THE ANALYSIS AND DESIGN OF STRUCTURES

WITH VARYING SECTION PROPERTIES

by

B.K. WILLCOCK, D.C.T. (Leeds). C.Eng.

A.M.I.Struct.E.

THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DECEMBER 1966

Department of Civil Engineering,

The University of Leeds.

Acknowledgements.

The author wishes to express his thanks to Professor R.H. Evans, C.B.E., D.Sc., D.es.Sc., Ph.D., M.I.C.E., M.I.Mech.E., M.I.Struct.E., the Head of the Civil Engineering Department of Leeds University and Mr. F. Sawko, M.Sc., under whose supervision this work was carried out, for the continual help, encouragement and constructive advice.

Thanks are due to Mr. S. Rider the Chief Technician, for his advice given during the model testing and to Mr. H. Robinson for his care and attention to detail during the construction of the perspex model.

The author is also indebted to the members of the Advisory Service at the Leeds University Computing Laboratory for the assistance and advice given during the writing of the computer programs.

Financial assistance was given by the Scientific Research Council.

CONTENTS

Page

Chapter 1

Introduction

1.1) Analysis Methods

Structures composed of members with varying section properties are quite often encountered in engineering practice. Two reasons for this are, firstly the more elegant appearance of the structure resulting from the use of non-uniform members, and secondly a more efficient use of materials, as the profile of the members can reflect more honestly the distribution of stresses throughout the structure. The variation in section can be along the full length of the member, as in precast concrete or welded steel portal frames which have tapered legs and rafters, or the member may vary in section only at the ends when haunch stiffeners are employed. Large multispan bridges often consist of longitudinal girders with a variation in depth giving an elegant arch like appearance to the structure. In the case of large bridge structures where dead weight forms a greater proportion of the load, the improved structural efficiency resulting from the use of varying section members, can reduce the overall weight of the structure thereby making further savings in the cost of the supporting structure and foundations.

Before the availability of electronic computers the analysis of all but the simplest forms of structures with varying section properties required considerable time and skill. Many of the established methods of structural analysis can be applied to frames composed of varying section members and many modifications have been introduced to enable specific types of structure to be analysed by hand.

Column analogy can easily be applied to beams and portal frames with stepped members or members with a linear variation in the moment of inertia. The method becomes tedious and lengthy as the variation in moment of inertia becomes more complicated. Robinson (3) demonstrated the method of semi-graphical integration which can be applied to non-prismatic members in plane frames, but again any complex variation in the moment of inertia will make the calculations extremely laborious. The moment distribution method can be applied to varying section members once the distribution and carry over factors have been obtained. Many standard text books contain tabulated values of stiffness and carry over factors, and values of fixed end moments for beams with straight, prismatic or parabolic haunches. Computer programmes have also been written to calculate these coefficients (4). The method can be applied to continuous beams or plane frame structures, although the rate of convergance is sometimes rather slow where members have comparatively high carry over factors. A method proposed by Fok and Mosuro (5) uses the tabulated values of stiffness and carry over factors to form three moment equations, which are solved to give the redundant moments. The method is only suitable for continuous beams and offers little advantage over the normal moment distribution process. Diwan (6) also uses the three moment equations to derive bending moments, and demonstrates that influence lines can be obtained by applying unit displacements to the ends of members. The application is again restricted to continuous beams. Sami (7) has evolved a method of solution by using redundent coefficients, which are presented in tabular form and substituted into a set of redundant moment equations. A computer programme has been written to derive the coefficients. A different set are required for each degree of redundancy and for each set of span ratios. The method has also been restricted to prismatic beams with parabolic depth variation. Without considerable amounts of tabulated data the use of this method is limited.

All the preceding methods have been developed and presented as hand methods for the analysis of plane frame structures. All of the methods make simplifying assumptions or rely on tabulated data to enable acceptable results to be obtained. In all cases only bending strains have been considered, but to obtain an accurate solution in all cases it is necessary to also consider axial strains. Sawko has shown (8) that in the case of a vierendeel girder, an error in the value of bending moment of up to 34% is committed when axial strains are neglected.

Another important class of structure often requiring analysis is the grid frame. Because the high degree of indeterminancy makes an exact solution by hand extremely laboMrious, several other methods have been developed but, as far as is known, only two of these make provision for non-prismatic members. The most widely used method is that by Guyon and Massonet in which the grillage is considered as an orthotropic plate of equivalent flexural and torsional stiffness. Morice and Little (9) produced a design procedure and presented the coefficients in graph form. The method has since been used by Goldstein, Lightfoot and Sawko (10) to analyse three span continuous grillage bridges with varying section properties, by using the simply supported span technique. A discrete girder is first analysed under the action of the total load to determine the bending moment along the beam, points of contraflecture and deflected profile. The central portion of the deck between the points of contraflecture, assumed to lie at right angles to the longitudinal girder, is considered as an equivalent simply supported slab. The variation in moment of inertia is small at the centre of the bridge and an average value is considered when calculating the bending and torsional stiffness parameters. Distribution coefficients are then obtained which, when multiplied by the average moment per girder, give the distribution of longitudinal bending moment. The deflected profile is also obtained using the distribution coefficients but the deflections obtained are related to the line of contraflecture and not the original position of the deck. Because only the first term of the loading series is considered, an increase of 10% to the maximum values of bending moment and deflection is recommended. The same coefficients obtained for the middle of the central span are assumed to apply to all other parts of the bridge, and hence, bending moments over the piers and in the end spans are obtained. Coefficients can be obtained to give the transverse bending moment distributions at any point in the quasi slab, but no recommendations are given for the treatment of side spans. This method has been used for the design of a right bridge (10) and a 24° skew beidge (32), although in the latter case model tests were also conducted as an additional check on the results.

The Hendry Jaeger method (11) relies upon the application of harmonic and basic function analysis. This method is more powerful than the former; grids supported on two, three or four sides may be analysed with fixed or free support conditions. The method can be applied to both skew and continuous grillages and it also allows for some variation in longitudinal stiffness. Basic function analysis is used for any type of grid with any boundary conditions, whereas the harmonic analysis, can be used only for simply supported grids. Coefficients presented in the form of charts and graphs make the method easy to apply. A comparison between the Hendry Jaeger method and the Distribution Coefficient method by Sawko and Saha (12) has shown that the results given by the former method were 8% higher in the case of longitudinal bending moment and 15% higher in the case of transverse bending moment for a single span uniform bridge deck. No comparison is available for continuous grillages with varying section properties.

The methods outlined above for the analysis of grid and plane frame structures are all hand methods and, although they are suitable for use in a design office in slide rule calculations, the methods are not easily programmed to give a general solution using the electronic computer. The advent of the computer brought about a change in structural analysis methods. Although matrix methods had been known for many years, their use had been restricted to very simple structures because of the large amounts of repetitive numerical manipulation involved. The computer is, however, able to perform this type of work extremely rapidly.

Madu (13) has applied a finite difference approach to plates of varying depth, using an electronic computer to solve the equations. The method has been applied to grid frames with a variation in depth but the application was restricted to simply supported right grillages.

From the use of matrix methods two different approaches have emerged. The flexibility method (14) renders the structure statically determinate by applying stress resultant releases. The displacement discontinuities can then be removed by the application of bi-actions. The values of the redundants are then determined by the solution of a set of 'n' simultaneous equations, where 'n' is the degree of indeterminancy. The solution of the problem is reduced to a series of methodical matrix operations. By making use of the matrix interpretive schemes available as part of the softwear of most computers, these operations can be performed rapidly and accurately. In this case the structure is reduced to its statically determinate state and the basic matrices formed, by hand. It is for this reason that the flexibilty method is not easily programmed to give a general solution. It is however, possible to programme specific cases and this method is used to calculate member properties in the program described in Section (2.3).

The stiffness method (15) renders the structure kinematically determinate by clamping each of the nodes. This usually results in stress resultant discontinuities at these nodes. The discontinuities can then be removed either by iteration, as in HardyCross moment distribution, or by solving a set of linear simultaneous equations, as in slope deflection. Krynicki and Mazurkirwics (16) have shown how slope deflection equations can be applied to the solution of frames consisting of solid bars of varying cross section. The method relies upon tabulated coefficients and is, therefore, a hand method which cannot be used to give a general solution.

The computer can be used to set up and solve a generalised set of slope deflection equations, and in this way an automatic solution can be programmed. Unlike the flexibility method, the degree of indeterminacy need not be known and the data need consist only of details of the frame geometry, applied loading and support conditions. Programmes have been written to analyse grid and plane frameworks (17, 18) using this method, but previously mainly prismatic members have been considered. It is possible to simulate a varying section member by considering it as a series of shorter uniform members, thus forming a stepped member. This procedure has the disadvantage of increasing the number of joints and hence the time for solution. Existing programmes have also been adapted to analyse specific cases. Litton, Roper and Thompson (19) describe a method of modification to consider members with sloping haunches at each end. The members are considered in terms of equivalent section properties, but each member is required to have two values of moment of inertia to give the correct bending and shearing stiffnesses. A method by which standard computer programmes for the analysis of frames with uniform members, may be used for the analysis of frames consisting of symmetrical non-uniform members, has been derived by Sawko (35). The member stiffness and carry over factors are first obtained by hand calculations or from standard tables. By using the published graphs an equivalent member is obtained, which has uniform haunches at each end. The dimensions of the haunches are obtained so that the equivalent member can be analysed exactly, by considering it as three separate uniform members; additional joints have to be inserted at the points of discontinuity. The two preceding methods are restricted in application to the particular types of non-uniform members for which they were specifically written.

After considering all the methods outlined above, it was apparent that there was a need for a method capable of analysing structures composed of varying section members, in which no restriction would be placed upon either the type of members, or degree of variation in section along the members. The existing hand methods are only suitable for small structures and many of the methods also impose restrictions upon the type of members. The computer programmes could in certain cases produce an 'exact' analysis, but in other cases it was necessary to introduce certain simplifying assumptions. The flexibility method, whilst providing an accurate means of analysis, becomes cumbersome when applied to highly redundant structures because of the nature of the necessary data.

Within this thesis computer programmes are developed for the accurate analysis of structures with varying section members. Two types of structure are considered: the plane frame and the grid frame. Programmes to calculate influence lines or surfaces for these two types of structures are also developed. The main application of these programmes is thought to be in the analysis of building frames and bridges. If the facilities are available to analyse accurately structures where the members may have any required variation in section, the engineer will not be deterred from producing an elegant and efficient structure, for fear that it cannot be analysed with sufficient accuracy. This is particularly the case with bridge engineering, where modern techniques in prestressed concrete and welded steel have enabled engineers to build large multispan bridges of slender proportions, and therefore, an accurate method of analysis is required to ensure adequate safety.

A series of model tests are described, which were conducted to investigate the accuracy of the grid frame programme, and the degree of approximation involved when the grillage with varying sections is used as a mathematical model to simulate plated structures.

In a recent paper (20) the author and his supervisor Mr.F. Sawko have demonstrated how the computer techniques developed in the investigation can be applied to bridges with varying section properties.

1.2) Design Methods

Since the advent of the electronic computer, programmes have been developed to analyse many types of structures, so that at the present time it is possible for the majority of structures to be analysed either 'exactly' or by using a convenient mathematical model. In the normal design process these programmes are used for the analysis of trial systems. There is no direct method of designing statically indeterminate structures and a 'trial and error' approach must be adopted. The structure is first designed approximately by hand, to obtain a set of dimensions and section sizes. It is then analysed, using an electronic computer to determine the stress distribution. If at any point the strucutre is overstressed the section sizes are modified and the structure is re-analysed, taking into account the changes in member stiffnesses. When using a computer to check trial systems it is usual to accept a solution which is over designed, as being practically correct. The initial design of a large structural system is a lengthy process involving a considerable amount of hand calculation. This process is often the most expensive part of the design, as computer analysis is now very rapid and economical.

Consideration of these facts led to the development of automatic design programmes, the computer being ideally suited to this type of work. It is able to perform repetitive calulations extremely rapidly and a large number of trial and error cyles can be applied to statically indeterminate structures, thereby arriving at an economical solution. Many of the routine preliminary calculations can be performed by the computer, so that the overall cost of the design is also reduced.

During the first decade that computers were available, efforts were concentrated upon writing programmes for the analysis of structures and it is only recently that automatic design programmes have been written. This is mainly due to the large amounts of random access store that is required by a design programme. The design process is usually an iterative procedure of trial and error analysis, and therefore, the full analysis procedure must be contained within the program, in addition to many other instructions. During the running of the program the computer is also required to calculate and store large amounts of information, and it is only in the past few years that computers of sufficient store have been available.

. Using modern computers many design programmes have been developed. Almost all of these programmes consist of elastic or plastic design of frame structures composed of uniform members and these were therefore considered to be outside the scope of this work.

The design of structures with varying section properties presents particular problems, as there are an infinite number of possible solutions to any one structural problem. So far very little work has been carried out on the automatic design of these structures. The Portland Cement Association (21) developed a design procedure for continuous prestressed concrete bridge beams with varying section properties. This approach was originally intended for use as a hand method but it was later used by Aziz (22) as the basis of a computer programme for the automatic design of bridge beams. The programme developed will design a single continuous prestressed concrete bridge girder, subjected to the HA loading specified by BS 153 (1), and the design produced is in accordance with CP 115 (2). The programme calculates relative EI values at all points along the girder and derives a set of influence lines for bending moment based upon the relative I values. The flexibility method is employed and the influence lines are formed by placing unit point loads at successive points along the girders. Using the influence lines, maximum and minimum live load bending moments are obtained. A ratio of dead/live bending moment is assumed and an initial set of section sizes is derived. From the actual sizes the bending moment ratio is checked and the section re-designed if necessary. The magnitude of the prestressing force is calculated and the cut off points for the cables established. The upper and lower bounds of the limiting zone are calculated and a cable profile designed, such that the line of thrust acts within this zone. The programme also calculates the total cost of the girder based upon the quantities of concrete, steel and shuttering. By using this programme the effects of span and depth ratios upon the cost of a bridge, were investigated. The programme produces a rigorous design for the HA loading and, as this loading acts on all girders in a bridge, the design need only consider a discrete girder. In order to examine the effects of the abnormal vehicle HB loading, the complete bridge must be considered, taking into account the distribution of load across the deck. The programme written by Aziz is unable to consider these effects because when the programme was developed a sufficiently large computer was not available.

The latter part of this work consists of an investigation into the problems of writing a comprehensive programme capable of automatically designing multispan highway bridges, with varying section properties, subjected to both HA and HB load systems. A study of the effects of variation in span lengths and relative EI values is described. The whole of the deck is considered under the action of both loading systems, and a range of parameters is investigated to determine the least weight design.

Chapter 2.

Computer Programs For The Analysis of Grid and Plane Frameworks

2.1) Introduction

The first program for the analysis of rigid jointed plane frames was devised by Livesley in 1953 (17) for the Ferranti Mark 1 machine at Manchester University. Several other plane frame programs were later written by different authors, including a program by Rooney, then of Babcock and Wilcox Ltd., for the Ferranti-Pegasus machine. The similarities between plane frameworks and grid frameworks were appreciated by Lightfoot and the program by Rooney was transcribed by Sawko to analyse grid frames in 1958 (18).

Although grid and plane frame structures are different in type the internal organisation of the programs is very similar. Both can be considered to be special cases of a general space frame structure, each one taking into accoumt the relevant strains. The basic difference is therefore in the stiffness matrix elements for individual members.

In order that the programs developed should be as efficient as possible they were written specifically for the KDF9 machine, which at present forms the basis of the installation at the Leeds University Computing Laboratory. The language Algol 60 (34) has been used for all programs. Algol is a more universally accepted language and one that the KDF9 compilers are able to translate efficiently. In rewriting the grid and plane frame programs in Algol, facilities were incorporated to deal accurately with varying section members. The internal organisation was also considerably modified, to produce a version making efficient use of the storage capacity of the computer, and also reducing the amount of data required. Only the grid framework program is described in detail because of the similarities between the two types of program. A copy of the grid framework program is contained in Appendix One.

2.11) The Electronic Computer

The installation at the Leeds University Computing Laboratory uses, as a basic machine, the English Electric Leo Marconi KDF9, which is a medium sized simultaneous digital computer. For most of the period of this research a main store of 16K, (16 x 1024 words) of 48 bit core store, was available. This has recently been increased to 32K units of main store, although not all of this is normally available for use by one program.

The times for the basic operations are as follows:-

Add	½ micro seconds	for 48 bit numbers.
Multiply	13 " "	
Divide	26 " "	

The main store has a cycle time of 6 microseconds and characters can be transferred between the main store and the arithmetic unit at 1.3×10^6 per second.

In addition to the main control machine there are several peripheral devices. Magnetic tape decks are used as additional storage units, as well as holding the range of soft-wear. Information is recorded simultaneously on 16 tracks across a 1" wide tape, with a density of 400 characters per inch. The transfer rate is 40,000 characters per second, which is considerably slower than within the main machine. The physical operations involved in using magnetic tape also greatly increase the time. The acceleration and deceleration of the tape takes 3-4 milli-seconds and the rewind time for a 2300 foot tape is 4 minutes. Paper tape readers have an input rate of 1000 characters per second and paper tape output punches have a process rate of 110 characters per second, or 12 words per second. This slow output rate has the effect of 'holding up' the central processing unit and an alternative faster method is to output results on to magnetic tape at a rate of 4000 words per second. The magnetic tape is processed through an off line printer, which has a maximum operating speed of 1000 lines per minute.

The central processing unit has the fastest operating speed and must, therefore, be adjusted through the input/output devices to accommodate the slowest piece of peripheral equipment in use. This means for the majority of the time it is being used at less than maximum efficiency. To overcome this deficiency the facility of time sharing can be used, whereby more than one program can be run 'simultaneously'. Whilst one program is transferring data from one phase to the next, the central processing unit takes over another program, thus alternating between several programs which are operating at the same time. This is why the whole of the fast store is not normally available for use by one program only; efficient use of the installation dictates that at least two programs should be running together.

Before the program can be obeyed by the computer, it must first be translated from the program language, in this case Algol, to the machine code. The KDF9 softwear contains two compilers to do this. The Whetstone compiler translates a programe rapidly, in a matter of seconds, and during this process performs a thorough check on the syntax of the Algol program. The resulting translation is, however, obeyed relatively slowly under the Whetstone controller. The Kidsgrove translater, by contrast, produces a translation which will be obeyed ten or more times faster, but the time taken to produce this more efficient version can be as much as ten minutes. The failure messages output under the Kidsgrove system are far less informative.

To make efficient use of both these compilers the program is first run using the Whetstone compiler, and use made of the comprehensive failure messages, until the author is satisfied that the program is entirely correct. It can next be translated using the Kidsgrove compiler and the resulting translated version stored on magnetic tape. It can then be used subsequently, without the need for a lengthy re-translation and the results are also produced in the quickest time. The program is given a unique title and placed in a common library of programs. Any data to be run with a program stored on magnetic tape is headed by a 'call' sequence, which causes the computer to search for the required program and transfer it into the main store.

All the programs described in later chapters are available in Kidsgrove binary versions and the operating times quoted are obtained using this compiler.

2.2) Assumptions

The programs analyse rigid jointed structures composed of straight members of either varying or uniform section properties. The structures are considered to be linear, i.e., in which all displacements and internal loads are linear functions of the applied loads.

The bending theory equation $\frac{EId^2y}{dx^2} = M$ is assumed to apply so that slope deflection equations can be written for every member in the frame. The frame is analysed in its undeflected form so that the small deflections, induced through the curvature of the members under load, are neglected and direct forces in grillages are not considered. The effects of axial loading, in plane frameworks, upon the stiffness and restraint factors is similarly neglected. Force equilibrium equations are set up for each joint, taking into account bending and torsional moments and shear force for grid frameworks, or axial and shear force and bending moments for plane frameworks. Consequently external loads have to be considered as acting at joints only. Any system of loads can, however, be expressed in terms of fixed end moments and shears. The final forces are calculated by superimposing the local member effects on to the results obtained from the computer.

2.3) Basic Framework Equations

2.3.1) Grid Framework

Consider a general member 1-2, as shown in Fig. (2.1) forming part of a grid framework. The member is considered relative to its own co-ordinate system where axis p runs along the member, axis q at right angles and the z axis normal to the plane.

Applying the basic slope deflection equations to member 12 At end 1:-

$$M_{q12} = K_{12}\theta_1 + C_{21}K_{21}\theta_2 - \left(K_{12} + C_{21}K_{21}\right)\left(\frac{\delta_{z1} - \delta_{z2}}{L}\right) \qquad \text{....(2.1)}$$

in which K_{ij} is the bending stiffness at end i

C_{ij} is the carry over factor from end i to end j.

For a uniform member the bending stiffness i.e., moment to produce a unit rotation with the far end fully fixed, is $\frac{4EI}{L}$, where E is Young's modulus and I is the effective second moment of area about the bending axis, and L is the length of the member. The value of carry over for a uniform member is $\frac{1}{2}$.

The torsional moment can be expressed as:-

$$M_{p12} = T\,(\theta_{p1}-\theta_{p2}) \qquad \ldots (2.2)$$

in which T is the torsional stiffness of the member.

For a uniform member T equals $\frac{GJ}{L}$, where G is the rigidity modulus, J is the effective second moment of area about the torsion axis, and L is the length of the member.

Similarly at end 2:-

$$M_{q21}=K_{21}\theta_2+C_{12}K_{12}\theta_1 - K_{21}+C_{12}K_{12})\ \frac{(\delta_{z1}-\delta_{z2})}{L} \qquad \ldots (2.3)$$

Shear force $F_{z12} = \frac{M_{q12}+M_{q21}}{L}$

and $F_{z21} = -F_{z12}$

The above equations are summarised in matrix form in Table (2.1) which can be expressed briefly as:-

$$F_1 = K_{12}D_1 + R_{12}D_2$$
$$F_2 = R_{21}D_1 + K_{21}D_2 \quad \ldots (2.4)$$

in which K_{ij} is the stiffness matrix.

R_{ij} is the restraint matrix.

or simply as:-

$$\left[F_{pq}\right] = \left[K\right]\left[D_{pq}\right] \quad \ldots (2.5)$$

These equations refer to moments and forces relative to the individual member displacements. In order to relate all the members of the frame, it is necessary to be able to express moments and forces relative to an arbitrary system of general co-ordinates, denoted by x and y.

If member 12 is inclined at an angle α to the general co-ordinate system as shown in Fig. (2.2a) it can be seen that:-

$$M_{x1} = M_{p12}\cos\alpha - M_{q12}\sin\alpha$$

$$M_{y1} = M_{p12}\sin\alpha + M_{q12}\cos\alpha$$

The force F_z remains unchanged as it acts normal to the axis of transformation.

The transformation matrix T thus becomes:-

$$\begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the transformed member forces can be expressed briefly as:-

$$\left[F_{xy}\right] = \left[T\right]\left[F_{pq}\right] \qquad \text{... (2.6)}$$

Similarly the relationship between displacement vectors can be expressed as:-

$$\left[D_{pq}\right] = \left[T\right]^T \left[D_{xy}\right] \qquad \text{... (2.7)}$$

This may be done by contragradience or from geometrical considerations.

By substituting Eqn. (2.7) to Eqn. (2.5):-

$$\left[F_{pq}\right] = \left[K\right]\left[T\right]^T\left[D_{xy}\right] \qquad \text{... (2.8)}$$

and by substituting Eqn. (2.8) to Eqn. (2.6) the transformed moments and forces can be expressed as:-

$$\left[F_{xy}\right] = \left[T\right]\left[K\right]\left[T\right]^T \left[D_{xy}\right] \qquad \text{... (2.9)}$$

Thus Equations (2.4) can be written:-

$$F_1' = K_{12}'D_1' + R_{12}'D_2'$$

$$F_2' = R_{21}'D_1' + K_{21}'D_2' \qquad \text{... (2.10)}$$

$$\text{where } K_{ij}' = [T][K][T]^T$$

$$\text{and } R_{ij}' = [T][R][T]^T \qquad (i,j = 1,2)$$

The complete set of generalised slope deflection equations are given in Table (2.2)

If the external forces F_1^E acting upon joint one, are considered together with all members meeting at joint one, the conditions of static equilibrium may be written:-

$$F_1^E = \sum(K_{12}')\,D_1' + \sum(R_{12}'D_2') \qquad \ldots\ldots (2.11)$$

Equations can be formed in the same way for every joint in the frame thus enabling all the equivalent external forces to be related to the various displacements in the general form:-

$$[F] = [K][D] \qquad \ldots\ldots (2.12)$$

where $[F]$ represents the force vector, $[K]$ is the stiffness matrix for the whole structure and $[D]$ represents the displacement vector.

It is therefore possible for a solution to this equation to be found to give joint displacements in the frame. Terminal forces and moments in the x, y directions can be calculated by back substitution in the equations given in Table (2.2) and by simple resolution the forces and moments in the member direction are found. Alternatively the displacements can be resolved into the individual member directions and back substituted into equations given in Table (2.1) to give the same final result.

2.3.2) Plane Framework Equations

Consider a general member 1-2, forming part of a plane framework, as shown in Fig. (2.3). In the same way that equations were set up for bending and shear in a grid framework, slope deflection equations can also be set up for bending moment and shear force for a member in a plane framework. As the load is now acting in the plane of the structure there will be no deformations due to torsion. There will however, be deformations caused by the axial loads in members and these may be expressed as:-

$$S_{12} = A(\delta_{p1} - \delta_{p2})$$

and $$S_{21} = -S_{12}$$

where A is the axial stiffness of the member which would be $\frac{Ea}{L}$ for a uniform member, where 'a' is the cross sectional area.

The slope deflection equations for an individual member are summarised in Table (2.3)

As before it is necessary to be able to refer to a member relative to a general system of co-ordinates. If member 12, as shown in Fig (2.4) is inclined at an angle α to the general co-ordinate axis then it can be seen that:-

$$S_{x1} = S_{p12} \cos\alpha - Q_{p12} \sin\alpha$$

$$Q_{y1} = S_{p12} \sin\alpha + Q_{p12} \cos\alpha$$

The moment M_1 remains unchanged by the transformation.

The transformation matrix T is therefore;-

$$\begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equations (2.10) are still valid and the full set of generalised slope deflection equations can be calculated as before. These are shown in Table (2.4).

By considering every member in the frame a set of equations can be built up, finally culminating in the form:-

$$[F] = [K][D]$$

The solution of this equation gives the joint displacements which are back substituted in the equations in Table (2.4) to give the forces and bending moment in the x,y directions. These are resolved to give the terminal forces and moments in the p,q directions.

2.4) Determination of Member Properties and Fixed End Moments

For every member in a frame it is necessary to know the bending stiffness and carry over factor at each end. The torsional stiffness and axial stiffness are also required in the case of grid frameworks and plane frameworks respectively. These properties are required in order to set up the slope deflection equations and are calculated using strain energy to derive influence coefficients for the end conditions. The derivation given here is based upon that given by Morice (31).

Consider a fixed ended beam as shown in Fig. (2.5a). The structure is two times statically indeterminate and is released by relaxing the end fixities. The end fixing moments are then made arbitrary constants x_1 and x_2. The bending moments caused by unit values of x_1 and x_2 will be as shown in Fig. (2.6b) and (2.6d). The total bending moment acting at any point on the beam will be that due to the applied loading on the released structure plus that due to the arbitrary constants:-

$$M = m_o + m_1 x_1 + m_2 x_2$$

The total strain energy in the beam will be:-

$$U = \int_s \frac{M^2}{EI}\, ds. \qquad \ldots (2.13)$$

By applying the theorum of least work, two equations for the determination of x_1 and x_2 can be obtained:-

$$\frac{\partial U}{x_1} = \int_s \frac{\partial}{\partial x_1}\left(\frac{M^2}{2EI}\right) ds = 0$$

$$\frac{\partial U}{\partial x_2} = \int_s \frac{\partial}{\partial x_2}\left(\frac{M^2}{2EI}\right) ds = 0 \qquad \ldots (2.14)$$

Replacing M by its components, equations (2.14) become:-

$$\frac{\partial U}{\partial x_1} = \int_s \frac{m_1}{EI} \quad (m_o + m_1 x_1 + m_2 x_2)\, ds = 0$$

$$\frac{\partial U}{\partial x_2} = \int_s \frac{m_2}{EI} (m_o + m_1 x_1 + m_2 x_2)\, ds = 0 \quad \ldots (2.15)$$

Expanding, equations (2.15) become:-

$$x_1 \int_s \frac{m_1^2}{EI} ds + x_2 \int_s \frac{m_1 m_2}{EI} ds + \int_s \frac{m_1 m_o}{EI} ds = 0$$

$$x_1 \int_s \frac{m_1 m_2}{EI} ds + x_2 \int_s \frac{m_2^2}{EI} ds + \int_s \frac{m_2 m_o}{EI} ds = 0 \quad \ldots (2.16)$$

Consider now Castigliano's second theorem, viz:-

$$\frac{\partial U}{\partial x} = \delta$$

and apply it to the displacement δ_1 at the position and in the direction of x_1:-

$$\delta_1 = \frac{\partial U}{\partial x_1} = x_1 \int_s \frac{m_1^2}{EI} ds + x_2 \int_s \frac{m_1 m_2}{EI} ds + \int_s \frac{m_1 m_o}{EI} ds \quad \ldots (2.17)$$

If the applied load $m_o = 0$ and $x_2 = 0$ then:-

$$\delta_1 = x_1 \int_s \frac{m_1^2}{EI} ds = x_1 f_{11} \text{ (say)} \quad \ldots (2.18)$$

Thus the rotation of the released structure under the action of a unit bending moment x_1 only, is:-

$$\theta_1 = \int_s \frac{m_1^2}{EI} ds \quad \ldots (2.19)$$

If the applied load $m_o = 0$ and $x_1 = 0$ then:-

$$\delta_1 = x_2 \int_s^{\rho} \frac{m_1 m_2}{EI} \, ds = x_2 \, f_{12} \text{ (say)} \qquad \ldots (2.20)$$

Thus for a unit value of x_2 applied at end 2 the rotation at end 1 will be:-

$$\alpha_{12} = \int_s^{\rho} \frac{m_1 m_2}{EI} \, ds \qquad \ldots (2.21)$$

If $x_1 = x_2 = 0$ then

$$\delta_1 = \int_s^{\rho} \frac{m_1 m_o}{EI} \, ds = u_1 \text{ (say)} \qquad \ldots (2.22)$$

Thus u_1 is the displacement at the position and in the direction of x_1 due to the applied load.

The first equation (2.16) can thus be written:-

$$f_{11}x_1 + f_{12}x_2 = -u_1 \qquad \ldots (2.23a)$$

Similarly the second equation relating to x_2 can be written:

$$f_{21}x_1 + f_{22}x_2 = -u_2 \qquad \ldots (2.23b)$$

where $f_{21} = \int_s^{\rho} \frac{m_1 m_2}{EI} \, ds = \alpha_{21}$

$$f_{22} = \int_s^{\rho} \frac{m_2^{\,2}}{EI} \, ds = \theta_2$$

Equations (2.23) can be written in matrix form thus:-

$$\begin{bmatrix} f_{11} + f_{12} \\ f_{21} \quad f_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \ldots (2.23c)$$

The solution these two simultaneous equations gives the values of arbitrary constants x_1 and x_2 and because $m_o = 0$ at the supports these will be the actual values of fixed end moments at end 1 and 2 respectively.

The stiffness and carry over factors can also be derived from the factors already calculated. Consider the member 12 simply supported at end 1 and fully fixed at end 2 as shown in Fig. (2.7a). The application of a unit bending moment at end 1 will cause a fixed end moment to act at end 2. The value of moment required to ensure full fixity will be $\frac{\alpha_{21}}{\theta_2}$ as given by Eqns, (2.19) and (2.21). This ratio is the carry over from end 1 to end 2, thus:-

$$C_{12} = - \frac{\int_s \frac{m_1 m_2}{EI}}{\int_s \frac{m_2^2}{EI} ds} = \frac{f_{12}}{f_{22}} \qquad \ldots (2.24)$$

and similarly $C_{21} = - \frac{\int_s \frac{m_1 m_2}{EI}}{\int_s \frac{m_1^2}{EI}} = \frac{f_{12}}{f_{11}}$ ... (2.25)

The effect of the moment being applied at end 2 will cause a rotation at end 1 which will equal $\frac{\alpha_{21}}{\theta_2} \times \frac{\int_s \frac{m_1 m_2}{EI} ds}{1}$

Therefore the total rotation at end 1 due to the application of unit moment M_1 at end 1 with end 2 fully fixed, is:-

$$\delta_1 = \int_s \frac{m_1^2}{EI} ds - \frac{\left[\int_s \frac{m_1 m_2}{EI} ds\right]^2}{\int_s \frac{m_2^2}{EI} ds}$$

thus $\delta_1 = f_{11} - \frac{f_{12}^2}{f_{22}}$ ... (2.26)

Similarly for δ_2 in Fig. (2.7b):-

$$\delta_2 = f_{22} - \frac{f_{12}^2}{f_{11}}$$

The bending stiffness of a member is the moment required to produce a unit rotation at the applied end with the remote end fully fixed.

Hence stiffness at end 1 is:-

$$K_{12} = \frac{1}{\delta_1} = \frac{1}{f_{11} - f_{12}^2/f_{22}} \quad \text{... (2.27)}$$

Stiffness at end 2 is:-

$$K_{21} = \frac{1}{\delta_2} = \frac{1}{f_{22} - f_{12}^2 / f_{11}} \qquad \ldots (2.28)$$

In order to calculate the stiffness, carry over factors and fixed end moments it is necessary to integrate the various bending moment diagrams. This is most easily accomplished by dividing the span of the beam into an even number of equal parts and by applying Simpson's rule to the ordinates of bending moment and $\frac{1}{EI}$ values. The accuracy of this method is discussed in Section (2.7).

2.5) Organisation of the Programs.

2.5.1) Standard Versions.

The basic operations can be summarised in the form of a flow diagram and are shown in Fig. (2.8).

The data is prepared in the standard way, as described in Section (2.6), and read into the computer as required by the program. It is not necessary to store the whole of the data within the computer before calculations begin.

Quantities are stored within the computer in one of two ways. 'Integers' are stored as fixed point numbers having an exact value. Any arithmetic operations are carried out exactly and the result is not subject to rounding off. Other variables which are 'real' numbers are stored in floating point form and are subject to rounding off errors. Values may be stored as single numbers, referred to by 'identifiers', or in blocks. The Algol language has the facilities to store blocks of numbers that can be referred to by subscripted variables. These may hold vectors and matrices or sets of numbers and are called 'arrays'. Arrays may be either of type 'integer' or'real'.

Upon commencement the computer reads in the basic parameters of the frame, consisting of: 'm' the number of members,'n' the number of joints, 'x' the number of types of members. Using these integers storage space can be alloted to contain the member list and member properties. The following eight properties are required for each member: $\cos\alpha$, $\sin\alpha$, L, T, C_{12}, C_{21}, K_{12}, K_{21}. In the case of plane frameworks the coefficient T is replaced by the axial stiffness A. These properties are also later required to set up the individual member slope deflection equations to calculate the final forces.and are stored in array P. To economise on storage space the members are grouped into sets in which all the members have the same properties. It is then only necessary to store one set of properties for each type of member. The size of array P is therefore $8\times x$. The members are denoted by the joints they connect and a list of members is stored in array M, which has $3 \times m$ elements of store. Because these joint numbers are later required to locate the positions of the elements in the stiffness matrix, they must not be rounded off and array M is therefore of type integer. The columns of M hold: joint at end 1, joint at end 2, type number of member.

The values of E and G are input and remain constant throughout the program. If any members in the frame have a different E or G value this can be allowed for by proportionately altering the I and J values for that member. The data for the first member is read in, the joint numbers being placed in array M and the x and y ordinates being placed in the first two columns in the P array. The number of members of this type, 'd' is also noted. The length L in inches is then calculated, being $L = 12 \times \sqrt{x^2+y^2}$, and placed in the third column. Using L the x and y ordinates are then converted to $\cos\alpha$ and $\sin\alpha$ respectively.

As mentioned previously the stiffness and carry over factors are calculated from the influence coefficients of the beam. Although at this stage the fixed end moments are not calculated the $\frac{1}{EI}$ values along the beam will later be required and are stored in array F. Array F is a vector occupying only one row. As before only the $\frac{1}{EI}$ values for each type of member are stored.

The computer next reads in the number of stations at which I and J are specified. This quantity is denoted by 't' and is stored in array T. The next 't' numbers are now read into array HF. The m_1 and m_2 values are formed in array H and the Simpson's rule factors in array Q. The influence coefficients f_{11}, f_{12}, f_{22} are then calculated by using Jensen's device which is a 'procedure' for finding the innerproduct of two vectors. A procedure is a sub-routine within the program that can be entered at any time by writing the name of the procedure followed by its parameters. The innerproduct procedure is given the name 'dot' and is included at the beginning of the program thus:-

```
real procedure  dot (a,b,p,q,r);
    value  p,q; real  a,b;
    integer  p,q,r;
    begin real s; s:=o;
    for r:=p step 1 until q do s:=s+axb;
    dot: = s;
    end;
```

The stiffness and carry over factors can now be calculated using Equations (2.24) to (2.28) and are stored in the last four places in the row of the P array.

The computer then reads the next 't' numbers from the data tape, being the corresponding J values at each 'station'.

The torsional stiffness is then calculated by summing the values along the length using Simpson's coefficients. For a plane frame the axial stiffness A is evaluated in the same way using the area at each 'station'.

The remaining members of this type are now input. As the properties for these members have just been calculated it is not necessary to specify any further data other than the joint numbers at end 1 and end 2. Hence in introducing economies of storage space, considerable time is also saved in data preparation.

The next d-1 pairs of numbers are then read into array M and are alloted a type number. Integer 'd' being the number of members of this type.

The process is continued until all the member properties have been calculated and stored in the property array. If any member in the frame is of uniform section the properties for that member are inserted directly into the appropriate storage positions. In this case only one value of I and J (or A) are required to describe the member and $C_{12} = C_{21} = 0.5.$, $K_{12} = K_{21} = \frac{4EI}{L}$

$T = \frac{GJ}{L}$ or $A = \frac{Ea}{L}$

The next section of the program sets up the stiffness matrix for the structure. The contributions from each member are inserted in turn, the members being considered in the order they appear in the M array. The computer selects the next member ij, in the M array and locates its corresponding properties in the P array. The diagonal terms are first calculated and placed in the stiffness matrix. K_{12}' and K_{21}' being placed in the locations corresponding to the ith and jth joints respectively. The computer then checks to ascertain whether the joint at end 1 is smaller than the joint at end 2. If this is so the off diagonal term R_{12}' is placed in the corresponding joint locations on the ith row and under the jth column. Alternatively, if the joint at end 1 is greater than the joint at end 2 then the R_{21}' term is used, being placed in the same position. In this case the member is in effect reversed.

This process is repeated for every member until the stiffness matrix for the complete structure has been formed.

A general joint forming part of a framework is shown in Figure (2.9) and the corresponding stiffness terms for members 13, 35 and 38 are shown inserted into the stiffness matrix.

In examining Tables (2.2) and (2.4) it will be noticed that matrix R_{21}' is the transpose of matrix R_{12}' and that matrices K_{12}' and K_{21}' are symmetrical about their diagonals.

Thus when the stiffness matrix is formed for the complete structure it is symmetrical about the leading diagonal.

If the form of the stiffness matrix is further examined it will be noticed that the physical form of the structure is reflected in the arrangement of the elements. If there is a member connecting joints i and j, the K_{12}' and K_{21}' terms will lie on the diagonal corresponding to the ith and jth joints respectively, and also the R_{12}' term will lie off the diagonal in the jth column, indicating a member connecting joints i and j. The form of the stiffness matrix is shown in Figure (2.10) where an asterisk represents a 3 x 3 matrix element. As the matrix is symmetrical about the leading diagonal, only the terms above and including the diagonal have been indicated. It can be seen that the elements form a band along the diagonal and the width of the band is dependent upon the width of the structure, being determined by the greatest difference between joint numbers. All terms lying outside the band are zero.

Utilising these facts, considerable savings can be made in the amount of space required to store the stiffness matrix. Because of symmetry only half the matrix need be stored. This can be further reduced, however, by only storing the elements that constitute the top half of the band. If a framework containing 25 x 3 joints is taken as an example. The storage capacity for the complete matrix would be $(3 \times 75)^2 = 50625$, there being three displacements at each joint. Storing only the upper half of the matrix band the space require will be $75 \times 4 \times 3^2 = 2700$.

Comparing this with the total number of 50625 elements it can be seen that, in this particular case, only about 5% of the original space is required to define the stiffness matrix.

When alloting the numbers to joints in a frame it is therefore advantageous to arrive at the system that will give the narrowest band. In the case of a grillage this is achieved by numbering across the narrowest width of the frame as demonstrated in Fig. (2.11).

As the computer is only able to store rectangular arrays the matrix band must be stored vertically as an array of size 3n x w, where 'n' is the number of joints and 'w' is the width of the matrix band.

The width 'w' is next calculated by the computer. Each member is considered in turn and the greatest difference between the joint numbers is found using the Algorithm:-

```
for i: = 1 step 1 until m. do
if  abs (M[i,1]- M[i,2])>w/3 - 1 then
w: = (abs (M[i,1]- M[i,2])+1)x3;
```

Having calculated 'w', an array S is declared to accommodate the stiffness matrix band containing 3n x w elements. All elements are given an initial value of zero. The computer then inserts the stiffness matrix terms for each member to build up the stiffness matrix for the complete structure. In placing the elements in array S due care is taken of the way the matrix is orientated. The form of the stiffness matrix and method of storage within the computer are shown in Fig. (2.12), where an asterisk represents one element in a K' or R' matrix.

The next operation is to form the loading vector. The program is able to analyse more than one loading case and so the loading vectors are stored in an array B. If 'y' is the number of loading cases, array B can be declared requiring 3nxy storage units. All spaces are initially given a value zero.

The terms in the loading array are formed by setting up equations (2.23) for each loaded beam, thereby finding the values of fixed end moments. The 'free' reactions are calculated and then corrected using the fixed end moments to restore statical equilibrium. The influence coefficients are reformed, as described previously, the $\frac{1}{EI}$ values for each beam having been stored for each type of member. To form the u_1 and u_2 values the free bending moment diagram ordinates are required at each 'station'. The computer calculates these values automatically, first considering the UDL, if any, acting on the beam:-

$$Ord_m = \frac{WL^2}{2}\,\frac{(s-1)(1-(s-1)/(n-1))}{(n-1)}$$

where W = load per foot

s = no. of stations

n = total number of stations

The contributions from point loads are now added being for ordinates to the left of the load:-

$$\text{Ord}_m = \frac{P\ (L-l)\ (n-1)}{(t-1)}$$

where l = distance from L.H. end.

and for ordinates to the right of the point load

$$\text{Ord}_m = \frac{Pl(t-n)}{(t-1)}$$

The total free bending moment ordinates at each 'station' are held in array 'N.

In this way the total free bending moment ordinates are built up by taking contributions from each load on the member. Only the ordinates at stations are considered therefore if a point load lies between two stations the bending moment peak will not be considered. This leads to slight local inaccuracies, which are discussed further in section (2.5).

Using procedure 'dot' matrices H'F and 'N are multiplied together giving the u_1 and u_2 values. These values are calculated for every loaded member and for each loading case, and are printed out in tabular form. This enables the local effects of bending moment and shear to be superimposed on to the general results output by the computer.

There now exists a set of simultaneous equations with the stiffness matrix forming the left hand side and the loading vectors on the right hand side. Before a solution is possible the conditions of zero deformations have to be incorporated, otherwise the structure is capable of rigid body movement and a unique solution is not possible. If a particular deformation is zero, the equation governing the deformation can be removed, together with all the terms that are multiplied by the deformation. The corresponding loads are also made zero. Thus the number of equations remains equal to the number of unknowns. To allow continuity in the process of solution it is necessary for the diagonal pivot term to remain in the stiffness matrix. Since there is now only one term on the left hand side being equated to a zero load the resulting deformation will become zero.

The joints having zero displacements are read from the data tape and the corresponding terms in the stiffness and loading arrays set to zero. If at joint 3 in a frame the vertical displacement is zero then the third term corresponding to joint 3 would be modified as shown in Fig. (2.12).

For the solution of the equations the 'Square-root' method is employed being a modification of the Choleski method (23).

As matrix S is symmetric it can be resolved into two triangular matrices one of which is the transpose of the other:

i.e. $S = U'U$

where U is an upper triangular matrix.

The elements of U can be evaluated by considering the rules for multiplying two matrices:

$$S_{ij} = U_{1i}U_{1j} + U_{2i}U_{2j} + \dots + U_{ii}U_{jj}, \quad (i>j)$$

$$S_{ii} = U_{1i}^2 + U_{2i}^2 + U_{3i}^2 + \dots + U_{ii}^2, \quad (i=j)$$

Hence the elements U_{ij} can be determined thus:-

$$U_{ii} = \sqrt{S_{ii} - \sum_{t=1}^{i-1} U_{ti}^2}, \quad (i>o) \quad \dots (2.29)$$

$$U_{ij} = \frac{S_{ij} - \sum_{t=1}^{i-1} U_{ti}U_{tj}}{U_{ii}}, \quad (j>i) \quad \dots (2.30)$$

Solving the system is the same as solving two triangular systems, thus:- $SD = F$

can be written:-

$U'K = F$ and $UD = K$

The elements in the K vector are determined by recurrent formulas analogous to the formulae for U_{ij}. Namely:-

$$K_1 = \frac{F_1}{U_{11}}$$

$$K_i = \frac{F_i - \sum_{t=1}^{i-1} U_{ti}K_t,}{U_{ii}} \qquad (i>1) \qquad \ldots (2.31)$$

The final solution is formed by the formulae:-

$$D_n = \frac{K_n}{U_{nn}}$$

$$D_i = \frac{K_i - \sum_{t=i+1}^{n} U_{it}D_t}{U_{i1}} \qquad \ldots (2.32)$$

The actual programming of the solution takes only a few lines and illustrates the power of the Algol programming language. Using the procedure 'dot' the upper triangular matrix is formed thus:-

```
for i: = 1 step 1 until 3xn do
 begin S[i,1]: = sqrt (S[i,1]-dot (S[r,i-r+1]↑2, 1,
      if i>w then i-w+1 else 1, i-1, r));
      k:= if 3xn-i>w-1 then w else 3xn-i+1;
      for j:=2 step 1 until k do
      S[i,j]:= (S[i,j]- dot (S[r,i-r+1], S[r,j+i-r], if
      j+i> w+1 then j+i-w else 1, i-1, r ))/S[i,1];
end formation of upper triangular matrix;
```

The loading vectors B are now transformed in the same way:-

```
for j: = 1 step 1 until y do
for i: = 1 step 1 until 3xn do
B[i,j]: = (B[i,j]- dot (S[r,i-r+1], B[r,j], if i>w
then i-w+1 else 1,i-1,r))/S[i,1];
```

Back substitution now leads to the final solution of the set of equations. As the deformations are evaluated they are placed into the positions originally occupied by the loads. Therefore array B eventually holds the values of deformation at each joint and for each loading case.

```
for j: = 1 step 1 until y do
begin B[3xn,j]:= B[3xn,j]/S[3xn,1];
for i:=3xn-1 step-1 until 1 do
B[i,j]:=(B[i,j]-dot (S[i,r-i+1], B[r,j],
i+1, if 3xn-i>w-1 then i+w-1 else 3xn,r))/S[i,1];
end back substitution;
```

It should be noted that the extra amount of calculation involved when the number of loading cases is increased is quite small. The main body of calculation is involved in forming the upper triangular matrix U. Once this has been evaluated any number of loading cases can be dealt with as alternative right hand sides. In running the program the additional time for calculation of additional loading cases and the corresponding increase in cost, is quite small. A large structure with many loading cases will have considerable final output and for an economical solution these results should be first placed onto magnetic tape and later processed through an off line printer.

The deformations are then back substituted into the slope deflection equations in Table (2.2) to give the forces in the x and y directions. These are then resolved into the member directions giving the forces viz. bending moments at each end, torsion moments, (or axial load for plane frameworks) and the shear force, for each member in the frame.

For each loading case the computer prints out in tabular form the forces preceeded by the member location. The values of deformation are also output. Although only values of deflection are normally of practical use the values of rotation are included for completeness. An example is given in Section (2.9).

2.5.2) Partitioned Grid Framework Program.

Computers are currently being manufactured with large fast store capacities and it is likely that in the future larger machines will be built thus enabling frameworks of several hundred joints to be analysed. The installation at the Leeds Computing Laboratory has recently been increased to contain 32K 'words' of fast store which enables grid frameworks of up to 450 joints to be solved. The core store will, however, always remain finite, thus placing a theoretical restriction on the size of problem that can be solved as a complete frame. By using the technique of partitioning any size of frame can be analysed, provided it can be reduced to a series of linearly connected sub-frames. A partitioned version of the grid framework program has therefore been written.

The basic method of solution is similar to that employed for the standard version. The difference occurs in the method of storing the various matrices. Data is prepared in the way described in Section (2.6.2). The basic operations are summarised in the flow diagram shown in Fig. (2.13).

In forming the upper triangular matrix and back substituting to obtain final deflections, only the terms on the preceeding 'w' lines are required - where 'w' is the width of the matrix band. Thus once the first 'w' lines of the upper triangular matrix have been formed, to form line w+1, line one is not required and could be transferred from the fast store on to magnetic storage tape. As it is more efficient to transfer several lines at any one time the lines are moved in sub-frame blocks. To form the stiffness matrix it is necessary to have access to the previous sub frame in order to include the terms from members common to both sub-frames. It is also necessary to impose zero displacements twice for each sub-frame as an initial zero may have been overwritten when terms from the common members have been added. Hence two adjacent sub-frames have to be stored in the fast store whilst the stiffness matrix is being formed. It is this fact that determines the size of individual sub-frames. The structure has to be sub-divided such that no two adjacent sub-frames exceed the available fast store.

Once a structure has been partitioned there is a marked increase in the time required for solution as the use of magnetic tape is a comparatively slow method of storing information. A recent set of data for a frame of 170 joints having 5 loading cases, took 25 minutes for solution. When using this program on a time sharing machine, only the effective time used is normally taken into account when charging for computer time used. Thus although the elapsed time may be higher, the time in terms of cost will only be slightly more than quoted in Section (2.8).

2.6) Preparation of Data

2.6.1) Standard Programs.

Each data tape is headed by a 'call' sequence containing the program's unique identifier. This causes the computer to select the correct program and to transfer it into the computer store. Following the 'call' sequence is a title describing the particular problem, which is transferred directly to the head of any output.

The basic frame parameters follow and consist of:-

Total number of joints in the frame.

Total number of members in the frame.

Number of types of mamber

Number of previous loading cases analysed.

Modulus of elasticity, E.

Torsional constant, G. (for grid frames only).

The number of previous loading cases is included to allow sets of output for the same frame to run consecutively. Thus if five loading cases have already been analysed the output will commence at Loading Case No. 6.

The member properties are next specified. They are first sorted into types in which all properties are identical and these properties given for the first member of each type. They consist of:- joint number at end 1, joint number at end 2, number of members of the type, x and y ordinates in feet, number of stations at which sectional properties are given, member properties. The ordinates are individual member ordinates, that is the relationship of end 2 to end 1. Sectional member properties consist of moment of inertia (ins^4) and either torsional rigidity (ins^4) for grid frameworks or cross sectional area (ins^2) for plane frameworks. As Simpson's rule is used to integrate the various functions, member properties should be specified at an odd number of points along the member. All other members of this type are now given and consist solely of the joint at end 1 and the joint at end 2.

The number of loading cases is stated and each loading case specified. Loads consist of a UDL acting along the full projected length of the number and/or a series of point loads acting at a distance from end 1. Each loading case has the following data:- number of loaded members; for each loaded member- joint at end 1, joint at end 2, UDL per foot of projected length, number of point loads, for each point load - load, distance from end 1 (ft). In the case of plane frameworks loads can act in both x and y directions and the loads on each member are therefore described twice, first in the x direction and second in the y direction.

Next the number of zero displacements in the frame is given. These are specified as the joint number followed by an integer to describe the direction of restraint, as given in Table (2.5)

TABLE (2.5)			
Grid Framework		Plane Framework	
Integer	Direction	Integer	Direction
1	θ_x	1	δ_x
2	θ_y	2	δ_y
3	δ_z	3	θ

In setting out the data, spaces and new lines may be used wherever necessary as these symbols are ignored by the compiler. Each item of data must be followed by a terminator which may consist of any symbol, other than a letter of the alphabet, digit, or decimal point. A semi-colon is generally used for this purpose. The whole of the data is terminated by an 'end message' symbol.

2.6.2) Paritioned Grid Framework.

The structure is sub-divided into frames such that no joint is common to more than two sub-frames. The joint numbers must also run consecutively within each sub-frame.

The data is headed by the 'call' sequence followed by the title. The basic parameters for the frame are stated in the order given below:-

Number of joints in complete structure.

Number of sub-frames.

Number of loading cases.

Number of loading cases previously analysed.

Maximum number of zeros imposed in any one sub-frame.

Matrix band width.

Maximum depth of stiffness matrix required for any two adjacent sub-frames.

For each sub-frame is stated the number of members, number of joints and number of types of members. The sub-frame data is then given in the same way as for the standard grid frame program viz. member data, loading data, impose zeros. All loading cases must be included in every sub-frame. If a particular loading case does not act on a sub-frame a zero is entered as the number of loaded members. Similarly if a sub-frame has no zero displacements the number of imposed zeros is entered as zero. The data is terminated by an end message symbol.

2.7) Accuracy.

Simpson's Rule, used to determine stiffness, carry over factors, fixed end moments and shears, is a central difference formulae which can be expressed in the form:

$$\int_{x_1}^{x_1} = \frac{h}{3}\left(y_1 + 4y_0 + y_{-1}\right)$$ h = distance between ordinates

if fourth and higher powers are neglected.

Any discontinuities should preferably occur at odd ordinates although this is not always possible in the case of point loads. Fig. (2.15) shows that for a part parabolic member, stiffness and carry over factors are calculated to within 1% if nine 'stations' are used. The discrepancy is virtually eliminated by using thirteen 'stations'. It can also be seen that more points are required to specify the loading than stiffness and carry over factors. The choice of ordinates must then depend upon circumstances. As few as three ordinates are required for a member whose moment of inertia varies linearly, and which is unloaded. It has been found in practice that little benefit is gained by selecting more than thirteen ordinates.

2.8) Capacity and Running Time of Programs.

Multispan continuous bridges frequently have large numbers of joints. The design engineer must know if the standard program has sufficient capacity or whether other techniques, such as partitioning need to be employed.

The amount of computer store required for the solution of a framework problem is given by the inequality.

$$4m + 8t + 3j\left[3(g+1) + l\right] + 6250 \leqslant x$$

where m = no. of members.

t = no. of types of members.

j = no. of joints.

g = no. of girders (or 3(g+1) = band width).

l = no. of loading cases.

x = storage capacity of computer.

The quantity 6250 is made up of the size of the program itself and the director which is a control routine program always present in the store.

The majority of the store is used to hold the stiffness matrix for the complete structure and it is this that largely governs the size of framework able to be solved. As shown in Section (2.4) the geometry effects the matrix band width and subsequently the size of the stiffness matrix for any given structure. The maximum number of joints based upon the number of main girders are shown in Fig. (2.14) where storage is measured in K units. $4K = 64^2 = 4096$ 'words' of core store. For example it can be seen that using 16K store a grid framework with eight main girders may have up to 90 joints, whereas a grid frame with four main girders may have up to 130 joints. The capacity of the plane frame program is derived in the same way.

The running time for the grid frame program is similarly dependent upon the total number of joints and the width of the structure. The run time is given approximately thus:

$$t = \frac{nxW}{180}$$

where the time 't' is in minutes, 'n' is the total number of joints, 'W' is the number of main girders across the structure, assuming the joints to be numbered across the frame as shown in Fig. (2.11b). The elapse time will be approximately one minute longer, as this includes the time to read in the data and output results through an on line printer.

2.9) Example.

A 3 bay portal frame as shown in Fig. (2.16) was analysed using the plane frame program. The members are uniform apart from a haunch at the column ends of the rafters. It is quite common to include such haunches for additional stiffness but the effects are often omitted from hand calculations. The increase in stiffness can be quite large and have a noticeable effect on the deflections and distributions of bending moments.

A copy of the data is given in Table (2.6) which follows the system given in Section (2.5). The loading acts only in the y direction therefore the loading to each rafter consists of:

zero U.D.L. in x direction

zero number of point loads in x direction

value of U.D.L. in y direction

zero number of point loads in y direction

The member properties for the rafters must be given at an equidistant number of points. As the length of the haunch is 6'0" the rafter was divided into ten equal sections, each 3'0" feet long and member properties specified at 11 points along each rafter. The columns are uniform in section and therefore require only one value of I and A.

The complete computer output of results is given in Table (2.7); the total time taken was 28 seconds. The fixed end moments and shear given in the first section must be superimposed on to the member forces given in the second section. The final bending moment distribution has been obtained in this manner and is plotted, together with the deflected form in Fig. (2.19). The frame was also analysed neglecting the effects of the haunches. These results are also plotted in Fig. (2.17) being the values in square brackets.

In general it can be seen that the haunches have the effect of 'attracting' more bending moment towards the columns. The values of deflections are reduced by up to 30% and in one case at joint 9 the inclusion of the haunches causes an upward deflection.

Thus it can be seen that although the additional material is relatively small, quite a large reduction in deflections is obtained. Without using the plane frame program for varying section members it would not be possible to benefit from this redistribution of bending moments and reduction in deflection values.

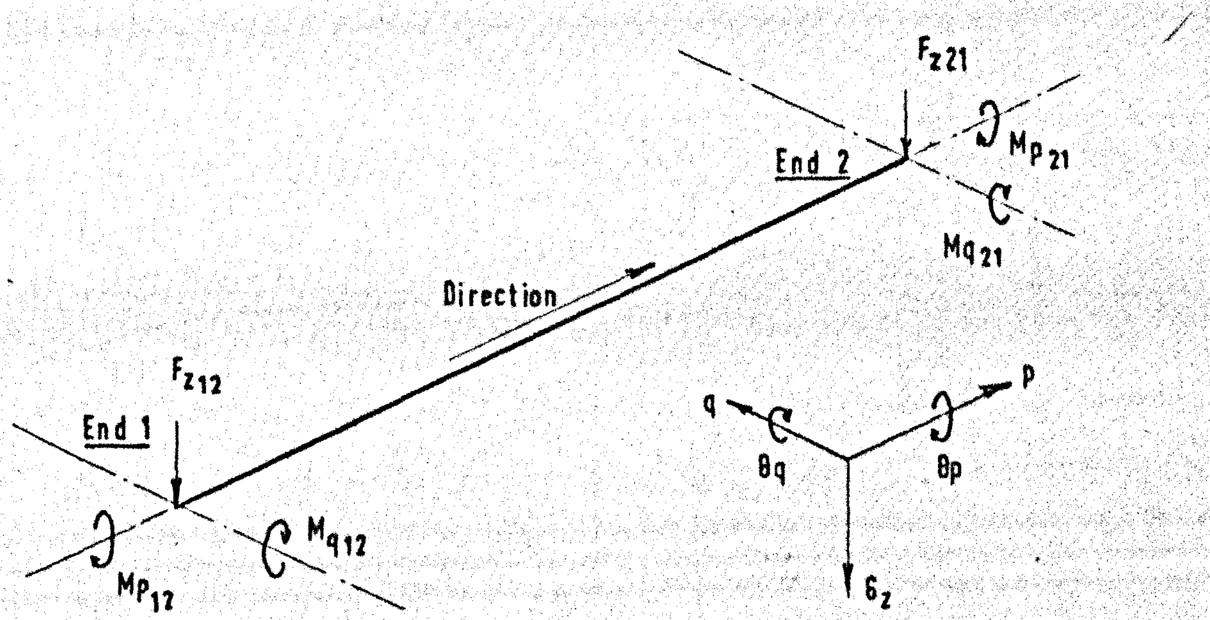


FIG. 2.1 Forces and Displacements for Grid Framework Member Related to Member Axes.

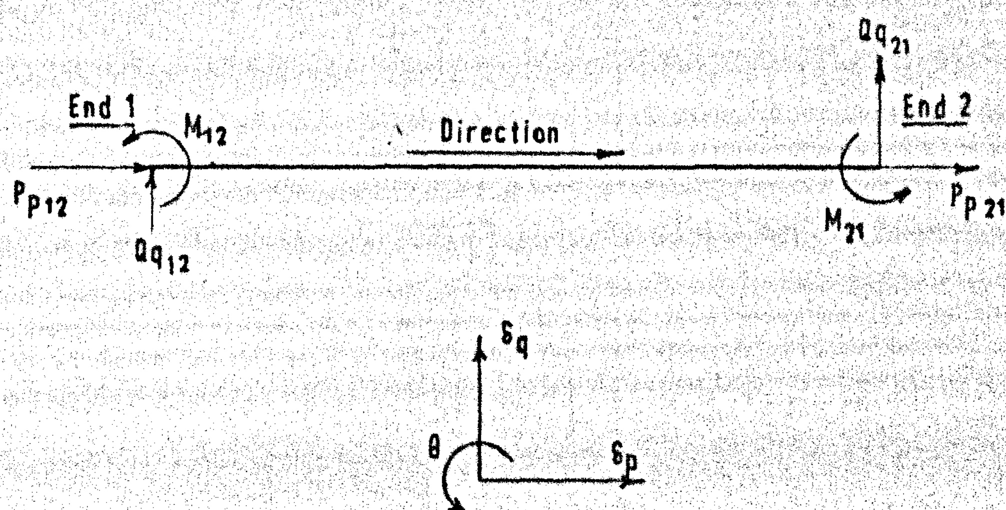


FIG. 2.3 Forces and Displacements for Plane Framework Member Related to Member Axes.

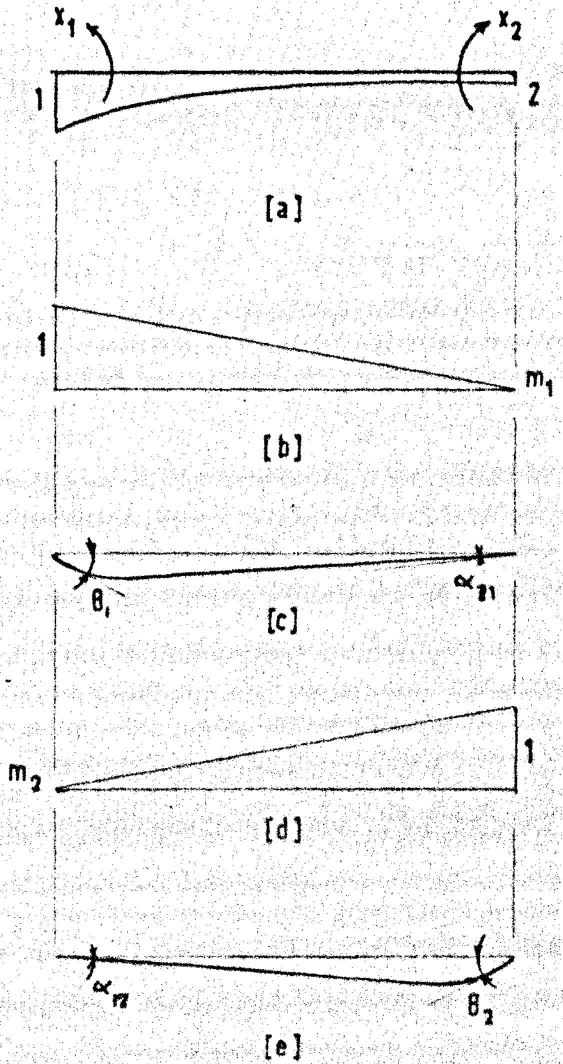


Fig. 2.6

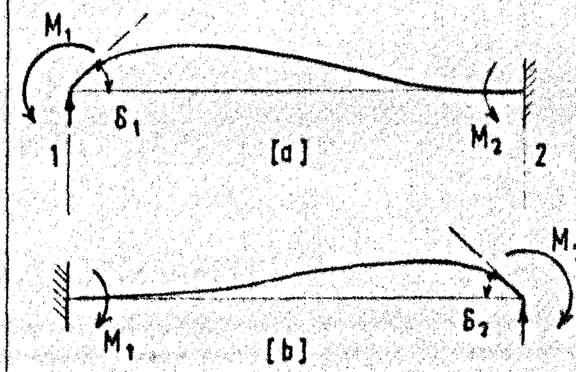


Fig. 2.7

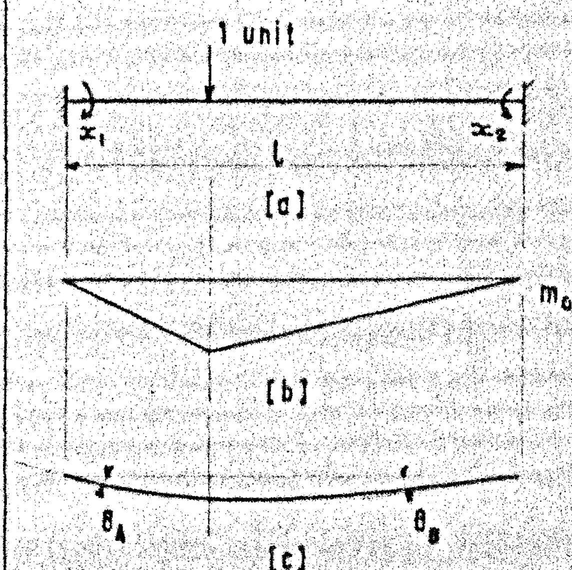


Fig. 2.5

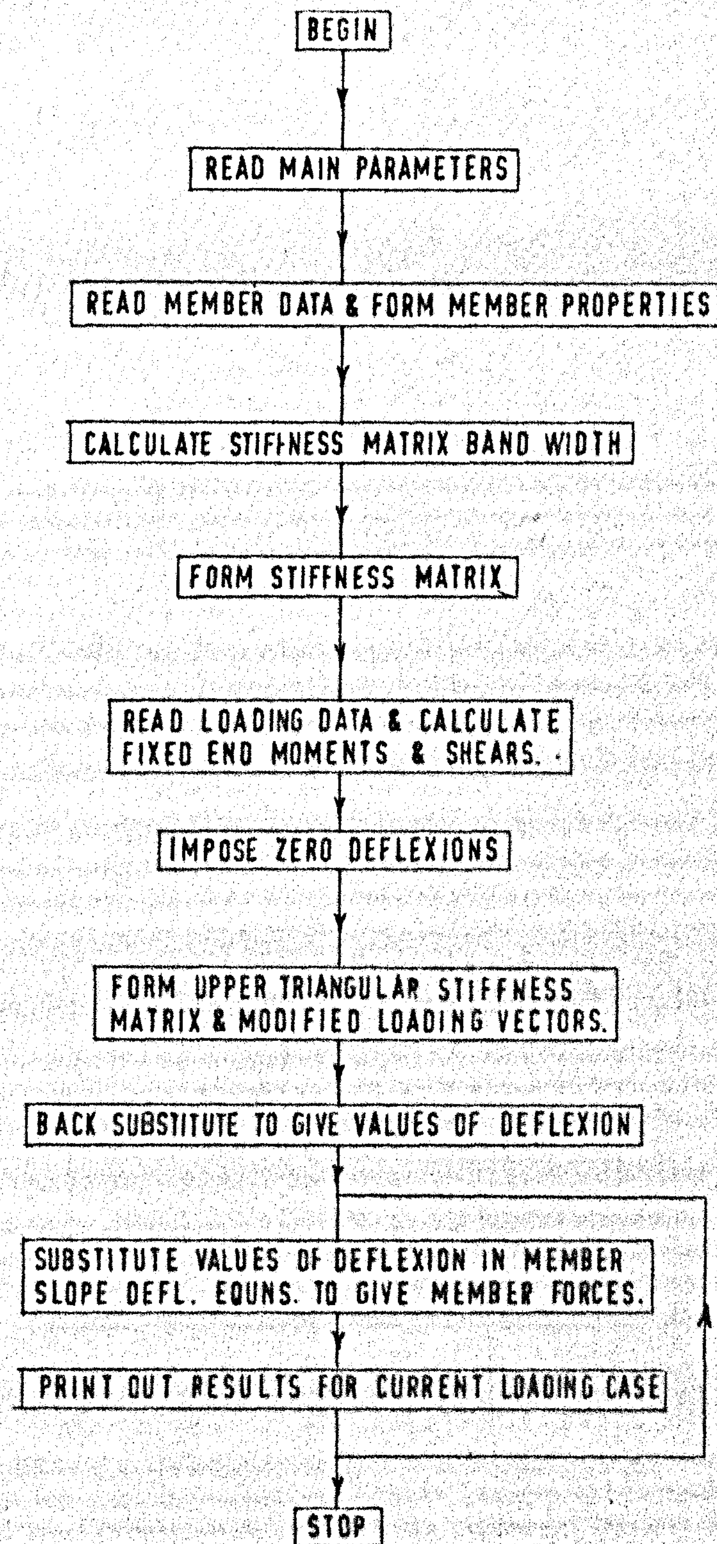
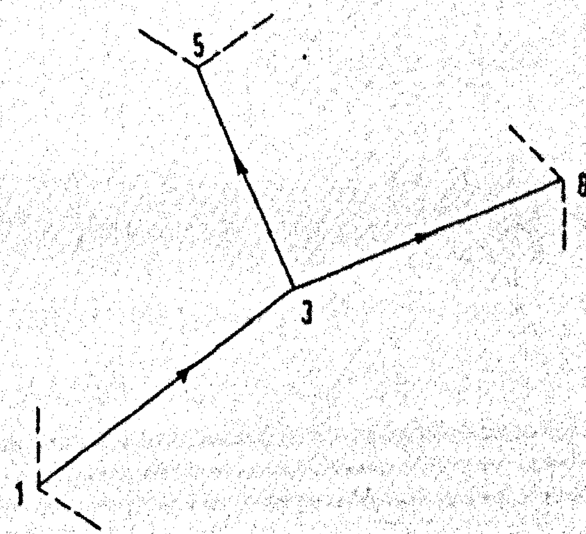


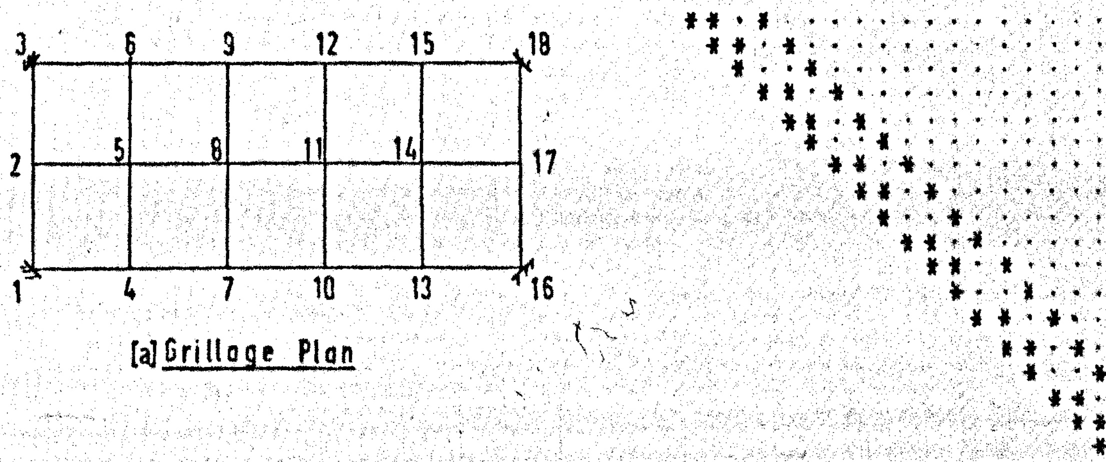
FIG. 2.8 FLOW DIAGRAM



Joint

	1	2	3	4	5	6	7	8		n
1	S_{13}		R_{13}							
2										
3	R_{31}		S_{31}, S_{35} S_{38}		R_{35}			R_{38}		
4										
5			R_{53}		S_{53}					
6										
7										
8			R_{83}					S_{83}		
n										

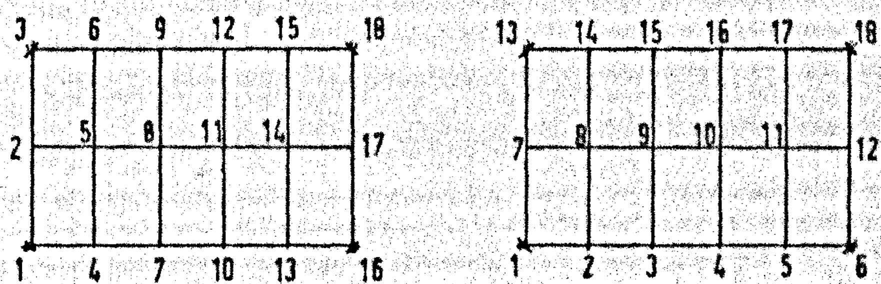
FIG. 2.9 Member Contributions to Stiffness Matrix.



[a] Grillage Plan

[b] Stiffness Matrix Terms

FIGURE 2.10



[a] Band Width = 12.

[b] Band Width = 21.

FIGURE 2.11

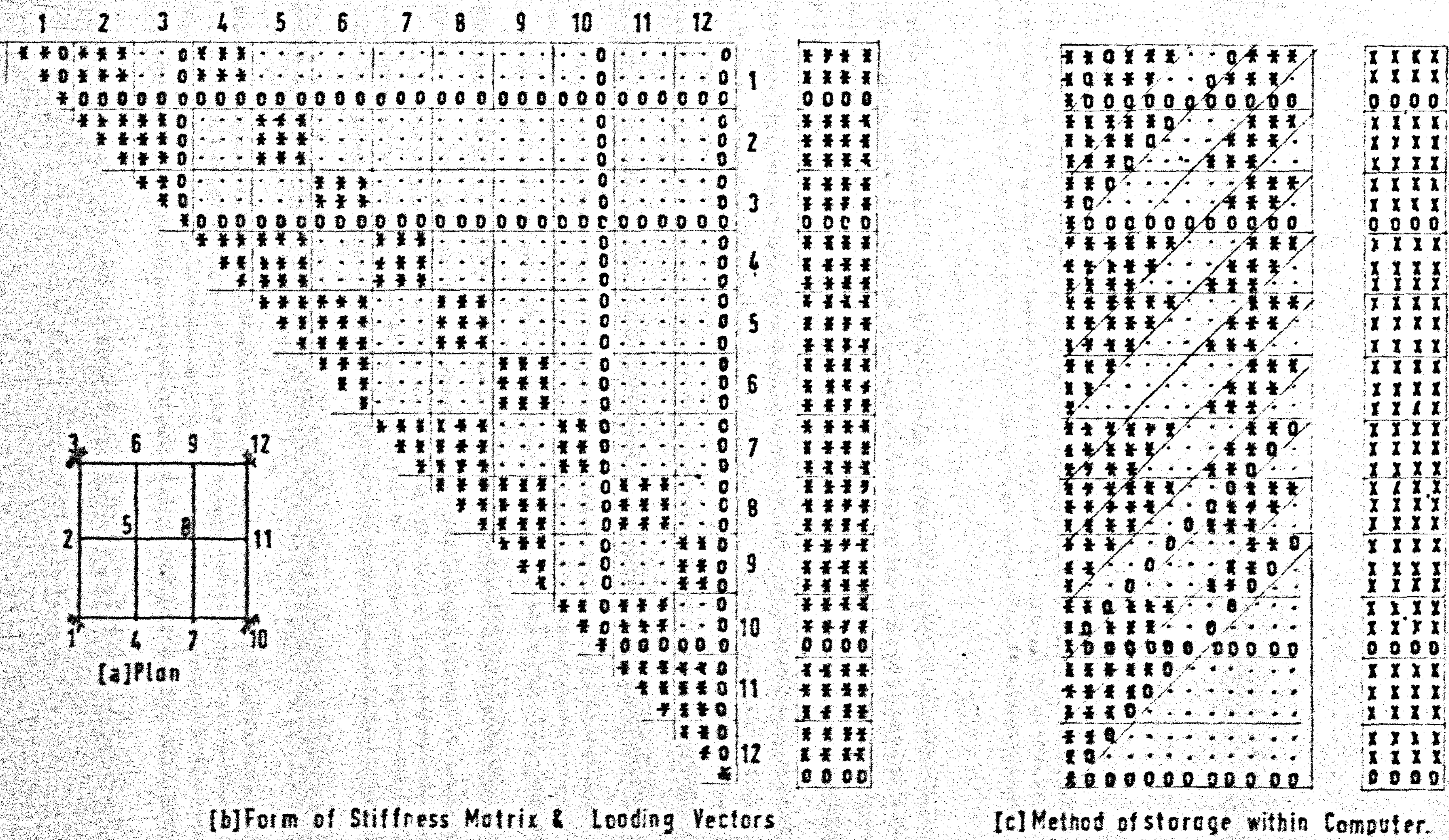

1 2 3 4 5 6 7 8 9 10 11 12
3 6 9 12
2 5 8 11
1 4 7 10
[a]Plan
[b]Form of Stiffness Matrix & Loading Vectors
[c]Method of storage within Computer.

FIGURE 2.12

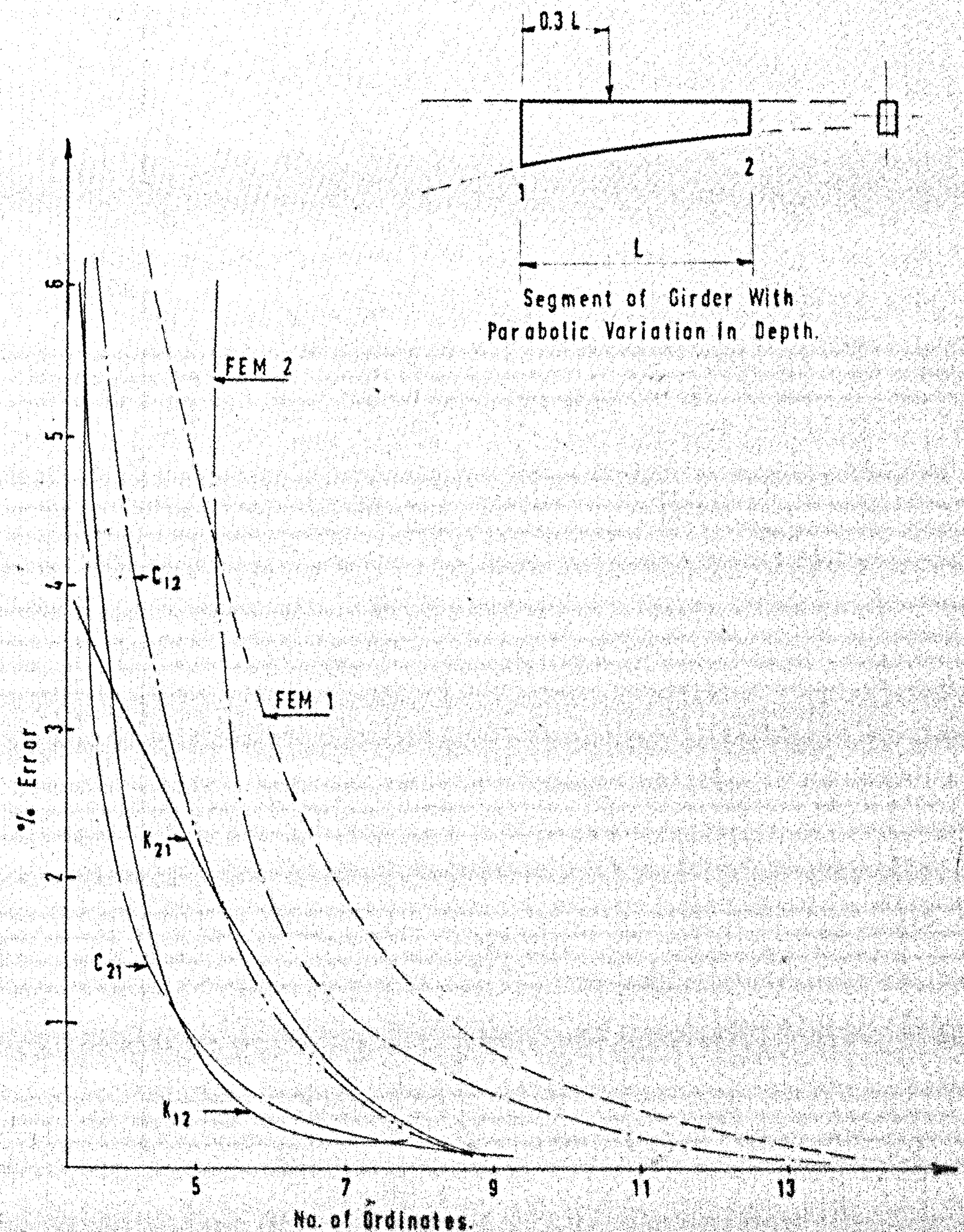


Fig. 2.13 Percentage Error x Nº of Ordinates.

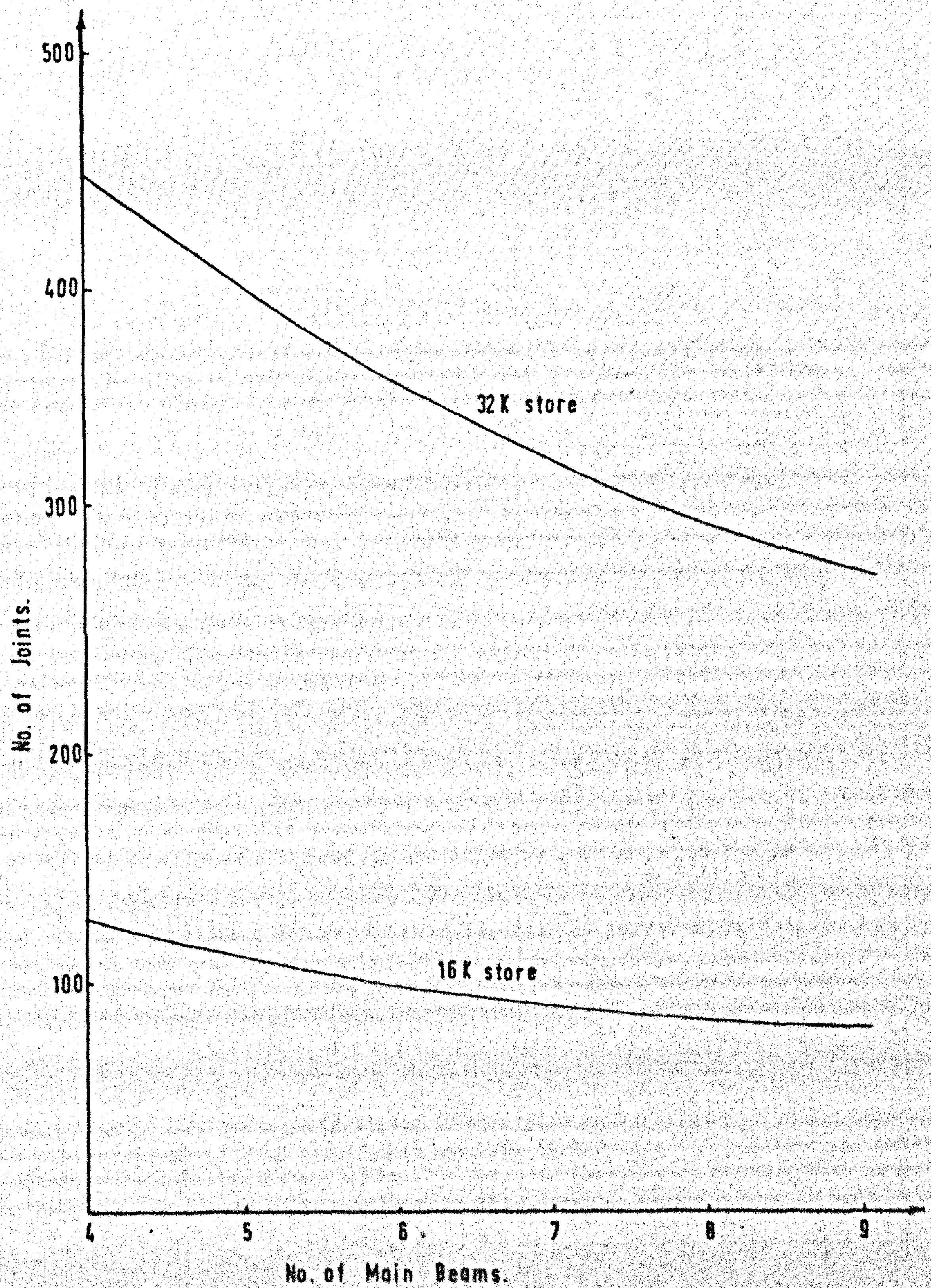


Fig. 2.14 Approx. Capacity of Grid Framework Program.

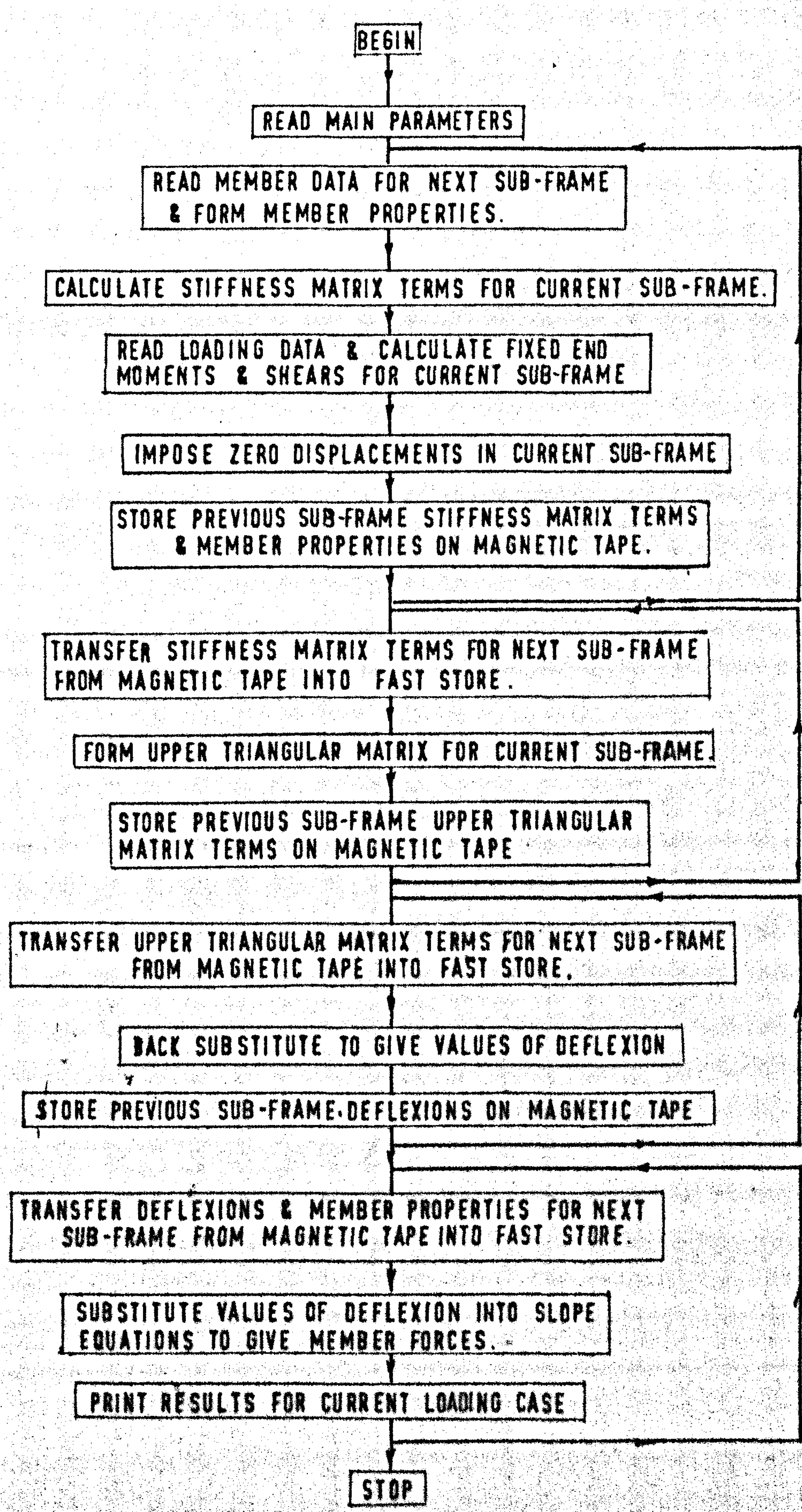
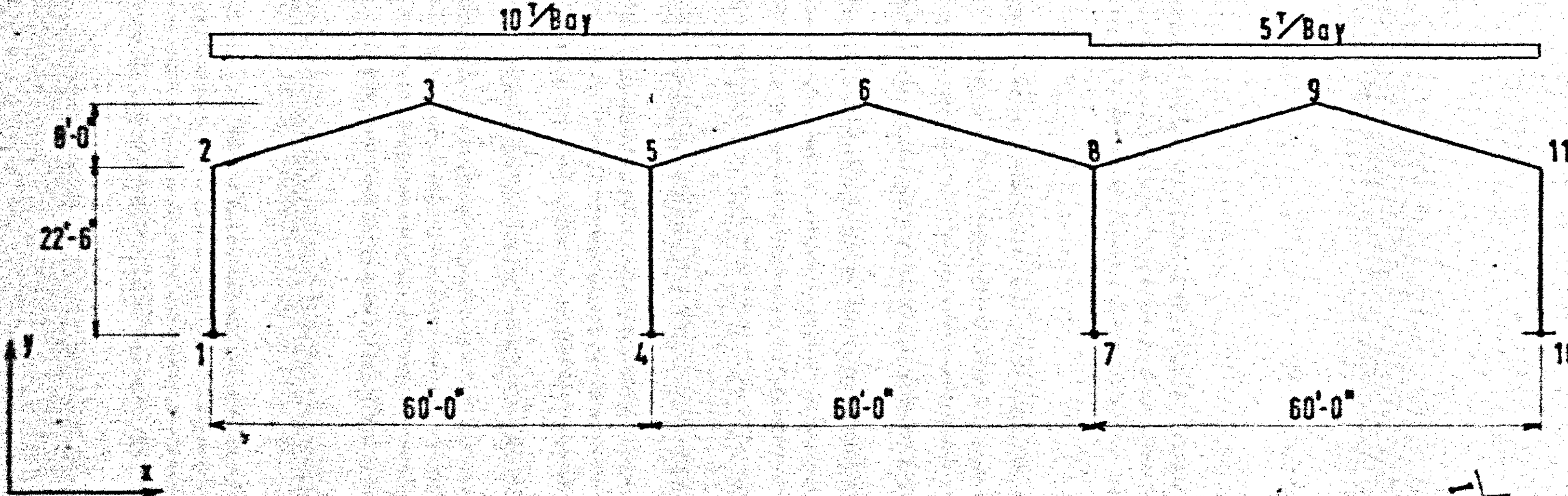
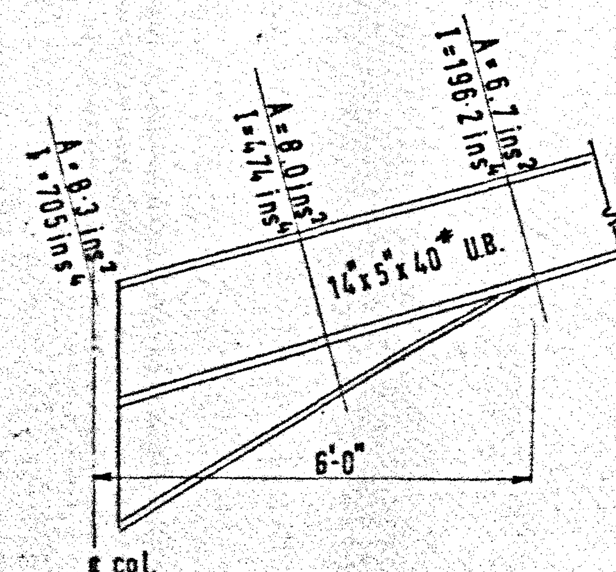


Fig. 2.15 Flow Diagram for Partitioned Grid Frame Program.

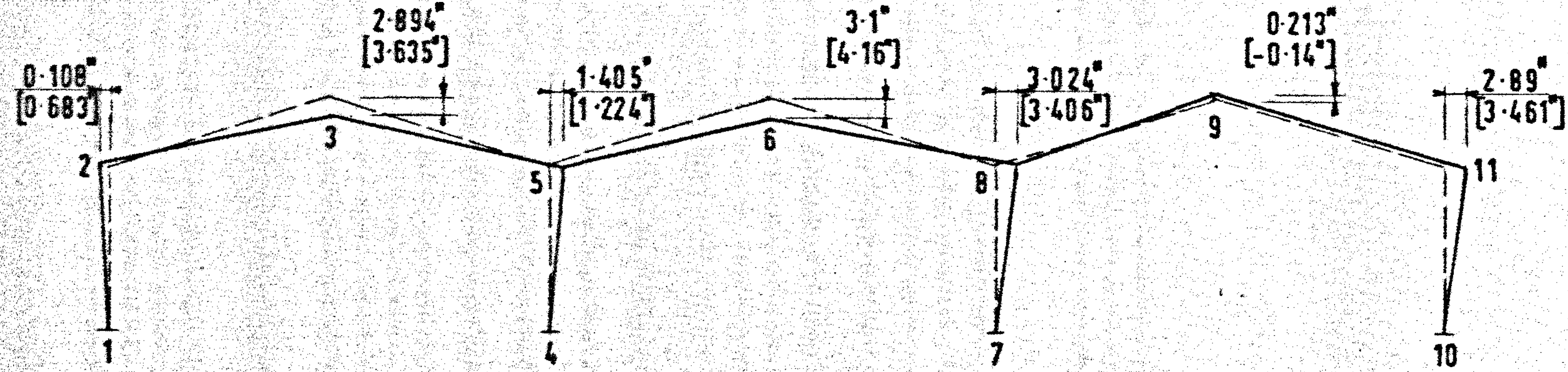


Member	Section	I ins^4	A ins^2
Outer Columns	15"x 6"x 35# U.B.	385.5	10.29
Inner —"—	6"x 6"x 20# U.C.	41.9	5.93
Rafters	14"x 5"x 40# U.B.	196.2	6.7

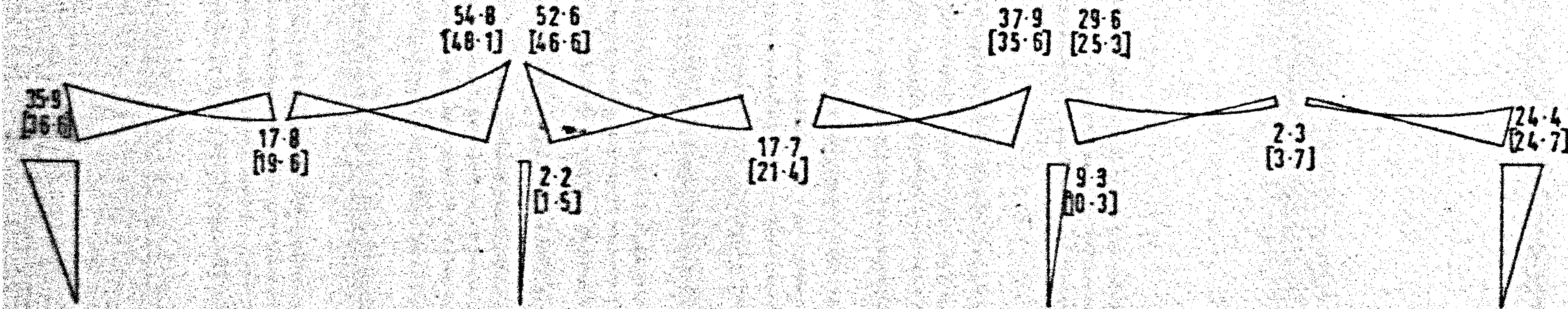


Detail of Haunch.

Fig. 2.15 Details of 3 Bay Portal Frame.



DEFLECTED FORM.



BENDING MOMENT DISTRIBUTION Ton.Ft.

Fig. 2.17 Computer Results. Values in Brackets thus [ ] Neglecting Haunches.

$$
\begin{bmatrix} Mp_1 \\ Mq_1 \\ Fz_1 \end{bmatrix} = \begin{bmatrix} T & 0 & 0 \\ 0 & K_{12} & \dfrac{K_{12}+C_{21}K_{21}}{L} \\ 0 & \dfrac{K_{12}\left[1+C_{12}\right]}{L} & \dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^3} \end{bmatrix} \begin{bmatrix} \theta p_1 \\ \theta q_1 \\ \delta z_1 \end{bmatrix} + \begin{bmatrix} -T & 0 & 0 \\ 0 & C_{21}K_{21} & -\dfrac{\left[K_{12}+C_{21}K_{21}\right]}{L} \\ 0 & \dfrac{K_{21}\left[1+C_{21}\right]}{L} & -\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2} \end{bmatrix} \begin{bmatrix} \theta p_2 \\ \theta q_2 \\ \delta z_2 \end{bmatrix}
$$

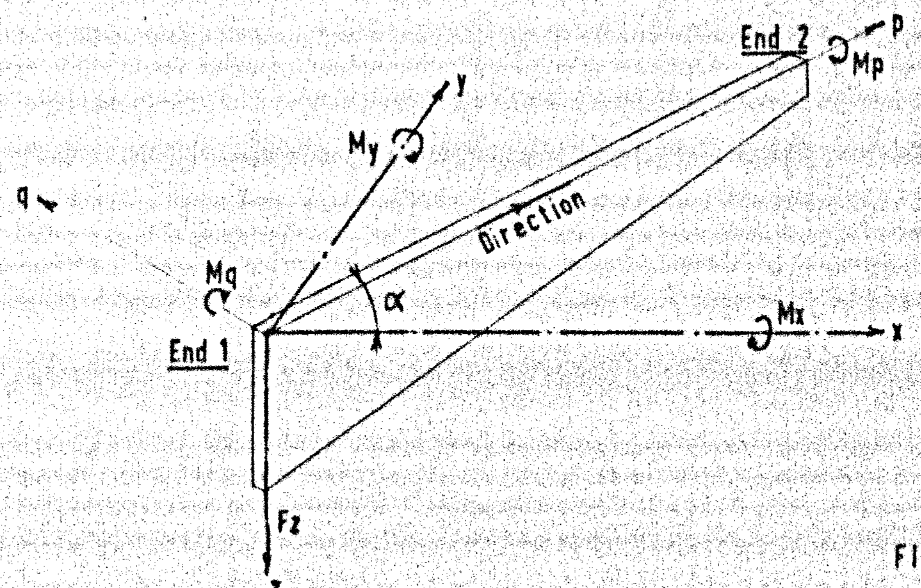
$$
\begin{bmatrix} Mp_2 \\ Mq_2 \\ Fz_2 \end{bmatrix} = \begin{bmatrix} -T & 0 & 0 \\ 0 & C_{12}K_{12} & \dfrac{K_{21}+C_{12}K_{12}}{L} \\ 0 & -\dfrac{K_{12}\left[1+C_{12}\right]}{L} & -\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2} \end{bmatrix} \begin{bmatrix} \theta p_1 \\ \theta q_1 \\ \delta z_1 \end{bmatrix} + \begin{bmatrix} T & 0 & 0 \\ 0 & K_{21} & -\dfrac{\left[K_{21}+C_{12}K_{12}\right]}{L} \\ 0 & -\dfrac{K_{21}\left[1+C_{21}\right]}{L} & \dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2} \end{bmatrix} \begin{bmatrix} \theta q_2 \\ \theta q_2 \\ \delta z_2 \end{bmatrix}
$$

TABLE 2.1

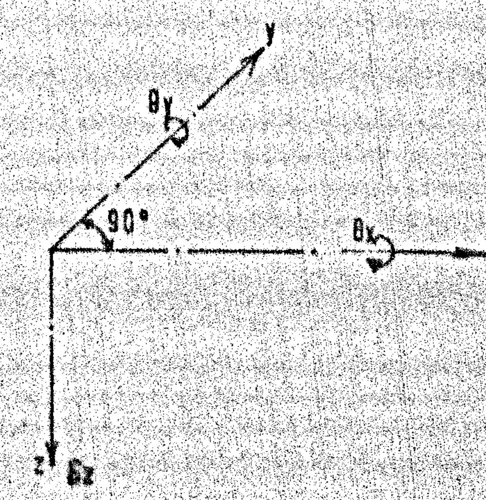
$$
\begin{bmatrix} Mx_{12} \\ My_{12} \\ Fz_{12} \end{bmatrix} =
\begin{bmatrix}
c^2T + s^2K_{12} & sc\left[T - K_{12}\right] & -\dfrac{s\left[K_{12} + C_{21}K_{21}\right]}{L} \\
sc\left[T - K_{12}\right] & s^2T + c^2K_{12} & \dfrac{c\left[K_{12} + C_{21}K_{21}\right]}{L} \\
-\dfrac{sK_{12}\left[1 + C_{12}\right]}{L} & \dfrac{cK_{12}\left[1 + C_{12}\right]}{L} & \dfrac{\left[K_{12} + K_{12}C_{12} + K_{21} + C_{21}K_{21}\right]}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_1 \\ \theta y_1 \\ \delta z_1 \end{bmatrix} +
\begin{bmatrix}
-c^2T + s^2C_{21}K_{21} & -sc\left[T + C_{21}K_{21}\right] & \dfrac{s\left[K_{12} + C_{21}K_{21}\right]}{L} \\
-sc\left[T + C_{21}K_{21}\right] & -s^2T + c^2C_{21}K_{21} & -\dfrac{c\left[K_{12} + C_{21}K_{21}\right]}{L} \\
-\dfrac{sK_{21}\left[1 + C_{21}\right]}{L} & \dfrac{cK_{21}\left[1 + C_{21}\right]}{L} & -\dfrac{\left[K_{12} + C_{12}K_{12} + K_{21} + C_{21}K_{21}\right]}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_1 \\ \theta y_1 \\ \delta z_1 \end{bmatrix}
$$

$$
\begin{bmatrix} Mx_{21} \\ My_{21} \\ Fz_{21} \end{bmatrix} =
\begin{bmatrix}
-c^2T + s^2C_{12}K_{12} & -sc\left[T + C_{12}K_{12}\right] & -\dfrac{s\left[K_{21} + C_{12}K_{12}\right]}{L} \\
-sc\left[T + C_{12}K_{12}\right] & -s^2T + c^2C_{12}K_{12} & \dfrac{c\left[K_{21} + C_{12}K_{12}\right]}{L} \\
\dfrac{sK_{12}\left[1 + C_{12}\right]}{L} & -\dfrac{cK_{12}\left[1 + C_{12}\right]}{L} & -\dfrac{\left[K_{12} + C_{12}K_{12} + K_{21} + C_{21}K_{21}\right]}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_2 \\ \theta y_2 \\ \delta z_2 \end{bmatrix} +
\begin{bmatrix}
c^2T + s^2K_{21} & sc\left[T - K_{21}\right] & \dfrac{s\left[K_{21} + C_{12}K_{12}\right]}{L} \\
sc\left[T - K_{21}\right] & s^2T + c^2K_{21} & -\dfrac{c\left[K_{21} + C_{12}K_{12}\right]}{L} \\
\dfrac{sK_{21}\left[1 + C_{21}\right]}{L} & -\dfrac{cK_{21}\left[1 + C_{21}\right]}{L} & \dfrac{\left[K_{12} + C_{12}K_{12} + K_{21} + C_{21}K_{21}\right]}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_2 \\ \theta y_2 \\ \delta z_2 \end{bmatrix}
$$

TABLE 2.2



MOMENT AND FORCES



DISPLACEMENTS

FIGURE 2.2

$$\begin{bmatrix} Sp_1 \\ Qq_1 \\ M_1 \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ 0 & \frac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2} & \frac{K_{12}\left[1+C_{12}\right]}{L} \\ 0 & \frac{\left[K_{12}+C_{21}K_{21}\right]}{L} & K_{12} \end{bmatrix} \begin{bmatrix} \delta p_1 \\ \delta q_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} -A & 0 & 0 \\ 0 & -\frac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2} & \frac{K_{21}\left[1+C_{21}\right]}{L} \\ 0 & -\frac{\left[K_{12}+C_{21}K_{21}\right]}{L} & C_{21}K_{21} \end{bmatrix} \begin{bmatrix} \delta p_2 \\ Sq_2 \\ \theta_2 \end{bmatrix}$$

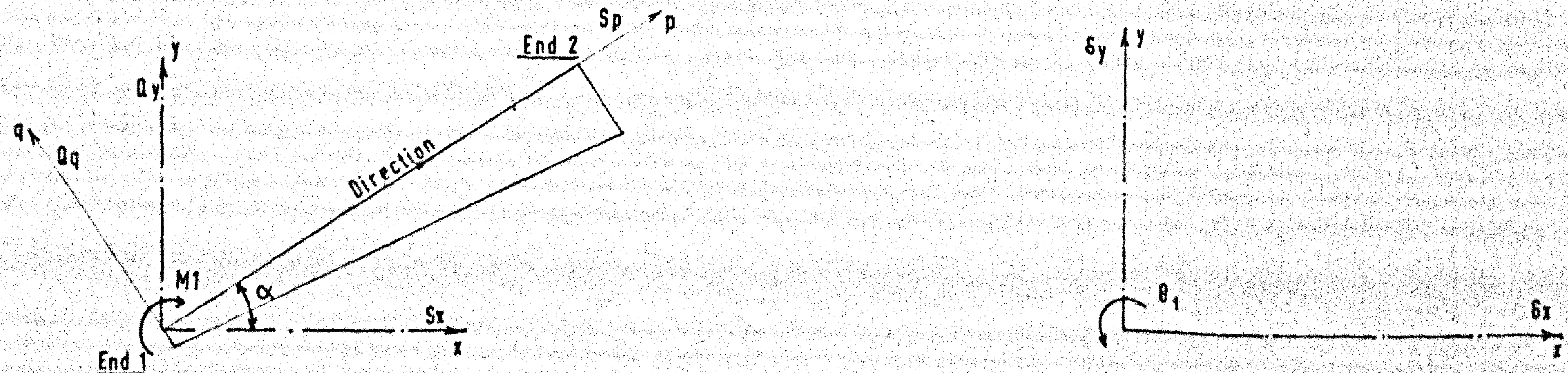
$$\begin{bmatrix} Sp_2 \\ Qq_2 \\ M_2 \end{bmatrix} = \begin{bmatrix} -A & 0 & 0 \\ 0 & -\frac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2} & -\frac{K_{12}\left[1+C_{12}\right]}{L} \\ 0 & \frac{\left[K_{21}+C_{12}K_{12}\right]}{L} & C_{12}K_{12} \end{bmatrix} \begin{bmatrix} \delta p_1 \\ \delta q_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} A & 0 & 0 \\ 0 & \frac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2} & -\frac{\left[K_{21}+C_{12}K_{12}\right]}{L} \\ 0 & -\frac{\left[K_{21}+C_{12}K_{12}\right]}{L} & K_{21} \end{bmatrix} \begin{bmatrix} \delta p_2 \\ \delta q_2 \\ \theta_2 \end{bmatrix}$$

TABLE 2.3

$$
\begin{bmatrix} S_{12} \\ Q_{12} \\ M_{12} \end{bmatrix} =
\begin{bmatrix}
s^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}+c^2A & sc\left[A-\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}\right] & -s\dfrac{K_{12}\left[1+C_{12}\right]}{L} \\
sc\left[A-\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}\right] & c^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}+s^2A & c\dfrac{K_{12}\left[1+C_{12}\right]}{L} \\
-s\dfrac{\left[K_{12}+C_{21}K_{21}\right]}{L} & c\dfrac{\left[K_{12}+C_{21}K_{21}\right]}{L} & K_{12}
\end{bmatrix}
\begin{bmatrix} \delta x_1 \\ \delta y_1 \\ \theta_1 \end{bmatrix}
+
\begin{bmatrix}
-s^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}-c^2A & -sc\left[A-\dfrac{K_{12}\ C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}\right] & -s\dfrac{K_{21}\left[1+C_{21}\right]}{L} \\
-sc\left[A-\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}\right] & -c\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}-s^2A & c\dfrac{K_{21}\left[1+C_{21}\right]}{L} \\
s\dfrac{\left[K_{12}+C_{21}K_{21}\right]}{L} & -c\dfrac{\left[K_{12}+C_{21}K_{21}\right]}{L} & C_{21}K_{21}
\end{bmatrix}
\begin{bmatrix} \delta x_2 \\ \delta y_2 \\ \theta_2 \end{bmatrix}
$$

$$
\begin{bmatrix} S_{21} \\ Q_{21} \\ M_{21} \end{bmatrix} =
\begin{bmatrix}
-s^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}-c^2A & sc\left[\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}-A\right] & s\dfrac{K_{12}\left[1+C_{12}\right]}{L} \\
sc\left[\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}-A\right] & -c^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}-s^2A & -c\dfrac{K_{12}\left[1+C_{21}\right]}{L} \\
-s\dfrac{\left[K_{21}+C_{12}K_{12}\right]}{L} & c\dfrac{\left[K_{21}+C_{12}K_{12}\right]}{L} & C_{12}K_{12}
\end{bmatrix}
\begin{bmatrix} \delta x_1 \\ \delta y_1 \\ \theta_1 \end{bmatrix}
+
\begin{bmatrix}
s^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}+c^2A & sc\left[A-\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}\right] & s\dfrac{K_{21}\left[1+C_{12}\right]}{L} \\
-sc\left[\dfrac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2}-A\right] & c^2\dfrac{\left[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}\right]}{L^2}+s^2A & -c\dfrac{\left[K_{21}+C_{12}K_{12}\right]}{L} \\
s\dfrac{\left[K_{21}+C_{12}K_{12}\right]}{L} & -c\dfrac{\left[K_{21}+C_{12}K_{12}\right]}{L} & K_{21}
\end{bmatrix}
\begin{bmatrix} \delta x_2 \\ \delta y_2 \\ \theta_2 \end{bmatrix}
$$

TABLE 2.4



(a) MOMENTS AND FORCES

(b) DISPLACEMENTS

FIGURE 2.4

```
3*BAY*PORTAL*FRAME.;
10;  11;  4;
13000;
1;  2;  2;  0;  +22.5;  1;  385.5; 10.29;
10;11;
4;  5;  2;  0;  +22.5;  1;  41.9;  5.93;
7;  8;
2;  3;  3;  +30.0;  +8.0;  11; 705; 474; 196.2; 196.2; 196.2;
                                  196.2; 196.2; 196.2; 196.2;
                                  196.2; 196.2; 8.3; 8.0; 6.47;
                                  6.47; 6.47; 6.47; 6.47; 6.47;
                                  6.47; 6.47; 6.47;
5;  6;
8;  9;
3;  5;  3;  +30.0;  -8.0;  11; 196.2; 196.2; 196.2; 196.2;
                                  196.2; 196.2; 196.2; 196.2;
                                  196.2; 474; 705; 6.47; 6.47;
                                  6.47; 6.47; 6.47; 6.47; 6.47;
                                  6.47; 6.47; 8.0; 8.3;
6;  8;
9;  11;

1;
6;
2;  3;  0;  0;  -0.1666;  0;
3;  5;  0;  0;  -0.1666;  0;
5;  6;  0;  0;  -0.1666;  0;
6;  8;  0;  0;  -0.1666;  0;
8;  9;  0;  0;  -0.0833;  0;
9;  11; 0;  0;  -0.0833;  0;

8;
1;  1;
1;  2;
4;  1;
4;  2;
7;  1;
7;  2;
10; 1;
10; 2;
→
```

Table 2.6. Computer Data for Plane Frame Analysis.

PLANE FRAMEWORK - VARYING SECTION MEMBERS
F.S.and B.K.W. Leeds University 16/6/65.
3 BAY PORTAL FRAME.

LOADING CASE No. 1

MEMBER REACTIONS	TONS					END MOMENTS	TONS FT
	FORCE IN	X-DIRECTION		Y-DIRECTION			
No.	MEMBER	END 1	END 2	END 1	END 2	END 1	END 2
1)	2 - 3	+0.00	+0.00	+2.69	+2.30	+16.530	-10.693
2)	3 - 5	+0.00	+0.00	+2.30	+2.69	+10.693	-16.530
3)	5 - 6	+0.00	+0.00	+2.69	+2.30	+16.530	-10.693
4)	6 - 8	+0.00	+0.00	+2.30	+2.69	+10.693	-16.530
5)	8 - 9	+0.00	+0.00	+1.35	+1.15	+8.265	-5.347
6)	9 - 11	+0.00	+0.00	+1.15	+1.35	+5.347	-8.265

LOADING CASE No. 1

MEMBER		FORCES	TONS	MOMENTS TON FT	
No.	MEMBER	AXIAL	SHEAR	END 1	END 2
1)	1 - 2	+4.683	-1.597	+0.000	-35.926
2)	10 - 11	+2.430	+1.086	+0.000	+24.427
3)	4 - 5	+10.557	+0.097	+0.000	+2.180
4)	7 - 8	+7.320	+0.414	+0.000	+9.319
5)	2 - 3	+2.055	+1.511	+19.396	+27.516
6)	5 - 6	+2.106	+2.078	+36.112	+28.405
7)	8 - 9	+1.364	+0.900	+20.295	+7.653
8)	3 - 5	+2.218	-2.120	-27.516	-38.292
9)	6 - 8	+1.980	-1.602	-28.405	-21.349
10)	9 - 11	+1.328	-0.767	-7.653	-16.163

DISPLACEMENTS RADIANS AND INCHES.

JOINT	X DIRECTION	Y DIRECTION	ROTATION
1)	+0.00000000	+0.00000000	+0.00426977
2)	-0.10764072	-0.00945228	-0.00734354
3)	+0.65260635	-2.89432489	+0.00260518
4)	+0.00000000	+0.00000000	-0.00736413
5)	+1.40479938	-0.03697471	-0.00088062
6)	+2.21241488	-3.10031994	-0.00204954
7)	+0.00000000	+0.00000000	-0.02043681
8)	+3.02361219	-0.02563696	+0.00727793
9)	+2.95382104	+0.21355896	-0.00054932
10)	+0.00000000	+0.00000000	-0.01333475
11)	+2.88971474	-0.00490494	-0.00543844

Table 2.7. Computer Output for Plane Frame Analysis.

Chapter 3

Computer Programs For Determining Influence Lines And Surfaces

3.1) Introduction

Although the design engineer has the facilities to analyse grid and plane frame structures accurately, he is still confronted with the problem of finding which loading system will produce maximum stresses. Previous experience is sometimes sufficient to enable these loading positions to be selected but often they are not so obvious. Multispan bridges with varying flexural rigidity present particular difficulties. The abnormal vehicle of up to 180 tons in weight can be placed virtually anywhere on a complex structural system. The problem is often made more complicated by the bridge being skewed on plan. In the early stages of design, influence lines or surfaces are therefore extremely useful in selecting loading cases for rigorous analysis.

3.2) Theory

There are two methods of determining influence lines or surfaces:

1) Unit point load methods.

A unit load is placed at successive points along the structure. The stresses or stress resultants at the required section are then plotted to give the influence line. This method is suitable when the number of points for which influence lines are required are large, and the number of point loads relatively small. If standard computer programs are used for this analysis the output will consist of stresses and stress resultants at all sections. The required influence lines have to be extracted from those results, a process which can be long and tedious. Programs to output only the relevant information have been written for determining influence lines for members with varying section properties (24,25). The method of analysis employed is the 'flexibility' approach. In using this method the degree of indeterminacy must be known and a system of releases derived to first render the structure statically determinate. Unit point loads are then placed upon the reduced structure and the ordinates of the free bending moment diagrams calculated and input as data. It is for these reasons that the method is not suitable for programming to give a general solution for highly redundant structures and examples have been confined to single girders.

2) Unit deformation methods.

These methods are based upon the Müller-Breslau theorum. The influence line for a stress resultant at any point is given by the deflected shape of the structure if a unit deflection is applied in the line of action of the stress resultant. This approach has been used by Sawko (8) using the 'stiffness' method of analysis to determine influence lines and surfaces for structures composed of prismatic members. In employing the 'stiffness' method the degree of redundancy need not be known and therefore this method is most suitable for grid frameworks. The approach used by Sawko is here extended to include for the effects of varying flexural rigidity. As before the method is explained for a grillage only, as the application to plane frameworks is similar.

A structural member 1-2 shown in Fig. (3.1a) subjected to torsion and bending moments M_p and M_q and a shear force F_z undergoes deflections θ_p, θ_q and δ_z at the two ends, as in Fig. (3.1b). The relationships between forces and displacements are as given in Table (2.1). These equations can be expressed briefly as:-

$$\begin{aligned} F_1 &= K_{12}D_1 + R_{12}D_2 \\ F_2 &= R_{21}D_1 + K_{21}D_2 \end{aligned} \qquad (3.1)$$

Similar equations can be written for every member of the structure. Suppose the influence line for bending moment is required at end 1 of member 1-2. End 1 must now undergo a unit displacement in the plane of bending whilst the displacements θ_p and δ_z remain unchanged. All other members in the structure are not affected by this displacement and remain unaltered. In the equations governing member 1-2 the rotation θ_p becomes (θ_p-1) as shown in Fig. (3.2)

Similarly, unit deflections can be induced in the planes of torsion and shear resulting (θ_p-1) and (δ_z-1) respectively.

By substituting (D_1-1) for D_1 in equations (3.1), that is $(\theta_p -1, \theta_q-1, \delta_z-1)$ for $(\theta_p, \theta_q, \delta_z)$ and expanding, the equations become:-

$$\begin{aligned} F_1 &= K_{12}D_1 + R_{12}D_2 - K_{12}x1 \\ F_2 &= R_{21}D_1 + K_{21}D_2 - R_{21}x1 \end{aligned} \qquad (3.2)$$

and are written fully in Table (3.1).

These equations apply to moments and forces in the individual member axis. To form the stiffness matrix for the complete structure, forces must be expressed in relation to a general system of co-ordinates denoted by x and y (Fig. 3.3)

Using a transformation matrix T the member forces and displacements in Fig. (3.1a) can be related to the general co-ordinates.

$D = TD'$ and $F = TF'$ for ends 1 and 2 of the member.

Substituting in equations (3.2)

$$\begin{aligned} TF_1 &= K_{12}TD_1' + R_{12}TD_2' - K_{12} \\ TF_2 &= R_{21}TD_1' + K_{21}TD_2' - R_{21} \end{aligned}$$

or

$$F_1' = K_{12}'D_1' + R_{12}'D_2' - K_{12}^u$$
$$F_2' = R_{21}'D_1' + K_{21}'D_2' - R_{21}^u \qquad (3.3)$$

where $K_{ij}' = T^{-1}K_{ij}T$

$R_{ij}' = T^{-1}R_{ij}T \qquad (i, j = 1, 2)$

$K^u = T^{-1}K$

$R^u = T^{-1}R$

These equations are given in Table (3.2)

The equations for all other members in the frame are as shown in Table (2.2) and the complete set of equations can be summarised in the form:

$$[F] = [K][D] - \begin{bmatrix} K^u \\ R^u \end{bmatrix} \qquad (3.4)$$

where $[K]$ represents the stiffness matrix for the complete structure, $[D]$ is the displacement vector and $\begin{bmatrix} K^u \\ R^u \end{bmatrix}$ are as defined above. Vector $[F]$ holds the sum of internal forces at every node and is normally equal to the external loading applied to the structure. In this case however, there is no external loading and force vector $[F]$ must equal zero. Equations (3.4) can therefore be written as:

$$\begin{bmatrix} K^u \\ R^u \end{bmatrix} = [K][D] \qquad (3.5)$$

The values of the $\begin{bmatrix} K^u \\ R^u \end{bmatrix}$ vector are given in Table (3.1) but can also be obtained by considering the physical properties of the member. A cut is made at end 1 of member 1-2 and forces applied to induce a unit displacement whilst end 2 remains fixed.

To apply a unit rotation $\theta_q = 1$ at end 1 the force required will equal the stiffness K_{12} at end 1. The force required to prevent any rotation at end 2 will be the stiffness at end 1 multiplied by the carry over from end 1 to end 2, $K_{12}C_{12}$. Shear forces $\frac{K_{12}(1+C_{12})}{L}$ act at end 1 and end 2 of the member.

The forces required to produce unit displacements are given in Fig. (3.4) and are identical to those values obtained from Table (3.1).

After undergoing a unit displacement in the required direction the member is rejoined to the structure. Upon 'release' the structure behaves as if under load: the joints translate and rotate to take up a position of equilibrium to give the influence line required.

The assembled stiffness matrix for any structure does not depend upon the unit displacement imposed as it is defined uniquely by the geometry of the structure. As the unit deflection vectors are treated as loads, several influence lines can be determined simultaneously by treating the deflection vectors as alternative loads.

3.3) Computer Programs.

The stiffness matrix for the complete frame is identical to that set up for the analysis of the structure under the action of loading. The method of specifying the geometry of the structure and evolving the stiffness matrix is therefore the same as for plane or grid frame analysis programs. In place of the loading data the members for which influence lines are required, are listed each followed by two integer parameters. The first is either 1 or 2 and indicates which end of the member is to be considered, the second gives the stress resultant required as summarised in Table (3.3) below:

TABLE 3.3

Grid Framework		Plane Framework	
Integer	Stress Resultant	Integer	Stress Resultant
1	Torsional Moment	1	Thrust
2	Bending Moment	2	Shear
3	Shear.	3	Bending Moment

As explained previously the displacement vectors can be obtained from the member properties. These have already been calculated in order to form the stiffness matrix and are stored in array P.

The zero displacements are imposed upon the set of equations and the solution obtained using the 'square root' method as before.

The output consists of three displacements at each joint. Normally the vertical ordinates only are used, but the horizontal displacements for plane frameworks can be used to find the effects of horizontal forces, such as accelerating or braking vehicles. Rotations are sometimes useful in plotting results when a greater degree of accuracy is required.

Because member forces are not calculated, the time taken for solution of any problem is slightly faster than would be required for a full analysis.

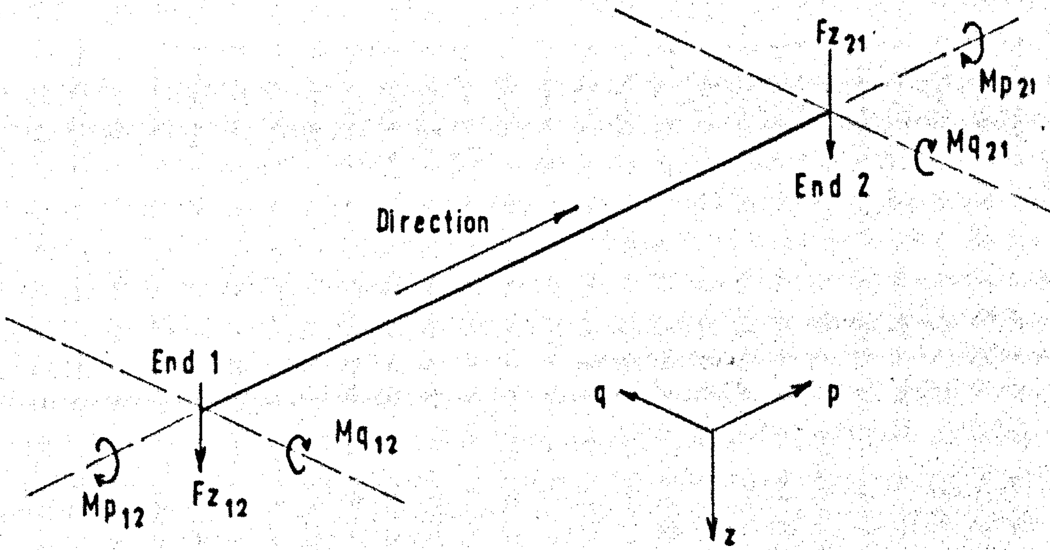
3.4) <u>Example.</u>

A 3 span continuous bridge with varying section longitudinal girders and skewed 45° on plan, as shown in Fig. (3.5) is considered to obtain the influence surface for bending moment in the edge girder over the support i.e. at end 2 of member 30-35. This example has been analysed by Sawko (8) where the varying section girders were considered as consisting of a series of stepped uniform sections. The uniform sections were taken as acting between the joints indicated and average values of I and J were used for analysis purposes.

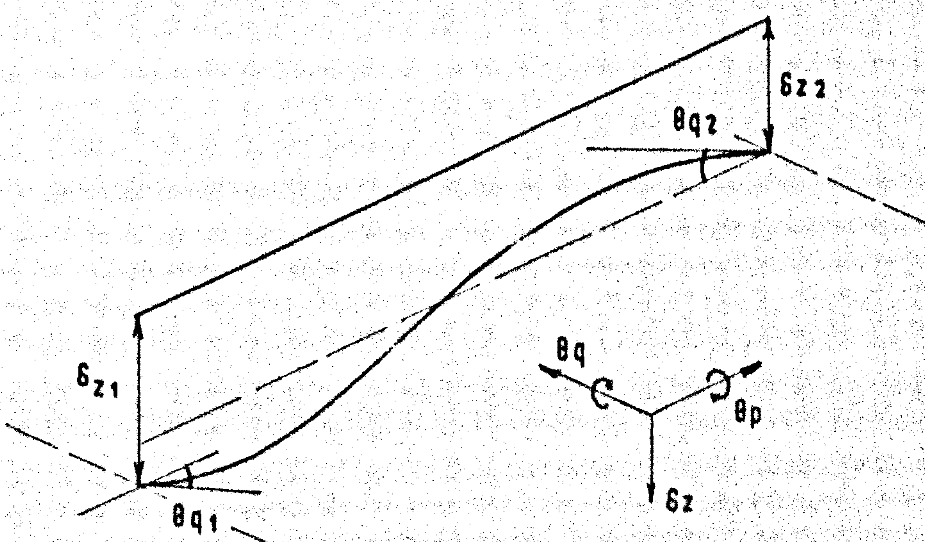
Using the program described earlier in this chapter, this example was recalculated taking into account the variation in longitudinal stiffness. Member properties were specified at five points along each longitudinal member and at one point for the uniform transverse members. A copy of the data is given in Table(3.4) and the computer output is given in Table (3.5). The time taken for the analysis was 1 min. 10 secs.

These results are plotted in Fig. (3.6) together with those obtained by Sawko - shown in square brackets. It can be seen that at the maximum ordinate at joint 30, the results due to Sawko underestimate those obtained by more rigorous analysis, by almost 8%. Thus by taking average values of member properties, the stiffness of the members at the supports is underestimated leading to an apparent reduction in bending moment at these points. A corresponding increase in midspan bending moments can be expected when using the latter method.

The accuracy of results obtained using the program for uniform members can be increased by increasing the number of joints. The change in section at joints is then reduced resulting in a truer approximation of the actual girder profile. Unfortunately this has the disadvantage of increasing the time for solution of the problem, as this is dependent upon the number of joints. By using the varying section program the members can be considered in terms of their actual stiffness and restraint factors, and hence a more accurate solution is obtained without any increase in time.



a) Forces.



b) Displacements.

FIG. 3.1 Forces and Displacements Related to Member Axes.

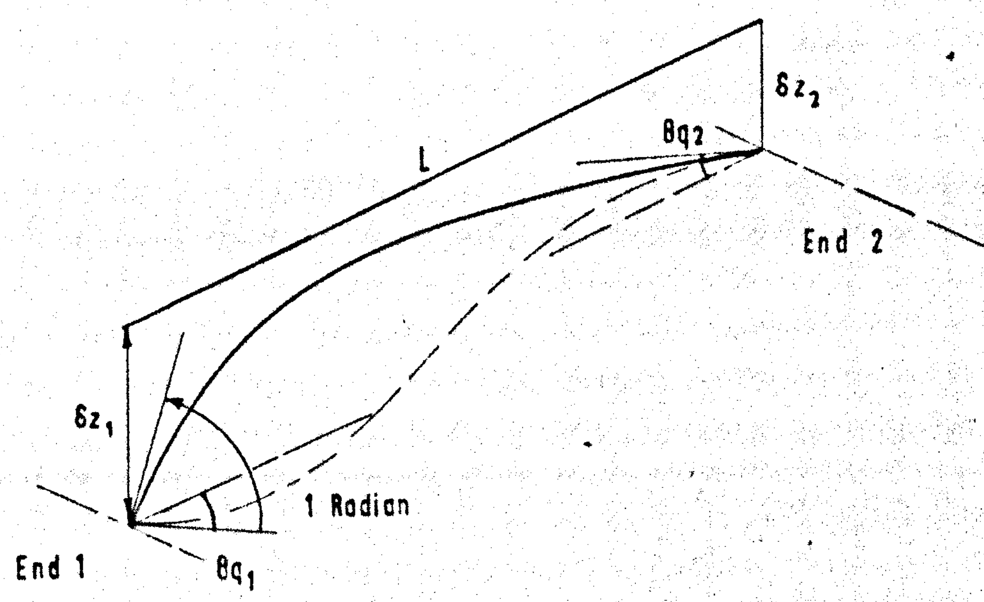


FIG. 3.2 Unit Displacement for Bending Moment.

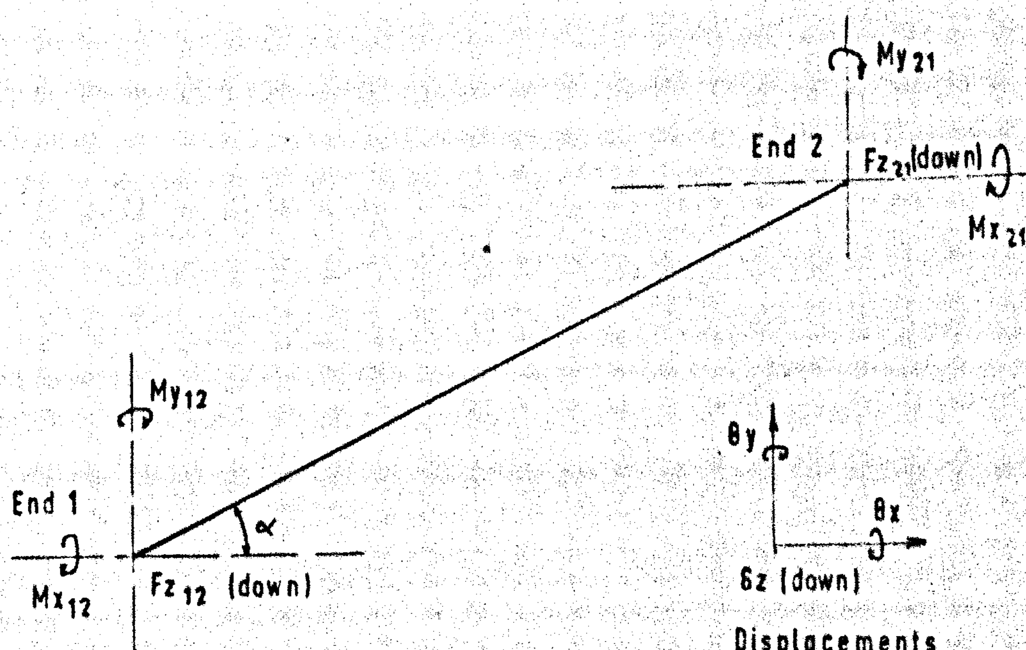


FIG. 3.3 Forces and Displacements Related to General Axes.

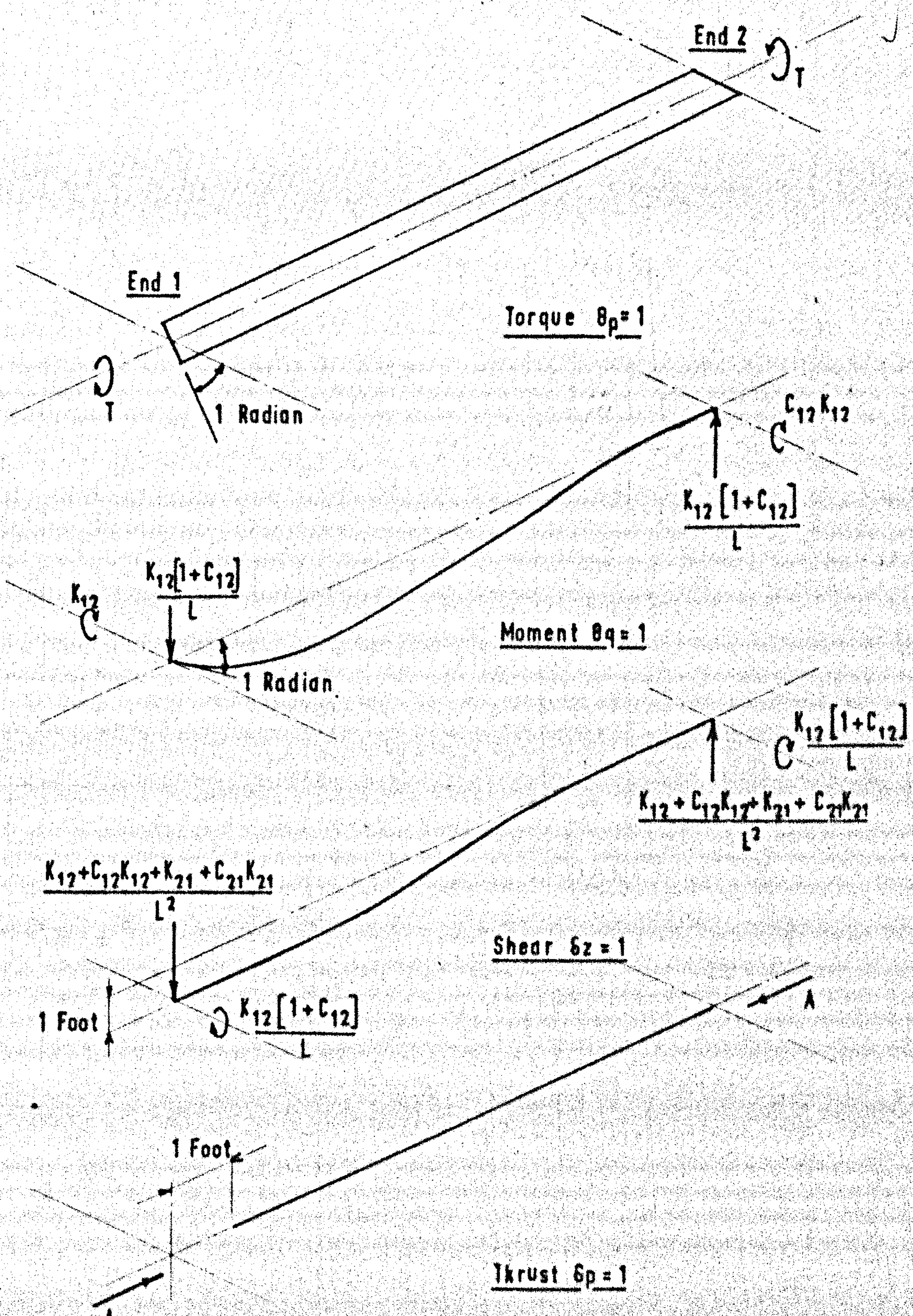
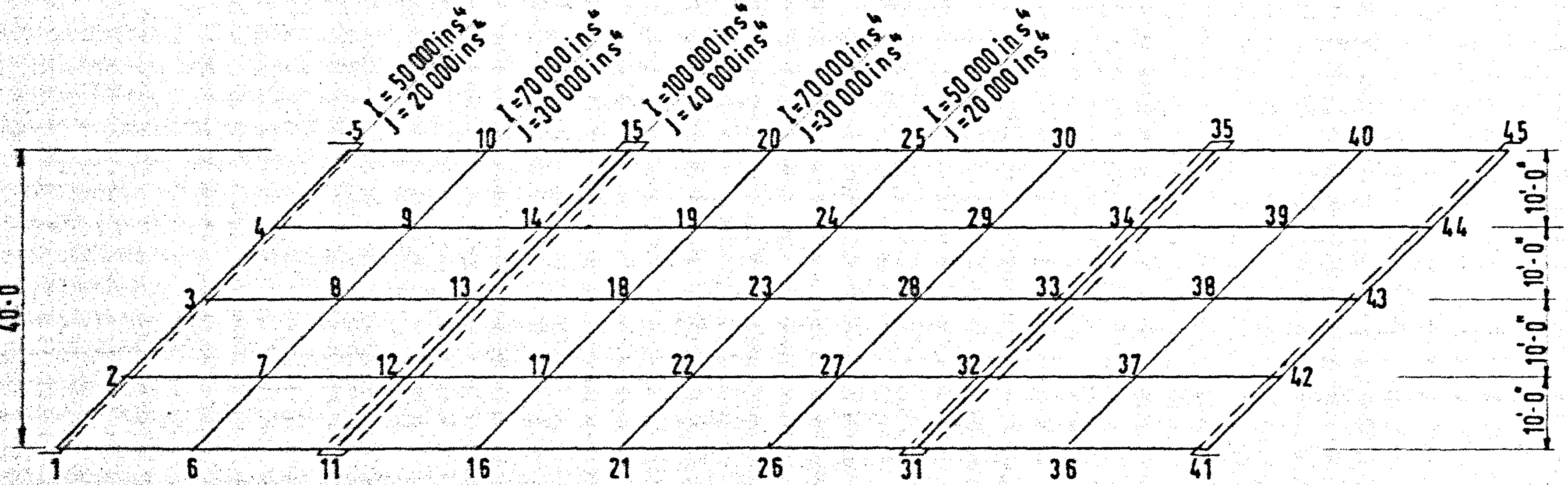
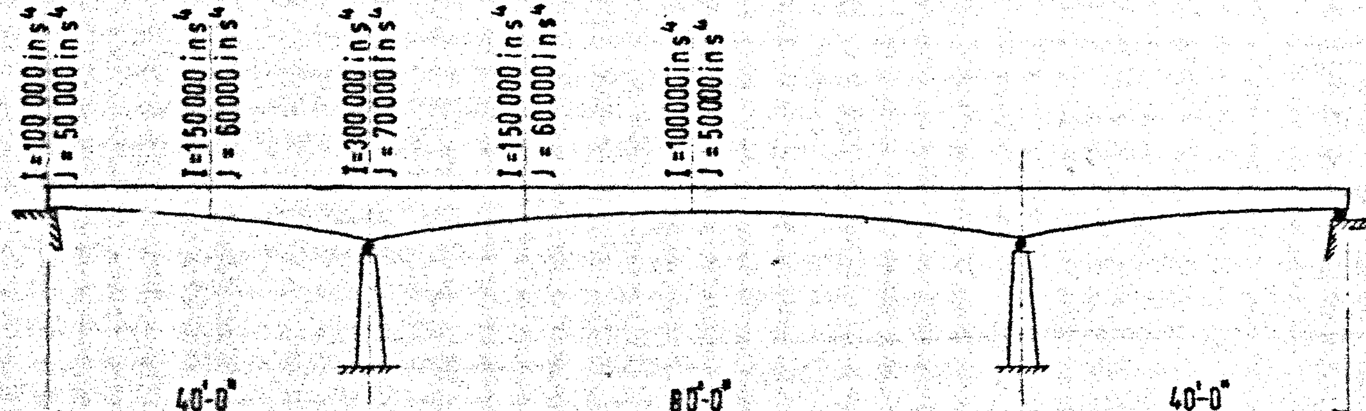


FIG. 3.4 Forces and Moments to Produce Unit Displacement at End 1 of Member 1-2.



Plan of Deck.



Elevation

Fig 3.5 3 Span Bridge for Influence Surface Analysis.

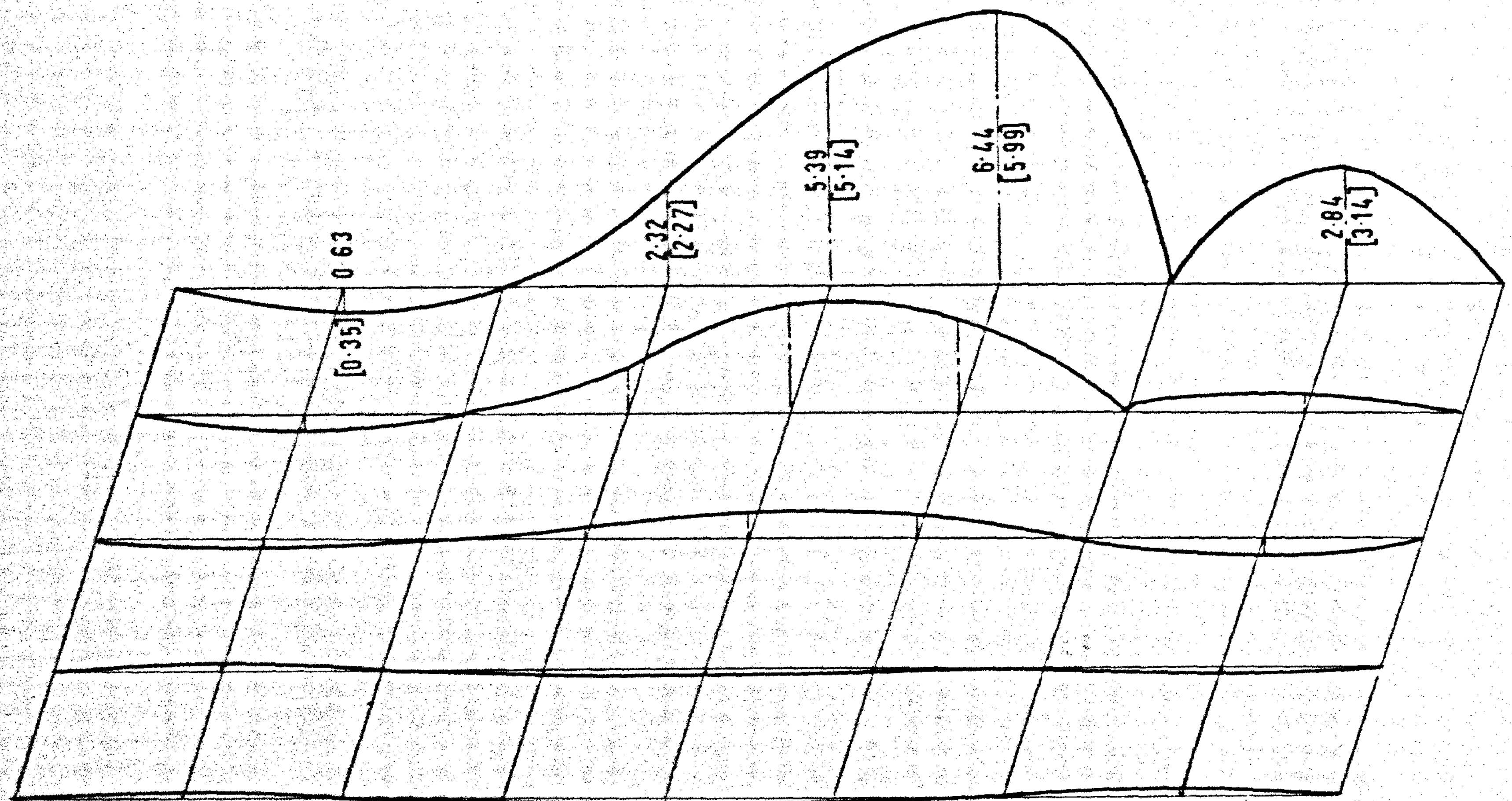


Fig 3.6 Influence Surface Ordinates for Bending Moment in Outer Girder Over Pier.

$$
\begin{bmatrix} Mp_{12} \\ Mq_{12} \\ Fz_{1} \end{bmatrix} =
\begin{bmatrix} T & 0 & 0 \\ 0 & K_{12} & \frac{K_{12}+C_{21}K_{21}}{L} \\ 0 & \frac{K_{12}[1+C_{12}]}{L} & \frac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2} \end{bmatrix}
\begin{bmatrix} \theta p_{1} \\ \theta q_{1} \\ \delta z_{1} \end{bmatrix} +
\begin{bmatrix} -T & 0 & 0 \\ 0 & C_{21}K_{21} & -\frac{[K_{12}+C_{21}K_{21}]}{L} \\ 0 & \frac{K_{21}[1+C_{21}]}{L} & -\frac{[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}]}{L^2} \end{bmatrix}
\begin{bmatrix} \theta p_{2} \\ \theta q_{2} \\ \delta z_{2} \end{bmatrix} -
\begin{bmatrix} T & 0 & 0 \\ 0 & K_{12} & \frac{K_{12} \quad C_{21}K_{21}}{L} \\ 0 & \frac{K_{12} \quad 1 \quad C_{12}}{L} & \frac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L^2} \end{bmatrix}
$$

$$
\begin{bmatrix} Mp_{21} \\ Mq \\ Fz_{21} \end{bmatrix} =
\begin{bmatrix} -T & 0 & 0 \\ 0 & C_{12}K_{12} & \frac{K_{21}+C_{12}K_{12}}{L} \\ 0 & -\frac{K_{12}[1+C_{12}]}{L} & -\frac{[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}]}{L^2} \end{bmatrix}
\begin{bmatrix} \theta p_{1} \\ \theta q_{1} \\ \delta z_{1} \end{bmatrix} +
\begin{bmatrix} T & 0 & 0 \\ 0 & K_{21} & -\frac{[K_{21}+C_{12}K_{12}]}{L} \\ 0 & -\frac{K_{21}[1+C_{21}]}{L} & \frac{K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}}{L} \end{bmatrix}
\begin{bmatrix} \theta p_{2} \\ \theta q_{2} \\ \delta z_{2} \end{bmatrix} -
\begin{bmatrix} T & 0 & 0 \\ 0 & C_{12}K_{12} & \frac{K_{21}+C_{12}K_{12}}{L} \\ 0 & -\frac{K_{12}[1+C_{12}]}{L} & -\frac{[K_{12}+C_{12}K_{12}+K_{21}+C_{21}K_{21}]}{L^2} \end{bmatrix}
$$

TABLE 3.1

$$
\begin{bmatrix} Mx_{12} \\ My_{12} \\ Fz_{12} \end{bmatrix}
=
\begin{bmatrix}
c^2T + s^2K_{12} & sc\left[T - K_{12}\right] & -s\dfrac{\left[K_{12} + C_{21}K_{21}\right]}{L} \\
sc\left[T - K_{12}\right] & s^2T + c^2K_{12} & c\dfrac{\left[K_{12} + C_{21}K_{21}\right]}{L} \\
\dfrac{-sK_{12}\left[1 + C_{12}\right]}{L} & \dfrac{cK_{12}\left[1 + C_{12}\right]}{L} & \dfrac{\left[K_{12} + C_{12}K_{12} + C_{21}K_{21} + K_{21}\right]}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_1 \\ \theta y_1 \\ \delta z_1 \end{bmatrix}
$$

$$
+
\begin{bmatrix}
-c^2T + s^2C_{21}K_{21} & -sc\left[T + C_{21}K_{21}\right] & s\dfrac{\left[K_{12} + C_{21}K_{21}\right]}{L} \\
sc\left[T + C_{21}K_{21}\right] & s^2T + c^2C_{21}K_{21} & c\dfrac{\left[K_{12} + C_{21}K_{21}\right]}{L} \\
\dfrac{sK_{21}\left[1 + C_{21}\right]}{L} & \dfrac{cK_{21}\left[1 + C_{21}\right]}{L} & \dfrac{\left[K_{12} + C_{12}K_{12} + C_{21}K_{21} + K_{21}\right]}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_2 \\ \theta y_2 \\ \delta z_2 \end{bmatrix}
-
\begin{bmatrix}
cT & -sK_{12} & -s\dfrac{\left[K_{12} + C_{21}K_{21}\right]}{L} \\
sT & cK_{12} & c\dfrac{\left[K_{12} + C_{21}K_{21}\right]}{L} \\
0 & \dfrac{K_{12}\left[1 + C_{12}\right]}{L} & \dfrac{K_{12} + C_{12}K_{12} + K_{21} + C_{21}K_{21}}{L^2}
\end{bmatrix}
$$

$$
\begin{bmatrix} Mx_{21} \\ My_{21} \\ Fz_{21} \end{bmatrix}
=
\begin{bmatrix}
-c^2T + s^2C_{12}K_{12} & -sc\left[T + C_{12}K_{12}\right] & -s\dfrac{\left[K_{21} + C_{12}K_{12}\right]}{L} \\
sc\left[T + C_{12}K_{12}\right] & -s^2T + c^2C_{12}K_{12} & c\dfrac{\left[K_{21} + C_{12}K_{12}\right]}{L} \\
\dfrac{sK_{12}\left[1 + C_{12}\right]}{L} & \dfrac{cK_{12}\left[1 + C_{12}\right]}{L} & -\dfrac{\left[K_{12} + C_{12}K_{12} + C_{21}K_{21} + K_{21}\right]}{L}
\end{bmatrix}
\begin{bmatrix} \theta x_1 \\ \theta y_1 \\ \delta z_1 \end{bmatrix}
$$

$$
+
\begin{bmatrix}
c^2T + s^2K_{21} & sc\left[T - K_{21}\right] & s\dfrac{\left[K_{21} + C_{12}K_{12}\right]}{L} \\
sc\left[T - K_{21}\right] & s^2T + c^2K_{21} & -c\dfrac{\left[K_{21} + C_{12}K_{12}\right]}{L} \\
\dfrac{sK_{21}\left[1 + C_{21}\right]}{L} & \dfrac{-cK_{21}\left[1 + C_{21}\right]}{L} & \dfrac{K_{12} + C_{12}K_{12} + C_{21}K_{21} + K_{21}}{L^2}
\end{bmatrix}
\begin{bmatrix} \theta x_2 \\ \theta y_2 \\ \delta z_2 \end{bmatrix}
-
\begin{bmatrix}
-cT & -sC_{12}K_{12} & -s\dfrac{\left[K_{21} + C_{12}K_{12}\right]}{L} \\
-sT & cC_{12}K_{12} & c\dfrac{\left[K_{21} + C_{12}K_{12}\right]}{L} \\
0 & \dfrac{K_{12}\left[1 + C_{12}\right]}{L} & -\dfrac{\left[K_{12} + C_{12}K_{12} + K_{21} + C_{21}K_{21}\right]}{L^2}
\end{bmatrix}
$$

TABLE 3.2

```
3*SPAN*SKEW*BRIDGE*DECK.;
76;  45;  7;
13000;  6500;
1;   2;   12;   +10.0;   +10.0;   1;   5.0₁₀4;   2.0₁₀4;
2;   3;
3;   4;
4;   5;
21;  22;
22;  23;
23;  24;
24;  25;
41;  42;
42;  43;
43;  44;
44;  45;
6;   7;   16;   +10.0;   +10.0;   1;   7.0₁₀4;   3.0₁₀4;
7;   8;
8;   9;
9;   10;
16;  17;
17;  18;
18;  19;
19;  20;
26;  27;
27;  28;
28;  29;
29;  30;
36;  37;
37;  38;
38;  39;
39;  40;
11;  12;   8; +10.0; +10.0; 1; 1.0₁₀5;  4.0₁₀4;
12;  13;
13;  14;
14;  15;
31;  32;
32;  33;
33;  34;
34;  35;
1;   6;   10; +20.0; 0; 5; 1.0₁₀5; 1.06₁₀5; 1.15₁₀5; 1.28₁₀5; 1.5₁₀5;
                           5.0₁₀4; 5.25₁₀4; 5.5₁₀4; 5.75₁₀4; 6.0₁₀4;
2;   7;
3;   8;
4;   9;
5;   10;
21;  26;
22;  27;
23;  28;
24;  29;
25;  30;
6;   11;   10;   +20.0; 0; 5; 1.5₁₀5; 1.78₁₀5; 2.1₁₀5; 2.5₁₀5; 3.0₁₀5;
                              6.0₁₀4; 6.25₁₀4; 6.5₁₀4; 6.75₁₀4; 7.0₁₀4;
7;   12;
8;   13;
```

Cont/

```
9;  14;
10; 15;
26; 31;
27; 32;
28; 33;
29; 34;
30; 35;
11; 16;   10; +20.0; 0; 5; 3.0⏨5; 2.5⏨5; 2.15⏨5; 1.78⏨5; 1.5⏨5;
                           7.0⏨4; 6.75⏨4; 6.5⏨4; 6.25⏨4; 6.0⏨4;
12; 17;
13; 18;
14; 19;
15; 20;
31; 36;
32; 37;
33; 38;
34; 39;
35; 40;
16; 21;   10; +20.0; 0; 5; 1.5⏨5; 1.28⏨5; 1.15⏨5; 1.06⏨5; 1.0⏨5;
                           6.0⏨4; 5.75⏨4; 5.5⏨4; 5.25⏨4; 5.0⏨4;
17; 22;
18; 23;
19; 24;
20; 25;
36; 41;
37; 42;
38; 43;
39; 44;
40; 45;

1;
30;  35;  2;  2;

20;
1;  3;
2;  3;
3;  3;
4;  3;
5;  3;
11; 3;
12; 3;
13; 3;
14; 3;
15; 3;
31; 3;
32; 3;
33; 3;
34; 3;
35; 3;
41; 3;
42; 3;
43; 3;
44; 3;
45; 3;
→
```

Table 3.4 Computer Data.

INFLUENCE SURFACE ORDINATES
GRID FRAMEWORK - VARYING SECTION MEMBERS
F.S. AND B.K.W. Leeds University 7/12/65.

3 SPAN SKEW BRIDGE DECK.
CASE No. 1

DEFLEXIONS RADIANS AND FEET

JOINT	X ROTATION	Y ROTATION	VERTICAL
1)	+0.00343721	+0.00390273	+0.00000000
2)	+0.00109802	+0.00098091	+0.00000000
3)	-0.00502075	-0.00622825	+0.00000000
4)	-0.01888783	-0.01960471	+0.00 00000
5)	-0.02850938	-0.03992787	+0.00000000
6)	+0.00269713	+0.00033526	+0.05531440
7)	+0.00628464	-0.00018197	+0.01043356
8)	+0.01312115	-0.00154634	-0.10002036
9)	+0.02164510	-0.00412670	-0.31326880
10)	+0.02435490	-0.00814199	-0.62674123
11)	-0.00585738	-0.00606759	+0.00000000
12)	-0.00139476	-0.00129298	+0.00000000
13)	+0.00942582	+0.01096872	+0.00000000
14)	+0.03465678	+0.03484151	+0.00000000
15)	+0.05868099	+0.07305845	+0.00000000
16)	-0.01704487	-0.00926937	-0.17078284
17)	-0.02266941	-0.00153338	-0.03539301
18)	-0.03369444	+0.02087765	+0.33077474
19)	-0.03053976	+0.07151782	+1.09681780
20)	-0.00583793	+0.14696993	+2.32126748
21)	-0.01693383	-0.00106814	-0.30156091
22)	-0.03851797	+0.00029743	-0.05674638
23)	-0.10755876	+0.00855945	+0.68270073
24)	-0.19193217	+0.05083586	+2.52087129
25)	-0.18866953	+0.13095021	+5.39189581
26)	-0.00417346	+0.00962023	-0.19753113
27)	-0.02328205	+0.00195997	-0.03308580
28)	-0.11621265	-0.02312124	+0.47305517
29)	-0.37029232	-0.08374469	+2.27503851
30)	-0.40328865	-0.04957389	+6.44161922
31)	+0.00471178	+0.00691014	+0.00000000
32)	-0.00095138	-0.00318846	+0.00000000
33)	-0.03411671	-0.02275532	+0.00000000
34)	-0.02249580	-0.05713980	+0.00000000
35)	+0.19526049	+0.36208904	+0.00000000
36)	+0.01105815	-0.00155218	+0.05228439
37)	+0.01740052	-0.00384415	-0.11728300
38)	-0.00232912	-0.00204650	-0.30541806
39)	-0.13416130	+0.03896811	+0.34806665
40)	-0.23014448	-0.02758954	+2.83728386
41)	+0.00113940	-0.00180307	+0.00000000
42)	+0.01155385	+0.01123594	+0.00000000
43)	+0.01507492	+0.01830184	+0.00000000
44)	-0.06763574	-0.05663361	+0.00000000
45)	-0.18892473	-0.19962443	+0.00000000

Table 3.5 Computer Output - Influence Surface Ordinates.

Chapter 4.

Experimental Tests.

4.1) Introduction.

In order to examine the accuracy of the programs, and the degree of approximation involved when the grillage with varying section members is used as a mathematical model to simulate plated structures, a comparison of practical and theoretical results was thought desirable. There are very few recorded results of tests carried out on structures with varying section properties. Madu (13) conducted tests on aluminium bridge decks, but confined the investigation to simply supported spans containing only three longitudinal girders. Model tests were carried out as part of the design procedure for the Clifton Bridge, Nottingham (32). A perspex model of one of the cantilever end span grillages was tested to obtain deflections for load and prestress, but no values of bending moment were obtained. A close correlation of deflections does not necessarily indicate the same agreement in bending moments, and therefore these results were considered to be inadequate. The Cement and Concrete Association have carried out a series of model tests on the cantilevers of the Medway Bridge using a prestressed concrete scale model. The bridge deck consists of three hollow box girders with 12' 0" side slabs. The girders are hollow throughout their lengths without transverse diaphragms and therefore it was felt that the grillage analogy could not be applied with any degree of accuracy.

In view of the lack of suitable experimental results, it was decided to carry out a series of model tests as part of this research. A three span continuous model bridge, having a parabolic variation in longitudinal depth, was chosen. The model was fabricated from standard perspex sheets, cemented at the joints. Although the use of perspex assisted greatly during the fabrication of the model, certain problems arose during the testing. Perspex is a non-elastic material that creeps under sustained loading and therefore the loads had to be kept relatively small and they could only be applied for short periods.

During the fabrication of the model it was possible to test it at three stages of construction and so apply the method of grillage analysis as a mathematical model to various types of structures. The model was also analysed using the simply supported span technique so that the tests also provided a means of comparing the two methods of analysis.

The model was built and tested in three stages:

Stage One consisted of an open grillage with a variation in section in longitudinal members, as shown in Plate 1. In this form the structure is closest to the mathematical model analysed, and was used to verify the accuracy of the computer program. The restriction in width of the members made it impractical to attach strain measuring devices and only deflections were measured at this stage.

Stage Two consisted of a torsionally weak system of interconnected tee beams formed by the addition of a top plate to stage 1. Electrical resistance strain gauges were fixed to the top plate and strains and deflections were measured.

Stage Three consisted of a cellular deck of interconnected hollow box beams varying in longitudinal stiffness, as shown in Plate 4. This was achieved by the addition of bottom plates to the model in the previous stage. The structure at this stage is a three dimensional assemblage of plates and quite far removed from the mathematical model of an open grillage used in the computer solution. Strains and deflections were again obtained.

4.2.) Description of Model and Testing.

A three span continuous bridge with a longitudinal variation in depth was represented by a perspex model. After construction the dimensions were checked and these are shown in Fig. (4.1). A depth of approximately ½" at the centre varying parabolically to 2" over the support gives a variation in the moment of inertia of 1:64 for the open grillage and 1:15.6 for the grillage plus top and bottom deck plates. The main diaphragms are ¼" thick and the top and bottom decks 1/8" thick. To facilitate easier construction and alignment of the main grillage, positioning grooves were machined into the main girders. All diaphragms were a tight push fit so that this form of construction should not lead to a reduction in the effective sections at these points.

The assembly of the grillage took several hours and consequently it was necessary to use a jointing cement that, as well as providing sufficient strength, also remained workable for the whole of this period. Several types of adhesive were investigated and it was found that Tensal No. 3(26) best suited these requirements. It is an all acrylic cement that remains in a liquid state until hardened by polymerisation. The manufacturers claim a 'bond strength' of 6000 lb/sq.ins. The cement was prepared in the ratio seven parts stabilised methyl methacrylate monomer liquid to one part methyl methacrylate monomer powder. The powder contains a proportion of photo-catalyst. The mixture was allowed to stand in a dark place for 24 hours before use. Hardening was effected by light polymerisation using mercury vapour fluorescent tubes. A special cabinet was built containing three tubes held in position one foot above the model, as shown in Plate 3. At each stage of assembly the model was exposed for approximately twelve hours to the ultra violet light.

The deck is supported along four lines of support each capable of rotation; the end supports also prevent upward displacement. The end support columns are rigid at one end and pinned top and bottom at the other thus allowing any horizontal movement to take place unimpeded. The central supports are adjustable to ensure contact along the lines of support. The model is mounted about seven inches above a steel plate base giving a rigid support and also allowing magnetic based deflection gauges to be fixed underneath.

Loads are applied to the deck through a pinned lever arm which has a hanger to receive weights. The point of contact is through a sliding ferrule, as shown in Plate 2. Any load applied is increased because of the lever arm effect, which accounts for the apparent lack of uniformity in the applied loads given in the results.

Vertical displacements were measured at each stage of testing to give transverse deflection profiles at the centre of the main and end spans. Dial gauges measuring to 0.0001" were used for this purpose. It was found necessary to lightly tap the model at dial gauge points after the application of each load to improve the response.

Strains were measured in stages 2 and 3 and bending moments calculated using the relationship

$$M = \frac{E\, y_2\, \varepsilon}{I} \qquad \ldots (4.1)$$

where E is the modulus of elasticity - taken as 440,000 lbs./sq. in.

y_2 is the distance from the neutral axis to the extreme fibre.

ε is the measured strain.

I is the moment of inertia at the centre of the gauge.

Saunders Rowe ½" linear foil strain gauges were used, attached to the model with Eastmans 901 adhesive and GA - 1A1 accelerator. The strain gauges were connected to a multiway junction box and readings taken on a Peekel strain indicator, type B103U. Only one dummy gauge was used for all readings.

It was found during testing that the model tended to creep under sustained loading, therefore readings were taken in groups of four only, the load being released and reapplied each time. Each loading was repeated four times and the experimental results plotted in Section (4.4) of this chapter are the average of these readings.

4.3) Material Properties.

To establish the material properties of the perspex, a test specimen was cut from the sheet used to fabricate the main girders. Ten E.R. strain gauges were attached in pairs on opposite sides of the specimen as shown in Fig. (4.2) and the specimen tested in a Hounsfield tensometer. The load is applied through a steel plate spring and registered as a mercury column in a graduated tube. The average results are plotted in Fig. (4.2) from which it was established that the modulus of elasticity $\underline{E}$ = 440,000 lb/sq. in. and the modulus of rigidity $\underline{G}$ = 157,000 lbs/sq. in.

Tests were also carried out to ascertain the strength of Tensol No. 3 cement. Tensile specimens gave erratic results which seemed to be due to the specimens being loaded eccentrically. Bond tests proved to be more regular and an average of 4225 lb/sq. in. was recorded. The specimens consisted of a perspex block cemented between two side pieces. The whole was clamped in a frame to prevent horizontal movement and the centre block pushed out. In some cases the parent perspex fractured in preference to the cemented joints. Therefore it was felt that although the strength claimed by the manufacturers had not been attained, Tensol No. 3 could reliably be used for all joints in the model.

4.4.1) Results.

The experimental deflections were obtained directly from the dial gauge readings and the bending moments were calculated using Eq. (4.1). The values of y_2 in this equation were obtained from the results of the computer program given in Appendix 2.

Computer results were obtained using the grid framework program described in Chapter 2. Moments of inertia were calculated in the normal way, taking the effective width of beam flanges to be equal to the centres of webs, i.e. 1" for longitudinal beams and $2\frac{1}{2}$" for transverse beams. The torsional constants for stages 1 and 2 were derived using the formula:

$$J = \sum kbd^3 \qquad \ldots (4.2)$$

The contributions from the top plate in stage 2 were reduced by half to include for the overall continuity in the transverse and longitudinal directions, of this plate.

The torsional constants for the box girders were calculated using the formula for thin walled sections:-

$$J = \frac{4A^2}{\oint ds/t} \qquad \ldots (4.3)$$

The sectional properties were specified at nine points in each section of longitudinal girder. A short computer program was written to calculate member properties and the results are given in Appendix 2. For the purpose of this program the factor k in Eq. (4.2) was evaluated using the expression stated by Kantorovich and Krylov (27):

$$k = \left[1 - \sqrt{\frac{2}{5}}\ \frac{b}{d}\ \tanh \sqrt{\frac{5}{2}}\ \frac{b}{d}\right] /3 \qquad \ldots (4.4)$$

The centre lines of the webs were used as the layout of members for grillage analysis. This is an obvious choice for stages 1 and 2. At stage 3 the moments of inertia were easily calculated as I sections about the web centre lines, and the torsional constants were assumed to be concentrated along webs. Half of this value was taken for the outer girders.

As the model is symmetrical only half of it, (consisting of 72 joints), was considered for computer analysis. The loading for the symmetrical and anti-symmetrical cases was proportioned such that the superposition of the two sets of results gave the correct solution for the whole frame i.e. $+\frac{W}{2}+\frac{W}{2}$ and $\frac{W}{2}-\frac{W}{2}$. For the symmetrical case zero rotations were imposed along the centre line and zero vertical displacements for the anti-symmetrical case. The total time taken for the analysis of three loading cases was slightly less than $4\frac{1}{2}$ minutes. The case of a unit load was analysed at each loading position and these results scaled linearly to coincide with the exact load applied to the model. Bending moments were calculated at the centres of strain gauges by interpolating linearly between the terminal moments for the member.

The simply supported span technique was also used to derive bending moments and displacements for the model. The modified approach outlined in Chapter 1 was used. The values of average bending moment and deflection were obtained by placing an equal load at each joint along the centre of the deck and analysing this case with the computer program.

4.4.2) Discussion of Results.

Stage One.

The deflection profiles are shown in Figs.(4.3 - 4.6). Initially, section properties of the perspex grillage were used for computer analysis and the agreement between computer and model was not very good. The model appeared to be torsionally stiffer than predicted theoretically. The reason for this was thought to be the omission of the contribution to the torsional stiffness of the $\frac{1}{2}$" square bearing strips at the central supports. Because of the high steel/perspex modula ratio these strips have the effect of considerably increasing the torsional stiffness of the transverse girders over the central supports, and these effects were included in all subsequent calculations. Fig. (4.3) shows the marked improvement in agreement. The agreement between computer results and experimental values at the centre of the main span is very good, but the equivalent simply supported span technique does not give an accurate assessment of maximum deflections and, because these values are plotted from the line of contraflexure, they do not follow the actual deflected profile.

With the load applied in the end span the correlation between grillage and model was not as close. The model appeared to be more flexible, although this could have been caused by slight deflections in the $\frac{1}{2}$" dia. supporting rod. It was found at this stage that when loads were applied to the end span the far intermediate support tended to lift. This was remedied by placing weights along this line to re-establish contact with the roller. The distribution coefficients largely overestimate the maximum deflections in the side span.

The discrepancies between analytical and experimental results are probably due to the model being unable to exactly simulate the program support conditions. The end supports are assumed to rotate about the mid depth of the transverse girder. In reality the axis of rotation is at the centre of the ½" dia. steel rod.

Stage Two.

The model at this stage consisted of a system of interconnected tee-beams. The deflection profiles are plotted in Figs. (4.7 - 4.10) where there is good agreement between measured deflections and computer results. When the load is applied to the side spans there is an improvement in the correlation of results, compared with stage one. In this case the deck is torsionally stiffer and the effects of the steel support rod are probably less noticeable. As before the distribution coefficients do not predict the maximum deflections or deformed profiles. When the load is placed at the centre of the deck at X there is reversal of curvature at the edge of the deck, shown clearly in Figs. (4.8 and 4.10). This is probably the Poisson's ratio effect which cannot be reflected in the computer results as the tee-beams are assumed to act independently and the overall continuity of the top plate is not considered.

The bending moments in the longitudinal direction are shown in Figs. (4.11 - 4.18). There is generally good correlation between the grillage analysis and experimental values. No definite trend can be seen in these results as the computer both overestimates and underestimates the experimental values. If the correctness of the computer results are accepted it appears that the discrepancies are due to normal variations in experimental readings. The distribution coefficients however, consistently overestimate the values of maximum bending moments by as much as 100% in Fig. (4.15). With the load placed at the mid-span point X there is almost a complete redistribution of negative moment over the internal supports. The computer analysis is able to predict this redistribution whereas by applying the mid span coefficients at the support the simply supported span technique is unable to do this, as shown in Fig. (4.14).

Transverse strains were also measured and from these the bending moments in the diaphragms were calculated using the full section as being effective at each point. This was an invalid assumption because in the vicinity of the concentrated load the effective width will be very much smaller. Figs. (4.20 and 4.22) show that a much larger bending moment than predicted by the computer, is obtained when measured strain is assumed to act across the full section. Figs. (4.19 and 4.21) show that when the load is applied to a longitudinal girder, thus avoiding local stress concentrations, a much better agreement is obtained. The grillage analogy can predict the average stress occurring in a member but is unable to indicate local stress concentrations when loads are placed between the main girders. The simply supported span technique does not make provisions for calculating the distribution of transverse bending moment in the side spans.

Stage Three.

The addition of the bottom plates to the model of stage two produced a cellular deck varying in longitudinal depth. The deflection profiles are plotted in Figs. (4.23 - 4.26). In the initial interpretation of Equation (4.3) the area and the perimeter of the hole were used to determine the torsion constants as recommended by Morice and Little (9). As can be seen from Fig. (4.24) this led to a underestimate of the torsional strength of the deck. The grillage was then re-analysed assuming the median line to lie on the centre lines of the flanges and web, which increased the torsional stiffness quite considerably and gave a much better correlation with the model as shown in Fig. (4.24). After a recent investigation of this problem Acton (28) arrived a similar conclusions. The agreement of displacements in the side span is again not very good. This is probably due to the nature of the supports.

Longitudinal bending moments are shown in Figs. (4.27 - 4.34) where there is generally good agreement between experimental and computer values. Fig. (4.27) shows again the improvement in agreement by using the modified torsion constants. The same re-distribution of hogging moment seen in stage two is again evident in Fig. (4.30). The deflection profile for a load placed at X shows a slight reverse in curvature at the edge of the deck in Fig. (4.23). This effect is also reflected in the longitudinal moments in Figs. (4.28) and (4.32) and shows again quite clearly the Poisson's ratio effect present in continuous plated structures. When using the simply supported span technique the bending moments are again overestimated.

Transverse bending moments are plotted in Figs. (4.35 - 4.38) where it can be seen that the same localised stress concentrations occur when the load is placed at point X.

4.5.) Conclusions.

By considering the sets of results obtained from the three different types of model the following conclusions have been drawn:

(1) The computer program for analysing grid framework structures composed of members with varying section properties gives accurate results for thus type of structure, provided member properties are specified at sufficient stations to enable accurate values of stiffness and restraint factors to be obtained.

(2) The grillage is also a valid mathematical model for analysing grillages composite with top slab and fully torsional cellular structures. The analysis is able to predict the maximum values of bending moment and deflection , but is unable to indicate local effects viz. localised stress concentrations and Poisson's ratio effects. The latter phenomenon is likely to be particularly noticeable in a perspex structure which has a high Poisson's ratio $\mu \simeq 0.35$ compared with concrete where $\mu \simeq 0.15$.

(3) The equivalent simply supported span technique does not give acceptable results. In some cases the values of maximum deflections were underestimated and in all cases the values of bending moments were overestimated by as much as 100%. The recommended increase of 10% to maximum value is therefore either insufficient or unnecessary. The method assumes that the simply supported span rests

on straight unyielding supports at right angles to the span. Fig. (4.37) shows this assumption to be untrue especially for loads placed at the edge of the deck, an effect first observed by Sawko (29). The use of this method would lead to a design which was 'on the safe side' but also quite uneconomical.

(4) When interpreting the formula for torsion constants of cellular sections the median line should be taken to lie on the centre lines of the webs and flanges.

(5) The Cement and Concrete Association recommendations that the full slab width between web centre lines should be considered effective, gave good agreement for both moments and deflections and is therefore valid.

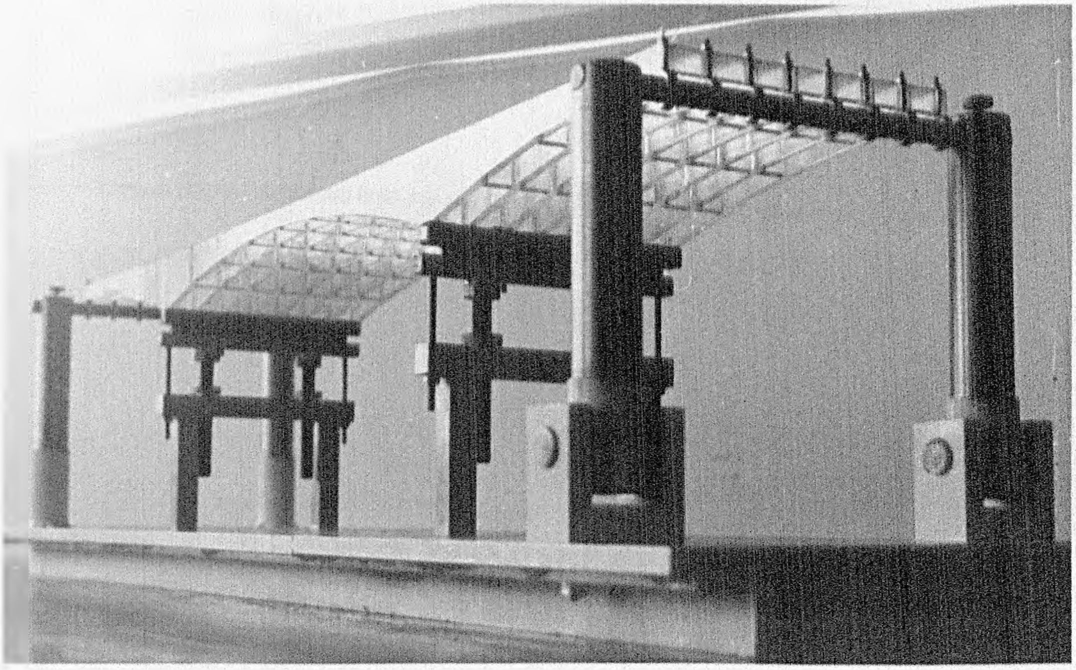


Plate 1. Perspex Model Stage 1.

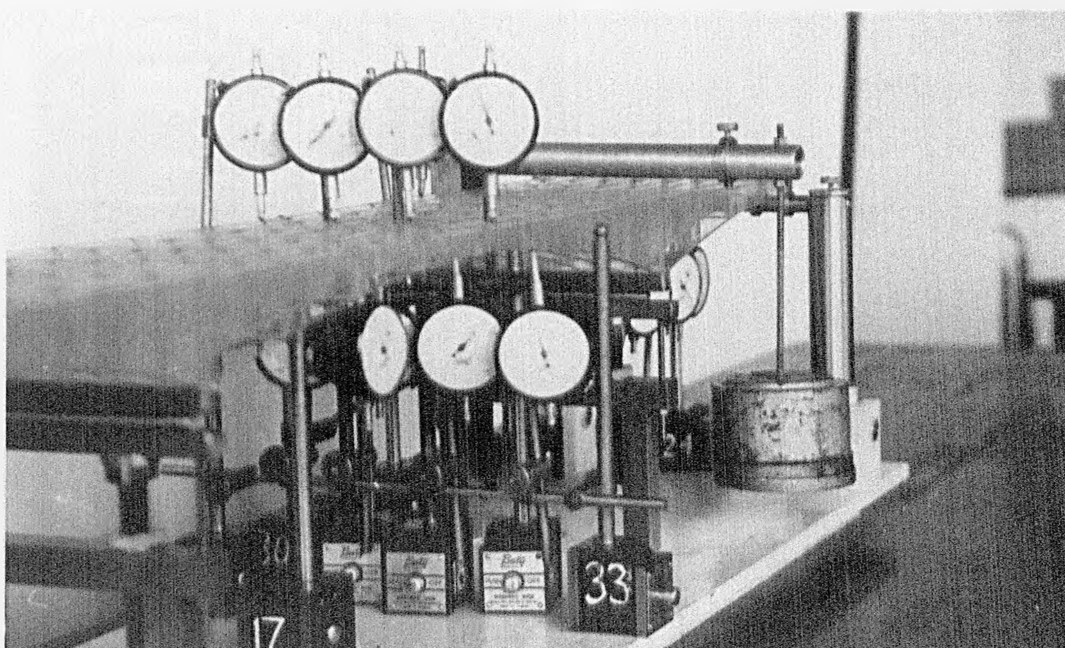


Plate 2. Method of Testing Stage 1.

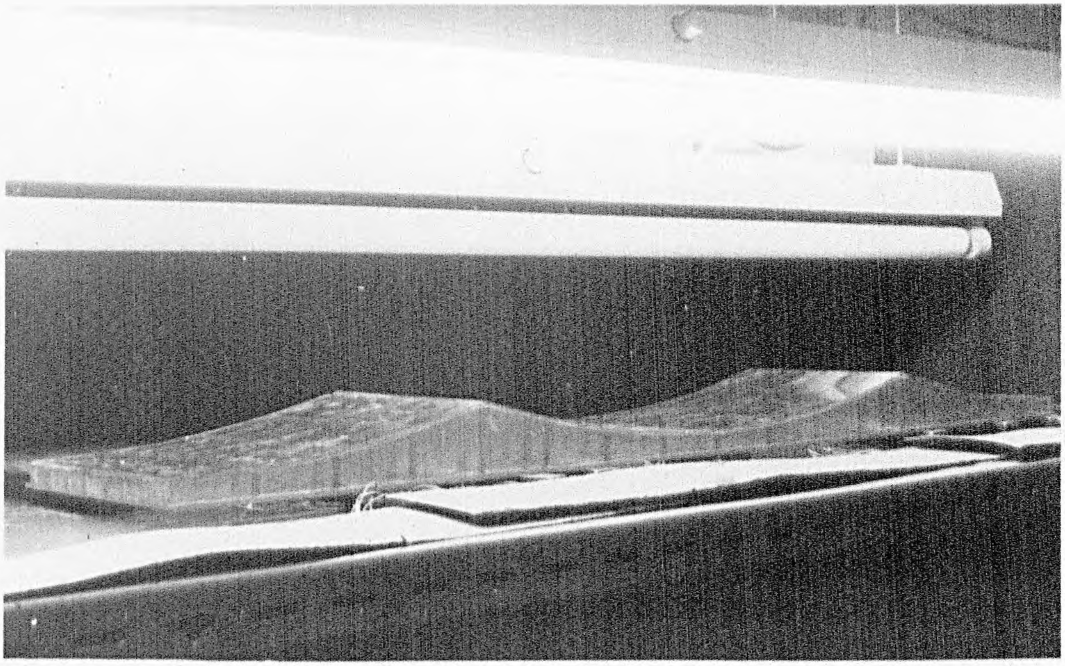


Plate 3. Light Polymerisation of Model.

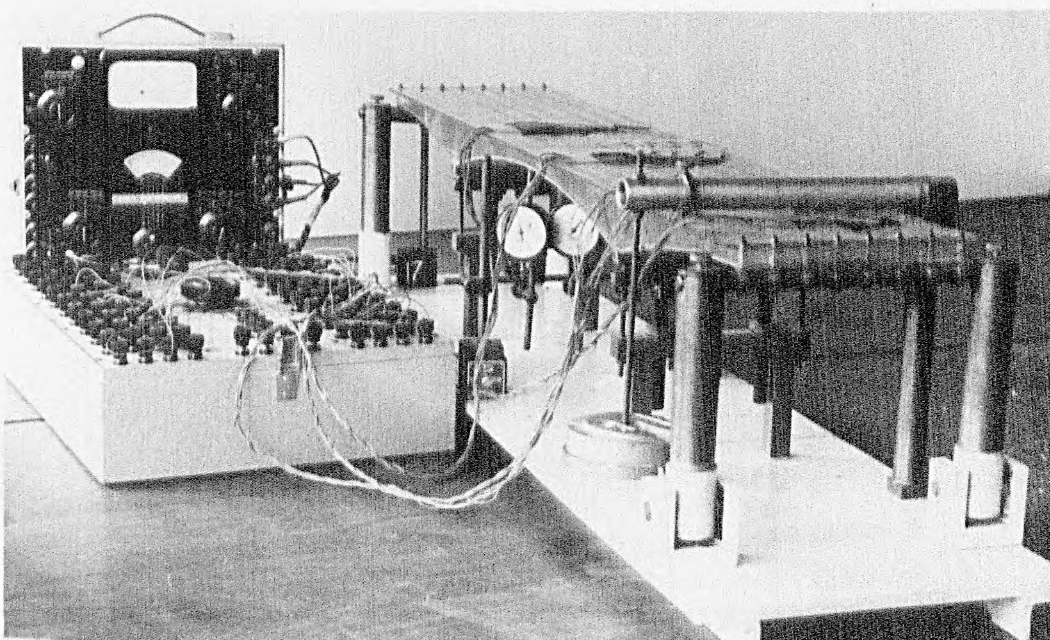


Plate 4. Method of Testing Stage 3.

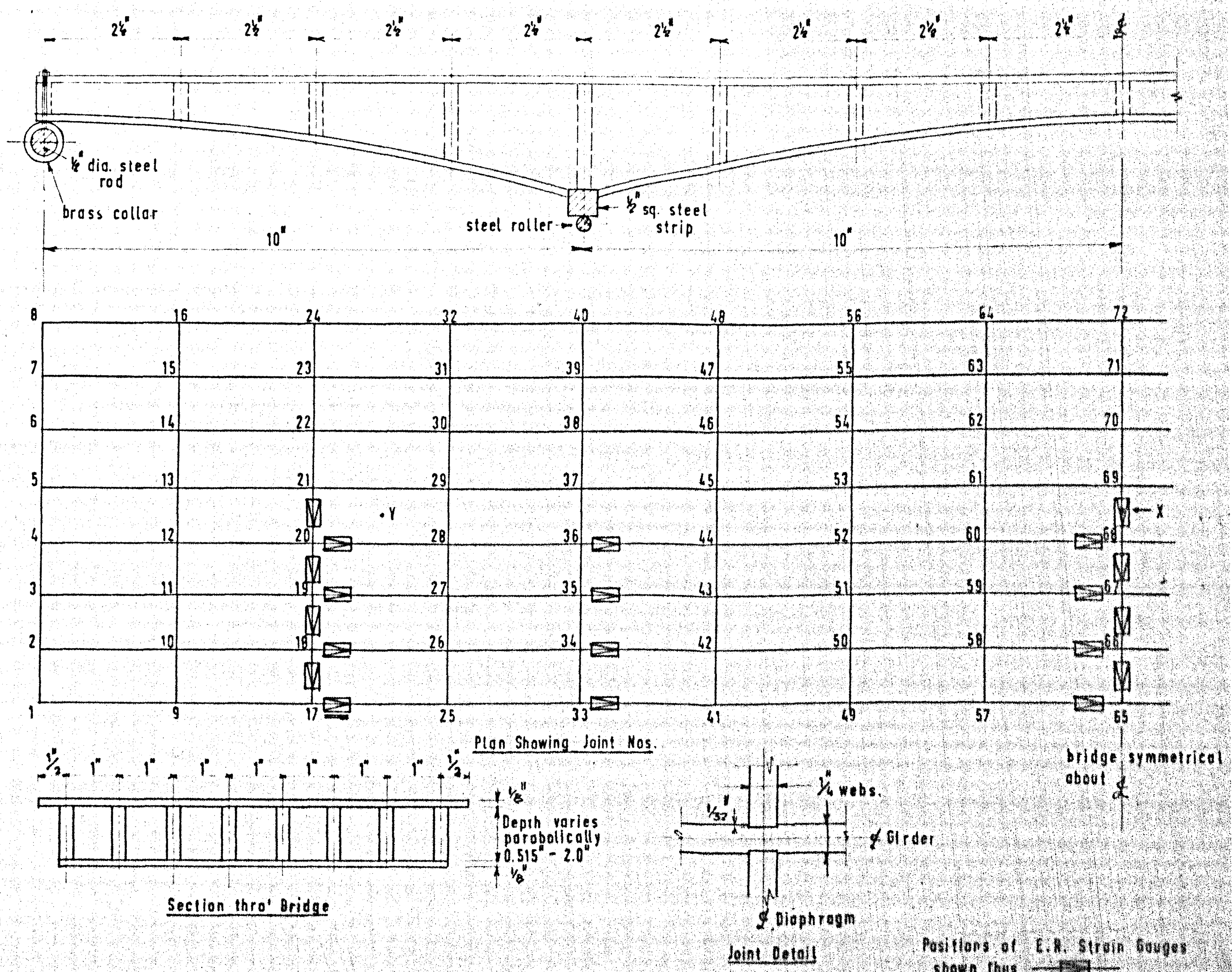
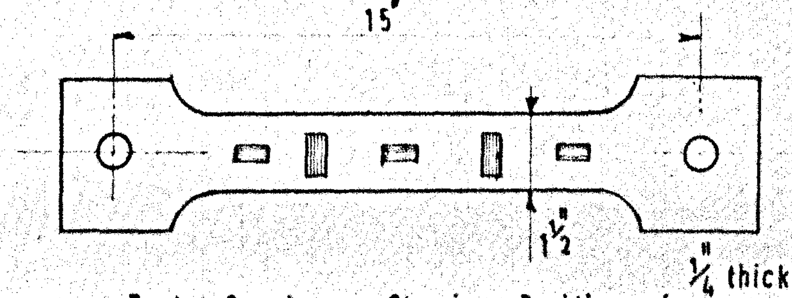


Fig. 4.1 DETAILS OF MODEL PERSPEX BRIDGE.



Test Specimen Showing Position of E.R. Strain Gauges.

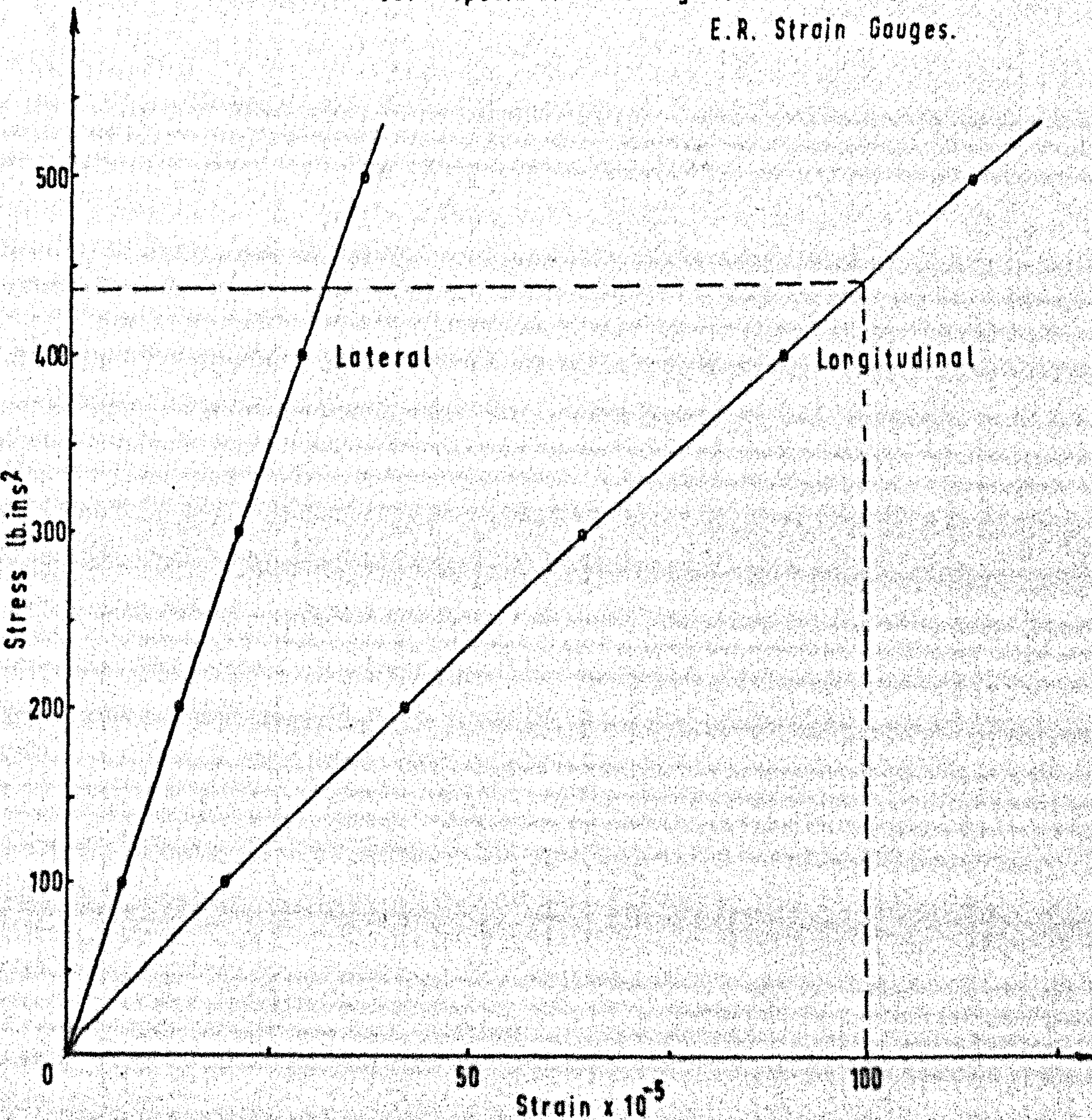
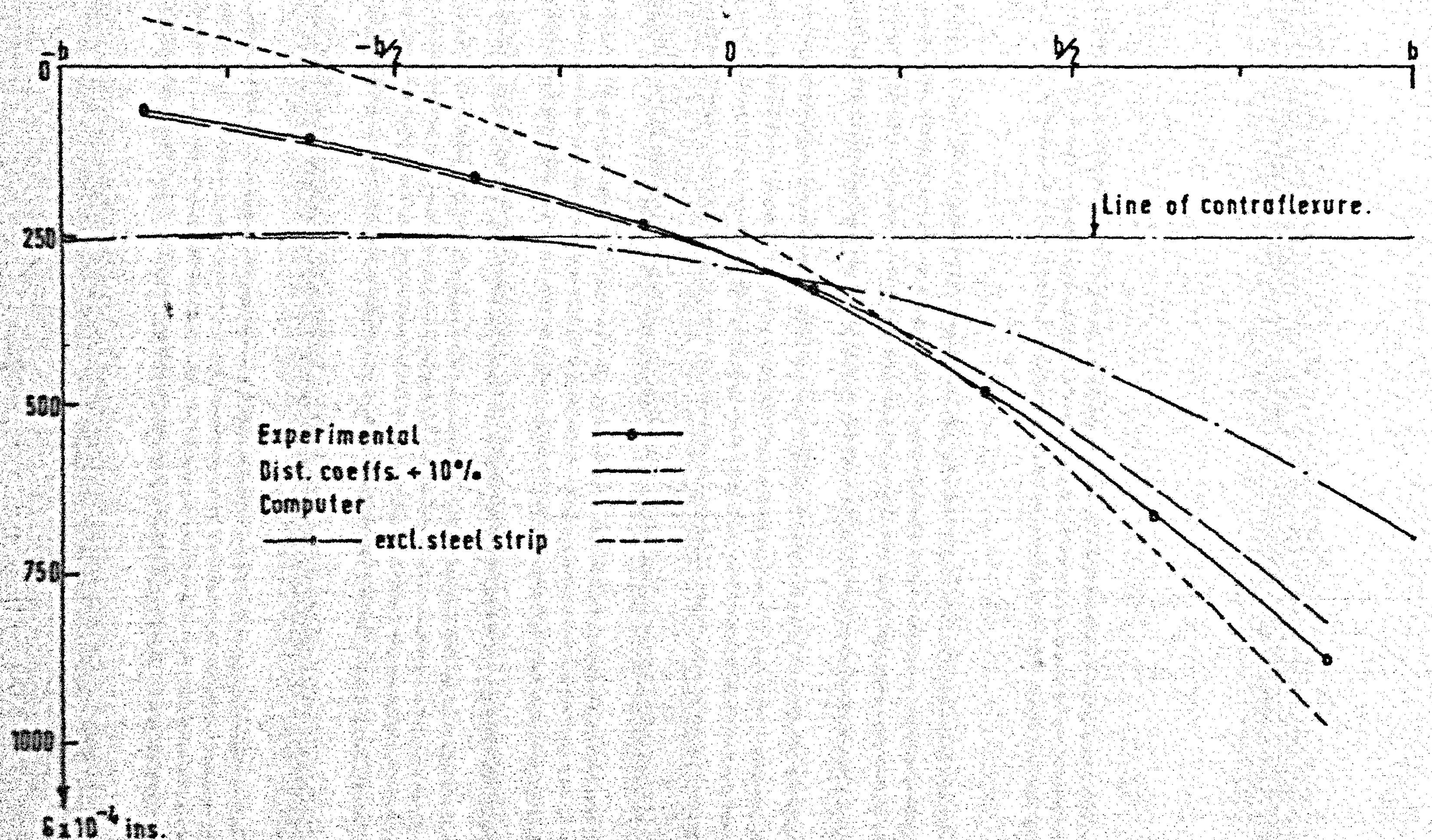


Fig. 4.2 Stress Strain Characteristics of Perspex.



Stage 1.

FIG. 4.3 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 65-72
23·5 lbs. at 72

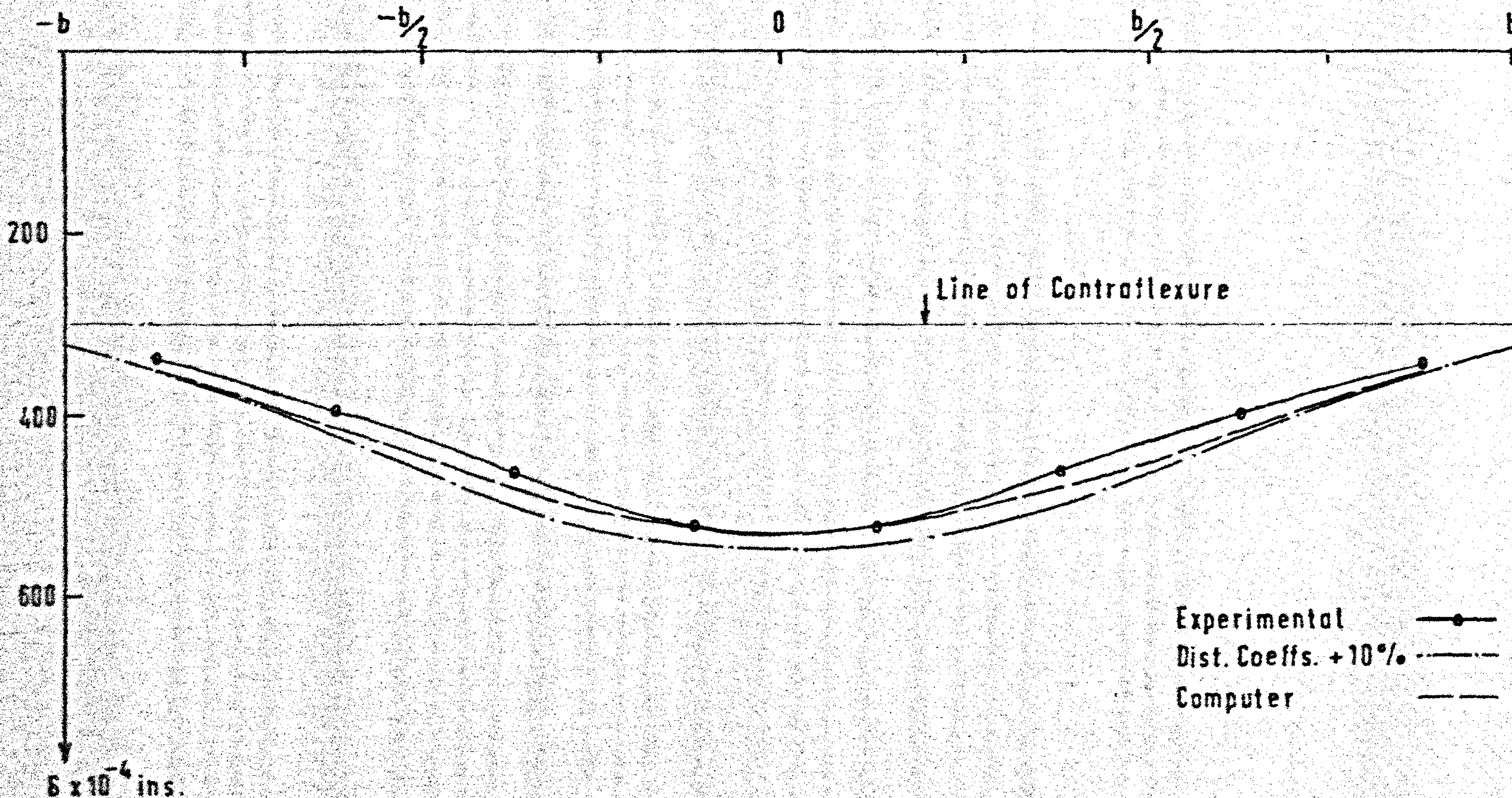


FIG. 4.4 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 65-72
30 lbs. at X

Stage 1.

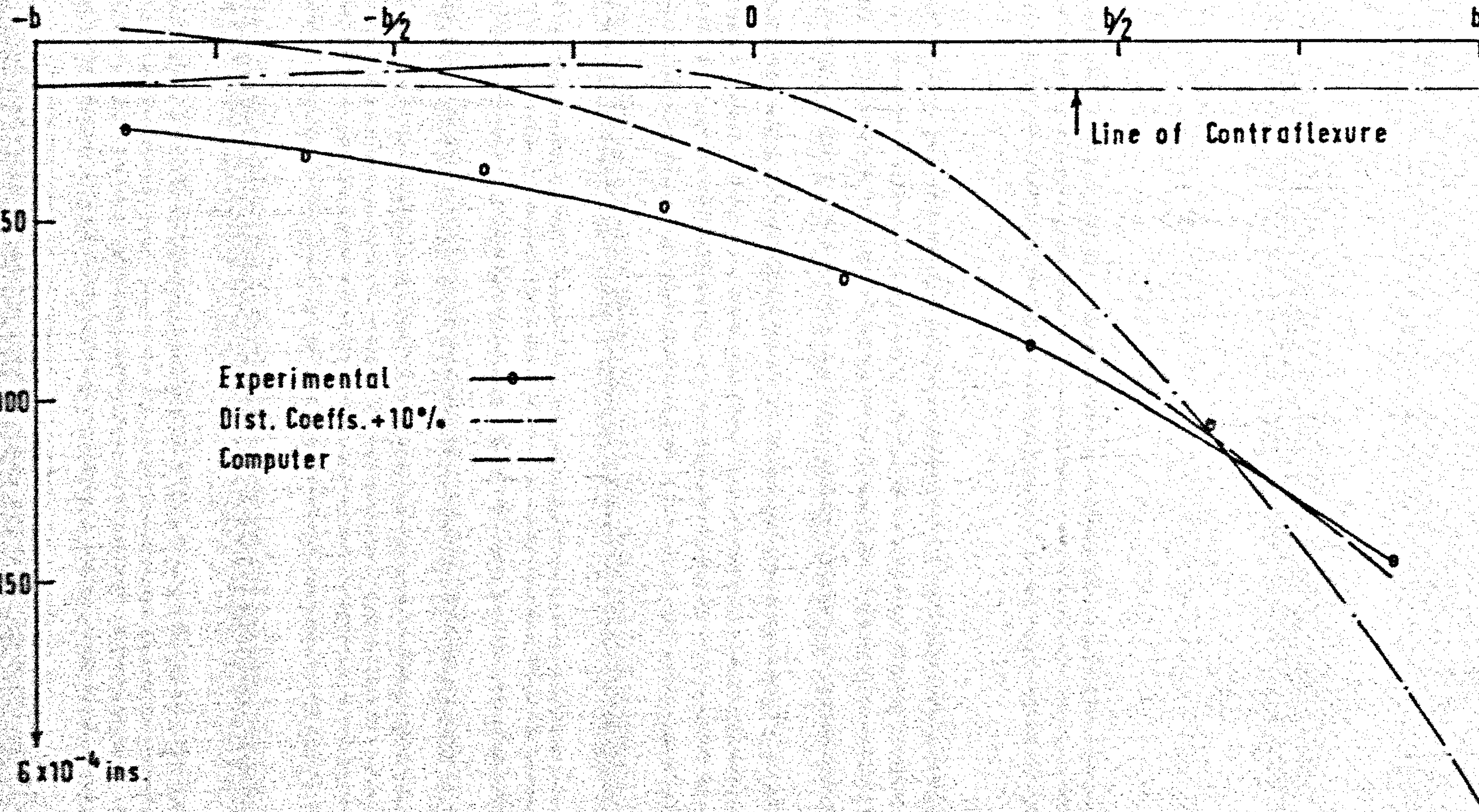


FIG. 4.5 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHAGM 17-24

17·6 lbs. at 24

Stage 1

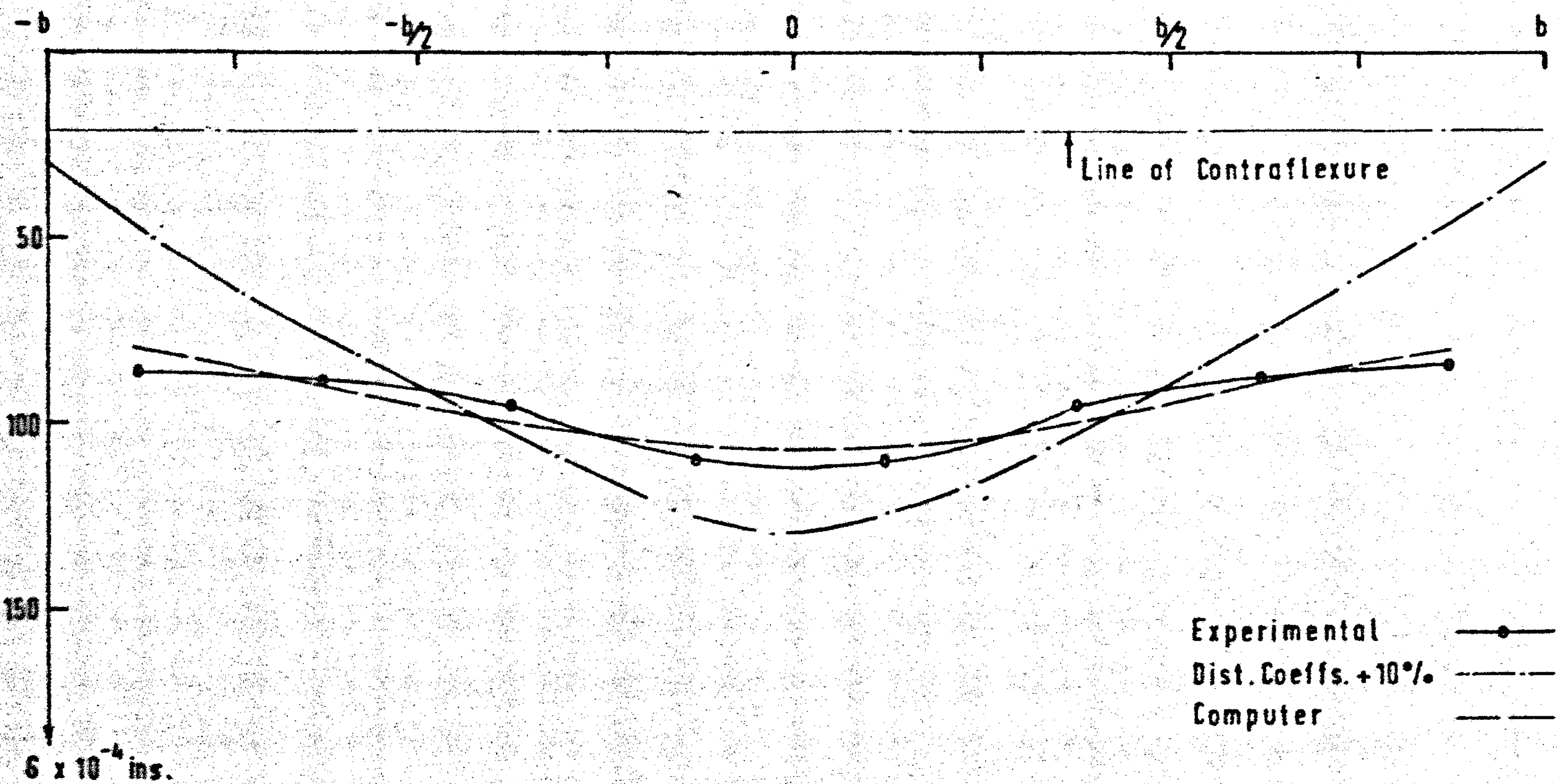


FIG. 4.6 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAM 17-24

30 lbs. at Y.

Stage 1

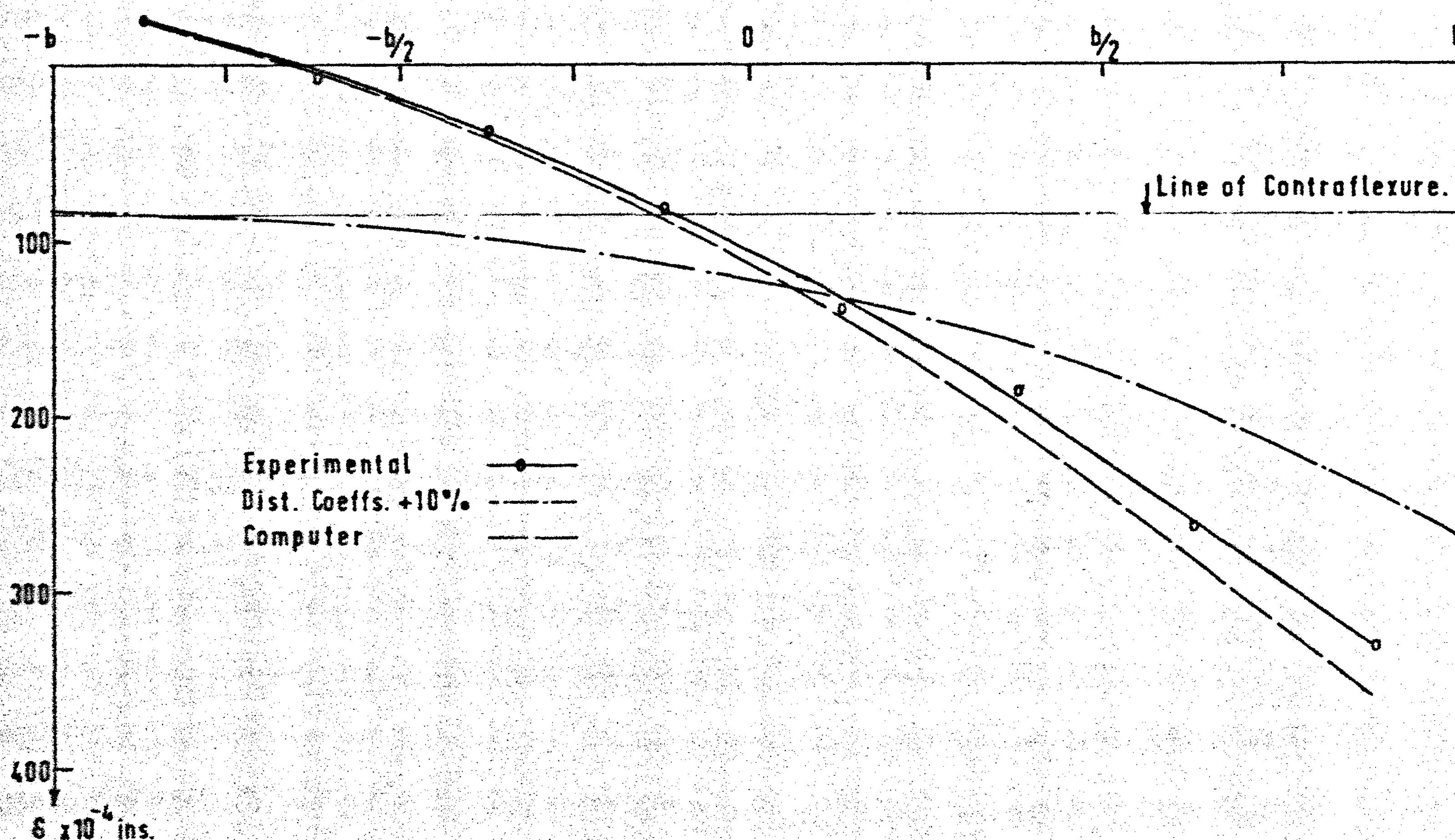


FIG. 4.7 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 65-72

22·8 lbs. at 72

Stage 2

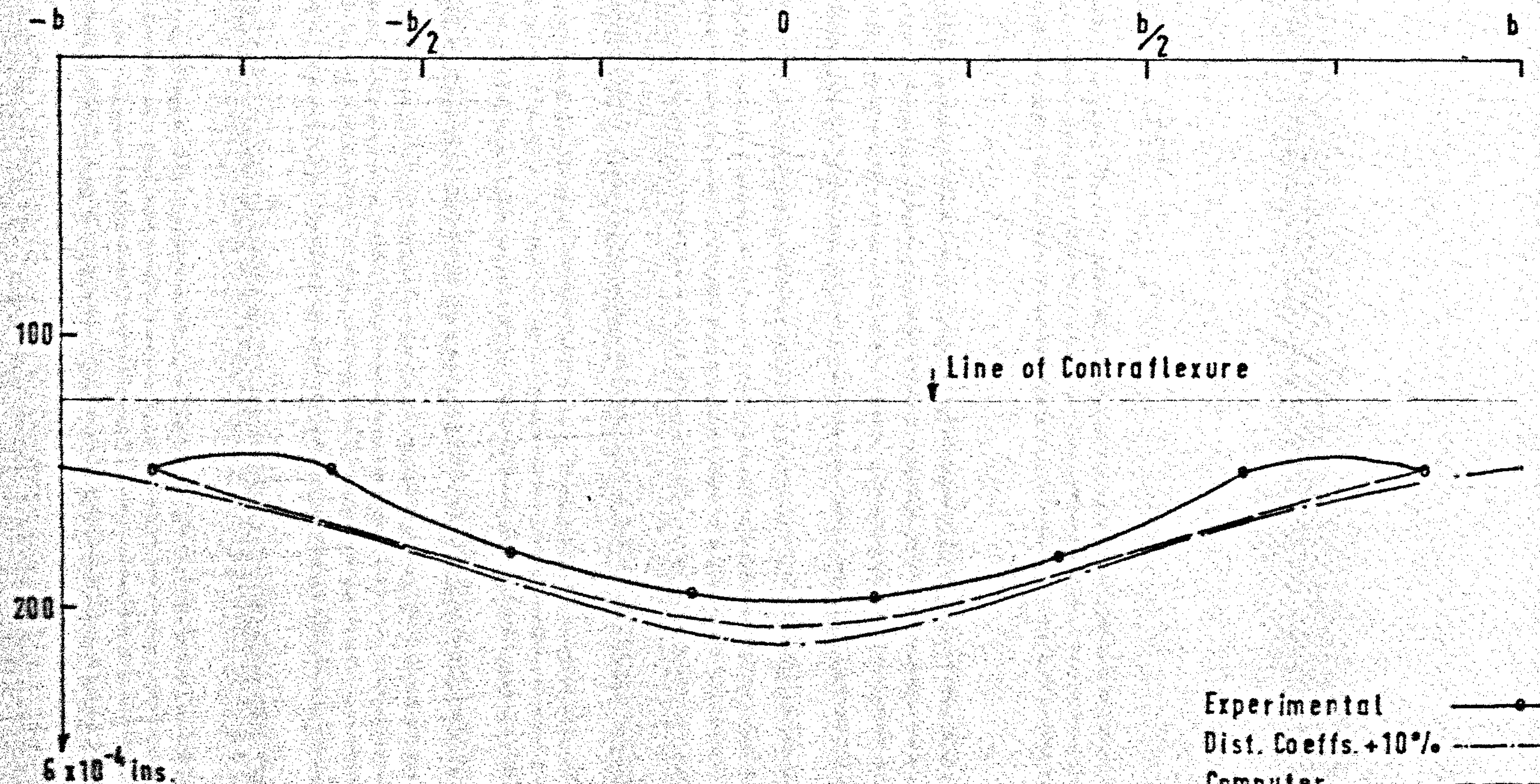


FIG. 4.8 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 65-72
30 lbs. at X

Stage 2

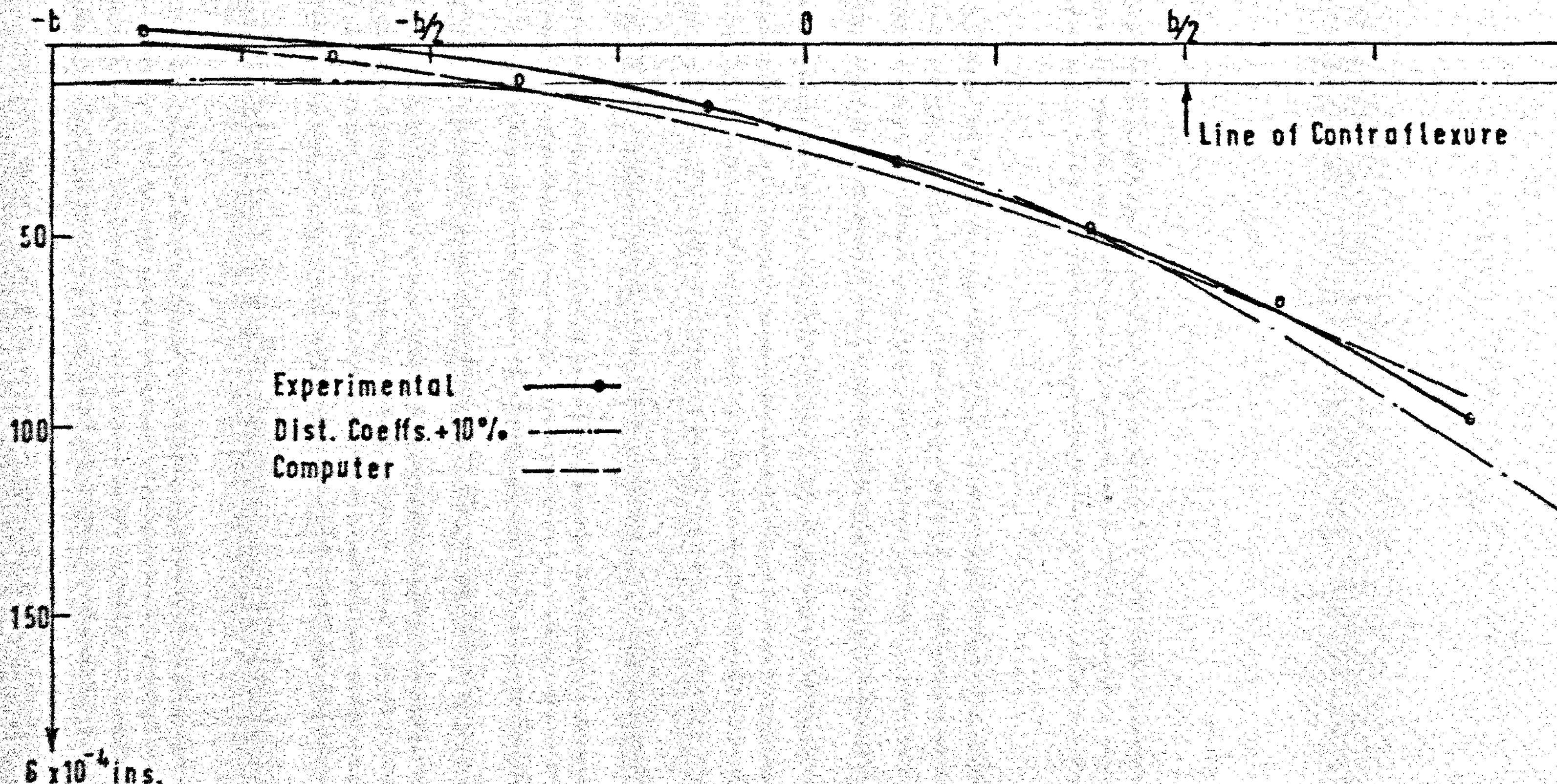


FIG. 4.9 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHAGM 17-24

22·5 lbs at 24

Stage 2

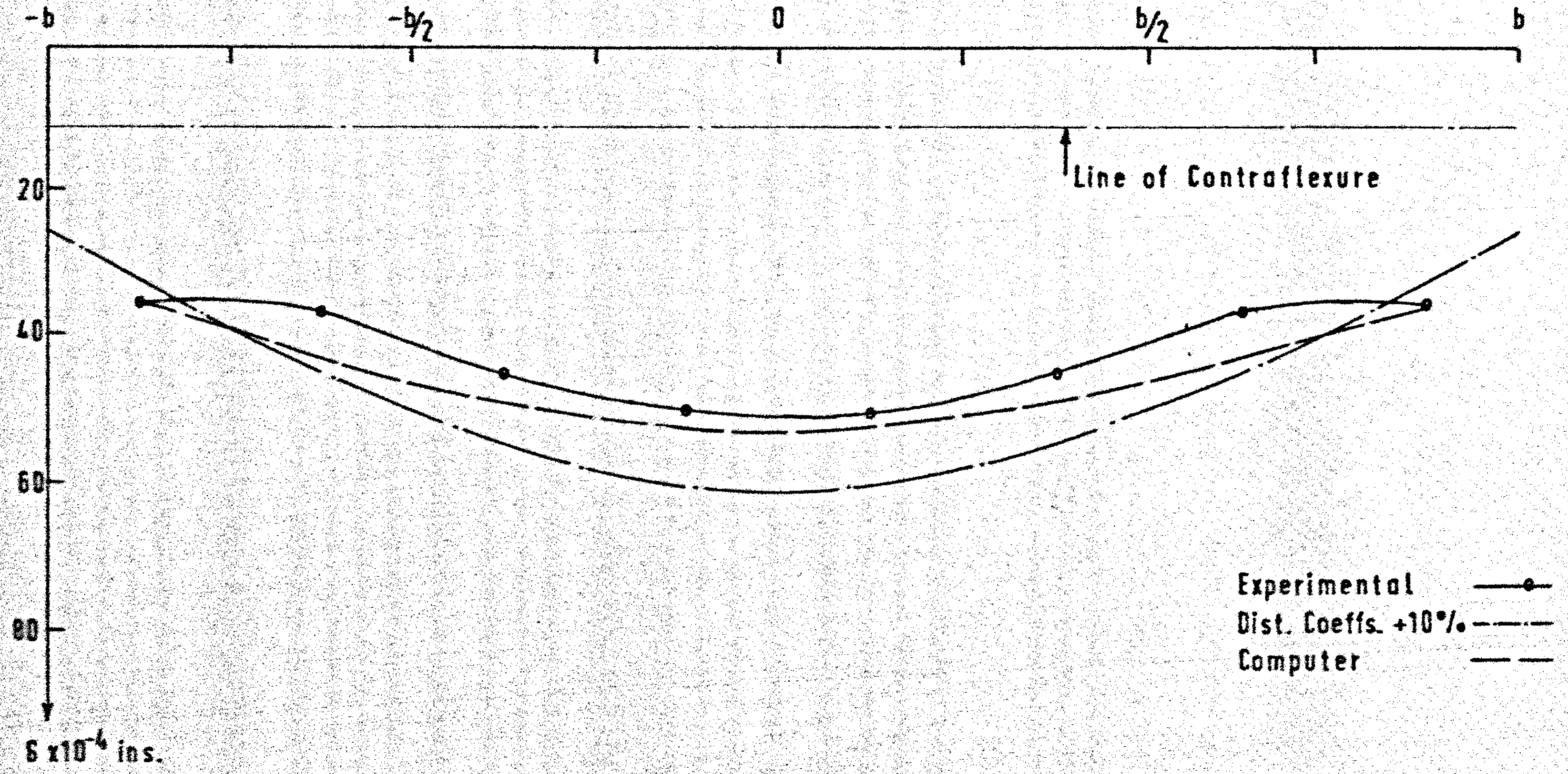


FIG. 4.10 TRANSERSE DEFLEXION PROFILE ALONG DIAPHAGM 17-24
30 lbs. at Y

Stage 2

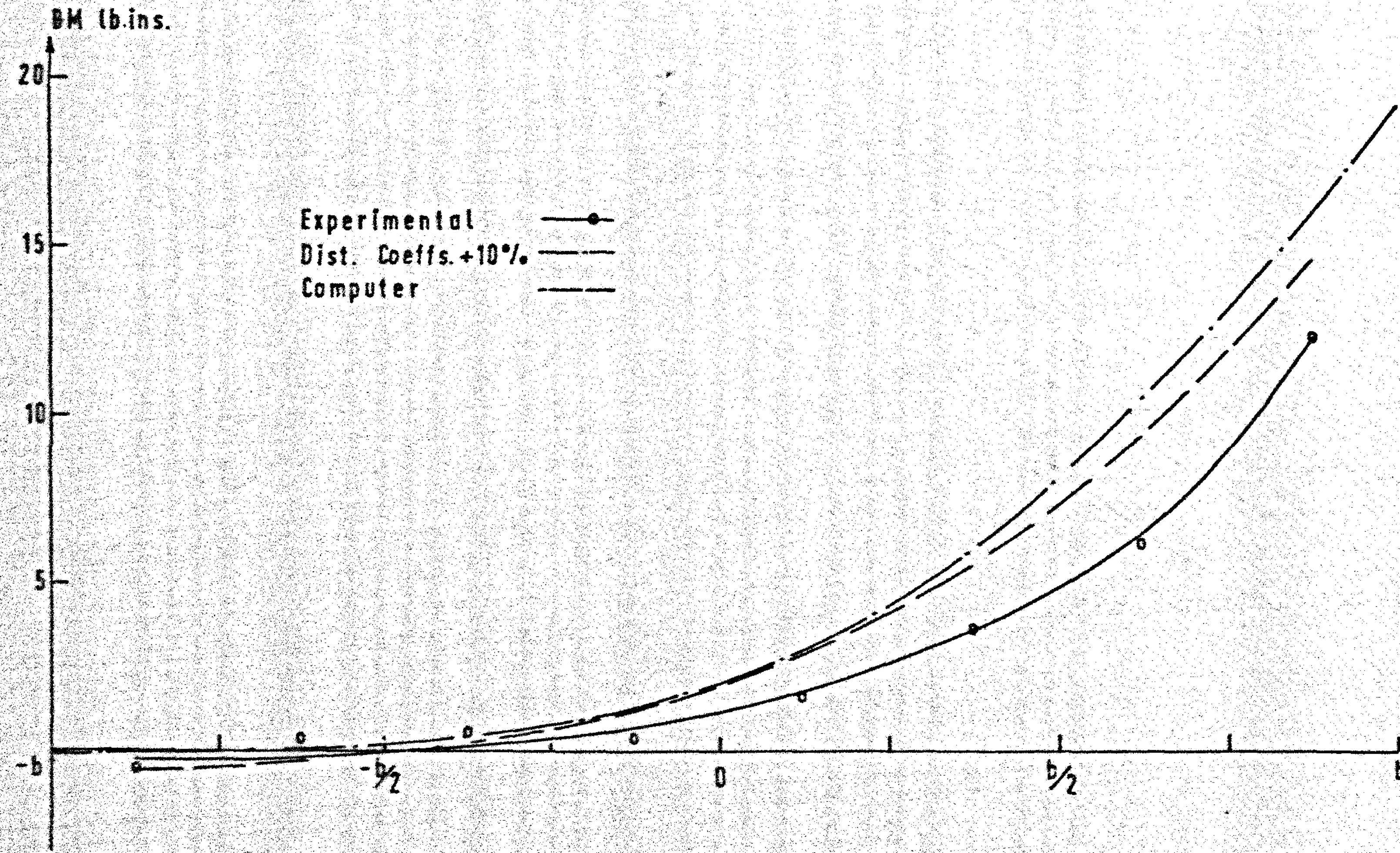


FIG. 4.11 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 65 72
30 lbs. at 72

Stage 2

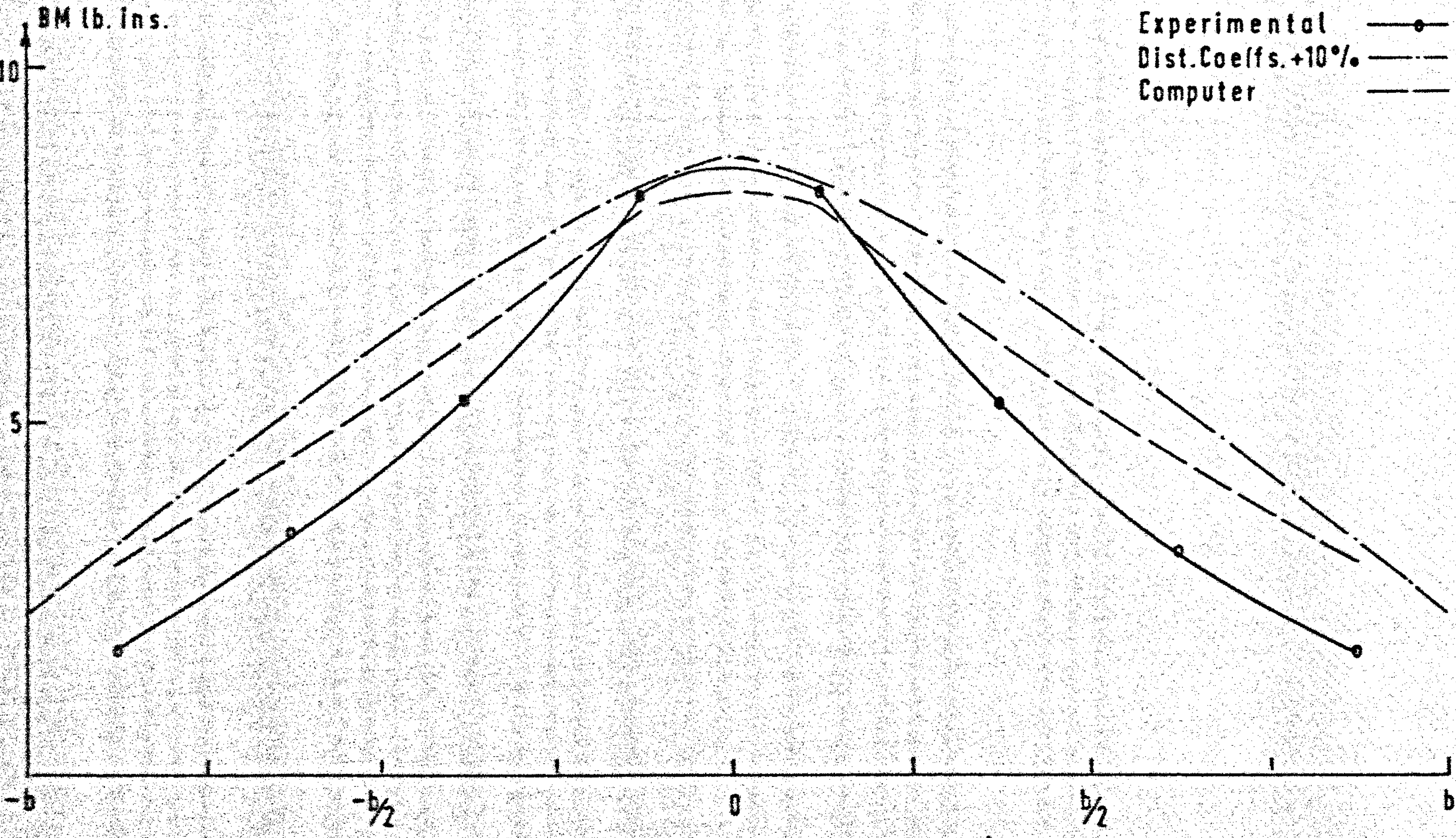


FIG. 4.12 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 65-72

30lbs. at X

Stage 2

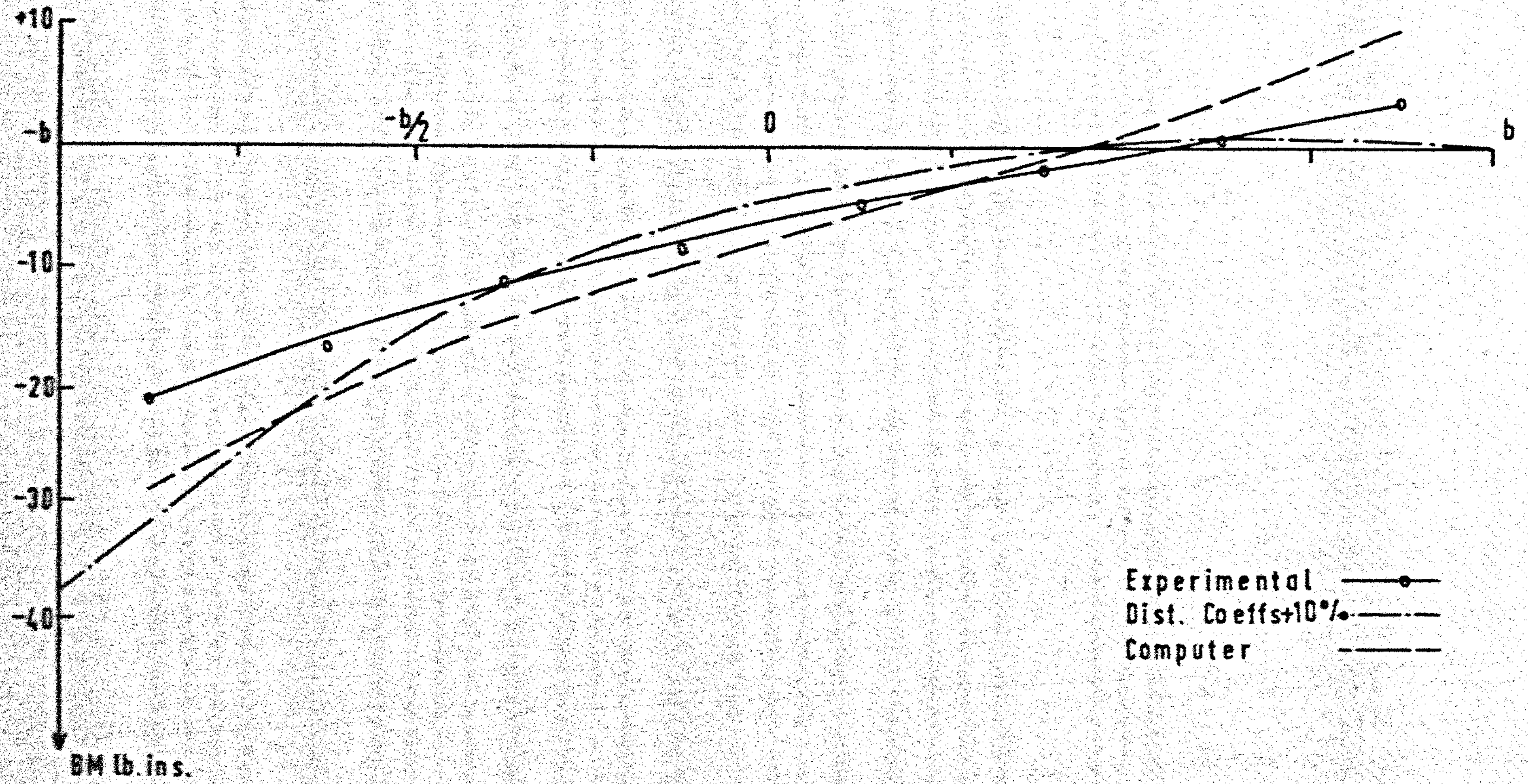


FIG. 4.13 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40
23·6 lb at X

Stage 2

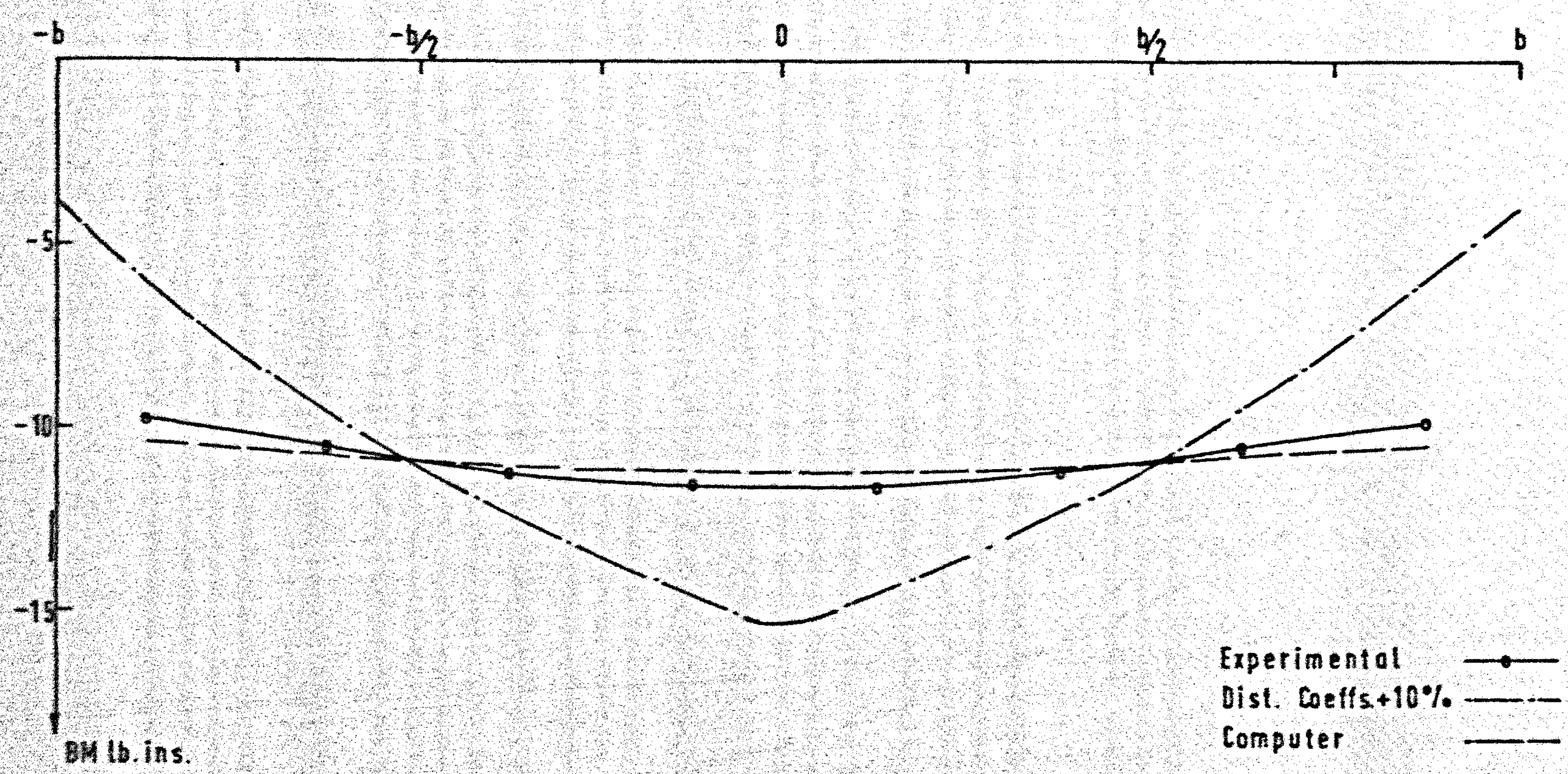


FIG. 4.14 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40

30 lbs. at X

Stage 2

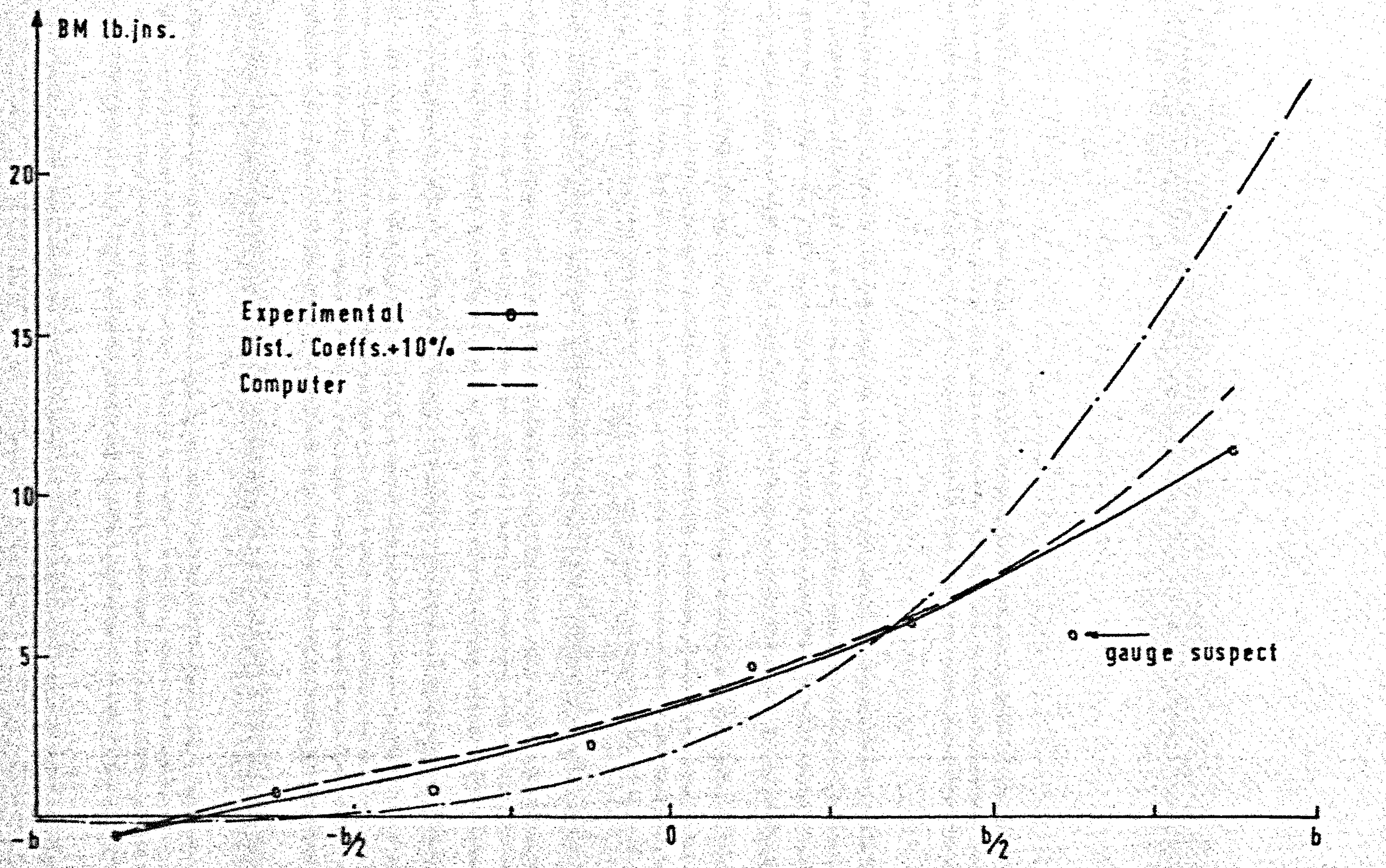


FIG. 4.15 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 17-24

22·8 lbs. at 24

Stage 2

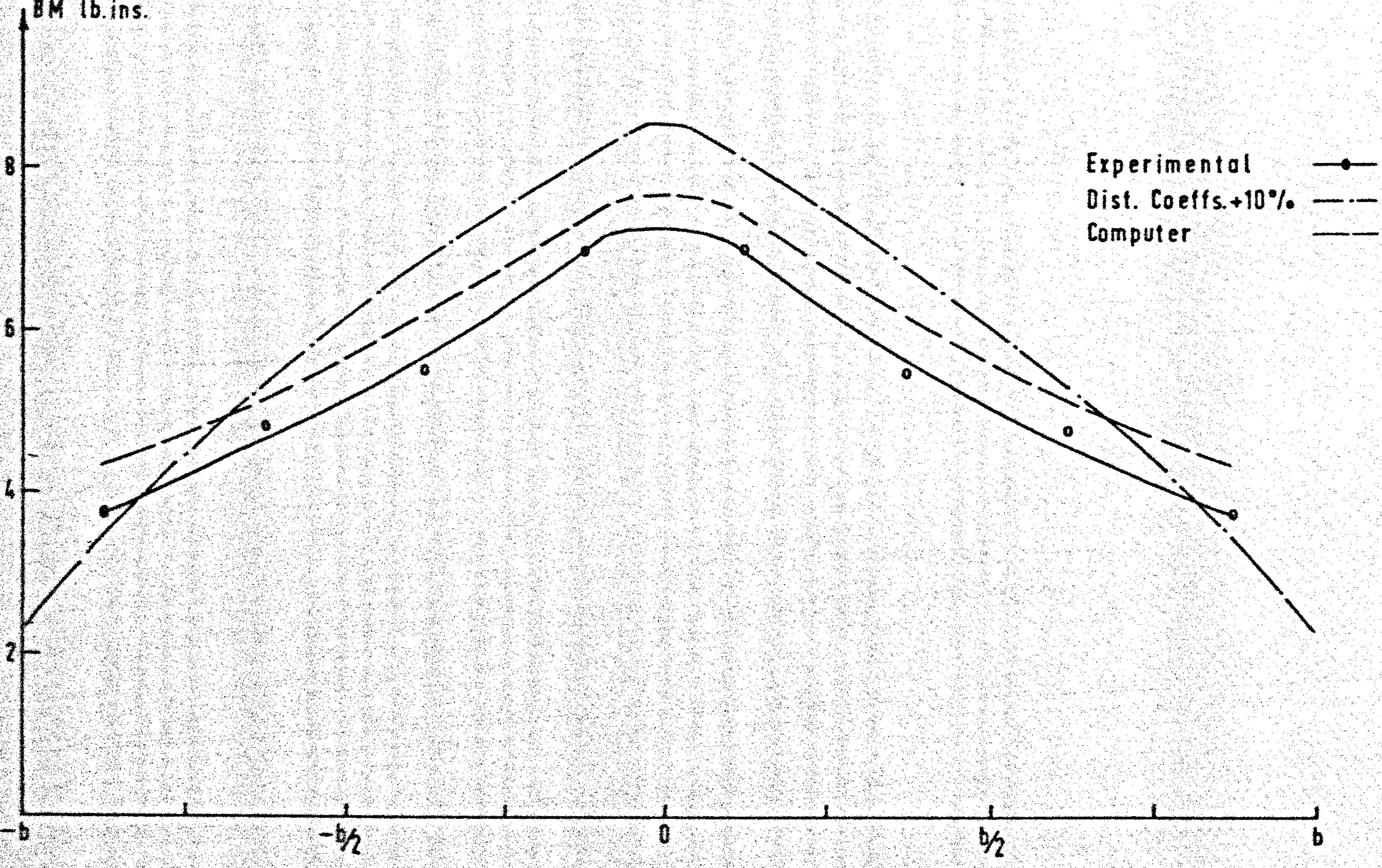


FIG. 4.16 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 17-24

30 lbs. at Y

Stage 2

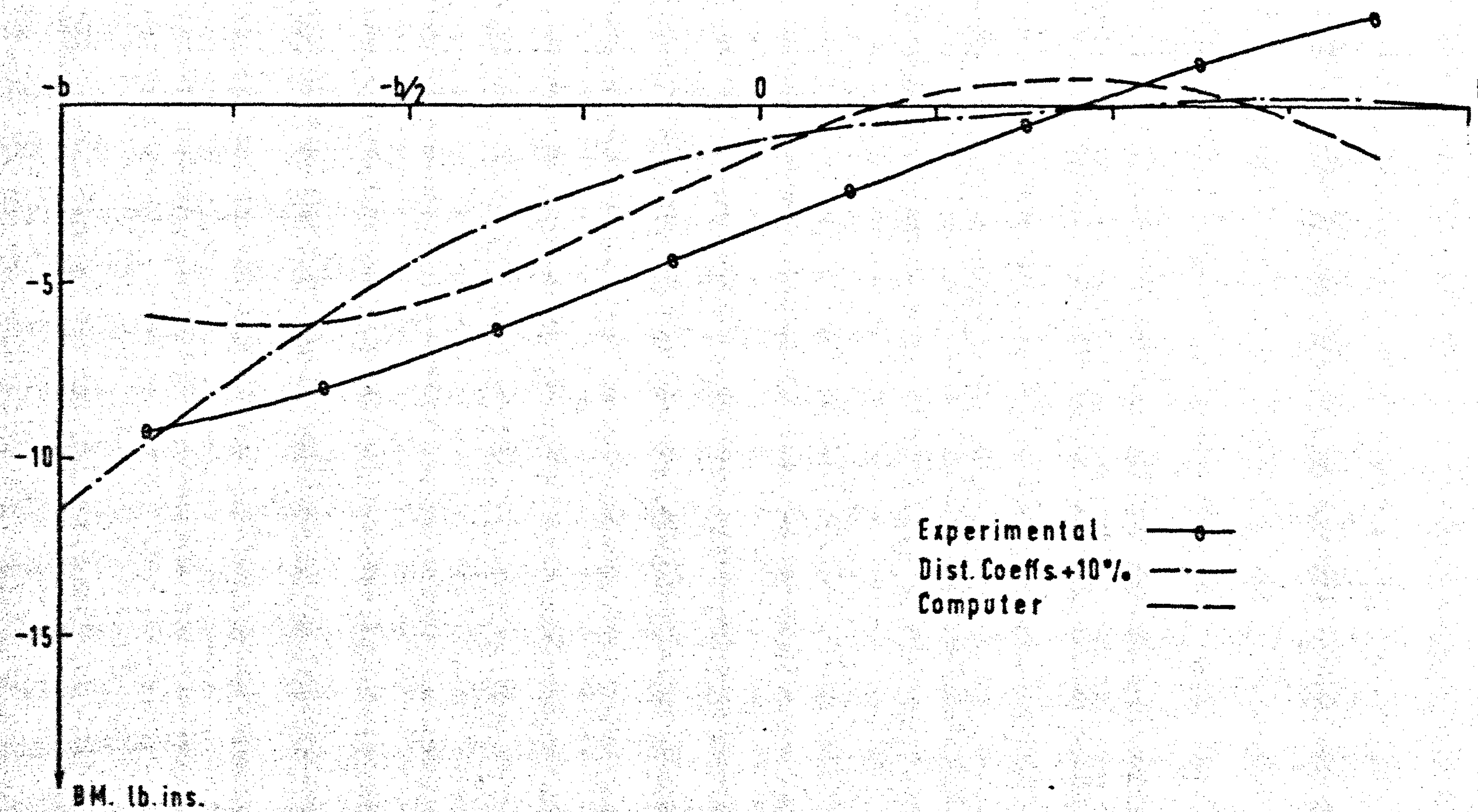


FIG. 4.17 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40
22·8 lbs. at 24

Stage 2

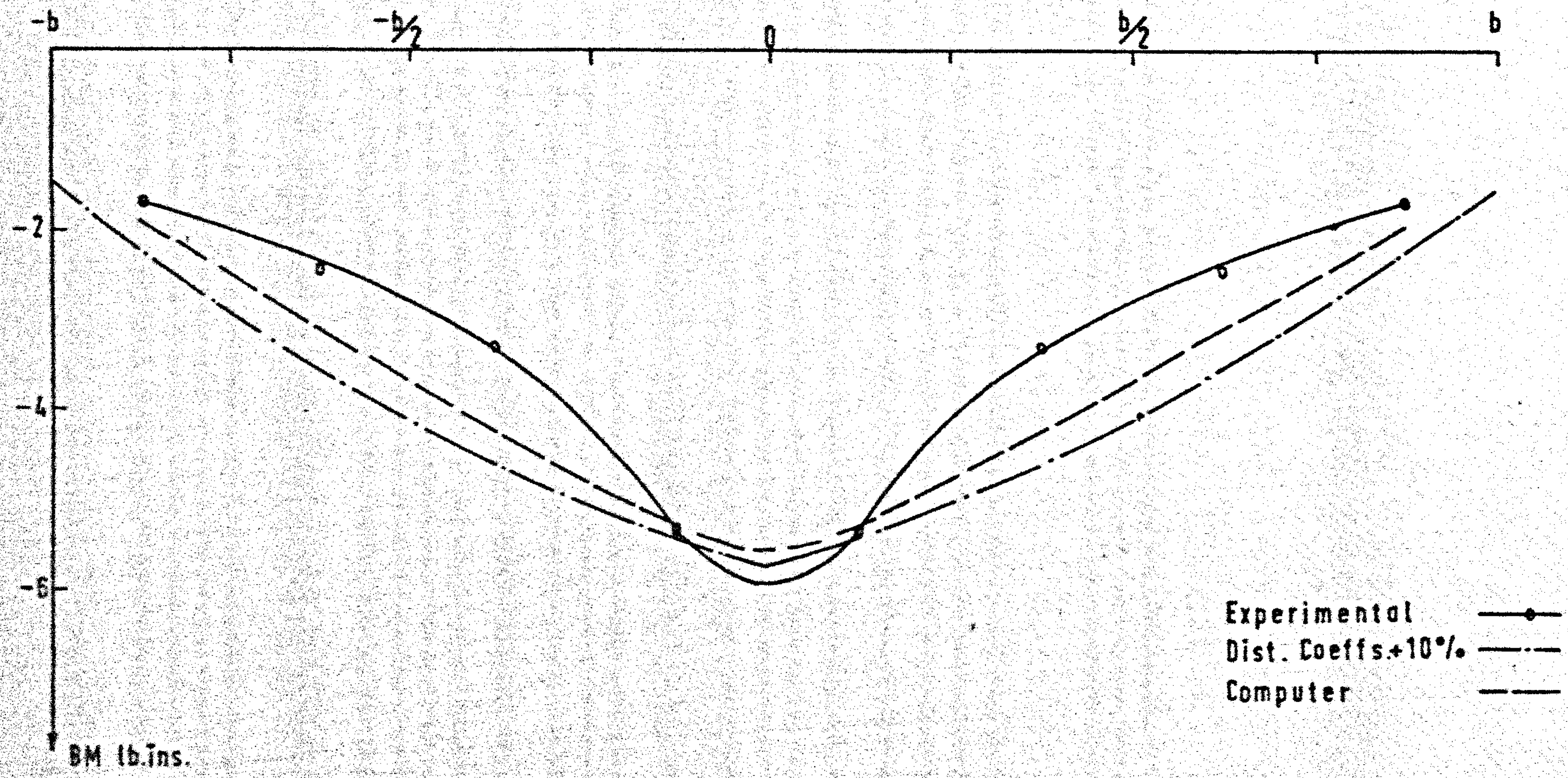


FIG. 4.18 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40

30 lbs. at Y

Stage 2

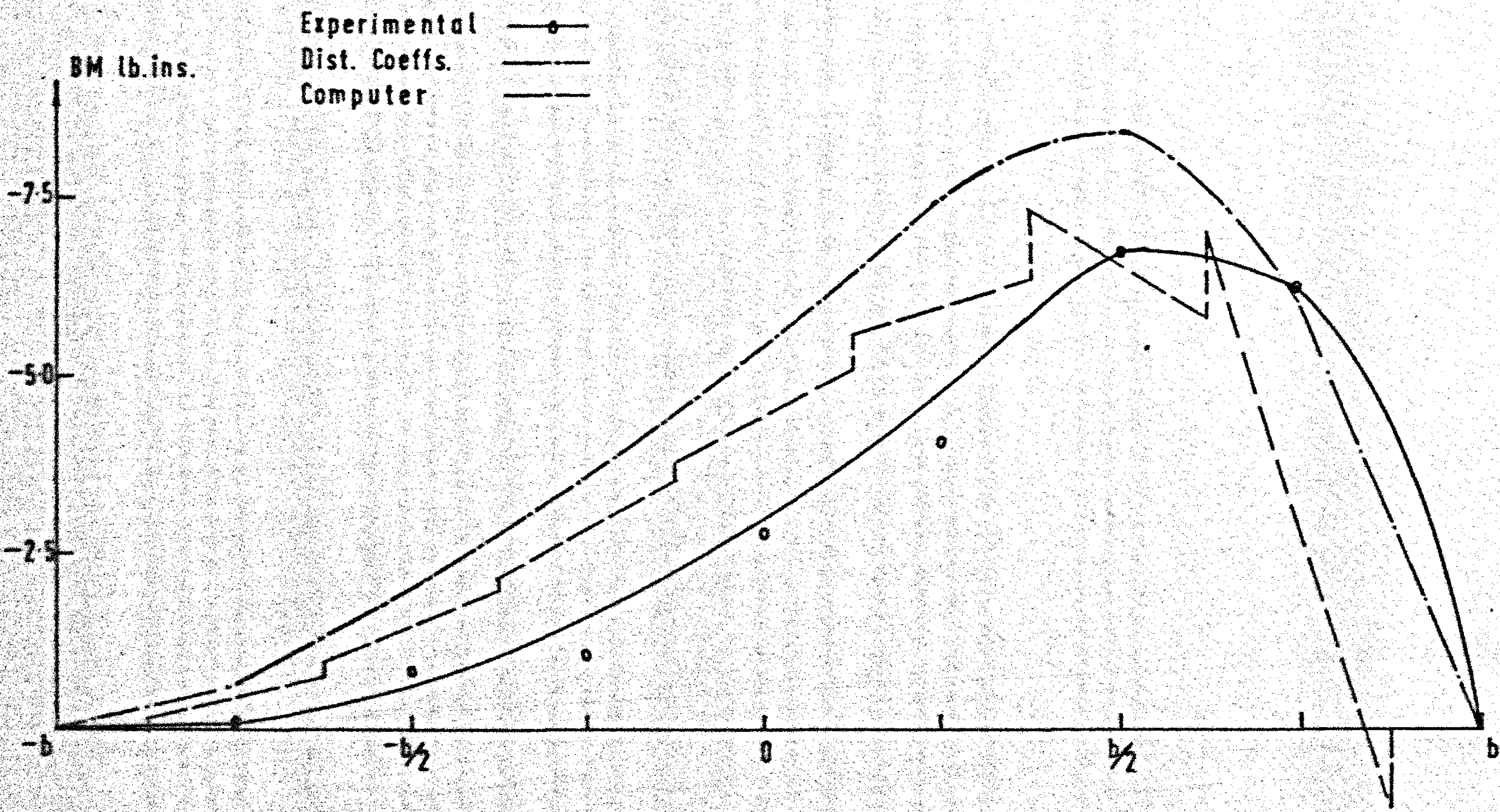


FIG. 4.19 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 65-72

23·6 lbs at 72.

Stage 2

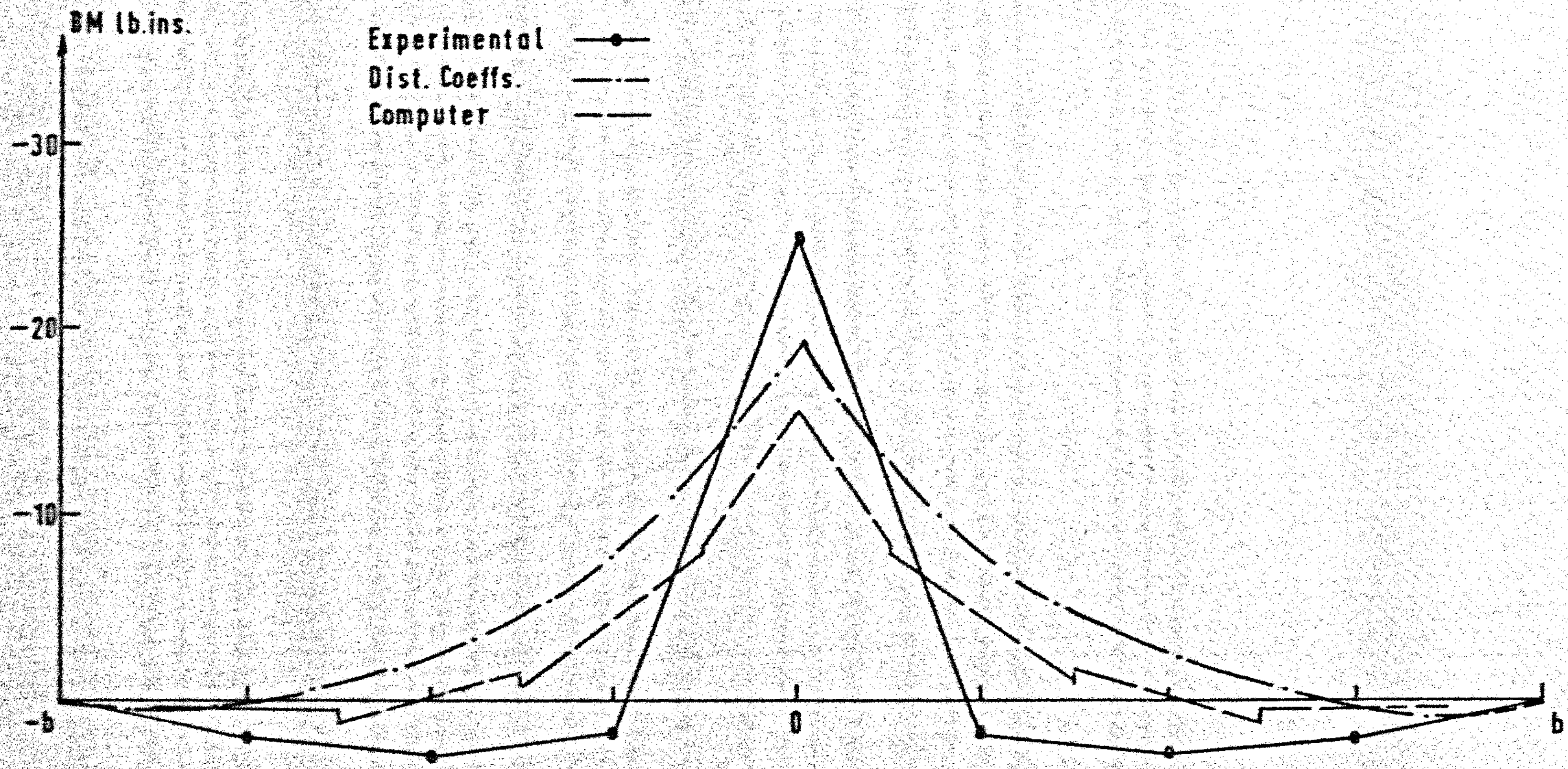


FIG. 4.20 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 65-72

30 lbs. at X.

Stage 2

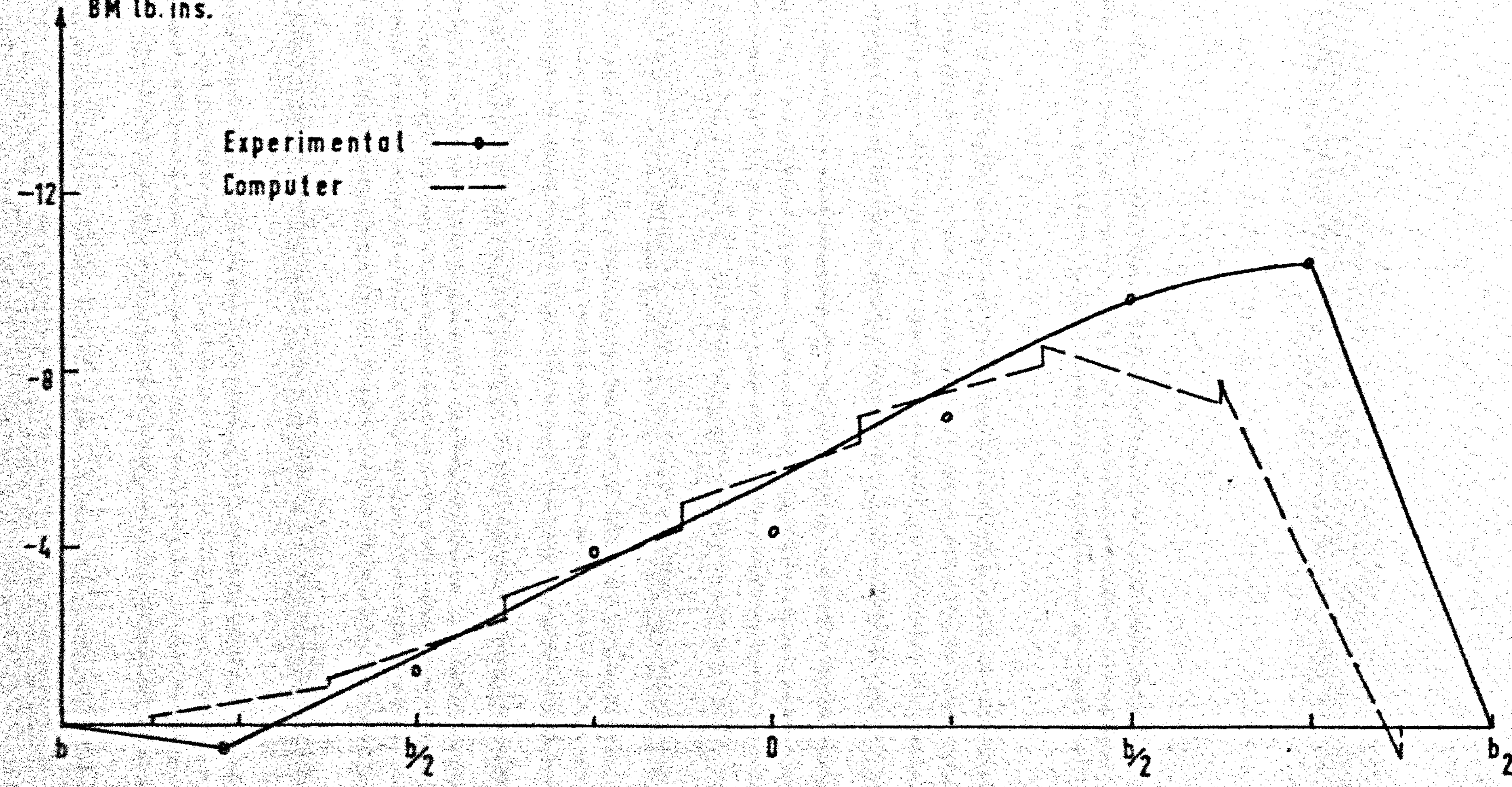


FIG. 4.21 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 17-24

22·8 lb. at 24

Stage 2

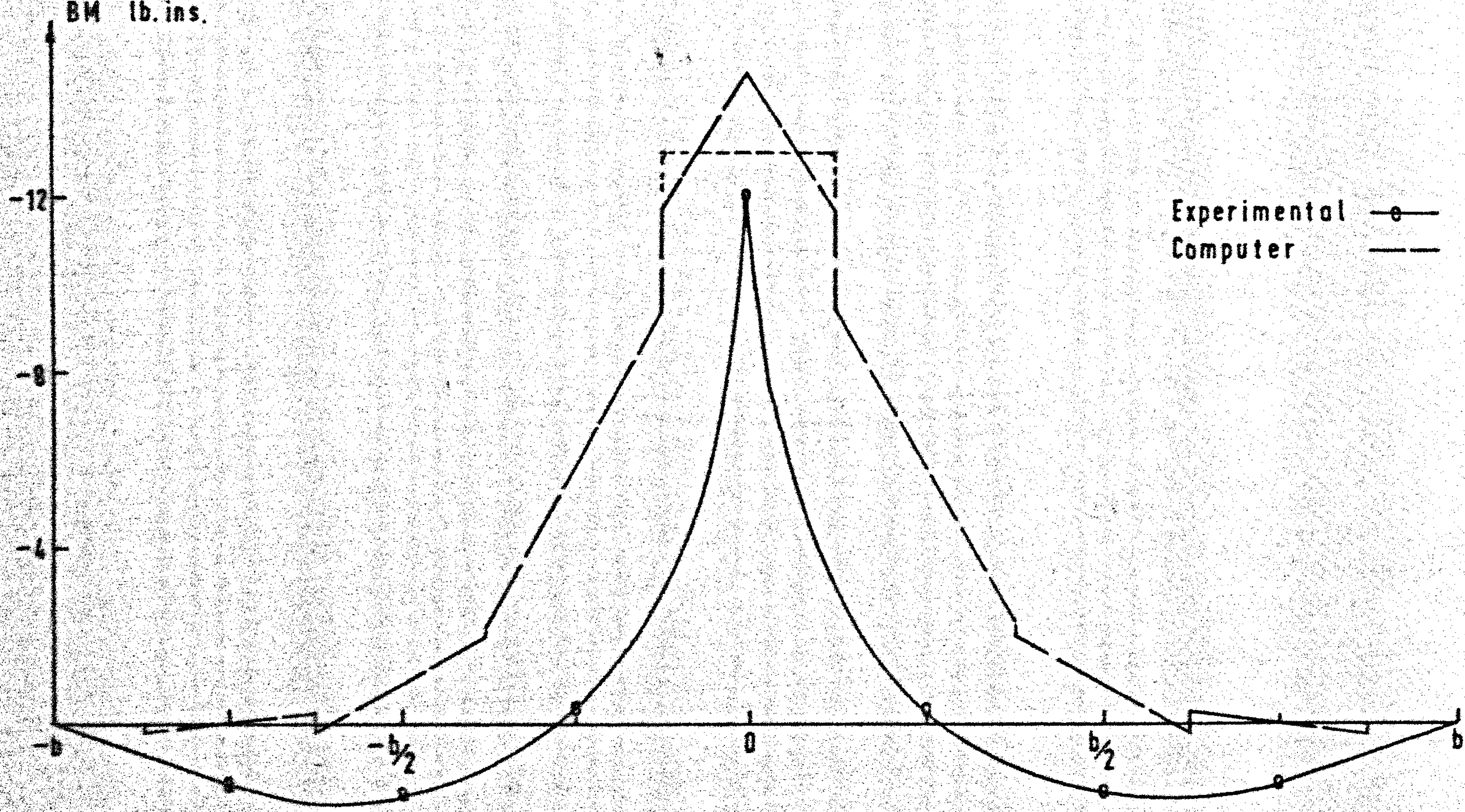


FIG. 4.22. TRANSVERSE BENDING MOMENT IN DIAPHRAGM 17-24

30 lbs. at Y

Stage 2

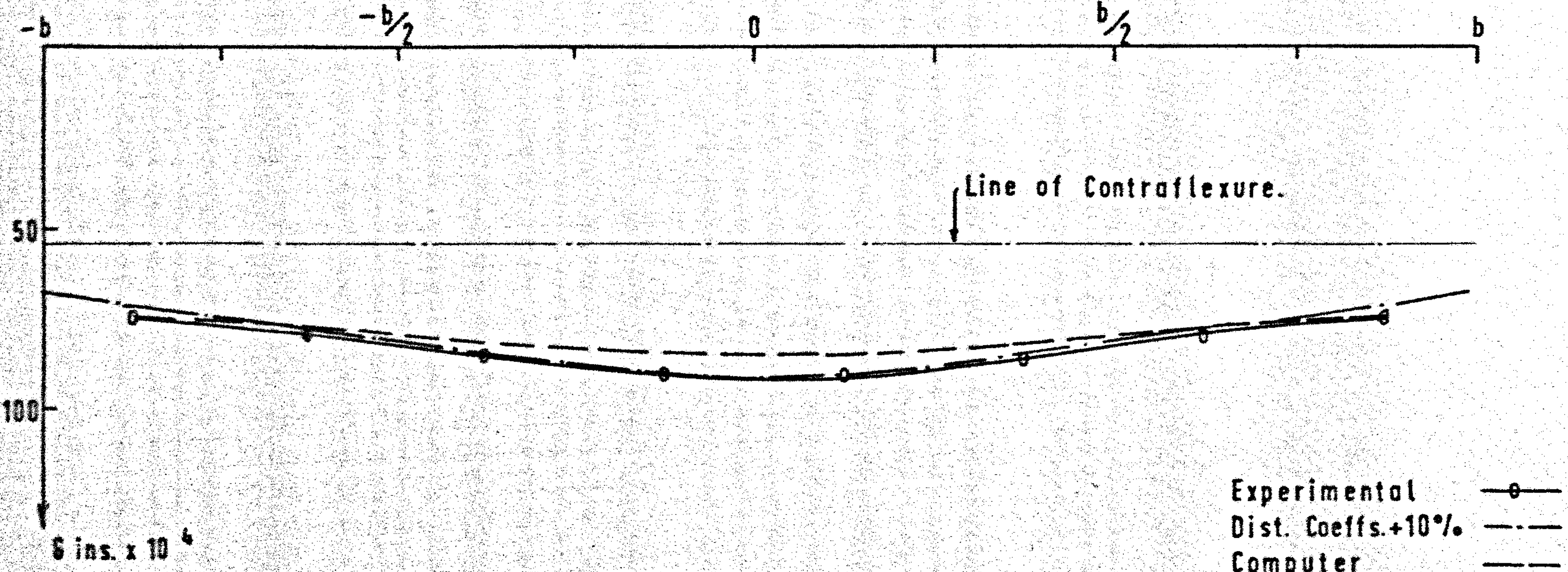


FIG. 4.23 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 65-72

30 lb. at X

Stage 3

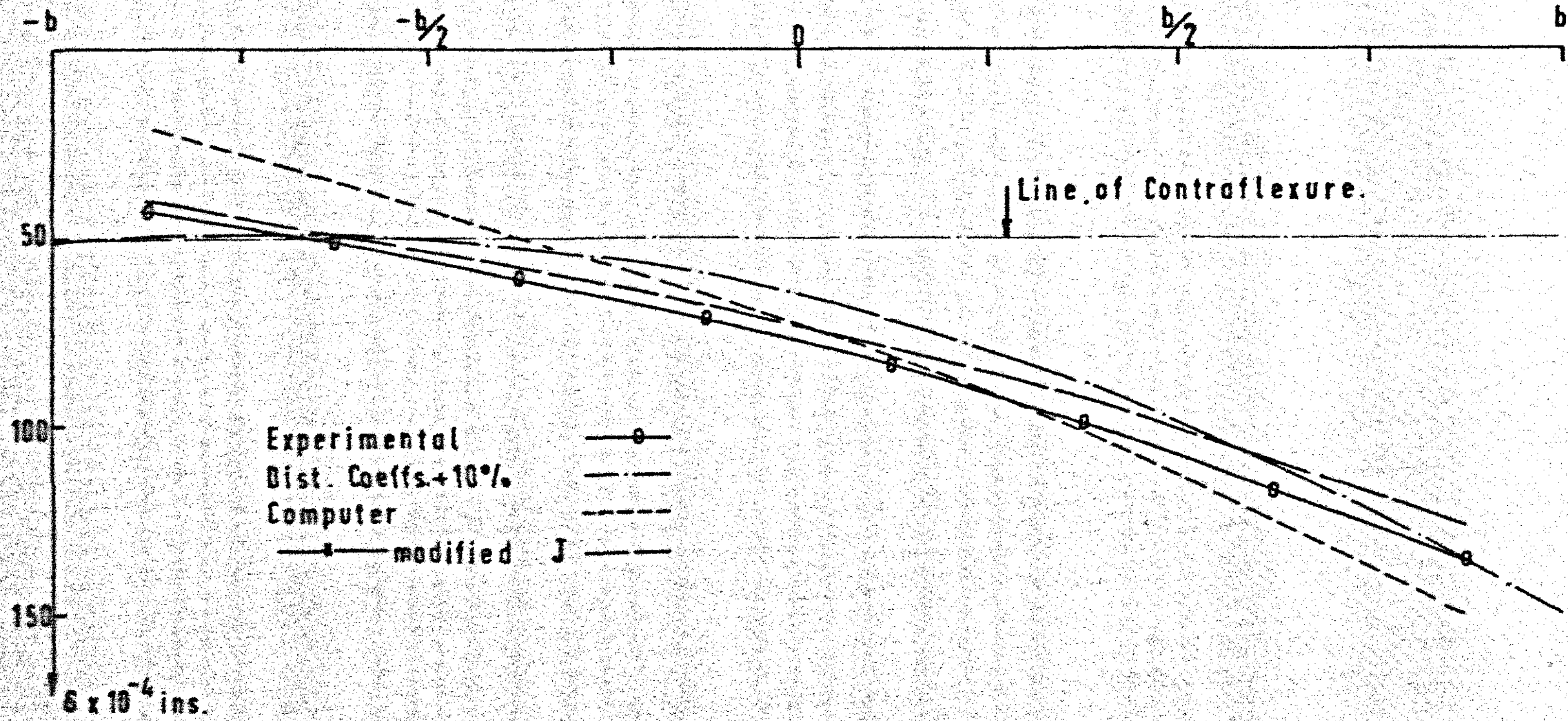


FIG. 4.24 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 65-72
29·4 lbs. at 72

Stage 3

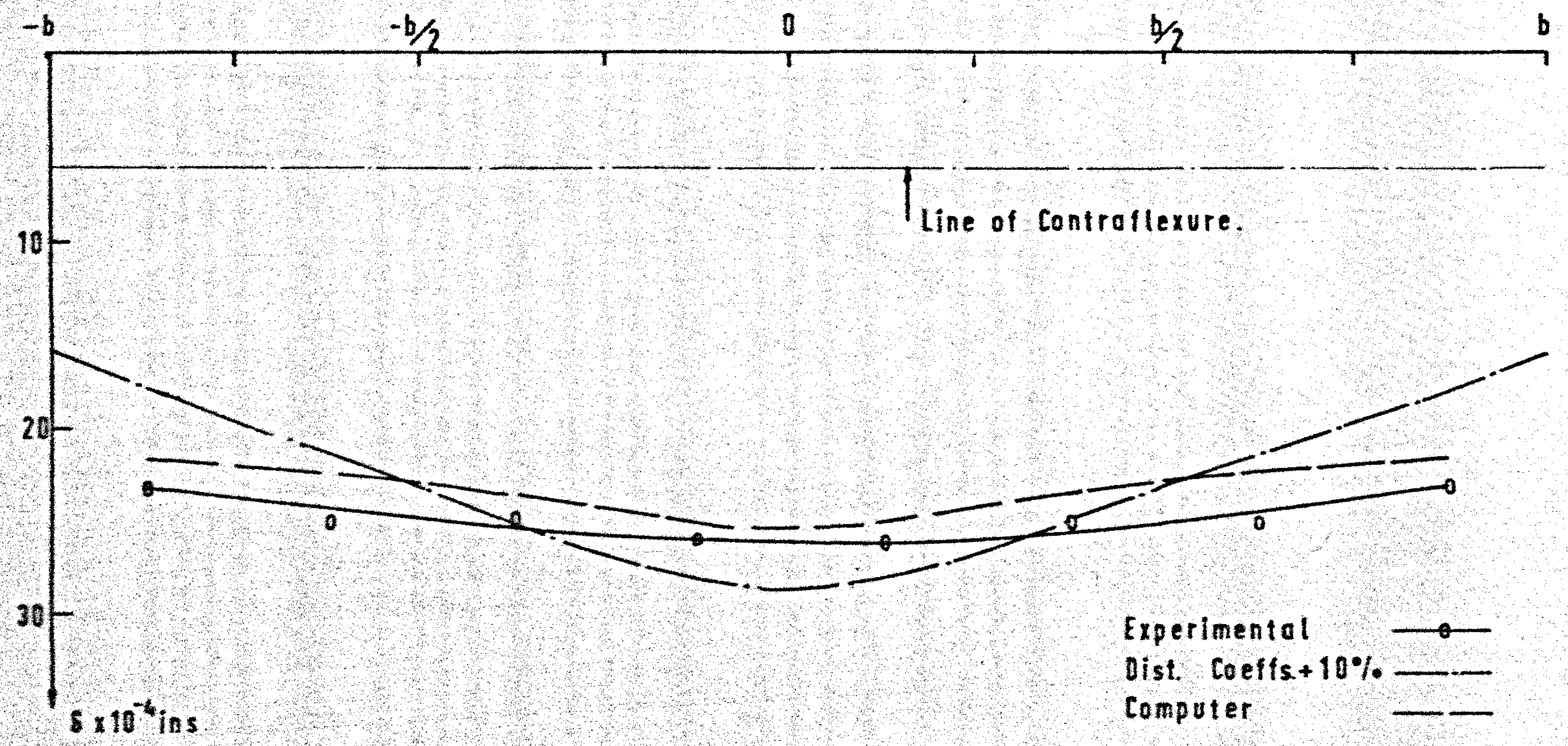


FIG. 4.25 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 17-24

40 lbs. at Y

Stage 3

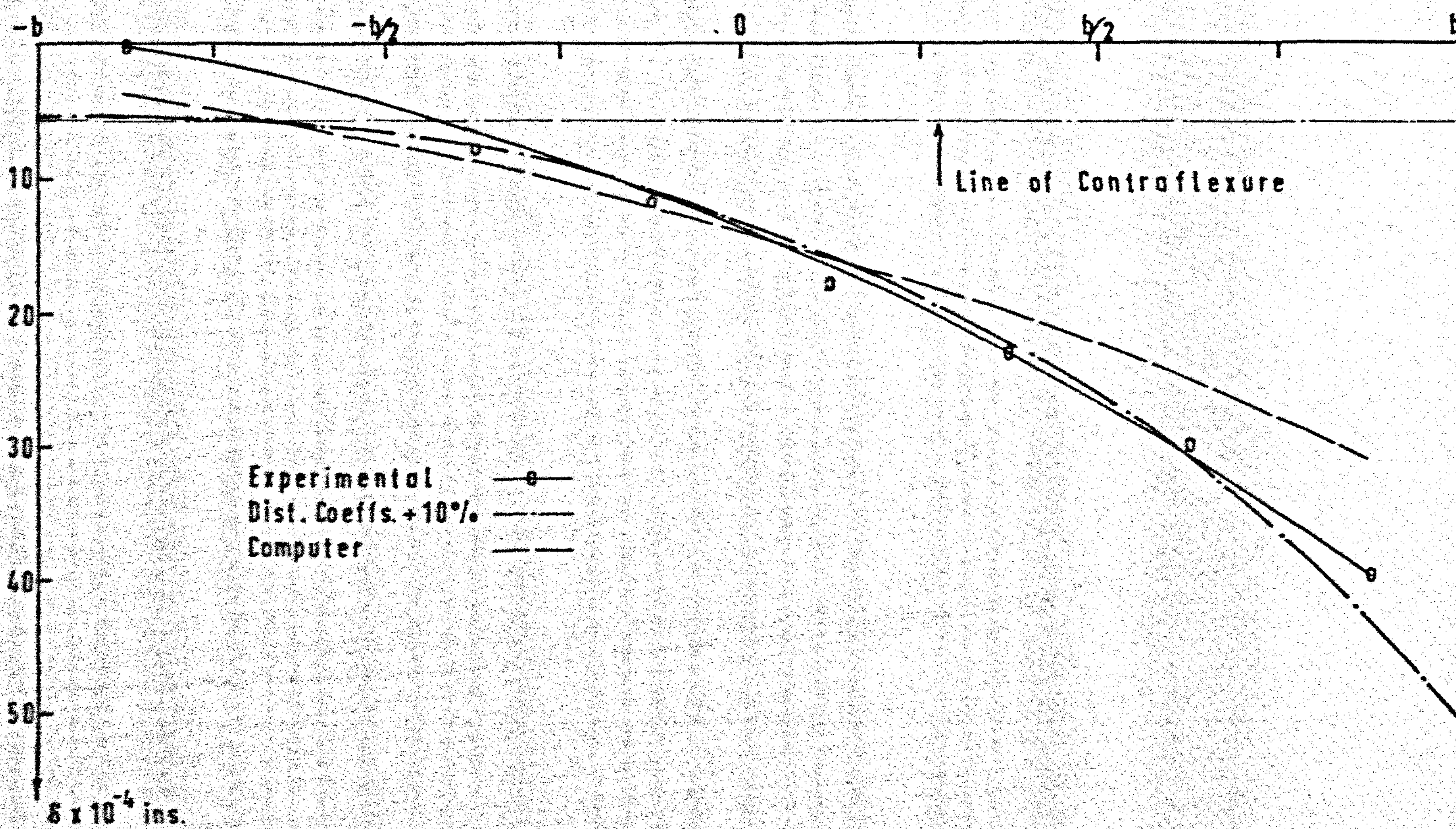


FIG. 4.26 TRANSVERSE DEFLEXION PROFILE ALONG DIAPHRAGM 17-24
29·4 lbs. at 24.

Stage 3

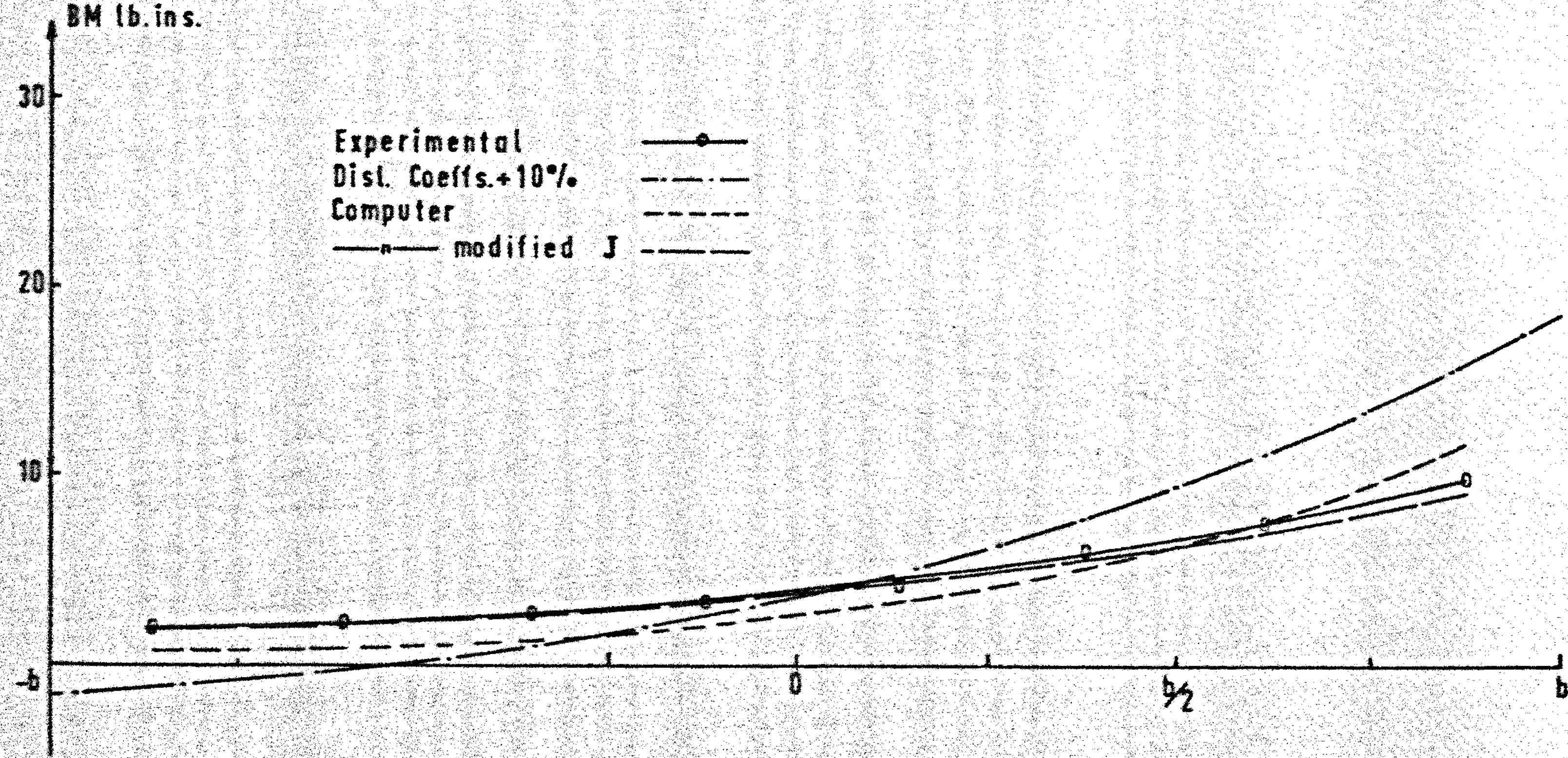


FIG. 4.27 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 65-72
23·4 lbs. at 72.

Stage 3

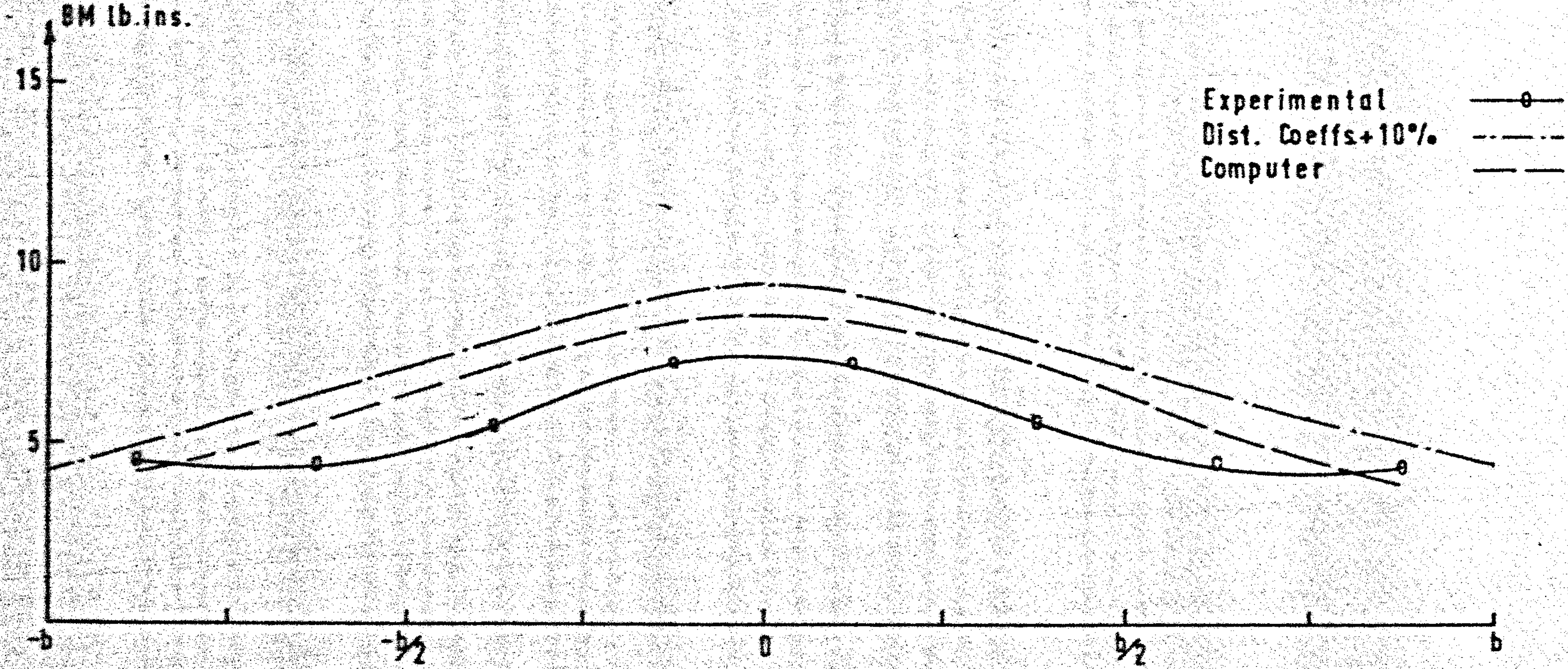


FIG. 4.28 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 65-72
30 lbs. at X.

Stage 3

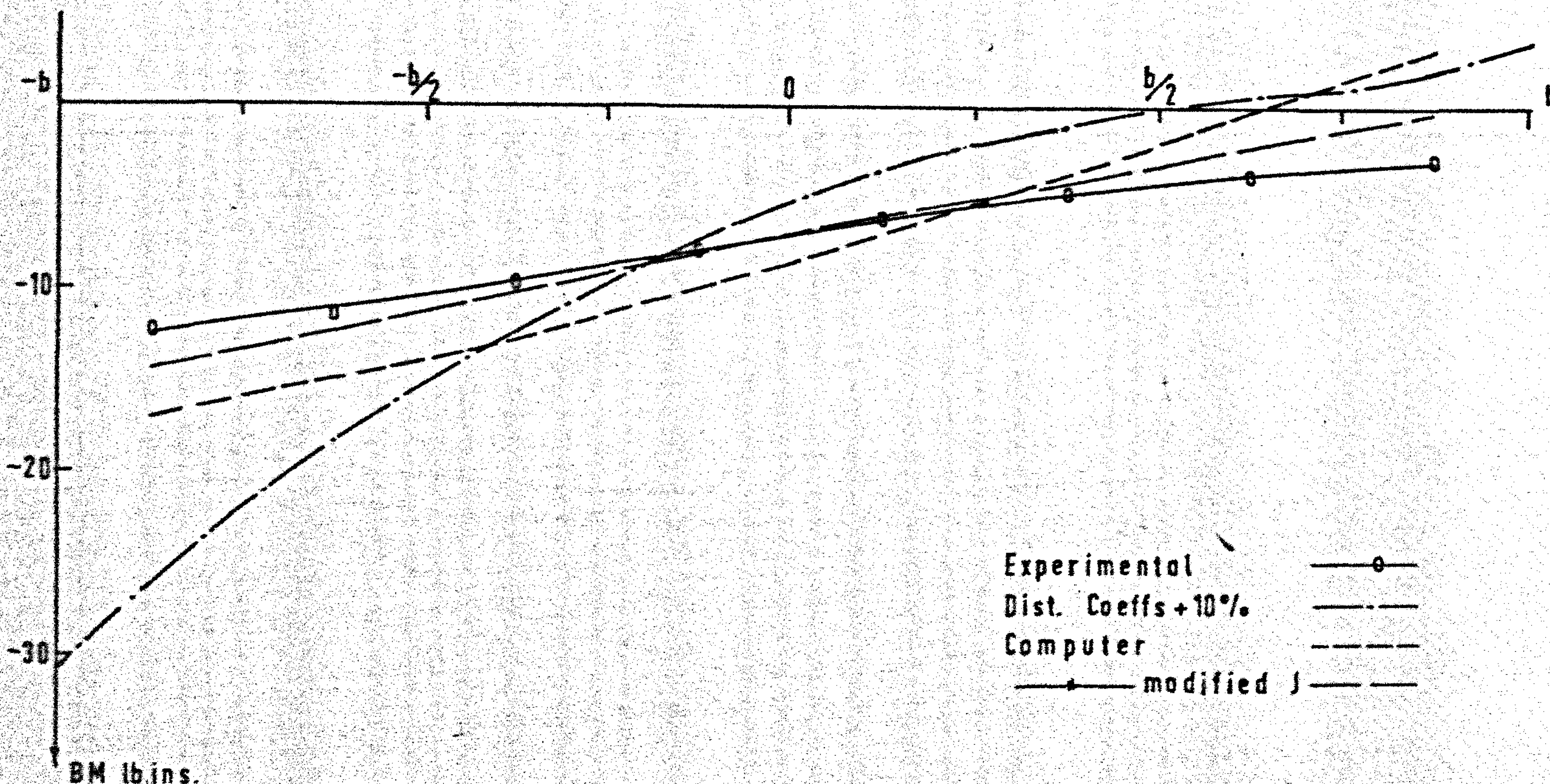


FIG. 4.29 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40

23·4 lbs. at 72

Stage 3

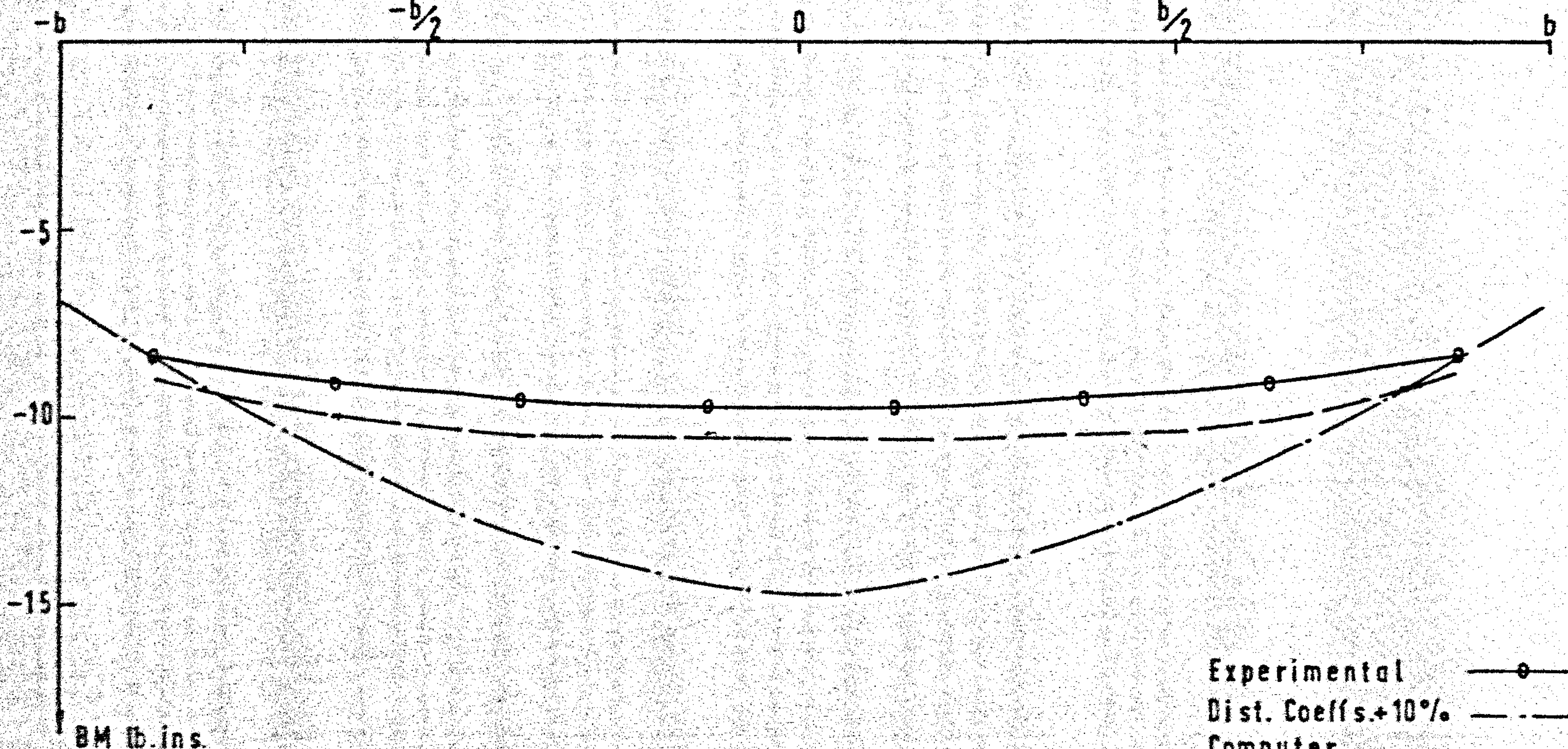


FIG. 4.30 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40
30 lbs. at X

Stage 3

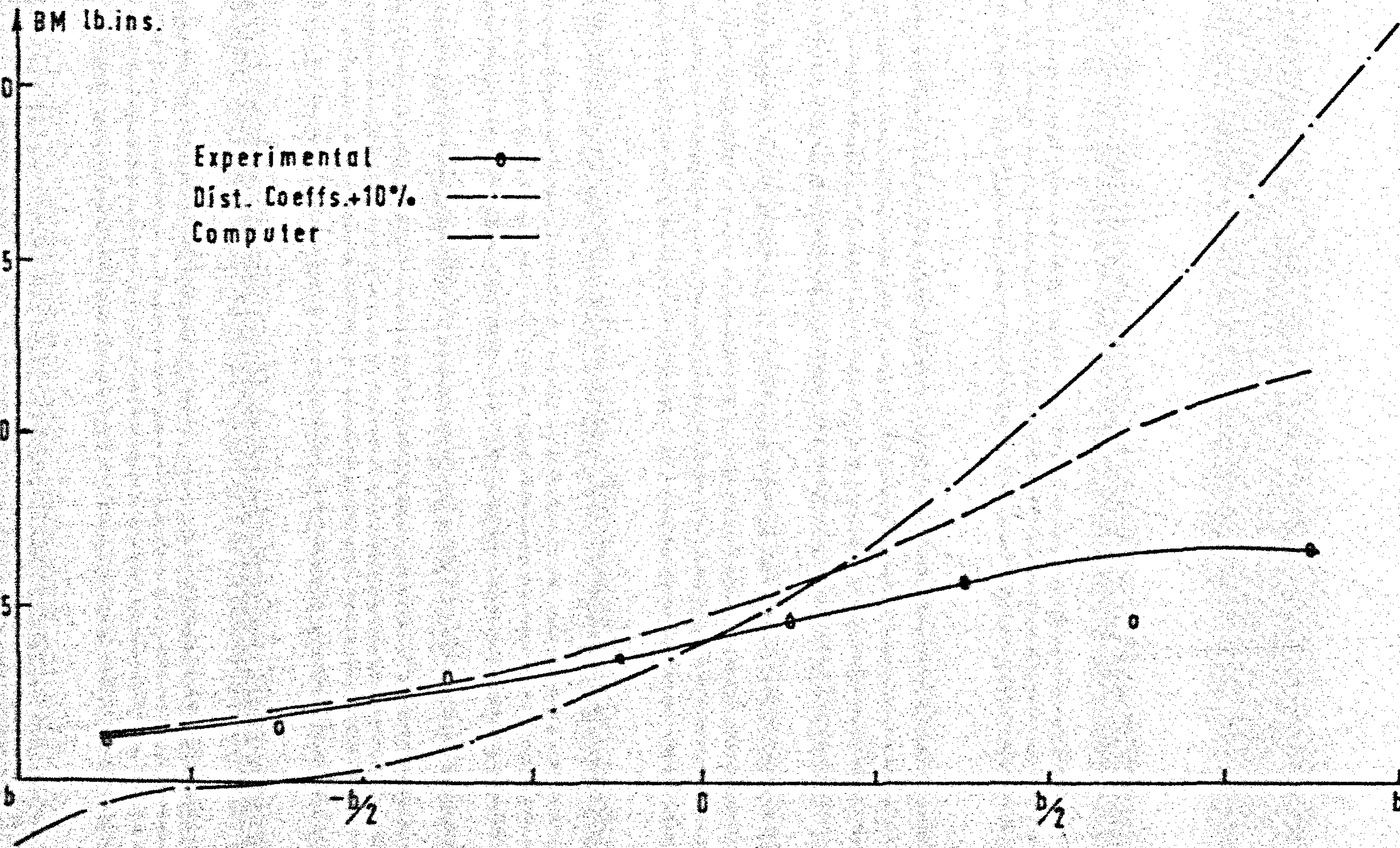


FIG. 4.31 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 17-24
29·4 lbs. at 24

Stage 3.

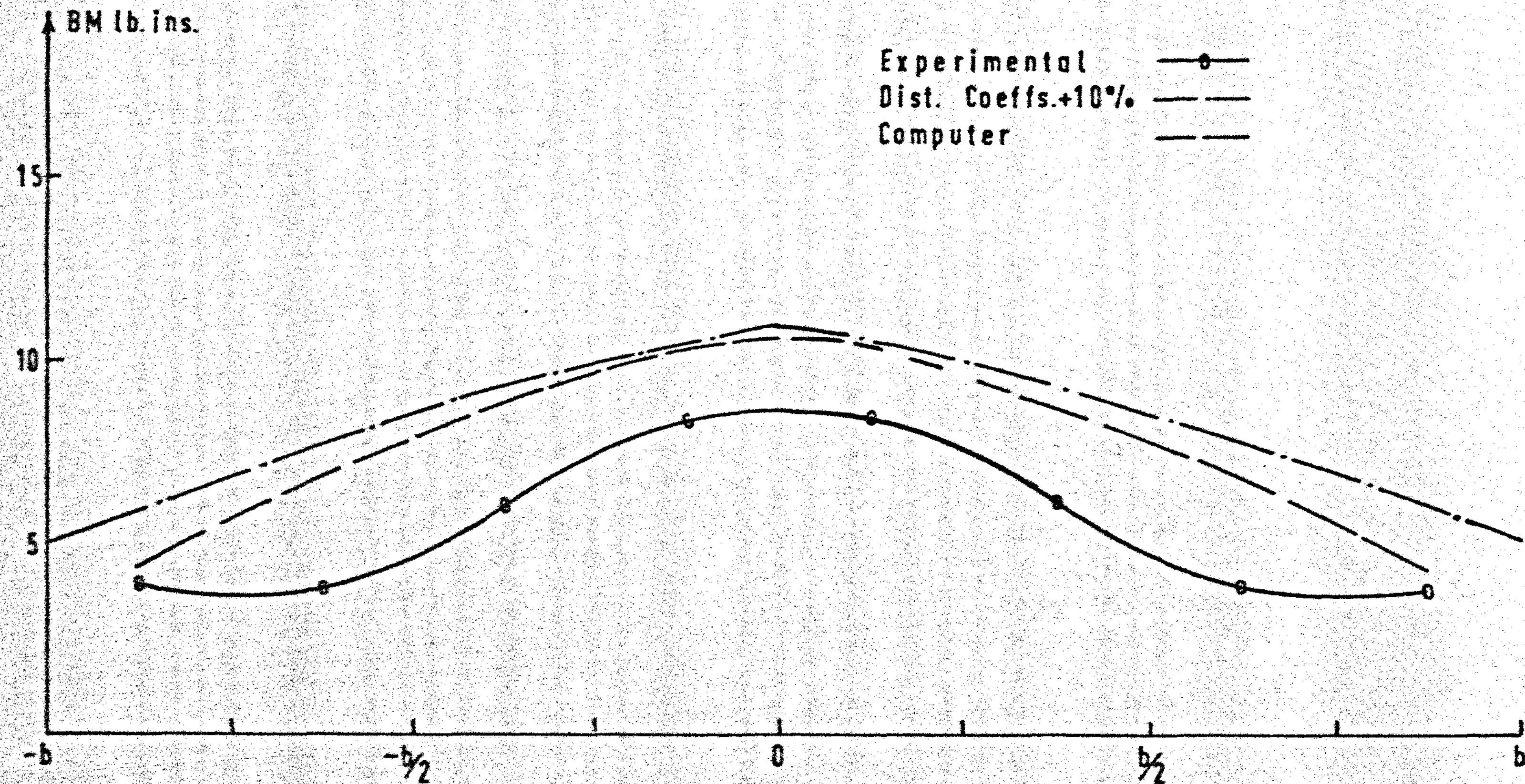


FIG. 4.32 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 17-24
40 lbs: at Y.

Stage 3

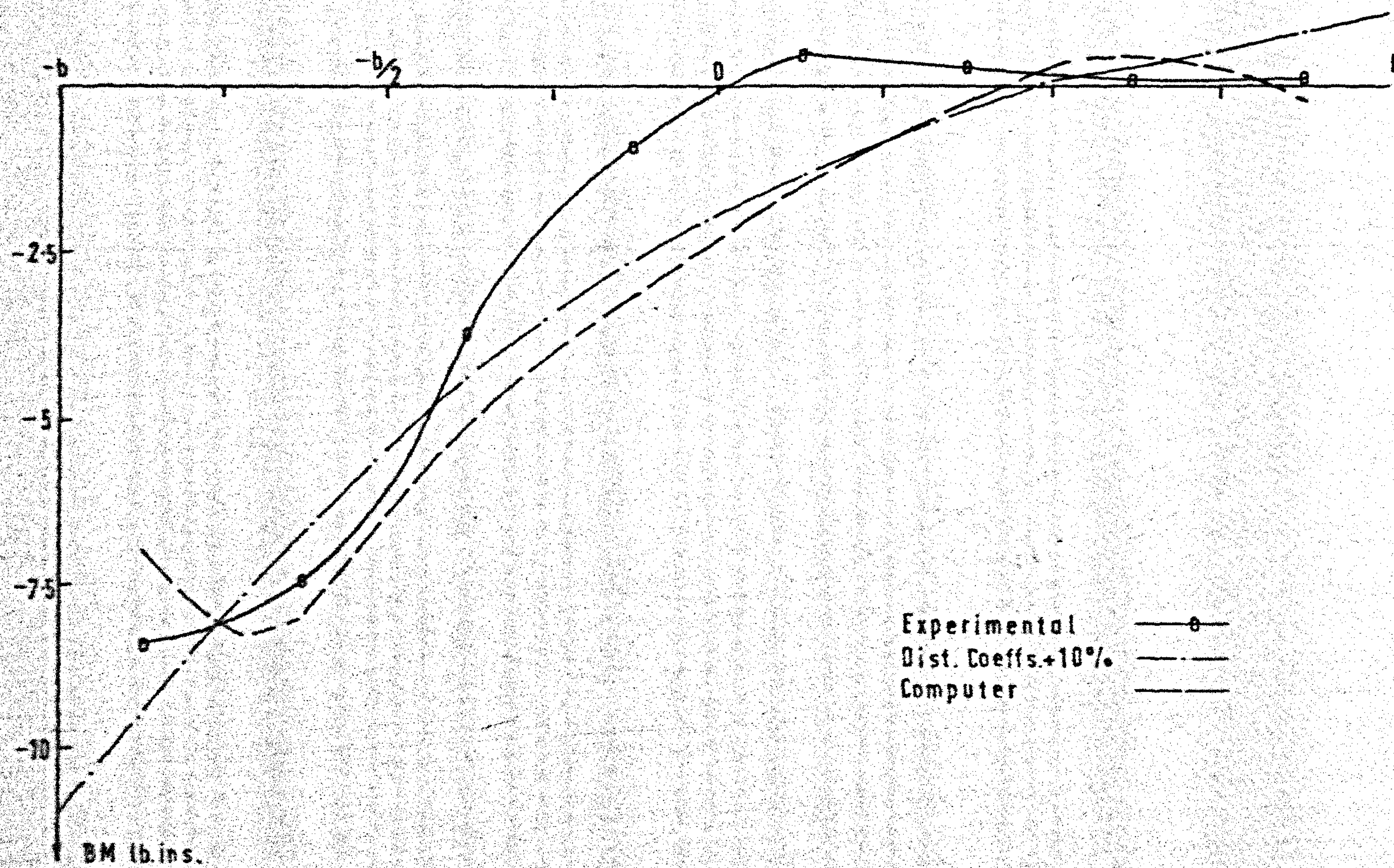


FIG. 4.33 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40
29·4 lb. at 24

Stage 3.

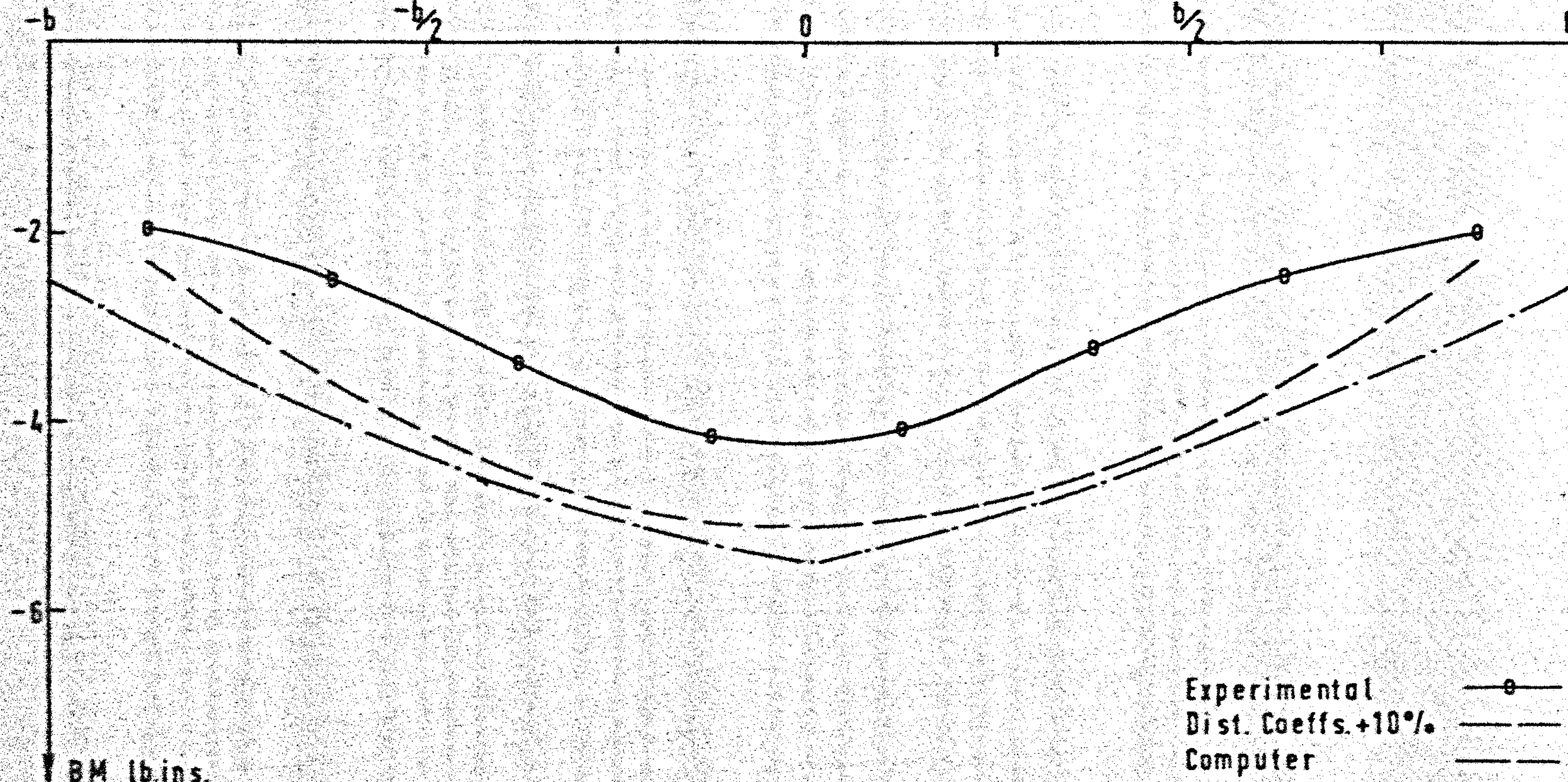


FIG. 4.34 LONGITUDINAL BENDING MOMENT AT DIAPHRAGM 33-40
30 lbs. at Y

Stage 3.

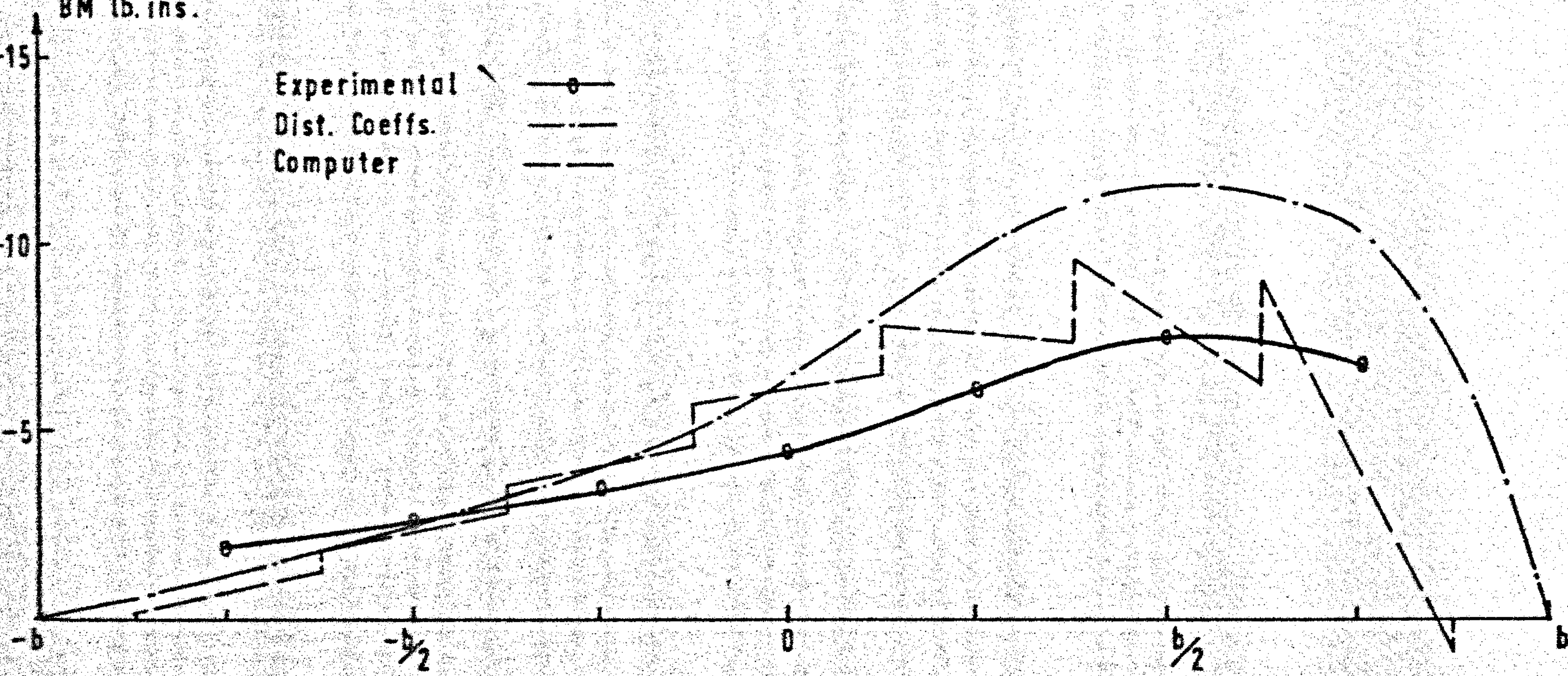


FIG. 4.35 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 65-72
23·6 lbs. at 72.

Stage 3

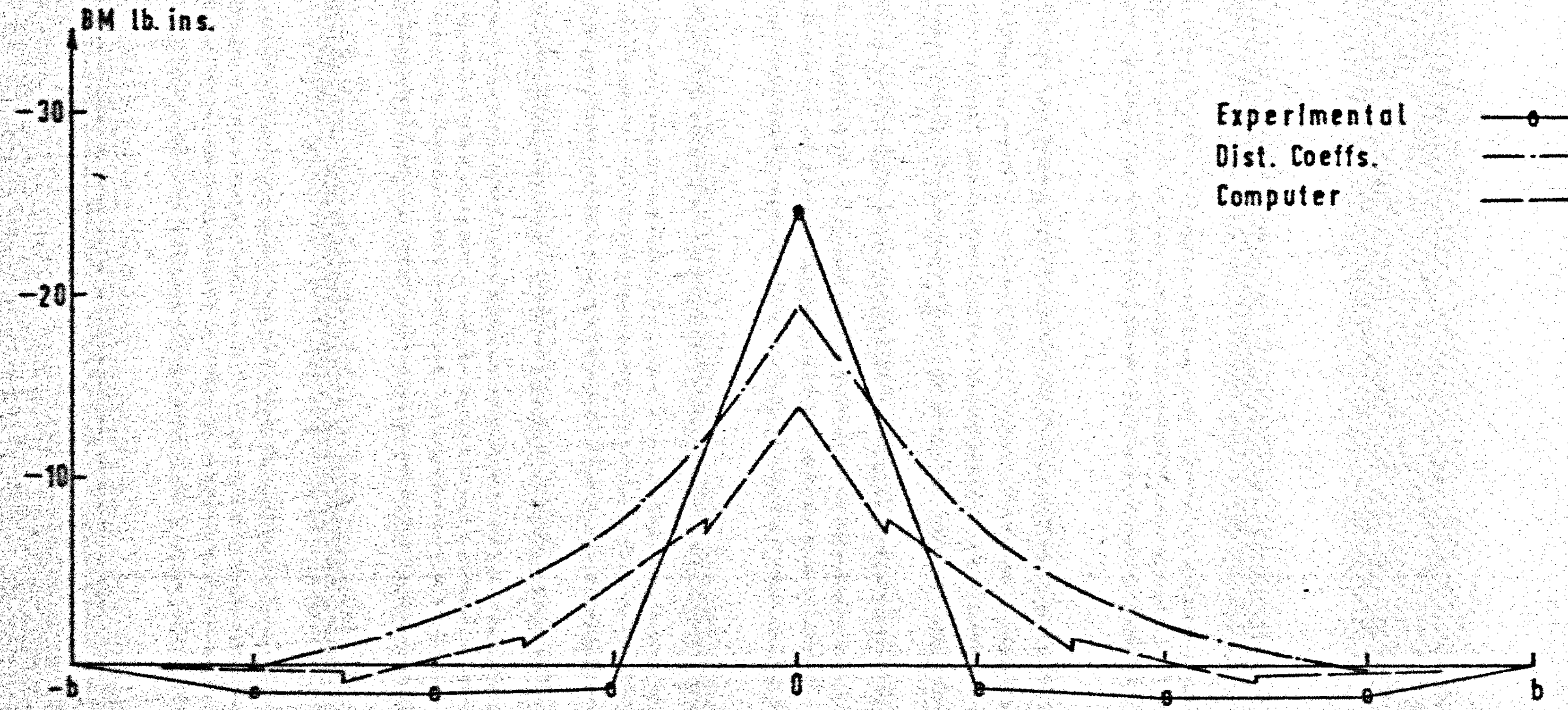


FIG. 4.36 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 65-72
30 lbs. at X

Stage 3.

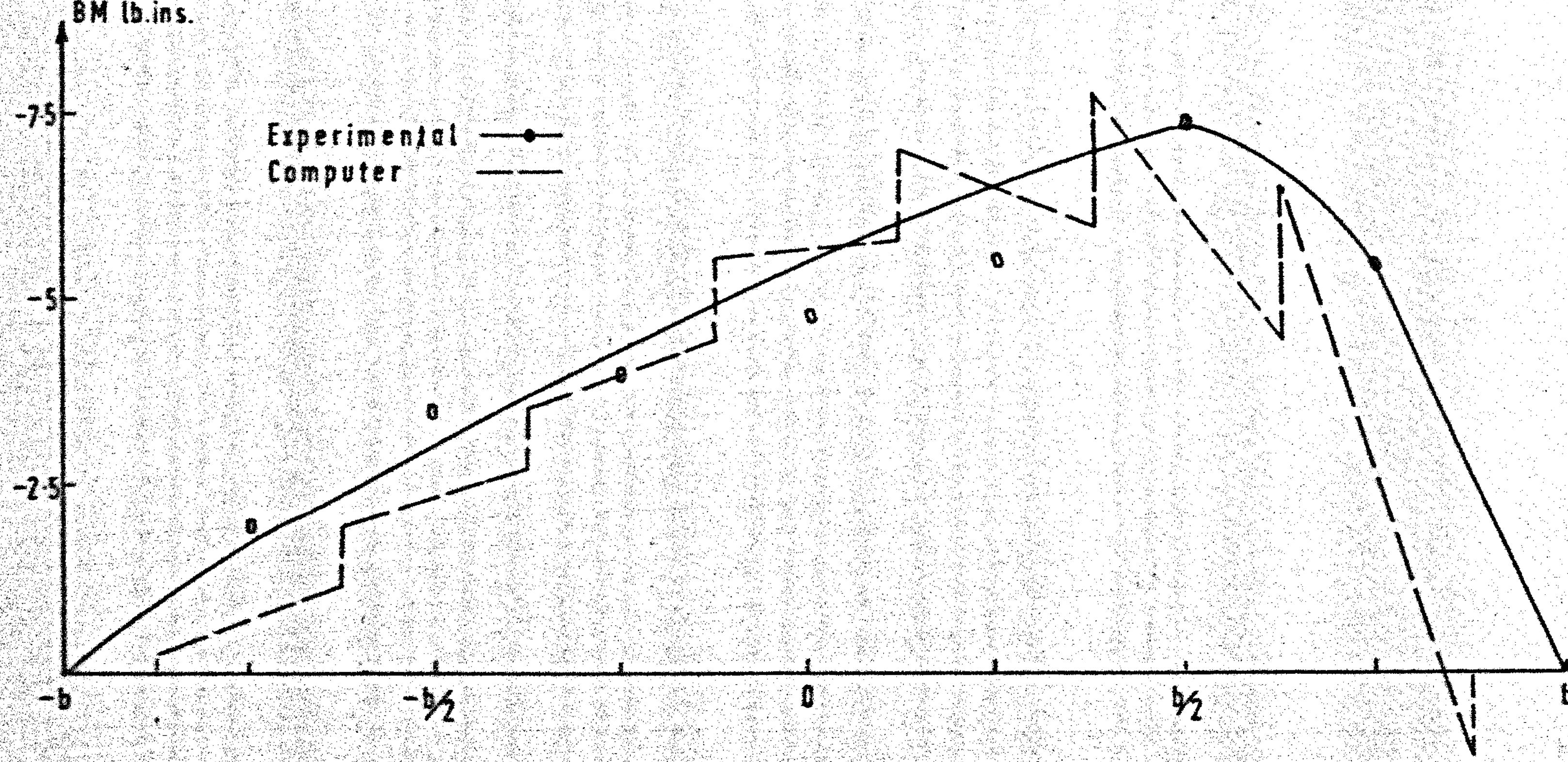


FIG. 4.37 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 17-24
29·4 lbs. at 24

Stage 3

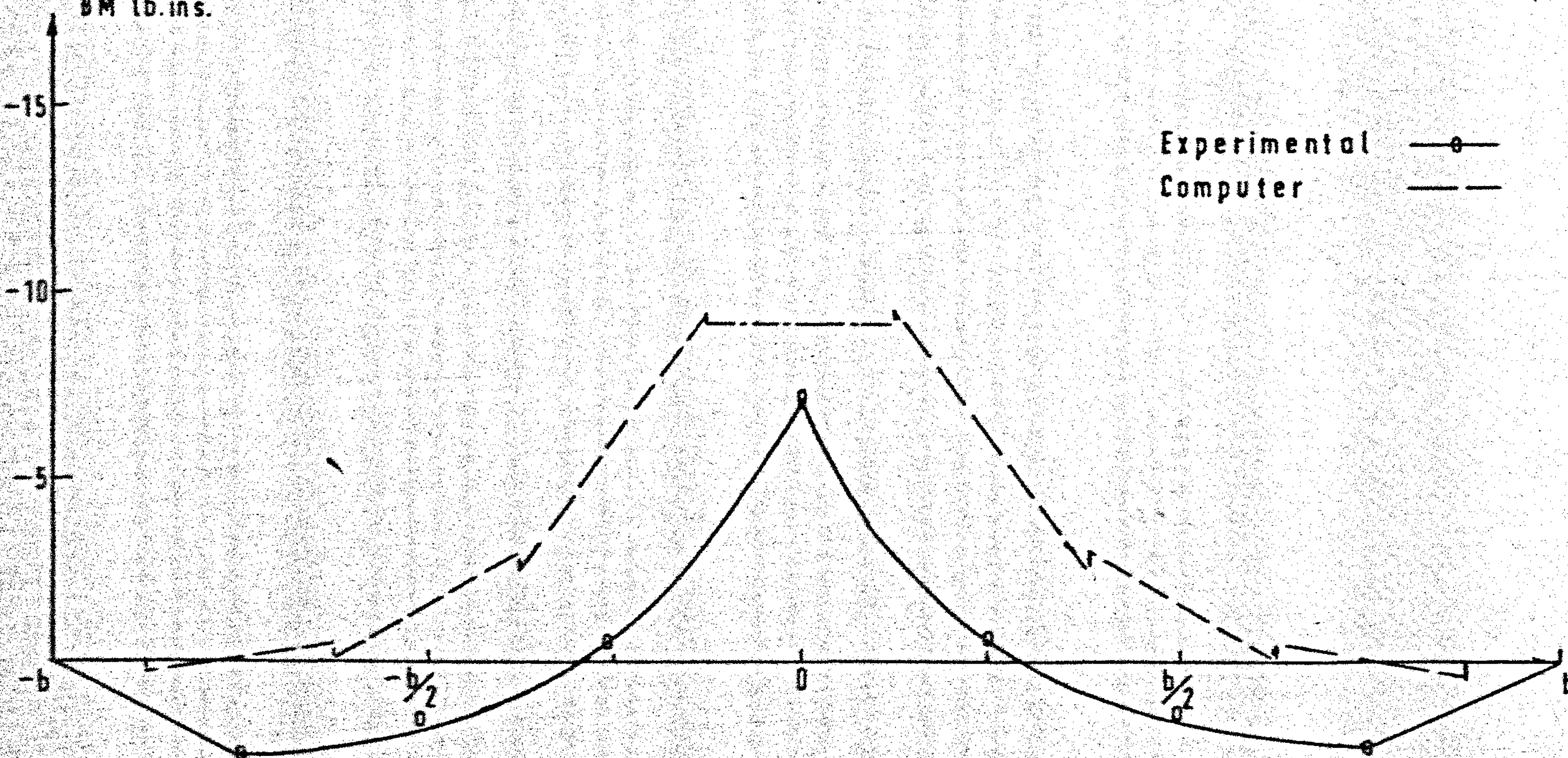


FIG. 4.38 TRANSVERSE BENDING MOMENT IN DIAPHRAGM 17-24 Stage 3
30 lbs. at Y

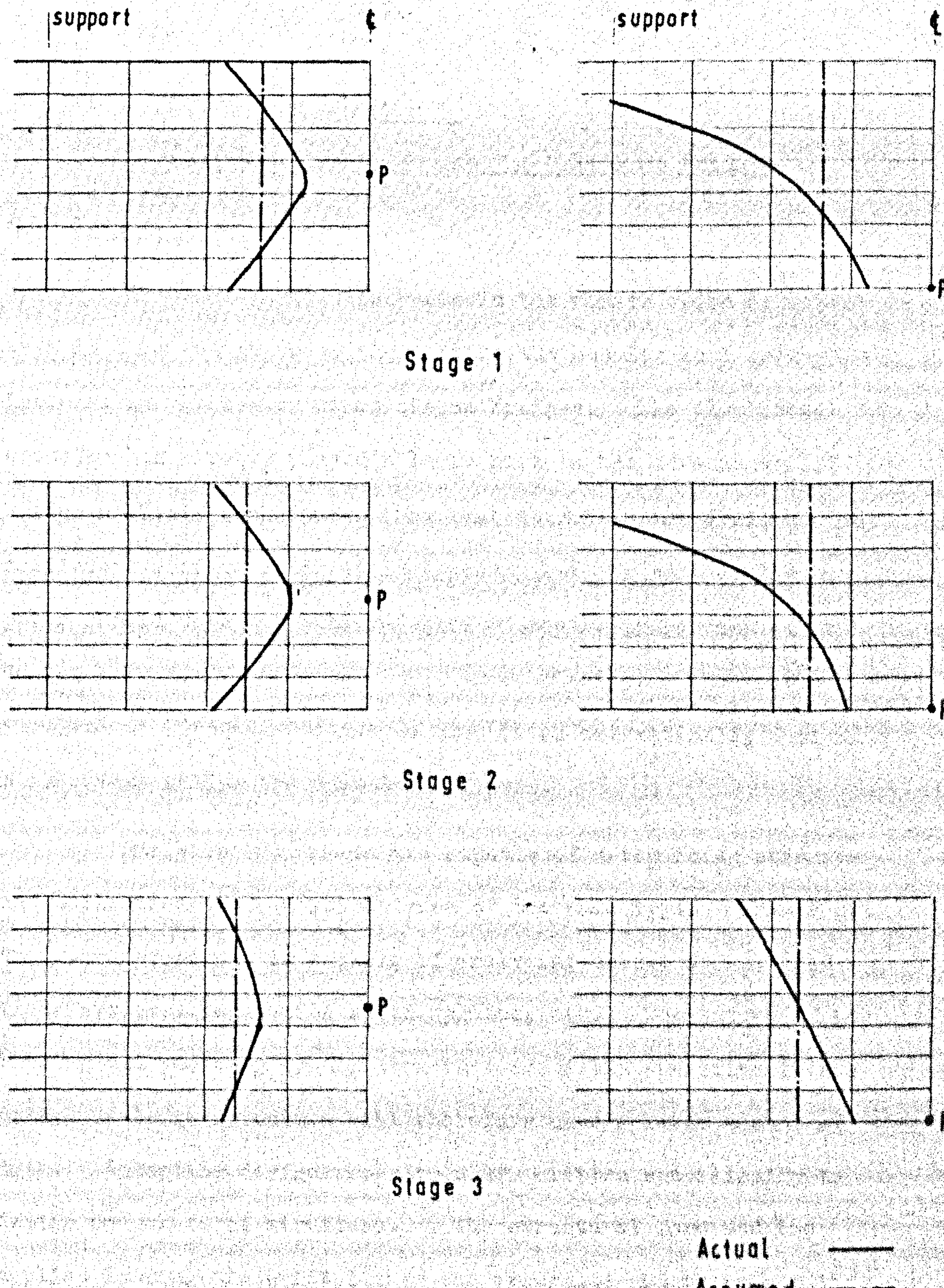


FIG. 4.39 Lines of Contraflexure Under Point Load P

Chapter 5

Automatic Design Programme for Multispan Bridges

5.1) Introduction

The considerable increase in the construction of motorways in this country over the past decade has led to a corresponding increase in the number of multispan bridges. The electronic computer has already played a large part in the advancement of bridge analysis techniques. Now that computers are available with sufficient storage space, it is logical that the role of the computer should change to one in which a greater proportion of the design calculations are performed automatically. The problems involved in writing a programme capable of automatically designing a multispan bridge are investigated here.

Analysis programmes are capable of determining stresses and deflections for many structures of the same type. Thus a grid framework programme is capable of analysing bridge decks, grid floor systems, foundation rafts, plates, etc. The nature of the loading and geometry of the structure are fully specified in the data. A large structure will therefore have a large amount of data. Automatic design programmes are written specifically to design one class of structure. The structural form and the applied loading are written into the programme and therefore the amount of data can be considerably reduced by sacrificing some of the generality.

Before the general propotype for the class of bridge to be designed could be derived, several factors had to be considered. The most usual form of multispan bridge is the three span type, used to cross a river or similar obstruction. The relative span legnths are generally predetermined by the site conditions, where aritificial obstructions such as railways, roads and buildings, or natural effects such as foundation peculiarities, influence the final choice. If the designer has a free choice the span ratios can be chosen to obtain the most economic structure, and obviously this will depend upon many factors. Aziz (22) has investigated the effects of varying span ratios and stiffnesses, upon the cost of the superstructure of bridges that are subjected to HA loading.

Using the programme described in this Chapter for the design of a complete bridge, the effects of span ratio and relative EI values are investigated for bridges subjected to both HA and HB load systems. This investigation is described in Chapter 6.

The aesthetic quality of the structure is largely a matter of individual taste. Emphasis can be placed upon many factors, such as the structural system, the profile, the overall efficiency, patterns of light and shade, etc. Although some conditions can be satisfied by non-structural elements, the profile still has an important influence upon the final appearance. It was therefore decided that the final ratio of girder stiffnesses should be specified as part of the data. A parabolic profile has been chosen, but the design remains general as a zero rise will give uniform section beams.

The transverse section is determined by structural efficiency and economy of construction. The hollow box girderhas been chosen to satisfy these two conditions. The girders are assumed to be continuous at the soffit, as shown in Fig.(5.1), giving a deck that is torsionally very stiff, thus giving the maximum redistribution of loads.

The overall economics of the bridge should also be considered. It is felt that the structural system of continuous varying section longitudinal box girder results in a highly efficient structure, which is also relatively simple in construction, and that this system is compatable with the economic requirements, as a minimum of materials are employed.

The prototype of the bridge was therefore established as a three span, cast in-situ prestressed concrete, continuous bridge with hollow box girders varying in section in the longitudinal direction. Transverse diaphragms are incorporated forming trapezoidal transverse box beams. It was also assumed to be supported on rollers at each support and the whole restrained against horizontal movement. The bridge is designed to withstand the full loading specified by BS 153(1).

5.2) Basic Approach

The method of design is summarised in the flow diagram shown in Fig. (5.2). A single longitudinal continuous girder is first designed to carry the HA loading, and the maximum positive and negative bending moments are determined. The complete bridge deck is then analised under the HB loading condition, taking into account the distribution of load across the deck. The abnormal vehicle is placed on the edge of the deck, at the centre of the midspan section, to give maximum longitudinal moment, and at the centre of the midspan to give maximum transverse moment. The maximum longitudinal moments from the HA and HB loading conditions are compared, and the worst condition used to determine the final section sizes and prestressing force.

5.3) Organisation and Theory of the Programme

5.3.1) Relative Section Properties

The programme first considers a single girder under the HA loading. As there are an infinite number of varying section girders capable of carrying this load, it is assumed that the relative moments of inertia in all spans are given by the relationship:-

$$\frac{I_x}{I_c} = (1 + R\,()1 - \frac{2x}{L})^2)^2 \qquad (5.1)$$

I_x = I at point x distance from L.H. end

I_c = I at centre of midspan

R = Ratio $I_s : I_c$, thus:- $I_s = I_c (1 + R)^2$

I_s = I at inner support.

This by assuming $I_c = 1$ it is possible to obtain relative values of I_x at all points along the girder.

5.3.2) Derivation of Influence Lines

Using the relative I values given by Equation (5.1), a set of bending moment influence lines are calculated. The method used is that described in Chapter 3, in which the influence line is given by the deflected profile of the girder, caused by the application of a unit displacement at the point being considered. The ordinates of deflection are given at joints only, and in order to obtain an accurate set of influence lines for a 3 span girder, it is necessary to sub-divide each span into a series of shorter members, connected by imaginary joints. Each span is divided into 20 shorter members and the joints are numbered automatically from 1 to 61, as shown in Fig. (5.3). The stiffness and carry over factors are calculated using the method described in Section (2.4), each member having five stations at which relative I values are evaluated. The member properties are stored in array P, and consist of C_{12}, C_{21}, K_{12}, K_{21}, length. The girder is considered acting in its own axis only, and each joint is capable of rotation θ_q and deflection δ_z. The general force displacement relationships, given for a member considered relative to its own co-ordinate system in Table (2.1), are therefore applicable. The torsion moment M_p will be zero at all points along the girder, so that the K and R terms in Table (2.1) can be reduced to 2 x 2 elements. The stiffness matrix for the complete girder is then formed as described in Section (2.4). The members are linearly connected resulting in a band width of 4, and therefore the stiffness array S will always be 122 x 4 for the 61 joint girder, each joint being capable of displacement in two directions only.

Influence lines for bending moment are calculated at 21 points along the girder. In the end spans, which are generally shorter, odd joint numbers are considered from 1 to 21 and each joint in the centre span from 22 to 31 incl. Only one half of the girder is considered because of symmetry. The loads to produce unit rotations, $\theta q = 1$, as in Fig. (3.4), are formed in array B, which has 122 x 21 units of store. Zero vertical displacements are imposed at the four supports, i.e. at joints, 1, 21, 41 and 61, and the corresponding terms in the stiffness matrix and loading vectors modified. The set of simultaneous equations can now be solved to give the displacements at 61 joints along the girder, for each of the 21 bending moment influence lines required. Solution is by the 'square root' method, described in Section (2.3). The design programme requires the solution of two sets of simultaneous equations, stored in banded form, and so the solution has been programmed as a special procedure, and given the name 'solve'. As the solution proceeds, the loads in array B are replaced by the corresponding joint displacements, and therefore array B finally holds the ordinates of the required influence lines.

5.3.3) Calculation of Loading

The H.A. loading is specified by B.S.153, Table 1, (1) and is dependent upon the span of the bridge or base length of the influence line. The influence line for bending moment always maintains the same sign in any span, being zero at the supports. Therefore a set of loads based upon each span length, can be calculated which also apply to the influence lines.

The loading can be approximated by the following formula stated by Aziz (22) :-

$$y = \frac{K}{\sqrt{x}}$$

y = U.D.L. per linear foot

x = length of influence line

K = constant

20 000 for 75 x 400

21 000 for 400 x 500

23 000 for 500 x 1900

Maximum overload of 4.7% at x = 550

Maximum underload of -2.98% at x = 150

Average overload = 1.78% } neglecting values that

Average underload = 1.66% } agree with tabulated loading.

For x = 20 → 75 y = 2200 lb/ft.

The load per square foot equals y/10, for lane widths of less than 10 feet, and y/la, where 'la' is the lane width greater than 10 feet. In addition to the U.D.L. a line load of 27,000 lbs. per lane acts anywhere in the span to produce the worst effect, which is at the maximum influence line ordinate.

The weight of surface finishes has also been included in the live load, to produce the greatest range of bending moment. An average value of 37.5 lb/sq.ft. has been taken as acting across the full width of the deck.

The H.B. loading consists of an abnormal vehicle of up to 45 units (180 tons), carried on a system of four axles, as shown in Fig. (5.4). One lane is loaded with type H.B. loading only and all other lanes are loaded with one third H.A. loading.

5.3.4) Design of Discrete Girder Under H.A.Loading.

The maximum positive and negative bending moments at the 21 points along the girder are next calculated. Simpson's rule is used to determined the area enclosed by the influence line for each span, which is then multiplied by the U.D.L. for that span. The maximum ordinate is multiplied by the line load and the total moment added to array ML. The positive values are held in column 1 and the negative values in column 2, thus after all spans have been considered the total maximum and minimum live load moments at the 21 points along the girder, are known. As the influence lines have been derived from assumed relative I values and the live loading is dependent upon the span lengths, these values of bending moment will be valid throughout the whole design process.

The centres of webs are fixed as - longest span/30, but not less than 6' 0". The final sizes might be slightly less than these figures, in order to make up the bridge width of equidistant webs.

The thickness of the top flange is uniform over the whole width of the bridge and was determined by placing two $11\frac{1}{4}$ ton wheel loads at 3 feet centres, at the critical point between the webs i.e., so that the centre line of the slab is midway between the centroid of the load system of the first wheel. The section was considered to be fully fixed at the webs and assuming a permissable concrete stress of 1000 lbs/sq.ins. the following thicknesses were derived:-

6" for $cs < 6$

7" for $6 < cs \leqslant 8$

8" for $8 < cs \leqslant 10$

9" for $10 < cs \leqslant 12$

12" for $12 \leqslant cs$

cs = centres of webs in feet.

The thickness of the web is determined either by the minimum width required to contain the prestressing cables allowing sufficient cover, or the minimum width able to withstand the principal tensile stress. The formula for approximate web thickness given by Evans and Bennett (30) is used :-

$$tw > \frac{0.85\ V_u}{C_t\ \ D} \tag{5.2}$$

where V_u = ultimate shearing force.

c_t = permissable tensile strength of concrete.

D = overall depth.

The maximum shear stress is assumed to occur at $\frac{1}{4}$ point and a load factor of 2 is applied to live plus dead load, to obtain V_u. The two values of web thickness are compared and the computer selects the highest value, which is then rounded off to the nearest inch above.

The design of the sections along the girder is based upon standard prestressed concrete theory, and follows the procedure given by Evans and Bennett (30).

The prestress in the concrete section is :-

At bottom of section $f_b = \frac{P}{A} - \frac{Pe}{Z_1}$ (5.3)

At top of section $f_t = \frac{P}{A} + \frac{Pe}{Z_2}$ (5.4) where P = prestressing force at transfer

e = cable eccentricity - negative when below N.A.

A = cross section area of concrete.

There are two loading conditions at which the concrete stresses are critical. The first occurs when the minimum bending moment is combined with the maximum prestress, to give the stress condition :-

$$\text{At bottom of section} \quad f_b - \frac{M_{min}}{Z_1} < f_{ct} \qquad (5.5)$$

$$\text{At top of section} \quad f_t + \frac{M_{min}}{Z_2} > f_{mint} \qquad (5.6)$$

where f_{ct} = permissable compressive stress at transfer

f_{mint} = permissable tensile stress at transfer.

The second loading condition occurs after the maximum loss of prestress has occured, in conjunction with the maximum bending moment where the following stress conditions can be written :-

$$\text{At bottom of section} \quad R_o f_b - \frac{M_w}{Z_1} > f_{minw} \qquad (5.7)$$

$$\text{At top of section} \quad R_o f_t + \frac{M_w}{Z_2} < f_{cw} \qquad (5.8)$$

where f_{cw} = permissable compressive working stress

f_{minw} = permissable tensile working stress

R_o = ratio of loss in prestressing force.

The minimum section moduli can now be found.

Eliminating f_b from inequalities (5.5) and (5.7) :-

$$Z_1 > \frac{M_w - R_o M_{min}}{R_o f_{ct} - f_{minw}} \qquad (5.9)$$

Eliminating f_t from inequalities (5.6) and (5.8) :-

$$Z_2 > \frac{M_w - R_o M_{min}}{f_{cw} - R_o f_{mint}} \qquad (5.10)$$

In deriving inequalities (5.9) and (5.10) it has been assumed that the bending moment is positive (sagging moment). When the bending moment is negative (hogging moment) the formula for minimum section moduli must be rewritten thus :-

$$Z_1 > \frac{M_w - R_o M_{min}}{f_{cw} - R_o f_{mint}} \qquad (5.10a)$$

$$Z_2 > \frac{M_w - R_o M_{min}}{R_o f_{ct} - f_{minw}} \qquad (5.10b)$$

From these two formuli, it is clear that the dimensions of the concrete are dependent upon the range of loading, the permissable stress and the loss ratio. At this stage in the design, the bending moment due to dead load is not known, but the numerator of inequalities (5.9) and (5.10) can be writter as :- (M live.max + M live.min). When the loss in prestress is zero, i.e., $R_o = 1$, this value will correspond to that given by the inequalities, but normally R_o will have a value of less than one resulting in a slight error. Hence approximate values for the minimum section moduli are obtained. By assuming the bottom flange is equal in thickness to the top flange and neglecting the web contribution, it is possible to arrive at an approximate depth for the section. The distance from the neutral axis to the centroid of the flange is given by the quadratic root :-

$$y = \sqrt{\frac{Z + Z^2 + 4cs\,Hf^2\,Z}{4cs\,Hf}} \qquad (5.11)$$

where Hf = thickness of top flange

cs = centres of webs in inches

Hence $D = 2 \times y + Hf$

The depths at the supports and centre of the main span can thus be found. As the ratios of moment of inertia must satisfy equation (5.1) the depths are also checked to ensure they conform to the relationship :-

$$d_s = d_c \, (1 + R) \qquad (5.12)$$

where d_s = depth of innersupport

d_c = depth of centre of main span.

If this equation is not satisfied either d_c or d_s is increased proportionately, in which case the maximum permissable stresses will not be attained at that section.

Hence an approximate trial section has been derived, with the dimensions known at the internal supports and centre of the main span. The variation in depth is parabolic, so that the section in any point along the girder can be calculated. Before the section can be finalised the design of the prestress has to be completed, which is dependent upon the dead load bending moment.

These trial sizes are used to determine the position of the prestressing force. By combining equation (5.5) with (5.7) the magnitude of the prestress will be given :-

Bottom of section $$\frac{f_{minw}}{R_o} + \frac{M_w}{R_o Z_1} < f_b < f_{ct} + \frac{M_{min}}{Z_1}$$

Top of section $$f_{mint} - \frac{M_{min}}{Z_2} < f_t < \frac{f_{ow}}{R_o} - \frac{M_w}{R_o Z_2}$$

The lowest value of prestress will normally be chosen so that the two formulae can be shortened to :-

$$f_b > \frac{f_{minw}}{R_o} + \frac{M_w}{R_o Z_1} \qquad (5.12)$$

$$f_t > f_{mint} - \frac{M_{min}}{Z_2} \qquad (5.13)$$

These equations are for positive values of bending moment and must be rewritten for negative bending moments :-

$$f_b > f_{mint} + \frac{M_{min}}{Z_1} \qquad (5.12a)$$

$$f_t > \frac{f_{minw}}{R_o} - \frac{M_w}{R_o Z_2} \qquad (5.13a)$$

The values of dead load bending moment at the 21 joints along the girder, are found by integrating the dead weight and influence line ordinates at the 61 points along the girder, using Simpson's factors. The dead load moments are placed in the third column of array M1. The exact thickness of the bottom flange is known only at the centre of the main span and at the inner supports, and so for the purpose of calculating the dead load, the bottom flange thickness is assumed to vary parabolically.

By eliminating P from equations (5.3) and (5.4), the position of the prestressing force can be expressed :-

$$e = \frac{Z_1 Z_2 (f_t - f_b)}{A (f_b Z_1 + f_t Z_2)} \qquad (5.14)$$

The minimum bending moment usually occurs with the dead weight acting together with live load in the remote spans. In long span bridge beams the dead load is often large compared with the live load, which decreases as the span increases. A large prestressing force is therefore required to counteract the minimum moment stresses. When the minimum moment exceeds a certain critical value, the position of the prestressing force given by equation (5.14) falls below the soffit of the section, in order to achieve an ideal prestress in the concrete at the top and bottom of the section. In practice the cable must be located at the lowest practical limit and increased in magnitude to maintain the same prestress at the bottom of the section (f_b). This, however, results in a reduced negative prestress at the top of the section, so that under minimum moment the stress in the concrete is increased. The minimum concrete stress f_{mint}, is therefore greater than it need be.

The minimum concrete stress at transfer can be found by substituting for f_t and f_b from equations (5.3) and (5.4), in equations (5.6) and (5.7), and eliminating P from the resulting two equations:-

$$f_{mint} = \frac{M_{min}}{Z_2} + \frac{Z_1 \, (Z_2 + Ae)}{R_o \, Z_2 \, (Z_1 - Ae)} \left(\frac{M_w}{Z_1} + f_{minw} \right) \quad (5.15)$$

If the value of minimum stress f_{mint} is greater than originally used, the value of Z_2 given by equation (5.10) is recalculated, using the new value.

There is no direct method of finalising the section because the exact magnitude of the dead load is not known. The programme therefore, proceeds by successive correction. The preliminary dimensions are based upon the required section modulus and top flange thickness. From these dimensions the total dead load and dead load bending moment are calculated. By using equation (5.14) the position of the prestressing force is calculated. If this falls below the practical limit of 3" above the soffit (allowing for cover to the cables) the position is fixed at this lower limit. The minimum concrete stress is recalculated using equation (5.15) and a new value of Z_2 is calculated. Z_1 and Z_2 do not have the same value and in order to obtain a more efficient section the thickness of the bottom flange is fixed by the ratio:

$$tbf = \frac{ttf . Z_1}{Z_2} \qquad (5.16)$$

where tbf = thickness of bottom flange.

With the values of Z_1, Z_2 ttf and tbf known at the supports and centre of the main span, an approximate depth of section can be found. If the web contributions are neglected, the distance from the section centroid to the top flange centroid is given by the quadratic root :-

$$y = \frac{Z_2 \sqrt{Z_2^2 + 4 cs (ttf + ttf^2/tbf) Z_2 \, ttf/2}}{2 cs (ttf + ttf^2/tbf)} \qquad (5.17)$$

A provisional depth to the nearest inch above is calculated, which will be greater than necessary to provide the minimum values of Z_1 and Z_2, because the web contributions were neglected. The moment of inertia and centroid position is now calculated using the known flange thicknesses and assumed depth, from which the actual values of Z_1 and Z_2

are found. A ' procedure' to calculate the moment of inertia and centroid depth is written into the programme, and given the same 'find I'. The actual Z values are compared with the minimum required values given by equations (5.9) and (5.10). If the actual values are higher than the required values the depth is reduced by $\frac{1}{2}$", and by again using procedure 'find I', new values of actual Z_1 and Z_2 are calculated. This cycle continues until either one of the actual Z values falls below the minimum required Z value. The depth is then increased by $\frac{1}{2}$", so that the depth is fixed to the nearest $\frac{1}{2}$" above the minimum required depth. In this way, the depths at the centre of midspan and at the internal supports are calculated. The ratio of moments of inertia is then compared with that given by equation (5.1). If this relationship is not satisfied, the depth at either inner support or centre of main span is increased. If the moment of inertia at the support is insufficient to satisfy equation (5.1), a new value is found :-

$$I_{support} = I_{centre} (I + R)^2 \qquad (5.1a).$$

Neglecting the web contributions, an approximate depth can be found. The distance from the centroid of the section to the centroid of the top flange is given by :-

$$y = \sqrt{\frac{I_{reqd}}{cs\,ttf + \frac{cs\,ttf^2}{tbf}}} \qquad (5.18)$$

By successively reducing the depth in $\frac{1}{2}$" increments and using procedure 'find I', a depth to the nearest $\frac{1}{2}$" above is found, to give the required value of moment of inertia. The section at this point will be understressed at maximum load conditions.

Using the set of dimensions calculated, the total dead weight of the girder is calculated and compared with the assumed value. If the discrepancy is greater than 1%, the programme recalculates the section sizes based upon the current weight. The design of the actual sizes is written as a procedure and given the name 'prelim sizes'. Because it is possible to arrive at a set of trial dimensions based upon minimum Z values independent of the dead load, the section design cycle converges very quickly. It has been found that generally only three or four cycles are required.

At this stage in the programme, a single girder has been designed to withstand the HA uniform loading. The depths of the section and thicknesses of the bottom flanges are known at the inner supports and at the centre of the midspan. All spans must satisfy equation (5.1) and therefore the section dimensions at the outer supports will be the same as at the midpoint of the centre span.

5.3.5) Effects of HB Loading on Bridge Deck

The HB loading consists of an abnormal vehicle of up to 45 units (180^T) in weight, carried on a system of four axles as shown in Fig. (5.4a). One lane is loaded with type HB load only, and all other lanes are loaded with one third HA loading. To obtain the maximum longitudinal moment the abnormal vehicle is placed as near to the edge of the deck as possible. The width of the wheels is 15" and the centre of the edge wheel is placed 12" from the edge of the carriageway i.e., 4½" clearance. To produce the maximum moment the centre line of the bridge should be midway between the centroid of the load system and the first axle, as shown in Fig. (5.4a). This presents an unsymmetrical case for analysis which necessitates the whole of the bridge being analysed. The available core store is insufficient to enable this to be done and in the programme the load is placed symmetrically upon the bridge, together with one third HA load in the remaining lanes, in the centre span, as shown in Fig. (5.4b). BS 153 (1) allows the permissable stress to be increased by 25% under the abnormal vehicle loading condition. In order that the HA and HB loading condition may be compared in the final section sizes design, the abnormal vehicle has been reduced in magnitude by 20%.

The value of maximum bending moment obtained in this way will differ only slightly from the actual maximum bending moment because the centre span is generally large compared with the dimensions of the abnormal vehicle. In this way, considerable savings in both storage space and time for solution, are effected. To obtain maximum transverse moment the load is placed centrally in the carriageway at the centre of the midspan. One third HA loading is applied to the deck on either side, although these areas do not necessarily correspond to the actual lane widths, as shown in Fig. (5.4c).

Despite the fact that only half of the bridge is considered, the available fast store is still insufficient for this analysis. To reduce the number of joints in the equivalent grillage, the transverse diaph ragms have been replaced by seven diaphragms of equivalent stiffness. This procedure has been used previously in the analysis of a continuous varying section grillage, by Goldstein, Lightfoot and Sawko (10), where no noticeable loss in accuracy was incurred. It is assumed that the centres of the diaphragms are 2.5 x centres of webs. Equivalent properties about the seven imaginary centre lines are calculated. The longitudinal members of the equivalent grillage are considered to act at the centres of the box girders i.e., midway between the webs. The equivalent grillage for the analysis of HB loading is shown in Fig. (5.5).

The programme employs the method of analysis described in Chapter 2, but instead of using information from a data tape the computer derives all the required data automatically. From the design of a discrete girder, the dimensions at the supports and at the centre of the midspan, are known. The depth of the girders varies parabolically and the depth of the bottom flange is again assumed to vary parabolically. The centres of the equivalent grillage members are known, so that the programme is able to automatically number the grillage joints and calculate the member properties for each type of member. The numbering proceeds across the deck commencing at the outer support, as shown in Fig. (5.5b). As there are seven transverse diaphagms the total number of joints is 7 (nw - 1), where 'nw' is the number of webs. The member joint and type numbers are stored in array M. The eight member properties viz., cos α, sin α, L, T, C_{12}, C_{21}, K_{12}, K_{21}, are calculated for each type of member. The transverse members are considered first. The depths of the girders at the seven transverse diaphragms are found and I and J values calculated using this depth. The J values are found using Equation (4.3) for thin walled box sections, the median line acting at the midpoint of the flanges and webs. Only one value of I and J is required to calculate the bending and torsional stiffness, being $\frac{4EI}{L}$ and $\frac{GJ}{L}$ respectively for uniform members. At any one diaphragm all the members will have the same properties, therefore, there are seven types of transverse numbers. The properties for the longitudinal members are determined by calculating the flexibility influence coefficients for each type of member. The members are divided into twelve equal sections and the $\frac{1}{EI}$ values calculated at thirteen stations. Simpson's rule is used to integrate the various functions and the final member properties are derived using Equations (2.24) to (2.28). Each section of longitudinal member across the bridge is the same type and, as there are always seven transverse members, there are six types of longitudinal girder. There are a total of thirteen types of members in the half of the bridge being analysed. The eight member properties are stored in array P, which has 13 x 8 elements.

After calculating the member joint numbers and properties, the programme then proceeds to set up the complete stiffness matrix for the half of the bridge being considered. The upper half of the stiffness matrix band is stored in array S. Following the joint numbering system shown in Fig.(5.5b), the band width is 3 x no. of webs; the total number of joints is 21 (nw - 1) and total number of elements in array S is 21 (nw - 1) x 3 nw. The individual member slope deflection equations are given in Table (2.2), where the x and y axes lie along and across the bridge respectively, as shown in Fig. (5.5b). The members are considered in the order they are stored in array M and the stiffness matrix terms inserted into the S array, as described in Section (2.4).

The loading can only be applied to the grillage joints, and therefore it is necessary to express all the applied loading as an equivalent system of loads, acting at joints only. Normally the loading is expressed as end moments and shears, but to simplify the programme only the shear forces have been considered here. This does not lead to a significant error when considering long span bridges with the abnormal vehicle placed at the centre of the main span. The omission of the applied end moments will cause slight local discrepancies, which will be small in comparison with the overall bending moment. The applied loading is transferred to the joints by statics. The loads are first transferred to the longitudinal girder. For each girder a value of UDL, due to one third HA loading, is held in array HA, and the values of point loads from the abnormal vehicle are held in array AV. The loads on the longitudinal girder are then transferred by statics to the joints. The two loading vectors i.e., abnormal vehicle on edge of deck and abnormal vehicle at the centre of deck, are stored in array B.

The zero displacement conditions are next imposed, being at the piers a zero vertical displacement and along the transverse centre line a zero rotation about the y axis. The set of simultaneous equations are solved using procedure 'solve'. The elements in the loading array B are replaced by the corresponding joint displacements for the two loading cases. The displacements are substituted into the individual member slope deflection equations in Table (2.2), to obtain the terminal moments and forces for each member . The complete set of results for the two loading cases are printed out in tabular form.

5.3.6) Design of Final Section Sizes.

The maximum values of bending moment, produced by the HA and HB loading conditions, are now compared. If the greatest bending moment is produced by the HA loading condition the sizes already derived will be the final sizes. If, however, the HB load produces a higher moment, the section sizes must be redesigned. The procedure 'prelim sizes' is activated using the new values of live load bending moment. The first cycle calculates the dead load bending moment from the section sizes derived to carry the HA loading. The programme continues in the design loop until the assumed dead weight of a single girder is to within 1% of the actual dead weight.

The section sizes are derived by considering the bending moments at the intermediate supports and centre of midspan only. To design the prestressing force at other sections along the girders a bending moment envelope is required. This has already been established at 21 points along the girder under the HA conditions, but it is not known for the HB loading condition, as the abnormal vehicle has only been placed at the midspan of the bridge. If the

HB Loading condition is the criterion, a bending moment envelope is obtained by applying the same percentage increase that occurs at midspan, to the positive values of bending moment occurring at the 20 other points along the beam. The percentage increase at the inner support is similarly applied to the negative bending moment values.

The final section is designed at the 21 points along the girder up to the centre line. Using equation (5.1) the required I values at these points can be found. The thickness of the top flange is uniform overall the bridge and the depth of the section varies parabolically between midspan and supports, so that the only variable is the thickness of the bottom flange. This is found by commencing with the thickness of the bottom flange equal to the greatest thickness, either at midspan or support, and reducing this amount by 0.2 inches until the actual value of I becomes less than the required value, given by equation (5.1). The previous value of thickness is taken, therefore the bottom flange thickness is calculated to the nearest 0.2 inches above. The procedure 'find I' is used to calculate the actual I value at each stage. The depth of the centroid is also found at each trial, so that the value of Z_1 and Z_2 can be calculated for the final section. In this way, the section dimensions and properties are calculated at the 21 points along the girder and placed into array M1.

The magnitude of the prestressing force is obtained by eliminating c from equations (5.3) and (5.4) :-

$$P = \frac{A\,(f_b\,Z_1 + f_t\,Z_2)}{Z_1 + Z_2} \qquad (5.18)$$

The values of f_b and f_t being the required prestress, are found from equations (5.12) and (5.13). The interpretation of these equations depends upon the sign of the bending moment i.e., hogging or sagging, which is determined by the sign of the dead load bending moment. In this way the required prestress at 21 points along the girder is found. This value will vary at each point, depending upon the required stresses and Z values at that point. The variation in prestress is developed by curtailing the cables as they are no longer required. The prestress at each point is provided by a number of cables which, it is assumed, are all equally stressed. Therefore, the actual prestress will be slightly higher than that required by equation (5.18), being equal to the number of cables multiplied by the force in each. Where there is a permissable zone in which the line of action of the cable should lie, this should not effect the stress conditions but at the critical intermediate support or midspan section, where the cable probably has a unique position, there is a danger of overstressing immediately after the prestress is applied and before the losses have occurred. In practice the whole of the prestress is not applied simultaneously and therefore a percentage loss will have taken place before the last cable is stressed.

The cable eccentricity must satisfy four conditions in order that the permissable stresses shall not be exceeded. These conditions are obtained by substituting for f_t and f_b from equations (5.3) and (5.4), in the relations (5.5) and (5.8). For positive bending moments :-

$$e > \frac{Z_1}{A} - \frac{Z_1 f_{ct}}{P} - \frac{M_{min}}{P}$$

$$e > \frac{-Z_2}{A} + \frac{Z_2 f_{mint}}{P} - \frac{M_{min}}{P}$$

$$e < \frac{Z_1}{A} - \frac{Z_1 \, f_{min}}{R_o \, P} - \frac{M_w}{R_o \, P}$$

$$e < -\frac{Z_2}{A} + \frac{Z_2 \, f_{cw}}{R_o \, P} - \frac{M_w}{R_o \, P} \qquad (5.19)$$

For negative bending moments :-

$$e < -\frac{Z_2}{A} + \frac{Z_2 f_{ct}}{P} - \frac{M_{min}}{P}$$

$$e < \frac{Z_1}{A} - \frac{Z_1 f_{mint}}{P} - \frac{M_{min}}{P}$$

$$e > -\frac{Z_2}{A} + \frac{Z_2 \, f_{minw}}{R_o \, P} - \frac{M_w}{R_o P}$$

$$e > \frac{Z_1}{A} - \frac{Z_1 f_{cw}}{R_o P} - \frac{M_w}{R_o P} \qquad (5.19a)$$

By using inequalities (5.19) the upper and lower limits of the line of action of the cable are found. These are placed into array M1, which finally holds in the columns: 1) positive live load moment 2) negative live load moment, 3) dead load moment, 4) Z_1, 5) Z_2 6) cross sectional area of concrete, 7) prestressing force, 8) lower cable limit 9) upper cable limit, 10) thickness of bottom flange, 11) overall depth.

5.3.7) Design of Cable Profile

The zone in which the prestressing force must act is given by equations (5.19), but because of the indeterminacy of the girder, secondary bending moments act, which cause the line of action of the cable to be displaced. Therefore, although the actual cables lie within the required zone, the displaced line of thrust may lie outside this zone.

The resultant line of thrust is found using influence coefficients, the method used is that given by Morice (31) and is similar to that used to determine the member properties, described in Section (2.3.4). The girder is rendered statically determinate by removing the continuity over the intermediate supports, as shown in Fig.(5.6a). Arbitrary moments x_1 and x_2 are applied at these joints to produce moments m_1 and m_2, as shown in Fig.(5.6b). The free bending moment upon the released structure m_o is obtained by multiplying the prestressing force by the eccentricity at each station. Two simultaneous equations for the solution of x_1 and x_2 are derived, which can be written in the form :-

$$\begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (5.20)$$

where

$$f_{11} = \int_s \frac{m_1^2}{EI} ds$$

$$f_{22} = \int_s \frac{m_2^2}{EI} ds$$

$$f_{12} = f_{21} = \int_s \frac{m_1 m_2}{EI} ds$$

$$u_1 = \int_s \frac{m_1 m_o}{EI} ds$$

$$u_2 = \int_S \frac{m_2 m_o}{EI} ds$$

The solution of equations (5.20) gives the value of arbitrary constants x_1 and x_2. In this particular case x_1 and x_2 will have the same value because the girder and loading are symmetrical.

The final moment distribution is given by :-

$$M = m_o + m_1 x_1 + m_2 x_2$$

but $m_o = P \times e$ hence

$$e' = e + \frac{m_1 x_1 + m_2 x_2}{P} \qquad (5.21)$$

where e is the actual cable eccentricity

e' is the line of action of the cable

P is the value of prestressing force.

If the cable was placed along the line of e', there would be no secondary moments caused and the line of thrust would correspond to the actual cable position. This is known as the concordant cable profile.

The cable profile cannot be derived directly, so that the computer must proceed in a series of 'trial and error' steps. An initial cable profile is chosen to have zero eccentricity at the outer supports, and lie at centre of the prestressing zone at the inner supports and centre of the midspan. If the cable zone passes outside the section the cable is located midway between the maximum practical eccentricity and the limit within the section. The cable is assumed to

WHY ?

be straight between these fixed points. The line of action is then found by setting up equations (5.20) and solving for x_1 and x_2, which are then substituted into equation (5.21). The eccentricity of the line of thrust at the inner supports and midspan is then checked to see if this falls within the prestressing zone. If these conditions are not satisfied the cable is then either raised or lowered at the outer supports, according to the required direction, in 2" increments until the line of action of the cables falls within the cable zone. The final position can be adjusted by hand, as the cable may be raised or lowered at the inner supports by any required amount. The resultant line of thrust is not affected by this transposition.. This important property of continuous prestressed concrete beams, first enunciated by Guyon, may be stated as follows.

"In a continuous prestressed concrete beam, if the prestressing force is displaced vertically at any of the intermediate supports by any amount, but without alteration to the intrinsic shape of the line of the force between the supports, the resultant line of thrust is unchanged."

5.4) <u>Preparation of Data</u>

The data is headed by a title describing the current bridge to be designed and this title is printed at the head of the computer output. The remainder of the data consists of :-

1) For each span is given - the length of the spans followed by the base length of the full parabola for that span. Thus end spans can be any division of parabola but usually they are one half for appearance. The centre span and parabola base length will be the same.

2) Ratio 'R' of moments of inertia at inner support and centre of main span. From Equn. (5.1) $\frac{I_{support}}{I_{centre}} = (1 + R)^2$

3) Total width of deck. If is assumed that the deck carries one way traffic only. Bridges with two carriageways have to be designed carrying the same load in both halves and can therefore be considered as a one way bridge by taking onehalf the total width of deck.

4) Width of traffic lanes. It is assumed that all lanes are of equal width.

5) Number of traffic lanes, including hard shoulder, if any.

6) Width from inner edge of deck to first lane or hard shoulder.

7) Width from outer edge of deck to traffic lane.

8) Number of units of abnormal vehicle. Maximum load = 45 units = 180^T

9) Modulus of Elasticity E.

10) Modulus of Rigidity G.

11) Minimum thickness of web required to accommodate prestressing cables.

12) Permissable compressive stress at transfer.

13) Permissable tensile stress at transfer.

14) Permissable compressive stress under working load.

15) Permissable tensile stress under working load.

16) Permissable principal tensile stress at ultimate load.

17) Percentage loss of prestress in cables.

18) Prestressing force per cable.

5.5) Computer Output

The full output consists of :-

1) Full analysis of structural model under both HB loading conditions.

2) Thickness of top flange

3) Thickness of webs

4) Centres of webs

5) Details of section at 21 points along the girder

6) Position of prestressing cable. ← Is the no. of prestressing cables being output as well ??

Item (1) enables the designer to calculate the required reinforcement in the transverse direction. The second loading case, with the abnormal vehicle placed at the centre of the deck, produces the maximum transverse moment. The bending moment is given at the points of the equivalent diaphragms, shown in Fig.(5.4a), and the joint numbering follows the system, shown in Fig.(5.4b). Item (5) gives in tabular form the section details at 21 points along the girder. Points 1 to 11 and 11 to 21 correspond to ten equidistant section along the end span and centre span respectively, point 11 being the inner support. At each point is given the overall depth, prestressing foce, upper and lower prestressing zone and thickness of bottom flange. Item (6) gives the position of the prestressing force at the supports and centre of the midspan. Although the line of the cables has been assumed to be straight between these points, the cable profile will be relatively flat and in practice a slight curvature would be introduced to, facilitate easier positioning of the cables.

5.6) Calculation Time

The total size of the design program is 5400 'words' when using the Whetstone translater. The storage required to use the Kidsgrove translater is approximately 10,000 words, which is in excess of that permitted by the KDF9 machine. Consequently, all development work had to be run using the slower Whetstone translater and it was found that the complete design of a bridge could not be accomplished in less than 45 minutes. This amount of time is impractical and expensive for normal running of such a program. The English Electric Leo Marconi Co. are at present preparing the necessary softwear to enable large programs to be accommodated within the available store. To do this it is necessary to divide the program into smaller segments, which are stored on magnetic tape. The segments of the program are transferred into the fast store of the machine as required and it is not necessary to hold the complete program within the store, which results in a saving of space. At the time of writing the design program, the segmentation softwear was not available and the program had to be segmented manually, as a temporary measure. The program was divided into three separate programs. The results output from one program form the input data for the next stage, and in this way it is possible for the complete analysis to be accomplished using approximately ten minutes of computer time. The amount of data preparation and total time required are considerably increased by using this method.

Some runs of the complete program were also obtained using the Atlas machine, operating at the National Institute for Research in Nuclear Science, Chilton, Berkshire. The compiled version produced by this machine is extremely efficient and operates at some sixty times faster than under the Whetstone controller. A complete design can be obtained in approximately two minutes, including the time required for translation.

5.7) Conclusions and Future Work.

After developing the program for the automatic design of three span prestressed concrete bridges, the following conclusions have been drawn. These observations are also applicable to all large design programs.

1) The program is able to produce a complete design extremely rapidly and economically. The short time required for solution is the main advantage, although the solution obtained is not exact because certain simplyfing assumptions have been made. Several alternative design schemes can be investigated in a short space of time, and a 'best' solution obtained.

2) In order to run the program economically, a computer with a large fast random access store is required. This is to contain the translated version of the program and large amounts of information evolved by the program during calculation. If insufficient store is available to produce an efficient machine code version, the time required for solution will make the program impractical for normal use.

3) An efficient design can be produced for statically indeterminate structures. The design is inherently difficult since the answer in

effect must be known, or assumed, beforehand. The program employs a method of successive correction for solution and is able to rapidly perform a sufficient number of cycles to obtain an accurate solution.

4) The programs have to be written specifically to design one type of structure and they are not as general in application as are analysis programs. The amount of data required for the computer is, however, substantially reduced.

5) The development of design programs is both lengthy and costly. A large amount of computer time is required for testing and correcting each section. It is unlikely that such programs will be developed and used exclusively by one organisation because of the limited application and expensive development costs. Large design programs could be a practical proposition, if made available to a large number of users through national computing centres and program libraries.

The role of electronic computers in all branches of civil engineering is gradually changing, as larger machines with increased hardwear become available. The automatic design program for three span bridges in its present form can be extended to accommodate a varying number of spans and also allow for the effects of skew. Once this has been achieved, one program will be able to design many types of bridges, fromsingle to multispan, uniform or parabolic profile, right or skew. From programs of this type, in which the number of spans, span ratio, girder profile, girder spacing and diaphragm spacing are all fixed either by the data or within the program, the future development will be towards programs which produce an optimum design for the problem under consideration. The data would consist of the overall length of the bridge and current materialcosts. From this data the program will examine the effects of the number of spans, span ratios, relative stiffnesses, girders spacing and profile, upon the cost of the bridge, and produce the most economical design for the given problem. This design would then form the basis of the actual design as other factors, such as appearance and construction techniques, need also to be considered.

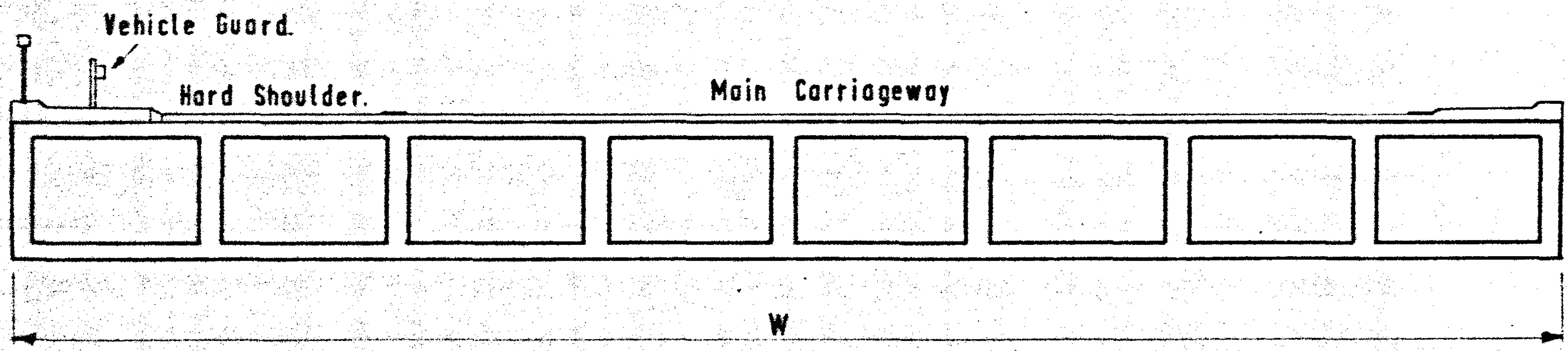


Fig. 5.1. Transverse Section of Prototype.

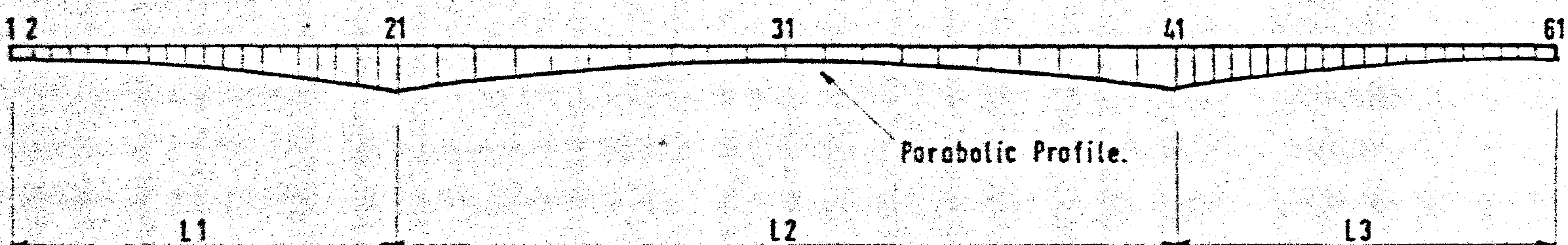


Fig. 5.3 Joint Numbers for Influence Line Analysis.

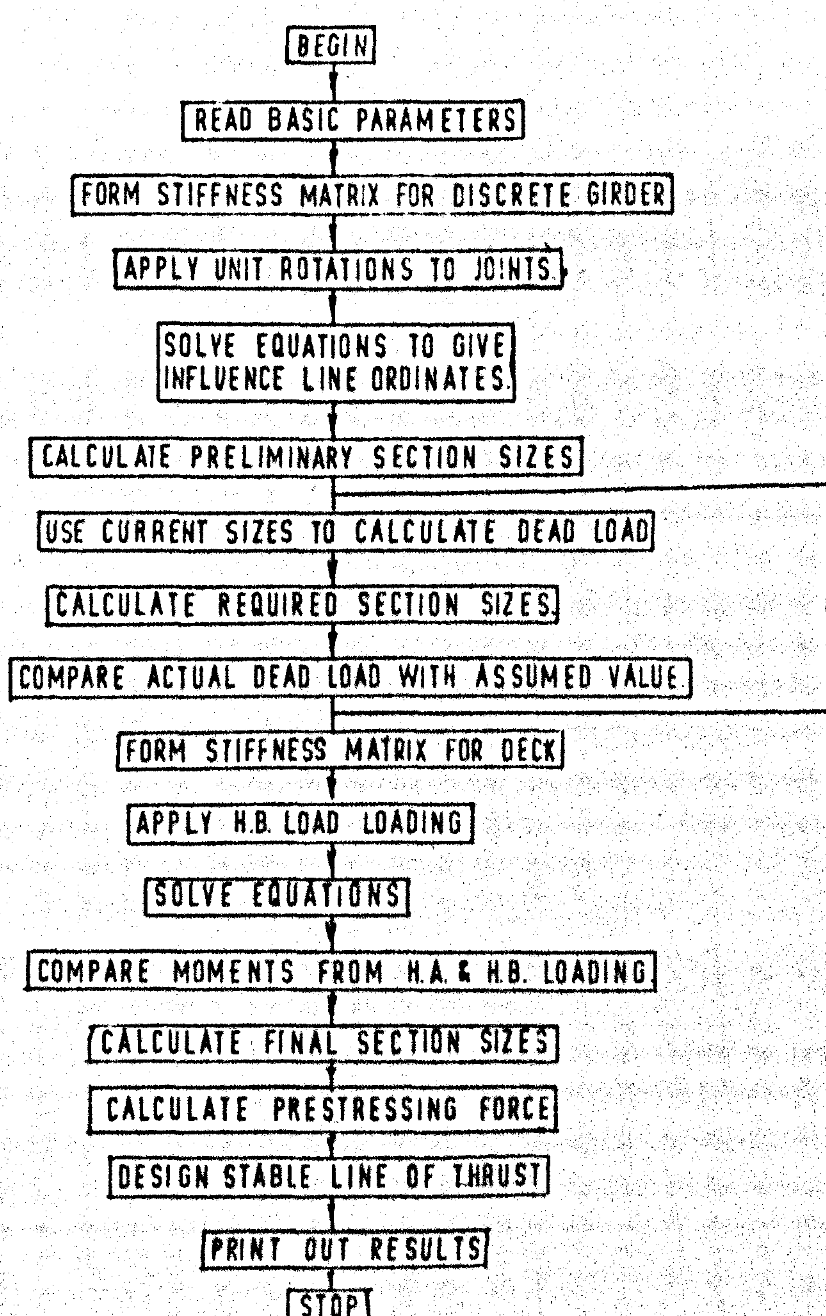
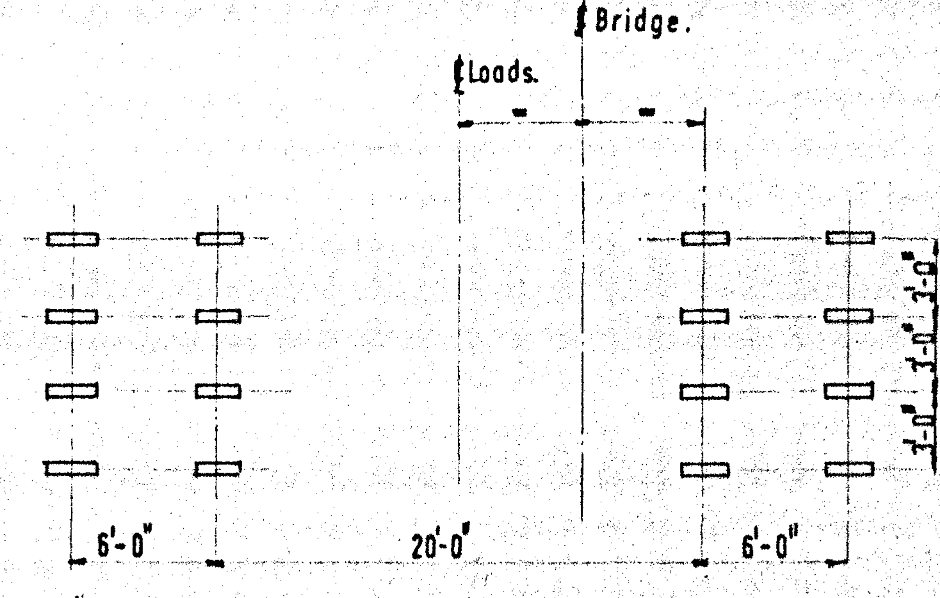
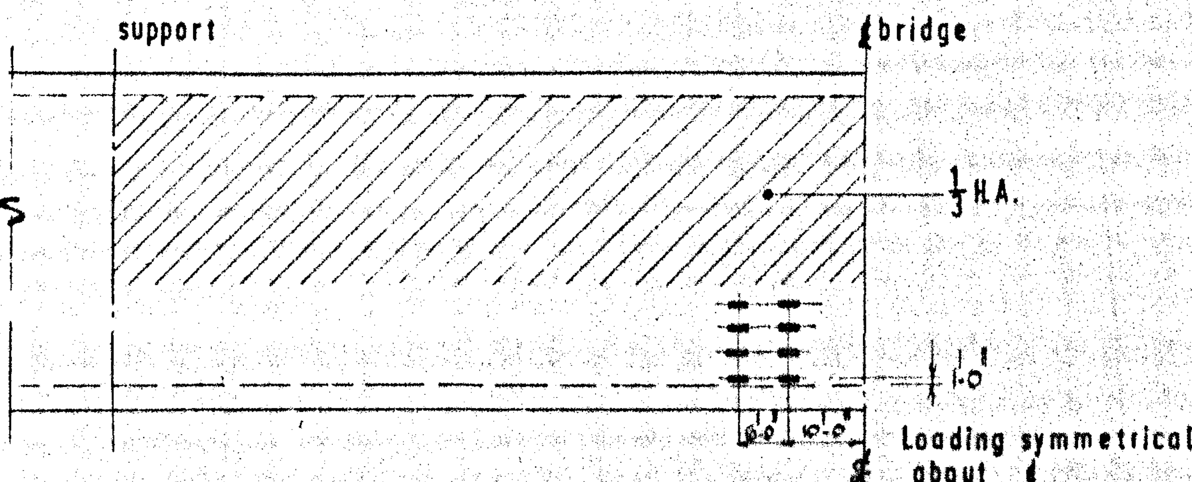


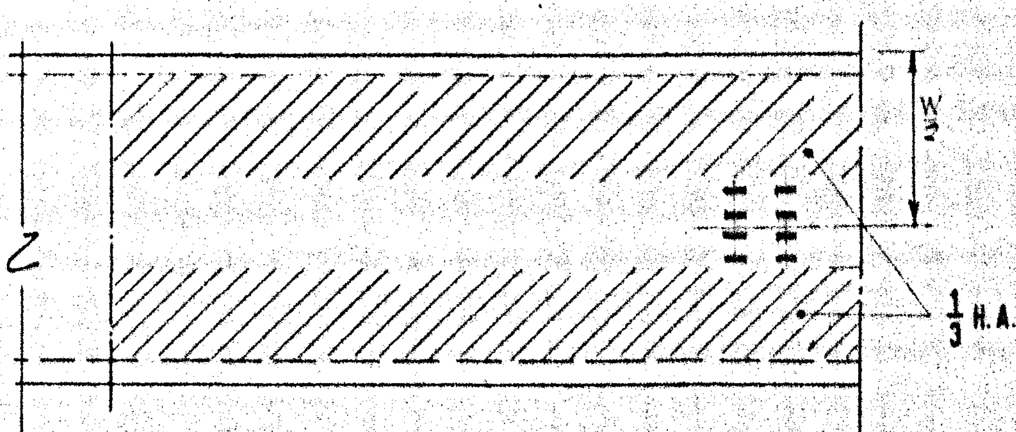
FIG.5.2 Design Program Flow Diagram



[a.] Position of Abnormal Vehicle for Maximum Moment.

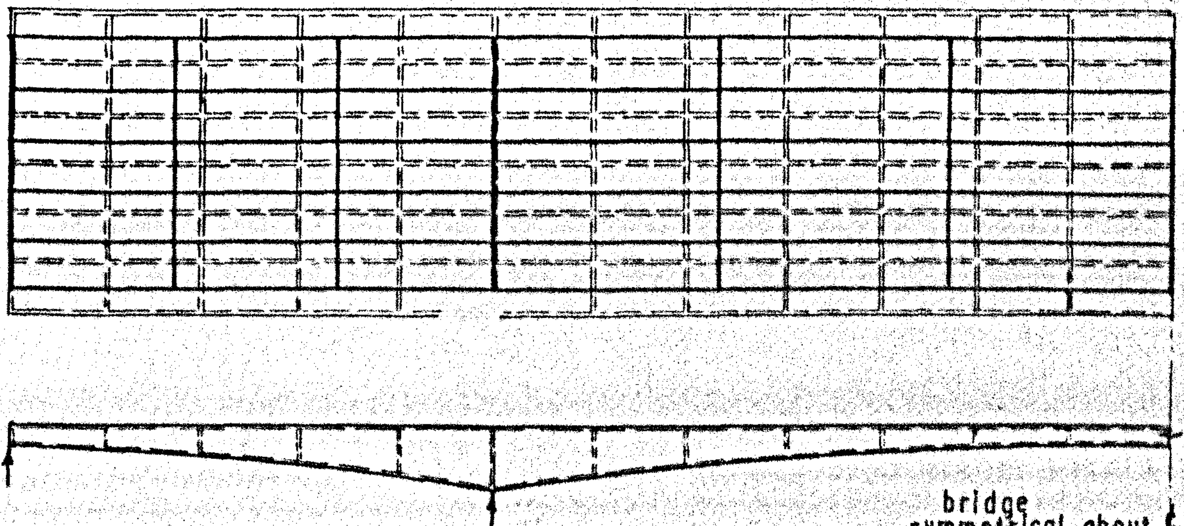


[b] Actual Loading Applied for Max. Long. Moment.

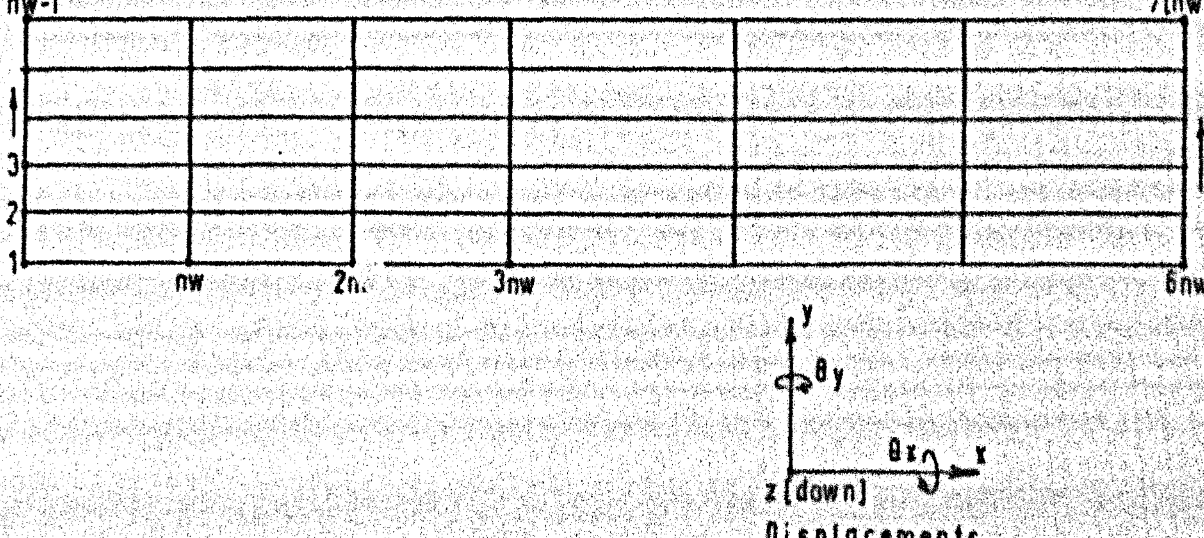


[c] Actual Loading Applied for Max. Transverse Moment.

Fig. 5.4 H.B. Loading Condition.

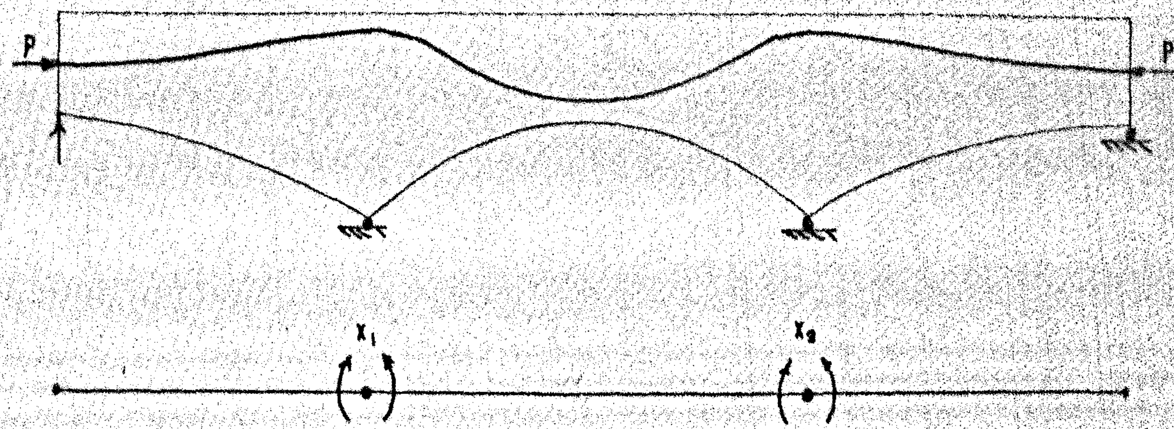


[a] Centre Lines for Grillage Analysis.

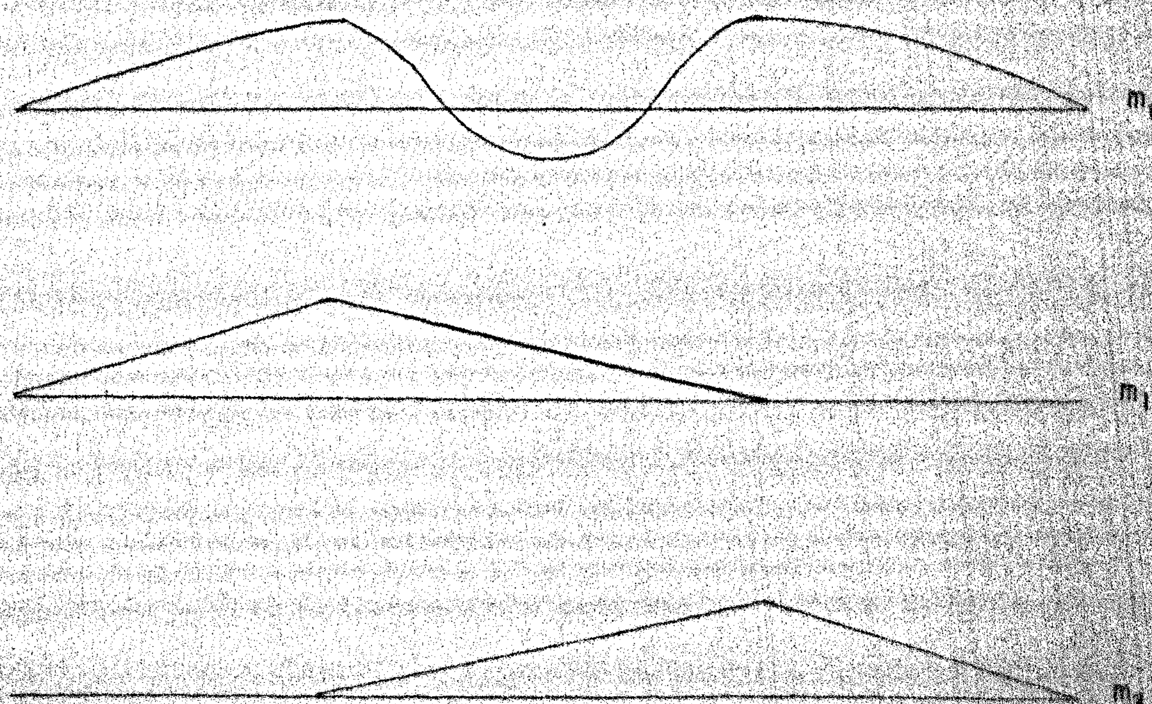


[b] Joint Nos. & Displacements.

Fig.5.5 Computer Analysis of Bridge Deck.



[a] Release System.



[b] Moments

Fig. 5.6 Determination of Prestressing Line of Action

Chapter 6

Investigation into Least Weight Design of Three Span Prestressed Concrete Bridges.

6.1) Introduction

The total cost of a bridge is governed by many interdependent conditions which cannot be studied in isolation from each other. The design must satisfy several conditions apart from structural safety; the structure should be aesthetically pleasing and the system of construction must be feasable. The cost of the superstructure is dependent upon many variable factors, namely - geographical location, relative costs of various materials and labour, method of construction, complexity of structural geometry, imposed site conditions, degree of standardisation and available plant.

For each multispan bridge there will be an optimum layout for the number of spans, relative lengths of spans, relative stiffness of spans and spacing of girders and diaphragms. These parameters are influenced by the relative cost of materials, and therefore the determination of the most economical system can only be achieved for individual schemes. The initial choice of layout parameters does, however, determine to alarge extent the overall cost of a bridge structure. An investigation into the effects of relative span lengths and stiffnesses upon the cost of individual bridge girders has been carried out by Aziz (22). The investigation was carried out using a program for the design of a discrete three span prestressed concrete bridge girder under the action of the HA loading condition only. The quantities of concrete and prestressing cables were calculated and a cost derived from these items. Two ratios of unit cost were considered, the cost of the steel per ton being 9.72 and 7.0 times that of the concrete cost per ton. In this case it was found that the span ratio had the most influence upon cost and that an optimum value of 0.3 i.e., the end spans being 0.3 x overall length, produced the most economical design over a wide range of span stiffness values. The span stiffnesses were found to have a lesser effect upon the total cost. It was also found that, within the range investigated, the change in relative unit costs had little effect upon the choice of other parameters.

The investigation was based upon assumed costs of steel and concrete and therefore the results must be interpreted for these values only. The program used was only able to consider the effects of the HA loading upon a discrete girder, and the effects of the abnormal vehicle HB loading were not considered in the investigation.

By using the program described in Chapter 5, it was possible to undertake a more rigorous study of the effects of span ratio and relative stiffness, upon the overall efficiency of a bridge. The total weight of the bridge has been used as a measure of efficiency and a set of 'least weight' parameters have been determined. It should be stressed that the least weight design is not necessarily the most economical, as only the volume of concrete has been considered.

6.2) Investigation

For the investigation all the bridges were of the same form, consisting of three span symmetrical prestressed concrete right decks, composed of hollow box girders. The overall span was 500 feet and the width 48 feet. The bridges comprised three traffic lanes, each 13 feet wide, an edge strip at the left hand side of 5 feet to the traffic lane and an edge strip of 4 feet at the right hand side. This was considered to be a 'typical' single carriageway bridge being part of a motorway system. It was assumed that the carriageway carrying the traffic in the opposite direction was structurally independent.

The spans of the bridge were chosen to give a range varying from three equal spans, to the end span being half the centre span. The variation in span lengths was expressed as the ratio of end span ' l' to the total span 'L', so that :-

$$\phi = \frac{l}{L}$$

The range of spans investigated is given in Table (6.1)

Table 6.1.

Span Ratio ϕ	End Span Ft.	Central Span Ft.
0.25	125	250
0.275	137.5	225
0.3	150	200
0.33	166.6	166.6

The range of stiffnesses investigated for each span ratio were, from R = 0 to R = 4, where R is the ratio of moment of inertia at the centre of the main span to the support, as given by Eqn. (5.1) thus :-

$$\frac{I_{support}}{I_{centre}} = (1 + R)^2$$

The range of depths thus varied to give a bridge composed of uniform girders when, R = 0, to a bridge consisting of arch like girders when, R = 4.

All other variable parameters in the design were fixed, so that the effects of span ratio and relative span stiffnesses could be studied in isolation. The centres of the main girders were 8 feet and the thickness of the webs 18 inches in all cases.

6.3) Results

The complete set of results obtained are given in Table (6.2). The variation in weight for a changing R value and fixed Ø value, are plotted in Fig.(6.1).

6.4) Discussion of Results.

From Fig.(6.1), it can be seen that the least weight for all span ratios is given when the variation in moment of inertia R = 0.5 → 1.0. This range of R values results in a bridge with only a slight curve in the longitudinal direction, the depth at the inner support being approximately 1.5 to 2.0 times that at the centre of the main span. Within thisrange the structure is able to develop the greatest stresses at all points along the girder and the most efficient use of materials is obtained.

If a continuous girder of uniform depth is employed over large spans the structural efficiency decreases considerably. The total weight when R = 0 is 8.25% greater than the minimum weight when ϕ = 0.33 and 13.5% greater when ϕ = 0.25. This decrease in structural efficiency is likely to be particularly noticeable in long span continuous bridge girders, as the majority of the load is due to self weight. Hence any materials, which are not being subjected to the maximum possible stresses, will add unnecessarily to the design bending moments. It was found that for a central span of 250 feet the dead load bending moment was approximately twice the live load bending moment. Thus it can be seen that quite large economies in concrete quantities can be effected by introducing only a slight variation in depth along the beam such that R = 0.5. It is probable that, in this case, the shuttering costs would be slightly higher but this increase should be more than offset by the saving in concrete quantity.

As the ratio of moment of inertia at inner support to moment of inertia at centre of main span is increased above R = 1, the total weight of the bridge begins to increase above the minimum value indicating that the structural efficiency is decreasing. Within the range of R values investigated, it was found that when $R = 4$ the total weight of the bridge, when compared with the minimum value, was increased by 6.8% when $\emptyset = 0.25$ and 11.75% when $\emptyset = 0.33$. The value of $R = 4$ will give an inner support depth equal to approximately five times that at the centre of the main span, which would probably exclude such a bridge for aesthetic, as well as economic, reasons.

The effect of the span ratio $\emptyset$ can be seen clearly from Fig. (6.1). The least weight is obtained when all three spans are of equal length, resulting in the minimum bending moment in all spans. In practice it is seldom possible to arrange the spans in this way, and therefore, the central span should be made as near to one third the total span, as possible. Even if a span ratio of $\emptyset = 0.25$ is chosen, the total weight when $R = 1$ is still less than the total weight when three equal spans are used, with a uniform depth continuous girder.

Throughout the whole range of span and depth ratios investigated, it was found that for the type of bridge deck employed (continuous soffit slab giving a torsionally stiff bridge deck), the HA uniform loading condition always produced the maximum bending moments. A bridge of total length 200 feet and having a central span of 100 feet, was also designed by the program where again it was found that the HA loading was the criterion. Bridges of 200 feet overall length or less are unlikely to be constructed with three spans, as this would be uneconomical. Therefore, for all large three span bridges, the HA loading is always going to produce the maximum longitudinal bending moment, provided the deck is fairly stiff transversely. Morice (36) has also found that, for simply supported spans of over 40 feet, the normal uniformly distributed loads produce the greater longitudinal moment and it has been shown in Chapter 4, that a continuous grillage provides a greater redistribution of loads than a simply supported slab. The maximum transverse moment will be produced by the abnormal vehicle HB loading system. This case is automatically analysed by the design program.

6.5) Conclusions

From the results obtained the following conclusions have been drawn for the range of bridges investigated.

1) The least weight design is obtained when a bridge of three equal spans is employed with a moment of inertia ratio of R = 1.0.

2) The choice of R value has the greatest influence upon the total weight of the bridge. A variation in total weight of up to 13.5% can occur depending upon the value given to R.

3) The ratio of span lengths has a large influence upon the total weight, which may vary by up to 11.75%, depending upon the value given to Ø.

4) The uniform HA loading produces the maximum longitudinal bending moments for large multispan bridges having torsionally stiff deck systems.

Before definite trends can be established, it will be necessary to carry out further investigations over a wider range of variables. For three span bridges the study should include overall spans of between 100 feet and 1000 feet. For these results to be applied in practice, it will also be necessary to investigate four and five span bridges. Such investigations require many runs of computer programs, and therefore, they are quite time consuming.

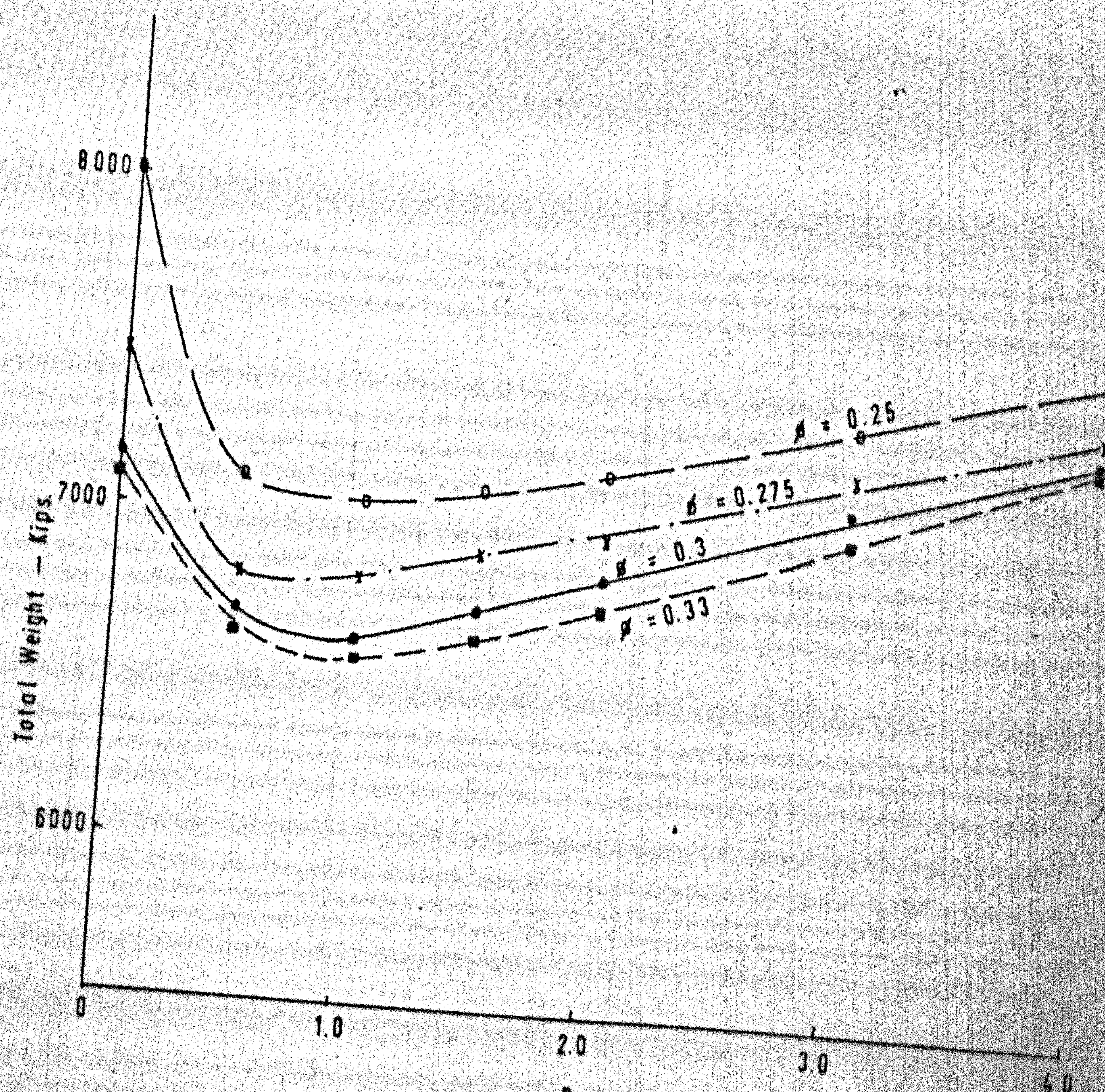


Fig. 6.1 Total Weight v. R

Table 6.2		
Ø	R	Total Weight Kips
0.25	0	8000
	0.5	7050
	1.0	7030
	1.5	7100
	2.0	7160
	3.0	7360
	4.0	7530
0.275	0	7480
	0.5	6800
	1.0	6820
	1.5	6900
	2.0	6980
	3.0	7200
	4.0	7340
0.30	0	7150
	0.5	6600
	1.0	6630
	1.5	6730
	2.0	6850
	3.0	7100
	4.0	7300
0.33	0	7100
	0.5	6540
	1.0	6530
	1.5	6640
	2.0	6750
	3.0	7000
	4.0	7300

Chapter 7.

General Conclusions and Future Work

7.1) General Conclusions

7.1.1) The series of model tests carried out, show that the analysis programs are capable of accurately analysing structures with varying section properties. The grillage is also a valid model for the idealisation of continuous plated grillages and cellular structures, having a spanwise variation in depth. The programs can accommodate any variation in section, occuring throughout the full length of the member or only at certain sections along the member. The programs make efficient use of storage space and have a high operational efficiency. The preparation of data is relatively straight forward and the user is not required to have a knowledge of how the programs operate.

7.1.2) The bridge design program is able to produce automatically an accuate design inwhich all the design conditions have been considered. The calculations are performed extremely rapidly by the computer and consequently it is possible to obtain a far more accurate design, than if hand methods were employed. The design value of dead load bending moment is evaluated to within 1% of the actual dead load moment and, because of this, an efficient design is produced, in that the minimum of materials are employed.

Automatic design programs cannot normally be as general in application as analysis programs, because the layout of the structure and applied loading are contained within the program. The amount of data can be quite small and the complete list of design data can be specified by less than twenty parameters. The length of the program will, however, be quite large, and it is this fact that has largely prevented the advancement of design programs in the past. The automatic design program is therefore limited inapplication and costly to develop. To obtain economical useage of such programs it is necessary that they should be available to a large number of users, either through computer manufacturers, or at national computing centres.

Recent developments in computer softwear have led to the development of 'problem orientated languages'. These new systems are written specifically for one class of user, unlike Algol which is a 'universal' language that can be used for many problems in all fields of science and technology. A structural analysis and design system has been developed at the Massachusetts Institute of Technology, and is and is known as STRUDL (38). By using this system, an engineer is able to design many types of structures. It is not necessary for the user to have knowledge of complicated programming languages, as all the program instructions are given in familiar terminology. The system is unable to carry out a design fully automatically and the user must insert further instructions as the program proceeds. He must, for instance, ask for the stresses to be computed at critical points and, if necessary, adjust the structural sizes and re-analyse. Such design work would be carried out through a remote terminal facility, possibly in the design office itself, and in this way the engineer has an extremely powerful tool to assist in his design work. The system has to be able to deal with many types of structures and problems, and this wide application leads to some loss in efficiency. By contrast, the automatic design program described here, is only able to consider onetype of structure, viz. three span bridges. The design is, however, carried out completely automatically and "rigorously" and the completed design is very "accurate" and detailed. It is possible to prepare detail drawings from results, all dimensions and prestressing cable details being given. The main advantages of such programs are that the solution is obtained completely automatically, and that it is very detailed.

It is felt that both these basically different design approaches have a place in future design operations. The design system because of its general applicability and the automatic design program because of its rapid and accurate solution.

7.2) Future Work.

7.2.1) The basic grid frame and plane frame programs can be extended to take into account several other effects. One such modification, to enable influence lines and surfaces to be derived, has been carried out and reported here. Sawko (37) has shown how the grid frame program can be extended to allow for the effects of elastic retraint at joints. The use of this program would be beneficial when analysing foundation rafts and earthwork structures.

Facilities can be incorporated which would take into account the effects of pinned ended members and pinned joints. The latter case exists in bridge decks consisting of simply supported suspended slabs with cantilever supports. Although, in this case, the bridge has an overall statical determinacy, the individual elements are still highly redundant and the complete bridge must be considered for an accurate analysis.

The method of analysis employed by the plane and grid frame programs can be used for solution of space frame structures with varying section members. Such a program would be applied to the analysis of building frames, when the interaction between plane frames has to be considered, or in bridges, where it is necessary to take into account the effects of elastic displacements of column or frame supports. A useful modification to the grid frame program would allow for certain joints to have six degrees of freedom, whilst all the remaining joints would still only have three degrees of freedom. Such a program would be capable of analysing bridge decks in which the displacement at supports could be taken into account. The amount of additional storage space required in this case would be quite small, as only those joints capable of three dimensional movement would require additional terms in the stiffness matrix.

The programs described in this thesis all make important economies in the method of storing the stiffness matrix elements, only the top half of the diagonal matrix band being held. This band does, however, contain zero elements and further economies in storage space could be introduced, bystoring only the non-zero stiffness matrix elements.

When a structure is analysed by computer program it is usual to consider several cases of alternative loading. This usually consists of dead load and several alternative live load cases. The output for each of these cases is given separately, and the alternative loading cases are combined by hand to obtain the worst case. By storing the output for each loading case within the computer, it would be possible for the results to give the combined effect of two, or more, alternative loading cases.

7.2.2) The design program for the design of three span bridges can be extended, to include for varying numbers of spans, from one up to five. The effects of skew could also be included. These modifications would produce a program with a wider field of application. Future design programs are likely to undertake more of the intial planning and layout work. To obtain an optimum scheme for a bridge many factors of cost and prevalent conditions have to be considered. Computers are ideally suited to this type of work, as they are able to operate at high speeds and consider many alternative schemes in a short space of time.

The range of computer hardwear is gradually increasing, and the use of this equipment will enable automatic design programs to perform more of the routine work, normally undertaken in the design office. The use of graph plotters will enable design programs to produce certain results in the form of 'working drawings'. These machines are also able to plot results in the form of bending moment and shear force diagrams. The programs themselves can be increased in scope, but this will also bring about an increase in program size. At the moment it is the size of such programs, that is governing development.

List of References.

1. British Standards Institution
B.S.153 Part 3A:1954, Girder Bridges. Section A: Loads.

2. British Standards Institution
CP115. 1959. The Use of Prestressed Concrete in Buildings.

3. Robertson, R.G. "Semi Graphical Integration Applied to the Analysis of Rigid Frames". The Structural Engineer, Vol. 27, No. 11, November, 1949.

4. Rowe, R.E. "The Electronic Computer: A New Tool for Structural Engineers". The Institution of Structural Engineers. Fiftieth Anniversary Conference, Session 3A, October, 1958.

5. Fok, D.Y.K. and Mosuve, T.F. "Analysis of Non-prismatic Continuous Structures" American Society of Civil Engineers, Vol. 92, STI, February, 1966.

6. Diwan, A.F.S. "Three Moment Equation for Variable Depth Beams" American Society of Civil Engineers, Vol. 90, ST6, December, 1964.

7. Sami, S. "The Use of Electronic Computations in the Analysis of Continuous Girder and Rigid Frame Bridges with Variable Moments of Inertia". Proc. 1st Conference of Electronic Computation. American Society of Civil Engineers, 1958.

8. Sawko, F. "Determination of Influence Lines and Surfaces by Electronic Computer". The Structural Engineer, Vol. 42, No. 8, August, 1964.

9. Morice, P.B. and Little, G. "The Analysis of Right Bridge Decks Subjected to Abnormal Loading". Cement and Concrete Association, July, 1956.

10. Goldstein, A., Lightfoot, E. and Sawko, F. "Analysis of a Three Span Continuous Grillage Having Varying Section Properties". The Structural Engineer, Vol. 39, No. 8, August, 1961.

11. Hendry, A.W. and Jaeger, L.G. "The Analysis of Grid Frameworks and Related Structures". Chatto and Windus, London, 1958.

12. Sawko, F. and Saha, G.P. "A Critical Review of Existing Methods for Bridge Deck Analysis". Indian Concrete Journal, August, September, 1965.

13. Madu, R. "Simply Supported Right Bridge Decks with Structural Members of Spanwise Variable Stiffness Under Concentrated Loads". Ph.D. Thesis, University of Southampton, 1962.

14. Henderson, J. C.deC., Edwards, A.D. and Munro, J. "Influence Coefficient Methods of Linear Structural Analysis". Symposium on the Use of Computers in Structural Engineering", Southampton, September, 1959.

15. Livesley, R.K. "Matrix Methods of Structural Analysis". Pergamon Press, 1964.

16. Krynicki, E.J. and Mazurkiewics, R. "Frames of Solid Bars of Varying Cross Section". American Society of Civil Engineers, Vol. 90, ST4, August, 1964.

17. Livesley, R.K. "Analysis of Rigid Frames by an Electronic Digital Computer". Engineering, Vol. 176, August, 1953.

18. Lightfoot, E. and Sawko, F. "Structural Frame Analysis by Electronic Computer: Grid Framework Resolved by Generalised Slope Deflection". Engineering, Vol. 182, January, 1959.

19. Litton, E., Roper, J.S. and Thompson, T. "Computer Calculations for High Buildings in Hong Kong". International Symposium; The Use of Electronic Digital Computers in Structural Engineering, University of Newcastle, 1966.

20. Sawko, F. and Willcock, B.K. "Computer Analysis of Bridges Having Varying Section Properties". To be published.

21. Portland Cement Association, Chicago. "Continuous Hollow Concrete Bridges". 2nd Edition.

22. Aziz, E.M. "The Design of Continuous Prestressed Concrete Bridge Girders by Computer". M.Sc., Thesis, London University, 1963.

23. Fadeeva, V.N. "Computational Methods of Linear Algebra" Dover Publications, 1959.

24. Cram, A.A. and Hall, A.S. "Influence Lines for Non-uniform Continuous Beams" Civil Engineering Transactions, Institute of Engineers, Australia, Vol. CE6, No. 1, March, 1964.

25. Jenkins, W.M. "Influence Line Computations for Structures with Members of Varying Flexural Rigidity Using the Electronic Digital Computer". The Structural Engineer, Vol. 39, No. 9. September, 1961.

26. "The Cementing of Perspex Acrylic Materials" Imperial Chemical Industries Ltd.,

27. Kantorovich, L. and Krylov, N. "Approximate Methods of Higher Analysis". P. Noordhoff Ltd., Groningen, 1958.

28. Acton, J.E. "A Computer Program for the Analysis of the Effects Of Torsion" Concrete and Constructional Engineering, Vol. 60, No. 8, August, 1965.

29. Sawko, F. "The Analysis of Grid Framework and Related Structures" M.Sc. Dissertation, Leeds University, 1960.

30. Evans, R.H. and Bennett, E.W. "Prestressed Concrete - Theory and Design". Chapman and Hall Ltd., London, 1962.

31. Morice, P.B. "Linear Structural Analysis" Thames and Hudson, London, 1959.

32. Finch, R.M. and Goldstein, A. "Clifton Bridge, Nottingham. Initial Design Studies and Model Test". The Institution of Civil Engineers, Vol. 12, March, 1959.

34. English Electric Leo Marconi Computers Ltd., "KDF9 - Algol Programming".

35. Sawko, F. "Simulation of Haunched Members and Hinged Joints in Computer Programs." Civil Engineering and Public Works Review, January, 1966.

36. Morice, P.B. "Concentrated Loads on Prestressed Concrete Bridge Decks". Journal of the Institution of Engineers (India). Vol. 35, No. 7, Part 1, August, 1955.

37. Sawko, F. "Grid Frameworks with Normal Elastic Restraints at Joints". Civil Engineering and Public Works Review, August, 1963.

38. Logcher, R.D., Sturman, G.M. and Biggs, J.M. "Structural Design Language - A Computer System for the Total Structural Design Process". International Symposium. The Use of Electronic Digital Computers in Structural Engineering, University of Newcastle, 1966.

Appendix 1.

Grid Framework Program.

```
begin comment A program for the elastic analysis of grid
        frameworks consisting of members with varying
                                     section properties;
    library A0,A6,A14;
    real E,G;
    integer m,n,pc,dv,x,y;
    real procedure dot(a,b,p,q,r);
        value p,q;
        real a,b; integer p,q,r;
        comment innerproduct with lower and upper bounds;
      begin real s; s:=0;
          for r:=p step 1 until q do s:=s+a×b;
          dot:=s;
      end dot;

    dv:=70; open(dv); open(20);
    write text(dv,[[8s]GRID*FRAMEWORK*-*VARYING*SECTION
                 *MEMBERS[c][8s]F.S.AND*B.K.W.****Leeds*
                 University****30/3/65.[c][8s]]);
    copy text(20,dv,[;]);
    m:=read(20); n:=read(20); x:=read(20); pc:=read(20);
    E:=read(20); G:=read(20);

    begin array P[1:x,1:8],F[1:x,1:13];
          integer array M[1:m,1:4],T[1:x];
          integer a,c,d,f,g,i,j,r,t,w,z;
          real L; i:=0;w:=0;

          for g:=1 step 1 until x do
          begin real L;  i:=i+1;
                   for j:=1,2 do M[i,j]:=read(20);
               M[i,3]:=g;  d:=read(20);
               for j:=1,2 do P[g,j]:=read(20);
               P[g,3]:=L:=12×sqrt(P[g,1]↑2+P[g,2]↑2);
               P[g,1]:=12×P[g,1]/L;
               P[g,2]:=12×P[g,2]/L;
               t:=read(20);
               if t>1 then
            begin array HF[1:2,1:t],H[1:2,1:t],
                        Q[1:t],J[1:t];
                real f11,f12,f22;
               T[g]:=t;
                for j:=1 step 1 until t do  HF[1,j]:=read(20);
                forj:=1 step 1 until t do J[j]:=read(20);
                for j:=1 step 1 until t do
                begin H[1,j]:=(t-j)/(t-1);
                        H[2,j]:=(j-1)/(t-1);
                end;
                Q[1]:=Q[t]:=1;
                for j:=2 step 2 until t-1 do Q[j]:=4;
                for j:=3 step 2 until t-2 do Q[j]:=2;
```

```
            for j:=1 step 1 until t do
            HF[1,j]:=HF[2,j]:=(Q[j]×L)/(E×HF[1,j]×3×(t-1));
            for j:=1 step 1 until t do F[g,j]:=HF[1,j];
            for c:=1,2 do
            for j:=1 step 1 until t do
            HF[c,j]:=H[c,j]×HF[c,j];

            f11:=dot(HF[1,j],H[1,j],1,t,j);
            f12:=dot(HF[1,j],H[2,j],1,t,j);
            f22:=dot(HF[2,j],H[2,j],1,t,j);

            P[g,5]:=f12/f22;
            P[g,6]:=f12/f11;
            P[g,7]:=1/(f11-P[g,5]×f12);
            P[g,8]:=1/(f22-P[g,6]×f12);

            for j:=1 step 1 until t do
            J[j]:=(G×J[j]×Q[j])/(3×(t-1)×L);
            P[g,4]:=dot(J[j],1,1,t,j);
        end else
        begin t:=T[g]:=13;
            P[g,5]:=P[g,6]:=0.5;
            P[g,7]:=P[g,8]:=4×E×read(20)/L;
            P[g,4]:=G×read(20)/L;
            F[g,1]:=F[g,13]:=1;
            F[g,2]:=F[g,4]:=F[g,6]:=F[g,8]:=F[g,10]:=F[g,12]:=4;
            F[g,3]:=F[g,5]:=F[g,7]:=F[g,9]:=F[g,11]:=2;
        end;
        r:=i;
        for c:=1 step 1 until d-1 do
        begin i:=r+c;
            for j:=1,2 do M[i,j]:=read(20);
            M[i,3]:=g;
        end;
    end Array P holds (1)cos,(2)sin,(3)L,(4)T,(5)C12,(6)C21,
            (7)K12,(8)K21;

    for i:=1 step 1 until m do
        if abs(M[i,1]-M[i,2])>w/3-1 then
        w:=(abs(M[i,1]-M[i,2])+1)×3;
        comment Above statement calculates the width of the
                    stiffness band;

    begin array S[1:3×n,1:w];
            for i:=1 step 1 until 3×n do
            for j:=1 step 1 until w do S[i,j]:=0;

        for i:=1 step 1 until m do
        begin integer M1J3; real P1,P2,P3,P4,P5,P6,P7,P8;
            g:=M[i,3];
```

```
        P1:=P[g,1]; P2:=P[g,2]; P3:=P[g,3]; P4:=P[g,4];
        P5:=P[g,5]; P6:=P[g,6]; P7:=P[g,7]; P8:=P[g,8];
        Mij3:=3×M[i,1];
        S[Mij3-2,1]:=S[Mij3-2,1]+P1↑2×P4+P2↑2×P7;
        S[Mij3-2,2]:=S[Mij3-2,2]+P1×P2×(P4-P7);
        S[Mij3-2,3]:=S[Mij3-2,3]-P2×(P7+P6×P8)/P3;
        S[Mij3-1,1]:=S[Mij3-1,1]+P2↑2×P4+P1↑2×P7;
        S[Mij3-1,2]:=S[Mij3-1,2]+P1×(P7+P6×P8)/P3;
        S[Mij3,1]:=S[Mij3,1]+(P7+P6×P8+P8+P5×P7)/P3↑2;
            Mij3:=3×M[i,2];
        S[Mij3-2,1]:=S[Mij3-2,1]+P1↑2×P4+P2↑2×P8;
        S[Mij3-2,2]:=S[Mij3-2,2]+P1×P2×(P4-P8);
        S[Mij3-2,3]:=S[Mij3-2,3]+P2×(P8+P5×P7)/P3;
        S[Mij3-1,1]:=S[Mij3-1,1]+P2↑2×P4+P1↑2×P8;
        S[Mij3-1,2]:=S[Mij3-1,2]-P1×(P8+P5×P7)/P3;
        S[Mij3,1]:=S[Mij3,1]+(P7+P6×P8+P8+P5×P7)/P3↑2;
        comment above statements set up diagonal elements
                in the S matrix;

        t:=3×abs(M[i,1]-M[i,2]);
        if M[i,1]<M[i,2] then
        begin r:= 3×M[i,1];
            S[r-2,t+1]:=-P1↑2×P4+P2↑2×P6×P8;
            S[r-2,t+2]:=-P1×P2×(P4+P6×P8);
            S[r-2,t+3]:=+P2×(P7+P6×P8)/P3;
            S[r-1,t]:=-P1×P2×(P4+P6×P8);
            S[r-1,t+1]:=-P2↑2×P4+P1↑2×P6×P8;
            S[r-1,t+2]:=-P1×(P7+P8×P6)/P3;
            S[r,t-1]:=-P2×P8×(1+P6)/P3;
            S[r,t]:=P1×P8×(1+P6)/P3;
            S[r,t+1]:=-(P7+P6×P8+P8+P5×P7)/P3↑2;
        end
        else
        begin r:=3×M[i,2];
            S[r-2,t+1]:=-P1↑2×P4+P2↑2×P5×P7;
            S[r-2,t+2]:=-P1×P2×(P4+P5×P7);
            S[r-2,t+3]:=-P2×(P8+P5×P7)/P3;
            S[r-1,t]:=-P1×P2×(P4+P5×P7);
            S[r-1,t+1]:=-P2↑2×P4+P1↑2×P5×P7;
            S[r-1,t+2]:=+P1×(P8+P5×P7)/P3;
            S[r,t-1]:=+P2×P7×(1+P5)/P3;
            S[r,t]:=-P1×P7×(1+P5)/P3;
            S[r,t+1]:=-(P7+P6×P8+P8+P5×P7)/P3↑2;
        end;
    end above statements set up off diagonal elements;
    y:=read(20);
    begin array B[1:3×n,1:y]; integer k,s;
        for i:=1 step 1 until 3×n do
        for j:=1 step 1 until y do B[i,j]:=0;

        for f:=1 step 1 until y do
        begin integer p,q;
            real e,h,l;
            write text(dv,[[c][8s]LOADING*CASE*No.*]);
```

```
    write(dv,format([nd]),f+pc);
    write text(dv,[[2c][8s]MEMBER*REACTIONS***TONS
      [14s].[4s]FIXED*END*MOMENTS***TONS*FT[2c][8s]
      No.[4s]MEMBER[7s]END*1[7s]END*2[10s]END*1
      [7s]END*2[c]]);
  z:=read(20);
  for c:=1 step 1 until z do
  begin p:=read(20); q:=read(20);
      for i:=1 step 1 until m do
      if p=M[i,1] and q =M[i,2] then
      begin g:=M[i,3]; t:=T[g];
            L:=P[g,3];
      end;
      begin array N[1:t], HF[1:2,1:t], H[1:2,1:t];
          real f11,f12,f22,m1,m2,M1,M2,R1,R2,S1,S2,W;
          for i:=1 step 1 until t do N[i]:=0;
          W:=read(20);
          for a:=1 step 1 until t do
          N[a]:=W×L↑2×(a-1)×(1-(a-1)/(t-1))/(24×(t-1));
          R1:=R2:=W×L/24;
          h:=read(20);
          for e:=1 step 1 until h do
          begin W:=read(20); l:=read(20); l:=12×l;
              R1:=R1+W×(L-l)/L;
              R2:=R2+W×l/L;
              for a:=1 step 1 until t do
              if (a-1)×L/(t-1)≤l then
              N[a]:=N[a]+W×(L-l)×(a-1)/(t-1)
              else N[a]:=N[a]+W×l×(t-a)/(t-1);
          end;

          for j:=1 step 1 until t do
          begin H[1,j]:=(t-j)/(t-1);
              H[2,j]:=(j-1)/(t-1);
              HF[1,j]:=H[1,j]×F[g,j];
              HF[2,j]:=H[2,j]×F[g,j];
          end;

          f11:=dot(HF[1,j],H[1,j],1,t,j);
          f12:=dot(HF[1,j],H[2,j],1,t,j);
          f22:=dot(HF[2,j],H[2,j],1,t,j);
          m1:=dot(HF[1,j],N[j],1,t,j);
          m2:=dot(HF[2,j],N[j],1,t,j);
          M1:=(m1-f12×m2/f22)/(f11-f12↑2/f22);
          M2:=(m1-f11×M1)/f12;
          S1:=+R1+(M1-M2)/L;
          S2:=+R2-(M1-M2)/L;

          write(dv,format([8sndd]),c);
          write text(dv,[)]);
          write(dv,format([sndd]),p);
          write text(dv,[*-]);
          write(dv,format([ndd]),q);
```

```
                write(dv,format([4s+nddd.dd]),S1);
                write(dv,format([4s+nddd.dd]),S2);
                write(dv,format([6s+nddd.ddd]),-M1/12);
                write(dv,format([3s+nddd.dddc]),M2/12);

                B[3×p-2,f]:=B[3×p-2,f]-P[g,2]×M1;
                B[3×p-1,f]:=B[3×p-1,f]+P[g,1]×M1;
                B[3×p,f]:=B[3×p,f]+S1;
                B[3×q-2,f]:=B[3×q-2,f]+P[g,2]×M2;
                B[3×q-1,f]:=B[3×q-1,f]-P[g,1]×M2;
                B[3×q,f]:=B[3×q,f]+S2;
            end;
        end;
    end calculation of fixed end moments and shears;

    x:=read(20);
    for i:=1 step 1 until x do
    begin t:=3×read(20); r:=read(20);
        for j:=1 step 1 until y do B[t+r-3,j]:=0;
        for s:=1 step 1 until t+r-4 do if t+r-s-2 ≤w
            then S[s,t+r-s-2]:=0;
        for j:=2 step 1 until w do S[t+r-3,j]:=0;
    end above statements impose zero deflexions
            where x=no. of imposed zeros;

    solution of equations:
    for i:=1 step 1 until 3×n do
    begin S[i,1]:=sqrt(S[i,1]-dot(S[r,i-r+1]↑2,1,
                if i>w then i-w+1 else 1,i-1,r));
        k:=if 3×n-i>w-1 then w else 3×n-i+1;
        for j:=2 step 1 until k do
        S[i,j]:=(S[i,j]-dot(S[r,i-r+1],S[r,j+i-r],if
                j+i>w+1 then j+i-w else 1,i-1,r))/S[i,1];
        end formation of triangular matrix;

    back substitution:
    for j:=1 step 1 until y do
    begin for i:=1 step 1 until 3×n do
        B[i,j]:=(B[i,j]-dot(S[r,i-r+1],B[r,j],
            if i>w then i-w+1 else 1,i-1,r))/S[i,1];
        B[3×n,j]:=B[3×n,j]/S[3×n,1];
        for i:=3×n-1 step -1 until 1 do
        B[i,j]:=(B[i,j]-dot(S[i,r-i+1],B[r,j],i+1,
            if 3×n-i>w-1 then i+w-1 else 3×n,r))/S[i,1];
    end back substitution;

    for j:=1 step 1 until y do
    begin integer p,q; real Mx1,My1,Mx2,My2,Fz,Mq1,Mq2,Mp;
        write text(dv,[[3c][8s]LOADING*CASE*No.*]);
        write(dv,format([nd]),pc+j);
        write text(dv,[[2c][8s]MEMBER[3s]FORCES**TONS[15s]
            MOMENTS*TON*FT[2c][8s]No.[4s]MEMBER[7s]SHEAR
            [8s]TORSION[6s]END*1[8s]END*2[c]]);
```

```
for i:=1 step 1 until  m do
begin array d[1:14];  g:=M[i,3];
    d[1]:=P[g,1]↑2×P[g,4]+P[g,2]↑2×P[g,7];
    d[2]:=P[g,1]×P[g,2]×(P[g,4]-P[g,7]);
    d[3]:=-P[g,2]×(P[g,7]+P[g,6]×P[g,8])/P[g,3];
    d[4]:=P[g,2]↑2×P[g,4]+P[g,1]↑2×P[g,7];
    d[5]:=P[g,1]×(P[g,7]+P[g,6]×P[g,8])/P[g,3];
    d[6]:=(P[g,7]+P[g,6]×P[g,8]+P[g,8]+P[g,5]
            ×P[g,7])/P[g,3]↑2;
    d[7]:=-P[g,1]↑2×P[g,4]+P[g,2]↑2×P[g,6]×P[g,8];
    d[8]:=-P[g,1]×P[g,2]×(P[g,4]+P[g,6]×P[g,8]);
    d[9]:=-P[g,2]↑2×P[g,4]+P[g,1]↑2×P[g,6]×P[g,8];
    d[10]:=-P[g,2]×P[g,8]×(1+P[g,6])/P[g,3];
    d[11]:=P[g,1]×(P[g,8]+P[g,7]×P[g,5])/P[g,3];
    d[12]:=P[g,2]↑2×P[g,4]+P[g,1]↑2×P[g,8];
    d[13]:=P[g,1]×P[g,2]×(P[g,4]-P[g,8]);
    d[14]:=P[g,1]↑2×P[g,4]+P[g,2]↑2×P[g,8];

    p:=3×M[i,1]; q:=3×M[i,2];
    Mx1:=d[1]×B[p-2,j]+d[2]×B[p-1,j]+d[3]×
    (B[p,j]-B[q,j])+d[7]×B[q-2,j]+d[8]×B[q-1,j];
    My1:=d[2]×B[p-2,j]+d[4]×B[p-1,j]+d[5]×
    (B[p,j]-B[q,j])+d[8]×B[q-2,j]+d[9]×B[q-1,j];
    Fz:=d[3]×B[p-2,j]+d[5]×B[p-1,j]+d[6]×(B[p,j]-
    B[q,j])+d[10]×B[q-2,j]+d[11]×B[q-1,j];
    Mx2:=d[7]×B[p-2,j]+d[8]×B[p-1,j]+d[10]×
    (B[p,j]-B[q,j])+d[14]×B[q-2,j]+d[13]×B[q-1,j];
    My2:=d[8]×B[p-2,j]+d[9]×B[p-1,j]+d[11]×
    (B[p,j]-B[q,j])+d[13]×B[q-2,j]+d[12]×B[q-1,j];

    Mq1:=(-Mx1×P[g,2]+My1×P[g,1])/12;
    Mq2:=(-Mx2×P[g,2]+My2×P[g,1])/12;
    Mp:=(Mx1×P[g,1]+My1×P[g,2])/12;
    write(dv,format([8s ndd]),i);
    write text(dv,[)]);
    write(dv,format([sndd]),M[i,1]);
    write text(dv,[*-]);
    write(dv,format([ndd]),M[i,2]);
    write(dv,format([5s+ndd.ddd]),Fz);
    write(dv,format([4s+nddd.ddd]),Mp);
    write(dv,format([4s+nddd.ddd]),Mq1);
    write(dv,format([4s+nddd.dddc]),Mq2);
end;
write text(dv,[[3c][9s]JOINT[4s]X*ROTATION
    [8s]Y*ROTATION[8s]VERTICAL[c]]);
for i:=1 step 1 until n do
begin write(dv,format([9sndd]),i);
    write text(dv,[)]);
    write(dv,format([3s+nd.dddddddd]),B[3×i-2,j]);
    write(dv,format([6s+nd.dddddddd]),B[3×i-1,j]);
    write(dv,format([6s+nd.ddddddddc]),B[3×i,j]);
end;
    end; end;
  end;   end;  close(dv);  close(20);
end
→
```

Appendix 2.

Member Properties of Perspex Model.

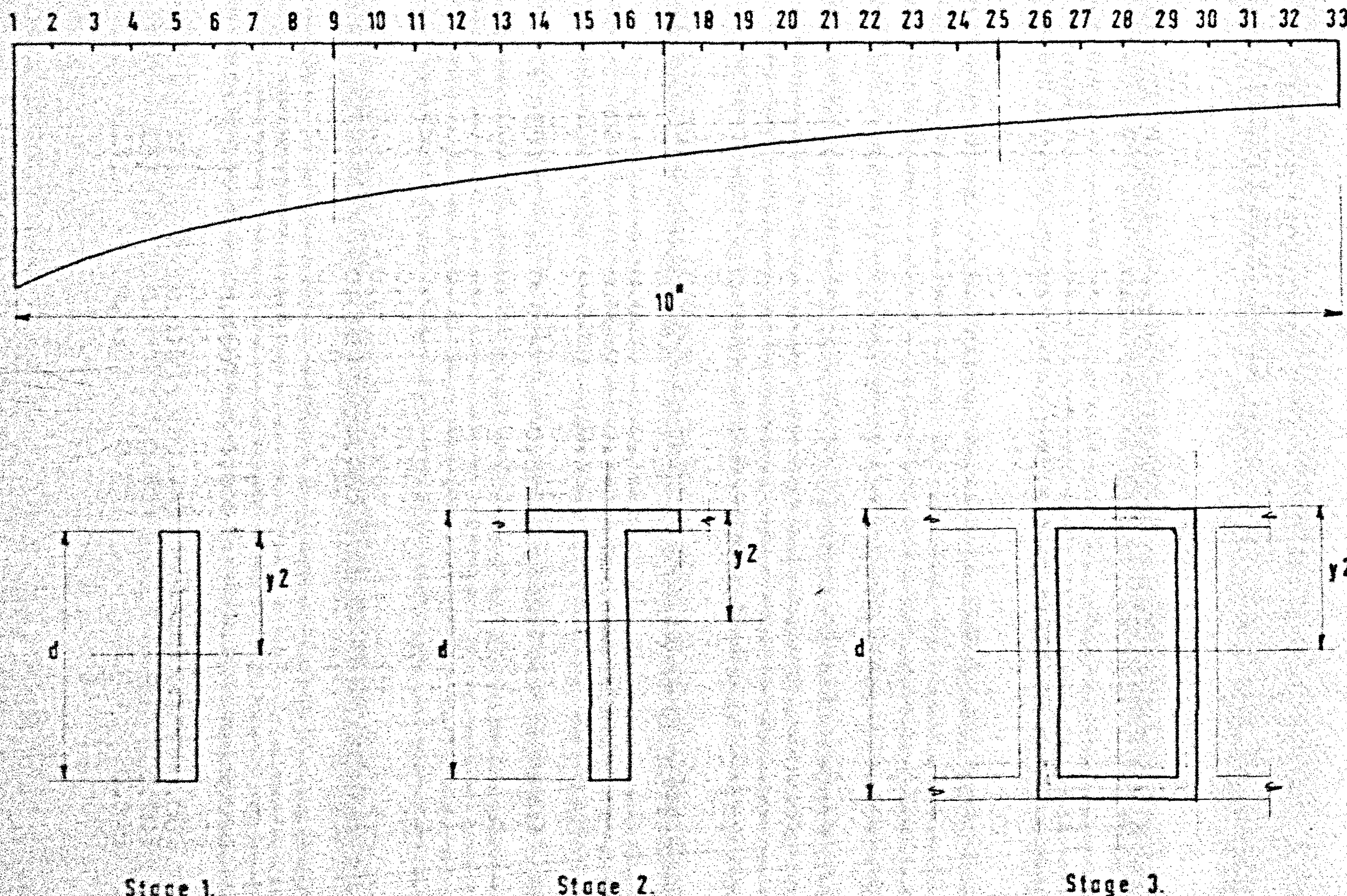


Fig. X1. Model Section Properties.

MEMBER PROPERTIES.
PERSPEX MODEL STAGE 1.

LONGITUDINAL GIRDERS.

STATION	I	y2	J	d
1)	0.1667	1.000	0.0096	2.000
2)	0.1449	0.954	0.0091	1.909
3)	0.1256	0.910	0.0087	1.820
4)	0.1087	0.867	0.0082	1.735
5)	0.0939	0.826	0.0078	1.652
6)	0.0810	0.786	0.0074	1.572
7)	0.0697	0.748	0.0070	1.495
8)	0.0598	0.711	0.0066	1.421
9)	0.0513	0.675	0.0062	1.350
10)	0.0439	0.641	0.0059	1.282
11)	0.0375	0.608	0.0055	1.217
12)	0.0321	0.577	0.0052	1.154
13)	0.0274	0.548	0.0049	1.095
14)	0.0233	0.519	0.0046	1.038
15)	0.0199	0.492	0.0043	0.985
16)	0.0170	0.467	0.0040	0.934
17)	0.0145	0.443	0.0038	0.886
18)	0.0124	0.421	0.0036	0.841
19)	0.0106	0.400	0.0033	0.799
20)	0.0091	0.380	0.0031	0.760
21)	0.0079	0.362	0.0029	0.724
22)	0.0069	0.345	0.0028	0.691
23)	0.0060	0.330	0.0026	0.660
24)	0.0053	0.316	0.0025	0.632
25)	0.0047	0.304	0.0023	0.608
26)	0.0042	0.293	0.0022	0.586
27)	0.0038	0.284	0.0021	0.567
28)	0.0035	0.276	0.0021	0.551
29)	0.0032	0.269	0.0020	0.538
30)	0.0031	0.264	0.0019	0.528
31)	0.0029	0.260	0.0019	0.521
32)	0.0029	0.258	0.0019	0.516
33)	0.0028	0.257	0.0019	0.515

TRANSVERSE GIRDERS.

STATION	I	y2	J	d
1)	0.1667	1.000	0.0096	2.000
9)	0.0513	0.675	0.0062	1.350
17)	0.0145	0.443	0.0038	0.886
25)	0.0047	0.304	0.0023	0.608
33)	0.0028	0.257	0.0019	0.515

MEMBER PROPERTIES.
PERSPEX MODEL STAGE 2

LONGITUDINAL GIRDERS.

STATION	I	y2	J	d
1)	0.2797	0.912	0.0105	2.125
2)	0.2474	0.868	0.0100	2.034
3)	0.2186	0.825	0.0095	1.945
4)	0.1928	0.784	0.0091	1.860
5)	0.1698	0.745	0.0087	1.777
6)	0.1494	0.706	0.0082	1.697
7)	0.1313	0.670	0.0078	1.620
8)	0.1153	0.634	0.0075	1.546
9)	0.1011	0.601	0.0071	1.475
10)	0.0886	0.569	0.0067	1.407
11)	0.0776	0.538	0.0064	1.342
12)	0.0679	0.509	0.0061	1.279
13)	0.0595	0.481	0.0058	1.220
14)	0.0521	0.455	0.0055	1.163
15)	0.0456	0.431	0.0052	1.110
16)	0.0400	0.407	0.0049	1.059
17)	0.0351	0.386	0.0047	1.011
18)	0.0309	0.366	0.0044	0.966
19)	0.0272	0.347	0.0042	0.924
20)	0.0241	0.329	0.0040	0.885
21)	0.0214	0.314	0.0038	0.849
22)	0.0191	0.299	0.0036	0.815
23)	0.0171	0.286	0.0035	0.785
24)	0.0154	0.274	0.0033	0.757
25)	0.0140	0.264	0.0032	0.733
26)	0.0129	0.254	0.0031	0.711
27)	0.0119	0.246	0.0030	0.692
28)	0.0111	0.240	0.0029	0.676
29)	0.0105	0.234	0.0029	0.663
30)	0.0101	0.230	0.0028	0.653
31)	0.0098	0.227	0.0028	0.646
32)	0.0096	0.225	0.0027	0.641
33)	0.0095	0.225	0.0027	0.640

TRANSVERSE GIRDERS.

STATION	I	y2	J	d
1)	0.3842	0.716	0.0110	2.125
9)	0.1400	0.446	0.0076	1.475
17)	0.0481	0.272	0.0052	1.011
25)	0.0188	0.182	0.0037	0.733
33)	0.0126	0.156	0.0032	0.640

MEMBER PROPERTIES.
PERSPEX MODEL STAGE 3.

INTERNAL LONGITUDINAL GIRDERS.

STATION	I	y2	J	d
1)	0.4492	1.125	0.4105	2.250
2)	0.4037	1.079	0.3889	2.159
3)	0.3624	1.035	0.3680	2.070
4)	0.3252	0.992	0.3480	1.985
5)	0.2916	0.951	0.3286	1.902
6)	0.2613	0.911	0.3101	1.822
7)	0.2341	0.873	0.2923	1.745
8)	0.2096	0.836	0.2753	1.671
9)	0.1877	0.800	0.2591	1.600
10)	0.1680	0.766	0.2436	1.532
11)	0.1504	0.733	0.2289	1.467
12)	0.1347	0.702	0.2149	1.405
13)	0.1207	0.673	0.2017	1.345
14)	0.1083	0.644	0.1892	1.289
15)	0.0972	0.617	0.1775	1.235
16)	0.0874	0.592	0.1665	1.184
17)	0.0787	0.568	0.1562	1.136
18)	0.0711	0.546	0.1467	1.091
19)	0.0644	0.525	0.1378	1.049
20)	0.0584	0.505	0.1297	1.010
21)	0.0533	0.487	0.1222	0.974
22)	0.0487	0.470	0.1154	0.940
23)	0.0448	0.455	0.1093	0.910
24)	0.0415	0.441	0.1038	0.882
25)	0.0386	0.429	0.0989	0.858
26)	0.0361	0.418	0.0946	0.836
27)	0.0341	0.409	0.0909	0.817
28)	0.0324	0.401	0.0879	0.801
29)	0.0311	0.394	0.0854	0.788
30)	0.0300	0.389	0.0834	0.778
31)	0.0293	0.385	0.0820	0.771
32)	0.0289	0.383	0.0812	0.766
33)	0.0288	0.382	0.0809	0.765

TRANSVERSE GIRDERS.

STATION	I	y2	J	d
1)	0.8730	1.125	0.6641	2.250
9)	0.3922	0.800	0.3778	1.600
17)	0.1751	0.568	0.2038	1.136
25)	0.0894	0.429	0.1174	0.858
33)	0.0677	0.382	0.0926	0.765

Appendix 3

Automatic Design Program

```
begin comment Automatic design program for 3 span prestressed
                                        concrete bridges;
    library A0,A6,A14;
    real R,W,la,fct,fcw,fmint,fminw,tw,cs,tbfs,tbfc,y1,x1,h,h1,
            ttf,Z1c,Z2c,Z1s,Z2s,y1c,y2c,y1s,y2s,Ic,Is,y,ds,dc,
            tf,Weight1,Weight2,ft,fb,Ac,e,fx,fpt,Ro,E,G,I,we,wc,
            nl,Pl,Mpos,Mneg,Ps;
    integer d,dv,f,i,j,k,r,s,st,w,x,nw,nu,g,m,t;
    real procedure dot(a,b,p,q,r);
        value p,q;
        real a,b;  integer p,q,r;
        comment innerproduct with lower and upper bounds;
        begin real s; s:=0;
            for r:=p step 1 until q do s:=s+a×b;
            dot:=s;
        end dot;
    procedure solve(s,b,n,w,lds);
        value lds,n,w; array s,b;
        integer n,w,lds;
        begin integer i,j,k;
            for i:=1 step 1 until n do
            begin s[i,1]:=sqrt(s[i,1]-dot(s[r,i-r+1]↑2,1,
                                if i>w then i-w+1 else 1,i-1,r));
                    k:=if n-i>w-1  then w else n-i+1;
                    for j:=2 step 1 until k do
                    s[i,j]:=(s[i,j]-dot(s[r,i-r+1],s[r,j+i-r],
                    if j+i>w+1 then j+i-w else 1,i-1,r))/s[i,1];
            end formation of triangular matrix;
            back substitution:
            for j:=1 step 1 until lds do
            begin for i:=1 step 1 until n do
                    b[i,j]:=(b[i,j]-dot(s[r,i-r+1],b[r,j],
                    if i>w then i-w+1 else 1,i-1,r))/s[i,1];
                    b[n,j]:=b[n,j]/s[n,1];
                    for i:=n-1 step -1 until 1 do
                    b[i,j]:=(b[i,j]-dot(s[i,r-i+1],b[r,j],i+1,
                    if n-i>w-1 then i+w-1 else n,r))/s[i,1];
            end solution of equations;
        end;
    procedure find I(dp,inertia,tfb,width,centroid);
            value tfb,dp,width;
            real dp,tfb,width,inertia,centroid;
        comment calculates moment of inertia from section sizes;
        begin real At,Ayt,Adt,It;
            procedure sum area(breadth,depth,ctr);
                value breadth,depth,ctr;
                real breadth,depth,ctr;
                begin real A,I,Ay,Ad;
                    A:=breadth×depth; At:=At+A;
                    I:=A×depth↑2/12; It:=It+I;
                    Ay:=A×ctr; Ayt:=Ayt+Ay;
                    Ad:=Ay×ctr; Adt:=Adt+Ad;
                end sum area;
```

```
        At:=Ayt:=Adt:=It:=0;
        sum area(width,tfb,tfb/2);
        sum area(tw,dp,dp/2+tfb);
        sum area(width,ttf,dp+tfb+ttf/2);
        y:=Ayt/At; It:=It+Adt-At×y↑2;
        inertia:=It;  centroid:=y;
    end find I;
procedure prelim sizes(m1,l1,qs,b);
    array m1,l1,b; integer array qs;
    comment designs size of section from range of bending moment;
    begin for j:=11,21 do
        begin Weight1:=0; m1[3,j]:=0;
            for f:=1,2,3 do
            begin h:=4×(tbfs-tbfc)/l1[f,2];
                h1:=4×(ds-tbfs-dc+tbfc)/l1[f,2];
                for i:=1 step 1 until d+1 do
                begin if f=1 and l1[f,1]<l1[f,2] then
                    x1:=l1[f,2]-l1[f,1]+(i-1)/d×l1[f,1]
                    else x1:=(i-1)/d×l1[f,1];
                    tf:=h×(x1-x1↑2/l1[f,2]); tf:=tbfs-tf;
                    y1:=h1×(x1-x1↑2/l1[f,2]); y1:=ds-ttf-tbfs-y1;
                    x1:=(12×cs×(ttf+tf)+tw×y1)×0.15×l1[f,1]/(432×d);
                    m1[3,j]:=m1[3,j]+x1×qs[i]×b[2×(i+(f-1)×d),j];
                    if j=21 then Weight1:=Weight1+x1×qs[i];
                end;
            end;
        end;
        Z1c:=12×(m1[1,21]+m1[3,21]-Ro×(m1[3,21]+m1[2,21]))/
                    (Ro×fct-fminw);
        Z2c:=12×(m1[1,21]+m1[3,21]-Ro×(m1[3,21]+m1[2,21]))/
                    (fcw-Ro×fmint);
        Z1s:=-12×(m1[3,11]+m1[2,11]-Ro×(m1[3,11]+m1[1,11]))/
                    (fcw-Ro×fmint);
        Z2s:=-12×(m1[3,11]+m1[2,11]-Ro×(m1[3,11]+m1[1,11]))/
                    (Ro×fct-fminw);
        ft:=fmint-12×(m1[2,21]+m1[3,21])/Z2c;
        fb:=fminw/Ro+12×(m1[1,21]+m1[3,21])/(Ro×Z1c);
        Ac:=tw×(dc-tbfc-ttf)+12×cs×(ttf+tbfc);
        e:=Z1c×Z2c×(ft-fb)/(Ac×(fb×Z1c+ft×Z2c));
        if e<-y1c+3 then
        begin e:=-y1c+3;
            fx:=12×(m1[3,21]+m1[2,21])/Z2c+(Z1c×(Z2c+Ac×e))/
            (Ro×Z2c×(Z1c-Ac×e))×(12×(m1[1,21]+m1[3,21])/Z1c+fminw);
            Z2c:=12×(m1[1,21]+m1[3,21]-Ro×(m1[3,21]+m1[2,21]))/
                                (fcw-Ro×fx);
            tbfc:=Z1c/Z2c×ttf; if tbfc<4 then tbfc:=4;
            tbfc:=entier(10×tbfc+0.5)/10;
        end;
```

```
y2c:=(Z2c+sqrt(Z2c↑2+48×cs×((ttf+ttf↑2/tbfc)×Z2c×ttf
       /2)))/(24×cs×(ttf+ttf↑2/tbfc));
dc:=y2c+ttf/2+y2c×ttf/tbfc+tbfc/2; dc:=entier(dc+1);
recal:find I(dc-ttf-tbfc,Ic,tbfc,12×cs,y);
if Ic/y>Z1c or Ic/(dc-y)>Z2c then
begin x1:=Ic; y1c:=y; y2c:=dc-y; dc:=dc-0.5;
    goto recal;
end;
Ic:=x1;
ft:=fminw/Ro-12×(m1[3,11]+m1[2,11])/(Ro×Z2s);
fb:=fmint+12×(m1[3,11]+m1[1,11])/Z1s;
Ac:=(ds-ttf-tbfs)×tw+12×cs×(ttf+tbfs);
e:=Z1s×Z2s×(ft-fb)/(Ac×(fb×Z1s+ft×Z2s));
if e>y2s-3 then
begin e:=y2s-3;
    fx:=-12×(m1[3,11]+m1[1,11])/Z1s+(Z2s×(Z1s-Ac×e))/(Ro×Z1s
          ×(Z2s+Ac×e))×(-12×(m1[3,11]+m1[2,11])/ Z2s+fminw);
    Z1s:=-12×(m1[3,11]+m1[2,11]-Ro×(m1[3,11]+m1[1,11]))/
                    (Ro×fct-fx);
    if Z1s>Z2s then tbfs:=ttf×Z1s/Z2s;
    tbfs:=entier(10×tbfs+0.5)/10;
end;
y1s:=(Z1s+sqrt(Z1s↑2+48×cs×((tbfs+tbfs↑2/ttf)×Z1s×tbfs/
      2)))/(24×cs×(tbfs+tbfs↑2/ttf));
ds:=y1s+tbfs/2+y1s×tbfs/ttf+ttf/2;
ds:=entier(ds+1);
recal1:find I(ds-ttf-tbfs,Is,tbfs,12×cs,y);
if Is/y>Z1s or Is/(ds-y)>Z2s then
begin x1:=Is;y1s:=y;  y2s:=ds-y; ds:=ds-0.5;
      goto recal1;
end;
Is:=x1;
if Is<Ic×(1+R)↑2 then
begin Is:=Ic×(1+R)↑2;
    y2s:=sqrt(Is/(12×cs×ttf+12×cs×ttf↑2/tbfs));
    ds:=y2s+ttf/2+y2s×ttf/tbfs+tbfc/2;
    ds:=entier(ds+1);
    recal4: findI(ds-ttf-tbfs,x1,tbfs,12×cs,y);
    if x1>Is then
    begin y1s:=y; y2s:=ds-y;
        ds:=ds-0.5; Is:=x1; goto recal4;
    end;
    ds:=y2s+y1s;
end else
begin Ic:=Is/(1+R)↑2;
    y2c:=sqrt(Ic/(12×cs×ttf+12×cs×ttf↑2/tbfc));
    dc:=y2c+ttf/2+y2c×ttf/tbfc+tbfc/2; dc:=entier(dc+1);
    recal5: findI(dc-ttf-tbfc,x1,tbfc,12×cs,y);
```

```
            if x1>Ic then
            begin y1c:=y; y2c:=dc-y; dc:=dc-0.5;
                Ic:=x1; goto recal5;
            end;
            dc:=y2c+y1c;
        end;
        Weight2:=0;
        for f:=1,2,3 do
        begin h:=4×(tbfs-tbfc)/l1[f,2];
            h1:=4×(ds-tbfs-dc+tbfc)/l1[f,2];
            for i:=1 step 1 until d+1 do
            begin if f=1 and l1[f,1]<l1[f,2] then x1:=l1[f,2]-
                    l1[f,1]+(i-1)/d×l1[f,1]
                    else x1:=(i-1)/d×l1[f,1];
                tf:=h×(x1-x1↑2/l1[f,2]);  tf:=tbfs-tf;
                y1:=h1×(x1-x1↑2/l1[f,2]); y1:=ds-ttf-tbfs-y1;
                Weight2:=Weight2+(12×cs×(ttf+tf)+tw×y1)×0.15×
                    l1[f,1]×qs[i]/(432×d);
            end;
        end;
    end;

    dv:=70; open(20);  open(dv);
    write text(dv,[[8s]AUTOMATIC*DESIGN*OF*3*SPAN*BRIDGE.[c][8s]
            F.S.AND*B.K.W.****LEEDS*UNIVERSITY****11/5/66.[c][8s]]);
    copy text(20,dv,[;]);
    s:=3;  d:=20; w:=4; st:=5;
    begin array L[1:s,1:2],UDL[1:s],M1[1:11,1:21];
        integer array Qs[1:d+1];
        Qs[1]:=Qs[d+1]:=1;
        for j:=2 step 2 until d do Qs[j]:=4;
        for j:=3 step 2 until d-1 do Qs[j]:=2;
        begin array HF[1:2,1:st],H[1:2,1:st],Q[1:st],P[1:d×3,1:5];
            real l,I,f11,f12,f22;
            for j:=1 step 1 until st do
            begin H[1,j]:=(st-j)/(st-1);
                  H[2,j]:=(j-1)/(st-1);
            end;
            for i:= 1,2,3 do
            for j:= 1,2 do L[i,j]:=read(20);
            R:=read(20);  W:=read(20);  la:=read(20); nl:=read(20);
            we:=read(20); wc:=read(20); nu:=read(20); E:=read(20);
            G:=read(20); tw:=read(20); fct:=read(20); fmint:=read(20);
            fcw:=read(20); fminw:=read(20); fpt:=read(20); Ro:=read(20);
            Ro:=(100-Ro)/100; Mpos:=Mneg:=0; Ps:=read(20);

            Q[1]:=Q[st]:=1;
            for j:=2 step 2 until st-1 do Q[j]:=4;
            for j:=3 step 2 until st-2 do Q[j]:=2;
```

```
for f:=1,2,3 do
begin l:=L[f,T]/d;
    for i:= 1 step 1 until d do
    begin for j:=1 step 1 until st do
        begin if f=1 and L[f,1]<L[f,2] then
            x1:=2×(L[f,2]-L[f,1]+(j-1)×l/(st-1)+(i-1)×l)
            else x1:=2×((j-1)×l/(st-1)+(i-1)×l);
            I:=(1+R×(1-x1/L[f,2])↑2)↑2;
            HF[1,j]:=HF[2,j]:=(Q[j]×l)/(I×12);
        end;
        for k:=1,2 do
        for j:=1 step 1 until st do HF[k,j]:=HF[k,j]×H[k,j];

        f11:=dot(HF[1,j],H[1,j],1,st,j);
        f12:=dot(HF[1,j],H[2,j],1,st,j);
        f22:=dot(HF[2,j],H[2,j],1,st,j);
        k:=(f-1)×d+i;

        P[k,1]:=f12/f22;
        P[k,2]:=f12/f11;
        P[k,3]:=1/(f11-P[k,1]×f12);
        P[k,4]:=1/(f22-P[k,2]×f12);
        P[k,5]:=l;
    end;
end;

begin array S[1:122,1:w],B[1:122,1:21];
    for i:=1 step 1 until 122 do
    begin for j:=1,2,3,4 do S[i,j]:=0;
          for j:=1 step 1 until 21 do B[i,j]:=0;
    end;

    for f:=1,2,3 do
    for i:=1 step 1 until d do
    begin k:=(f-1)×d+i;
            S[2×k-1,1]:=S[2×k-1,1]+P[k,3];
            S[2×k-1,2]:=S[2×k-1,2]+P[k,3]×(1+P[k,1])/P[k,5];
            S[2×k,1]:=S[2×k,1]+(P[k,3]+P[k,2]×P[k,4]+P[k,4]
                             +P[k,1]×P[k,3])/P[k,5]↑2;
            S[2×k+1,1]:=S[2×k+1,1]+P[k,4];
            S[2×k+1,2]:=S[2×k+1,2]-(P[k,4]+P[k,1]×P[k,3])/
                                                   P[k,5];
            S[2×k+2,1]:=S[2×k+2,1]+(P[k,3]+P[k,2]×P[k,4]+
                             P[k,4]+P[k,1]×P[k,3])/P[k,5]↑2;
            S[2×k-1,3]:=P[k,2]×P[k,4];
            S[2×k-1,4]:=-(P[k,3]+P[k,2]×P[k,4])/P[k,5];
            S[2×k,2]:=P[k,4]×(1+P[k,2])/P[k,5];
            S[2×k,3]:=-(P[k,3]+P[k,2]×P[k,4]+P[k,4]
                             +P[k,1]×P[k,3])/P[k,5]↑2;
    end formation of stiffness matrix;
    i:=0;
```

```
for k:=1 step 2 until 21,22 step 1 until 31 do
begin i:=i+1;
        B[2×k-1,i]:=P[k,3];
        B[2×k,i]:=(P[k,3]+P[k,2]×P[k,4])/P[k,5];
        B[2×k+1,i]:=P[k,2]×P[k,4];
        B[2×k+2,i]:=-B[2×k,i];
end formation of influence line loading vectors;

for k:=1,21,41,61 do
begin for j:=1 step 1 until 21 do B[2×k,j]:=0;
        for j:=2 step 1 until w do S[2×k,j]:=0;
        for x:=1 step 1 until 2×k-1 do if 2×k-x+1≤w
                then S[x,2×k-x+1]:=0;
end imposition of zero displacements;

solve(S,B,122,w,21);

begin real span; integer k1;
    if L[1,1]<L[2,2] then span:=L[2,2] else span:=L[1,1];
    if span>200 then cs:=span/30 else cs:=6.5;
    nw:=entier(W/cs+0.5)+1; cs:=W/(nw-1);
    if la≤10 then P1:=2.7×cs else P1:=27×cs/la;

    for f:=1,2,3 do
    begin if L[f,1]<75 then UDL[f]:=2.2 else
        begin if L[f,1]≥75 and L[f,1]≤400 then k1:=20;
              if L[f,1]>400 and L[f,1]≤500 then k1:=21;
              if L[f,1]>500 then k1:=23;
              UDL[f]:=k1/sqrt(L[f,1]);
        end;
        if la≤10 then UDL[f]:=UDL[f]×0.1
        else UDL[f]:=UDL[f]/la;
    end;

    for j:= 1 step 1 until 21 do
    begin real A,max,min;
        for f:=1,2,3 do
        begin A:=max:=min:=0;
            for i:=1 step 1 until d+1 do
            begin k:=((f-1)×d+i)×2;
                  if B[k,j]>max then max:=B[k,j];
                  if B[k,j]<min then min:=B[k,j];
            end;
            A:=dot(B[2×((f-1)×d+r),j],Qs[r],1,d,r)×L[f,1]/(3×d
               if A>0 then M1[1,j]:=M1[1,j]+(UDL[f]+0.0375)×
                                     cs×A+P1×max
               else M1[2,j]:=M1[2,j]+(UDL[f]+0.0375)×cs×
                                     A+P1×min;
        end;
    end;
    begin if cs≤6 then ttf:=6;
        if cs>6 and cs≤8 then ttf:=7;
        if cs>8 and cs≤10 then ttf:=8;
```

```
            if cs>10 and cs≤12 then ttf:=9;
            if cs>12 then ttf:=12;
            Z1c:=12×(M1[1,21]-M1[2,21])/(Ro×fct-fminw);
            Z2c:=12×(M1[1,21]-M1[2,21])/(fcw-Ro×fmint);
            Z1s:=12×(M1[1,11]-M1[2,11])/(fcw-Ro×fmint);
            Z2s:=12×(M1[1,11]-M1[2,11])/(Ro×fct-fminw);
            if Z2c>Z1c then x1:=Z2c else x1:=Z1c;
            y:=(x1+sqrt(x1↑2+48×cs×ttf↑2×x1))/(48×ttf×cs);
            dc:=2×y+ttf;
            if Z2s>Z1s then x1:=Z2s else x1:=Z1s;
            y:=(x1+sqrt(x1↑2+48×cs×ttf↑2×x1))/(48×ttf×cs);
            ds:=2×y+ttf;
            if ds<dc×(1+R) then ds:=dc×(1+R) else dc:=ds/(1+R);
            dc:=entier(dc+1); ds:=entier(ds+1);
            x1:=1.7×L[2,2]×(0.28×UDL[2]×cs+0.006×ttf×cs+(dc+dc×R/8
                  )×tw/4000)/((dc+dc×R/4)×fpt);
            if x1>tw then tw:=x1; tw:=entier(tw+1);
            tbfc:=tbfs:=ttf;
            y1c:=y2c:=dc/2;  y1s:=y2s:=ds/2;
            again: prelim sizes(M1,L,Qs,B);
            if abs(Weight2-Weight1)/Weight1> 0.05 then goto again;
      end;
        for j:=1 step 1 until 21 do
        for f:=1,2,3 do
        begin h:=4×(tbfs-tbfc)/L[f,2];
            h1:=(ds-tbfs-dc+tbfc)/L[f,2];
            for i:=1 step 1 until d+1 do
            begin if f=1 and L[f,1]<L[f,2] then x1:=L[f,2]-
                     L[f,1]+(i-1)/d×L[f,1]
                     else x1:=(i-1)/d×L[f,1];
                tf:=h×(x1-x1↑2/L[f,2]);  tf:=tbfs-tf;
                y1:=h1×(x1-x1↑2/L[f,2]); y1:=ds-ttf-tbfs-y1;
                M1[3,j]:=M1[3,j]+(12×cs×(ttf+tf)+tw×y1×0.15×L[f,1]
                         ×Qs[i]×B[2×(i+(f-1)×d),j])/(432×d);
            end;
        end;
     end;
   end;
end;

x:=(nw-1)×6+(nw-2)×7; w:=nw×3;
begin array M[1:x,1:3],P[1:13,1:8],S[1:21×(nw-1),1:w],B[1:21×(nw-1)
       ,1:2],HF[1:2,1:13],H[1:2,1:13],J[1:13];
  integer array Q[1:13];
  Q[1]:=Q[13]:=1;
  for i:=2 step 2 until 12 do Q[i]:=4;
  for i:=3 step 2 until 11 do Q[i]:=2;
  for i:=1 step 1 until 21×(nw-1) do
  begin for j:=1 step 1 until w do S[i,j]:=0;
        for j:=1,2 do B[i,j]:=0;
  end;
```

```
for j:=1 step 1 until 13 do
begin H[1,j]:=(13-j)/12;
      H[2,j]:=(j-1)/12;
end;
for j:=1,2,3,4,5,6,7 do
begin real I; if j<3 then
   begin h:=4×(tbfs-tbfc)/L[1,2];
         h1:=4×(ds-tbfs-dc+tbfc)/L[1,2];
         x1:=(j-1)×L[1,1]/3+L[1,2]-L[1,1];
         i:=1; x:=3;
   end;
   if j>3 then
   begin h:=4×(tbfs-tbfc)/L[2,2];
         h1:=4×(ds-tbfs-dc+tbfc)/L[2,2];
         x1:=(j-4)×L[2,2]/6; i:=2; x:=6;
   end;
   tf:=h×(x1-x1↑2/L[i,2]); tf:=tbfs-tf;
   y1:=h1×(x1-x1↑2/L[i,2]); y1:=ds-ttf-tbfs-y1;
   find I(y1,I,tf,12×L[i,1]/x,y);

   P[j,1]:=0; P[j,2]:=1;
   P[j,3]:=12×cs;
   P[j,4]:=(4×G×(12×L[i,1]/×(y1+ttf/2+tf/2))↑2)/(12×L[i,1]
          /x×(12×L[i,1]/(x×ttf)+L[i,1]/(x×tf)
          +(4×(y1+ttf/2+tf/2)/tw))×12×cs);
   P[j,5]:=P[j,6]:=0.5;
   P[j,7]:=P[j,8]:=4×E×I/P[j,3];
   if j=1 or j=7 then
   begin P[j,4]:=P[j,4]/2;
         P[j,7]:=P[j,8]:=P[j,7]/2;
   end;

   for i:=1 step 1 until nw-2 do
   begin k:=(j-1)×(nw-2)+i;
         M[k,1]:=(j-1)×(nw-1)+i;
         M[k,2]:=(j-1)×(nw-1)+i+1;
         M[k,3]:=j;
   end;
end;

begin array HA[12,1:nw-1],AV[1:2,1:nw-1];
   integer g,sum;
   real dt,dt1,db,f11,f12,f22;
   for i:=1,2 do
   for j:=1 step 1 until nw-1 do
   begin HA[i,j]:=0;
         AV[i,j]:=0;
   end;
   for i:=1,2 do
   begin if i=1 then dt:=we-2-cs/2 else dt:=(W-cs)/2-7.5;
      for g:=1 step 1 until 4 do
      begin dt:=dt+3; db:=0; sum:=0;
      [illegible]
```

```
            loop2: db:=db+cs; sum:=sum+1;
            if db/dt≥1 then goto calc1 else goto loop2;
            calc1: AV[i,sum]:=AV[i,sum]+nu×0.2×(db-dt)/cs;
            AV[i,sum+1]:=AV[i,sum+1]+nu×0.2×(cs-db+dt)/cs;
        end;
    end;

    for j:=1,2,3 do
    begin if j=1 then
        begin dt:=we+la-cs/2;
            dt1:=dt+(nl-1)×la;
            sum:=0; db:=0; i:=1;
        end;
        if j=2 then
        begin dt:=we-cs/2;
            dt1:=(W-cs)/2-4.5;
            sum:=0; i:=2; db:=0;
        end;
        if j=3 then
        begin dt:=(W-cs)/2+4.5;
            dt1:=W-wc-cs/2; i:=2; sum:=0;
            sum:=entier(dt/cs); db:=sum×cs;
        end;
        loop3: db:=db+cs; sum:=sum+1;
        if db/dt≥1 then goto calc2 else goto loop3;
        calc2: if db≥dt1 then goto calc3 else
        begin HA[i,sum+1]:=HA[i,sum+1]+(db-dt)×UDL[2]/3
            ×(cs-(db-dt)/2)/cs;
            HA[i,sum]:=HA[i,sum]+(db-dt)×UDL[2]/3×(db-dt)
                                          /(2×cs);
            dt:=db; db:=db+cs; sum:=sum+1;
            goto calc2;
        end;
        calc3: HA[i,sum]:=HA[i,sum]+(cs-db+dt1)×UDL[2]/3
                              ×(cs-(cs-db+dt1)/2)/cs;
               HA[i,sum+1]:=HA[i,sum+1]+(cs-db+dt1)×UDL[2]/3
                              ×(cs-db+dt1)/(2×cs);
    end;

    for f:=1,2,3,4,5,6 do
    begin if f<3 then x:=1 else x:=2;
        h:=4×(tbfs-tbfc)/L[x,2];
        h1:=4×(ds-tbfs-dc+tbfc)/L[x,2];
        for j:=1 step 1 until 13 do
        begin if f<3 then x1:=L[1,2]-L[1,1]+(f-1)/3×L[1,1]+
                      (j-1)×L[1,1]/36
            else x1:=(f-4)/6×L[2,2]+(j-1)×L[2,2]/72;
            tf:=h×(x1-x1↑2/L[x,2]); tf:=tbfs-tf;
            y1:=h1×(x1-x1↑2/L[x,2]); y1:=ds-ttf-tbfs-y1;
```

```
          find I(y1,I,tf,12×cs,y);
          HF[1,j]:=HF[2,j]:=Q[j]×(if f≤3 then L[1,1]/9
                else L[2,2]/18)/(I×E);
          J[j]:=(4×(12×cs×(y1+ttf/2+tf/2))↑2)/(12×cs×(
                12×cs/ttf+12×cs/tf+4×(y1+ttf/2+tf/2)/tw));
     end;
     for x:=1,2 do
     for j:= 1 step 1 until 13 do HF[x,j]:=HF[x,j]×H[x,j];

     f11:=dot(HF[1,j],H[1,j],1,13,j);
     f12:=dot(HF[1,j],H[2,j],1,13,j);
     f22:=dot(HF[2,j],H[2,j],1,13,j);
     P[f+7,1]:=1; P[f+7,2]:=0;
     P[f+7,3]:=if f≤3 then L[1,1]×4else L[2,2]×2;
     P[f+7,5]:=f12/f22;
     P[f+7,6]:=f12/f11;
     P[f+7,7]:=1/(f11-P[f+7,5]×f12);
     P[f+7,8]:=1/(f22-P[f+7,6]×f12);

     for j:=1 step 1 until 13 do
     J[j]:=(G×J[j]×Q[j])/(36×(if f≤3 thenL[1,1]/3
          else L[2,2]/6));
     P[f+7,4]:=dot(J[j],1,1,13,j);
     k:=7×(nw-2);
     for j:=1 step 1 until nw-1 do
     begin M[k+(f-1)×(nw-1)+j,1]:=(f-1)×(nw-1)+j;
           M[k+(f-1)×(nw-1)+j,2]:=(f-1)×(nw-1)+j+nw-1;
           M[k+(f-1)×(nw-1)+j,3]:=f+7;
     end;
end;
for f:=1,2 do
for i:=1 step 1 until nw-1 do
for j:=1,2,3 do
begin k:=(3×(nw-1)+(j-1)×(nw-1)+i)×3;
     B[k,f]:=B[k,f]+HA[f,i]×L[2,2]/12;
     k:=(3×(nw-1)+j×(nw-1)+i)×3;
     B[k,f]:=B[k,f]+HA[f,i]×L[2,2]/12;
end;
for f:=1,2 do
for i:=1 step 1 until nw-1 do
begin k:=3×(5×(nw-1)+i);
     B[k,f]:=B[k,f]+AV[f,i]×(60+96)/L[2,2];
     k:=3×(6×(nw-1)+i);
     B[k,f]:=B[k,f]+AV[f,i]×((L[2,2]/6-10)/
               (L[2,2]/6)+(L[2,2]/6-16)/(L[2,2]/6));
end;
x:=(nw-1)×6+(nw-2)×7;
m:=(nw-1)×6+(nw-2)×7;   y:=2;
```

```
for i:=1 step 1 until m do
begin integer Mij3; real P1,P2,P3,P4,P5,P6,P7,P8;
    g:=M[i,3];
    P1:=P[g,1]; P2:=P[g,2]; P3:=P[g,3]; P4:=P[g,4];
    P5:=P[g,5]; P6:=P[g,6];P7:=P[g,7]; P8:=P[g,8];
    Mij3:=3×M[i,1];
    S[Mij3-2,1]:=S[Mij3-2,1]+P1↑2×P4+P2↑2×P7;
    S[Mij3-2,2]:=S[Mij3-2,2]+P1×P2×(P4-P7);
    S[Mij3-2,3]:=S[Mij3-2,3]-P2×(P7+P6×P8)/P3;
    S[Mij3-1,1]:=S[Mij3-1,1]+P2↑2×P4+P1↑2×P7;
    S[Mij3-1,2]:=S[Mij3-1,2]+P1×(P7+P6×P8)/P3;
    S[Mij3,1]:=S[Mij3,1]+(P7+P6×P8+P8+P5×P7)/P3↑2;
    Mij3:=3×M[i,2];
    S[Mij3-2,1]:=S[Mij3-2,1]+P1↑2×P4+P2↑2×P8;
    S[Mij3-2,2]:=S[Mij3-2,2]+P1×P2×(P4-P8);
    S[Mij3-2,3]:=S[Mij3-2,3]+P2×(P8+P5×P7)/P3;
    S[Mij3-1,1]:=S[Mij3-1,1]+P2↑2×P4+P1↑2×P8;
    S[Mij3-1,2]:=S[Mij3-1,2]-P1×(P8+P5×P7)/P3;
    S[Mij3,1]:=S[Mij3,1]+(P7+P6×P8+P8+P5×P7)/P3↑2;
    comment above statements set up diagonal elements
                    in the S matrix;
    t:=3×abs(M[i,1]-M[i,2]);
    r:= 3×M[i,1];
    S[r-2,t+1]:=-P1↑2×P4+P2↑2×P6×P8;
    S[r-2,t+2]:=-P1×P2×(P4+P6×P8);
    S[r-2,t+3]:=+P2×(P7+P6×P8)/P3;
    S[r-1,t]:=-P1×P2×(P4+P6×P8);
    S[r-1,t+1]:=-P2↑2×P4+P1↑2×P6×P8;
    S[r-1,t+2]:=-P1×(P7+P8×P6)/P3;
    S[r,t-1]:=-P2×P8×(1+P6)/P3;
    S[r,t]:=P1×P8×(1+P6)/P3;
    S[r,t+1]:=-(P7+P6×P8+P8+P5×P7)/P3↑2;
end above statements set up off diagonal elements;

for i:=1 step 1 until nw-1,3×(nw-1)+1 step 1 until 4×(nw-1),
    6×(nw-1)+1 step 1 until 7×(nw-1) do
begin if i≤nw-1 then begin t:=3×i; r:=3; end;
    if i>nw-1 and i<6×(nw-1)+1then begin t:=3×i; r:=3; end;
    if i>6×(nw-1) then begin t:=3×i; r:=2; end;
    for j:=1 step 1until y do B[t+r-3,j]:=0;
    for s:=1 step 1 until t+r-4 do if t+r-s-2 ≤w
                  then S[s,t+r-s-2]:=0;
    for j:=2 step 1 until w do S[t+r-3,j]:=0;
end above statements impose zero displacements;

solve(S,B,21×(nw-1),3×nw,2);
```

```
for j:=1,2 do
begin integer p,q; real Mx1,My1,Mx2,My2,Fz,Mq1,Mq2,Mp;
    write text(dv,[[3c][8s]LOADING*CASE*No.*]);
    write(dv,format([nd]),j);
    if j=1 then write text(dv,[[c][8s]ABNORMAL*VEHICLE*
        ON*EDGE*OF*BRIDGE]) else write text(dv,[[c][8s]
        ABNORMAL*VEHICLE*AT*CENTRE*OF*BRIDGE]);
    write text(dv,[[2c][8s]MEMBER[3s]FORCES**KIPS[15s]
        MOMENTS*KIP*FT[2c][8s]No.[4s]MEMBER[7s]SHEAR
        [8s]TORSION[6s]END*1[8s]END*2[c]]);
    for i:=1 step 1 until m do
    begin array d[1:14];  g:=M[i,3];
        d[1]:=P[g,1]↑2×P[g,4]+P[g,2]↑2×P[g,7];
        d[2]:=P[g,1]×P[g,2]×(P[g,4]-P[g,7]);
        d[3]:=-P[g,2]×(P[g,7]+P[g,6]×P[g,8])/P[g,3];
        d[4]:=P[g,2]↑2×P[g,4]+P[g,1]↑2×P[g,7];
        d[5]:=P[g,1]×(P[g,7]+P[g,6]×P[g,8])/P[g,3];
        d[6]:=(P[g,7]+P[g,6]×P[g,8]+P[g,8]+P[g,5]
                  ×P[g,7])/P[g,3]↑2;
        d[7]:=-P[g,1]↑2×P[g,4]+P[g,2]↑2×P[g,6]×P[g,8];
        d[8]:=-P[g,1]×P[g,2]×(P[g,4]+P[g,6]×P[g,8]);
        d[9]:=-P[g,2]↑2×P[g,4]+P[g,1]↑2×P[g,6]×P[g,8];
        d[10]:=-P[g,2]×P[g,8]×(1+P[g,6])/P[g,3];
        d[11]:=P[g,1]×(P[g,8]+P[g,7]×P[g,5])/P[g,3];
        d[12]:=P[g,2]↑2×P[g,4]+P[g,1]↑2×P[g,8];
        d[13]:=P[g,1]×P[g,2]×(P[g,4]-P[g,8]);
        d[14]:=P[g,1]↑2×P[g,4]+P[g,2]↑2×P[g,8];

        p:=3×M[i,1]; q:=3×M[i,2];
        Mx1:=d[1]×BI[p-2,j]+d[2]×BI[p-1,j]+d[3]×
             (BI[p,j]-BI[q,j])+d[7]×BI[q-2,j]+d[8]×BI[q-1,j];
        My1:=d[2]×BI[p-2,j]+d[4]×BI[p-1,j]+d[5]×
             (BI[p,j]-BI[q,j])+d[8]×BI[q-2,j]+d[9]×BI[q-1,j];
        Fz:=d[3]×BI[p-2,j]+d[5]×BI[p-1,j]+d[6]×(BI[p,j]-
             B[q,j])+d[10]×BI[q-2,j]+d[11]×BI[q-1,j];
             Mx2:=d[7]×BI[p-2,j]+d[8]×BI[p-1,j]+d[10]×
             (BI[p,j]-BI[q,j])+d[14]×BI[q-2,j]+d[13]×BI[q-1,j];
        My2:=d[8]×BI[p-2,j]+d[9]×BI[p-1,j]+d[11]×
             (BI[p,j]-BI[q,j])+d[13]×BI[q-2,j]+d[12]×BI[q-1,j];

        Mq1:=(-Mx1×P[g,2]+My1×P[g,1])/12;
        Mq2:=(-Mx2×P[g,2]+My2×P[g,1])/12;
        Mp:=(Mx1×P[g,1]+My1×P[g,2])/12;
        for i:=1 step 1 until nw-1 do
        begin if p=3×(nw-1)+1 and q=4×(nw-1)+1 and
                                 Mq1<Mneg then Mneg:=Mq1;
              if p=5×(nw-1)+1 and q=6×(nw-1)+1 and
                                 Mq2>Mpos thenMpos:=Mq2;
       end;
```

```
            write(dv,format([8s ndd]),i);
            write text(dv,[)]);
            write(dv,format([sndd]),M[i,1]);
            write text(dv,[*-]);
            write(dv,format([ndd]),M[i,2]);
            write(dv,format([5s+ndd.ddd]),Fz);
            write(dv,format([4s+nddd.ddd]),Mp);
            write(dv,format([4s+nddd.ddd]),Mq1);
            write(dv,format([4s+nddd.dddc]),Mq2);
        end;
        write text(dv,[[3c][9s]JOINT[4s]X*ROTATION
            [8s]Y*ROTATION[8s]VERTICAL[c]]);
        for i:=1 step 1 until 7×(nw-1) do
        begin write(dv,format([9sndd]),i);
                write text(dv,[)]);
            write(dv,format([3s+nd.ddddddd]),B[3×i-2,j]);
            write(dv,format([6s+nd.ddddddd]),B[3×i-1,j]);
            write(dv,format([6s+nd.dddddddc]),B[3×i,j]);
        end;
    end;

    k:=0;
    if Mpos>M1[1,21] then
    begin x1:=Mpos/M1[1,21]; k:=k+1;
        for i:=1 step 1 until 21 do M1[1,i]:=M1[1,i]×x1;
    end;
    if Mneg<M1[2,11] then
    begin x1:=Mneg/M1[2,11]; k:=k+1;
        for i:=1 step 1 until 21 do M1[2,i]:=M1[2,i]×x1;
    end;
    if k>0 then goto pick;
    pick: prelim sizes(M1,L,Qs,B);
    if abs(Weight2-Weight1)/Weight1>0.05 then goto pick;
    final sizes: for f:=1,2 do
    begin real l; h1:=4×(ds-dc)/L[f,2];
        l:=if f=1 then L[1,1]/10 else L[2,2]/20;
        for i:=1 step 1 until 11 do
        begin if f=1 then x1:=L[f,2]-L[f,1]+(i-1)×l
            else x1:=(i-1)×l;
            h:=Ic×(1+R×(1-x1/L[f,2])↑2)↑2;
            tf:=iftbfc>tbfs then tbfc else tbfs;
            y1:=h1×(x1-x1↑2/L[f,2]); y1:=ds-y1;
            loop: find I(y1-ttf-tf,I,tf,12×cs,y);
            if I>h then
            begin tf:=tf-0.2;
                goto loop;
            end;
            k:=(f-1)×10+i;
            M1[10,k]:=tf+0.2; M1[12,k]:=y;
            M1[11,k]:=y1;
            M1[4,k]:=I/y;
            M1[5,k]:=I/(y1-y);
            M1[6,k]:=12×cs×(ttf+M1[10,k])+y1×tw;
            if M1[3,k]>0 then
            begin ft:=fmint-12×(M1[2,k]+M1[3,k])/M1[5,k];
                fb:=fminw/Ro+12×(M1[1,k]+M1[3,k])/(Ro×M1[4,k]);
                M1[7,k]:=M1[6,k]×(fb×M1[4,k]+ft×M1[5,k])/
                        (M1[4,k]+M1[5,k]);
                M1[7,k]:=Ps×(entier(M1[7,k]/Ps)+1);
```

```
                if x1>M1[8,k] then M1[8,k]:=x1;
                x1:=M1[4,k]/M1[6,k]-M1[4,k]×fminw/(Ro×M1[7,k]
                    )-(M1[3,k]+M1[1,k])/(Ro×M1[7,k]);
                M1[9,k]:=-M1[5,k]/M1[6,k]+M1[5,k]×fcw/(Ro×M1[7,k])
                    -(M1[3,k]+M1[1,k])/(Ro×M1[7,k]);
                if x1<M1[9,k] then M1[9,k]:=x1;
            end else
            begin ft:=fminw/Ro-12×(M1[3,k]+M1[2,k])/(Ro×M1[5,k]);
                fb:=fmint+12×(M1[3,k]+M1[2,k])/M1[4,k];
                M1[7,k]:=M1[6,k]×(fb×M1[4,k]+ft×M1[5,k])/
                                 (M1[4,k]+M1[5,k]);
                M1[7,k]:=Ps×(entier(M1[7,k]/Ps)+1);
                x1:=-M1[5,k]/M1[6,k]+M1[5,k]×fct/M1[7,k]
                    -(M1[3,k]+M1[1,k])/M1[7,k];
                M1[9,k]:=M1[4,k]/M1[6,k]-M1[4,k]×fmint/M1[7,k]-
                          (M1[3,k]+M1[1,k])/M1[7,k];
                if x1<M1[9,k] then M1[9,k]:=x1;
                x1:=-M1[5,k]/M1[6,k]+M1[5,k]×fminw/(Ro×M1[7,k])-
                    (M1[2,k]+M1[3,k])/(Ro×M1[7,k]);
                M1[8,k]:=M1[4,k]/M1[6,k]-M1[4,k]×fcw/(Ro×M1[7,k])-
                    (M1[2,k]+M1[3,k])/(Ro×M1[7,k]);
                if x1>M1[8,k] then M1[8,k]:=x1;
            end;
        end;
    end;

    write text(dv,[[8s]THICKNESS*OF*TOP*FLANGE*=*]);
    write(dv,format([nd.dd]),ttf); write text(dv,[*INCHES[c]]);
    write text(dv,[[8s]THICKNESS*OF*WEBS*=*]);
    write(dv,format([nd.dd]),tw); write text(dv,[*INCHES[c]]);
    write text(dv,[[8s]CENTRES*OF*WEBS*=*]);
    write(dv,format([nd.dd]),cs); write text(dv,[*FEET[c]]);
    write text(dv,[[11s]DEPTH****PRESTRESSING[7s]CABLE*ZONE[8s]
                THICKNESS[c][8s].[12s]FORCE[8s]UPPER[5s]LOWER
                [6s]BTM.*FLG.[c]]);
    for i:=1 step 1 until 21 do
    begin write(dv,format([8snd]),i);
        write text(dv,[)]);
        write(dv,format([snd.dd]),M1[11,i]);
        write(dv,format([4s+ndsdddsddd]),M1[7,i]);
        write(dv,format([4s-nd.dd]),M1[11,i]-M1[12,i]-M1[9,i]);
        write(dv,format([4s-nd.dd]),M1[11,i]-M1[12,i]-M1[8,i]);
        write(dv,format([6s+nd.ddc]),M1[10,i]);
    end;
end;

begin real f11,f12,f22,f21,a1,a2,e1,e2,e3,Mom,add;
    integer total;
    array B[1:31];
```

```
begin array H[1:2,1:31],HF[1:2,1:31];
    integer array Q[1:31];
    for i:=1,2 do
    for j:=1 step 1 until 31 do H[i,j]:=0;
    for i:=1 step 1 until 11 do
    begin H[1,i]:=(i-1)/10;
          H[1,i+10]:=(11-i)/10;
          H[2,i+10]:=(i-1)/10;
          H[2,i+20]:=(11-i)/10;
    end;
    Q[1]:=Q[31]:=1;
    for i:=2 step 2 until 30 do Q[i]:=4;
    for i:=3 step 2 until  29 do Q[i]:=2;
    for i:=1 step 1 until 10 do
    HF[1,32-i]:=HF[2,32-i]:=HF[1,i]:=HF[2,i]:=Q[i]×L[1,2]/
          (M1[12,i]×120);
    for i:=1 step 1 until 6 do
    HF[2,i+10]:=HF[2,22-i]:=HF[1,i+10]:=HF[1,22-i]:=Q[i]×
          L[2,2]/(120×M1[12,9+2×i]);
    for k:=1,2 do
    for j:=1 step 1 until 31 do HF[k,j]:=HF[k,j]×H[k,j];
    f11:=dot(HF[1,j],H[1,j],1,31,j);
    f12:=dot(HF[1,j],H[2,j],1,31,j);
    f22:=dot(HF[2,j],H[2,j],1,31,j);
    f21:=dot(HF[2,j],H[1,j],1,31,j);
    add:=e1:=0; total:=0;
    e2:=(M1[8,11]+M1[9,11])/2;
    e3:=(M1[8,21]+M1[9,21])/2;
    again: e1:=e1+add; total:=total+1;
    for i:=1 step 1 until 10 do
    B[i]:=B[32-i]:=M1[7,i]×(e1+(e2-e1)×(i-1)/10);
    for i:=1 step 1 until 6 do
    B[10+i]:=B[22-i]:=((e3-e2)×(i-1)/10+e2)×M1[7,9+2×i];
    a1:=dot(HF[1,j],B[j],1,31,j);
    a2:=dot(HF[2,j],B[j],1,31,j);
    Mom:=-a1/(f11+f12);
    for i:=1 step 1 until 11 do
    e:=Mom×(i-1)/(10×M1[7,i]);
    for i:=2 step 1 until 6 do
    e:=Mom/M1[7,9+2×i];
    e:=x1:=0;
    if e2+Mom/M1[7,11]>M1[9,11] or e3+Mom/M1[7,21]>M1[9,21]
            then add:=+1 else add:=0;
    if e2+Mom/M1[7,11]<M1[8,11] or e2+Mom/M1[7,21]<M1[8,21]
            then add:=-1 else add:=0;
    if add≠0 and total<10 then goto again;
    write text(dv,[[c][8s]ECCENTRICITY*OF*CABLE*AT*
            OUTER*SUPPORT*=*]);
```