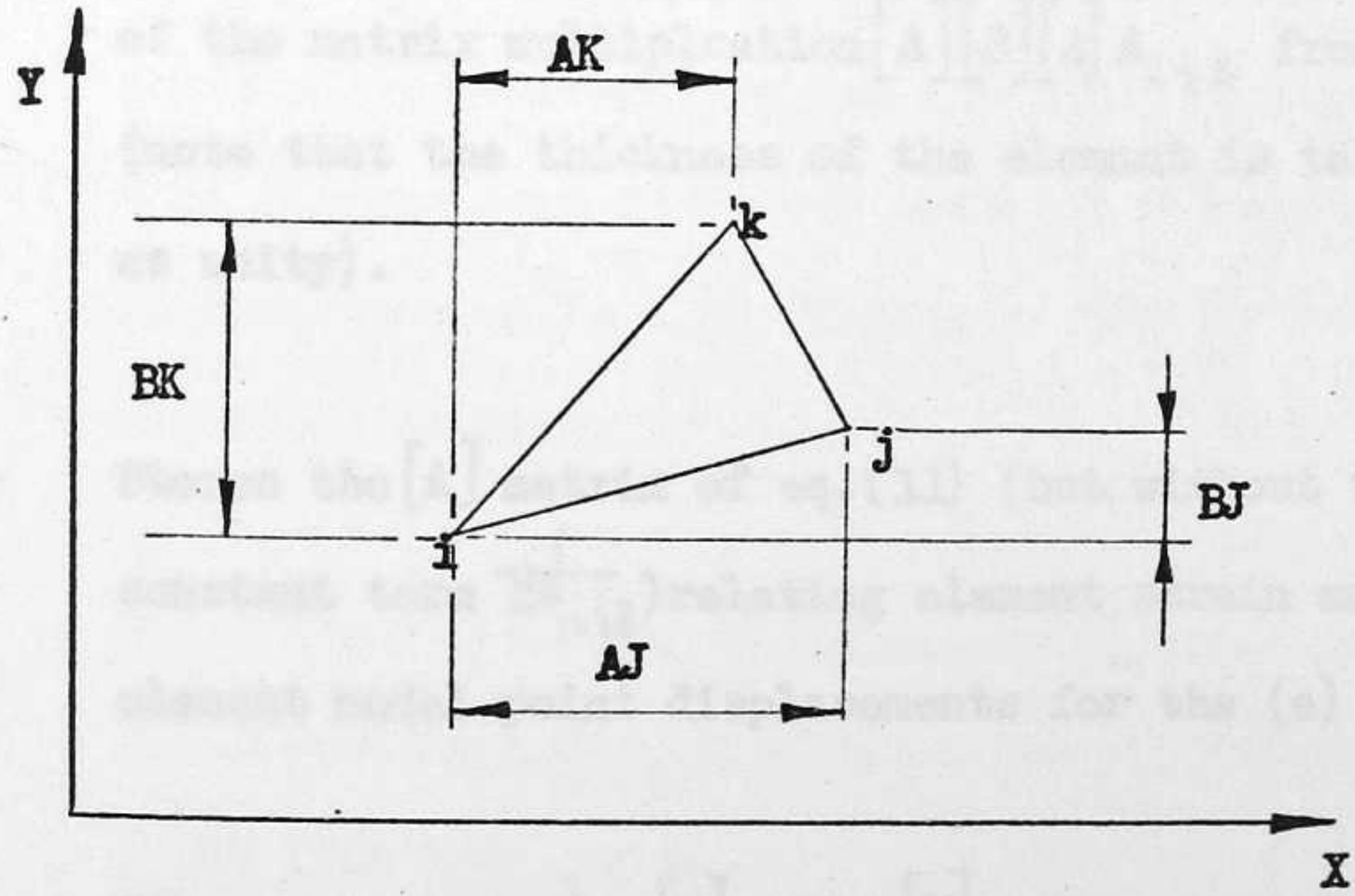


Fig.58 element co-ordinates see also fig. 1 page 17.

0104 Computes the area of the element from eq.(6).

0105 Check area of element.

If area is zero, or negative then there is an error

in the input data and the program will be terminated by statement line 0351. The offending element number will be indicated on the line printer by statement line 0346.

line no.

0107 to 0165 — Do loop to form the 6x6 element stiffness matrix $K^{(e)}$ and to assemble $K^{(e)}$ in the assembled substructure stiffness matrix $K^{(T)}$.

0108 — Computes the area of the (e) element A_{ijk} from eq.(6)

0109 — Computes the constant term

$$COMM = \frac{E}{4(1-\nu^2)A_{ijk}} \tag{91}$$

of the matrix multiplcation $[A]^T[B][A]A_{ijk}$ from eq.(21) (note that the thickness of the elemnt is taken as unity).

0110 to 0127 — Stores the $[A]$ matrix of eq.(11) (but without the constant term $\frac{1}{2A_{ijk}}$) relating element strain and element nodal point displacements for the (e) element.

0128 to 0136 — Stores matrix $\frac{1}{4A_{ijk}}[B]$. The $[B]$ matrix from eq.(12a) and (12b) relates elememt stresses to element strains for the (e) element.

0137 to 0141 — Preforms the matrix multiplcation $\frac{1}{2}[B][A]$ for the (e) element. The multiplcation $[B][A]$ from eq.(13)

relates element stresses to nodal point displacements. The program uses the 6x6 element stiffness array S for storage of $\frac{1}{2}[B][A]$.

line no.

0142 to 0144 — Transposes $\frac{1}{2}[B][A]$ to obtain $\frac{1}{2}[A]^T[B] = \frac{1}{2}[A]^T[B]$ since $[B]$ is a symmetric matrix. Array B is used for storage of $\frac{1}{2}[A]^T[B]$ thus leaving array S free for the storage of $K^{(e)}$.

0145 to 0149 — Preforms the multiplcation $[B]2A_{ijk}[A]$; where array B contains $\frac{1}{2}[A]^T[B]$, thus we have effectively formed $[A]^T[B][A]A_{ijk}$ the element stiffness matrix $K^{(e)}$ given by eq.(21) (the thickness t being taken as unity).

Line 0149 completes the formation of the element stiffness matrix $K^{(e)}$, which has now to be stored within $K^{(r)}$.

0150 to 0165 — Formation of assembled substructure stiffness matrix $K^{(r)}$ for the (r) substructure, as indicated on pages 58 to 62 Only the non-zero stiffness terms of $K^{(r)}$ are stored in the computer. A flow diagram of the assembly procedure is given by fig. 59.

0150 to 0152 — Store under array LM, the three nodal point numbers associated with the (e) element who's stiffness matrix $K^{(e)}$ has just been formed.

Storage of $K^{(e)}$ is achieved by locating each of the nine 2x2 submatrix of $K^{(e)}$ within arrays SXX, SXY, SYX, SYY using do loops, line no. 0153 and line no. 0154 to control integer variables L, and M.

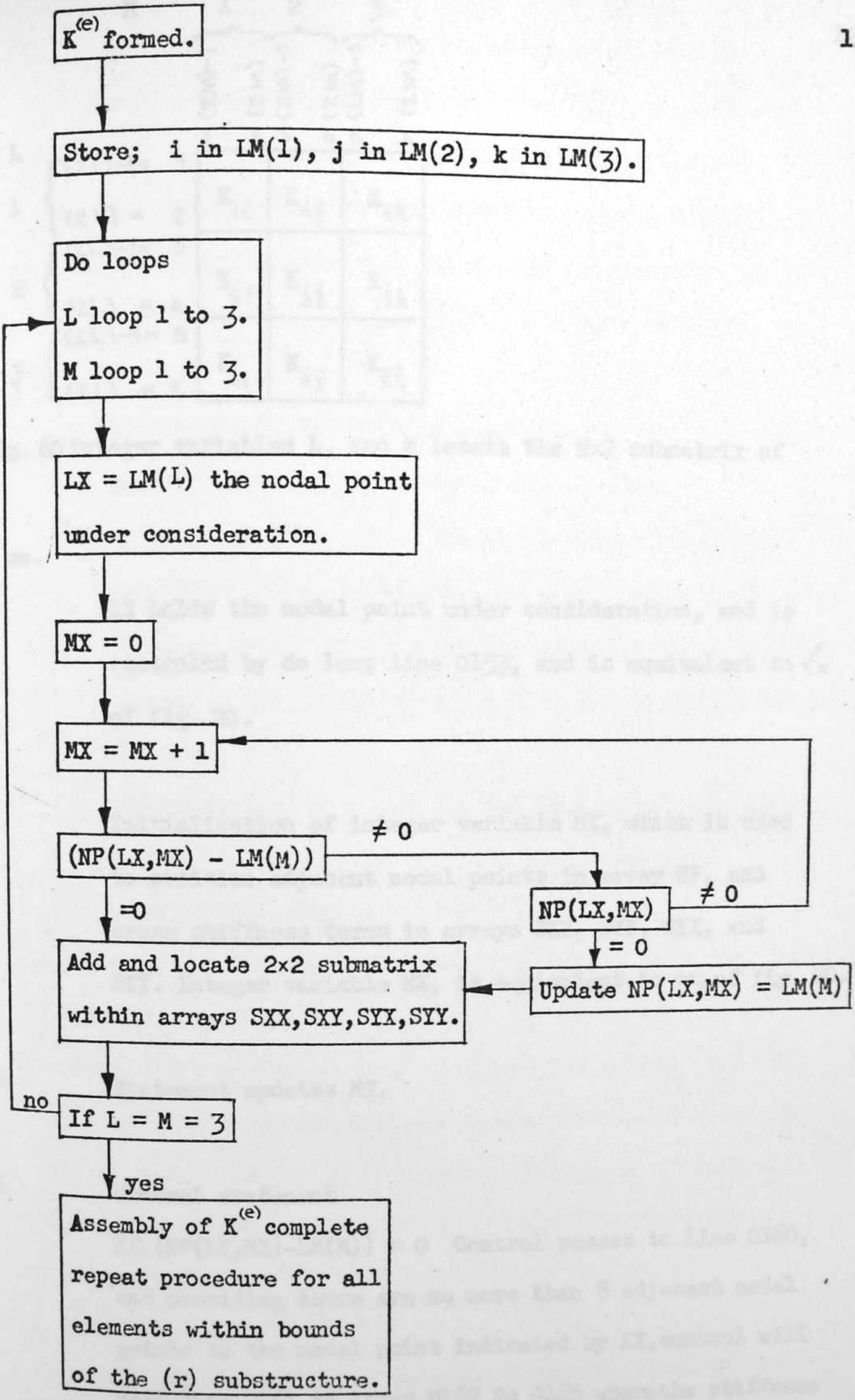


Fig. 59 flow diagram of assembly procedure for $K^{(r)}$, use with description on pages119to122.

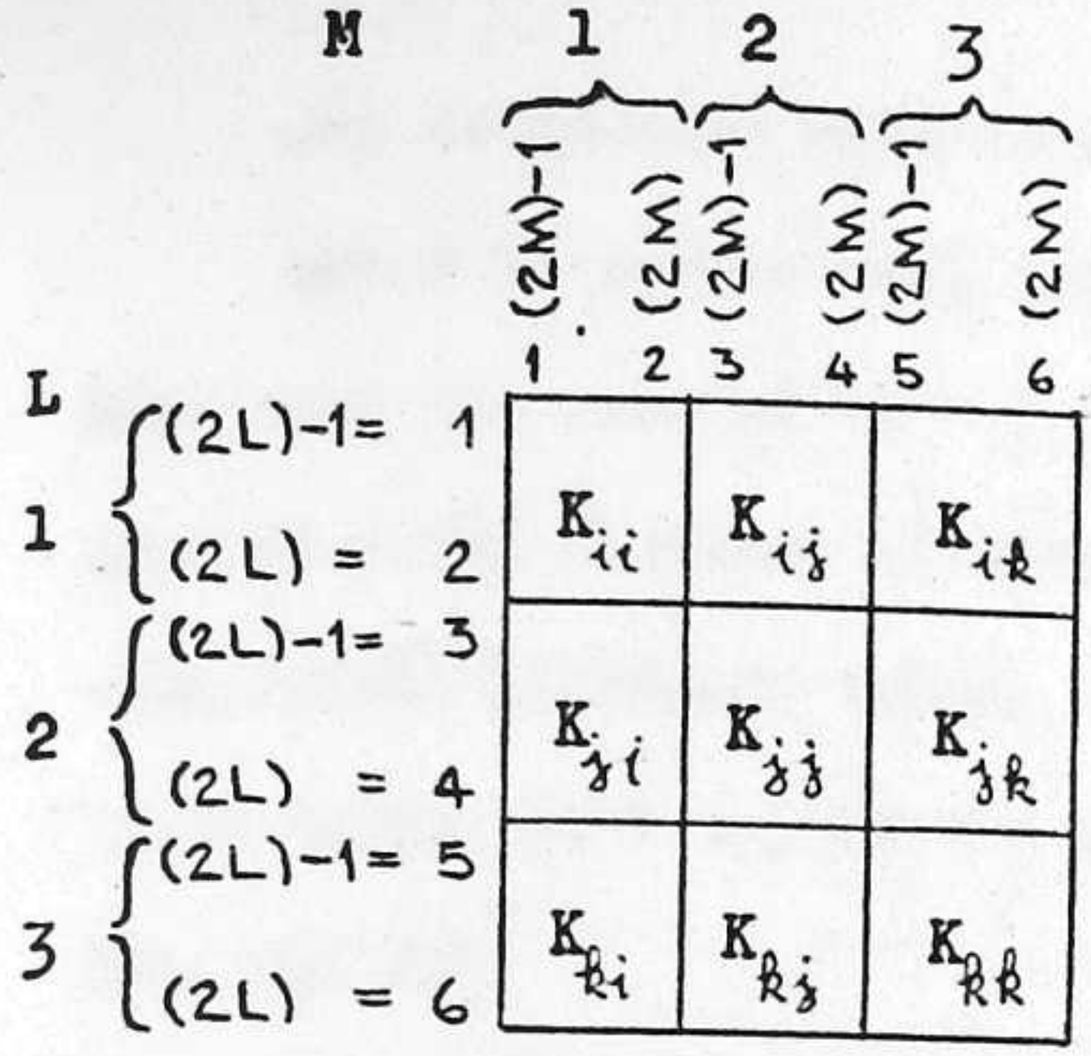


Fig. 60 integer variables L, and M locate the 2x2 submatrix of

line no.

0155 LX holds the nodal point under consideration, and is controled by do loop line 0153, and is equivalent to ℓ_x of fig. 20.

0156 Initialization of integer variable MX, which is used to position adjacent nodal points in array NP, and cross stiffness terms in arrays SXX, SXY, SYX, and SYY. Integer variable MX, is equivalent to m_x of fig. 20.

0157 Statement updates MX.

0158 Control statement

If (NP(LX,MX)-LM(M)) = 0 Control passes to line 0160, and providing there are no more than 8 adjacent nodal points to the nodal point indicated by LX, control will pass line 0161 to lines 0162 to 0165 where the stiffness terms K_{xx}, K_{xy}, K_{yx}, and K_{yy} of the 2x2 submatrix that we

are concerened with (controled by L, and M), is added to arrays SXX, SXY, SYX, and SYY respectively. Note that the case of MX = 1 and L = M, then NP(LX,1) = LX (due to initialization at line 0095) and we are dealing with direct stiffness terms, which will be stored in the first column MX = 1 , of the LX row of arrays SXX, SXY, SYX, and SYY.

If (NP(LX,MX)-LM(M)) ≠ 0 Control passes to line 0159. Here the contents of array NP are checked.

If (NP(LX,MX)) = 0 then control passes to line 0160 and NP(LX,MX) is updated to the nodal point indicated by array LM(M). In this way the adjacent nodal points for any nodal point LX is built up and stored in the array NP.

If (NP(LX,MX)) ≠ 0 then control passes to line 0157 and MX is updated and the loop continues until (NP(LX,MX)-LM(M)) = 0

The sequence of operations between lines 0153 and 0165 is repeated untill L=M=3

$K^{(e)}$ for the (e) element will be positioned within SXX, SXY, SYX, and SYY. This procedure is repeated for each element within the bounds of the (r) substructure, and effectively stores the non-zero terms of the assembled substructure stiffness matrix $K^{(r)}$.

line no.

0166 to 0170 Do loop to count and store in array NAP, the number of adjacent nodal points + 1 for each nodal point in the (r) substructure.

line no.

0171 to 0174 — Initializes arrays NOTE and NCODE.

0175 to 0181 — Rearranges array NOTE to suit the partitioned numbering order of $K^{(r)}$. The corresponding global boundary nodal points numbering for the (r) substructure boundary nodal points is stored in the array NCODE at line 0180.

Arrays NOTE and NCODE will now be in the general form as required by fig. 18.

0182 to 0184 — Write to first magnetic tape; arrays NOTE and NCODE for the (r) substructure , for input to programs PART B, and C.

0185 — Stores under NUMBER the size of the square matrix $K_{ii}^{(r)}$ which is to be inverted.

0186 to 0208 — Formation of $K_{ii}^{(r)}$. This matrix is to be held in core, including all zero terms by array SII.

$K_{ii}^{(r)}$ is formed by calling the appropriate stiffness terms from arrays SXX, SXY, SYX, and SYY and placing them in 2×2 submatrix form into array SII in column order using integer variables K and I. Since we are dealing with $K_{ii}^{(r)}$ in partitioned formation we must refer to array NOTE (using integer variables M and J) to select the nodal point numbers associated with $K_{ii}^{(r)}$. Both the

rows and columns of $K_{ii}^{(\tau)}$ are associated with integer nodal points.

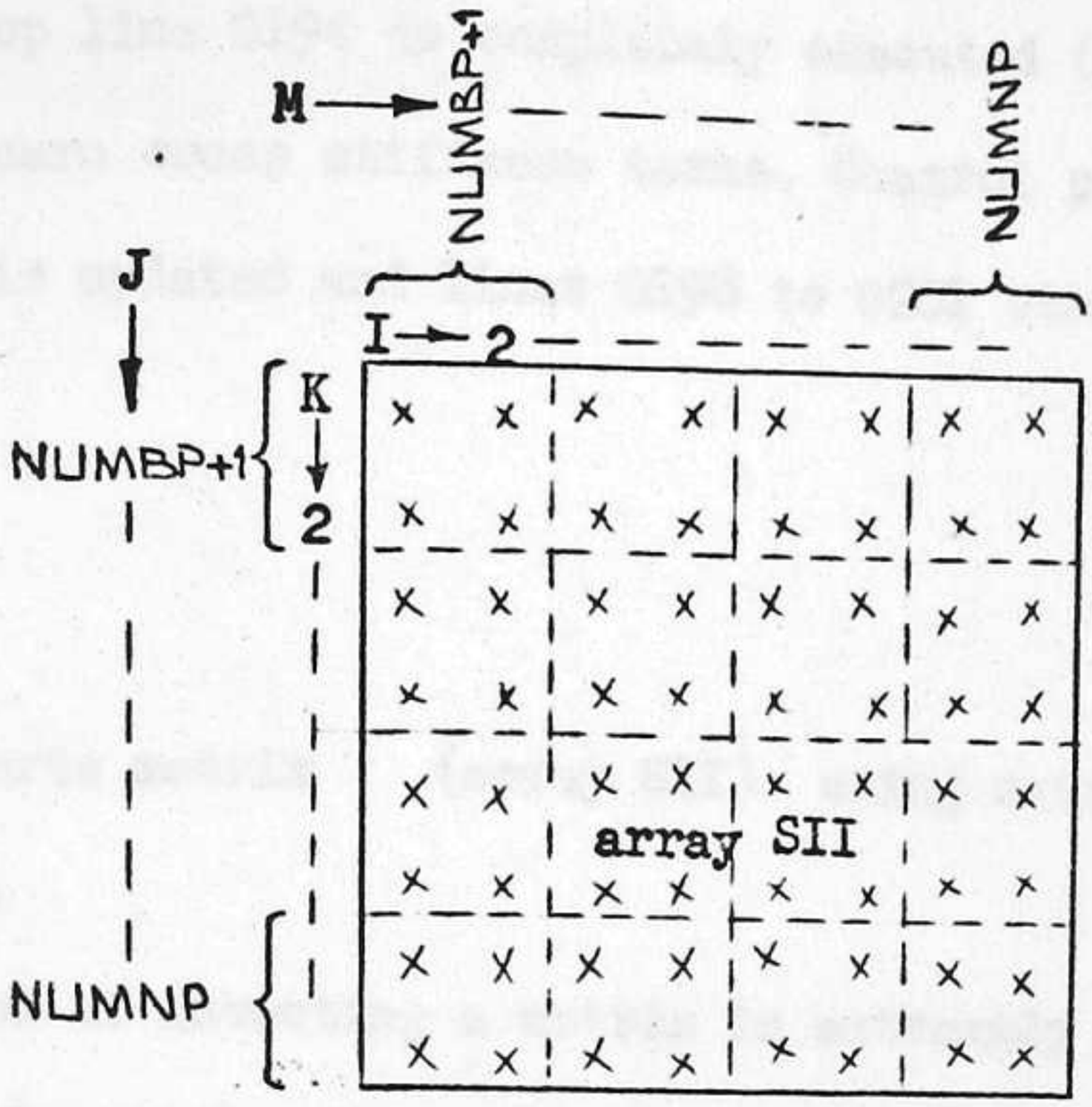


Fig. 61 integer variables used in the formation of array SII.

line no.

0189 MM stores the column nodal point number of array SII.

0192 JJ stores the row nodal point number under consideration.

0193 NUM store sthe number of adjacent nodal points + 1 for nodal point JJ.

0194 Compares nodal point JJ and its adjacent nodal points with nodal point MM.

If (NP(JJ,L)=MM) then we are dealing with either direct stiffness terms (JJ=MM) or non-zero cross stiffness terms Control passes to line 0203 where K is updated by 2 (since we are storing 2×2 submatrix) and lines 0204 to 0207 store the non-zero stiffness terms inarray SII, using integer

variables K and I to locate the submatrix.

If do loop line 0194 is completely executed (ie L=NUM) then we have zero cross stiffness terms. Control passes to line 0197, K is updated and lines 0198 to 0201 store the zero terms.

line no.	
0209 to 0238	Inverts matrix (array SII) using matrix partitioning.

This method of inverting a matrix is extreamly efficient it involves only simple addition, subtraction,multiplcation, and division, there is no elimination and no searching for pivots etc. The flow diagram for the invertion is shown in fig. 62 .

The theory of the matrix partioning is given in Appendix 1.

Summerising from appendix 1 to invert SII, we require

to invert SII_{11} to obtain SII_{11}^{-1}

compute $SII_{11}^{-1}\ SII_{12}$ (92)

compute $SII_{21}\ SII_{11}^{-1}$ (93)

compute $SII_{21}\ SII_{11}^{-1}\ SII_{12}$ (94)

compute $SII_{22} - SII_{21}\ SII_{11}^{-1}\ SII_{12}$ (95)

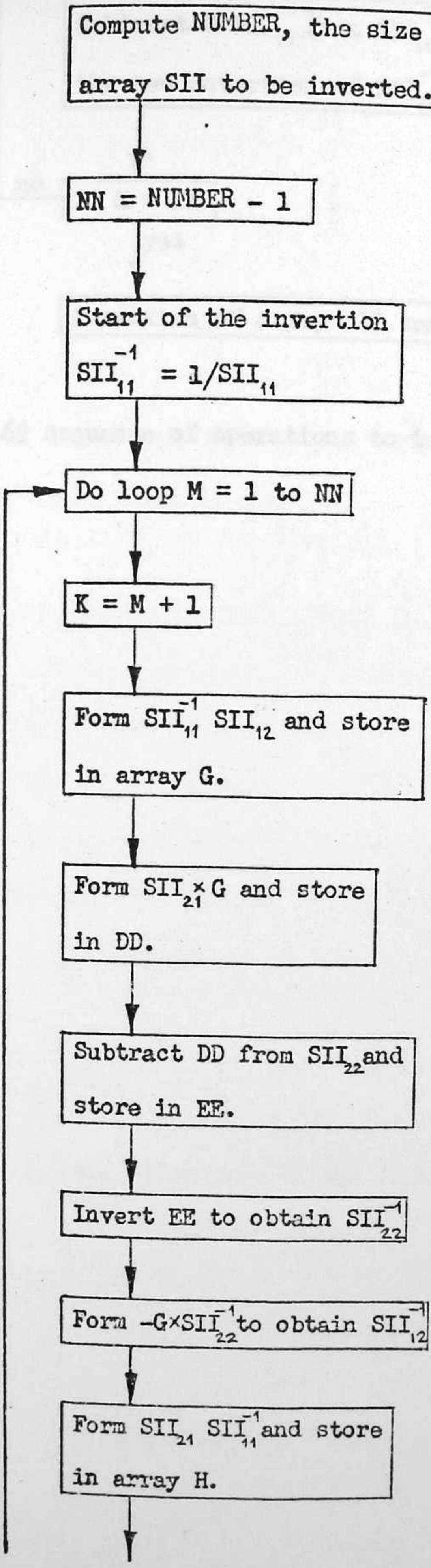
Then the inverted matrix is

$$SII_{22}^{-1} = (SII_{22} - SII_{21}\ SII_{11}^{-1}\ SII_{12})^{-1} \quad (96)$$

$$SII_{12}^{-1} = -SII_{11}^{-1}\ SII_{12}\ SII_{22}^{-1} \quad (97)$$

$$SII_{21}^{-1} = -SII_{22}^{-1}\ SII_{21}\ SII_{11}^{-1} \quad (98)$$

$$SII_{11}^{-1} = SII_{11}^{-1} - SII_{11}^{-1}\ SII_{12}\ SII_{21}^{-1} \quad (99)$$


Compute NUMBER, the size of the array SII to be inverted.
NN = NUMBER - 1
Start of the invertion
$SII_{11}^{-1} = 1/SII_{11}$
Do loop M = 1 to NN
K = M + 1
Form SII_{11}^{-1} SII_{12} and store in array G.
Form $SII_{21} \times G$ and store in DD.
Subtract DD from SII_{22} and store in EE.
Invert EE to obtain SII_{22}^{-1}
Form $-G \times SII_{22}^{-1}$ to obtain SII_{12}^{-1}
Form SII_{21} SII_{11}^{-1} and store in array H.

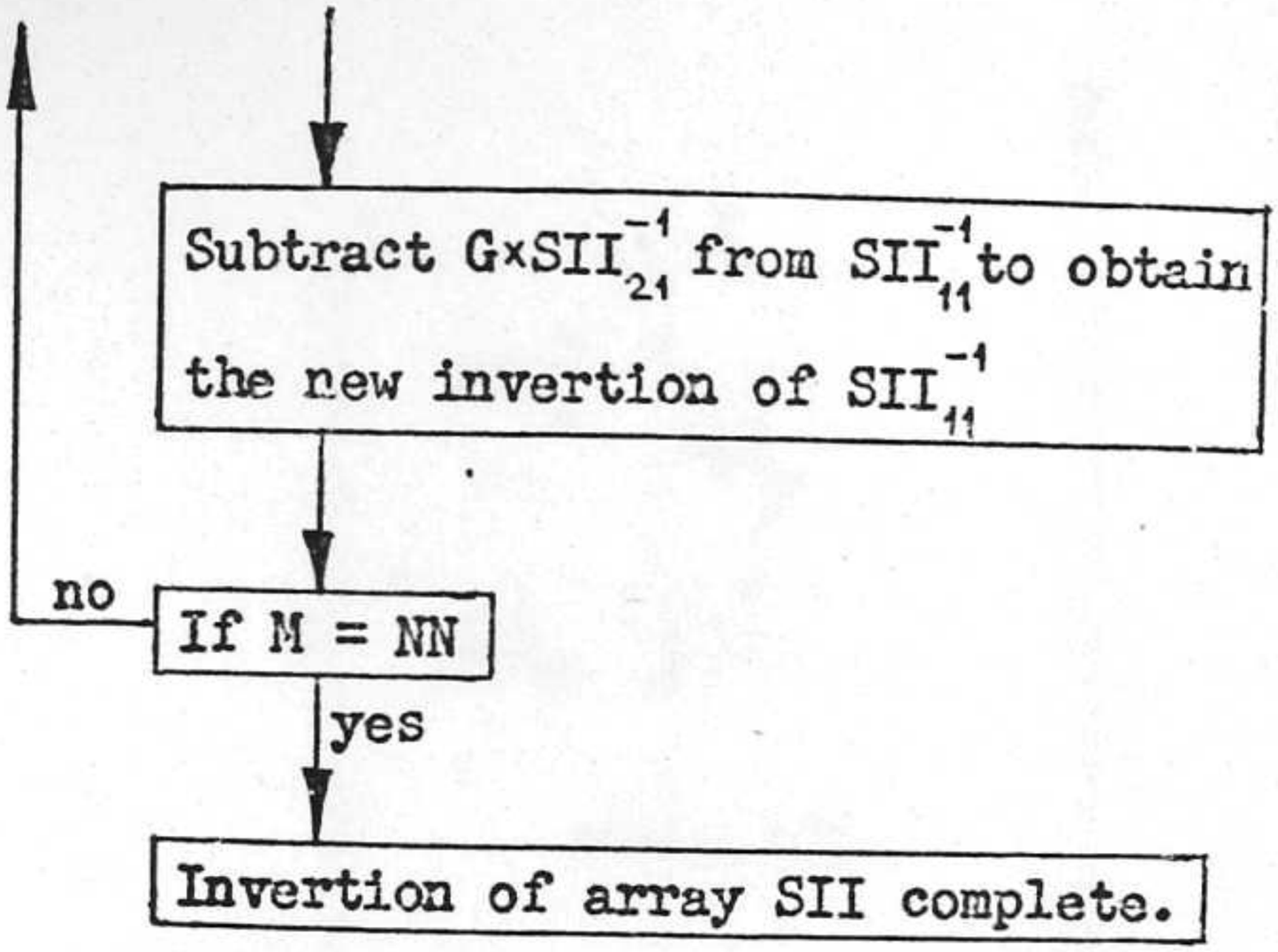


Fig.62 sequence of operations to invert $K_{ii}^{(r)}$ by partitioning.

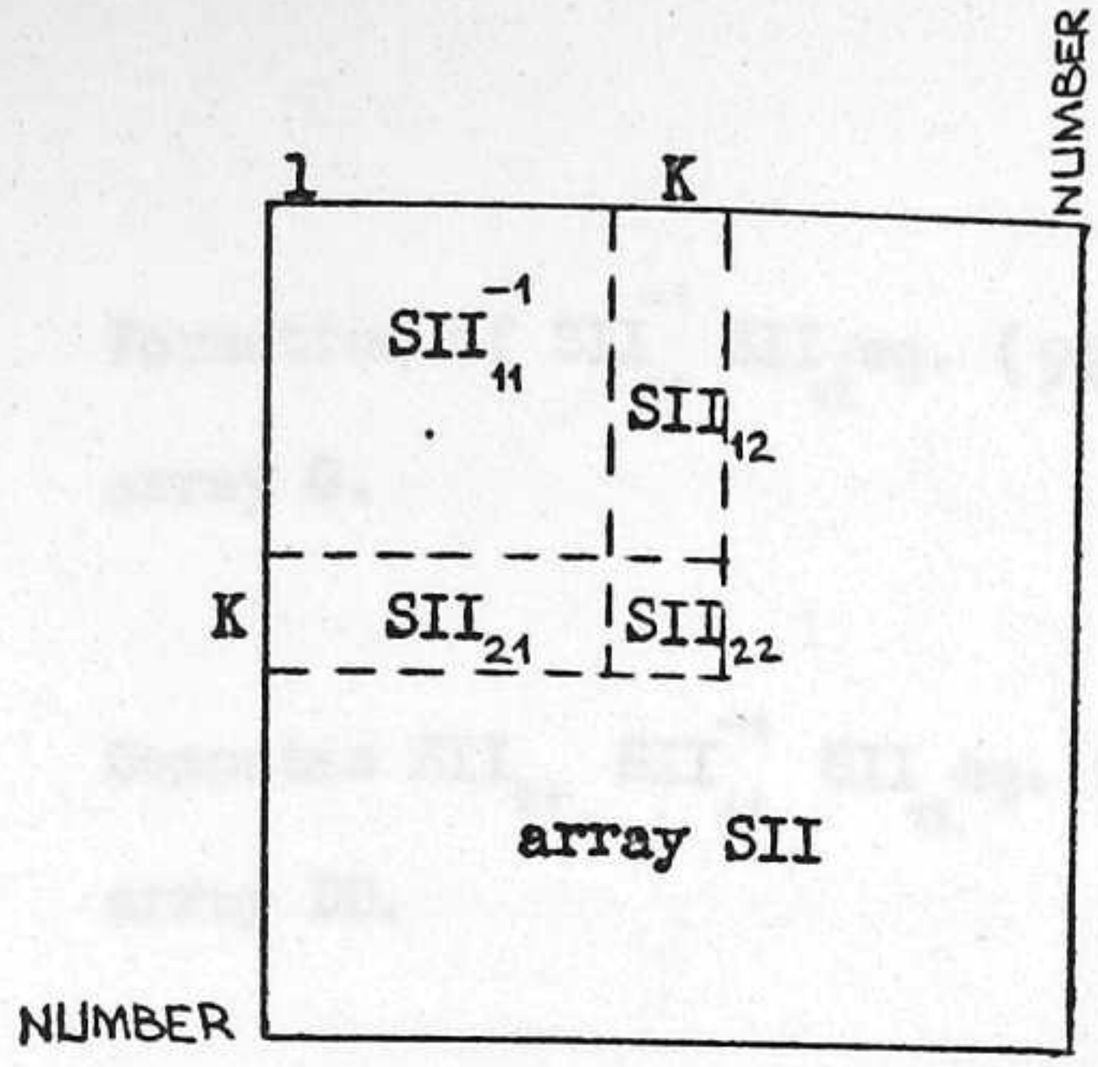


Fig.63 partitioning of matrix SII. SII_{11}^{-1} is the previously inverted part of the matrix.

The basis of the method is that the first term of the array SII(1,1) is inverted to SII_{11}^{-1} The matrix partitioning is then applied to SII_{11}^{-1} and the next row and column (the K row and column) as shown by fig.63.When inverted the square matrix bounded by size 1 to K becomes the new size of SII_{11}^{-1} The procedure is repeated taking each row in turn till K = NUMBER.

line no.

0209 NN stores the number of rows - 1 being inverted.

0210 The first term of the first row is inverted to SII_{11}^{-1} to start the invertion.

0211 Do loop cycling from row /column number, 2 to number.

0212 K stores the row /column number which is to be inverted.

line no.

0213 to 0217 — Formation of $SII_{11}^{-1}\ SII_{12}$ eq. (92), and is stored in array G.

0219 to 0221 — Computes $SII_{21}\ SII_{11}^{-1}\ SII_{12}$ eq. (94), and is stored in array DD.

0222 — Computes $SII_{22} - SII_{21}\ SII_{11}^{-1}\ SII_{12}$ eq. (95), and store in EE.

0223 — Computes $(SII_{22} - SII_{21}\ SII_{11}^{-1} SII_{12})^{-1}$ eq. (96), to form SII_{22}^{-1}

0224 to 0226 — Computes $- SII_{11}^{-1}\ SII_{12}\ SII_{22}^{-1}$ eq. (97), to form SII_{12}^{-1}

0227 to 0231 — Forms $SII_{21}\ SII_{11}^{-1}$ eq. (93), and stores in array H.

0232 to 0234 — Computes $- SII_{22}^{-1}\ SII_{21}\ SII_{11}^{-1}$ eq. (98), to form SII_{21}^{-1}

0235 to 0238 — Computes $SII_{11}^{-1} - SII_{11}^{-1}\ SII_{12}\ SII_{21}^{-1}$ eq. (99), to form the new values of SII_{11}^{-1}

This completes one row/column of SII. The complete inversion of SII is terminated by do loop line 0211 at cycle NN.

line no.

0239 to 0249 — Computes $U_i^{(r)}{}_{u_b=0}$ the interior nodal point displacements due to interior nodal point external forces $P_i^{(r)}$ with the boundary fixed. ie $U_b^{(r)} = 0$.

0241 to 0242 — Initializes the nodal point displacements in both X and Y directions UIX, and UIY.

0245 — JJ stores the nodal point number under consideration.

0244 to — Preforms the matrix multiplcation $(K_{ii}^{(r)})^{-1} P_i^{(r)}$ eq.(61), to form the X and Y interior nodal point displacements for interior nodal point number JJ.

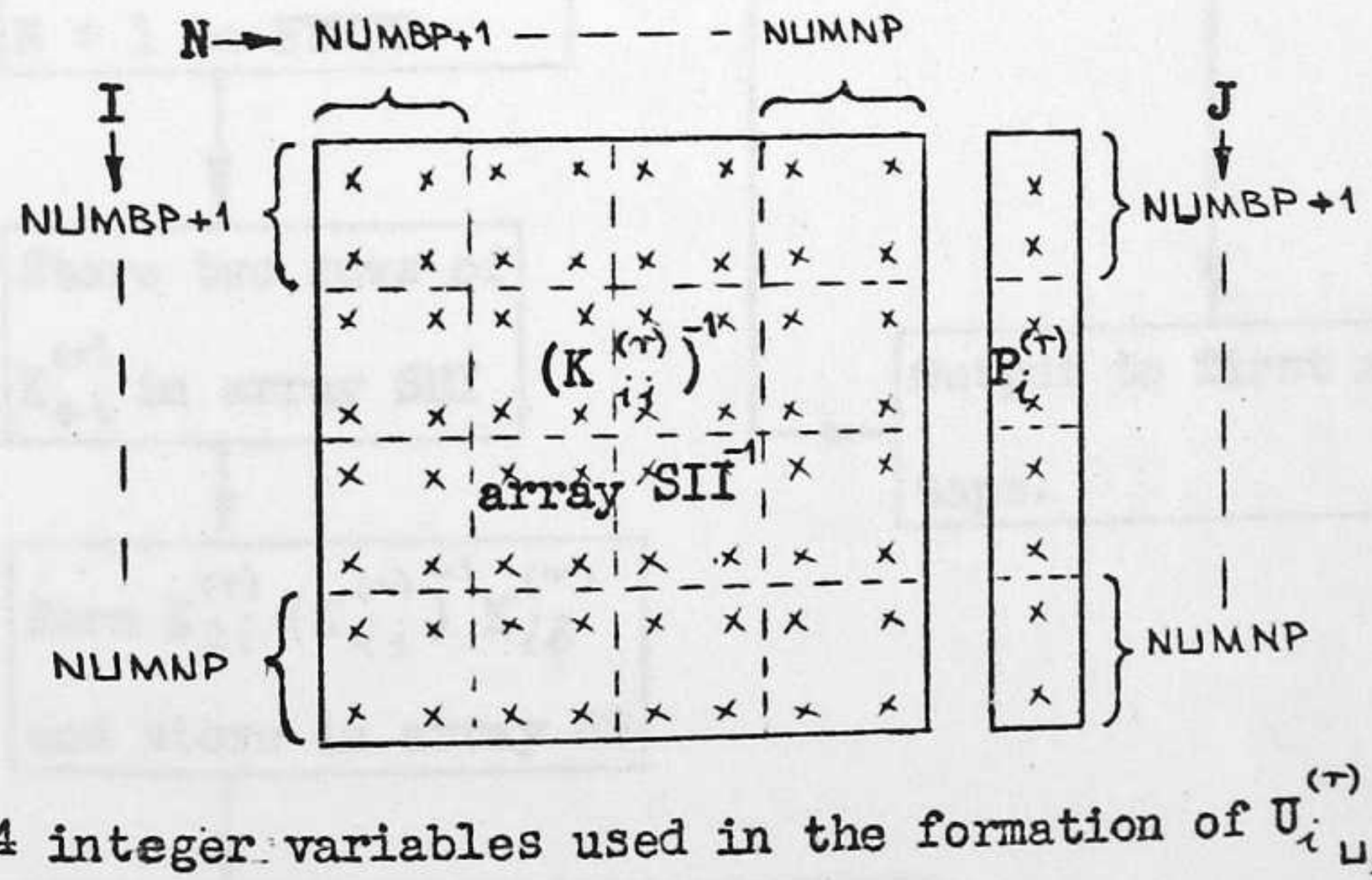
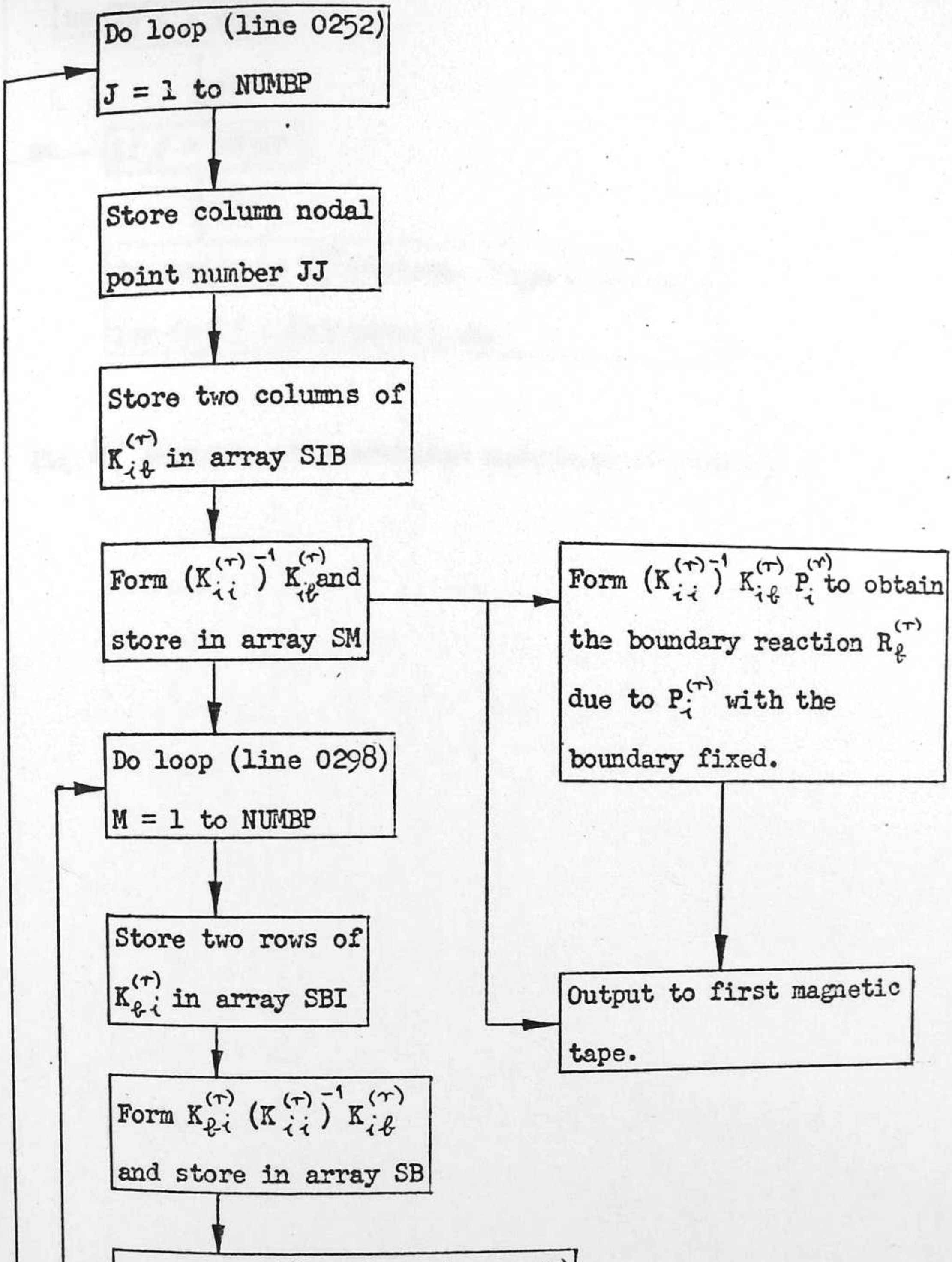


Fig. 64 integer variables used in the formation of $U_i^{(r)}{}_{u_b=0}$

0252 to 0344 — Do loop to form the substructure boundary stiffness matrix $K_b^{(r)}$ for the (r) substructure.

The procedure of forming $K_b^{(r)}$ is outlined on pages 64 to 65 and the flow diagram is given by fig. 65.


Do loop (line 0252)
J = 1 to NUMBP
Store column nodal point number JJ
Store two columns of $K_{ib}^{(\tau)}$ in array SIB
Form $(K_{ii}^{(\tau)})^{-1} K_{ib}^{(\tau)}$ and store in array SM
Form $(K_{ii}^{(\tau)})^{-1} K_{ib}^{(\tau)} P_i^{(\tau)}$ to obtain the boundary reaction $R_b^{(\tau)}$ due to $P_i^{(\tau)}$ with the boundary fixed.
Do loop (line 0298)
M = 1 to NUMBP
Store two rows of $K_{bi}^{(\tau)}$ in array SBI
Form $K_{bi}^{(\tau)} (K_{ii}^{(\tau)})^{-1} K_{ib}^{(\tau)}$ and store in array SB
Subtract $K_{bi}^{(\tau)} (K_{ii}^{(\tau)}) K_{ib}^{(\tau)}$ from $K_{bb}^{(\tau)}$ to form a 2×2 submatrix of $K_b^{(\tau)}$ and store in array SB
Output the 2×2 submatrix SB to first magnetic tape.
Output to first magnetic tape.

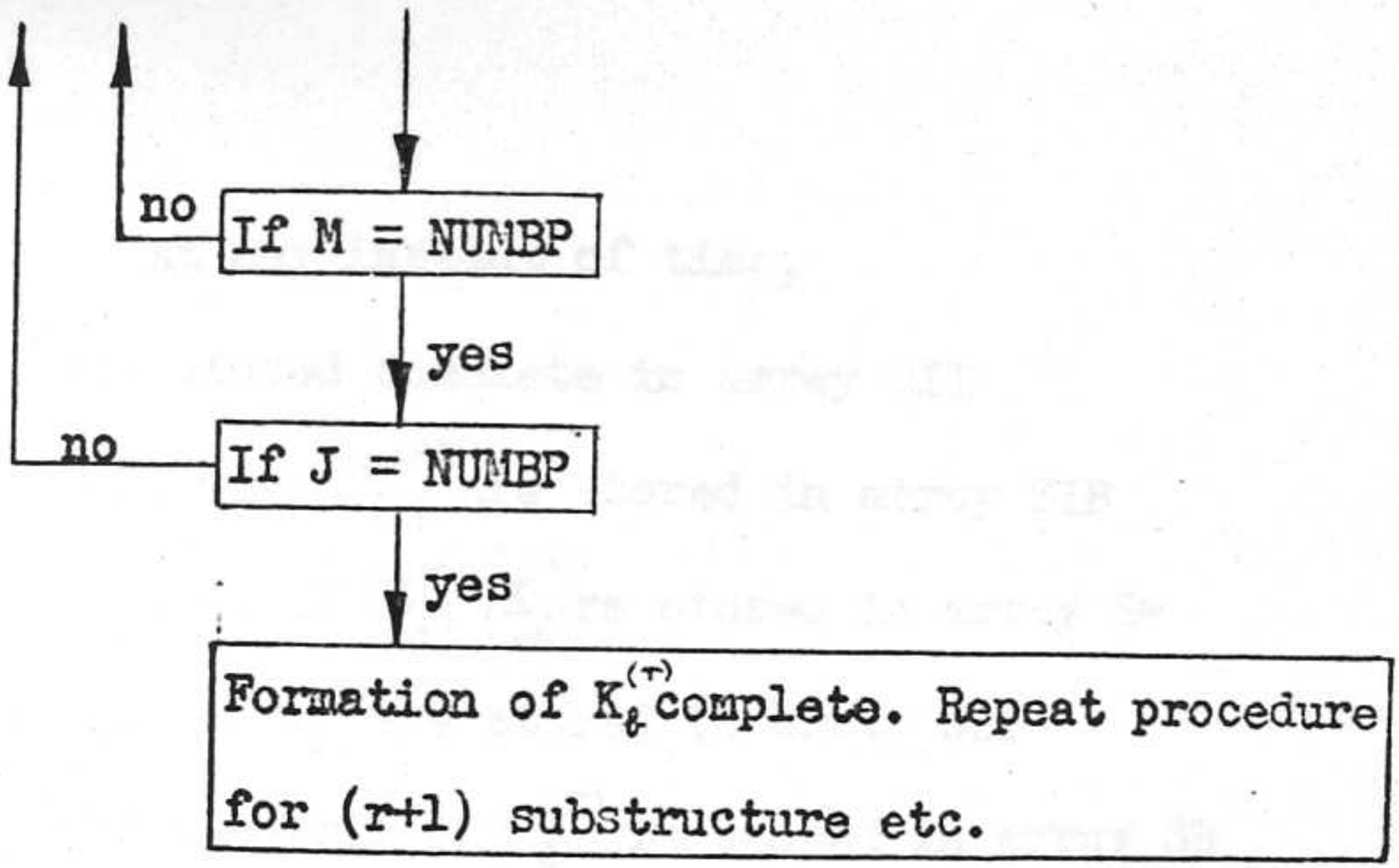


Fig. 65 sequence of operations necessary to form $K_{\beta}^{(r)}$.

At any instant of time,

$(K_{ii}^{(\tau)})^{-1}$ is stored complete in array SII

Two columns of $K_{i\ell}^{(\tau)}$ are stored in array SIB

Two columns of $(K_{ii}^{(\tau)})^{-1}K_{i\ell}^{(\tau)}$ are stored in array SM

Two rows of $K_{\ell i}^{(\tau)}$ are stored in array SBI

A 2x2 sub matrix of $K_{\ell}^{(\tau)}$ is stored in array SB

line no.

0253 to Stores two columns of $K_{i\ell}^{(\tau)}$ in array SIB, fig. 66 .

0272 The two columns are formed by calling in the appropriate stiffness terms from arrays SXX, SXY, SYX, and SYY using integer variables J, and I to obtain the nodal point numbers from array NOTE.

Selection of stiffness terms is achieved by do loop lines 0258 to 0260.

If (NP(II,L)=JJ) then nodal point JJ is equal to (direct stiffness) or is adjacent to (non-zero cross stiffness) nodal point II, and control passes to lines 0267 to 0272.

If the loop is completely executed (ie L=NUM) we store zero cross stiffness terms; lines 0262 to 0265.

0273 to 0283 Preforms the matrix multiplcation SII^{-1} SIB to form two columns of $(K_{ii}^{(\tau)})^{-1}K_{i\ell}^{(\tau)}$ and stores in the array SM, fig.67.

0284 to 0286 Output the two columns of $(K_{ii}^{(\tau)})K_{i\ell}^{(\tau)}$ stored in array SM, to the first magnetic tape for input to PART C.

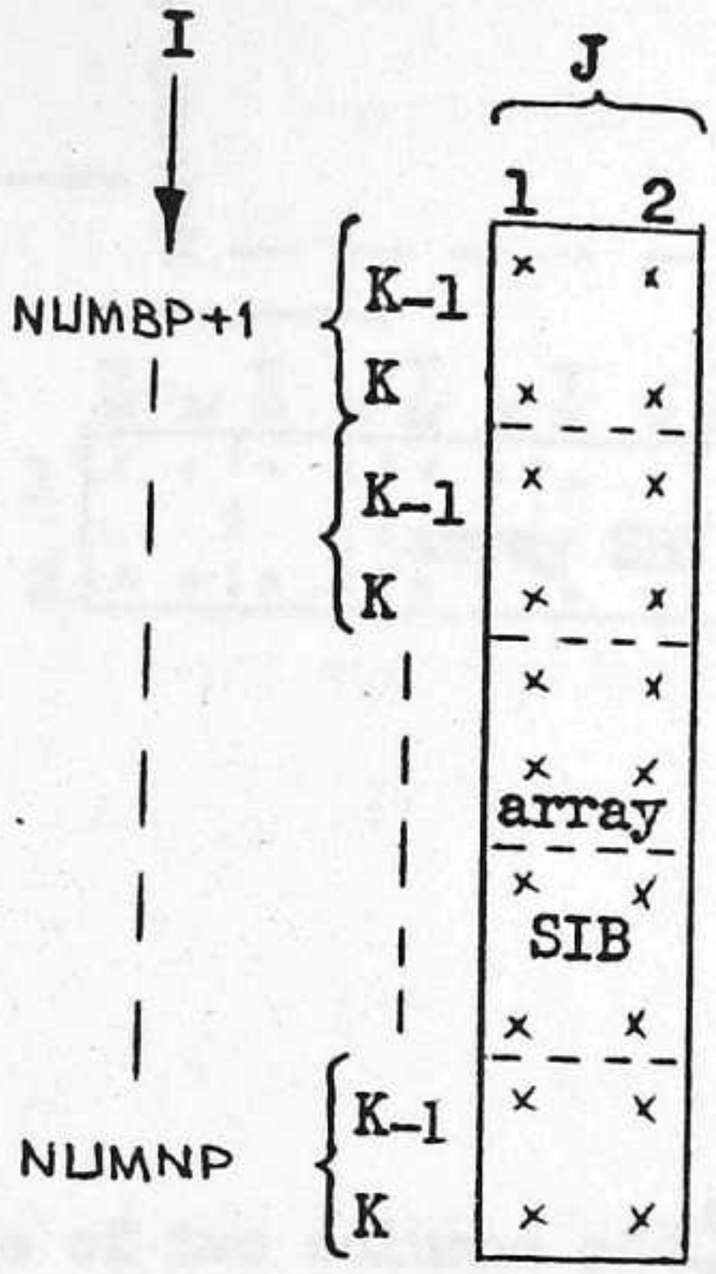


Fig. 66 storage of two columns of $K_{i\ell}^{(r)}$. Integer variable K is updated by two each time a 2x2 submatrix is stored.

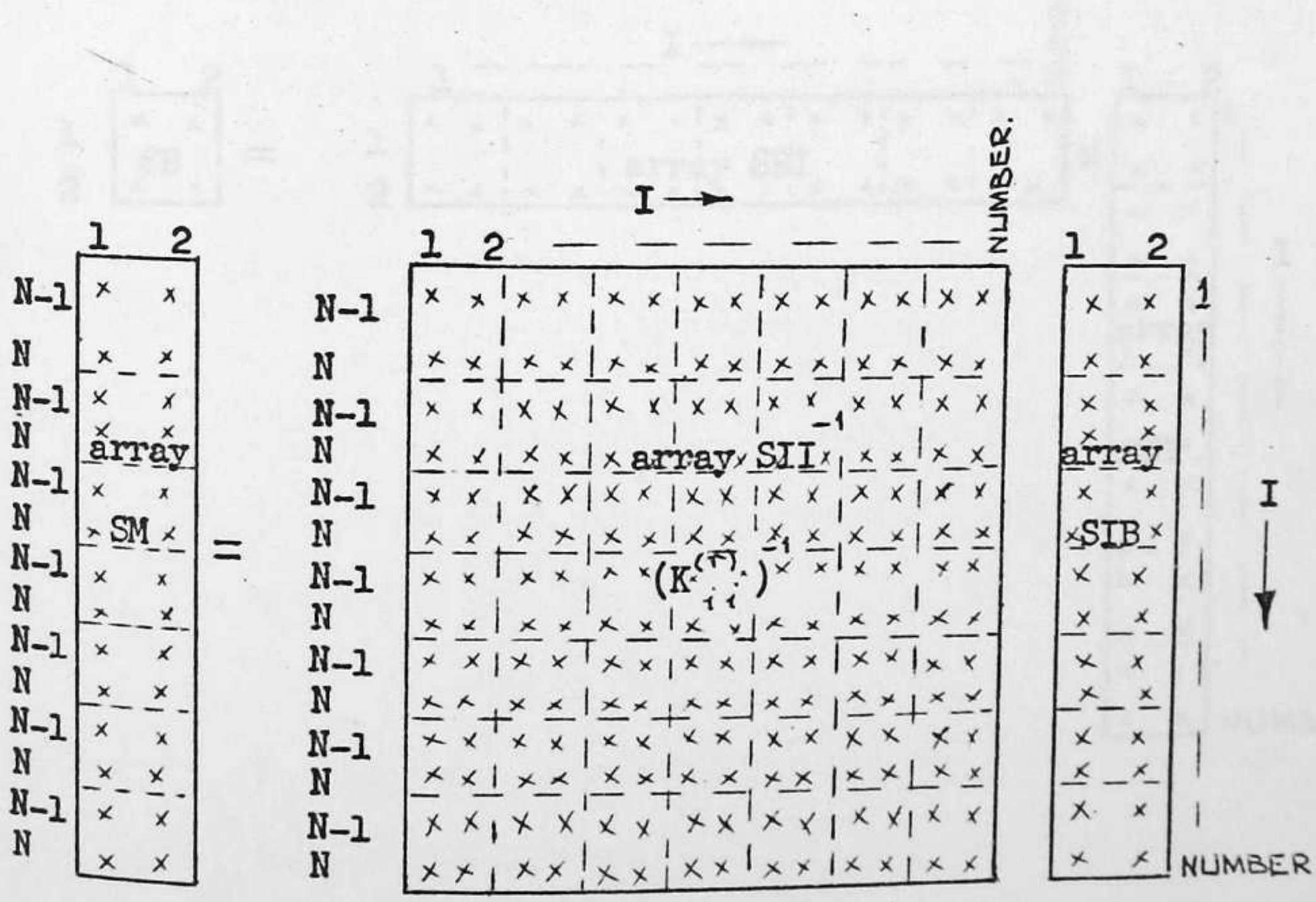


Fig. 67 formation of two columns of $(K_{ii}^{(r)})^{-1}K_{i\ell}^{(r)}$. Integer variable N is updated by two each time a 2x2 sub matrix is stored.

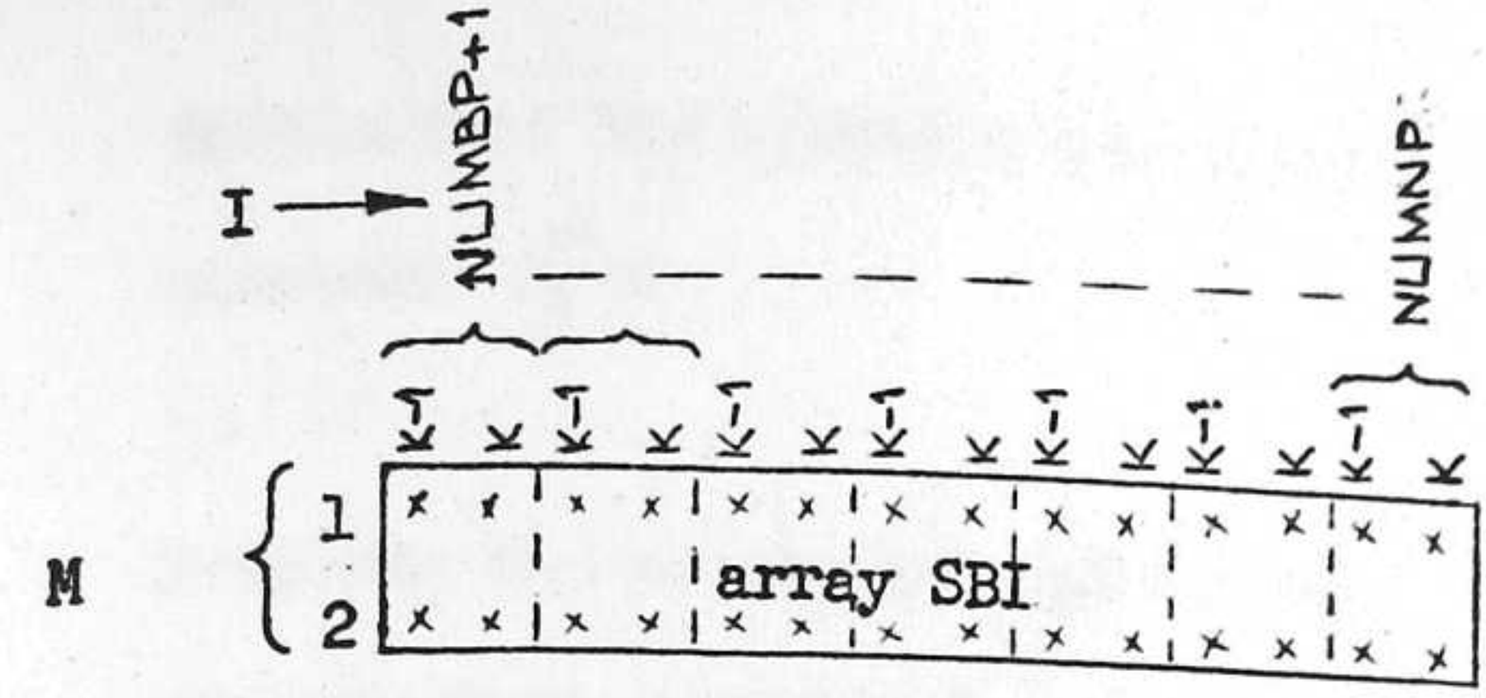


Fig. 68 storage of two columns of $K_{\beta i}^{(r)}$. Integer variable K is updated by two each time a 2×2 submatrix is stored.

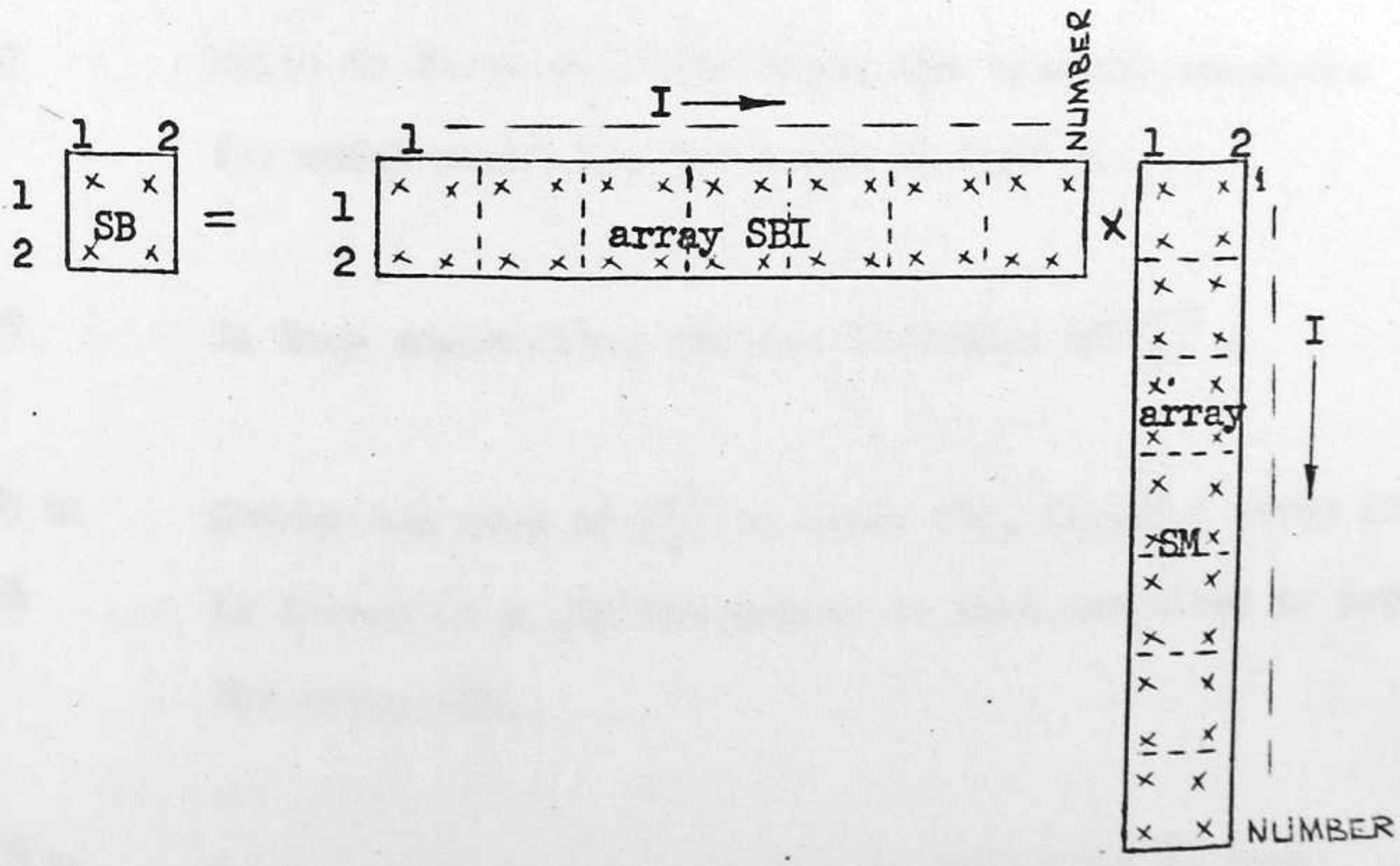


Fig. 69 storage of 2×2 submatrix of $K_{\beta i}^{(r)} (K_{ii}^{(r)})^{-1} K_{i\beta}^{(r)}$

line no.

0287 to 0288 Initializes the boundary reactions $R_{\ell}^{(r)}$ necessary to maintain $U_{\ell}^{(r)}=0$

0291 to 0296 Preforms the matrix multiplcation$(K_{ii}^{(r)})K_{i\ell}^{(r)}P_{i}^{(r)}$to obtain the boundary reaction $R_{\ell}^{(r)}$ (eq.(62)), for nodal point II.

Since $K_{\ell i}^{(r)} = (K_{i\ell}^{(r)})^{T}and(K_{ii}^{(r)})^{-1}$is symmetric. Therefore $((K_{ii}^{(r)})^{-1} K_{i\ell}^{(r)})$ $= K_{\ell i}^{(r)}(K_{ii}^{(r)})^{-1}$

The storage of SM does not actually take place. The effect of the transpose is achieved by arranging the subscripts of arrays PIX, and PIY to select the correct stiffness terms from array SM.

0297 Write to first magnetic tape, the boundary reactions for nodal point II, for input to PART B.

0298 Do loop controlling the row formation of $K_{\ell i}^{(r)}$.

0299 to 0318 Stores two rows of $K_{\ell i}^{(r)}$ in array SBI, fig.68 . Array SBI is formed in a similar manner to that described on page 133 for array SIB.

0319 to 0328 Preforms the matrix multiplcation$K_{\ell i}^{(r)}(K_{ii}^{(r)})K_{i\ell}^{(r)}$to form a 2×2 submatrix of$K_{\ell i}^{(r)}(K_{ii}^{(r)})K_{i\ell}^{(r)}$and stores this in 2×2 array SB. Fig. 69 .

line no.

0329 to 0342 — Preforms the final formation of the 2×2 submatrix of $K_{\ell}^{(r)}$ by subtracting the contents of array SM from the associated stiffness terms of $K_{\ell\ell}^{(r)}$.

0329 to 0331 — Selects the required stiffness terms of $K_{\ell\ell}^{(r)}$ from SXX, SXY, SYX, and SYY.

If (NP(MM,L) = JJ) then control passes to lines 0337 to 0342.

If the do loop is completely executed then we have zero cross stiffness terms and control passesto lines 0332 to 0336

The contents of array SB , is now a 2×2 submatrix of $K_{\ell}^{(r)}$

0343 — Output array SB to first magnetic tape for input to PART B.

Looping back to line 0298 NUMBP-1 times, a complete column of $K_{\ell}^{(r)}$ is formed and outputed to the first magnetic tape.

Looping back to line 0252 and repeating the sequence NUMBP-1 times forms the complete substructure boundary stiffness matrix for the (r) substructure.

Looping back to line 0032, NUMSUB-1 times forms $K_{\ell}^{(r)}$ for each substructure.

When the run is complete line 0353 outputs on line printer PART A. COMPLETE.

0354 — Closes the magnetic tape file.

line no.

0355 to Format statements.

0395

0396 to Terminates PART A. program run.

0398

Fig. 70 indicates the main steps of program PART B. of the computer analysis. The main object of this program is to obtain the boundary nodal point displacements.

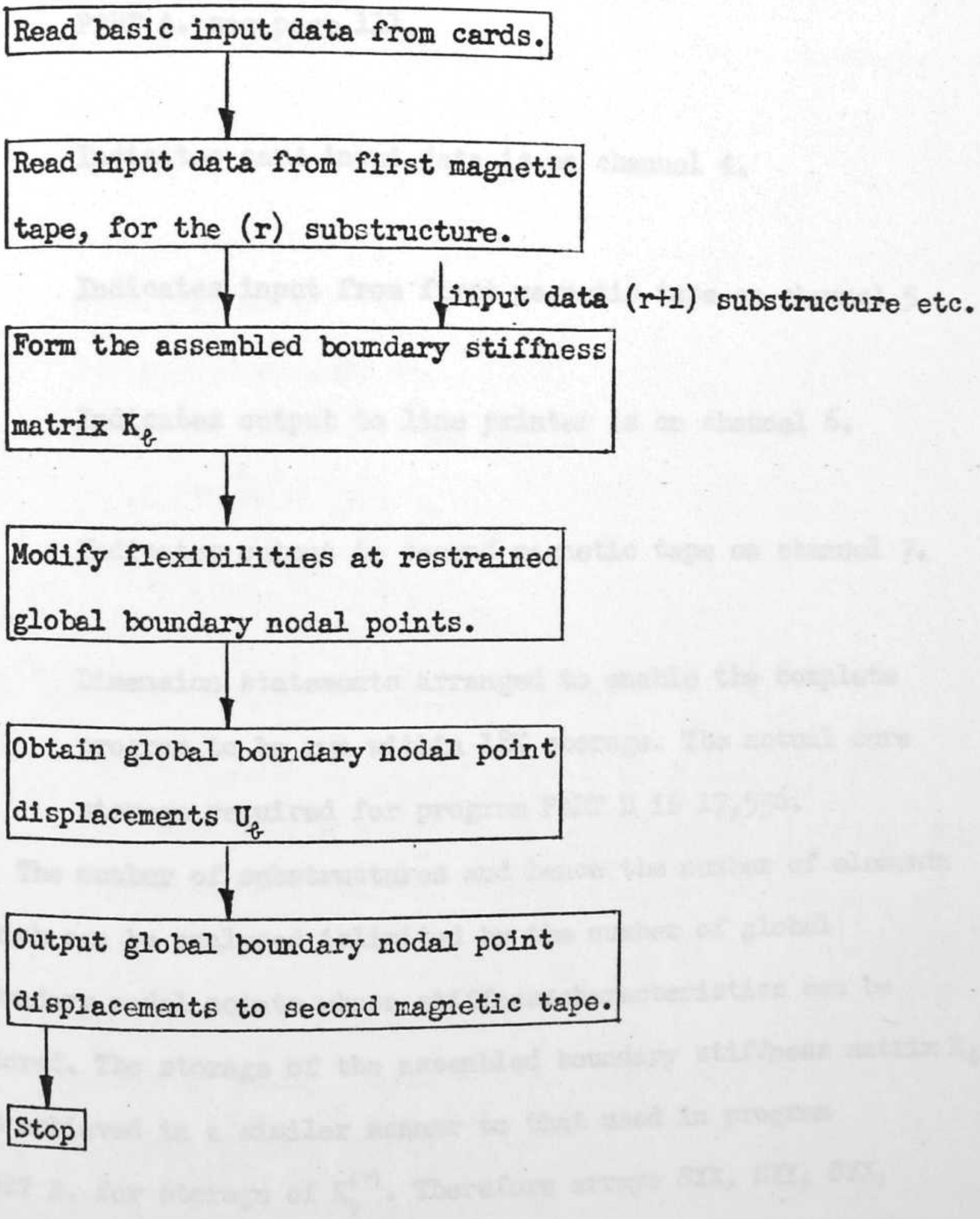


Fig. 70 flow diagram for program PART B.

Refer to pages212to225for line print out of program PART B.

line no.

0005 Compress integer and logical.

This statement has the same meaning as in program PART A. see page 111.

0006 Indicates card input data is on channel 4.

0007 Indicates input from first magnetic tape on channel 5.

0008 Indicates output to line printer is on channel 6.

0009 Indicates output to second magnetic tape on channel 7.

0017 to 0020 Dimension statements arranged to enable the complete program to be run within 18K storage. The actual core storage required for program PART B is 17,536.

The number of substructures and hence the number of elements which can be analysed islimited by the number of global boundary nodal points whose stiffnesscharacteristics can be stored. The storage of the assembled boundary stiffness matrix K_ℓ is achieved in a similar manner to that used in program PART A. for storage of $K_\ell^{(\tau)}$. Therefore arrays SXX, SXY, SYX, and SYY are again used to store stiffness terms, and Identity array NP is again used to record the position of adjacent nodal point. The dimensions of these arrays will however be considerably larger.

In this program, 54 global boundary nodal points, and up to 23 adjacent global boundary nodal points to any one global boundary nodal point may be specified.

The dimension of the arrays may be altered at will, providing the computer storage capacity available is not exceeded. Refer to page185 when altering dimension statements.

line no.

0021 Read from card. problem description.

0022 Read from cards

Number of substructures, NUMSUB.

Number of global boundary nodal points, NUMGBP.

0023 Read from cards.

Cycle print interval, NCPIN.

Output interval of results, NOPIN.

Iteration cycle limit, NCYCM.

Iteration tolerance limit, TOLER.

Over-relaxation factor, XFAC.

0024 Read from cards.

Number of global boundary nodal points with non-zero external forces, NONZBF.

Number of global boundary nadal points with non-zero initial displacements, NONZBD.

Number of restrained boundary nodal points, NUMRBP.

line no.

0025 to 0030 — Initialize to zero.

Global boundary nodal point external forces, arrays FBX, and FBY.

Global boundary nodal point displacements, arrays UBX, and UBY.

0032 to 0043 — Do loop to read from cards.

The non-zero initial estimates of the global boundary nodal point displacements U_{ℓ} , and to store these displacements into arrays UBX, and UBY, using the global boundary nodal point number held by NN.

Arrays UBX, and UBY will now contain the X and Y direction, initial global boundary nodal point displacements respectively; required for the first cycle of the iteration.

0045 — Read from cards for each restrained boundary nodal point.

Boundary nodal point number (global numbering), array NRBP.

Boundary condition code (see page 147), array NFIX.

Slope ϕ of fig. 28, array SLOPE.

0046 to 0053 — Initialize to zero.

Arrays SXX, SXY, SYX, and SYY for storage of non-zero stiffness terms of K_{ℓ} ,

Array NP for storage of adjacent nodal points.

0054 — Initializes the first column of array NP, to give a consecutive column of numbers as required by fig. 20 .

line no.

0056 Write to second magnetic tape for input to program PART C.

0057 to 0064 Output the input information to line printer.

0065 Read from first magnetic tape.

Young's modulus, E.

Poisson's ratio, XU.

These constants will not be used in this program.

0066 to 0102 Do loop controling the assambly of the assembled boundary stiffness matrix $K_b^{(r)}$.

0067 Read from first magnetic tape

Substructure number, NSUB.

Number of elements within substructure, NUMEL.

Number of nodal points,within substructure, NUMNP.

Number of substructure boundary nodal points within substructure, NUMBP.

0068 Read from first magnetic tape,

Element numbers with associated nodal point numbers.

This information is not required by program PART B, and is therefore not stored.

0069 Read from first magnetic tape,

Nodal point numbers with associated X and Y co-ordinates.

This information is not required by program PART B, and is therefore not stored.

line no.

0070 to 0074 — Do loop to initialize and read from first magnetic tape arrays NOTE, and NCODE for the (r) substructure.

0075 to 0078 — Do loop to initialize array SB, which is used to temporary store two rows of $K_{\ell}^{(r)}$.

0079 — Computes the size of array $K_{ii}^{(r)}$.

0081 — Read from first magnetic tape;

The interior nodal point displacements $U_{i}^{(r)}{}_{u_{\ell}=0}$

This information is not required for program PART B and is therefore not stored.

0082 to 0102 — Formation of assembled boundary stiffness matrix K_{ℓ} for the complete structure as indicated on pages 58 to 62 only the non-zero stiffness terms of K_{ℓ} are stored in the computer under arrays SXX, SXY, SYX, and SYY.

The assembly process for K_{ℓ} is similar to that described on pages119to122 for the assembly of $K^{(r)}$.

The only differences are

a/ Instead of assembling stiffness terms of an element with three nodal points, we have a substructure (which can be considered to be a large complex element) having a total of NUMBP, nodal points. Hence integer variables L (do loop line 0082) and M

(do loop line 0091) range from 1 to NUMBP.

b/ The stiffness terms will be stored in global numbering order therefore reference is made to array NCODE when assembling terms. Array NCODE has the same function as array LM in program PART A.

line no.

0083 LX stores the global boundary nodal point number under consideration.

0084 Read from first magnetic tape.

Two columns of $(K_{ii}^{(r)})^{-1} K_{ib}^{(r)}$.

This information is not required for program PART B, and is therefore not stored.

0085 Read from first magnetic tape.

The boundary reaction $R_b^{(r)}$ for nodal point number LX.

0086 and 0087 Subtracts the boundary reaction $R_b^{(r)}$ (just read from magnetic tape) from the global boundary nodal point external force P_b at global boundary nodal point LX.

When every substructure has been considered, arrays FBX and FBY will contain the resultant boundary nodal point forces in X and Y directions respectively as required by eq. (65).

0088 to 0090 Do loop to read from first magnetic tape, and store in array SB, two rows of the substructure boundary stiffness matrix $K_b^{(r)}$ for the (r) substructure.

$K_b^{(r)}$ as output from program PART A. is in column order.

Since $K_b^{(r)}$ is symmeteic then $K_b^{(r)} = K_b^{(r)T}$. Line 0089 has been arranged

to read the 2×2 submatrix of $K_\ell^{(r)}$ and to automatically transpose the submatrix into array SB.

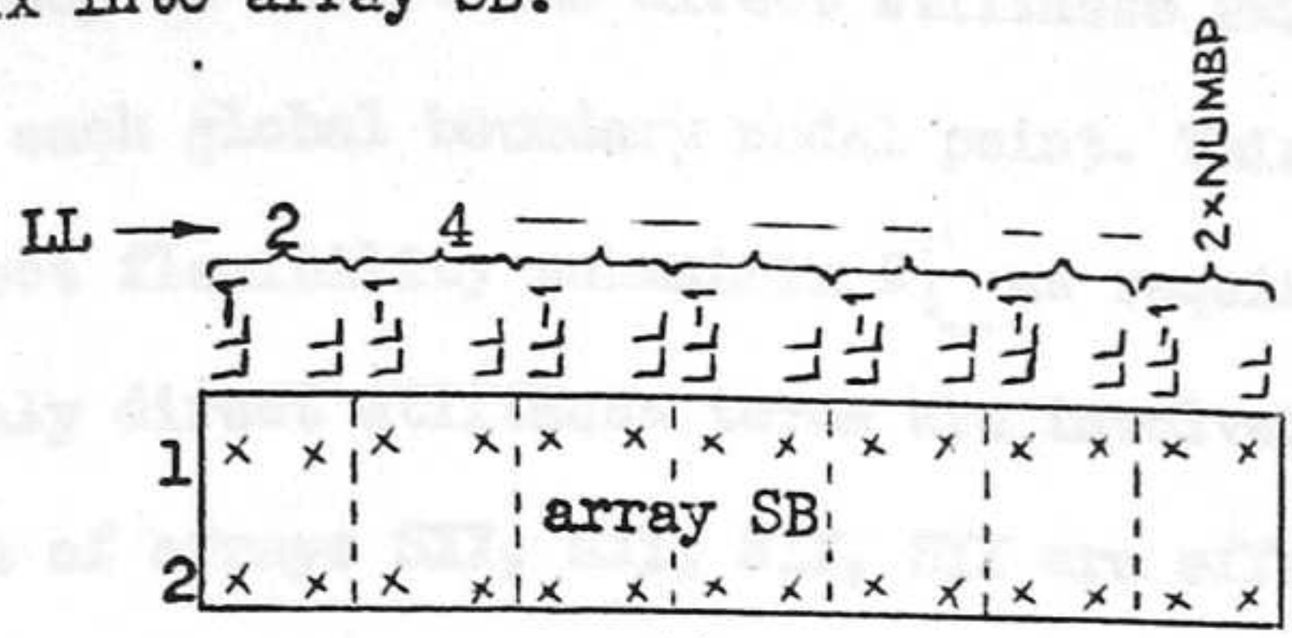


Fig. 71 storage of two rows of $K_\ell^{(r)}$ integer variable LL is updated each time a 2×2 submatrix is stored.

line no.

0091 to 0102 Assemble the two rows of $K_\ell^{(r)}$ (just stored in array SB) into arrays SXX, SXY, SYX, and SYY.

The assembly procedure is basicly identical to that used in program PART A. to assemble $K^{(r)}$, and described on pages 119 to 122 The main difference is that reference is made to array NCODE rather than array LM when comparing nodal point numbers.

Looping back to line 0082 NUMBP-1 times effectively assembles the non-zero terms of the (r) substructure boundary stiffness matrix $K_\ell^{(r)}$ into arrays SXX, SXY, SYX, and SYY.

Looping back to line 0066 NUMSUB-1 times effectively assembles the non-zero stiffness terms of each substructure boundary stiffness matrix to form the non-zero terms of K_ℓ the assembled boundary stiffness matrix.

0103 to 0107 Do loop to count and store in array NAP, the number of adjacent nodal points + 1 for each global boundary nodal point in the complete structure.

line no.

0108 to 0114 Do loop to invert the direct stiffness 2×2 submatrix for each global boundary nodal point. This obtains the direct flexibility submatrix $K^{-1}_{\ell_{nn}}$ as required by eq. (70).

Since only direct stiffness terms are involved then only the first column of arrays SXX, SXY, SYX, SYY are effected.

0115 to 0132 Do loop to modify the direct flexibilities to take account of boundary conditions.

0116 Integer variable M stores the global boundary nodal point flexibility being modified.

0117 Puts the value held in the M row and first column of the identity array NP to zero. The necessity for this will be indicated in the iteration process.

0118 Control statement, for type of boundary restraint.

If NFIX = 0

We have the case of a nodal point fixed in both X and Y directions. Control passes to line 0128 and the terms of eq. (82) become

$$f^*_{xx} = f^*_{xy} = f^*_{yx} = f^*_{yy} = 0.0$$

If NFIX = 1

We have the case of $\phi = 90^\circ$ ie the nodal point is fixed in the X direction. (since the computer cannot handle $\text{TAN}\phi = \infty$) Control passes to line 0126 and the terms of eq. (82) become

$$f^{*}_{yy} = f_{yy} - \frac{f_{yx}\, f_{xy}}{f_{xx}}$$

$$f^{*}_{xx} = f^{*}_{xy} = f^{*}_{yx} = 0.0$$

If NFIX = 2

We have the case of a nodal point restrained to move at an angle ϕ (see fig.28) where $0.0 \leqslant \phi < \pi/2$ Control passes to lines 0119 where the modified flexibility terms of eq. (82) are computed.

line no.

0133 to 0164 Preforms the GAUSS - SEIDEL iteration process on the global boundary nodal point displacements.

0133 to Initializes to zero.

0135 The iteration cycle counter, NCYCLE.

Print interval cycle counter, NUMPT.

Output interval of results counter, NUMOPT.

0136 Initializes to zero.

The sum of the force unbalance, SUM at the start of each iteration cycle.

0137 Do loop controlling the iteration process.

M stores the global boundary nodal point under consideration.

0138 NUM stores the number of adjacent boundary nodal points+1 associated with global boundary nodal point M.

lind no.

0139 If (SXX(M,1)+SYY(M,1)) = 0 then we are dealing with a nodal point fixed in both X and Y directions and therefor there is to be no change of displacement in either direction. Control immediately by-passes the iteration to line 0152.

0140 and 0141 Stores under FRX and FRY the resultant boundary force in X and Y directions respectively, for global boundary nodal point M.

0142 to 0145 Do loop to preform the matrix multiplcation and summation

$$\left(F_{\ell_n} - \sum_{i=1,n-1} K_{\ell_{ni}} U_{\ell_i}^{(s+1)} - \sum_{i=n+1,N} K_{\ell_{ni}} U_{\ell_i}^{(s)}\right)$$

as required by eq. (74).

0146 and 0147 Completes eq. (74) to form the change in displacement $\Delta U_{\ell_n}^{(s,s+1)}$ between successive. cycles of iteration.

0148 and 0149 Computes the new displacement $U_{\ell_n}^{(s+1)}$ from eq.(75) and stores this in arrays UBX and UBY (the X and Y directions respectively)

0151 Computes the force unbalance at nodal point number M. and adds this to the contents of SUM.

The force unbalance is obtained by dividing $\Delta U_{\ell_n}^{(s,s+1)}$ the difference in displacement between cycles of iteration by the nodal point flexibilities SXX(M,1) and SYY(M,1). The force unbalance of restrained nodal points is not computed (controlled

by line 0150) since SXX(M,1) and SYY(M,1) may be zero for these nodal points.

line no.

0153 Iteration cycle counter, NCYCLE is updated when one cycle of iteration is complete ie M = NUMGBP.

0154 If NCYCLE = NUMPT the cycle number NCYCLE and the sum of the force unbalance is outputed on the line printer.

0155 Print interval cycle counter, NUMPT is updated.

0158 The sum of the force unbalance is compared with the tolerance limit, TOLER (accuracy factor).

If SUM $\leqslant$ TOLER the iteration has converged, and the final global boundary nodal point displacements are printed on line printer (line 0166) and second magnetic tape (line 0179) for input to program PART C.

If SUM > TOLER control passes to line 0159.

0159 If NCYCM = NCYCLE then the iteration has failed to converge within the specified iteration cycle limit, NCYCM.

The latest results are printed on the line printer and on the second magnetic tape (line 0170). The program run is terminated by line 0177.

If NCYCM≠NCYCLE then control passes to line 0160.

0160 If NCYCLE = NUMOPT then control immediately passes

to line 0136. SUM is initializes and a new iteration cycle started.

If NCYCLE = NUMOPT then a print out of the latest results is obtained on line printer before control is passes to line 0136.

line no.	
0180 and 0181	Closes the magnetic tape files.
0182	Prints on line printer PART B. COMPLETE. this statement is printed only if the iteration has successfully converged.
0183 to 0224	Format statements.
0225 to 0227	Terminates PART B. program run.

The limitation on the number of elements which may be incorporated into a structure using substructure analysis will be dependent on the maximum number of global boundary nodal points which can be accomodated by the program. The actual restriction being due to the storage requirements of K_{ℓ} the assembled boundary stiffness matrix. In the programs just described the number of global boundary nodal points was restricted to 54 and the number of adjacent global boundary nodal points to any one global boundary nodal point is 23 This keeps the program PART B to within 18K.

To improve on these figures the following program arranges to place K_{ℓ} on Direct Access Disc storage(backing store). Since there will be a relatively small number of global boundary nodal points compared to the total number of nodal points in the complete structure, there is a good chance that the time required to obtain a solution may not be excessive.

The only part of the computer analysis affected is program PART B. All input information, and programs PART A and C are unchanged.

Refer to pages 226 to 247 for the line print out of program PART B with disc storage.

The flow diagram for this program is exactly the same as that outlined in fig. 70 .

The only difference in the program listing is in the method of handling the arrays SXX, SXY, SYX, SYY, and NP. The terms of which represent the non-zero values of K_{ℓ} and these have to be placed on the backing store.

Five scratch files, channel numbers 1 to 5, each of length 10000 have been allocated to store arrays SXX, SXY, SYX, SYY, and NP on backing store respectively.

The arrays are transfered by columns to the backing store. Therefore the form of the arrays may be represented by fig. 72. (this may be compared with fig's.20 and 21 .)

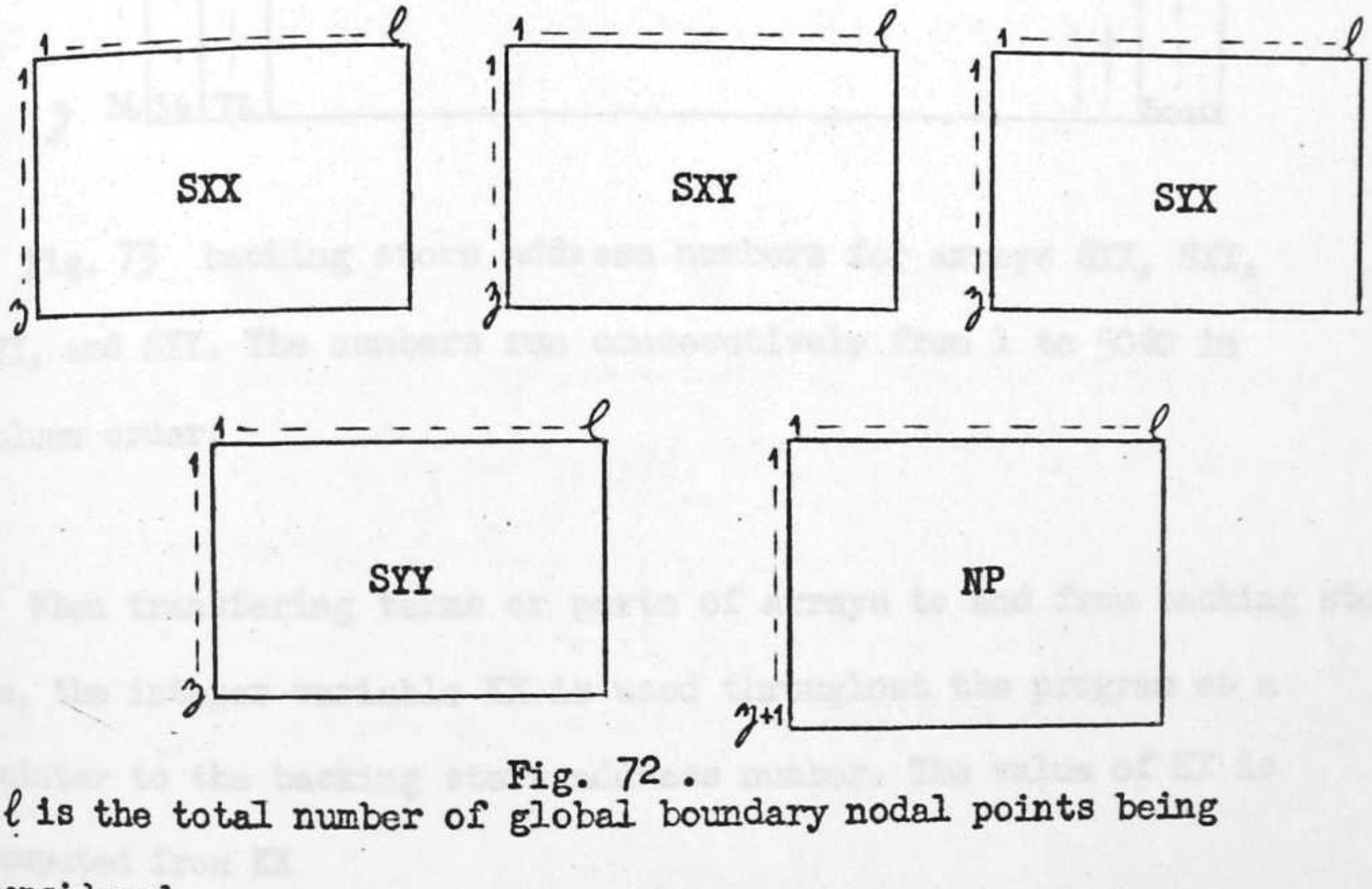


Fig. 72 .

ℓ is the total number of global boundary nodal points being considered.

$\mathcal{Z} = x+1$ where x is the maximum number of adjecent global boundary nodal points that any one global boundary nodal point is likely to have.

In the present program ℓ and x have been limited to 140 and 35 respectively.

Each term of an array read to a backing store file is allocated an address number. For arrays SXX, SXY, SYX, and SYY the address numbering order of their respective channel numbers is as shown in fig. 73 . For array NP the columns will be of length 37 and the address numbers will run consecutively from 1 to 5180.

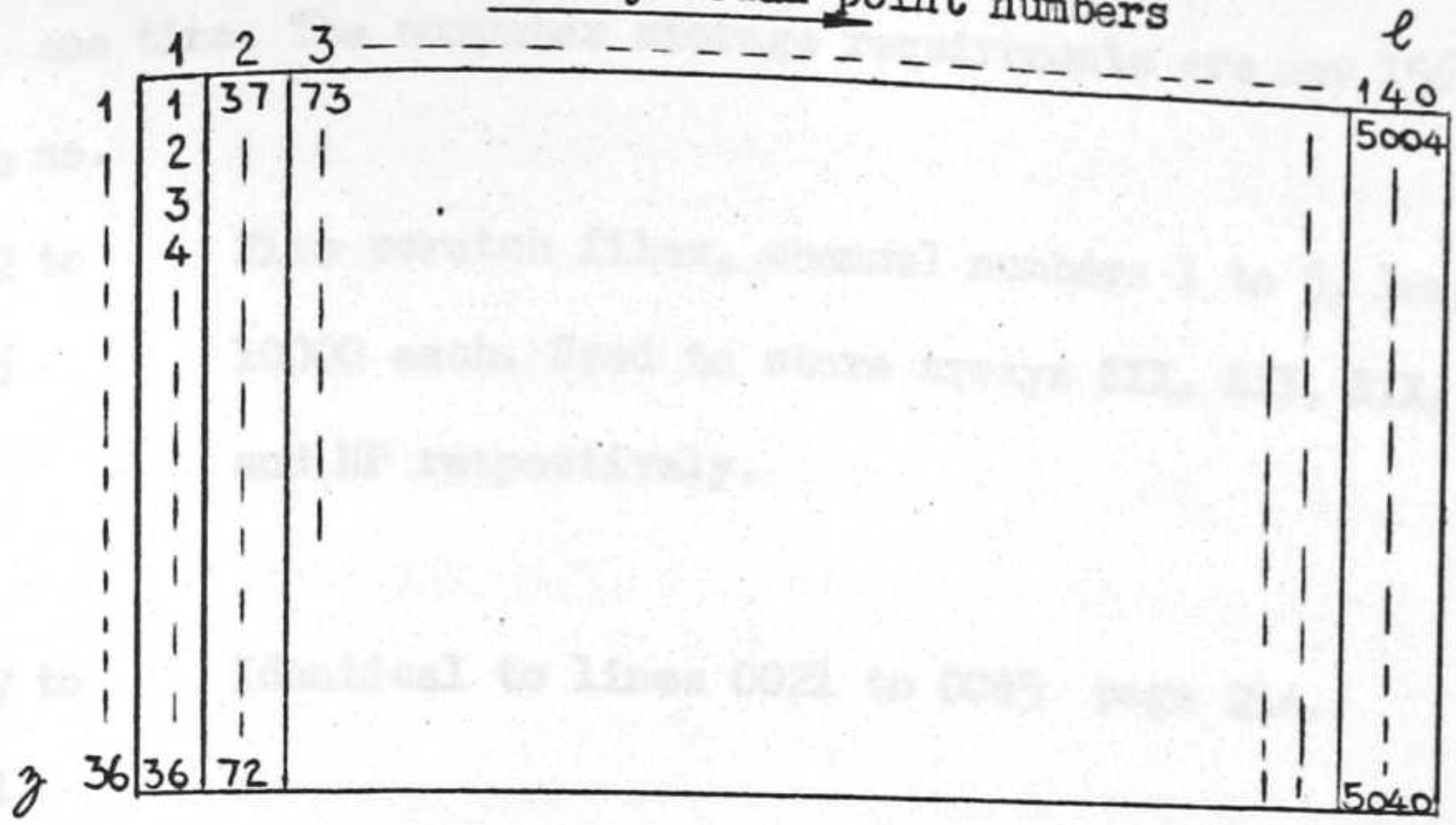


Fig. 73 backing store address numbers for arrays SXX, SXY, SYX, and SYY. The numbers run consecutively from 1 to 5040 in column order.

When transfering terms or parts of arrays to and from backing store, the integer variable KX is used throughout the program as a pointer to the backing store address number. The value of KX is computed from KK

Where KK is the address number of that term at which the start of a transfer of terms is to take place.

lines no.

0018 to 0021 Dimension statements, these now allow up to 140 global boundary nodal points, and the maximum number of adjacent global boundary nodal points to any one global boundary nodal point is now 35.

To keep the total number of transfers of information between backing store down to a minimum, the stiffness terms of 20 global boundary nodal points are held in core at any

one time. The computer storage requirements are now 15680.

line no.

0022 to 0026 — Five scratch files, channel numbers 1 to 5, length 10000 each. Used to store arrays SXX, SXY, SYX, SYY, and NP respectively.

0027 to 0051 — Identical to lines 0021 to 0045 page 214.

0052 to 0082 — Have the same result as lines 0046 to0054 page 215.

Lines 0068 to 0079 arrange to transfer 20 contiguous columns of arrays SXX, SXY, SYX, SYY, and NP to backing store.

0083 to 0092 — Identical to lines 0055 to 0064 page 216.

0093 to 0155 — Have the same result as lines 0065 to 0102 page 216.

Lines 0119 to 0130 arrange to transfer 1 complete column of arrays SXX, SXY, SYX, SYY, and NP from backing store.

Lines 0143 to 0154 arranges to transfer 1 complete column to backing store.

0156 to 0163 — Have the same result as lines 0103 to 0107 page 218.

Lines 0158 to 0160 arranges to transfer 1 complete column of array NP from backing store.

lines no.

0164 to 0188 Have the same result as lines 0108 to 0114 page 218.
Lines 0165 to 0173 arrange to transfer 1 term of each array SXX, SXY, SYX, and SYY from backing store.
Lines 0180 to 0187 arrange to transfer 1 term of each array SXX, SXY, SYX, and SYY to backing store.

0189 to 0227 Have the same result as lines 0115 to 0132 page219.
Lines 0191 to 0199 arrange to transfer 1 term of each array SXX, SXY, SYX, and SYY from backing store.
Lines 0201 to 0203 arrange to transfer 1 term of array NP to backing store.
Lines 0218 to 0226 arrange to transfer 1 term of each array SXX, SXY, SYX, and SYY to backing store.

0228 to 0298 Have the same result as lines 0133 to 0179 page 220.
Lines 0235 to 0246 arrange to teansfer 20 contiguous columns of arrays SXX, SXY, SYX, and SYY from backing store.

0299 to 0303 Closes the direct access disc backing store files;

0304 and 0305 Closes the magnetic tape files

0306 Prints on line printer PART B. COMPLETE.
This will only be printed if the iteration has

successfully converge.

0307 to 0348 — Format statements.

0349 to 0351 — Terminates PART B program run.

3.4
Program PART C.

Fig 74 indicates the main steps of program PART C. of the computer analysis. The main object of this program is to obtain the element stresses, and nodal point displacements for each substructure, and to output these results on the line printer.

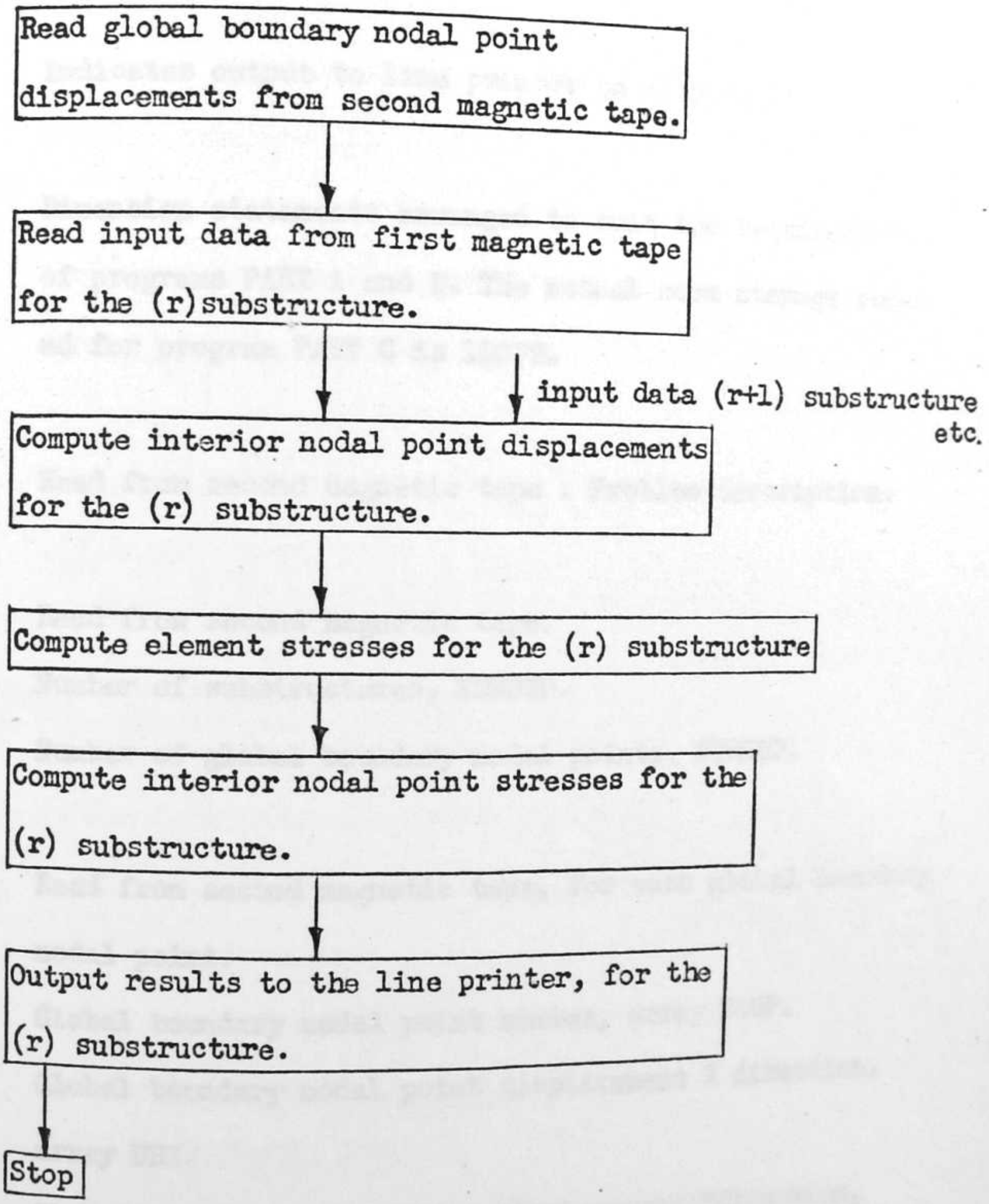


Fig. 74 flow diagram for program PART C.

Refer to pages 248 to 259 for line print out of program PART C.

line no.

0005 Indicates input from second magnetic tape on channel 4.

0006 Indicates input from first magnetic tape on channel 5.

0007 Indicates output to line printer on channel 6.

0015 to 0019 Dimension statements arranged to suit the requirements of programs PART A and B. The actual core storage required for program PART C is 11072.

0020 Read from second magnetic tape . Problem description.

0021 Read from second magnetic tape.

Number of substructures, NUMSUB.

Number of global boundary nodal points. NUMGBP.

0022 Read from second magnetic tape, for each global boundary nodal point.

Global boundary nodal point number, array NGBP.

Global boundary nodal point displacement X direction, array UBX.

Global boundary nodal point displacement Ydirection, array UBY.

0023 Read from first magnetic tape.

Youmg's modulus, E.

Poisson's ratio, XU.

line no.

0024 to 0158 Do loop controlling the formation of the results of each substructure.

0025 Read from first magnetic tape for the (r) substructure.

Substructure number, NSUB.

Number of elements, NUMEL.

Number of nodal points, NUMNP.

Number of boundary nodal points, NUMBP.

0026 Read from first magnetic tape for each element in the (r) substructure.

Element number, NUME.

i, nodal point number, NPI.

j, nodal point number, NPJ.

k, nodal point number, NPK.

0027 Read from first magnetic tape for each nodal point in the (r) substructure.

Nodal point number,(substructure number),NPNUM.

X---co-ordinate, XORD.

Y----co-ordinate, YORD.

0028 Read from first magnetic tape for the (r) substructure.

Substructure boundary nodal point numbers, array NOTE.

Corresponding global boundary nodal point numbers, array NCODE.

line no.

0029 to 0032 Initialization of nodal point displacements, array UX and UY.

The arrays UX and UY will eventually hold the complete set of nodal point displacements for the (r) substructure.

0033 Computes NUMBER = twice the number of interior nodal points in the (r) substructure.

0034 to 0039 Do loop to read from first magnetic tape, for the (r) substructure.

$U_{i\,u_\ell=0}^{(r)}$ the interior nodal point displacement with the boundary fixed, UIX and UIY and to store these displacements into arrays UX and UY respectively, using substructure numbering order.

0040 to 0045 Do loop to read matrix $(K_{ii}^{(r)})^{-1} K_{i\ell}^{(r)}$ from first magnetic tape for the (r) substructure, and to store this in array SM.

0041 Read from first magnetic tape, two columns of $(K_{ii}^{(r)})^{-1} K_{i\ell}^{(r)}$

0042 Read from first magnetic tape the boundary reaction $R_\ell^{(r)}$

These terms are not required by this program and are therefore not stored.

line no.

0044 Read from first magnetic tape.

Substructure boundary stiffness matrix $K_{\ell}^{(r)}$.

These terms are not required by this program and are therefore not stored.

0046 to 0054 Do loop to compute the difference in element coordinates from eq.(90) for the (r) substructure.

0055 to 0060 Do loop to store the global boundary nodal point displacements into arrays UX and UY.

J stores the substructure boundary nodal point number.

JJ stores the global boundary nodal point number.

Arrays UX, and UY now hold , in substructure numbering.

a/ The (r) substructure boundary nodal point displacements $U_{\ell}^{(r)}$ in X and Y directions respectively.

b/ The (r) substructure interior nodal point displacements $U_{i}^{(r)}_{U_{\ell}=0}$ with the boundary fixed.

0062 to 0072 Do loop to preform the matrix multiplcation and subtraction $U_{i}^{(r)}_{U_{\ell}=0} - (K_{ii}^{(r)})^{-1} K_{i\ell}^{(r)} U_{\ell}^{(r)}$ from eq.(68) and hence obtain $U_{i}^{(r)}$, the (r) substructure interior nodal point displacementsin both X and Y directions with the boundary relaxed.

Arrays UX and UY now contain the (r) substructure nodal point displacements in X and Y directions respectively.

line no.

Output to line printer.

0077 The nodal point displacements for the (r) substructure.

0080 to Do loop to form the element stresses, using the
0106 relationship of eq,(13) for elements within bounds of the (r) substructure.

0081 to I, J, and K hold the nodal point numbers associated
0083 with the (e) element.

0084 to Preforms the matrix multiplcation $2A_{ijk}[A][U]$.
0087 The [A] matrix of eq.(11) relates elements strains and element nodal point displacements.

0088 Computes the constants term

$$COMM = \frac{E}{(1-\nu^2)2A_{ijk}}$$

0089 to Computes the stresses of the (e) element using the
0091 relationship of eq.(13).

X stores the normal stress in the X direction. σ_x

Y stroes the normal stress in the Y direction. σ_y

XY stores the shear stress. τ_{xy}

0092 to Computes the maximum and minimum principal stresses
0104 and their direction, from elementary mechanics.

XMAX stores the max. principal stress.

YMAX stores the min. principal stress.

PA stores the direction of the max. principal stress. 164
from the X axis.

line no.

0092 to 0094 Arrays SIGXX, SIGYY, SIGXY store the element stresses σ_x, σ_y, and τ_{xy} respectively, for use when computing interior nodal point stresses.

0105 Output to line printer .

The element stresses for the (e) element of the (r) substructure.

0109 to 0155 Do loop to compute the interior nodal point stresses of the (r) substructure.

0110 to 0113 Do loop to check whether nodal point M under consideration is an interior nodal point .

MM stores the row number of the nodal point M, in array NOTE.

If NCODE(MM) is zero then the nodal point is an interior nodal point.(see fig. 18)

0114 to 0119 Initialization of stress components.

0120 to 0142 Do loop to preform the summation required by eq's(83)(84) and (85).

line no.

0132 and 0133

A1 stores $a^{(e)}$ from eq.(86).

B1 stores $b^{(e)}$ from eq.(87).

0143 to 0145

Computes the interior nodal point stresses from eq.(83), eq.(84) and eq.(85) for interior nodal points within the (r) substructure.

SRX stores S_x from eq.(88).

SRY stores S_y from eq.(89).

R stores N the number of elements adjacent to the interior nodal point under consideration.

X stores the normal stress σ_x

Y stores the normal stress σ_y

XY stores the shear stress τ_{xy}

0146 to 0155

Computes for the interior nodal point M under consideration. The max. and min principal stresses and direction

XMAX stores the max. principal stress

XMIN stores the min. principal stress

PA stores the direction of the max. principal stress from the X axis.

0156

Outputs the interior nodal point stresses to the line printer.

Looping back to line 0024 NUMSUB times, computes the stresses and nodal point displacements for each substructure.

line no.

0159 Prints on line printer PART C. COMPLETE.

0160 and Closes the magnetic tape files.

0161

0162 to Format statements.

0188

0189 to Terminates the program run.

0191

This chapter lists the sequence of operations necessary to analyse a structure and to indicate any limitations to the programs.

The programs use triangular elements and are restricted to plane stress (or plane strain**) structures of constant thickness with in-plane forces, and homogeneous material properties.

The maximum number of elements which can be analysed is limited by the computer storage available. For the programs listed in this thesis the storage requirements are

Program PART A. 16832.

Program PART B. 17536.

Program PART B (disc storage). 15680.

Program PART C. 11072.

For a computer with 18K storage it is possible to analyse a structure (for example see page. 95) with about 500 elements.

If program PART B with disc storage is used then about 1500 elements can be accommodated.

The sequence of operations necessary to analyse a structure is as follows.

1/ Each structure is divided up into a nuber of substructures of quite arbitary shape. Each substuucture is given a unique

** The program may be used for plane strain structures providing that the material elastic constants are modified as follows.

$$E^* = \frac{E}{1-\nu^2} \quad ; \quad \nu^* = \frac{\nu}{1-\nu} \; ;$$

number (the substructure number). The structure may be divided into
any number of substructures providing the following limits listed below under umder paragraphs (a) to (c) are checked out.

The substructure boundary need not conform to the physical boumdary of the structure, nor need the substructure boundary form a closed loop.

2/ Global boundary nodal points are specified along the boundarys of the substructures, numbering from 1 upwards in consecutive order. This is known as global numbering.

The global boundary nodal points must consist of

All restrained nodal points.

All nodal points which are common to more than one substructure.

A nodal point associated to only one substructure may be considered as a boumdary nodal point if so desired.

To analyse a substructure within 18K, the following must be checked out before proceding further.

(a) The number of global boundary nodal points must not exceed 54.

(b) The number of global (or substructure) boundary nodal points associated with any one substructure must not exceed 16. For example substructure 3 fig. 75 has 5 global boumdary nodal points, namely 9,10,11,12,13.

(c) THe number of global boundary nodal points adjacent to any

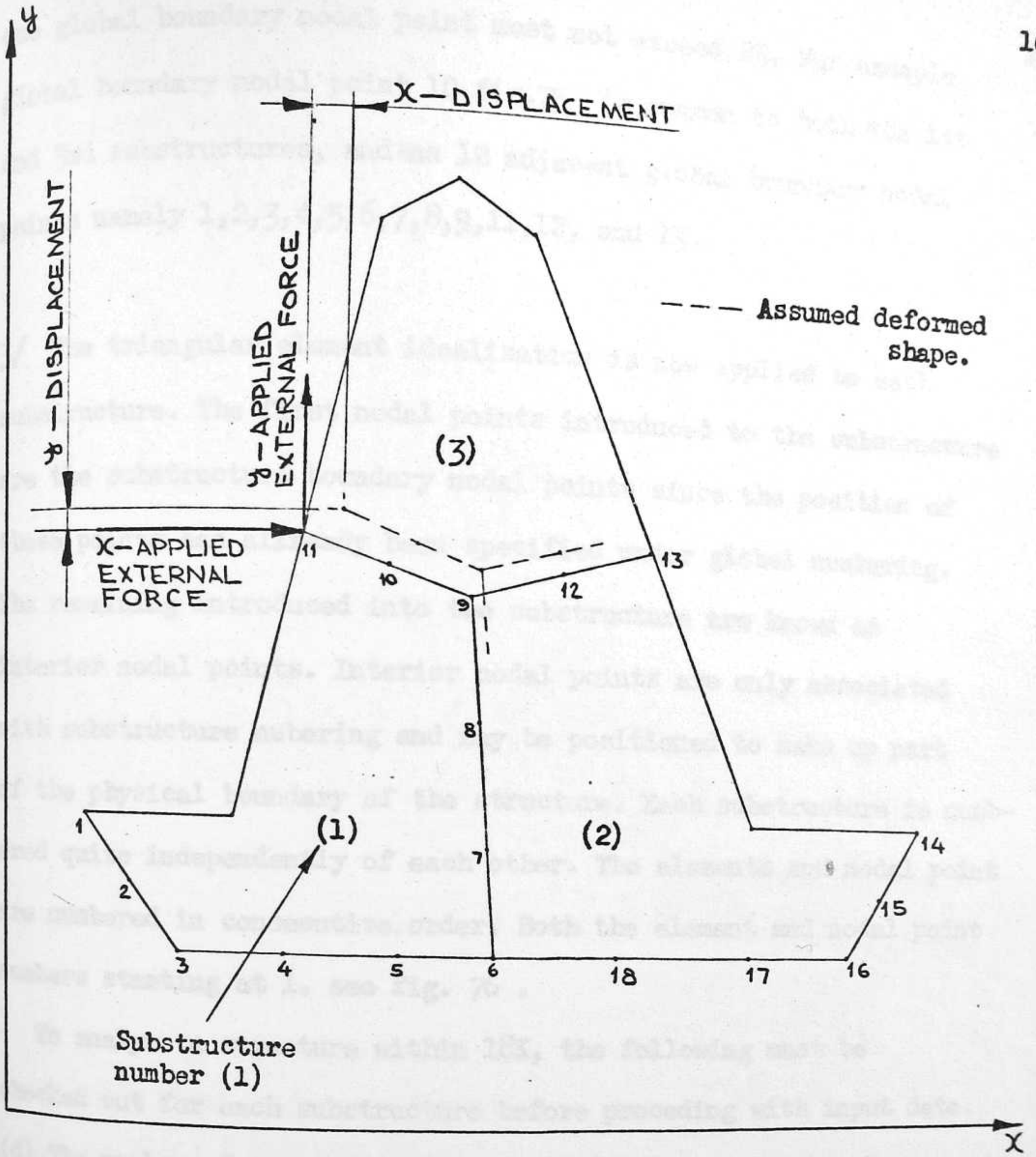


Fig.75 global numbering for a structure divided into three substructures. The directions of forces and displacements shown are positive.

Number of substructures = 3.

Number of global boundary nodal points = 18.

Number of restrained global boundary nodal points = 11.

Restrained global boundary nodal points are 1,2,3,4,5,6,14,15,16, 17, and 18.

one global boundary nodal point must not exceed 23. For example global boundary nodal point 10 fig.75 is common to both the 1st and 3rd substructures, and Has 12 adjacent global boundary nodal points namely 1,2,3,4,5,6,7,8,9,11,12, and 13.

3/ The triangular element idealization is now applied to each substructure. The first nodal points introduced to the substructure are the substructure boundary nodal points since the position of these points has allready been specified under global numbering. The remaining introduced into the substructure are known as interior nodal points. Interior nodal points are only associated with substructure nubering and may be positioned to make up part of the physical boundary of the structure. Each substructure is numbered quite independently of each other. The elements and nodal point are numbered in consecutive order. Both the element and nodal point numbers starting at 1. see fig. 76 .

To analyse a structure within 18K, the following must be checked out for each substructure before proceding with input data.

(d) The number of interior nodal points in any one substructure must not exceed 25.

(e) The total number of interior and substructure boundary nodal points in any one substructure must not exceed 41.

(f) The number of elements in any one substructure must not exceed 65.

(g) The number of adjacent substructure nodal points to any one

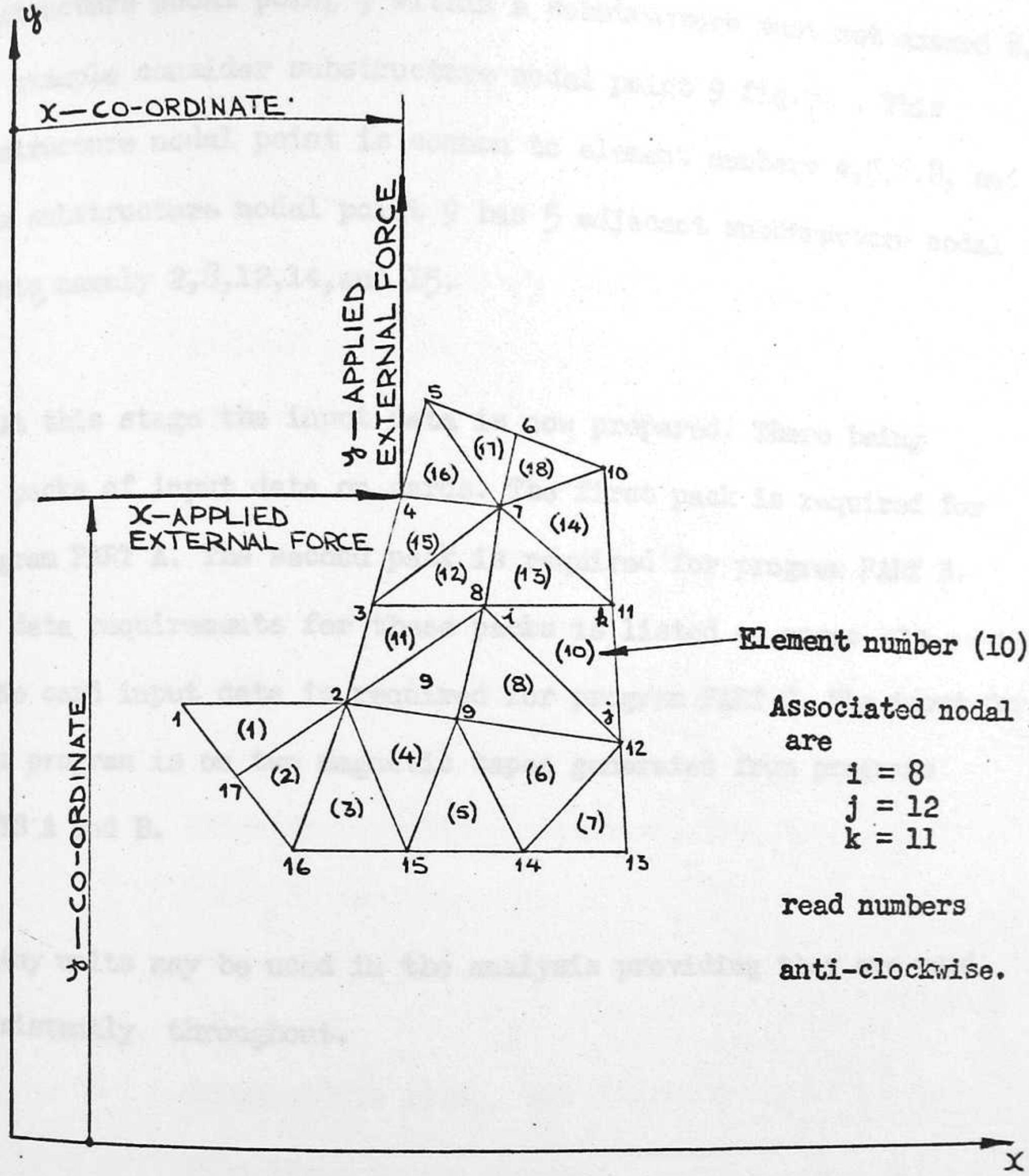


Fig.76 substructure numbers for substructure (1) of fig. 75

The direction of the force shown are positive.

Substructure number = 1.

Number of elements = 18.

Number of nodal points = 17.

Number of substructure boundary nodal points = 11.

Number of interior nodal points = 6.

Substructure boundary nodal points are 1,5,6,10,11,12,13,14,15,16,17.

substructure nodal point , within a substructure must not exceed 8. For example consider substructure nodal point 9 fig.76 . This substructure nodal point is common to element numbers 4,5,6,8, and 9 Thus substructure nodal point 9 has 5 adjacent substructure nodal points, namely 2,8,12,14,and 15.

4/ At this stage the input data is now prepared. There being two packs of input data on cards. The first pack is required for program PART A. The second pack is required for program PART B. The data requirements for these packs is listed on pages 173 TO 181.

No card input data is required for program PART C. The input for this program is on two magnetic tapes generated from programs PARTS A and B.

Any units may be used in the analysis providing they are used consistently throughout.

section

(1) Title card, one card (71H).

columns 2——72 should contain information to be printed with the results, usually job title and name.

(2) Control card, one card free format (2I0).

NUmber of substructures.

Number of global boundary nodal points.

(3) Material properties, one card, free format (2E0.0).

Young's modulus.

Poisson's ratio.

(4) Substructure array, the following refers to one substructure.

(4a) Substructure control card, one card, free format(5I0)

Substructure number.

Number of elements witnin the substructure.

Number of nodal points within the substructure.

Number of substructure boundary nodal points within the substructure.

Number of interior nodal points with non-zero external forces within the substructure.(if there are no interior nodal point external forces, read 0)

(4b) Numbering array, one card per substructure boundary nodal point within the substructure. Free format(2I0).

Substructure boundary nodal point number (substructure numbering).

Corresponding global boundary nodal point number.

Cards of the numbering array may be read in any order of substructure boundary nodal point number.

(4c) Element array, one card per element within the substructure. Free format(4I0).

Element number.

Nodal point number i.

Nodal point number j. Substructure numbering.

Nodal point number k.

The nodal points i,j,k of each element must be read ANTI-CLOCKWISE.

The cards of the element array may be read in any order of element number.

(4d) Nodal point array, one card per nodal point within the substructure. Free format(I0,2E0.0).

Nodal point number(substructure numbering-.

X —— ordinate of nodal point.

Y —— ordinate of nodal point.

Cards of the nodal point array may be read in any order of nodal point number.

section (4e)

Interior force array, one card per interior nodal point with a non-zero external force. Free format(I0, 2E0.0). (ignore this section if there are no interior nodal points with external forces).

Interior nodal point number (substructure numbering).

Applied external force in the X direction.

Applied external force in the Ydirection.

The computer analysis is for a structure of unit thickness, it is suggested that the force per unit thickness be used as input. This will result in the output being in actual magnitude without the need to scale.

Section (4) is completed for each substructure. The set of substructure arrays may be read in any order of substructure number.

This completes the data requirements of program PART A.

If the program run is terminatid successfully then the following statement is outputed on line printer PART A. COMPLETE.
Run time.

The total time required to run program PART A is approximately 100n + 55 seconds.
Where n is the number of substructures.

Error Returns.

Two errors are outputed in program PART A. The first is a result of wrong input data. The second is a result of idealizing a

structure outwith the computer storage capacity specified.

Error 1/

ZERO OR NEGATIVE AREA, EL. NO.=

This error is usually a result of wrong dimensioning of one or more of the nodal points of the element indicated.

Error 2/

OVER 8 N.P. ADJACENT TO N.P. NO.

Read paragraph (g) page 170 .

section

(1) Title card, one card. (71H).

Columns 2 ——72 should contain information to be printed with the results, usually job title and name.

(2) Control card, one card. Free format(2I0).

Number of substructures.

Number of global boundary nodal points.

(3) Iteration control card, one card. Free format(3I0, 2E0.0).

Cycle print interval, (usually about 5 or 10). Outputs the iteration cycle number and the sum of the force umbalance at every cycle print interval.

Output interval of results (usually about 20 or 30); outputs the global boundary nodal point displacements, together with the iteration cycle number and sum of the force unbalance at every output interval.

Iteration cycle limit (usually about 150). If the iteration has not converged within this figure, the latest results are outputed and the program run terminated.

Tolerance limit (usually about 0.5 lb.f, or the equivelent in whatever units are being used). If the sum of the force unbalance is less than this figure,

then the iteration has converged. The latest results are outputed and the program run is terminated.

Over-relaxation factor β (usually about 1.89). To speed the rate of convergence. Must lie within the range $1 \leqslant \beta < 2$

The cycle print interval, and output interval of results are used mainly as a check on the convergence of the iteration process.

(4) Controlcard, one card. Free format(3I0).

Number of global boundary nodal points in the complete structure with non-zero external forces. (if there are no global boundary nodal point external forces read 0).

Number of global boundary nodal points in the complete structure with non-zero initial displacements (if this figure = 0, all global boundary nodal points displacements are automatically initialy set to zero in both X and Y directions.)

Number of restrained global boundary nodal points

(5) Boundary force array, one CARD per global boundary nodal point with non-zero external forces. (ignore this section if there are no global boundary nodal point external forces). Free format(I0,2E0.0).

Nodal point number (global numbering).

Applied external force inthe X direction.

Applied external force in the Y direction.

The computer analysis is for a structure of unit thickness; it is therefore suggested that the force per unit thicknessbe used as input. This will result in the output being in actual magnitude without the need to scale.

The boundary force array cards may be read in any order of nodal point number.

(6) Boundary displacement array, one card per global boundary nodal point with non-zero initial displacements. (ignore this section if initially all global boundary nodal point displacements are assumed to be zero). Free format (I0,2E0.0).

Nodal point number (global numbering).

Initial displacement X direction.

Initial displacement Y direction.

The boundary displacement array cards may be read in any order of nodal point number.

(7) Boundary condition array, one card per restrained global boundary nodal point. Free format(2I0,E0.0).

Nodal point number (global numbering).

Boundary condition code.

Read 0 if nodal point is fixed in both X and Y direction.

1 if nodal point is fixed in X direction.

2 if nodal point moves along a slope = tanϕ

Slope = Tanϕ where ϕ is measured from the X axis. For a nodal point fixed in the Y direction; this is

a case 2 nodal point with slope = 0.0

For case 0 and 1 nodal points, read any value for slope.

The boundary condition array cards may be read in any order of nodal point number. Forfurther information on boundary conditions refer to pages 69 to 71 .

This completes the card input data for program PART B, and for the complete computer analysis.

If the run terminates successfully then the following statement is outputed on the line printer: PART B. COMPLETE.

Run time

The total time required to run program PART B depends very mush on the assumed initial displacements. The approximate run time equals 0.023mz + 36 seconds.

Where m is the number of global boundary nodal points, and z is the number of iteration cycles to obtain convergence.

If a good estimate is made of the displacements, convergence will be obtained in about 130 cycles of iteration. Giving an approximate run time of 3.0m + 36 seconds.

Error returns

Two error returns are outputed in program PART B. The first is a result of idealizing a structure outwith the computer storage capacity specified. The second occures if the iteration fails to converge within the iteration cycle limit.

Error 1/

OVER 23 N.P. ADJACENT TO N.P. NO.

Read paragraph (c) page168 .

Error 2/

ITERATION FAILS TO CONVERGE IN CYCLES, SUM OF

FORCE UNBALANCE =

The iteration will fail to converge if the sum of the force unbalance is not less than the tolerance limit at the end of the iteration cycle limit. The latest updated results are outputed to line printer and the second magnetic tape.

One of the following desions has to be made.

a/ The program PART B may be run again, andeither, read in a better estimate of the displacements(using the latest results as a guide) and / or increase the iteration cycle limit. Program PART A does not have to be run again.

b/ If by comparing the latest results with the results of the previous output interval, the iteration has converged to the required number of significant figures. Then program PART C may be run and the analysis completed.

4.3 Program PART B input Data cards when using program with Direct Access Disc storage.

The card input requirements for this program are exactly the same as indicated on pages 177 to 181 , with the exception that now 140 global boundary nodal points, and up to 35 adjacent global boundary nodal points to any one global boundary nodal point is allowed.

If the run terminates successfully then the following statement is outputed on the line printer: PART B. COMPLETE.

Run time.

The approximate time required to run this program is

$2.6((m/20) + 0.5)z + 52$ seconds.

Where m is the number of global boundary nodal points, and z is the number of cycles of iteration.

Error returns are the same as indicated on pages 180 and 181 except that 35 adjacent nodal points are allowed.

No card input data is required for this program. Input data is obtained from two magnetic tapes generated from programs PART A, and B.

The output from this program completes the analysis, and consists of

X - DISPLACEMENT The nodal point displacement in the X direction.

Y - DISPLACEMENT The nodal point displacement in the Y direction.

X - STRESS The normal stress in the X direction.

Y - STRESS The normal stress in the Y direction.

XY - STRESS The shear stress.

MAX - STRESS The maximum principal stress.

MIN - STRESS The minimum principal stress.

DIRECTION The direction of the maximum principal stress from the X axis, in degrees.

Both element and interior nodal point stresses are outputed.

The results are for a structure of unit thickness. If the force per unit thickness has been used as input then the results will be of the correct magnitude.

All the results are outputed in substuucture numbering, therefore reference must be made to the substructures when analysing the results.

If the run is terminated successfully the following statement is outputed on line printer PART C. COMPLETE.

Run time.

The time required to run program PART C. is approximately 30n + 33 seconds.

Where n is the number of substructures.

The following table indicates the necessary alterations to the subscripts of the program dimension statements, when altering the following parameters by amount n.

Parameter being altered	PROGRAM PART	ROW/COLUMN SUBSCRIPT TO BE ALTERED	SUBSCRIPT ALTERED BY	Arrays to be altered
Element	A	Row	n	NUME, NPI, NPJ, NPK, AJ, AK, BJ, BK.
	C	Row	n	NUME, NPI, NPJ, NPK, AJ, AK, BJ, BK, SIGXX, SIGXY, SIGYY.
Interior nodal points	A	Row	n	NPNUM, XORD, YORD, NAP, NOTE, NCODE, PIX, PIY, NP, SXX, SXY, SYX, SYY.
			2n	SIB, G.
		Column	2n	H, SBI, SM.
		Row & column	2n	SII.
	B, B DISC	Row	n	NOTE, NCODE.
	C	Row	n	NPNUM, XORD, YORD, NOTE, NCODE, UX, UY.
			2n	SM

cont.

Parameter being altered	PROGRAM PART	Row/COLUMN SUBSCRIPT TO BE ALTERED	SUBSCRIPT ALTERED BY	Lines and / or Arrays to be altered
Substructure boundary nodal points	A	Row	n	NPNUM, XORD, YORD, NAP, NOTE, NCODE, NBP, NBPG, NP, SXX, SXY, SYX, SYY.
	B, B disc	Row	n	NOTE, NCODE.
		Column	2n	SB
	C	Row	n	NPNUM, XORD, YORD, NAP, NOTE, NCODE, UX, UY.
		Column	2n	SM.
Global boundary nodal points	B	Row	n	SXX, SXY, SYX, SYY, NP,
	B, B disc	Row	n	NFIX, SLOPE, NAP, FBX, FBY, UBX, UBY, NRBP.
	C	Row	n	NGBP, UBX, UBY.
Number of substructure nodal points adjacent to any one substructure nodal point	A	Column	n	NP, SXX, SXY, SYX, SYY, Line 0088, DO1170M=1,9±n Line 0094, NP(L,10±n)=0 Line 0161, IF(MX-10±n)1196,1702, 1702 Line 0350, WRITE(7,1113)(NP(LX, I),I=2,10±n) Line 0391, 1115 FORMAT (33H OVER 8±n N.P. ADJACENT TO N.P. NO.,I4)

cont.

Parameter being altered	PROGRAM PART	ROW/COLUMN SUBSCRIPT TO BE ALTERED	SUBSCRIPT ALTERED BY	Lines and /or Arrays being altered
Number of global boundary nodal points adjacent to any one global boundary nodal point	B	Column	n	NP, SXX, SXY, SYX, SYY, Line 0047, DO1170M=1,24±n Line 0053, NP(L,25±n)=0 Line 0097, IF(MX-25±n)1196,1702,1702 Line 0174, WRITE(6,1113)(NP(LX,I),I=2,25±n) Line 0220, 1115 FORMAT(34H OVER 23±n N.P. ADJACENT TO N.P. NO.,I4)
	B disc	Row	n	NP, SXX, SXY, SYX, SYY, Line 0054, DO1170M=1,36±n Line 0061, NP(37 n,L)=0 Line 0137, IF(MX-37±n)1196,1702,1702 Line 0288, WRITE(6,1113)(NP(1,I),I=2,37±n) Line 0344, 1115 FORMAT(34H OVER 35±n N.P. ADJACENT TO N.P. NO.,I4) All statements such as KK=(LL-20)*36±n+1 CALL PUTPART(1,KX,SXX,SXX(1,1),SXX(36±n,20)) CALL GETPART(5,KX,NP,NP(1,1),NP(37±n,1)) given by line no's 0068,0070,0072,0074,0076,0077,0079,0119,0121,0123,0125,0127,0128,0130,0143,0145,0147,0149,0151,0154,0158,0160,0165,0191,0201,0218,0235,0237,0239,0241,0243,0244,0246.

5.PROGRAM LISTING

5.1 Program PART A.

```
0001          SEND TO (ED,PROGRAM DUMP.FORTY)
0002          RUN
0003          MAP
0004          PROGRAM(PDEC16021PDE)
0005          COMPRESS INTEGER AND LOGICAL
0006          INPUT 5=CRO
0007          USE 6=MTO/FORMATTED(AWMT)
0008          OUTPUT 7=LPO
0009          END
0010            MASTER FINITE PART A
0011        C   PART A. OF FINITE ELEMENT PROGRAM
0012        C   WITH SUBSTRUCTURAL ANALYSIS
0013        C   PLANE STRESS SOLUTION
0014        C   USING TRIANGULAR ELEMENTS
0015        C   WITH IN PLANE FORCES.
0016              DIMENSION NPNUM(41),XORD(41),YORD(41),NAP(41),
0017             1NP(41,10),SXX(41,9),SXY(41,9),SYX(41,9),SYY(41,9),
0018             2NUME(65),NPI(65),NPJ(65),NPK(65),
```

```
0019          3AJ(65),AK(65),BJ(65),BK(65),
0020          4LM(3),A(6,6),B(6,6)S(6,6)
0021          5,SIB(50,2),SBI(2,50),SM(50,2),SB(2,2),
0022          6SII(50,50),G(50),H(1,50),
0023          7NOTE(41),NCODE(41),NBP(16),NBPG(16)
0024          8,PIX(41),PIY(41)
0025           WRITE(7,996)
0026           READ(5,1000)
0027           WRITE(7,1000)
0028          READ(5,999)NUMSUB,NUMGBP
0029          READ  (5,1005) E,XU
0030          WRITE(6,1119)E,XU
0031          WRITE(7,998)NUMSUB,NUMGBP,E,XU
0032           DO200JX=1,NUMSUB
0033           READ(5,1001)NSUB,NUMEL,NUMNP,NUMBP,NONZIF
0034           DO52I=1,NUMBP
0035           READ(5,53)NBP(I),NBPG(I)
0036         52 CONTINUE
```

```
0037                DO55I=1,NUMBP-1
0038                IP1=I+1
0039                DO55J=IP1,NUMBP
0040                IF(NBP(I)-NBP(J))55,55,56
0041             56 NTE=NBP(I)
0042                NBP(I)=NBP(J)
0043                NBP(J)=NTE
0044                NTE=NBPG(I)
0045                NBPG(I)=NBPG(J)
0046                NBPG(J)=NTE
0047             55 CONTINUE
0048                READ  (5,1002) (NUME(N)NPI(N),NPJ(N),NPK(N),N=1,NUMEL)
0049                READ (5,1003) (NPNUM(N),XORD(N),YORD(N),N=1,NUMNP)
0050                DO58I=1,NUMNP-1
0051                IP1=I+1
0052                DO58J=IP1,NUMNP
0053                IF(NPNUM(I)-NPNUM(J))58,58,59
0054             59 NTE=NPNUM(I)
```

```
0055                NPNUM(I)=NPNUM(J)
0056                NPNUM(J)=NTE
0057                TE=XORD(I)
0058                XORD(I)=XORD(J)
0059                XORD(J)=TE
0060                TE=YORD(I)
0061                YORD(I)=YORD(J)
0062                YORD(J)=TE
0063             58 CONTINUE
0064                WRITE(6,131)NSUB,NUMEL,NUMNP,NUMBP
0065                WRITE(7,997)NSUB,NUMEL,NUMNP,NUMBP
0066                WRITE(7,1104)
0067                WRITE(7,1105)(NBP(I),NBPG(I),I=1,NUMBP)
0068                WRITE(6,135)(NUME(N),NPI(N),NPJ(N),NPK(N),N=1,NUMEL)
0069                WRITE(7,1101)
0070                WRITE(7,1120)(NUME(N),NPI(N),NPJ(N),NPK(N),N=1,NUMEL)
0071                WRITE(6,133)(NPNUM(N),XORD(N),YORD(N),N=1,NUMNP)
0072                WRITE(7,1102)
```

```
0073              WRITE(7,1103)(NPNUM(N),XORD(N),YORD(N),N=1,NUMNP)
0074              DO70I=1,NUMNP
0075              PIX(I)=0.0
0076              PIY(I)=0.0
0077           70 CONTINUE
0078              IF(NONZIF.EQ.0)GOTO73
0079              WRITE(7,1106)
0080              DO71N=1,NONZIF
0081              READ(5,72)NN,TSX,TSY
0082              WRITE(7,1107)NN,TSX,TSY
0083              PIX(NN)=TSX
0084              PIY(NN)=TSY
0085           71 CONTINUE
0086           73 CONTINUE
0087               DO 1175 L=1,NUMNP
0088               DO 1170 M=1,9
0089               SXX(L,M)=0.0
0090               SXY (L,M)=0.0
```

```
0091                SYX(L,M)=0.0
0092                SYY(L,M)=0.0
0093          1170 NP(L,M)=0
0094               NP(L,10)=0
0095          1175 NP(L,1)=L
0096               DO 1180 N=1,NUMEL
0097               I=NPI(N)
0098               J=NPJ(N)
0099               K=NPK(N)
0100               AJ(N)=XORD(J)-XORD(I)
0101               AK(N)=XORD(K)-XORD(I)
0102               BJ(N)=YORD(J)-YORD(I)
0103               BK(N)=YORD(K)-YORD(I)
0104               AREA=(AJ(N)*BK(N)-BJ(N)*AK(N))/2.0
0105               IF (AREA) 1701,1701,0
0106         1180 CONTINUE
0107               DO 1200 N=1,NUMEL
0108               AREA=(AJ(N)*BK(N)-AK(N)*BJ(N))*0.5
```

```
0109          COMM=0.25*E/((1.0-XU**2.0)*AREA)
0110          A(1,1)=BJ(N)-BK(N)
0111          A(1,2)=0.0
0112          A(1,3)=BK(N)
0113          A(1,4)=00
0114          A(1,5)=-BJ(N)
0115          A(1,6)=0.0
0116          A(2,1)=0.0
0117          A(2,2)=AK(N)-AJ(N)
0118          A(2,3)=0.0
0119          A(2,4)=-AK(N)
0120          A(2,5)=0.0
0121          A(2,6)=AJ(N)
0122          A(3,1)=AK(N)-AJ(N)
0123          A(3,2)=BJ(N)-BK(N)
0124          A(3,3)=-AK(N)
0125          A(3,4)=BK(N)
0126          A(3,5)=AJ(N)
```

```
0127          A(3,6)=-BJ(N)
0128          B(1,1)=COMM
0129          B(1,2)=COMM*XU
0130          B(1,3)=0.0
0131          B(2,1)=COMM*XU
0132          B(2,2)=COMM
0133          B(2,3)=0.0
0134          B(3,1)=0.0
0135          B(3,2)=0.0
0136          B(3,3)=COMM*(1.0-XU)*0.5
0137          DO 1182 J=1,6
0138          DO 1182 I=1,3
0139          S(I,J)=0.0
0140          DO 1182 K=1,3
0141       1182 S(I,J)=S(I,J)+B(I,K)*A(K,J)
0142          DO 1183 J=1,6
0143          DO 1183 I=1,3
0144       1183 B(J,I)=S(I,J)
```

```
0145                DO 1184 J=1,6
0146                DO 1184 I=1,6
0147                S(I,J)=0.0
0148                DO 1184 K=1,3
0149           1184 S(I,J)=S(I,J)+B(I,K)*A(K,J)
0150                LM(1)=NPI(N)
0151                LM(2)=NPJ(N)
0152                LM(3)=NPKN)
0153                DO 1200 L=1,3
0154                LX=LM(L)
0155                DO 1200 M=1,3
0156                MX=0
0157           1185 MX=MX+1
0158                IF (NP(LX,MX)-LM(M)) 1190,1195,1190
0159           1190 IF (NP(LX,MX)) 1185,1195,1185
0160           1195 NP(LX,MX)=LM(M)
0161                IF (MX-10) 1196,1702,1702
0162           1196 SXX(LX,MX)=SXX(LX,MX)+S(2*L-1,2*M-1)
```

```
0163                SXY(LX,MX)=SXY(LX,MX)+S(2*L-1,2*M)
0164                SYX(LX,MX)=SYX(LX,MX)+S(2*L,2*M-1)
0165           1200 SYY(LX,MX)=SYY(LX,MX)+S(2*L,2*M)
0166                DO 1206 M=1,NUMNP
0167                MX =1
0168           1205 MX=MX1
0169                IF (NP(M,MX)) 1206,1206,1205
0170           1206 NAPM)=MX-1
0171                DO60I=1,NUMNP
0172                NOTE(I)=I
0173                NCODE(I)=0
0174             60 CONTINUE
0175                DO61I=1,NUMBP
0176                N=NBP(I)
0177                NN=NBPG(I)
0178                NOTE(N)=NOTE(I)
0179                NOTE(I)=N
0180                NCODE(I)=NN
```

```
0181          61 CONTINUE
0182             DO65L=1,NUMNP
0183             WRITE(6,132)NOTE(L),NCODE(L)
0184          65 CONTINUE
0185             NUMBER=2*(NUMNP-NUMBP)
0186             I=0
0187             DO100M=NUMBP+1,NUMNP
0188             I=I+2
0189             MM=NOTE(M)
0190             K=0
0191             DO100J=NUMBP1,NUMNP
0192             JJ=NOTE(J)
0193             NUM=NAP(JJ)
0194             DO103L=1,NUM
0195             IF(NP(JJ,L).EQ.MM)GOTO105
0196         103 CONTINUE
0197             K=K+2
0198             SII(K-1,I-1)=0.0
```

```
0199                SII(K,I-1)=0.0
0200                SII(K-1,I)=0.0
0201                SII(K,I)=0.0
0202              GOTO100
0203          105 K=K+2
0204              SII(K-1,I-1)=SXX(JJ,L)
0205              SII(K,I-1)=SYX(JJ,L)
0206              SII(K-1,I)=SXY(JJ,L)
0207              SII(K,I)=SYY(JJ,L)
0208          100 CONTINUE
0209              NN=NUMBER-1
0210              SII(1,1)=1.0/SII(1,1)
0211              DO150M=1,NN
0212              K=M+1
0213              DO151I=1,M
0214              G(I)=0.0
0215              DO151J=1,M
0216              G(I)=G(I)+SII(I,J)*SII(J,K)
```

```
0217          151 CONTINUE
0218              DD=0.0
0219              DO155I=1,M
0220              DD=DD+SII(K,I)*G(I)
0221          155 CONTINUE
0222              EE=SII(K,K)-DD
0223              SII(K,K)=1.0/EE
0224              DO180I=1,M
0225              SII(I,K)=-G(I)*SII(K,K)
0226          180 CONTINUE
0227              DO185J=1,M
0228              H(1,J)=0.0
0229              DO185I=1,M
0230              H(1,J)=H(1,J)+SII(K,I)*SII(I,J)
0231          185 CONTINUE
0232              DO190J=1,M
0233              SII(K,J)=-SII(K,K)*H(1,J)
0234          190 CONTINUE
```

```
0235                DO150I=1,M
0236                DO150J=1,M
0237                SII(I,J)=SII(I,J)-G(I)*SII(K,J)
0238              150 CONTINUE
0239                DO500I=2,NUMBER,2
0240                N=0
0241                UIX=0.0
0242                UIY=0.0
0243                IF(NONZIF.EQ.0)GOTO503
0244                DO501J=NUMBP+1,NUMNP
0245                JJ=NOTE(J)
0246                N=N+2
0247                UIX=UIX+SII(I-1,N-1)*PIX(JJ)+SII(I-1,N)*PIY(JJ)
0248                UIY=UIY+SII(I,N-1)*PIX(JJ)+SII(I,N)*PIY(JJ)
0249              501 CONTINUE
0250              503 WRITE(6,502)UIX,UIY
0251              500 CONTINUE
0252                DO200J=1,NUMBP
```

```
0253                JJ=NOTE(J)
0254                K=0
0255                DO201I=NUMBP1,NUMNP
0256                II=NOTE(I)
0257                NUM=NAP(II)
0258                DO202L=1,NUM
0259                IF(NP(II,L).EQ.JJ)GOTO203
0260            202 CONTINUE
0261                K=K+2
0262                SIB(K-1,1)=0.0
0263                SIB(K,1)=0.0
0264                SIB(K-1,2)=0.0
0265                SIB(K,2)=0.0
0266                GOTO201
0267            203 K=K+2
0268                SIB(K-1,1)=SXX(II,L)
0269                SIB(K,1)=SYX(II,L)
0270                SIB(K-1,2)=SXY(II,L)
```

```
0271                SIB(K,2)=SYY(II,L)
0272            201 CONTINUE
0273                DO210N=2,NUMBER,2
0274                SM(N-1,1)=0.0
0275                SM(N,1)=0.0
0276                SM(N-1,2)=0.0
0277                SM(N2)=0.0
0278                DO210I=1,NUMBER
0279                SM(N-1,1)=SM(N-1,1)+SII(N-1,I)*SIB(I,1)
0280                SM(N-1,2)=SM(N-1,2)+SII(N-1,I)*SIB(I,2)
0281                SM(N,1)=SM(N,1)+SII(N,I)*SIB(I,1)
0282                SM(N,2)=SM(N,2)+SII(N,I)*SIB(I,2)
0283            210 CONTINUE
0284                DO220N=1,NUMBER
0285                WRITE(6,225)SM(N,1),SM(N,2)
0286            220 CONTINUE
0287                RBX=0.0
0288                RBY=0.0
```

```
0289                LL=0
0290                IF(NONZIF.EQ.0)GOTO551
0291                DO550I=NUMBP+1,NUMNP
0292                II=NOTE(I)
0293                LL=LL+2
0294                RBX=RBX+SM(LL-1,1)*PIX(II)+SM(LL,1)*PIY(II)
0295                RBY=RBY+SM(LL-1,2)*PIX(II)+SM(LL,2)*PIY(II)
0296            550 CONTINUE
0297            551 WRITE(6,560)RBX,RBY
0298                DO200M=1,NUMBP
0299                MM=NOTE(M)
0300                NUM=NAP(MM)
0301                K=0
0302                DO117I=NUMBP+1,NUMNP
0303                II=NOTE(I)
0304                DO118L=1,NUM
0305                IF(NP(MM,L).EQ.II)GOTO119
0306            118 CONTINUE
```

```
0307                K=K+2
0308                SBI(1,K-1)=0.0
0309                SBI(2,K-1)=0.0
0310                SBI(1,K)=0.0
0311                SBI(2,K)=0.0
0312                GOTO117
0313            119 K=K+2
0314                SBI(1,K-1)=SXX(MM,L)
0315                SBI(2,K-1)=SYX(MM,L)
0316                SBI(1,K)=SXY(MM,L)
0317                SBI(2,K)=SYY(MM,L)
0318            117 CONTINUE
0319                SB(1,1)=0.0
0320                SB(2,1)=0.0
0321                SB(1,2)=0.0
0322                SB(2,2)=0.0
0323                DO120I=1,NUMBER
0324                SB(1,1)=SB(1,1)+SBI(1,I)*SM(I,1)
```

```
0325            SB(2,1)=SB(2,1)+SBI(2,I)*SM(I,1)
0326            SB(1,2)=SB(1,2)+SBI(1,I)*SM(I,2)
0327            SB(2,2)=SB(2,2)+SBI(2,I)*SM(I,2)
0328          120 CONTINUE
0329              DO121L=1,NUM
0330              IF(NP(MM,L).EQ.JJ)GOTO122
0331          121 CONTINUE
0332              SB1,1)=-SB(1,1)
0333              SB(2,1)=-SB(2,1)
0334              SB(1,2)=-SB(1,2)
0335              SB(2,2)=-SB(2,2)
0336              GOTO123
0337          122 CONTINUE
0338              SB(1,1)=SXX(MM,L)-SB(1,1)
0339              SB(2,1)=SYX(MM,L)-SB(2,1)
0340              SB(1,2)=SXY(MM,L)-SB(1,2)
0341              SB(2,2)=SYY(MM,L)-SB(2,2)
0342          123 CONTINUE
```

```
0343              WRITE(6,130)SB(1,1),SB(2,1),SB(1,2),SB(2,2)
0344            200 CONTINUE
0345              GOTO1999
0346           1701 WRITE(7,1114)N
0347              GOTO1998
0348           1702 WRITE(7,1115)LX
0349              WRITE(7,1112)
0350              WRITE(7,1113)(NP(LX,I),I=2,10)
0351           1998 FAIL=SQRT(-1)
0352           1999 CONTINUE
0353                WRITE(7,1121)
0354                ENDFILE 6
0355             53 FORMAT(2I0)
0356             72 FORMAT(I0,2E0.0)
0357            130 FORMAT(4E18.9)
0358            131 FORMAT(I2,3I3)
0359            132 FORMAT(I2,2X,I2)
0360            133 FORMAT(I3,2X,E15.6,2X,E15.6)
```

```
0361          135 FORMAT(4I5)
0362          136 FORMAT(4I2)
0363          225 FORMAT(E15.5,5X,E15.5)
0364          502 FORMAT(2E15.6)
0365          560 FORMAT(2E15.6)
0366          996 FORMAT(21H1PART A.  INPUT DATA )
0367          997 FORMAT(21H1SUBSTRUCTURE NUMBER=,I2//20H NUMBER OF ELEMENTS=,I4//
0368             124H NUMBER OF NODAL POINTS=,I4//40H NUMBER OF SUBSTRUCTURE BOUNDAR
0369             2Y POINTS=,I3//)
0370          998 FORMAT(25HONUMBER OF SUBSTRUCTURES=,I3//34H NUMBER OF GLOBAL BOUND
0371             1ARY POINTS=,I3//16H YOUNGS MODULUS=,E12.5//16H POISSONS RATIO=,
0372             2E12.5)
0373          999 FORMAT(2IO)
0374         1000 FORMAT (72H  BCD  INFORMATION
0375             1                    )
0376         1001 FORMAT(5IO)
0377         1002 FORMAT (4IO)
0378         1003 FORMAT (IO,2EO.O)
```

```
0379          1005 FORMAT (2E0.0)
0380          1101 FORMAT(40HO        EL          I          J          K)
0381          1102 FORMAT(32HO      NP        X-ORD        Y-ORD)
0382          1103 FORMAT(I8,2F12.3)
0383          1104 FORMAT(42HOSUBSTRUCTURE B.P.    CORRESPONDING G.B.P.)
0384          1105 FORMAT(I17,20X,I3)
0385          1106 FORMAT(25HONON-ZERO INTERIOR FORCES,//38H INTERIOR N.P.       XLOA
0386             1D       YLOAD)
0387          1107 FORMAT(I14,2F12.3)
0388          1112 FORMAT(20HOADJACENT NODES ARE.)
0389          1113 FORMAT(I20)
0390          1114 FORMAT(31H ZERO OR NEGATIVE AREA, EL.NO.=,I4)
0391          1115 FORMAT (33H OVER 8 N.P. ADJACENT TO N.P. NO.,I4)
0392          1116 FORMAT (2I4)
0393          1119 FORMAT(E12.5,2X,E12.5)
0394          1120 FORMAT (4I10)
0395          1121 FORMAT(18HOPART A,  COMPLETE)
0396               STOP
```

```
0397                    END
0398                    FINISH
```

5.2 Program PART B.

```
0001                SEND TO (ED,PROGRAM DUMP.FORTY)
0002                RUN
0003                MAP
0004                PROGRAM(PDEC16021PDE)
0005                COMPRESS INTEGER AND LOGICAL
0006                INPUT 4=CRO
0007                USE 5=MTO/FORMATTED(AWMT)
0008                OUTPUT 6=LPO
0009                USE 7=MT1/FORMATTED(ACW)
0010                  END
0011                  MASTER FINITE PART B
0012              C   PART B. OF FINITE ELEMENT PROGRAM
0013              C   WITH SUBSTRUCTURAL ANALYSIS
0014              C   PLANE STRESS SOLUTION
0015              C   USING TRIANGULAR ELEMENTS
0016              C   WITH IN PLANE FORCES.
0017                    DIMENSION NOTE(41),NCODE(41),SB(2,32),
0018                   1NRBP(54),NFIX(54),SLOPE(54),
```

```
0019          2SXX(54,24),SXY(54,24),SYX(54,24),SYY(54,24),NP(54,25),
0020          3NAP(54),FBX(54),FBY(54),UBX(54),UBY(54)
0021           READ(4,1000)
0022           READ(4,10)NUMSUB,NUMGBP
0023           READ(4,40)NCPIN,NOPIN,NCYCM,TOLER,XFAC
0024           READ(4,60)NONZBF,NONZBD,NUMRBP
0025           DO50I=1,NUMGBP
0026           FBX(I)=0.0
0027           FBY(I)=0.0
0028             UBX(I)=0.0
0029             UBY(I)=0.0
0030          50 CONTINUE
0031             IF(NONZBF.EQ.0)GOTO53
0032             DO51I=1,NONZBF
0033             READ(4,61)NN,PBX,PBY
0034              FBX(NN)=PBX
0035              FBY(NN)=PBY
0036           51 CONTINUE
```

```
0037          53 CONTINUE
0038             IF(NONZBD.EQ.0)GOTO54
0039             DO52I=1,NONZBD
0040             READ(4,61)NN,TSX,TSY
0041             UBX(NN)=TSX
0042             UBY(NN)=TSY
0043          52 CONTINUE
0044          54 CONTINUE
0045             READ(4,62)(NRBP(N),NFIX(N),SLOPE(N),N=1,NUMRBP)
0046             DO1175L=1,NUMGBP
0047             DO1170M=1,24
0048             SXX(L,M)=0.0
0049             SXY (L,M)=0.0
0050             SYX(L,M)=0.0
0051             SYY(L,M)=0.0
0052          1170 NP(L,M)=0
0053               NP(L,25)=0
0054          1175 NP(L,1)=L
```

```
0055          WRITE(7,1000)
0056          WRITE(7,1001)NUMSUB,NUMGBP
0057          WRITE(6,999)
0058          WRITE(6,1000)
0059          WRITE(6,1100)NUMSUB,NUMGBP,NUMRBP,NCPIN,NOPIN,NCYCM,TOLER,XFAC
0060          WRITE(6,1101)
0061          WRITE(6,1102)(N,FBX(N),FBY(N),UBX(N),UBY(N),N=1,NUMGBP)
0062          WRITE(6,1103)
0063          WRITE(6,1104)(NRBP(N),NFIX(N),SLOPE(N),N=1,NUMRBP)
0064          WRITE(6,1109)
0065          READ(5,43)E,XU
0066          DO1200JX=1,NUMSUB
0067          READ(5,11)NSUB,NUMEL,NUMNP,NUMBP
0068          READ(5,41)(N1,N2,N3,N4,N=1,NUMEL)
0069          READ(5,42)(N5,X1,X2,N=1,NUMNP)
0070          DO15I=1,NUMNP
0071          NOTE(I)=0
0072          NCODE(I)=0
```

```
0073          READ(5,16)NOTE(I),NCODE(I)
0074       15 CONTINUE
0075          DO20I=1,2*NUMBP
0076          SB(1,I)=0.0
0077          SB(2,I)=0.0
0078       20 CONTINUE
0079          NUMBER=2*(NUMNP-NUMBP)
0080          NX=NUMBER/2
0081          READ(5,17)(X3,X4,I=1,NX)
0082          DO1200L=1,NUMBP
0083          LX=NCODE(L)
0084          READ(5,26)(X5,X6,I=1,NUMBER)
0085          READ(5,27)RBX,RBY
0086          FBX(LX)=FBX(LX)-RBX
0087          FBY(LX)=FBY(LX)-RBY
0088          DO21LL=2,2*NUMBP,2
0089          READ(5,22)SB(1,LL-1),SB(1,LL),SB(2,LL-1),SB(2,LL)
0090       21 CONTINUE
```

```
0091                DO1200M=1,NUMBP
0092                MX=0
0093          1185  MX=MX+1
0094                IF(NP(LX,MX)-NCODE(M))1190,1195,1190
0095          1190  IF(NP(LX,MX))1185,1195,1185
0096          1195  NP(LX,MX)=NCODE(M)
0097                IF(MX-25)1196,1702,1702
0098          1196  SXX(LX,MX)=SXX(LX,MX)+SB(1,2*M-1)
0099                SXY(LX,MX)=SXY(LX,MX)+SB(1,2*M)
0100                SYX(LX,MX)=SYX(LX,MX)+SB(2,2*M-1)
0101                SYY(LX,MX)=SYY(LX,MX)+SB(2,2*M)
0102          1200  CONTINUE
0103                DO1206M=1,NUMGBP
0104                MX=1
0105          1205  MX=MX+1
0106                IF(NP(M,MX))1206,1206,1205
0107          1206  NAP(M)=MX-1
0108                DO1210M=1,NUMGBP
```

```
0109                COMM=SXX(M,1)*SYY(M,1)-SXY(M,1)*SYX(M,1)
0110                TEMP=SYY(M,1)COMM
0111                SYY(M,1)=SXX(M,1)/COMM
0112                SXX(M,1)=TEMP
0113                SXY(M,1)=-SXY(M,1)/COMM
0114          1210  SYX(M,1)=-SYX(M,1)/COMM
0115                DO1240L=1,NUMRBP
0116                M=NRBR(L)
0117                NP(M,1)=0
0118                IF (NFIX(L)-1) 1225,1220,1215
0119          1215  C=(SXX(M,1)*SLOPE(L)-SXY(M,1))/(SYX(M,1)*SLOPE(L)-SYY(M,1))
0120                R=1.-C*SLOPE(L)
0121                SXX(M,1)=(SXX(M,1)-C*SYX(M,1))/R
0122                SXY(M,1)=(SXY(M,1)-C*SYY(M,1))/R
0123                SYX(M,1)=SXX(M,1)*SLOPE(L)
0124                SYY(M,1)=SXY(M,1)*SLOPE(L)
0125                GO TO 1240
0126          1220  SYY(M,1)=SYY(M,1)-SYX(M,1)*SXY(M,1)/SXX(M,1)
```

```
0127                GO TO 1230
0128          1225 SYY(M,1)=0.0
0129          1230 SXX(M,1)=0.0
0130          1235 SXY(M,1)=0.0
0131               SYX(M,1)=0.0
0132          1240 CONTINUE
0133               NCYCLE=0
0134               NUMPT=NCPIN
0135               NUMOPT=NOPIN
0136          1244 SUM=0.0
0137               DO1290M=1,NUMGBP
0138               NUM=NAP(M)
0139               IF (SXX(M,1)+SYY(M,1)) 1275,1290,1275
0140          1275 FRX=FBX(M)
0141               FRY=FBY(M)
0142               DO 1280 L=2,NUM
0143               N=NP(M,L)
0144               FRX=FRX-SXX(M,L)*UBX(N)-SXY(M,L)*UBY(N)
```

```
0145          1280 FRY=FRY-SYX(M,L)*UBX(N)-SYY(M,L)*UBY(N)
0146               DX=SXX(M,1)*FRX+SXY(M,1)*FRY-UBX(M)
0147               DY=SYX(M,1)*FRX+SYY(M,1)*FRY-UBY(M)
0148               UBX(M)=UBX(M)+XFAC*DX
0149               UBY(M)=UBY(M)+XFAC*DY
0150               IF (NP(M,1)) 1285,1290,1285
0151          1285 SUM=SUM+ABS(DX/SXX(M,1))+ABS(DY/SYY(M,1))
0152          1290 CONTINUE
0153               NCYCLE=NCYCLE +1
0154               IF (NCYCLE-NUMPT) 1305,1300,1300
0155          1300 NUMPT=NUMPT+NCPIN
0156               WRITE(6,1105)
0157               WRITE (6,1106) NCYCLE,SUM
0158          1305 IF(SUM-TOLER)1401,1401,1310
0159          1310 IF (NCYCM-NCYCLE) 1400,1400,1315
0160          1315 IF (NCYCLE-NUMOPT) 1244,1320,1320
0161          1401 WRITE(6,1402)NCYCLE,SUM
0162               GOTO1400
```

```
0163          1320 NUMOPT=NUMOPT+NOPIN
0164         1400 CONTINUE
0165              WRITE(6,1107)
0166              WRITE(6,1108)(N,UBX(N),UBY(N),N=1,NUMGBP)
0167              IF (SUM-TOLER) 1998,1998,1430
0168        1430 IF(NCYCM-NCYCLE)1999,1999,1244
0169        1999 WRITE(6,1403)NCYCM,SUM
0170              WRITE(7,1108)(N,UBX(N),UBY(N),N=1,NUMGBP)
0171              GOTO1703
0172        1702 WRITE(6,1115)LX
0173              WRITE(6,1112)
0174              WRITE(6,1113)(NP(LX,I),I=2,25)
0175        1703 ENDFILE 5
0176              ENDFILE 7
0177              FAIL=SQRT(-1)
0178         1998 CONTINUE
0179              WRITE(7,1108)(N,UBX(N),UBY(N),N=1,NUMGBP)
0180              ENDFILE 5
```

```
0181            ENDFILE 7
0182            WRITE(6,1110)
0183         10 FORMAT(2I0)
0184         11 FORMAT(I2,3I3)
0185         16 FORMAT(I2,2X,I2)
0186         17 FORMAT(2E15.6)
0187         22 FORMAT(4E18.9)
0188         26 FORMAT(E15.5,5X,E15.5)
0189         27 FORMAT(2E15.6)
0190         40 FORMAT(3I0,2E0.0)
0191         41 FORMAT(4I5)
0192         42 FORMAT(I3,2X,E15.6,2X,E15.6)
0193         43 FORMAT(E12.5,2X,E12.5)
0194         60 FORMAT(3I0
0195         61 FORMAT(I0,2E0.0
0196         62 FORMAT(2I0,E0.0)
0197        999 FORMAT(20H1PART B.  INPUT DATA)
0198       1000 FORMAT ( 2H  BCD  INFORMATION
```

```
0199                1                                    )
0200          1001 FORMAT(I2,1X,I3)
0201          1100 FORMAT(25HONUMBER OF SUBSTRUCTURES=,I2//34H NUMBER OF GLOBAL BOUND
0202             1ARY POINTS=,I3//38H NUMBER OF RESTRAINED BOUNDARY POINTS=,I3//
0203             222H CYCLE PRINT INTERVAL=,I3//28H OUTPUT INTERVAL OF RESULTS=,I3//
0204             313H CYCLE LIMIT=,I4//17H TOLERANCE LIMIT=,F6.3//
0205             424H OVER-RELAXATION FACTOR=,F6.3//)
0206          1101 FORMAT(60HO    GNP        X-LOAD        YLOAD        X-DISP        Y-
0207             1DISP)
0208          1102 FORMAT(I8,2F13.3,2F13.8)
0209          1103 FORMAT(31HORESTRAINED BOUNDARY CONDITIONS,//27H    GNP      NFIX
0210             2   SLOPE)
0211          1104 FORMAT(I7,I9,3X,F8.4)
0212          1105 FORMAT(40H1         CYCLE      SUM OF FORCE UNBALANCE)
0213          1106 FORMAT(I12,5X,E20.6)
0214          1107 FORMAT(51HOGLOBAL BOUNDARY N.P. X-DISPLACEMENT Y-DISPLACEMENT)
0215          1108 FORMAT(I21,2E15.6)
0216          1109 FORMAT(18HOEND OF INPUT DATA)
```

```
0217          1110 FORMAT(18HOPART B.  COMPLETE)
0218          1112 FORMAT(20HOADJACENT NODES ARE.)
0219          1113 FORMAT(I20)
0220          1115 FORMAT(34H OVER 23 N.P. ADJACENT TO N.P. NO.,I4)
0221          1402 FORMAT(1H1,22HITERATION CONVERGES IN,I3,6HCYCLES,5X,
0222              123HSUM OF FORCE UNBALANCE=,E20.6,//)
0223          1403 FORMAT(32HOITERATION FAILS TO CONVERGE IN ,I4,7H CYCLES,
0224              129H      SUM OF FORCE UNBALANCE=,E20.6)
0225               STOP
0226               END
0227               FINISH
```

5.3 Program PART B with Disc storage.

```
0001                SEND TO (ED,PROGRAM DUMP.FORTY)
0002                RUN
0003                MAP
0004                PROGRAM(PDEC16021PDE)
0005                COMPRESS INTEGER AND LOGICAL
0006                INPUT 4=CRO
0007                USE 5=MTO/FORMATTED(AWMT)
0008                OUTPUT 6=LPO
0009                USE 7=MT1/FORMATTED(ACW)
0010                 END
0011                 MASTER FINITE PART B
0012          C     PART B. OF FINITE ELEMENT PROGRAM
0013          C     WITH SUBSTRUCTURAL ANALYSIS
0014          C     USING DIRECT ACCESS DISC STORAGE.
0015          C     PLANE STRESS SOLUTION
0016          C     USING TRIANGULAR ELEMENTS
0017          C     WITH IN PLANE FORCES.
0018                   DIMENSION NOTE(41),NCODE(41),SB(2,32),
```

```
0019          1NRBP(140),NFIX(140),SLOPE(140),
0020          2SXX(36,20),SXY(36,20),SYX(36,20),SYY(36,20),NP(37,20),
0021          3NAP(140),FBX(140),FBY(140),UBX(140),UBY(140)
0022           CALL WORKFILE(1,2HED,10000)
0023           CALL WORKFILE(2,2HED,10000)
0024           CALL WORKFILE(3,2HED,10000)
0025           CALL WORKFILE(4,2HED,10000)
0026           CALL WORKFILE(5,2HED,10000)
0027           READ(4,1000)
0028          READ(4,10)NUMSUB,NUMGBP
0029          READ(4,40)NCPIN,NOPIN,NCYCM,TOLER,XFAC
0030           READ(4,60)NONZBF,NONZBD,NUMRBP
0031           DO50I=1,NUMGBP
0032            FBX(I)=0.0
0033            FBY(I)=0.0
0034            UBX(I)=0.0
0035            UBY(I)=0.0
0036          50 CONTINUE
```

```
0037                IF(NONZBF.EQ.0)GOTO53
0038                DO51 I=1,NONZBF
0039                READ(4,61)NN,PBX,PBY
0040                FBX(NN)=PBX
0041                FBY(NN)=PBY
0042             51 CONTINUE
0043             53 CONTINUE
0044                IF(NONZBD.EQ.0)GOTO54
0045                DO52I=1,NONZBD
0046                 READ(4,61)NN,TSX,TSY
0047                 UBX(NN)=TSX
0048                 UBY(NN)=TSY
0049             52 CONTINUE
0050             54 CONTINUE
0051                 READ(4,62)(NRBP(N),NFIX(N,SLOPE(N),N=1,NUMRBP)
0052                 LL=0
0053             1169 DO1175L=1,20
0054                  DO1170M=1,36
```

```
0055                SXX(M,L)=0.0
0056                SXY(M,L)=0.0
0057                SYX(M,L)=0.0
0058                SYY(M,L)=0.0
0059                NP(M,L)=0
0060          1170 CONTINUE
0061                NP(37,L)=0
0062          1175 CONTINUE
0063                DO1174I=1,20
0064                L=LL+I
0065                NP(1,I)=L
0066          1174 CONTINUE
0067                LL=LL+20
0068                KK=(LL-20)*36+1
0069                KX=KK
0070                CALL PUTPART(1,KX,SXX,SXX(1,1),SXX(36,20))
0071                KX=KK
0072                CALL PUTPART(2,KX,SXY,SXY(1,1),SXY(36,20))
```

```
0073                KX=KK
0074                CALL PUTPART(3,KX,SYX,SYX(1,1),SYX(36,20))
0075                KX=KK
0076                CALL PUTPART(4,KX,SYY,SYY(1,1),SYY(36,20)
0077               KK=(LL-20)*37+1
0078               KX=KK
0079               CALL PUTPART(5,KX,NP,NP(1,1),NP(37,20))
0080               IF(LL.GE.NUMGBP)GOTO9
0081               GOTO1169
0082             9 CONTINUE
0083              WRITE(7,1000)
0084              WRITE(7,1001)NUMSUB,NUMGBP
0085              WRITE(6,999)
0086              WRITE(6,1000)
0087              WRITE(6,1100)NUMSUB,NUMGBP,NUMRBP,NCPIN,NOPIN,NCYCM,TOLER,XFAC
0088              WRITE(6,1101)
0089              WRITE(6,1102)(N,FBX(N),FBY(N),UBX(N),UBY(N),N=1,NUMGBP)
0090              WRITE(6,1103)
```

```
0091          WRITE(6,1104)(NRBP(N),NFIX(N),SLOPE(N),N=1,NUMRBP)
0092          WRITE(6,1109)
0093          READ(5,43)E,XU
0094          DO1200JX=1,NUMSUB
0095         READ(5,11)NSUB,NUMEL,NUMNP,NUMBP
0096          READ(5,41)(N1,N2,N3,N4,N=1,NUMEL)
0097          READ(5,42)(N5,X1,X2,N=1,NUMNP)
0098          DO15I=1,NUMNP
0099          NOTE(I)=0
0100          NCODE(I)=0
0101          READ(5,16)NOTE(I),NCODE(I)
0102        15 CONTINUE
0103          DO20I=1,2*NUMBP
0104          DO20J=1,2*NUMBP
0105          SB(I,J)=0.0
0106        20 CONTINUE
0107          NUMBER=2*(NUMNP-NUMBP)
0108          NX=NUMBER/2
```

ERRATA

Page 233 does not exist. Read program statements direct from page 232 to 234.

```
0109          READ(5,17)(X3,X4,I=1,NX)
0110          DO1200L=1,NUMBP
0111          LX=NCODE(L)
0112          READ(5,26)(X5,X6,I=1,NUMBER)
0113          READ(5,27)RBX,RBY
0114          FBX(LX)=FBX(LX)-RBX
0115          FBY(LX)=FBY(LX)-RBY
0116          DO21LL=2,2*NUMBP,2
0117          READ(5,22)SB(1,LL-1),SB(1,LL),SB(2,LL-1),SB(2,LL)
0118        21 CONTINUE
0119             KK=(LX-1)*36+1
0120             KX=KK
0121             CALL GETPART(1,KX,SXX,SXX(1,1),SXX(36,1))
0122             KX=KK
0123             CALL GETPART(2,KX,SXY,SXY(1,1),SXY(36,1))
0124             KX=KK
0125             CALL GETPART(3,KX,SYX,SYX(1,1),SYX(36,1))
0126             KX=KK
```

```
0127                CALL GETPART(4,KX,SYY,SYY(1,1),SYY(36,1))
0128                KK=(LX-1)*37+1
0129                KX=KK
0130                CALL GETPART(5,KX,NP,NP(1,1),NP(37,1))
0131                DO1202M=1,NUMBP
0132                MX=0
0133           1185 MX=MX+1
0134                IF(NP(MX,1)-NCODE(M))1190,1195,1190
0135           1190 IF(NP(MX,1))1185,1195,1185
0136           1195 NP(MX,1)=NCODE(M)
0137                IF(MX-37)1196,1702,1702
0138           1196 SXX(MX,1)=SXX(MX,1)+SB(1,2*M-1)
0139                SXY(MX,1)=SXY(MX,1)+SB(1,2*M)
0140                SYX(MX,1)=SYX(MX,1)+SB(2,2*M-1)
0141                SYY(MX,1)=SYY(MX,1)+SB(2,2*M)
0142           1202 CONTINUE
0143                KK=(LX-1)*36+1
0144                KX=KK
```

```
0145          CALL PUTPART(1,KX,SXX,SXX(1,1),SXX(36,1))
0146          KX=KK
0147          CALL PUTPART(2,KX,SXY,SXY(1,1),SXY(36,1))
0148          KX=KK
0149          CALL PUTPART(3,KX,SYX,SYX(1,1),SYX(36,1))
0150          KX=KK
0151          CALL PUTPART(4,KX,SYY,SYY(1,1),SYY(36,1))
0152          KK=(LX-1)*37+1
0153          KX=KK
0154              CALL PUTPART(5,KX,NP,NP(1,1),NP(37,1))
0155        1200 CONTINUE
0156             DO1206M=1,NUMGBP
0157             MX=1
0158             KK=(M-1)*37+1
0159             KX=KK
0160              CALL GETPART(5,KX,NP,NP(1,1),NP(37,1))
0161           1205 MX=MX+1
0162                IF(NP(MX,1))1206,1206,1205
```

```
0163          1206 NAP(M)=MX-1
0164               DO1210M=1,NUMGBP
0165               KK=(M-1)*36+1
0166               KX=KK
0167               CALL GETPART(1,KX,SXX,SXX(1,1),SXX(1,1))
0168               KX=KK
0169               CALL GETPART(2,KX,SXY,SXY(1,1),SXY(1,1))
0170               KX=KK
0171               CALL GETPART(3,KX,SYX,SYX(1,1),SYX(1,1))
0172               KX=KK
0173               CALL GETPART(4,KX,SYY,SYY(1,1),SYY(1,1))
0174               COMM=SXX(1,1)*SYY(1,1)-SXY(1,1)*SYX(1,1))
0175               TEMP=SYY(1,1)/COMM
0176               SYY(1,1)=SXX(1,1)/COMM
0177               SXX(1,1)=TEMP
0178               SXY(1,1)=-SXY(1,1)/COMM
0179               SYX(1,1)=-SYX(1,1)/COMM
0180               KX=KK
```

```
0181                CALL PUTPART(1,KX,SXX,SXX(1,1),SXX(1,1))
0182                KX=KK
0183                CALL PUTPART(2,KX,SXY,SXY(1,1),SXY(1,1))
0184                KX=KK
0185                CALL PUTPART(3,KX,SYX,SYX(1,1),SYX(1,1))
0186                KX=KK .
0187                CALL PUTPART(4,KX,SYY,SYY(1,1),SYY(1,1))
0188          1210  CONTINUE
0189                DO1240L=1,NUMRBP
0190                M=NRBP(L)
0191                KK=(M-1)*36+1
0192                KX=KK
0193                CALL GETPART(1,KX,SXX,SXX(1,1),SXX(1,1))
0194                KX=KK
0195                CALL GETPART(2,KX,SXY,SXY(1,1),SXY(1,1))
0196                KX=KK
0197                CALL GETPART(3,KX,SYX,SYX(1,1),SYX(1,1))
0198                KX=KK
```

```
0199              CALL GETPART(4,KX,SYY,SYY(1,1),SYY(1,1))
0200              NP(1,1)=0
0201              KK=(M-1)*37+1
0202              KX=KK
0203              CALL PUTPART(5,KX,NP,NP(1,1),NP(1,1))
0204              IF (NFIX(L)-1) 1225,1220,1215
0205         1215 C=(SXX(1,1)*SLOPE(L)-SXY(1,1))/(SYX(1,1)*SLOPE(L)-SYY(1,1))
0206              R=1-C*SLOPE(L)
0207              SXX(1,1)=(SXX(1,1)-C*SYX(1,1))/R
0208              SXY(1,1)=(SXY(1,1)-C*SYY(1,1))/R
0209              SYX(1,1)=SXX(1,1)*SLOPE(L)
0210              SYY(1,1)=SXY(1,1)*SLOPE(L)
0211              GOTO1242
0212         1220 SYY(1,1)=SYY(1,1)-SYX(1,1)*SXY(1,1)/SXX(1,1)
0213              GO TO 1230
0214         1225 SYY(1,1)=0.0
0215         1230 SXX(1,1)=0.0
0216         1235 SXY(1,1)=0.0
```

```
0217                SYX(1,1)=0.0
0218          1242 KK=(M-1)*36+1
0219               KX=KK
0220               CALL PUTPART(1,KX,SXX,SXX(1,1),SXX(1,1))
0221               KX=KK
0222               CALL PUTPART(2,KX,SXY,SXY(1,1),SXY(1,1))
0223               KX=KK
0224               CALL PUTPART(3,KX,SYX,SYX(1,1),SYX(1,1))
0225               KX=KK
0226               CALL PUTPART(4,KX,SYY,SYY(1,1),SYY(1,1))
0227          1240 CONTINUE
0228               NCYCLE=0
0229               NUMPT=NCPIN
0230               NUMOPT=NOPIN
0231          1244 SUM=0.0
0232               JM=0
0233               MM=0
0234          1291 MM=MM+20
```

```
0235                KK=(MM-20)*36+1
0236                KX=KK
0237                CALL GETPART(1,KX,SXX,SXX(1,1),SXX(36,20))
0238                KX=KK
0239                CALL GETPART(2,KX,SXY,SXY(1,1),SXY(36,20))
0240                KX=KK
0241                CALL GETPART(3,KX,SYX,SYX(1,1),SYX(36,20))
0242                KX=KK
0243                CALL GETPART(4,KX,SYY,SYY(1,1),SYY(36,20))
0244                KK=(MM-20)*37+1
0245                KX=KK
0246                CALL GETPART(5,KX,NP,NP(1,1),NP(37,20))
0247                DO1295M=1,20
0248                JM=JM+1
0249                NUM=NAP(JM)
0250                IF(SXX(1,M)+SYY(1,M))1275,1289,1275
0251           1275 FRX=FBX(JM)
0252                FRY=FBY(JM)
```

```
0253                DO1280L=2,NUM
0254                N=NP(L,M)
0255                FRX=FRX-SXX(L,M)*UBX(N)-SXY(L,M)*UBY(N)
0256          1280  FRY=FRY-SYX(L,M)*UBX(N)-SYY(L,M)*UBY(N)
0257                DX=SXX(1,M)*FRX+SXY(1,M)*FRY-UBX(JM)
0258                DY=SYX(1,M)*FRX+SYY(1,M)*FRY-UBY(JM)
0259                UBX(JM)=UBX(JM)+XFAC*DX
0260                UBY(JM)=UBY(JM)+XFAC*DY
0261                IF(NP(1,M))1285,1289,1285
0262          1285  SUM=SUM+ABS(DX/SXX(1,M))+ABS(DY/SYY(1,M))
0263          1289  IF(JM.EQ.NUMGBP)GOTO1290
0264          1295  CONTINUE
0265                GOTO1291
0266          1290  CONTINUE
0267                NCYCLE=NCYCLE+1
0268                IF (NCYCLE-NUMPT) 1305,1300,1300
0269          1300  NUMPT=NUMPT+NCPIN
0270                WRITE(6,1105)
```

```
0271                WRITE (6,1106) NCYCLE,SUM
0272          1305 IF(SUM-TOLER)1401,1401,1310
0273          1310 IF (NCYCM-NCYCLE) 1400,1400,1315
0274          1315 IF (NCYCLE-NUMOPT) 1244,1320,1320
0275          1401 WRITE(6,1402)NCYCLE,SUM
0276               GOTO1400
0277          1320 NUMOPT=NUMOPT+NOPIN
0278          1400 CONTINUE
0279               WRITE(6,1107)
0280                WRITE(6,1108)(N,UBX(N),UBY(N),N=1,NUMGBP)
0281                IF (SUM-TOLER) 1998,1998,1430
0282          1430 IF(NCYCM-NCYCLE)1999,1999,1244
0283          1999 WRITE(6,1403)NCYCM,SUM
0284               WRITE(7,1108)(N,UBX(N),UBY(N),N=1,NUMGBP)
0285               GOTO1703
0286          1702 WRITE(6,1115)LX
0287               WRITE(6,1112)
0288               WRITE(6,1113)(NP(1,I),I=2,37)
```

```
0289          1703 CALL FREEFILE(1)
0290               CALL FREEFILE(2)
0291               CALL FREEFILE(3)
0292               CALL FREEFILE(4)
0293               CALL FREEFILE(5)
0294               ENDFILE 5
0295               ENDFILE 7
0296               FAIL=SQRT(-1)
0297          1998 CONTINUE
0298               WRITE(7,1108)(N,UBX(N),UBY(N),N=1,NUMGBP)
0299               CALL FREEFILE(1)
0300               CALL FREEFILE(2)
0301               CALL FREEFILE(3)
0302               CALL FREEFILE(4)
0303               CALL FREEFILE(5)
0304               ENDFILE 5
0305               ENDFILE 7
0306               WRITE(6,1110)
```

```
0307          10 FORMAT(2IO)
0308          11 FORMAT(I2,3I3)
0309          16 FORMAT(I2,2X,I2)
0310          17 FORMAT(2E15.6)
0311          22 FORMAT(4E18.9)
0312          26 FORMAT(E15.5,5X,E15.5)
0313          27 FORMAT(2E15.6)
0314          40 FORMAT(3IO,2EO.O)
0315          41 FORMAT(4I5)
0316          42 FORMAT(I3,2X,E15.6,2X,E15.6)
0317          43 FORMAT(E12.5,2X,E12.5)
0318          60 FORMAT(3IO)
0319          61 FORMAT(IO,2EO.O
0320          62 FORMAT(2IO,EO.O)
0321          999 FORMAT(2OH1PART B.  INPUT DATA)
0322          1000 FORMAT ( 2H  BCD  INFORMATION
0323              1                    )
0324          1001 FORMAT(I2,1X,I3)
```

```
0325          1100 FORMAT(25HONUMBER OF SUBSTRUCTURES=,I2//34H NUMBER OF GLOBAL BOUND
0326             1ARY POINTS=,I3//38H NUMBER OF RESTRAINED BOUNDARY POINTS=,I3//
0327             222H CYCLE PRINT INTERVAL=,I3//28H OUTPUT INTERVAL OF RESULTS=,I3//
0328             313H CYCLE LIMIT=,I4//17H TOLERANCE LIMIT=,F6.3//
0329             424H OVER-RELAXATION FACTOR=,F6.3//)
0330          1101 FORMAT(60HO    GNP        X-LOAD        Y-LOAD        X-DISP        Y-
0331             1DISP)
0332          1102 FORMAT(I8,2F13.3,2F13.8
0333          1103 FORMAT(31HORESTRAINED BOUNDARY CONDITIONS,//27H    GNP     NFIX
0334                2   SLOPE)
0335          1104 FORMAT(I7,I9,3X,F8.4)
0336          1105 FORMAT(40H1        CYCLE     SUM OF FORCE UNBALANCE)
0337          1106 FORMAT(I12,5X,E20.6)
0338          1107 FORMAT(51HOGLOBAL BOUNDARY N.P. X-DISPLACEMENT Y-DISPLACEMENT)
0339          1108 FORMAT(I21,2E15.6)
0340          1109 FORMAT18HOEND OF INPUT DATA)
0341          1110 FORMAT(18HOPART B.  COMPLETE)
0342          1112 FORMAT(20HOADJACENT NODES ARE.)
```

```
0343          1113 FORMAT(I20)
0344         1115 FORMAT(34H OVER 35 N.P. ADJACENT TO N.P. NO.,I4)
0345         1402 FORMAT(1H1,22HITERATION CONVERGES IN,I3,6HCYCLES,5X,
0346             123HSUM OF FORCE UNBALANCE=,E20.6,//
0347         1403 FORMAT(32HOITERATION FAILS TO CONVERGE IN ,I4,7H CYCLES,
0348             129H      SUM OF FORCE UNBALANCE=,E20.6)
0349              STOP
0350              END
0351         FINISH
```

5.4 Program PART C.

```
0001                SEND TO (ED,PROGRAM DUMP.FORTY)
0002                RUN
0003                MAP
0004                PROGRAM(PDEC16021PDE)
0005                USE 4=MT1/FORMATTED(ACW)
0006                USE 5=MTO/FORMATTED(AWMT)
0007                OUTPUT 6=LPO
0008                END
0009                MASTER FINITE PART C
0010          C     PART C. OF FINITE ELEMENT PROGRAM
0011          C     WITH SUBSTRUCTURAL ANALYSIS
0012          C     PLANE STRESS SOLUTION
0013          C     USING TRIANGULAR ELEMENTS
0014          C     WITH IN PLANE FORCES.
0015                   DIMENSION NGBP(54),UBX(54),UBY(54),
0016                  1NUME(65),NPI(65),NPJ(65),NPK(65),
0017                  2AJ(65),AK(65),BJ(65),BK(65),SIGXX(65),SIGXY(65),SIGYY(65),
0018                  3NPNUM(41),XORD(41),YORD(41),NOTE(41),NCODE(41),SM(50,32)
```

```
0019          4,UX(41),UY(41)
0020           READ(4,1000)
0021           READ(4,1001)NUMSUB,NUMGBP
0022           READ(4,500)(NGBP(I),UBX(I),UBY(I),I=1,NUMGBP)
0023           READ(5,501)E,XU
0024           DO2000JX=1,NUMSUB
0025           READ(5,505)NSUB,NUMEL,NUMNP,NUMBP
0026           READ  (5,1002) (NUME(N),NPI(N),NPJ(N),NPK(N),N=1,NUMEL)
0027           READ (5,1003) (NPNUM(N),XORD(N),YORD(N),N=1,NUMNP)
0028            READ(5,510)(NOTE(I),NCODE(I),I=1,NUMNP)
0029            DO100I=1,NUMNP
0030            UX(I)=0.0
0031            UY(I)=0.0
0032          100 CONTINUE
0033             NUMBER=2*(NUMNP-NUMBP)
0034             DO512I=NUMBP+1,NUMNP
0035             II=NOTE(I)
0036             READ(5,511)UIX,UIY
```

```
0037                UX(II)=UIX
0038                UY(II)=UIY
0039            512 CONTINUE
0040                DO515L=2,2*NUMBP,2
0041                READ(5,520)(SM(I,L-1),SM(I,L),I=1,NUMBER)
0042                READ(5,521)X1,X2
0043                DO515LL=2,2*NUMBP,2
0044                READ(5,525)X3,X4,X5,X6
0045            515 CONTINUE
0046                DO 1180 N=1,NUMEL
0047                I=NPI(N)
0048                J=NPJ(N)
0049                K=NPK(N)
0050                AJ(N)=XORD(J)-XORD(I)
0051                AK(N)=XORD(K)-XORD(I)
0052                BJ(N)=YORD(J)-YORD(I)
0053                BK(N)=YORD(K)-YORD(I)
0054            1180 CONTINUE
```

```
0055                DO150I=1,NUMBP
0056                J=NOTE(I)
0057                JJ=NCODE(I)
0058                UX(J)=UBX(JJ)
0059                UY(J)=UBY(JJ)
0060            150 CONTINUE
0061                L=0
0062                DO200I=NUMBP+1,NUMNP
0063                L=L+2
0064                II=NOTE(I)
0065                M=0
0066                DO205J=2,2*NUMBP,2
0067                M=M+1
0068                MM=NOTE(M)
0069                UX(II)=UX(II)-SM(L-1,J-1)*UX(MM)-SM(L-1,J)*UY(MM)
0070                UY(II)=UY(II)-SM(L,J-1)*UX(MM)-SM(L,J)*UY(MM)
0071            205 CONTINUE
0072            200 CONTINUE
```

```
0073                WRITE(6,1121)
0074                WRITE (6,1000)
0075                WRITE(6,1130)NSUB
0076                WRITE (6,1107)
0077                WRITE(6,1108)(NPNUM(N),UX(N),UY(N),N=1,NUMNP)
0078                WRITE(6,1131)NSUB
0079                WRITE (6,1109)
0080                DO 1425 N=1,NUMEL
0081                I=NPI(N)
0082                J=NPJ(N)
0083                K=NPK(N)
0084                EPX=(BJ(N)-BK(N))*UX(I)+BK(N)*UX(J)-BJ(N)*UX(K)
0085                EPY=(AK(N)-AJ(N))*UY(I)-AK(N)*UY(J)+AJ(N)*UY(K)
0086                GAM=(AK(N)-AJ(N))*UX(I)-AK(N)*UX(J)+AJ(N)*UX(K)
0087               1  +(BJ(N)-BK(N))*UY(I)+BK(N)*UY(J)-BJ(N)*UY(K)
0088                COMM=E/((1.0-XU**2.0)*(AJ(N)*BK(N)-AK(N)*BJ(N)))
0089                X=COMM*(EPX+XU*EPY)
0090                Y=COMM*(EPY+XU*EPX)
```

```
0091                XY=COMM*GAM*(1.0-XU)*0.5
0092                SIGXX(N)=X
0093                SIGYY(N)=Y
0094                SIGXY(N)=XY
0095                C=(XY)2.0
0096                R=SQRT(((Y-X)/2.0)**2+XY**2)
0097                XMAX=C+R
0098                XMIN=C-R
0099                PA=0.5*57.29578*ATAN(2.0*XY/(Y-X))
0100                    IF (2.0*X-XMAX-XMIN) 1405,1420,1420
0101               1405 IF (PA) 1410,1420,1415
0102               1410 PA=PA+90.0
0103                    GO TO 1420
0104               1415 PA=PA-90.0
0105               1420 WRITE (6,1110) NUME(N),X,Y,XY,XMAX,XMIN,PA
0106               1425 CONTINUE
0107                    WRITE(6,1131)NSUB
0108                    WRITE (6,1111)
```

```
0109                DO 1900 M=1,NUMNP
0110                DO1901MM=1,NUMNP
0111                IF(NOTE(MM).EQ.M)GOTO1902
0112          1901  CONTINUE
0113          1902  IF(NCODE(MM).NE.0)GOTO1900
0114                X=0.0
0115                Y=0.0
0116                XY=0.0
0117                SRX=0.0
0118                SRY=0.0
0119                R=0.0
0120                DO 1860 N=1,NUMEL
0121                I=NPI(N)
0122                J=NPJ(N)
0123                K=NPK(N)
0124                IF (M-I) 1830,1850,1830
0125          1830  IF (M-J) 1835,1845,1835
0126          1835  IF (M-K) 1860,1840,1860
```

```
0127          1840 I=NPK(N)
0128               K=NPI(N)
0129               GO TO 1850
0130          1845 I=NPJ(N)
0131               J=NPI(N)
0132          1850 A1=ABS(XORD(J)+XORD(K)-2.0*XORD(I))
0133               B1=ABS(YORD(J)+YORD(K)-2.0*YORD(I))
0134               RY=B1/(A1B1)
0135               SRY=SRY+RY
0136               Y=Y+SIGYY(N)*RY
0137               RX=A1/(A1+B1)
0138               SRX=SRX+RX
0139               X=X+SIGXX(N)*RX
0140               R=R+1.0
0141               XY=XY+SIGXY(N)
0142          1860 CONTINUE
0143               X=X/SRX
0144               Y=Y/SRY
```

```
0145                XY=XY/R
0146                C=(X+Y)2.0
0147                R=SQRT(((Y-X)/2.0)**2+XY**2)
0148                XMAX=CR
0149                XMIN=C-R
0150                PA=0.5*57.29578*ATAN(2.0*XY/(Y-X))
0151                IF (20*X-XMAX-XMIN) 1805,1820,1820
0152          1805  IF (PA) 1810,1820,1815
0153          1810  PA=PA+90.0
0154                GO TO 1820
0155          1815  PA=PA-90.0
0156          1820  WRITE(6,1112)M,X,Y,XY,XMAX,XMIN,PA
0157          1900  CONTINUE
0158          2000  CONTINUE
0159                WRITE(6,9999)
0160                ENDFILE 4
0161                ENDFILE 5
0162           500  FORMAT(I21,2E15.6)
```

```
0163          501 FORMAT(E12.5,2X,E12.5)
0164          505 FORMAT(I2,3I3)
0165          510 FORMAT(I2,2X,I2)
0166          511 FORMAT(2E15.6)
0167          520 FORMAT(E15.5,5X,E15.5)
0168          521 FORMAT(2E15.6)
0169          525 FORMAT(4E18.9
0170          999 FORMAT(2I0)
0171         1000 FORMAT (72H  BCD  INFORMATION
0172              1                        )
0173         1001 FORMAT(I2,1X,I3)
0174         1002 FORMAT(4I5)
0175         1003 FORMAT(I3,2X,E15.6,2X,E15.6)
0176         1004 FORMAT (2I0,E0.0)
0177         1107 FORMAT (42HONODAL POINT X-DISPLACEMENT Y-DISPLACEMENT)
0178         1108 FORMAT (I12,2E15.6)
0179         1109 FORMAT(120HO  ELEMENT                X-STRESS                Y-STRESS
0180              1       XY-STRESS          MAX-STRESS      MIN-STRESS      DIRECTION)
```

```
0181          1110 FORMAT (I10,3F20.4,5X,3F15.2)
0182          1111 FORMAT(120HOINTERIOR N.P.                X-STRESS                Y-STRESS
0183             1         XY-STRESS          MAX-STRESS      MIN-STRESS     DIRECTION)
0184          1112 FORMAT(I14,3F19.4,5X,2F15.2,F13.2)
0185          1121 FORMAT(21H1PART C.  OUTPUT DATA)
0186          1130 FORMAT(13HOSUBSTRUCTURE,I3)
0187          1131 FORMAT(13H1SUBSTRUCTURE,I3)
0188          9999 FORMAT(18HOPART C,  COMPLETE)
0189               STOP
0190               END
0191               FINISH
```

ACKNOWLEDGMENTS

The author wishes to express his gratitude to Dr. T. Allan of the Mechanical Design Department for guidance during this project.

Thanks are also due to Mrs. Nicol, Mr. D. Evans, and personnel of the Computer Science Department for their guidance in the preparation of the computer programs.

REFERENCES.

1/ Zienkiewkz, O.C. , "The Finite Element Method in Structural and Continuum Mechanics," McGraw-Hill, 1967.

2/ Prizemieniecki, J.S., "Theory of Matrix Structural Analysis," McGraw-Hill, 1968.

3/ Prizemieniecki, J.S. , "Matrix Structural Analysis of Substructures," J. Am. Inst. Aeron. Astron., 1:138-147(1963).

4/ Prizemieniecki, J.S., "Analysis of Aircraft Structures by the Substructure Relaxation Method, " Bristol Aircraft Ltd. England, Tech. Rept. TOR 145, 1961.

5/ Wilson, E.L. "Finite Element Analysis of Two-Dimensional Structures," University of California, Rept. No. 63-2 (1963).

6/ Holand, I. and Bell, K., "Finite Element Methods in Stress Analysis," Tapir, Universtiy of Norway, 1969.

7/ Welbourne, D., "Numerical and Matrix Methods in Structural Mechanics," Commonwealth International Library.

8/ Jenkins, W.M., "Matrix and Digital Computer Methods in Structureal Analysis," McGraw-Hill, 1969.

9/ Turner, M. j., Martin, H. C., and Weikel, R. C., "Further Development and Applications of the Stiffness Method," AGARD, Structures and Materials Panel, Paris, France, July 1962.

APPENDIX

Matrix Inversion by Partitioning.

APPENDIX 1.

Matrix Inversion by Partitioning. (REF. 7).

Consider the matrix relationship

$$A\,B = I \tag{A1}$$

where $B = A^{-1}$ and I is a unit matrix.

writing (A1) in partitioned form

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A2}$$

expanding out we have

$$A_{11}\,B_{11} + A_{12}\,B_{21} = 1 \tag{A3a}$$

$$A_{11}\,B_{12} + A_{12}\,B_{22} = 0 \tag{A3b}$$

$$A_{21}\,B_{11} + A_{22}\,B_{21} = 0 \tag{A3c}$$

$$A_{21}\,B_{12} + A_{22}\,B_{22} = 1 \tag{A3d}$$

from (A3b) we have

$$B_{12} = -A_{11}^{-1}\,A_{12}\,B_{22} \tag{A4}$$

substitute (A4) into (A3d) we have

$$B_{22} = \left(A_{22} - A_{21}\,A_{11}^{-1}\,A_{12} \right)^{-1}$$

since $B\,A = I$ then

$$B_{21}\,A_{11} + B_{22}\,A_{21} = 0$$

$$B_{21} = -B_{22}\,A_{21}\,A_{11}^{-1}$$

$$B_{11} = A_{11}^{-1} - A_{11}^{-1}\,A_{12}\,B_{21}$$

summerising

Invert A_{11} to obtain A_{11}^{-1}

compute $A_{11}^{-1}\,A_{12}$

compute $A_{21}\,A_{11}^{-1}$

compute $A_{21}\, A_{11}^{-1}\, A_{12}$

compute $A_{22} - A_{21}\, A_{11}^{-1}\, A_{12}$.

then the inverted matrix is

$$B_{22} = \left(A_{22} - A_{21}\, A_{11}^{-1}\, A_{12} \right)^{-1}$$

$$B_{12} = -A_{11}^{-1}\, A_{12}\, B_{22}$$

$$B_{21} = -B_{22}\, A_{21}\, A_{11}^{-1}$$

$$B_{11} = A_{11}^{-1} - A_{11}^{-1}\, A_{12}\, B_{21}$$