DYNAMIC PROGRAMMING AND DIRECT ITERATION FOR THE OPTIMUM DESIGN OF SPACE STRUCTURES

by

G.C. Howell

A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science in the Faculty of Engineering University of Cape Town

Department of Civil Engineering
UNIVERSITY OF CAPE TOWN

April 1978

The University of Cape Town has been given the right to reproduce this thesis in whole or in part. Copyright is held by the author.

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

DECLARATION OF CANDIDATE

I, Graham Conrad Howell, hereby declare that this thesis is substantially my own work and that it has not been submitted for a degree at another University.

Signed by candidate

DEDICATION

I would like to dedicate this thesis to my parents Mr and Mrs H.C. Howell

ACKNOWLEDGEMENTS

My sincere appreciation is extended to:

Mr W.S. Doyle* under whose supervision this thesis was conducted, for his constant help, encouragement, enthusiasm and the vital suggestions which made this project possible.

The Council for Scientific and Industrial Research for their financial assistance.

Mrs Lee Behm for her immaculate typing of the manuscript.

Mr Harold Cable for the printing of this thesis.

* Senior Lecturer, Department of Civil Engineering
University of Cape Town

ABSTRACT

A computer technique is proposed for a simple practical method of automatically designing skeletal structures. Dynamic programming is used to find the optimum geometric configuration of the structural members, while the member sizes are proportioned by direct iteration.

The computational effort required to find the best possible design for large structures can become unmanageable without the use of Dynamic Programming. This technique simplifies this problem by a process of intermediate decisions which are made at each stage of the solution.

Dynamic Programming is applied to tower structures which can be regarded as discrete substructures. The configuration of each substructure is defined at its upper and lower interfaces by a set of state variables. An optimum weight design can be found by selecting the best configuration and hence the best state variables at each interface. Each alteration of the geometric configuration of a substructure effects its weight. Consequently, a series of decisions based on accumulated weight must be made so that the chosen configuration at each interface produces the optimum weight design for the entire structure.

NOTATION

Upper Case Characters

$\{A\}$	square symmetric matrix
A_i	cross sectional area of member i
C	initial value of state variable
$C_i(k)$	accumulate cost up to stage k
C_o	Euler critical stress
C_s	stress ratio
D_i, D_j	fixed dimensions at stage k
Fx,Fy,Fz	equivalent loads on interface
FX( ),FY( ),FZ( )	equivalent interface nodal loads
J	total cost
J_n	total cost for dimension n
$[K]$	stiffness matrix
K_2	load factor or coefficient
Mx,My,Mz	equivalent moments on interface
$\{P\}$	vector of nodal loads
Ph,Pv	external loads
U( )	admissible set of control variables
X( )	admissible set of state variables
X,Y,Z	coordinate axis system
YS	minimum yield stress in MPa

Lower Case Characters

$\underline{b}$	general vector of nodal loads
f	n-dimensional vector functional
k	stage index
ℓ	function for the cost of a single stage
ℓ/r	slenderness ratio
(or L/R)	slenderness ratio for computer use
m	cycle index
n	degree of dimensionality
$t_{ij}(k)$	cost of a single stage (k-1 to k) between value i and j
$\{u\}$	vector of displacements
$u_n(\ )$	n-dimensional control variable vector
$x_n(\ )$	n-dimensional state variable vector
$\underline{x}$	general vector of unknowns

Greek Characters

σ_a	actual member stress
σ_p	permissible member stress
η	function of the slenderness ratio

CONTENTS

CHAPTER 1

INTRODUCTION

The design of the most efficient structure is generally the ultimate goal of every practising engineer. Many different criteria exist for the achievement of this aim. These include: least weight; fully stressed design; and best geometric configuration. Methods have been devised using these criteria, for the optimum design of space truss structures in particular. A number of mathematical techniques are difficult to apply to practical structures. The main reason for this is that these methods become unmanageably complex when allowance is made for the many variables which exist.

In this thesis, complicated mathematical techniques have been avoided. More attention has been devoted to efficient and practical methods of automatic computer design. A technique has been used in which the optimum geometric configuration together with the optimum member sizes are determined simultaneously.

Practical aspects which have been included are the use of a list of actual section sizes and the subdivision of the structure into member groups. The most suitable member sizes are selected from the lists provided. This ensures that both the analysis and design processes are carried out using 'true' values. The reduction of the variety of sections in a structure to a desirable minimum, is another aspect of practical design. This has been accomplished by preselecting member groups. Many of the other techniques do not recognise these aspects of design and merely produce academic solutions.

The technique of Dynamic Programming has been chosen to determine the optimum member configurations since it lends itself admirably both to the design process and to the deployment of these practical considerations. This technique was originally devised by Bellman [1,2] for the solution of optimum control problems. Considerable work on the theory has been done by Dreyfus [2], Aris [3] and Larson [4]. The amount of calculation involved in large problems is, however, prohibitive. Consequently, methods have been found to curtail the calculations, as reported by Larson [5]. These methods include the Dynamic Programming Successive Approximations technique [6] which is used in this thesis.

The nature of the problems which can be solved by a Dynamic Programming technique, is essentially segmental. These methods all use a systematic decision process to arrive at an optimum control sequence, or, as applicable here, an optimum geometric configuration. This segmental approach has led to the idea of substructuring. The analysis calculations are greatly reduced by this process since each substructure, containing relatively few members, is analysed separately. Reports by Palmer and Sheppard [7,8,9,10] and P. Packia Raj and S. Olani Durrant [11] have used the conventional Dynamic Programming technique to design pin-jointed structures which include transmission towers. They were particularly concerned with the geometric configurations, the member design was only of secondary importance. Consequently, they used approximate methods of analysis, by degenerating the structure into a determinate system.

The displacement method of analysis is used in this thesis to accurately predict the member forces in each substructure. The DPSA technique is used in conjunction with this analysis method to produce an optimum configuration. For all structures, this technique degenerates the system into a series of single variable Dynamic Programming calculations. The Direct Iteration method [12] is used to select the optimum member sizes in each substructure. The member sizes are chosen so that the ratio between actual and permissible stresses for all members is as close to unity as possible.

The application of these techniques for the structural and geometric design of tower structures is discussed in detail with special reference to three types of towers:

(a) Plane truss towers

(b) Rectangular 3-dimensional towers

(c) Triangular 3-dimensional towers

The methods described, however, are not restricted to the specific design of towers. Domes, pyramids, space truss girders and indeed any cantilever type structure can be designed with this method.

Three computer programs have been written.

1. DYSPAN, is a general purpose program for the design of any type of space truss structure which has a specified geometric configuration. This program uses the Direct Iteration method to design the member sizes.

2. DYNGEO, uses the Dynamic Programming Successive Approximations technique, the Direct Iteration method and substructuring to calculate the optimum configuration and member sizes for least weight.

3. DYNPRE, uses the same techniques as in DYNGEO but also employs an interpolation method for the prediction of member forces to decrease computer calculation times.

The designs produced by these programs are very similar, although member forces are sometimes inconsistent. This is inconsequential, however, since these forces are negligible and their design is based entirely upon their slenderness ratios.

These three programs show that the techniques are indeed feasible for the design of tower structures. A survey of the published literature has revealed that optimum geometric configurations have not been correlated with optimum member sizes for skeletal tower structures. For this reason, no direct comparisons of results can be made. Consequently, the results obtained from these three programs are the only means of assessing the validity of the designs.

Additional Notes

The following points should be borne in mind when an appraisal of the techniques involved in this thesis are considered. These notes are included here in order to augment the basic assumptions which are made in the course of the argument.

(a) The structure can be sub-divided into substructures for analysis purposes. This therefore assumes that the forces transferred from one substructure to the next can be calculated from statics. The implication is that each substructure is effectively externally statically determinate, but it will be shown that this is not unduly significant for design purposes.

(b) As the structure is designed segmentally, it is assumed that the overall stability of the structures is adequate and checked by some other means. This is, however, outside the range of this thesis.

(c) In the design of individual members, the use of the ratio of the actual stress to permissible stress for optimization between member sizes is based on the assumption that the range of members being compared have approximately the same ratio of area to radius of gyration.

CHAPTER 2

STRUCTURAL DESIGN BY DYNAMIC PROGRAMMING

2.1 INTRODUCTION

Dynamic programming has long been recognised as an extremely powerful optimization technique, particularly for problems of a discontinuous nature. High-dimensional problems, however, result in a large amount of computation, but this can be reduced by a successive approximation method.

A 3-dimensional Dynamic Programming Successive Approximation (DPSA) technique is used here to obtain an optimum (least weight) geometrical configuration for the design of tower structures. Concurrently, a simple direct iteration procedure is used to select the optimum member sizes from any list of section properties provided.

If the structure were to be designed as a whole, a change of geometric configuration would need a re-analysis. This means that a large amount of computation would be required.

Substructuring has been introduced to reduce the overall problem into smaller stages so that the analysis can be performed more rapidly.

This Chapter deals with the basic aspects of Dynamic Programming. The DPSA technique, which degenerates the n-dimensional problem into a series of single-dimensional problems, is discussed. Examples of the use of this method are given in Chapter 4 and a discussion of the computer programming techniques in Chapter 3. Supplementary details of both the original Dynamic Programming method and the DPSA technique are given in Appendices B and C.

2.2 DYNAMIC PROGRAMMING - BASIC CONCEPTS

The computational effort required to find all the possible solutions for large problems can become unmanageable without the use of Dynamic Programming. This technique by-passes this problem by considering the possible decisions to be made at each stage of the solution.

Dynamic Programming can be described as a technique for methodically selecting an optimum solution of a multi-stage decision problem. This mathematical technique can be used for problems where a sequence of decisions are dependent upon one another and each decision influences the system's response to future decisions. A set of solutions can be categorized so that one may be judged to be better than another in some pre-defined manner.

In a sequence of decisions, the current state of the sequence is assessed as $f(x(k-1))$ and the succeeding one as $f(x(k))$. Without considering the whole chain of past and future decisions, except that they contribute to $f(x(k-1))$, the best decision can be found from:

$$f(x(k)) = \min[t(x(k); x(k-1)) + f(x(k-1))] \tag{2.1}$$

where t is the assessment between stages k-1 and k and

x is a state variable

Dynamic Programming is based on a repeated use of this idea.

Dynamic Programming is ideally suited to problems of a systems control nature. Examples of these include: the scheduling of airline flight plans, the optimum operation and planning of Water Resource systems and the optimum use of electrical power reticulation systems. All these problems have a common denominator - they are all segmental operations and therefore can be formulated as follows:

(i) From a set of linear or non-linear systems equations of the form:

$$x(k+1) = f[x(k),u(k),k] \tag{2.2}$$

where x is an n-dimensional state variable

u is an n-dimensional control variable

k is an index for the stage variable

and f is an n-dimensional vector functional

(ii) With a set of systems constraints

$$x \in X(k) \tag{2.3a}$$

$$u \in U(k) \tag{2.3b}$$

where X(k), U(k) are admissible sets of the state and control variables at stage k.

(iii) And an initial state given by

$$x(0) = C \tag{2.4}$$

(iv) An optimum solution, given by the control sequence u(0),...,u(K), of the variational performance criterion

$$J = \sum_{k=0}^{K} \ell[x(k),u(k),k] \tag{2.5}$$

where J = total cost

and ℓ = cost of a single state

can be found so that J is minimized, (or maximized depending on the type of problem) subject to the constraints imposed on the systems equations.

Each problem is solved segmentally by the repeated use of equation (2.1).

A simple network problem is given in Appendix B to demonstrate the process. In the course of the solution of this problem, two central ideas are used. The first is the idea of imbedding; this means that the overall optimization problem consists of a number of smaller problems imbedded within the whole, each can be solved independently. In the second idea, an optimum solution can be found from a sequence of decisions by imagining that the final solution is broken up into a series of simpler decisions.

The structural design problem is considered in the same terms as above. Equation (2.2), the systems equation, is required to describe the configuration of the tower structure at pre-defined positions. This can be done by relating the configuration at the $k+1^{th}$ stage to that of the k^{th} stage by a mathematical equation of the type shown (equation (2.2)). A second method is to relate the configuration to a set of convenient designer-chosen dimensions.

Since the object of this thesis is to produce a practical method of designing towers, the latter method has been used. Equation (2.2) can therefore be re-written as a series of constants:

$$x(k) = D_i \quad \text{for } i = 1 \text{ to } I \text{ at stage } k \tag{2.6a}$$

$$x(k+1) = D_j \quad \text{for } j = 1 \text{ to } J \text{ at stage } k+1 \tag{2.6b}$$

where D is a dimension at stage k or k+1

and i,j are indices for the number of dimensions at stage k and k+1 respectively.

This does not effect the method of solution or the equation of sequential decisions (2.1).

Problems of this type become very tedious when the order of the state variables ($x(k)$) becomes large. For example, various state variable vectors (denoted $x_n(k)$) may be present at any particular stage k of the calculation, which complicates the matter. For this reason, the dimension (n) of the problem is decreased to a single variable vector by the use of the Dynamic Programming Successive Approximation (DPSA) technique.

The geometric design of a tower structure can be degenerated into a series of simple decision processes as prescribed by the DPSA technique by using substructuring.

2.3 SUBSTRUCTURING

The benefit of substructuring in the design of towers is that the optimum section sizes can be determined rapidly within each substructure. The interaction between adjacent substructures is comparatively simple and can be simulated with reasonable ease which makes this technique particularly attractive.

The geometry at the interface of each substructure is uniquely defined by the coordinates of the member ends which are 'cut' at the interfaces. Hence in Figure 2.1 the coordinates at the right-hand side of substructure 1 are defined by x_o, y_o; x_1,y_1 and for substructure 2 by x_1,y_1; x_2,y_2 and so on for the other substructures.

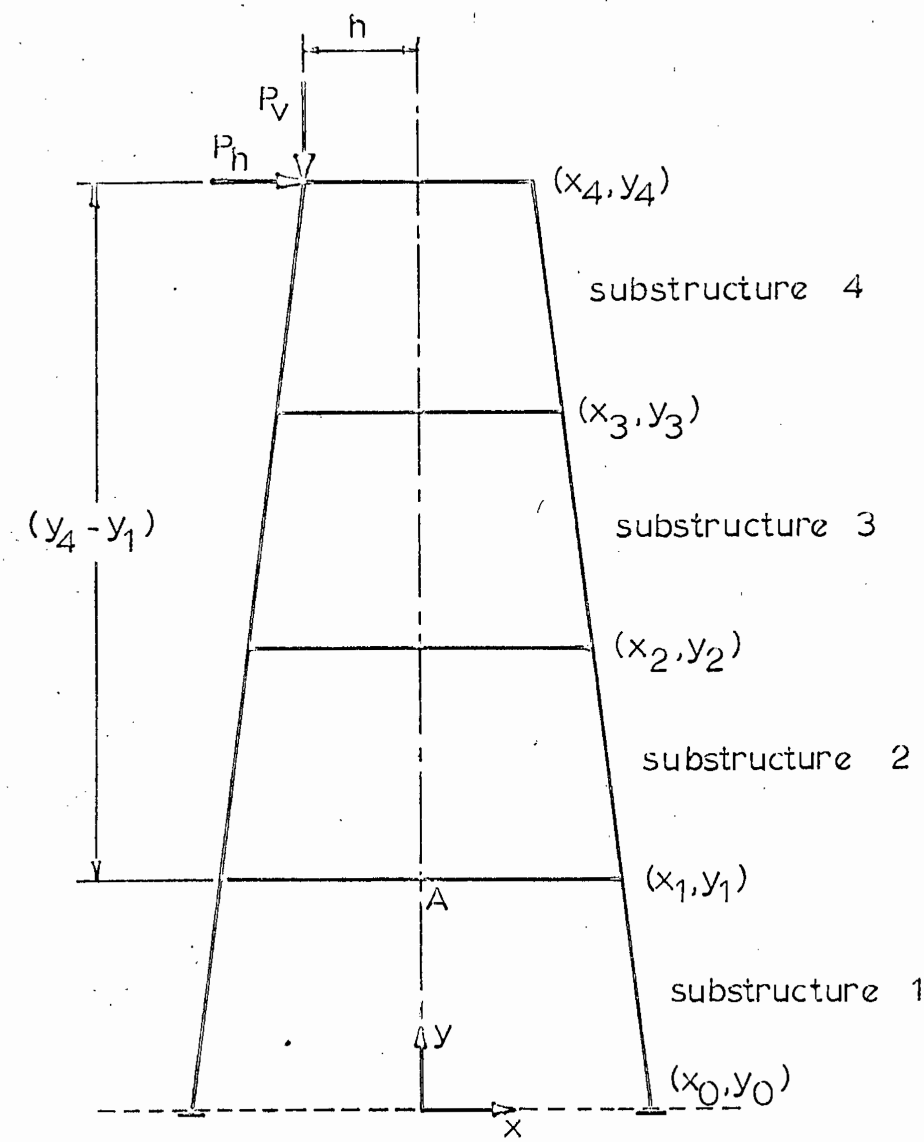


Figure 2.1 Tower Structure with Substructures

The assumption is made that the forces transferred from one substructure to the next can be calculated from statics. For example, the interaction forces at the interface between substructures 1 and 2 are found by applying equivalent loads and moments at the central point A, i.e. the vertical force P_v is resolved into a load P_v and a moment $P_v \times h$, while the horizontal force P_h is resolved into a load P_h and a moment $P_h \times (y_4 - y_1)$.

Each pin-jointed substructure is analysed by the displacement method, which requires the equivalent loads and moments at A to be applied as point loads at the interface nodes (x_i, y_1) and $(-x_i, y_1)$.

The substructures are now analysed and designed independently assuming that the lower interface nodes are constrained. The member sizes are selected by a simple direct iteration procedure. [The details of the required equivalent load calculations have been summarized in Appendix A. The displacement method, as applied to pin-jointed members, is discussed using an Energy Approach in Appendix D].

2.4 DIRECT ITERATION

A list of sections is provided so that the actual member sizes required in each substructure can be selected automatically by the direct iteration procedure. The calculated stresses are compared with permissible stresses so that the ratio between them is a minimum. The substructure is reanalysed each time the member sizes are revised until the results from successive iterations are identical.

2.5 THE DYNAMIC PROGRAMMING SUCCESSIVE APPROXIMATION (DPSA) TECHNIQUE

The coordinates of the interface nodes in Figure 2.1 can be changed at any stage and the members designed accordingly by the direct iteration method. Consequently, the weight of each configuration of members in all substructures can be calculated. The optimum solution is then the combination of possible configurations which results in the least weight of the total structure. The dynamic programming technique sets about the calculation in an organized methodical way.

Let us consider a tower similar to that in Figure 2.1. The interface nodes, which define the configuration of the tower, can be positioned in space by a set of three dimensional coordinate values. Let the structure be symmetrical about the XZ and YZ planes, therefore a node placed in the first octant will define the shape of the tower at that level. Let the coordinates of the interface node be ($x_1(k)$, $x_2(k)$, $x_3(k)$). This corresponds to the x, y and z coordinates of the primary node of substructure 1, where k also denotes the upper interface level of that substructure and k = 0 represents the ground level. The n-dimensional DPSA method is therefore well suited to structures of this type, where n = 3. For example, at level k = 2 some possible values of the state variables are shown in Table 1. The control variables $u_n(k)$ shown correspond to the identification number in each set.

STATE VARIABLES $x_n(k)$			IDENTIFICATION NO $u_n(k)$ n = 1 to 3
$x_1(2)$	$x_2(2)$	$x_3(2)$	$u(2)$
0,8	0,7	0,9	1
0,9	0,75	0,95	2
1,0	0,8	1,0	3
1,1	0,85	1,05	4
1,2	0,9	1,1	5

TABLE 1

A POSSIBLE SET OF STATE AND CONTROL VARIABLE VECTORS

The values of $u(k)$ (i.e. $u_1(k)$, $u_2(k)$, $u_3(k)$) and hence the values of $x_1(k)$, $x_2(k)$, $x_3(k)$ at each level k must be found so that the overall weight of the structure is a minimum.

A mathematical statement of the DPSA method for structural configuration problems can be formulated in similar terms to those used in equations (2.2) to (2.5). The addition of the successive approximation variables alters only the presentation of equation (2.5). The full formulation is as follows:

(i) A system of linear or non-linear equations of the form:

$$x_n(k+1) = f[x_n(k), u_n(k), k] \tag{2.7}$$

which for structural configuration problems is

$$x_n(k) = D_i \quad \text{for } i = 1 \text{ to } I \;\; \& \;\; k = 0 \text{ to } K \qquad (2.7)$$

where n is the degree of dimensionability

x is the state vector

u is the control vector

k is the index for the stage variable

f is a vector function

D is a dimension at stage k

and i is an index for the number of dimensions at stage k

(ii) With systems constraints

$$x_n(k) \in X_n(k) \tag{2.8a}$$

$$u_n(k) \in U_n(k) \tag{2.8b}$$

where $X_n(k)$, $U_n(k)$ are admissible sets of the state and control variables at stage k

(iii) And an initial state

$$x_n(0) = C \tag{2.9}$$

(iv) An optimum solution given by the control sequence $u_n(0),\ldots,u_n(k)$ can be found for the variational performance criterion:

$$J_n^{(m+1)} = \min J_n^{(m)}$$

where
$$J_n = \min \left[\sum_{k=0}^{k=K} \ell_n \, [x_n(k),u_n(k),k)] \right] \tag{2.10}$$

for $n = 1$ to N

and $m = 1$ to ∞

and subjected to cyclical constraints

$$[x_n(k)]_m = [x_n(k)]_{m-1}$$

$$[u_n(k)]_m = [u_n(k)]_{m-1} \tag{2.11}$$

for all n except $n \neq n$.

The DPSA method requires an initial solution to the problem. The average value of the state variables at each interface is a suitable initial solution. To proceed, only one state variable is altered, while the remainder stay at their initial values. In this way a single dimensional dynamic programming procedure with respect to this variable is carried out. The process is then continued, the new value of the first variable is retained and one of the other state variables is altered. All the variables are processed in this way, the first cycle is complete when all the state variables have been altered. Subsequent cycles follow the same procedure as above. The DPSA method has converged to its optimum solution when no weight difference is recorded between successive cycles. This method is found to converge rapidly to an optimum weight solution. The structure comprises a number of substructures. The geometric configuration between interfaces is regarded as a substructure.

The direct iteration method is used to find the most satisfactory structural design and hence the weight for every geometric configuration of each substructure. A least weight path is followed through all the substructures to determine the optimum configuration of the total structure. The explanation of the DPSA method can be simplified by the following elementary example.

2.5.1 Example

Consider a simple 2-dimensional tower which consists of 2 substructures - Figure 2.2.

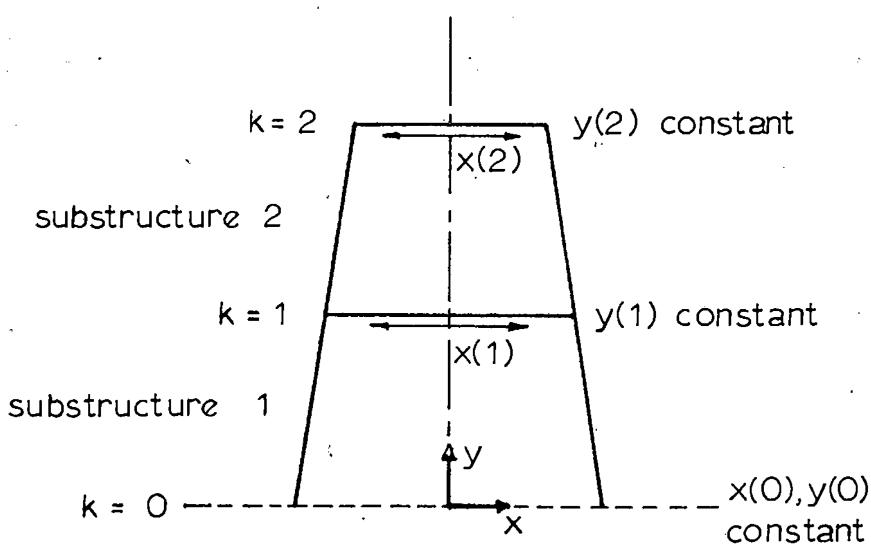


Figure 2.2 Tower Structure

Let the state variables $x(1)$ and $x(2)$ vary in three steps on each side of the vertical axis of symmetry and the other state variables i.e. $y(k)$ at $k = 0$ be constrained to a single value. The accumulated cost (or weight) of the substructures up to level k for each position of x can be denoted by:

$$C_i(k)$$

where i denotes the position of $x(k)$ on interface k.

Let the cost of a single substructure k be denoted by:

$$t_{ij}(k)$$

where i and j denote the positions of the state variables $x(k)$ for substructure k at the upper and lower interfaces respectively.

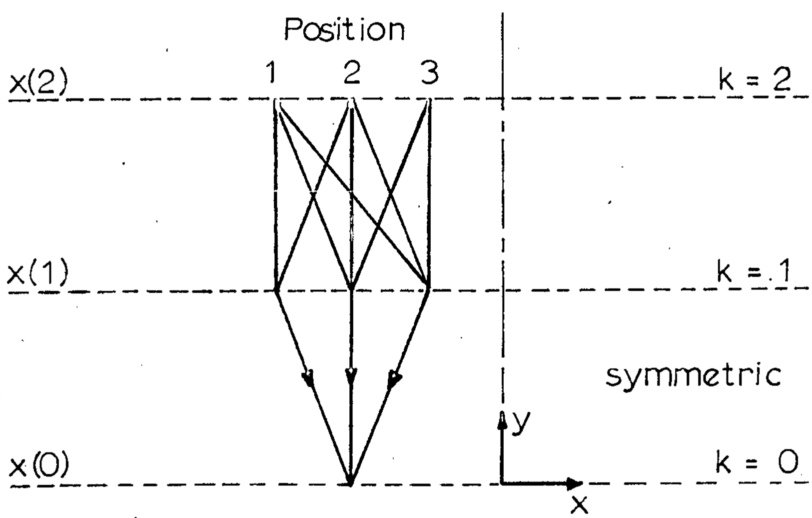


Figure 2.3 Possible Configurations

Figure 2.3 shows all the possible geometrical configurations within the constraints given.

The cost of each of the 3 configurations in the first substructure, due to the varying values of $x(1)$ are calculated by:

$$C_1(1) = t_{11}(1)$$

$$C_2(1) = t_{21}(1)$$

$$C_3(1) = t_{31}(1)$$

and are shown in Figure 2.4

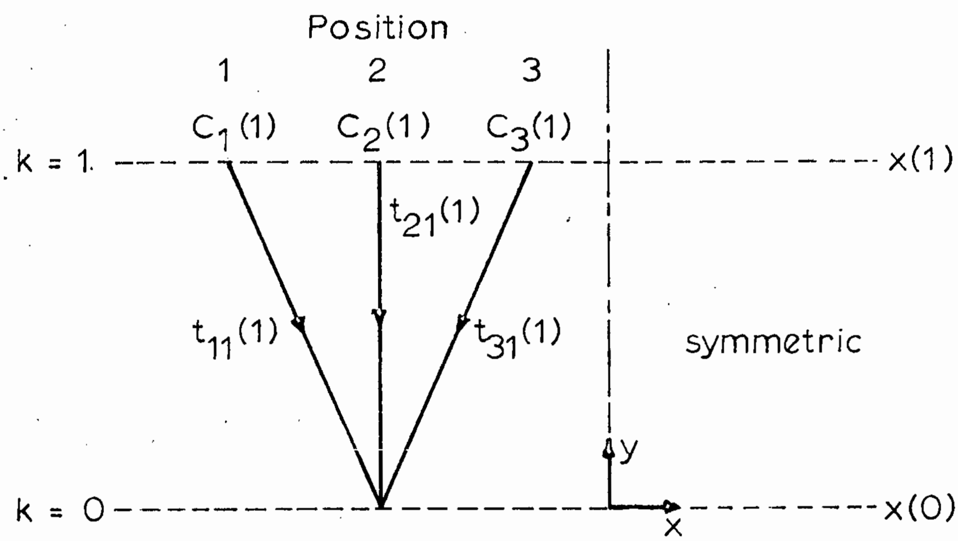


Figure 2.4 Three Cost Values

Similarly the values of $t_{ij}(2)$ can be obtained for the second substructure. The least accumulated weight at point i on interface k can be found from:

$$C_i(2) = \min\ [t_{ij}(2) + C_j(1)] \quad \text{for } j = 1 \text{ to } 3$$

The optimum configuration of the structure at level 2 is determined by choosing the least value of $C_i(2)$. The optimum path can then be traced back through the calculations to find the optimum configuration of the entire structure.

A comprehensive example, related directly to the optimum configuration of towers, is discussed in Appendix C. An actual tower design example would require a series of time consuming analyses to be performed which would confuse the underlying concepts involved in the DPSA method. Consequently, an example is used which is concerned with the optimum configuration of an air conditioning duct so that a minimum area of sheet metal is required. This example has been chosen since the surface area at each stage can be simply calculated.

2.6 SUMMARY

A computer technique is proposed for a simple practical method of automatically designing tower structures. Dynamic programming is used to find the optimum geometric configuration of the structural members, while the member sizes are proportioned by direct iteration.

Tower structures are particularly suited to this method of automatic design since the rapidity of the analysis and design depends primarily upon substructuring. Substructuring of towers is comparatively simple because interaction between adjacent substructures can be simulated with reasonable accuracy. Typical examples are presented to illustrate the method.

CHAPTER 3

COMPUTER PROGRAMMING

3.1 GENERAL

Four programs have been prepared for the design of general space truss structures and tower structures.

(i)	DYSPAN	general program for the structural design of space truss structures which has a specified configuration of members.
(ii)	DYNGEO	design of tower structures for minimum weight by considering both the structural and geometric configuration design and based on the application of Dynamic Programming and Direct Iteration.
(iii)	DYNPRE	program which performs a similar function to DYNGEO but incorporates some computer-time-saving processes.
(iv)	DRAW	program for the plotting of the optimum structure shape from the results of DYNGEO and/or DYNPRE

DYNGEO is a program for the design of space truss structures which uses a direct iteration technique to determine the member sizes. The designer specifies a geometric configuration for which an optimum structural design is found by the program. Hence, in order to find the best configuration for a optimum weight design, a number of configurations must be processed by the program DYSPAN. This requires a large quantity of data to be prepared by the designer who must then scan the designs manually to find the optimum.

The programs DYNGEO and DYNPRE perform these tasks automatically for tower structures. In those programs the structures are analysed and designed as separate substructures. The variation in the configurations are dependent upon the possible dimensions of the interface between substructures. Any possible interface dimension can be used for an initial configuration. Data preparation is thus kept to a minimum. The DPSA technique is then used to choose the best configuration for optimum weight design.

DYNGEO requires a displacement method analysis to be performed for each possible substructure configuration. This becomes time consuming for large problems. Consequently, DYNPRE was written to reduce the number of analyses, and hence the time taken to produce an optimum design.

All programs are written in standard FORTRAN V as it applied to the UNIVAC 1106 computer and the CALCOMP drum plotter.

A description of the programs follows for which it is assumed that the user has, at least,a basic knowledge of FORTRAN V.

User manuals, sample data and results are given in Appendices I, J and K together with computer storage techniques and general subroutine descriptions.

3.2 THE DIRECT ITERATION METHOD OF DESIGN: PROGRAM DYSPAN

3.2.1 Introduction

A computerized linear elastic analysis is used in conjunction with elastic structural design for the proportioning of the member sizes in pin-jointed space trusses.

The technique employed is one of cyclic revision after an initial geometric configuration and a set of section properties have been chosen. The design variables in each cycle relate to the member properties, viz the cross sectional area and their radii of gyration. In this program, a structural design is prepared for a single unique geometric configuration.

The method involves a series of structural modifications, each of which is subjected to an analysis prediction calculation. Subsequent cycles produce better approximations which converge to an optimum design by considering the stress levels in each member. Cyclical re-proportioning of section sizes ultimately leads to the optimal set of design variables being obtained.

The designer can use his experience to great advantage here. If a reasonable initial section selection is made, i.e. the sections chosen are close to the optimum, the computational effort is reduced substantially since the number of cycles required for convergence is reduced.

The technique relies on an analysis method which is of primary importance. The structural design, although it is dependent on the results of the analyses, is nevertheless a vital and integral subdivision of each cycle.

3.2.2 Basis of the Procedure

The Displacement Method is used to analyse the structure. The efficiency of computer storage of the matrices and the solution of the set of simultaneous equations makes this method particularly useful for space truss problems. Details of the solution technique and the computer storage method used in this thesis are given in Appendices E and F.

The procedure is conveniently formulated by a system of matrix equations of the form:

$$[K]\{u\} = \{P\} \tag{3.1}$$

where $\{u\}$ is a vector of unknown displacements or nodal degrees of freedom

$[K]$ is a square symmetric positive definite matrix of stiffness coefficients

and $\{P\}$ is a vector of nodal loads

The solution of equation (3.1) can be found by

$$\{u\} = [K]^{-1}\{P\} \tag{3.2}$$

It is usually unnecessary and undesirable to invert the stiffness matrix explicitly for large systems of equations. Far less computational effort is required to solve for the displacements directly from equation (3.1). The Crout Reduction Method (Appendix E) has been chosen for the solution of the equations since it is both a rapid and an exact method.

The design is carried out in accordance with the recommendations set out in BS 449, Part 2 of 1969.

The analysis of the structure is performed for specific values of the design variables*(cross sectional area and radius of gyration). A set of member forces are calculated from the differential node displacements. The ratio of member stress to permissible stress limits gives an indication of the efficiency of that particular member. By adjusting the design variables, the ratio between the actual and the permissible stresses are minimized to produce the most efficient possible design. Let the ratio of actual to permissible stress be C_s, therefore

$$\frac{\sigma_a}{\sigma_p} = C_s \tag{3.3}$$

For an ideal design C_s equals 1,0 for all members in the structure.

The effect on the forces in each member, P_i, is assumed to remain constant for small changes in member area. This assumption has been verified by a number of examples. Therefore, the required area, A_i, to produce a unit value of C_s is

$$A_i = \frac{P_i}{\sigma_p} \tag{3.4}$$

Let the force obtained from iteration n be $P_i^{(n)}$ and the design variable which was used to calculate this force be $A_i^{(n)}$. Consequently, an improved prediction of the required area for iteration n + 1 is

$$A_i^{(n+1)} = \frac{P_i^{(n)}}{\sigma_p} \tag{3.5}$$

which is used in the analysis.

The calculation of the permissible stress value σ_p for compression members is dependent upon their slenderness ratios (ℓ/r) and the permissible stress of the material used and is calculated from the equation given in the British Standard Code of Practice BS 449.

The permissible stress value of tension members is defined by the material used. In the example shown in this thesis only grade 43 steel has been considered with a permissible stress of 155 MPa.

A cyclical re-analysis and design calculation is carried out until no further improvements of the value of C_s can be found for any member. In this way an optimum stressed design is reached.

3.2.3 DYSPAN: Program Description

This program is structured in modular form i.e. it consists of a main program and a series of independent subroutines. The function of the main program is to read in basic data, coordinate the steps in the design procedure and to control the output of relevant results.

A macro-flow chart of the program (Figure 3.1) shows the steps required in the analysis and design of space truss structures.

Major features of the program include:

(i) Lists of sections from which the structural sections can be chosen.

(ii) Interactive or card input of the data.

(iii) Interactive mode, the designer has complete control of the design calculations, while the computer merely does the 'arithmetic'.

(iv) Re-design of the structure can be performed for different section types without the need for the re-input of a complete set of data.

(v) Loading cases include normal nodal loads, the combination of separate load cases and the inclusion of self-weight and temperature loading in the calculations.

(vi) Dynamic core storage for efficient computer use irrespective of the dimensions of the problem.

(vii) Data input is in 'free-format' throughout.

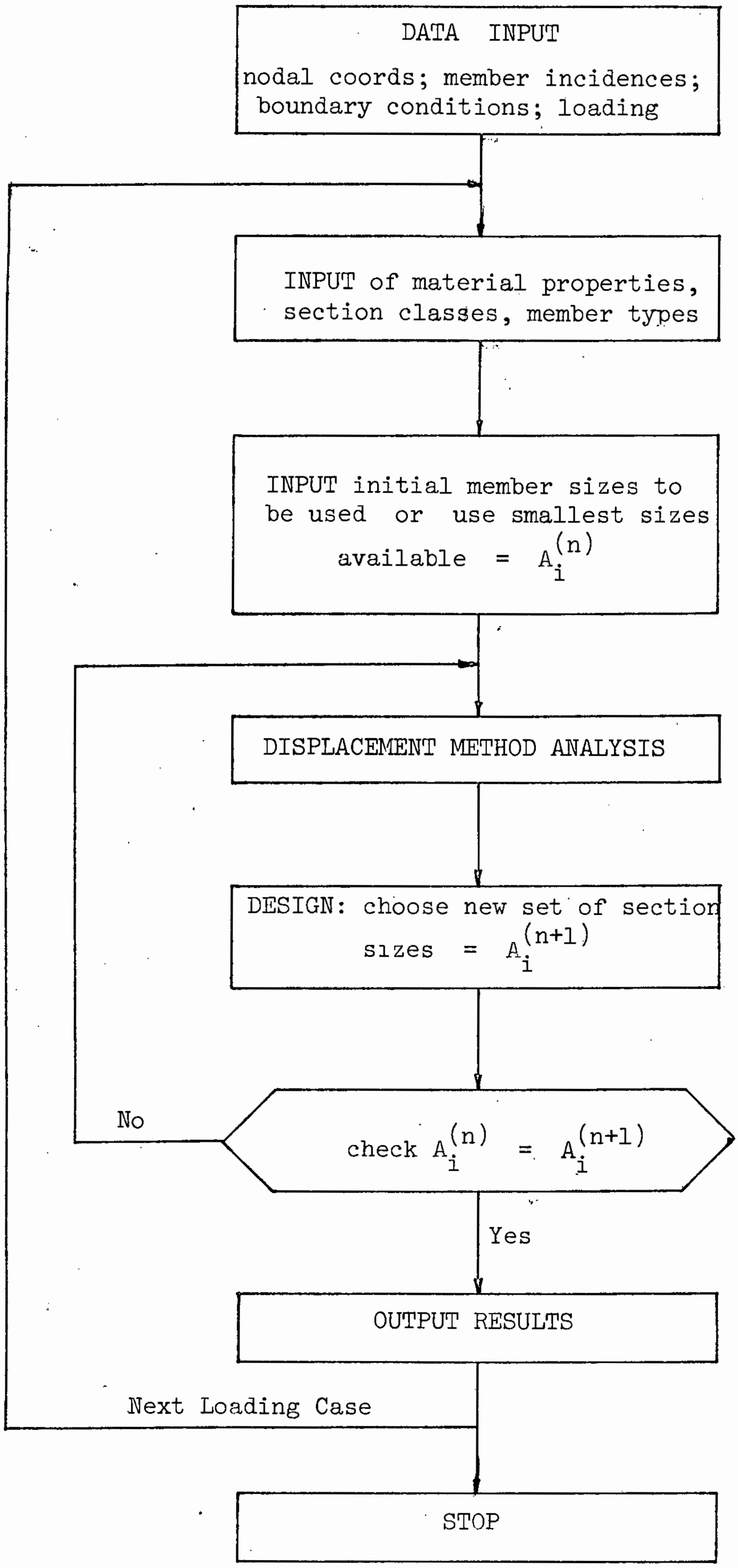


Figure 3.1 A Macro-flow Chart of the Program DYSPAN

3.2.3.1 Lists of Sections

In order to produce a practical, usable computer design, a list of available sections must be provided. The most suitable members are chosen by the program from this list and the analysis and design is performed with these actual listed sections.

The advantage of the method is that the best possible design with the materials available can be produced. Thus, any list of sections can be used and is dependent only on the fabricators available stock.

3.2.3.2 Types of Sections Specified

In order to use this feature in the program,four different section classes have been specified, each containing, for convenience, up to twenty different sized members. Each member is listed with its cross sectional area and radius of gyration. Members are arranged in ascending numerical order with respect to their cross sectional areas. For the purposes of the examples contained in this thesis, the lists of sections contain the following classes:

(i) pipe sections

(ii) single equal angles

(iii) double angles, i.e. two angles, back to back

(iv) channels

The lists of sections are arranged in four two-dimensional arrays labelled PP, AG, DA and CH respectively. The section areas and radii of gyration are automatically read into these arrays.

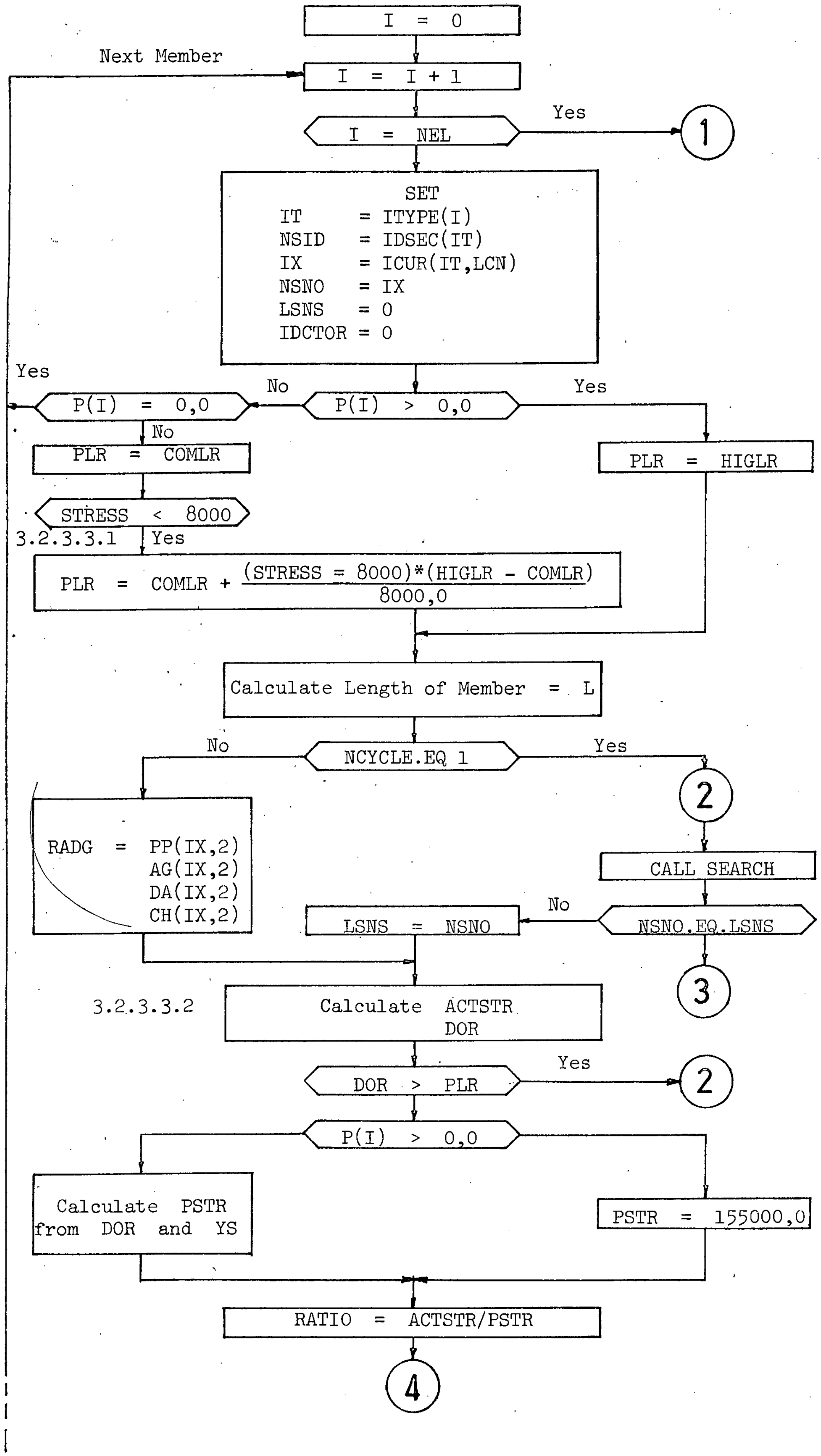
3.2.3.3 Member Design

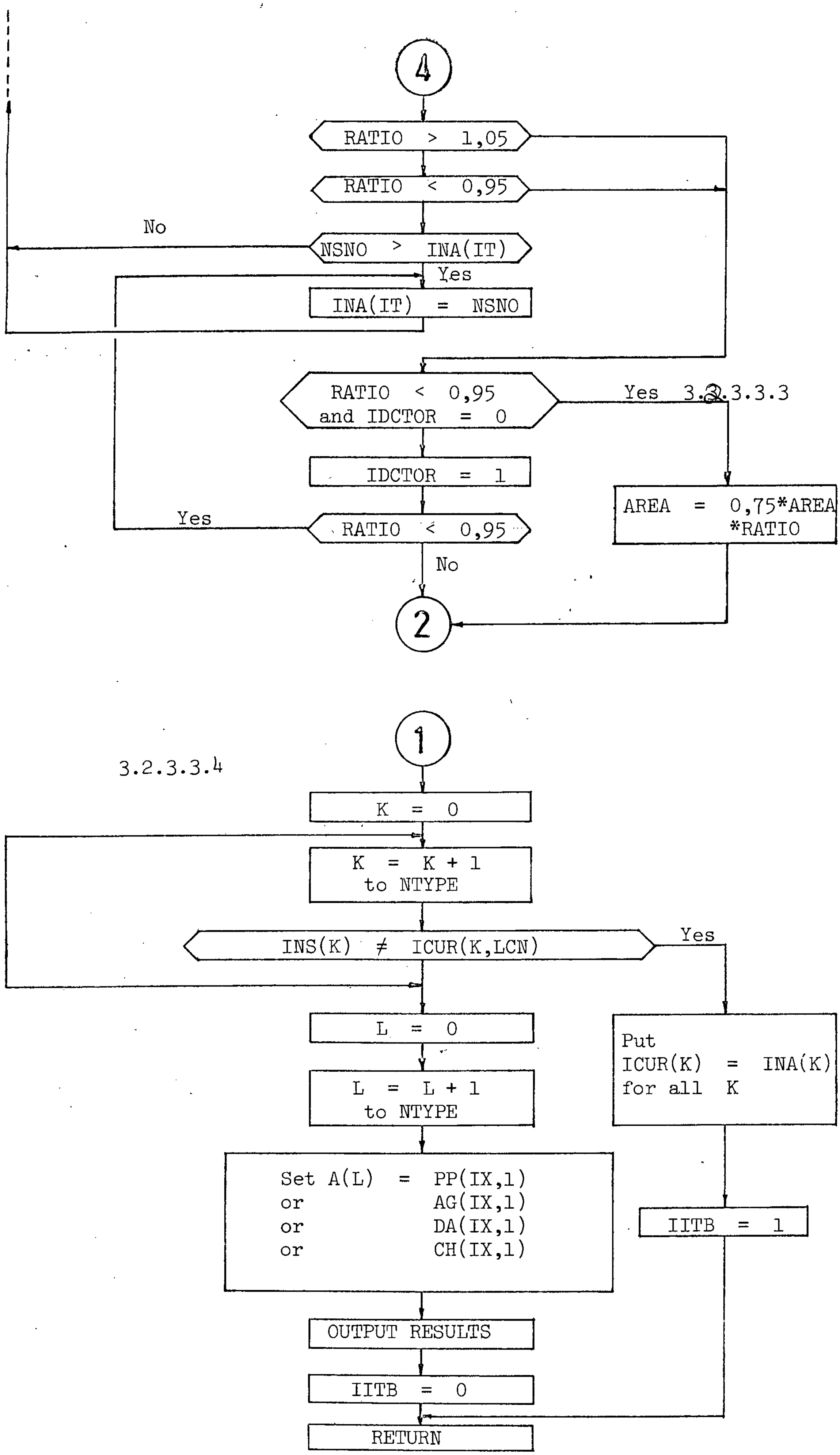
The subroutine DESIGN is used to check the design of the structural members and to re-proportion member sizes if required.

The design is consistent with the recommendations of BS 449 Part 2 (1969) and follows the steps discussed in equations (3.3) and (3.5). A detailed flow chart of this subroutine is shown in Figure 3.2.

Subroutine DESIGN - Figure 3.2 - Nomenclature

I	=	member number
NEL	=	maximum number of members
IT	=	member group number
NSID	=	member type identification number
	=	1 for pipes = array PP
	=	2 for angles = array AG
	=	3 for double angles = array DA
	=	4 for channels = array CH
IX	=	section number in list of sections
NSNO	=	new section size number
LSNA	=	last section size number
IDCTOR	=	satisfactory section size indicator
P(I)	=	member force
HIGLR	=	maximum allowable L/R ratio
COMLR	=	maximum allowable L/R ratio for struts
PLR	=	permissible L/R ratio
L	=	~~Lenght~~ length of member
ACTSTR	=	actual stress value
R	=	radius of gyration
DOR	=	L/R ratio
YS	=	yield stress
RATIO	=	ratio between actual and permissible stress
INA(IT)	=	array for storage of required section size for group IT
AREA	=	area required to produce a RATIO of 1.0
NTYPE	=	number of groups of members
K,L	=	counter for number of groups
ICUR(K)	=	current size of members for group K,L
IITB	=	termination indicator
	=	1, another iteration is required
	=	0, termination or start of next load case


I = 0
Next Member
I = I + 1
I = NEL
Yes
1
SET
IT = ITYPE(I)
NSID = IDSEC(IT)
IX = ICUR(IT,LCN)
NSNO = IX
LSNS = 0
IDCTOR = 0
Yes
P(I) = 0,0
No
P(I) > 0,0
Yes
No
PLR = COMLR
PLR = HIGLR
STRESS < 8000
3.2.3.3.1
Yes
PLR = COMLR + (STRESS = 8000)*(HIGLR - COMLR) / 8000,0
Calculate Length of Member = L
No
NCYCLE.EQ 1
Yes
2
RADG = PP(IX,2)
AG(IX,2)
DA(IX,2)
CH(IX,2)
CALL SEARCH
No
LSNS = NSNO
NSNO.EQ.LSNS
3
3.2.3.3.2
Calculate ACTSTR
DOR
Yes
DOR > PLR
2
P(I) > 0,0
Calculate PSTR
from DOR and YS
PSTR = 155000,0
RATIO = ACTSTR/PSTR
4



Numbers eg, 3.2.3.3.4 refer to sections in the text

Figure 3.2 Subroutine DESIGN

3.2.3.3.1 Permissible Slenderness Ratio (L/R)

Three values of permissible slenderness ratios can be specified in the data required for the program.

(i) HIGLR:

This value of the L/R ratio is applicable to members normally acting as ties but are subject to reversal of stress from the action of wind. The ratio must then not exceed 350 (BS 449 clause 44a). This L/R ratio has been chosen as an overall maximum value for any member.

(ii) COMLR:

The L/R ratio of struts subjected to loads other than wind is limited to 180; but for members subjected to wind loads, the value is 250 (clause 33).

An incompatibility results at low stress values between these two values i.e. HIGLR and COMLR. The stresses calculated from successive iterations can change sign depending on the change in areas. Therefore, the design L/R ratio can change dramatically between iterations, resulting in an unstable system which alternates between HIGLR and COMLR. Consequently, a transition between the two values has been incorporated as shown in Figure 3.3.

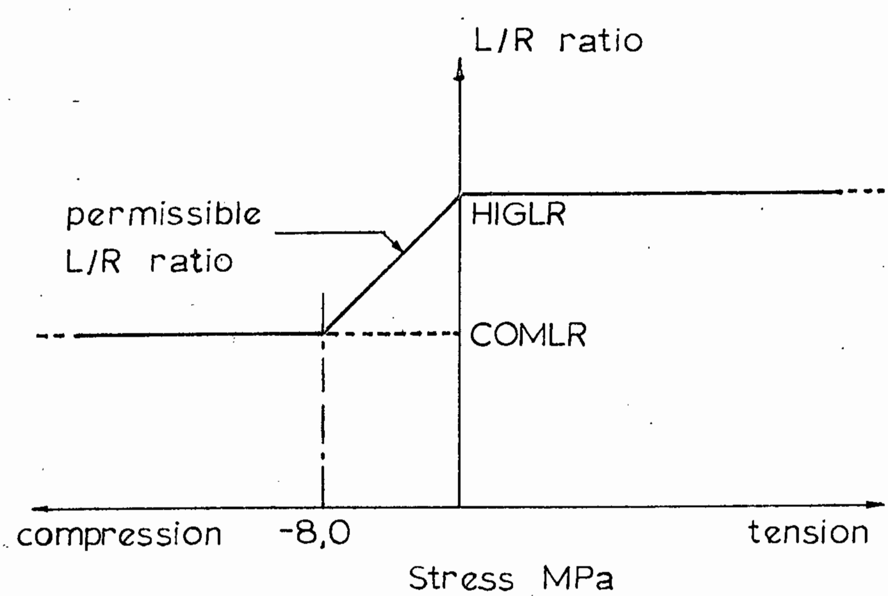


Figure 3.3 Permissible Range of Slenderness Ratios (L/R)

It is therefore assumed that the effective L/R ratio is proportionately increased at low values of compressive stress i.e. below 8 MPa.

3.2.3.3.2 Permissible Compressive Stress Values

The calculation of the permissible compressive stress of a member is dependent on the L/R ratio and the permissible elastic stress. It follows the interaction curve which is given in (BS 449, 1969 - Appendix B)

$$K_2\sigma_p = \frac{Y_s + (\eta + 1)\ C_o}{2} \sqrt{\left[\left(\frac{Y_s + (\eta + 1)\ C_o}{2}\right)^2 - Y_s C_o\right]} \qquad (3.6)$$

where. σ_p = permissible stress

K_2 = load factor of coefficient, taken as 1,7 for the purpose of the standard

Y_s = minimum yield stress in MPa

C_o = Euler critical stress

$= \dfrac{\pi^2 E}{(\ell/r)^2}$ MPa

$\eta = 0{,}3\ (\ell/100r)^2$

ℓ/r = slenderness ratio

= effective length to radius of gyration ratio

3.2.3.3.3 Choice of Acceptable Sections

The calculation of the value C_s (equation 3.3) determines the structural efficiency of a member. Acceptable limits of 1,05 and 0,95 have been specified. Suitable member areas are checked against the size required by members of the same group (cf position 3 in Figure 3.3). and the adjustment is made until the best value is selected. C_s ratios which fall outside the limits, require special attention. Ratios above

1,05 represent unacceptably large stresses. Hence, a larger section must be found by the subroutine SEARCH. An acceptable (i.e. adequate) design is found when the value of IDCTOR is 1.

Overdesign is indicated by a smaller ratio value than 0,95. A smaller area is calculated from the equation

$$\text{target area} = 0{,}75 \times C_s \times \text{current area} \qquad (3.7)$$

and the section size corresponding to the 'target area' is found by subroutine SEARCH. The process will probably produce an inadequate design (i.e. C_s above 1,05) but this can now be dealt with as a matter of course.

Further study of the relevant section of the flow chart will explain any contingencies arising.

3.2.3.3.4 Design Acceptability

A final design is considered to be acceptable when the section sizes chosen from consecutive iterations are identical.

For an acceptable design, control is transferred to a subroutine which prints out the salient results. The program is then free to continue with the next load case if applicable.

3.2.3.3.5 Search Subroutine

The list of sections provided by the designer, serves as a reservoir of possible designs. The design routine predicts a cross sectional area in numerical terms i.e. a target value. A systematic search is then carried out through the list to choose an actual section which is most suitable. Care is taken to ensure that both the area and the radius of gyration of the section chosen are acceptable. Figure 3.4 shows the details of the subroutine SEARCH.

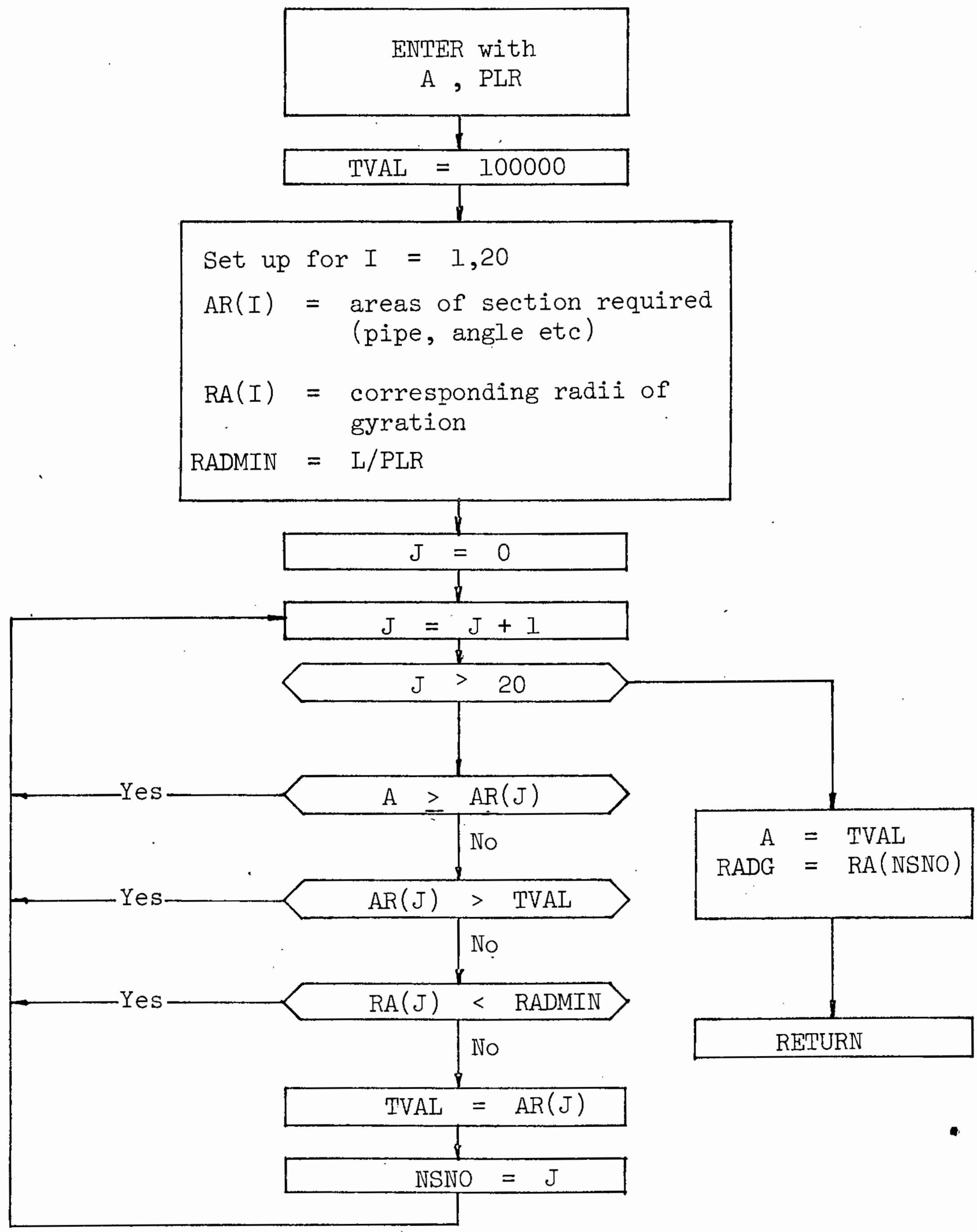


Figure 3.4 Subroutine SEARCH

Three questions must be answered in the negative for a section to be acceptable:

(i) Is the target area greater than the listed area ?

(ii) Is the listed area greater than a previously determined acceptable area (TVAL) ?

(iii) Is the listed radius of gyration less than the minimum allowable which is calculated from the effective length (L) and the permissible slenderness ratio (PLR)

A positive answer to any of the above questions renders the listed section unacceptable and a further search is required.

Subroutine SEARCH - Figure 3.4 - Nomenclature

A	=	area required to produce an actual/permissible stress ratio of 1,0
PLR	=	permissible L/R ratio
TVAL	=	temporary value of area
I,J	=	counter within list of sections
AR(I)	=	area of position I in list of sections
RA(I)	=	radius of gyration of position I in list of sections
RADMIN	=	L/PLR = minimum radius of gyration allowable
NSNO	=	new section size number

3.2.3.4 Load Cases

Various load cases can be entered into the program. A separate design is prepared for each load case. Although the load values can simply be arithmetically summed to produce a combination load case, the design variables (areas, radii of gyration) cannot, since the analysis of the structure requires different values in the stiffness matrix for each different load case. Consequently, the program has been written to combine relevant load cases and to calculate separate designs without the need for excessive data manipulation. In addition, the inclusion of self weight and/or temperature loading is available for each load case if required.

3.3 DYNAMIC PROGRAMMING AND DIRECT ITERATION FOR THE DESIGN OF TOWER STRUCTURES: PROGRAMS: DYNGEO, DYNPRE

3.3.1 Introduction

Two computer programs have been written to design tower structures. In both, the configuration of the structure and the selection of the member sizes are varied until an optimum design is determined. An optimum design is a configuration and member design which produces the minimum weight for a given set of loads.

The 'best' configuration is chosen by the application of the Dynamic Programming Successive Approximations Technique (Chapter 2). A direct iteration method is used to design the member sizes (section 3.2).

The structure is analysed and designed as separate substructures in order to reduce the computational time required to reach an optimum solution. The possible dimensions of the interface between substructures are specified by the designer. The method then employs a series of systematic decision sequences which determine the weight of the structural members in the various configurations of each substructure.

The program DYNGEO analyses and designs all the configurations within the substructures. To decrease the computer time spent on analysis, DYNPRE has been written. This program considers the design of the extreme configurations of each substructure by calculating member forces for four controlling configurations. Predictions of the forces and hence the design of other configurations of the substructure can then be found from these calculations. This method substantially reduces the computer time requirements over those of DYNGEO. A full description of this process is given in section 3.3.3.

The practical aspect of choosing an actual section size from a list of sections, is included in the program together with the other characteristics of the program DYSPAN. As in DYSPAN, a combination of load cases can be designed for, but self weight and temperature loading are excluded.

3.3.2 DYNGEO; Program Description

DYNGEO is structured in modular form, i.e. it consists of a main program and a series of independent subroutines. The main program correlates the calculations while the subroutine performs the specific tasks required in the DPSA and Direct Iteration procedures.

A macro-flow chart of the program (Figure 3.5) shows the steps required for the optimum design of space truss tower structures by this method.

Major features of the program include:

(i) A complete record of steps in the DPSA calculations, if required by the designer

(ii) The design of three different tower types

Plane truss

Rectangular plan shape

Triangular plan shape

(iii) Designer controlled configurations at each substructure interface. A maximum of five possible dimensions in each coordinate direction can be specified by the designer.

(iv) Any node in the structure can be positioned at a particular set of coordinate values which can be kept constant throughout the

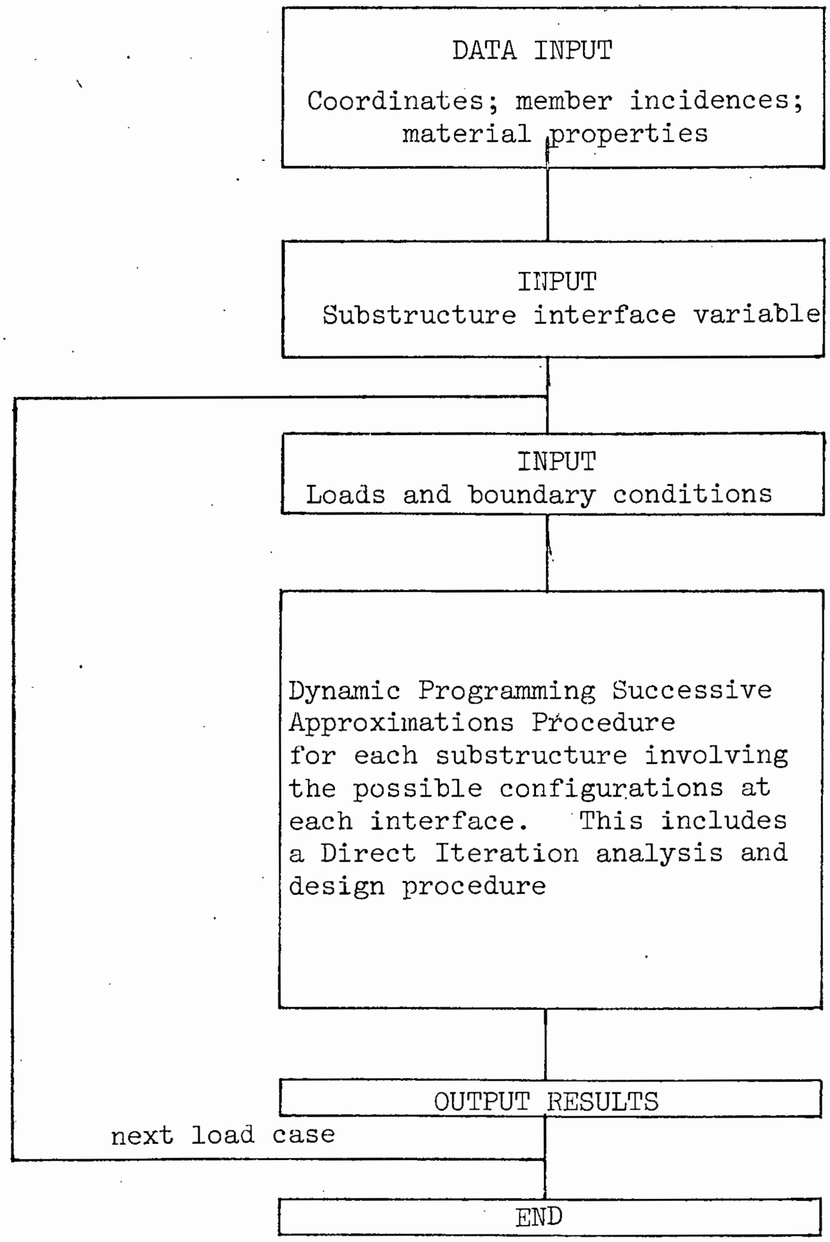


Figure 3.5 Macro-flow Chart for Programs DYNGEO; DYNPRE

the calculations. Since the shape of the structure changes for various configurations, this is particularly useful for the accurate positioning of specific nodal loading.

(v) The calculation of equivalent loads and support constraints for each substructure.

(vi) The calculation of the cost (i.e. weight) of the structure from the structural design.

(vii) Dynamic core storage for efficient computer use irrespective of the dimensions of the problem.

(viii) Data input is in 'free format' throughout.

(ix) A computer plot of the completed structure can be produced if required by the designer.

(x) Data is automatically generated for a possible structural design by the program DYSPAN.

3.3.2.1 Coordinate System

The coordinate system shown in Figure 3.6 has been used in the formulation of the program. The origin is specified as the centroid of the figure formed by the foundation nodes for three dimensional structures i.e. the centroid of a rectangular or triangular figure. Planar structures are defined in the X-Z plane (i.e. the y-coordinates are zero) and the origin is at the centre of the line joining the two foundation nodes. The elevation of both 2 and 3 dimensional structures is in the direction of increasing z-coordinate.

3.3.2.2 Substructure Interface Dimensions

The substructure geometry of a tower can be defined by the size, shape and elevation of its upper and lower interfaces. The designer has complete control over the selection of the dimensions

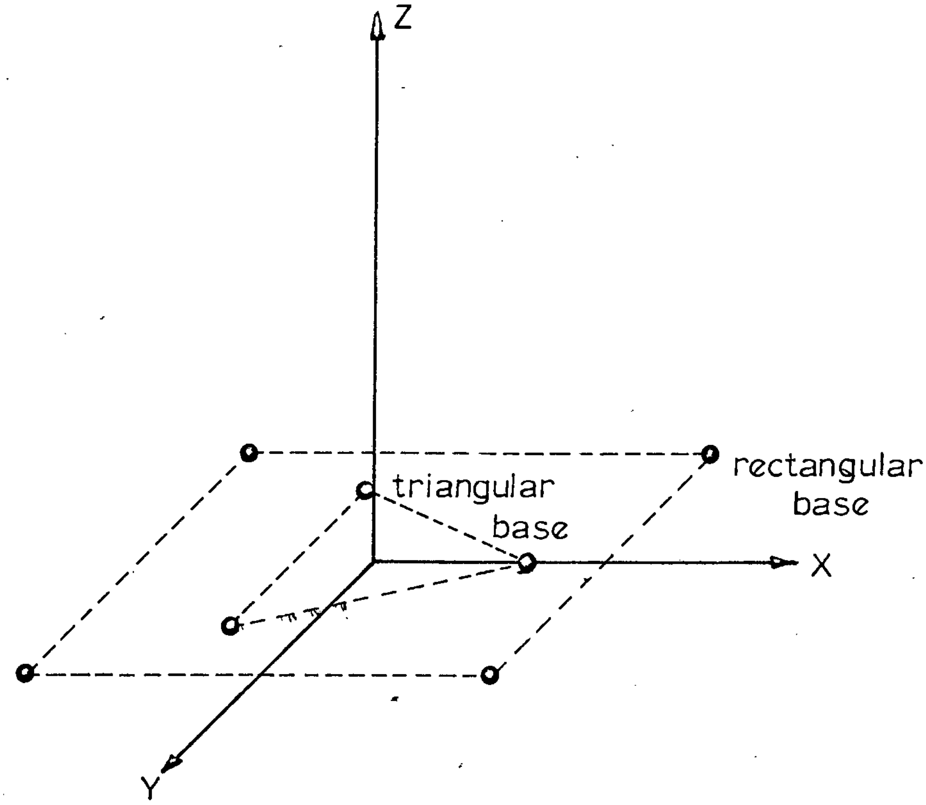


Figure 3.6 Coordinate Axis System

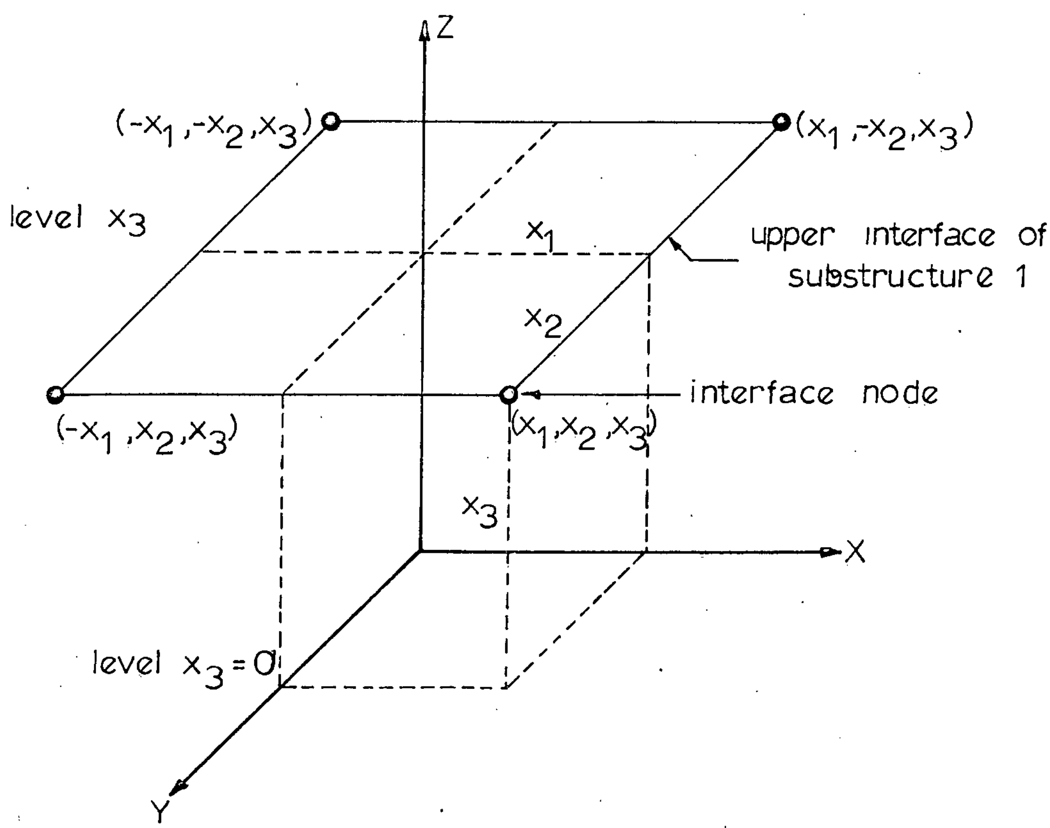


Figure 3.7 Definitions of State Variables

governing the interface. These dimensions or state variables x_1, x_2, x_3 are defined as being the coordinates of a primary interface node positioned in the 1st octant, relative to the X,Y,Z axes respectively (Figure 3.7).

A primary interface node defines the interface dimensions by producing mirror images in the other three +Z octants (Figure 3.7). For example, four interface nodes are (x_1,x_2,x_3); $(x_1,-x_2,x_3)$; $(-x_1,-x_2,x_3)$; $(-x_1,x_2,x_3)$.

The triangular plan shape structures are defined by state variables x_1 and x_2 where x_1 represents the radius of the circle enclosing the interface nodes and x_2 represents the elevation of the interface.

The geometric configuration of the members is uniquely defined by the specification of upper and lower interface dimensions (section 3.3.2.4).

The designer can choose five values for each state variable at every interface. The permutations of configurations produced from these values will then represent the total number thought to be practically feasible by the designer. The DPSA method contained in the program must then choose the best configuration to produce a minimum weight.

3.3.2.3 Initial Solution: Geometric Configuration

The DPSA technique requires an initial solution of the problem before calculations can commence. The designer provides this initial solution by proportioning the structure into some feasible shape. The coordinates and member incidences which define this shape are regarded as the initial solution.

3.3.2.4 Calculation of Coordinates for Changing Geometry

Consider a single substructure of a typical tower. The geometric configuration of the structure is defined by the dimensions of the interfaces. A change in the dimensions of the interfaces is required by the DPSA technique. Consider a simple plane truss substructure (Figure 3.8) whose upper and lower interfaces are initially defined as

$$x_1(1) = 5,0 \quad \text{and} \quad x_1(2) = 3,0$$

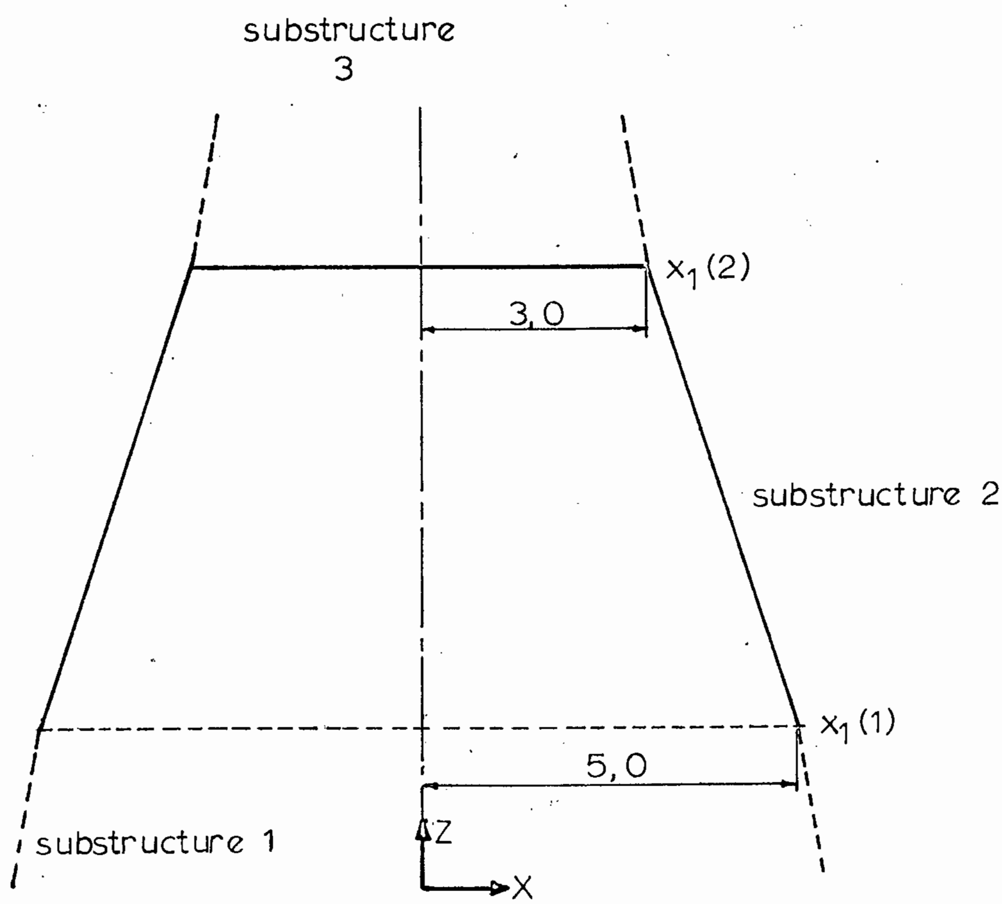


Figure 3.8 Substructure 2 of Typical Tower

Let the possible positions of x_1 at level 2 be 2,0; 3,0; 4,0 and at level 1 be 4,0; 5,0; 6,0. The DPSA method requires the permutation of configurations between these values to be investigated as shown in Figure 3.9.

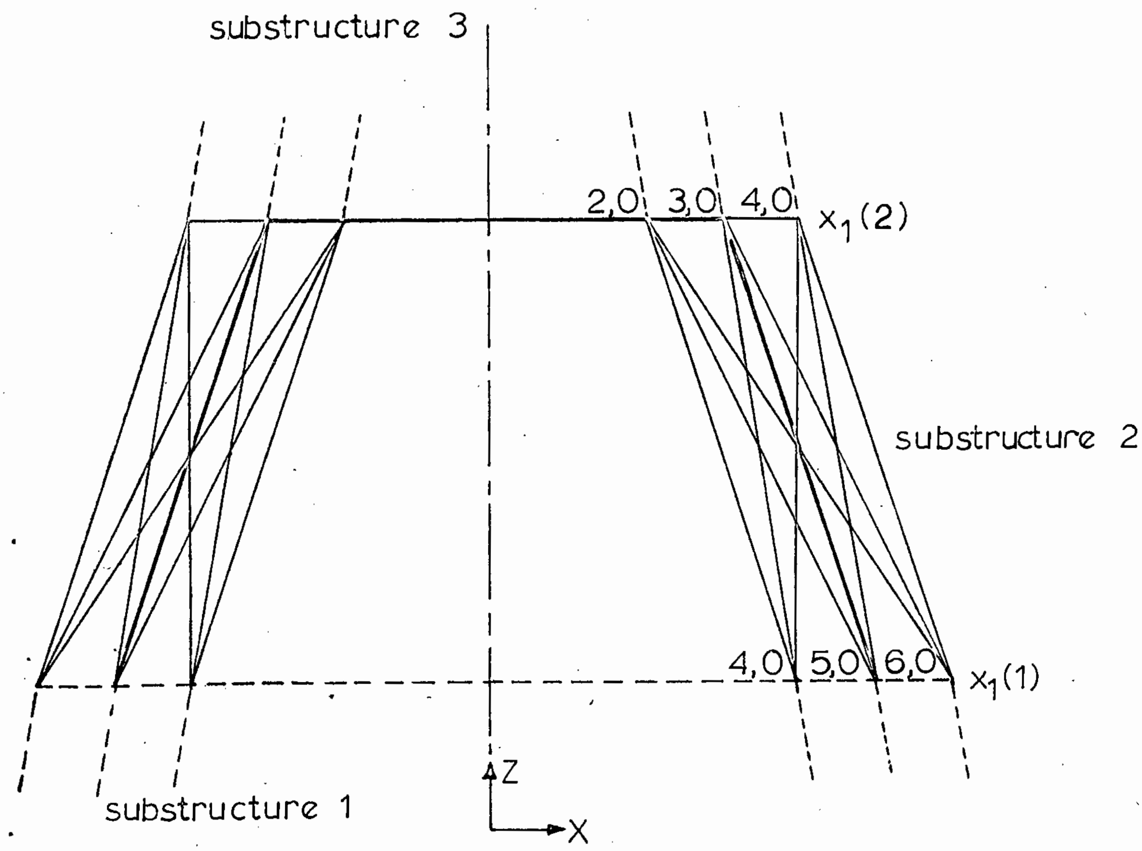


Figure 3.9 Possible Configurations of Substructure 2

The coordinates at levels 1 and 2 can be immediately deduced, but the coordinates of a general node A are unknown. The calculation of the new coordinates of A requires comparison of the dimensions of the interface of the original substructure (Figure 3.8) with those of the proposed substructure. From their relationship, the position of node A can be calculated by simple proportion.

The complete formulae required for this process are given in Appendix H.

3.3.2.5 Equivalent Loads and Support Constraints

The external loading applied to a substructure is assumed to act through equivalent loads on the interface between substructures. The equivalent loads are calculated from statics for the three types of tower configurations investigated. Subroutines EQUILD and TRILD calculate these loads for plane truss and rectangular plan tower, and for triangular plan towers respectively. Details of these equivalent loads are given in Appendix A.

Each substructure is assumed to be fully constrained at its lower interface nodes by equivalent support constraints. Only the stresses in the interface members are significantly effected by this assumption. These support constraints are controlled by subroutine EQUIBC.

3.3.2.6 <u>The DPSA Control Subroutine SUCOPT</u>

Control of the calculations performed in the DPSA is monitored by the subroutine SUCOPT. The subroutine controls the cycling of the state variables (as discussed in section 2.5 of Chapter 2) and the successive sets of configurations of each substructure in turn. A detailed flow chart is given in Figure 3.10.

This subroutine uses the previously discussed ideas to calculate a new set of coordinates for each alternate configuration of the substructure, (subroutines RECTCD, TRICD) equivalent loads and support constraints. The actual substructure design which is dependent on the above data, is controlled by subroutine CELDSN. This is a curtailed form of the program DYSPAN and includes both the analysis and design as discussed in section 3.2. The weight of each substructure configuration is calculated from the structural design by the subroutine COST. The accumulated cost of each state variable position at every level of the tower is stored until the best weight value and hence the configuration can be found.

Successive approximations decrease the overall weight until the results from consecutive Dynamic Programming calculations are identical This is checked in the subroutine CONCCK.

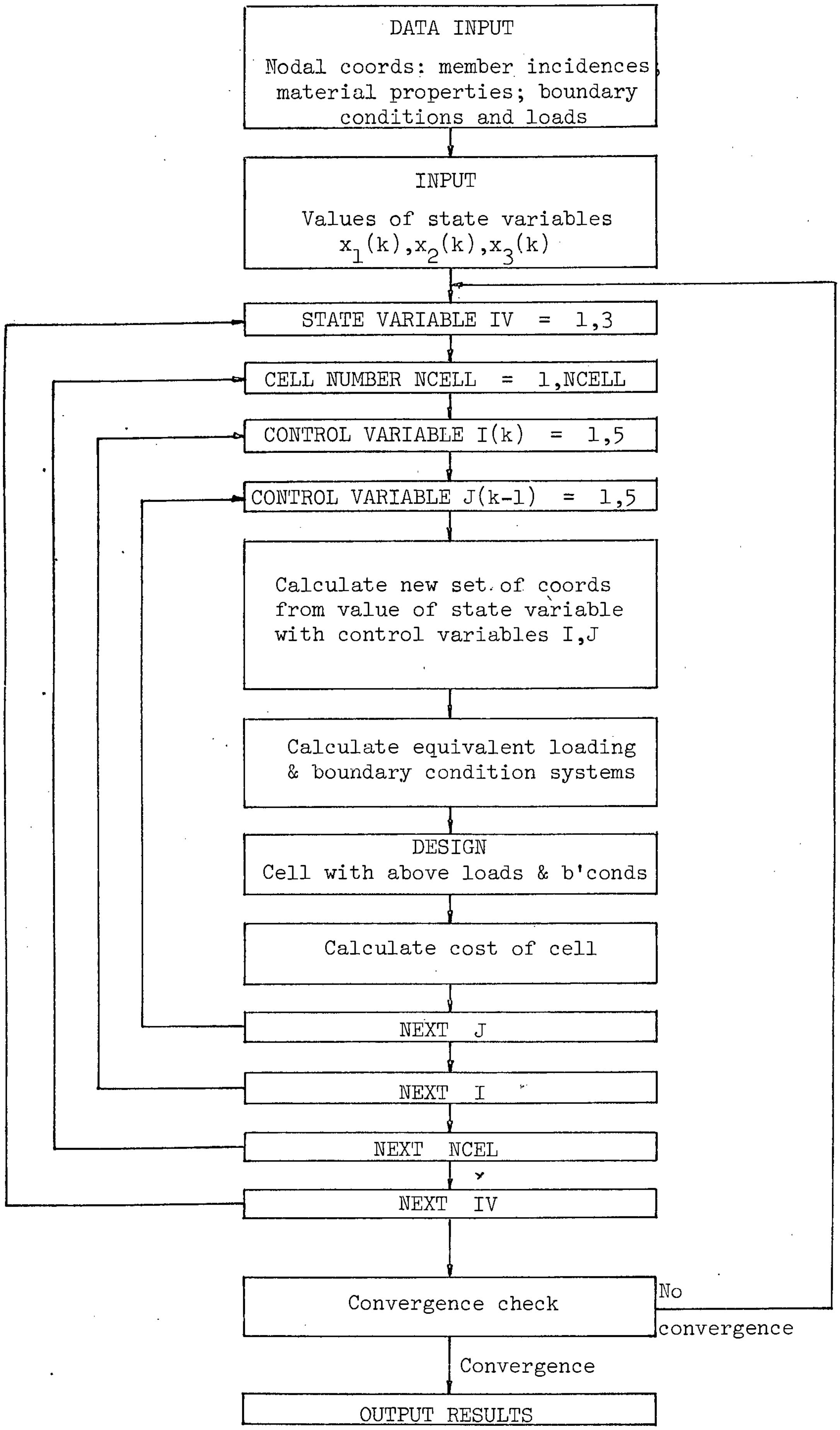


Figure 3.10 Flow-chart of Subroutine SUCOPT

Finally the pertinent results are printed by the subroutine OUTPUT. Control is then passed back to the main program, where the next load case can be initiated or the program execution terminated.

3.3.3 DYNPRE: Program Description

The previously discussed DYNGEO requires an analysis and design to be carried out for each combination of upper and lower interface variables for every substructure. To improve the computational time required for these calculations, the program DYNPRE has been written. Four extreme configurations are chosen. The forces obtained from each form a basis for four controlling designs. The structural design of any other configuration of members can then be found by interpolating between the four controlling designs.

Consider the possible configurations within a substructure. Let the total number of values of the upper and lower interface variables be I and J respectively (Figure 3.10). If the maximum number of values (i.e. I = 5; J = 5) is specified by the designer, then the total number of analyses for each substructure is 25.

In an attempt to decrease the number of analyses, DYNPRE has been written to design any general combination of I and J by prediction from four controlling designs

(a)	I = 1	J = 1
(b)	I = 5	J = 5
(c)	I = 1	J = 5
(d)	I = 5	J = 1

By a simple process of linear interpolation, the design of any combination of I and J can be found. The calibration (subroutine CALBRN) is carried out by considering the four controlling configurations (Figure 3.11). The axial forces in the members are calculated and

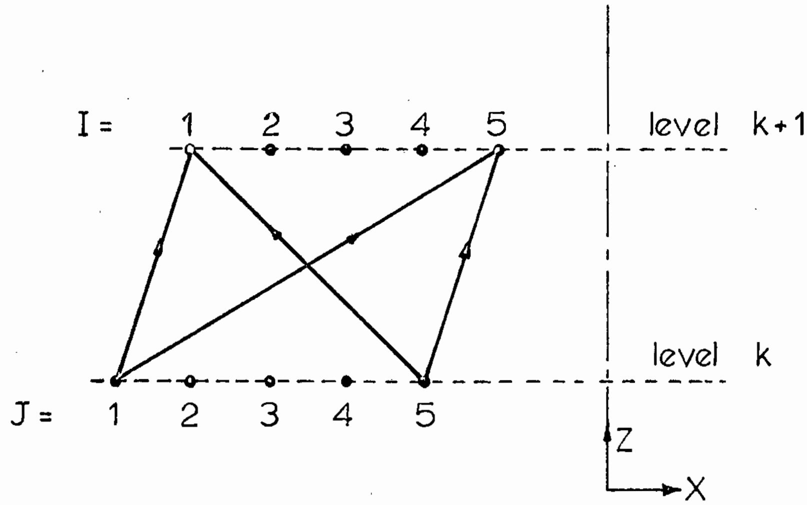


Figure 3.11 Four Controlling Configurations

stored for each configuration. The prediction of the forces and therefore the design of any combination of I and J can be found by linear interpolation between the forces of the control configurations. For example, the design of (I,J) = (1,3) can be found by interpolating between (1,1) amd (1,5). The design (I,J) = (2,3) is found by a double interpolation process. Firstly, (1,1) and (1,5) are used to find (1,3); and (5,1) and (5,5) used to find (5,3). (2.3) can then be found by interpolation between (1,3) and (5,3).

This technique has proved very successful with little practical difference being found between the results produced from the two programs DYNGEO and DYNPRE.

3.4 COMPUTER PLOTTER PROGRAM DRAW

The designer often requires a graphic display of his structure in order to consider the aesthetic and practical aspects involved and to check his data input. The designer must be able to recognize radical design configurations which could be generated by the programs DYNGEO and DYNPRE. Consequently, a program DRAW has been written for this purpose.

The program is entirely automatic. The required data (nodal coordinates, member incidences) is automatically read from a data-file set up for this purpose by the main programs. A drawing can either be plotted on the Graphics Terminal or alternatively on the drum plotter. A convenient method can be employed whereby an 'instantaneous' plot can be generated on the graphics terminal, and if satisfactory can be sent directly to the drum plotter.

Each drawing produced is enclosed in standard A4 format. A title block contains the information of drawing type and scale. As the structures are three dimensional pin-jointed structures with many members, an isometric view would be too confusing. Consequently, the structure is plotted in elevation and plan. A plan of every interface level is plotted to ensure completeness and accuracy.

3.5 SUMMARY

The program DYSPAN has been written to determine the optimum member sizes of general space trusses for any specific geometric configuration. This program uses a direct iteration procedure to choose member sizes from a list of sections provided. These members are designed so that the ratio between the actual and permissible stresses are as close to unity as possible.

The program DYNGEO analyses and designs structures by a process of substructuring. In each substructure, successively varying configurations are designed. The Dynamic Programming Successive Approximations method chooses an optimum configuration for the structure by a series of sequential decisions based upon a least weight criterion. DYNGEO also uses direct iteration to design the members.

The program DYNPRE is very similar to DYNGEO except that interpolated forces are used for the design of the members. Four controlling configurations are used in each substructure in the interpolation process. The computation time for this program is significantly reduced in this way. The ratio of computation times between DYNGEO and DYNPRE is approximately 25:4.

The plotting program DRAW has been written to automatically plot the elevations and substructure interface plans if required by the designer.

CHAPTER 4

NUMERICAL EXAMPLES

4.1 INTRODUCTION

Six examples of tower structures are given which have been carefully selected to illustrate various features of the three programs, DYSPAN, DYNGEO and DYNPRE discussed in this thesis. Although only tower examples are shown, DYSPAN can be used for any type of space truss structure, while DYNGEO and DYNPRE can also be used for the design of domes, pyramids and indeed for any cantilever type structure.

The six examples which have been 'automatically' designed with these programs are:

i) 3-substructure plane-truss tower

ii) 2-substructure triangular plan tower

iii) 2-substructure rectangular plan tower

iv) 3 legged transmission tower (from the literature)

v) Rectangular transmission tower (from the literature)

vi) Practical transmission tower

4.2 EXAMPLE 1 - PLANE-TRUSS TOWER (3 substructures)

This simple example has been chosen to illustrate the basic concepts of the method. The scope of the program is shown by the inclusion of all types of section classes (pipe, channel, angle sections) with no regard being taken of practical connection details. If desired, however, these classes can be changed at will to provide a more practical design.

Initial Configuration: Height: 6 m

Breadth at foundation level: 2 m

Breadth at interface level 1: 1,5 m

Breadth at interface level 2: 1,0 m

Breadth at upper level: 0,5m

The initial configuration is shown in Figure 4.1

Loading: Two loads:

at node 7: - 15 kN in z-direction

at node 8: - 10 kN in x-direction

Possible dimensions (state variables) at the 4 interface levels shown in Figure 4.1.

i) x-direction

	1	2	3	4	5
Base	1,6	1,8	2,0	2,2	2,4
Level 1	1,1	1,3	1,5	1,7	1,9
Level 2	0,8	0,9	1,0	1,1	1,2
Level 3	0,5	-	-	-	-

ii) z-direction

	1	2	3	4	5
Base	0,0	-	-	-	-
Level 1	1,8	1,9	2,0	2,1	2,2
Level 2	3,8	3,9	4,0	4,1	4,2
Level 3	6,0	-	-	-	-

Section classes:

i) Up-right members - channel sections

ii) Diagonal members - pipe sections

iii) Horizontal members - angle sections

Section lists: Appendix L

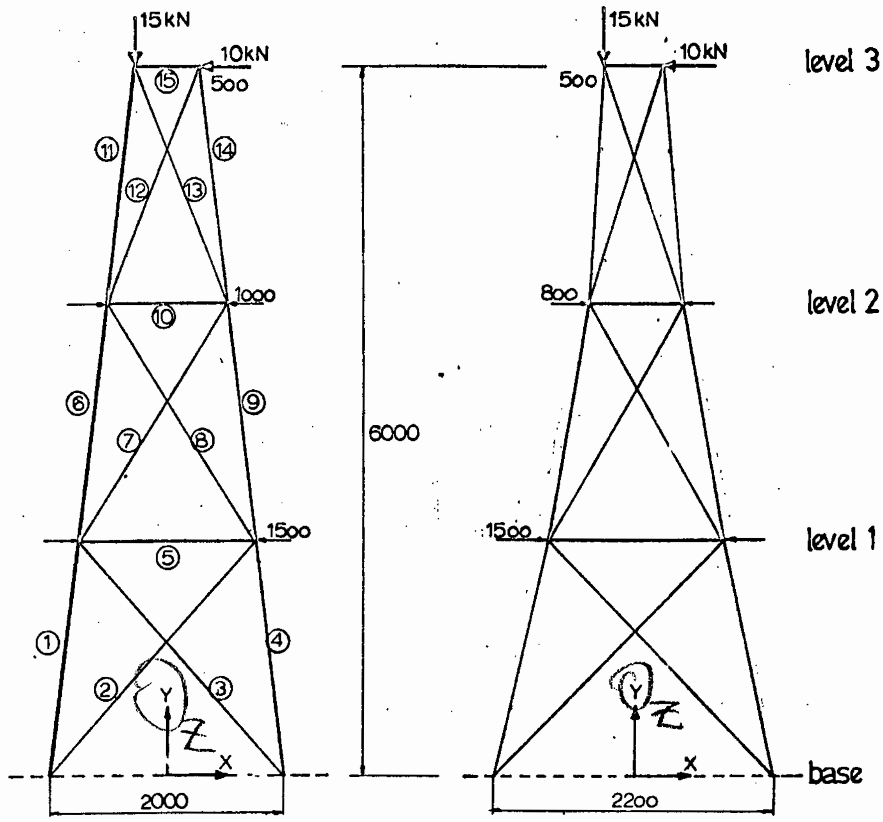


Figure 4.1 Original Tower Shape

Figure 4.2 Optimum Tower Shape

Results:

Member forces: The following diagram (Figure 4.3) shows a comparison between the forces obtained from the 3 programs DYSPAN, DYNGEO and DYNPRE for members shown in Figure 4.1 The difference in the forces exhibited in the program DYNPRE is caused by a slight difference in the overall optimum configuration of the structure. Programs DYNGEO and DYNPRE, however, produce structure weights of 1,39 kN and 1,461 kN, a difference of 4,8%

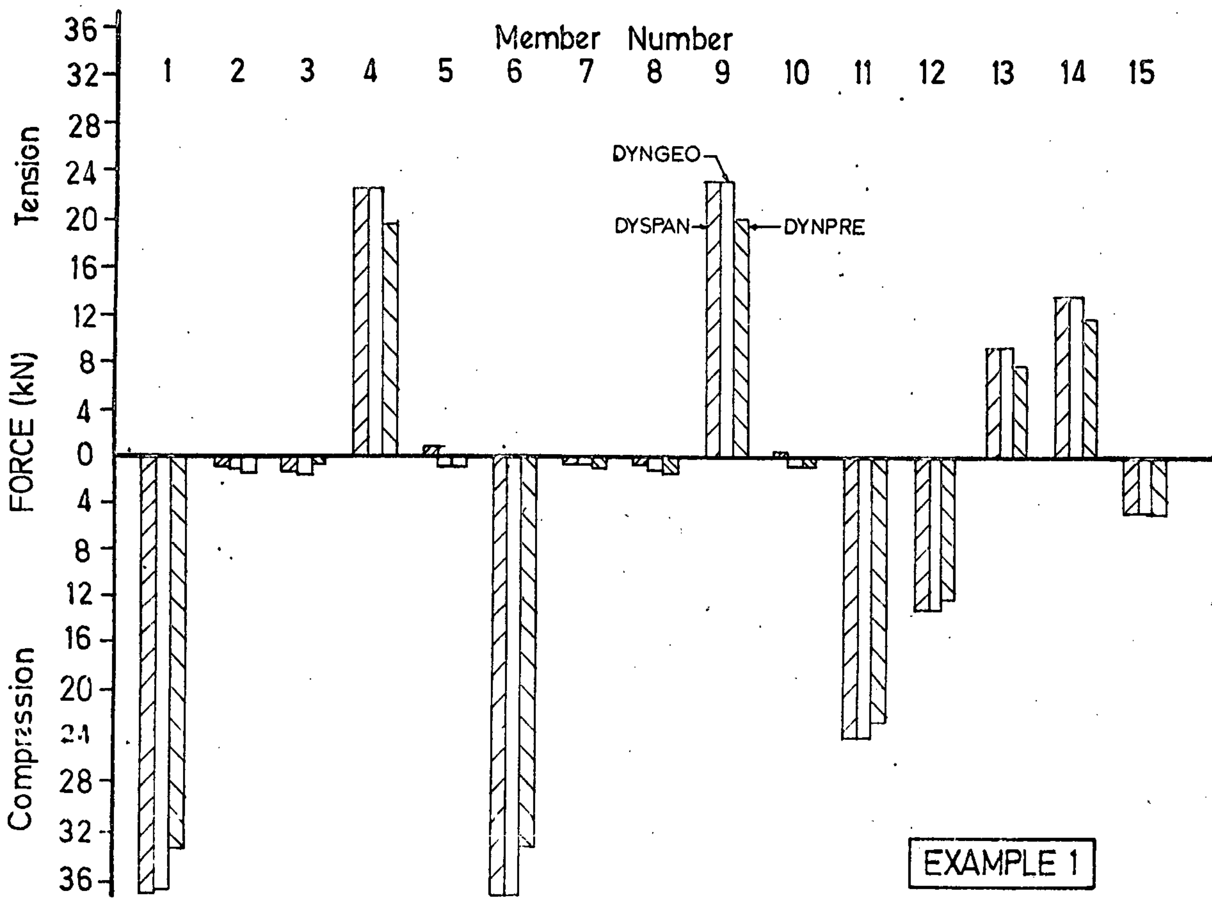


Figure 4.3 Member Forces

Structural design: The diagram (Figure 4.4) shows the structural design of the tower. Channels were used for member group number 1, pipes for member group number 2 and angle for member group number 3 in each substructure without regard to practical connection details. The member size numbers correspond to those in Appendix L. No differences are registered for the main members (group 1). However, small differences of design exist in the bracing members which are caused by the permissible values of stress for the particular slenderness ratio. These members are not critical since the maximum stress in the lower level bracing members is a negligible 5,8 MPa.

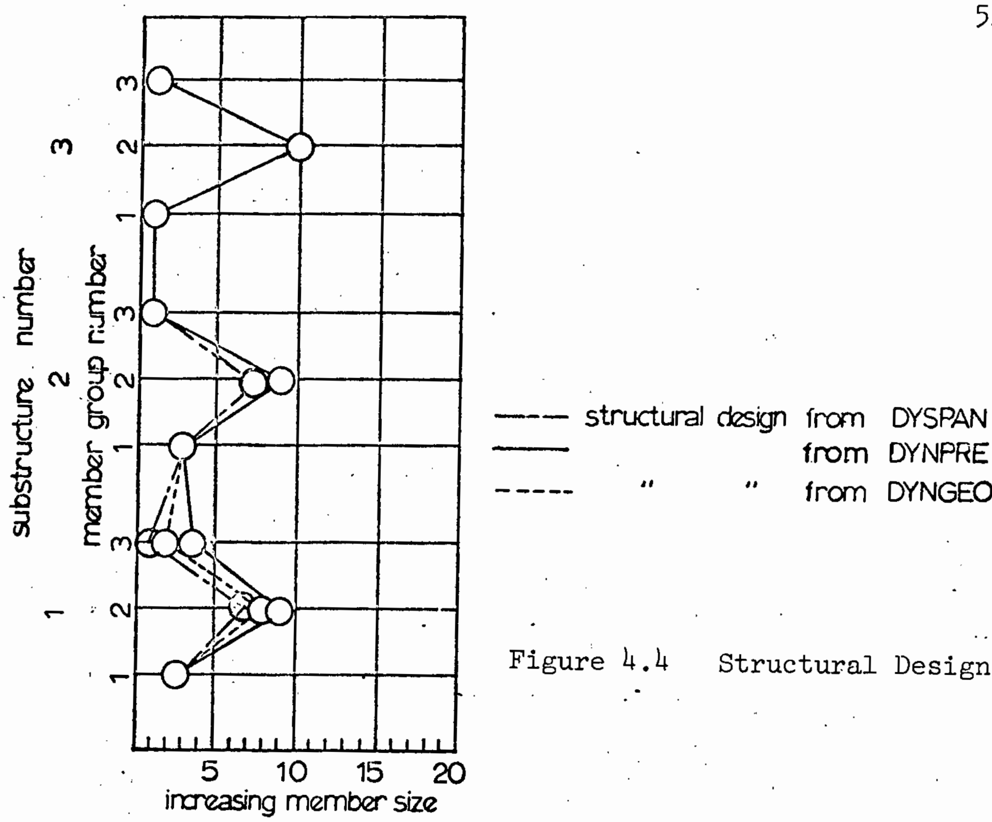


Figure 4.4 Structural Design

	DYSPAN:	2,24 sec
Computer times:	DYNGEO:	48,579 sec
	DYNPRE:	19,46 sec

As expected the computational times for DYSPAN is far less than for the other two programs, since DYSPAN does not alter the predetermined geometrical configuration of the structure.

DYNGEO reanalyses the structure for each variation of geometrical configuration while DYNPRE uses interpolated member forces and hence less computation is required.

4.3 EXAMPLE 2 - TRIANGULAR PLAN TOWER (2 substructures)

In the example the substructure height as well as the interface width is varied at each level. A relatively large range of possible dimensions have been chosen at each interface. This is only a small illustrative example.

Initial Configuration: Height: 4 m

Radius of foundation level: 3 m

Radius of level 1: 2 m

Radius of level 2: 1 m

Total number of nodes: 15

Total number of members: 42

Loading: Three loads:

at node 13: 10 kN in z-direction

at node 14: 15 kN in y-direction

at node 15: - 12 kN in x-direction

Possible dimensions (state variables) at the 3 levels

i) Radii

	1	2	3	4	5
Base	0,4	0,45	0,5	0,55	0,6
Level 1	0,3	0,35	0,4	0,45	0,5
Level 2	0,2	0,25	0,3	0,35	0,4

ii) Elevation, z-direction

	1	2	3	4	5
Base	0,0	-	-	-	-
Level 1	1,8	1,9	2,0	2,1	2,2
Level 2	3,8	3,9	4,0	4,1	4,2

Section classes:

All members to be pipe sections

Section lists: Appendix L

Results:

Two differing optimum configurations were produced by the programs DYNPRE and DYNGEO. The radius of the interface levels are:

	DYNPRE	DYNGEO
Base	0,6 m	0,5 m
Level 1	0,5 m	0,35 m
Level 2	0,2 m	0,35 m

Although this difference exists, the optimum weights from DYNPRE and DYNGEO were 1,091 kN and 1,115 kN respectively. This is a 2.15% difference. The design from DYNPRE is an approximation to that of DYNGEO, but since the prediction of the forces differ the choosing of a configuration is also different. The comparison of dominant forces for the three programs are:

Substructure Number	Member Type	DYSPAN	DYNGEO	DYNPRE
1	Main legs	-68,17 48,63	-75,68 54,54	-55,13 39,37
	Interface	-10,7	0,63	- 1,6
	Bracing	-15,8 -13,5 -17,3 -16,8	13,3 -10,8 -11,9 -11,14	-13,8 -12,6 -14,6 -13,9
2	Main legs	20,8 -25,2	19,8 -24,6	18,3 -23,9
	Interface	- 8,154	- 8,02	- 8,36
	Bracing	-25,6 -24,06	-25,8 -23,6	-12,083 -22,3

The structural design is shown in Figure 4.6 The difference in design for member groups 2 of substructure 1 is due to the varying degrees of accuracy of the forces in the bracing members. For substructure 2, the structural design from DYNGEO is exactly similar to DYSPAN.

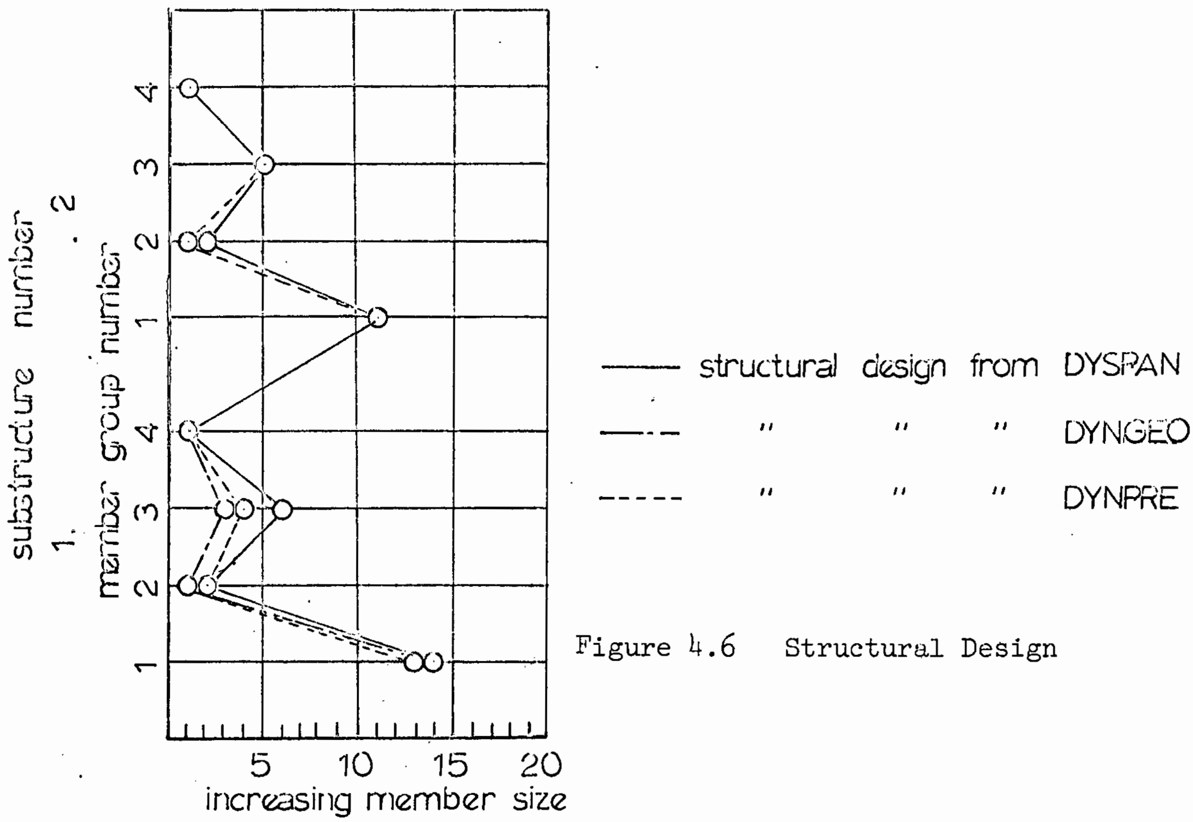


Figure 4.6 Structural Design

	DYSPAN:	8,286 sec
Computer times:	DYNGEO:	5 min 35,853 sec
	DYNPRE:	16,74 sec

Similar conclusions may be drawn to those given in example 1 as regards computer times.

Although large disparities occur in some member forces, each design is perfectly feasible. It is interesting to note that the programs DYNGEO and DYNPRE provide almost exactly the same optimum weight (2,15% difference).

4.4 EXAMPLE 3 - RECTANGULAR PLAN TOWER (2 substructures)

The reason for selecting this example is to determine how the program would select an optimum configuration if the tower were subjected to vertical loads only.

Initial Configuration:

Height: 4 m

Dimensions at foundation level: 2 x 2 m

Dimensions at level 1: 1 x 1 m

Dimensions at level 2: 1 x 1 m

Total number of nodes: 20

Total number of members: 60

Loading: Four loads

at node 17: -20 kN in z-direction

at node 18: -20 kN " "

at node 19: -20 kN " "

at node 20: -20 kN " "

Possible dimensions (state variables) at the 3 levels

i) x- and y-directions

	1	2	3	4	5
Base	0,6	1,0	1,4	1,8	2,2
Level 1	0,4	0,6	0,8	1,0	1,2
Level 2	0,4	0,6	0,8	1,0	1,2

ii) z-direction

	1	2	3	4	5
Base	0,0	-	-	-	-
Level 1	1,8	1,9	2,0	2,1	2,2
Level 2	3,8	3,9	4,0	4,1	4,2

Section classes:

All members to be pipe sections

Section lists: Appendix L

Results:

In this example a set of vertical loads is used to produce an optimum configuration. The example is designed to test the programs' ability to produce an effective structure. The results from both DYNGEO and DYNPRE show that the smallest dimensions are chosen at each interface and the height of the tower is reduced to the minimum allowed by the state variables. Only two member groups were used:

Member group 1 - pipe sections in substructure 1

Member group 2 - pipe sections in substructure 2

Substructure Number	Member Number	DYSPAN	DYNGEO	DYNPRE
1	1	-8,34	-8,198	-8,198
	5	3,6	0,923	0,923
	9	-6,78	-6,862	-6,862
	18	-6,30	-6,379	-6,379
	24	-6,78	-6,862	-6,862
2	31	-8,477	-8,242	-8,242
	35	2,711	2,766	2,766
	39	-6,368	-6,497	-6,497
	46	-6,368	-6,497	-6,497
	54	-6,368	-6,497	-6,497

The only considerable force difference is in member number 5 which is a horizontal member on the interface between substructures. The equivalent loading has been applied to its surrounding nodes and consequently the actual transfer of member forces between upper and lower substructures is not complete. This is a minor problem, however, since the actual stress in the member is 5 MPa.

The structural design chosen by the programs for the 2 member groups is (in mm)

Member Group	DYSPAN	DYNGEO	DYNPRE
1	42,9ØX2	42,9ØX2	42,9ØX2
2	42,9ØX2	42,9ØX2	42,9ØX2

	DYSPAN:	9,187 sec
Computer.times	DYNGEO:	8 min 43,295 sec
	DYNPRE:	2 min 59,964 sec

Computer times follow the trend set for examples (1) and (2). The main point of interest is that the tower is being reduced to the minimum cross sectional dimensions specified and the height of the tower is also reduced to the minimum specified dimension. This is a good indication that the program logic is correct.

4.5 EXAMPLE 4 - THREE LEGGED TRANSMISSION TOWER

The design of this tower (Figure 4.8) is compared with results given by Kuzmanovic et al [13]. The initial configuration is that specified by Kuzmanovic et al, while the loading is the metric (SI) equivalent of that given in the same paper.

Initial Configuration: Height: 14,63 m

Radius at foundation level: 3,167 m

Radius at level 1: 2,112 m

" " " 2: 1,76 m

" " " 3: 1,408 m

" " " 4: 1,056 m

Radius at level 5: 1,056 m

" " " 6: 1,056 m

Total number of nodes = 42

Total number of members = 132

The structure is shown in Figure 4.8.

Loading: Three load cases are specified

Load Case 1: Basic wind force in transverse direction

Position	Load, kN	Direction
A,B	26,7	-z
	8,9	-x
C	80,0	-z
	31,0	-x
D,E	15,6	-x
	40,0	-z

Load Case 2: 0,707 Basic wind force in transverse direction

0,707 Basic wind force in longitudinal direction

Position	Load, kN	Direction
A,B	16,0	-z
	5,4	-x
C	48,0	-z
	18,7	-x
D,E	9,4	-x
	11,6	+y
	24,0	-z

Load Case 3: No wind

Position	Load, kN	Direction
A,B	32,0	-z
C	160,0	-z
D,E	80,0	-z

Possible dimensions (state variables) at the 7 levels

i) Radii

	1	2	3	4	5
Base	3,4	3,3	3,167	3,0	2,8
Level 1	2,4	2,2	2,112	2,0	1,8
2	2,0	1,9	1,76	1,7	1,6
3	1,6	1,5	1,408	1,3	1,2
4	1,2	1,1	1,056	1,0	0,9
5	1,2	1,1	1,056	1,0	0,9
6	1,2	1,1	1,056	1,0	0,9

ii) Elevation

	1
Base	0,0
Level 1	5,486
2	7,315
3	9,144
4	10,973
5	12,803
6	14,630

Section classes:

All members to be angles

Section lists: Appendix L

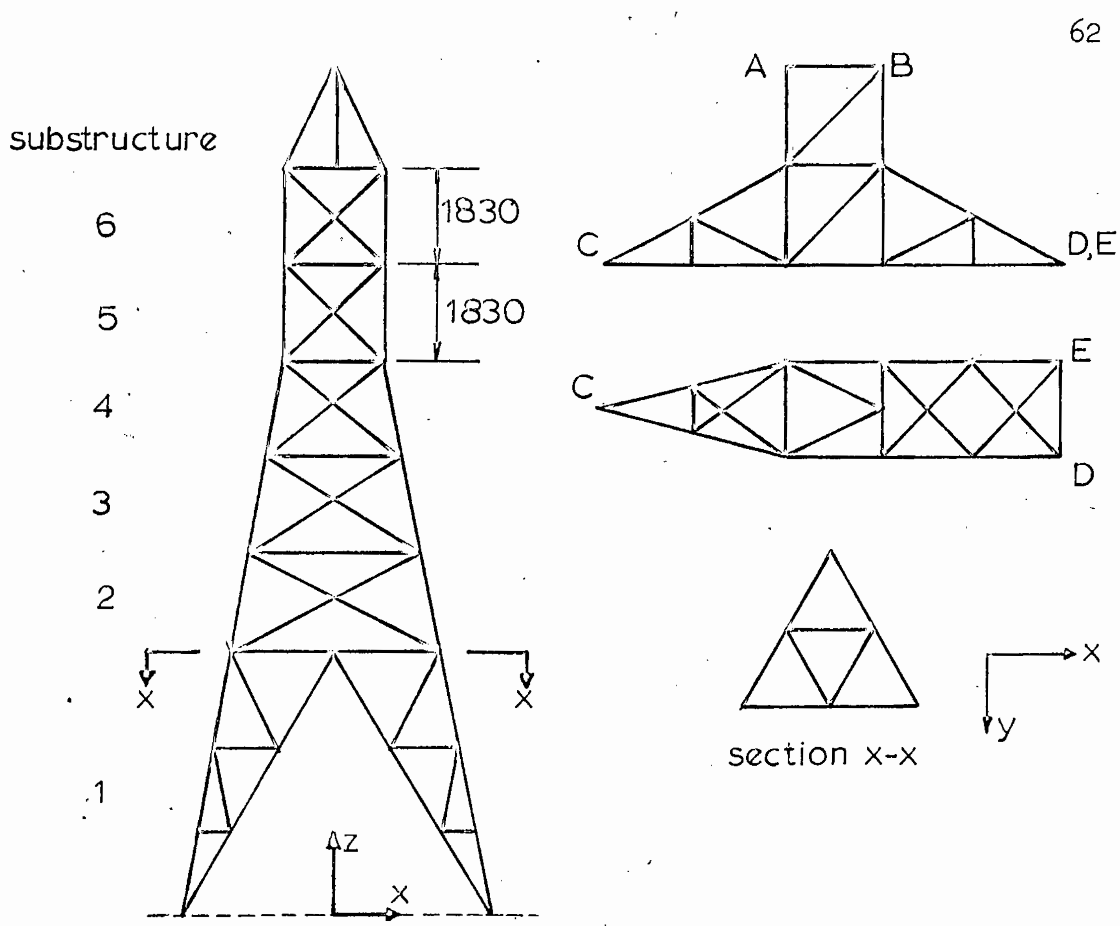


Figure 4.8 Three-legged Transmission Tower

Results:

Load Case 1: The configuration obtained from both programs DYNGEO and DYNPRE is identical. The comparison of forces are:

Substructure Number	Member Type	Member Number	DYSPAN	DYNGEO	DYNPRE
1	Main legs	1 4 10 16	-194,78 172,61 - 10,29 10,218	-194,78 172,61 - 10,29 10,28	Same as DYNGEO
	Interface	40 41	- 5,23 3,64	- 35,61 10,72	
	Bracing		minute	minute	minute
2	Main legs	73 74	-182,73 154,43	-170,77 149,89	

Substructure Number	Member Type	Member Number	DYSPAN	DYNGEO	DYNPRE
	Interface	76	1,605	2,655	Same
	Bracing	80 83	14,87 - 3,53	19,144 - 30,39	as
3	Main legs	85 86	-173,3 135,89	-153,32 131,53	DYNGEO
	Interface	88	0,25	2,36	
	Bracing	92 95	4,32 - 4,94	7,87 - 41,012	
4	Main legs	97 98	-147,4 99,3	-130,8 + 94,58	
	Interface	102	13,63	5,50	
	Bracing	103 104	- 22,71 19,652	- 19,314 23,04	
5	Main legs	109 110	-103,191 25,745	- 92,95 23,73	
	Interface	112	5,223	7,35	
	Bracing	115 116	- 34,95 27,98	- 33,6 29,33	
6	Main legs	121 122	- 28,09 11,83	- 24,72 15,78	
	Interface	126	3,00	4,186	
	Bracing	129 131	10,07 - 9,95	7,47 - 11,78	

Load Cases 2 and 3: Similar trends as above are shown for these load cases. The interface forces are extremely unreliable, but the main leg member forces approximate to those found by DYSPAN.

Configuration: The following configurations and total weights were found for the three Load Cases. State variable values:

Load Case	1		2		3	
	Radius	Z	Radius	Z	Radius	Z
Base	2,8	0,0	2,8	0,0	2,8	0,0
Level 1	1,8	5,486	1,8	5,486	1,8	5,486
2	1,6	7,315	1,6	7,315	1,6	7,315
3	1,2	9,144	1,2	9,144	1,2	9,144
4	1,0	10,973	0,9	10,073	0,9	10,973
5	0,9	12,803	0,9	12,803	0,9	12,803
6	0,9	14,603	0,9	14,603	0,9	14,603
Weight kN	23,256		16,786		18,508	

The optimum tower design chosen by this program is more slender than the original example presented by Kuzmanovic. However, the total weight (excluding tower 'arms'), of 23,25 kN, (Load Case 1), compares favourably with Kuzmanovic's value of 23,5 kN.

Structural Design: The structural design for the three load cases is shown in Figure 4.9. For all substructures, the design of the main leg members (member group 1) is consistent with that found by DYSPAN. This can also be seen from the comparison of forces. Design difference in the interface members are marked (member group 2). Little difference is however found in the design of the bracing (member group 3).

Computer Time	1	2	3 Load Case
DYSPAN		Accumulated time	4 min 41,273
DYNGEO	15 min 46,872	13 min 29,756	10 min 26,574
DYNPRE	6 min 27,081	4 min 12,173	3 min 36,89

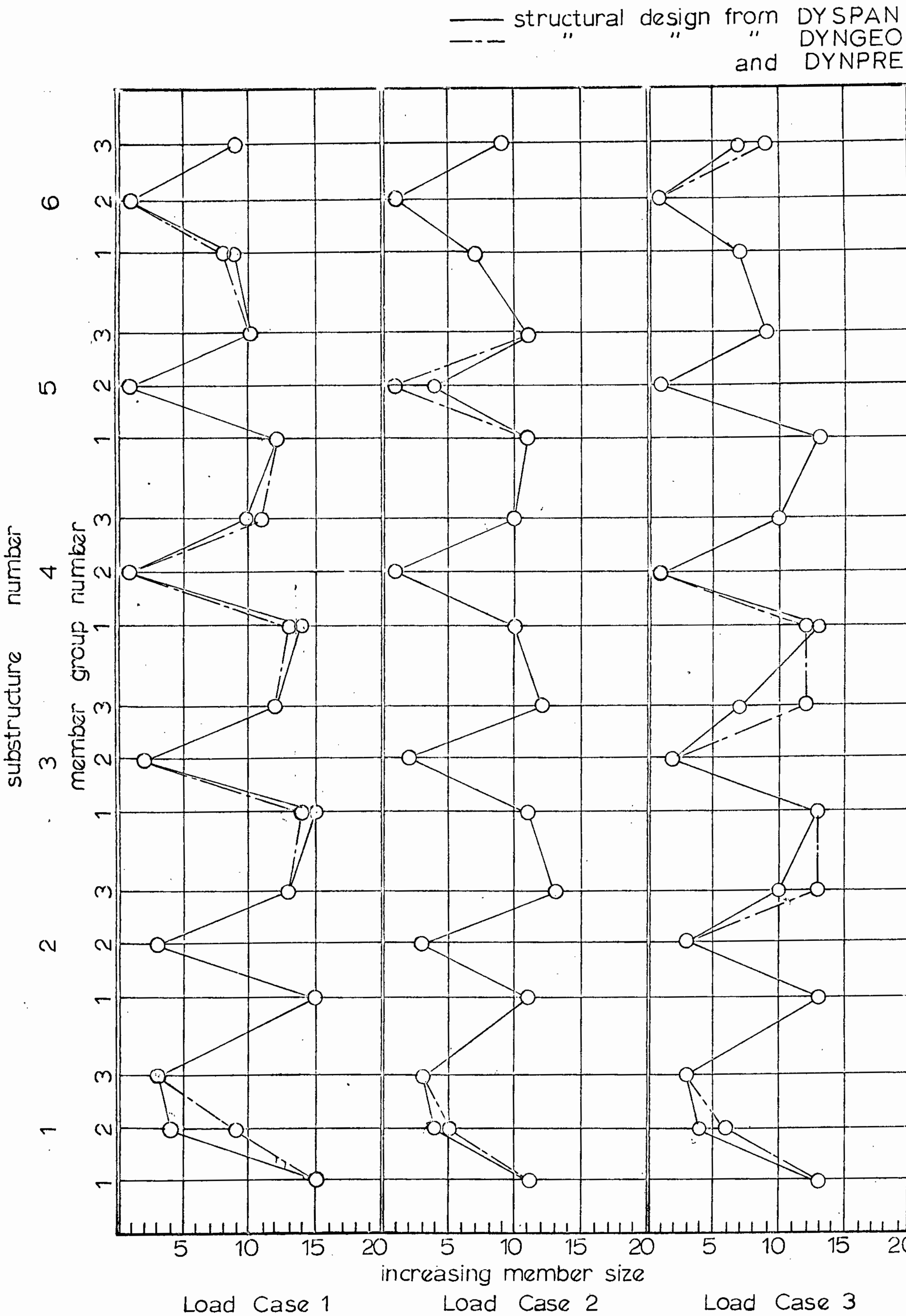


Fig 4.9 Structural Design

4.6 EXAMPLE 5 - RECTANGULAR TRANSMISSION TOWER (4 substructures)

This example is compared with the second discussed by Kuzmanovic et al [13] and is the design of a rectangular transmission tower subjected to the loads applied to example 4.

Initial Configuration: Height: 14,63 m

Dimensions at foundation level: 5,48 x 5,48 m

" " level 1: 3,66 x 3,66 m

" " " 2: 1,83 x 1,83 m

" " " 3: 1,83 x 1,83 m

" " " 4: 1,83 x 1,83 m

Total number of nodes: 56

Total number of members: 175

Loading: as for Example 4

Possible dimensions (state variables) at the 5 interface levels.

(Values used in Load Case 2 are shown in brackets)

i) x- and y-direction

	1		2		3		4	5
Base	2,74	(2,0)	-	(2,4)	-	(2,74)	- (3,0)	-(3,2)
Level 1	1,6	(1,4)	1,83	(1,6)	2,0	(1,83	(2,0)	(2,2)
Level 2	0,8	(0,6)	0,91	(0,8)	1,0	(0,91)	(1,0)	(1,2)
Level 3	0,8	(0,6)	0,91	(0,8)	1,0	(0,91)	(1,0)	(1,2)
Level 4	0,91	(0,6)	-	(0,8)	-	(0,91)	(1,0)	(1,2)

ii) z-direction

	1
Base	0,0
Level 1	5,49
Level 2	10,97
Level 3	12,8
Level 4	14,63

Section classes:

All members to be angles

Section lists: Appendix L

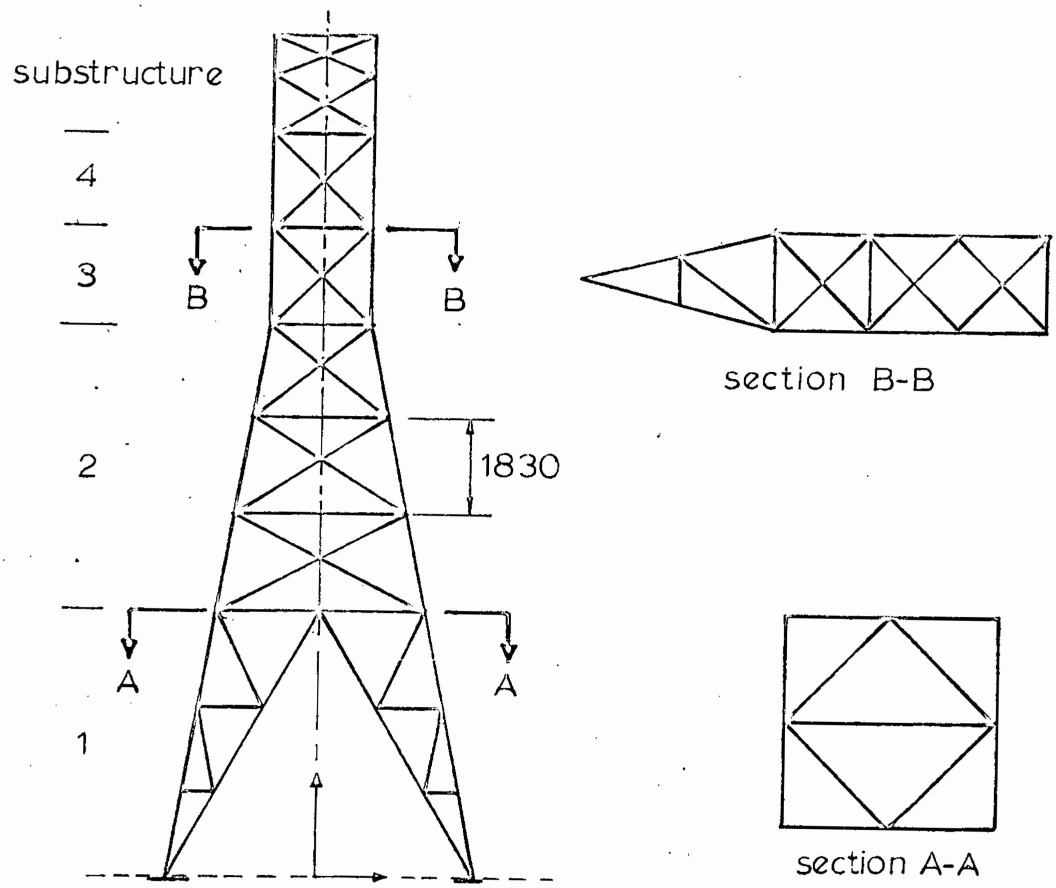


Figure 4.10 Rectangular Transmission Tower

Results:

A time limit of 30 minutes computer time was allowed for this structure. For Load Case 1 and 2, DYNGEO reached this limit without producing a result. With DYNPRE, however, feasible results were obtained for Load Cases 1, 2 and 3 in 15 min 43,137 sec; 27 min 48,335 sec and 18 min 19,666 sec respectively. Corresponding results from program DYSPAN are not available due to a violation of program limits for this problem. The configuration and structural designs from DYNPRE are given below.

Configuration:

The following configuration was produced for the state variables

Load Case	1			2			3		
	x	y	z	x	y	z	x	y	z
Base	2,74	2,74	0,0	2,0	2,0	0,0	2,74	2,74	0,0
Level 1	1,6	2,0	5,49	1,4	1,4	5,49	1,6	1,6	0,0
Level 2	0,80	0,8	10,97	0,8	0,91	10,97	0,8	0,8	10,97
Level 3	0,91	0,8	12,89	0,6	0,6	12,8	0,8	0,8	12,8
Level 4	0,91	0,91	14,63	0,6	0,6	14,63	0,91	0,91	14,63

Load Case 3, which is purely downward load, produces the most slender structure, i.e., an attempt has been made by the computer to construct a single strut to support the loads. This was also evident in Example 3.

Structural Design: Figure 4.11 shows the structural design of the tower for the 3 load cases. Although this diagram cannot be used to compare designs critically, it does show that a variety of load cases produce similar trends in the structural design. For example, member group 1 in all substructures are very similar. The torsion effects from load case 2 cause the design of substructure 1 to be considerably different for some member groups.

Structural weight:

Load Case 1	26,540 kN
2	23,515 kN
3	17,415 kN

The weight registered here of 26,540 kN once again compares favourably with Kuzmanovic's value of 26,765 kN.

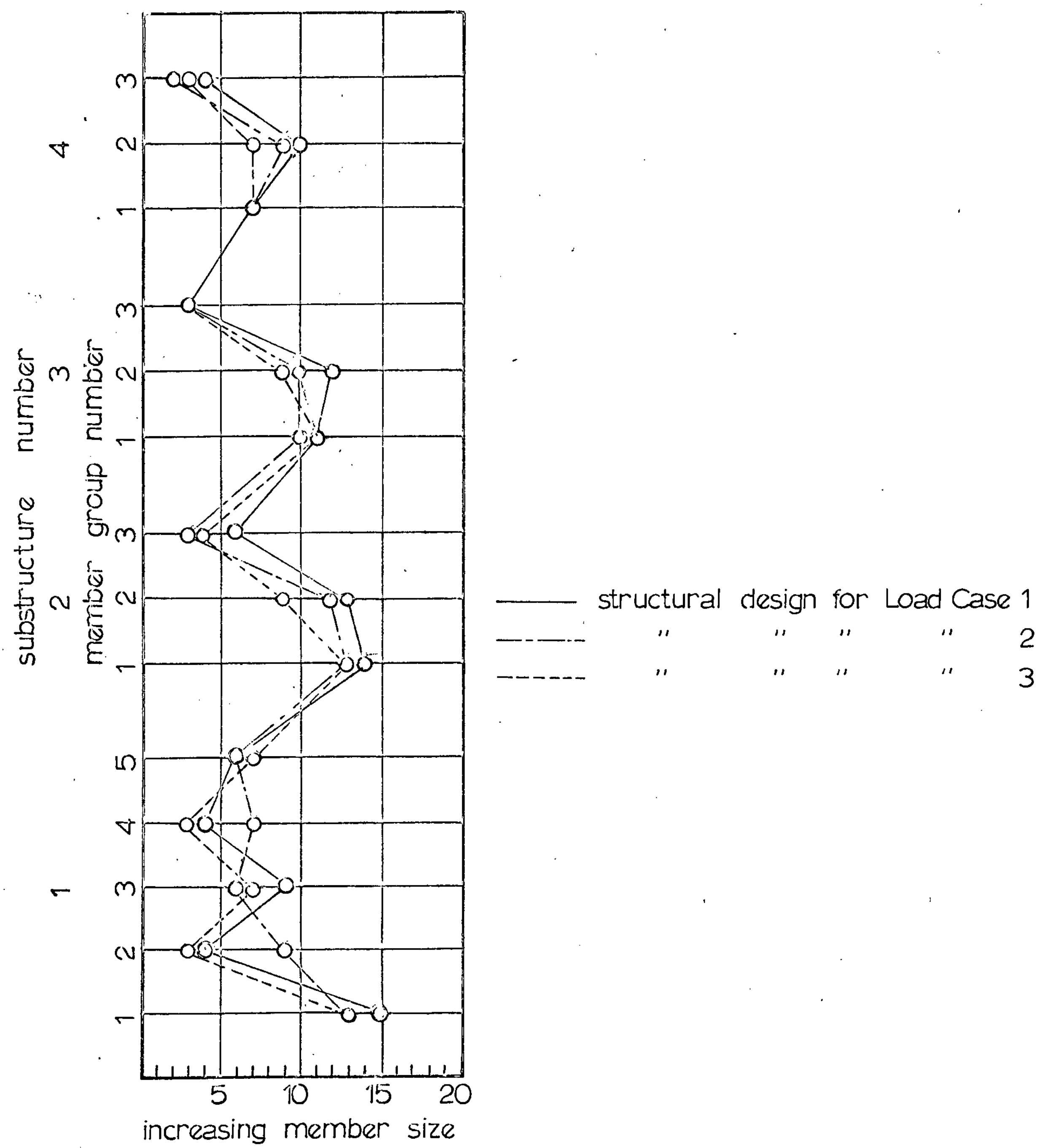


Fig 4.11 Structural Design

4.7 EXAMPLE 6 - PRACTICAL TRANSMISSION TOWER (3 substructures)

This transmission tower is similar to a typical tower used in South Africa. This example illustrates that a practical steel tower constructed from a large number of members can be 'automatically' designed with these programs.

Initial Configuration:	Height:	12,35 m	
	Dimensions at foundation level:		6,7 x 6,7 m
	" " level 1:		5,0 x 5,0 m
	" " " 2:		3,9 x 3,9 m
	" " " 3:		2,8 x 2,8 m
	" " " 4:		5,8 x 1,9 m
	Total number of nodes:		115
	Total number of members:		336

The tower is shown in Figure 4.14.

The total number of nodes and members was too great for the direct application of the programs. Therefore, the structure has been divided into two sections, each with its own set of equivalent loads. The lower section, which consits of 3 substructures was designed first. The best interface dimensions obtained were then used as the lower interface dimensions for the upper section. The final results was obtained by combining the two designs.

Possible dimensions (state variables) at the 4 levels.

i) x- and y-directions

	1	2	3
Base	3,35	3,6	
Level 1	2,5	2,8	
Level 2	1,95	1,8	
Level 3	1,4	1,2	1,6
Level 4	2,9		

ii) z-direction

	1
Base	0,0
Level 1	2,65
Level 2	4,45
Level 3	6,25
Level 4	13,45

Loading: An equivalent set of loads was applied to the lower structure at level 3. These loads at A,B,C,D in Figure 4.12 were:

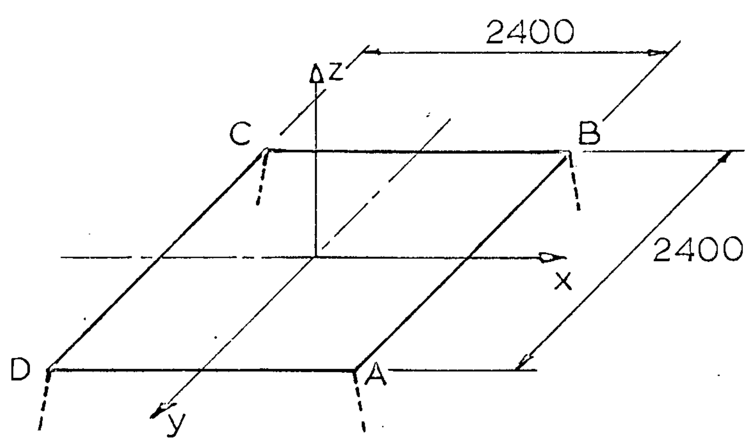


Figure 4.12 Level 3 Interface

Load case	1			2			3		
	x	y	z	x	y	z	x	y	z
A	-20,0	0,0	31,0	-14,9	8,0	33,6	0,0	0,0	-96,0
B	-20,0	0,0	31,0	- 4,6	8,0	37,9	0,0	0,0	-96,0
C	-20,0	0,0	-138,0	- 4,6	-2,3	-85,7	0,0	0,0	-96,0
D	-20,0	0,0	-138,0	-14,9	-2,3	-90,0	0,0	0,0	-96,0

Loads on the upper structure were (Figure 4.13)

E	-20,0	0,0	- 48,5	-24,15	5,4	-19,6	0,0	0,0	-96,0
F	-20,0	0,0	- 48,5	4,76	5,4	- 7,6	0,0	0,0	-96,0
G	-20,0	0,0	- 58,3	4,76	0,4	-32,5	0,0	0,0	-96,0
H	-20,0	0,0	- 58,3	-24,15	0,4	-44,48	0,0	0,0	-96,0

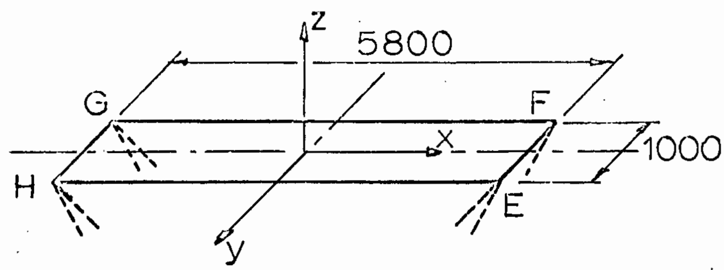


Figure 4.13 Level 4 Interface

Section classes:

- All members to be angles

Section lists: Appendix L

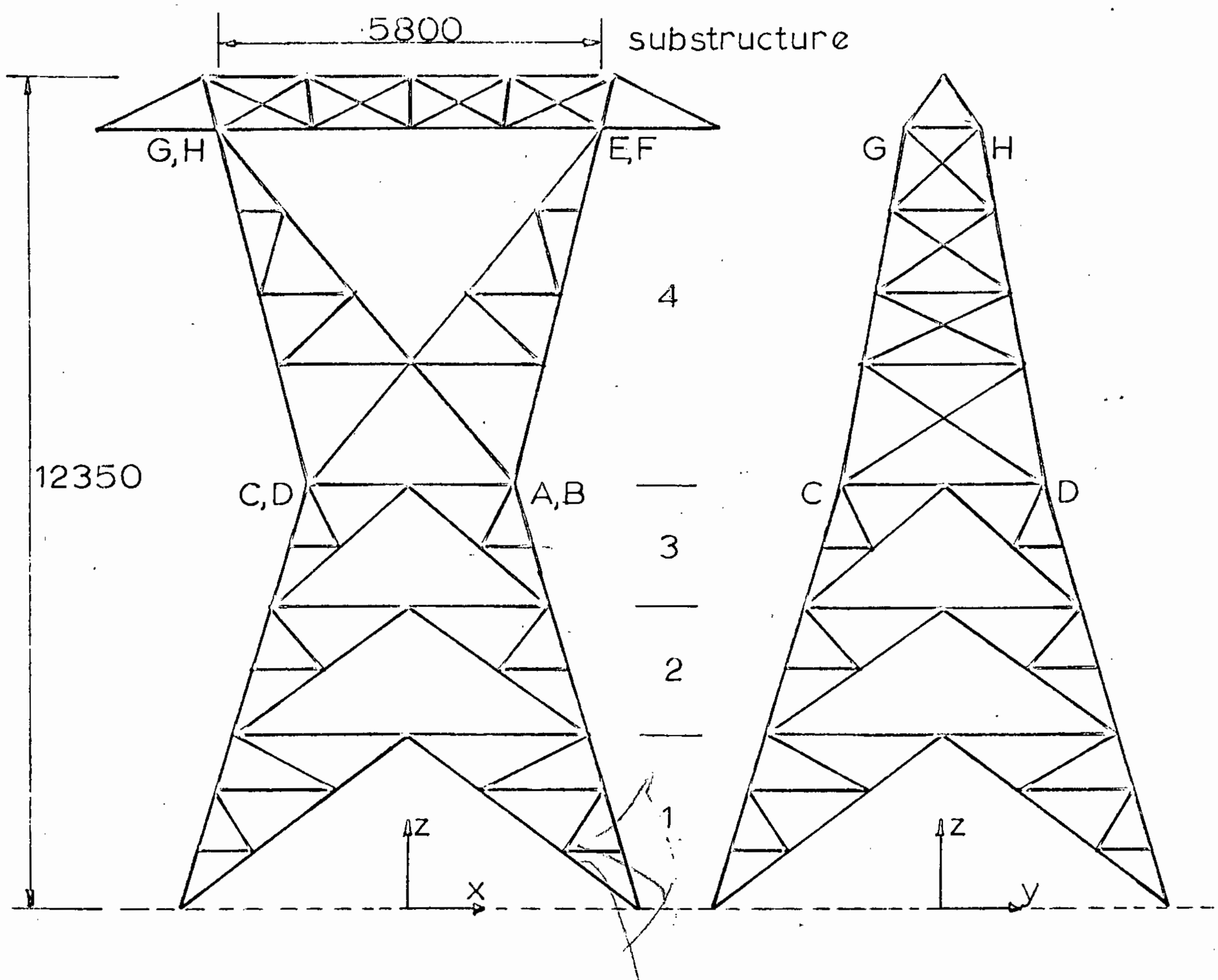


Figure 4.14 Practical Transmission Tower

Results:

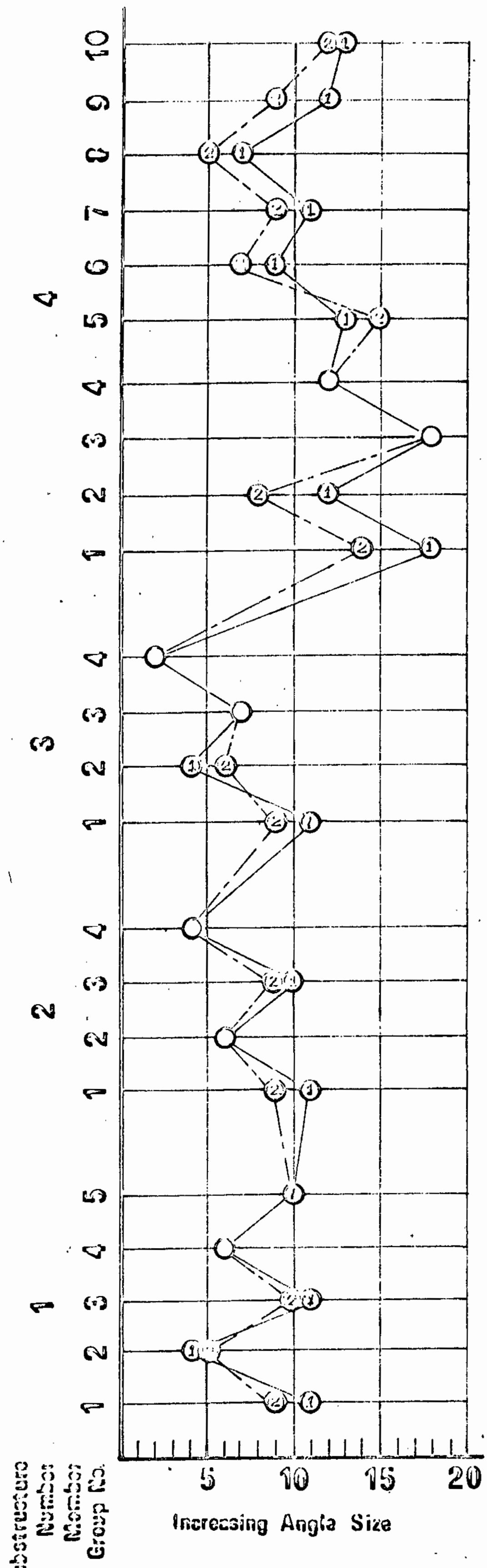
A time limit of 30 minutes computer time was allowed for the two halves of the structure separately. DYNGEO could not find a feasible solution in this time. DYNPRE produced solutions for Load Cases 1 and 2 for both upper and lower structures. No solution could be found for Load Case 3 for the lower structure. Load Case 3 on the upper structure required a section larger than that given in the section list and hence the calculation was terminated. Due to the considerable size of the whole structure, DYSPAN could not be used.

Structural Design: Figure 4.15 shows the structural design of the tower for the two load cases as mentioned above. A general trend of compatibility of designs from varying load cases is observed. The actual detailed loading on the tower is not known, but from the available data, it appears that the loads used in the program were approximately 4 times the actual values.

Main leg and boom members from the actual design were 88,9 x 88,9 x 9,53 angles. The corresponding members found by the program were:

	Load Case 1	Load Case 2
Main Leg Member	80 x 80 x 9,63	70 x 70 x 6,38
Boom Members	150 x 150 x 27,3	100 x 100 x 15

The large boom members were due to the very large downward loads from Load Case 1. A total of 276 kN was applied to each boom together with horizontal wind forces. Unfortunately, due to the unavailability of loading information a direct comparison cannot be made between the results obtained from the program and the actual practical designs. However, very plausible results were obtained from the assumed loading which appears to be far greater than that which was actually applied to this structure.



—— Structural design for load case 1

—·—· " " " " " 2

Figure 4.15 Structural Design

4.8 DISCUSSION OF RESULTS

The program DYSPAN is considered to be a basis of all comparisons, since it analyses and designs structures as a whole. Consequently, the forces obtained from DYSPAN represent a true elastic solution. In addition, the members are designed from these forces by proportioning the member sizes in order to produce a minimum difference between actual and permissible stresses for each member in the structure.

DYNGEO's use of substructures requires an approximation to be made for the interaction between substructures. Force transfer from one part of the structure to the next is calculated from statics. The equivalent set of loads are therefore assumed to approximate to the true interaction forces.

Each interface is subjected to an equivalent set of loads applied so that the overall forces and moments at that level balance the externally applied loads. The forces in the interface members are therefore inaccurately calculated since the effective stiffness at the interface level is not consistent with the actual state. The design of these members is, however, not critical since stress levels are very low (average 6 MPa). It has been found that the degree of force inconsistency is dependent on the distance between the external loads and the interface under consideration. Large differences are found at lower interfaces while values approaching the actual are found in the interface members near the top of the tower. The increase in the interface loads with decrease in elevation is due to moment effects caused by the external loads, which is analogous to the moments on a cantilever.

The design of the main leg members, which are the main load carriers, is all important. Excellent agreement between both the forces and the design has been found by using the two programs DYSPAN and DYNGEO. This can be seen from diagrams Figures 4.4; 4.6; 4,9 and 4,11. In particular, Figure 4.9

the structural design of the triangular plan tower, shows no difference between the designs by DYSPAN and DYNGEO.

Where vertical loads only are applied to a structure, the program DYNGEO finds the narrowest possible solution. This is logical since a simple strut would be the most efficient and the cheapest method of supporting the load. The design in this case is completely compatible with that found by DYSPAN.

An interesting observation is that similar trends in the structural design of a particular tower are evident even though the load cases are vastly different. Examples 4, 5 and 6 show this trend clearly.

The optimum configuration of the structure is dependent upon the accuracy of the prediction of the forces in the members. An inefficient prediction produces slightly different designs which in turn have varying 'costs' or weights. As the DPSA method chooses the design and configuration with the least cost, a different geometry may be chosen by the programs DYNGEO and DYNPRE. This is more critical for small structures where a change in configuration can cause considerable changes in forces. For example, the relative configuration (and hence force) change for a 4 metre tower subjected to a dimension change of 0,5 metres at some level, is far greater than the same dimension change on a 16 metre tower. Consequently, the inaccuracy in the prediction calculations used in DYNPRE are far more evident for small structures. This can be seen from the designs in examples 2 and 4. Both structures are triangular in plan but the former is 4 metres high while the latter is 14,63 metres high. The comparison of respective designs shows a larger overall difference between the results from the two programs. Indeed, there is no difference between designs in example 4.

The absence of published work for the combination of structural and geometrical design of towers makes it impossible to quantitatively criticize the designs produced by DYNGEO and DYNPRE. Therefore, the only yard stick for a comparison of any type was by using the results produced by the full elastic analysis and design program DYSPAN.

CHAPTER 5

CONCLUSIONS

A computer technique has been developed to determine the optimum geometric configuration and member sizes automatically for tower structures. A prohibitive amount of data preparation and computational time would be required if numerous configurations are selected manually. The application of Dynamic Programming, in particular the Dynamic Programming Successive Approximations technique, is suitable for the design of tower structures, since substructuring provides the necessary state variables. The Direct Iteration Method, in conjunction with Dynamic Programming, is a convenient way of finding optimum member sizes and a geometric configuration concurrently. Substructuring also has the added advantage of simplifying coding and checking of data.

Only a single data set is required to generate many configurations from an initial configuration. Variations in the dimensions of the structure can be produced by merely specifying values of the state variables at the interfaces.

Three types of tower structures which have been satisfactorily designed are:

(a) Plane truss towers

(b) Rectangular 3-dimensional towers

(c) Triangular 3-dimensional towers

The final designs selected by each of the three programs for the above examples are the same for all practical purposes. Some secondary member forces differ significantly, however. The design of these members is usually governed by their slenderness ratios and is not dependent upon their stresses, since these are negligible. This is a phenomenon which is often overlooked by structural analysts who prefer more rigorous methods of analysis and techniques for optimum solutions.

Large computer calculation times are required for many complex structures. Substructuring is essential to decrease both the analysis times and the computer storage requirements for these structures. It is important to degenerate any particular structure into as many substructures as possible in order to decrease the computational time to a minimum. This has been clearly shown by examples 4 and 5 in Chapter 4. The design for example 4, which contains 6 substructures, was determined in approximately half the time required by example 5 (4 substructures) even though the number of state variables has been restricted in the latter. The size of the substructures is also of major importance. The lower substructure in example 5 is substantially larger than the others. This results in an appreciable increase in computer time and storage requirements.

The method of predicting forces by simple linear interpolation from four controlling substructure configurations has proved successful for large structures. Prediction errors occur for small structures where the relative configuration changes are considerable in comparison with the same dimension changes in large structures. This technique has however shown that a substantial reduction in computer time is possible without sacrificing design efficiency. This interpolation method, used in the program DYNPRE requires only a quarter of the computer time used by the complete DPSA program DYNGEO. Greater accuracy could be attained if a higher order interpolation was used, however this will increase the computation times.

Scope for Further Research

The

other areas

structures c

that the met

substructure

Vari

1. Sele

It h

can be perfo

overall stru

computer tim

structure in

The

i) Anal

ii) Assu

iii) Desi

conf

iv) Calc

v) Use

from

vi) Re-a

vii) Revi

Disa

i) A true optimum design cannot be gauranteed since all possible substructure configurations will not be analysed as in the current technique

ii) A large computer will be required to analyse the whole structure for any reasonably complex practical problem.

The following refinements can be incorporated although they are probably unnecessary, since the interface member forces are generally negligible.

2. Overlapping

In order to determine the forces in the interface members more accurately, it is proposed that substructures be overlapped. Two adjoining substructures can be analysed together to predict the actual interface member forces.

3. Prescribed Displacements

Alternatively, the interface nodal displacements can be found from a full analysis. These displacements can then be used as prescribed support movements for each substructure. This could decrease the minor errors inherent in some member forces.

4. Applied Member Forces

Another method is to proceed from substructure to substructure from the top of the tower downward. The member forces can be calculated for each substructure in turn. Their effects can then be applied as loads to the lower substructures in a successive manner. The calculation of equivalent interface loads is then unnecessary.

Generally the computation times for large structures are still excessive. If a more rapid method can be found to determine member forces, the DPSA method would become even more satisfactory.

Limit state design has not been mentioned. The recommendations of the new steel code (BS 449, which is not yet available), can however easily be incorporated in these programs in the future.

REFERENCES

1. R.E. Bellman, Dynamic Programming, Princeton University Press. Princeton, N.J. 1957.
2. R.E. Bellman and S.E. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, N.J. 1962
3. R. Aris, Discrete Dynamic Programming, Blaisdell Publishing Company, N.Y. 1964.
4. R.E. Larson, State Increment Dynamic Programming. American Elsevier, New York, 1968.

Survey of Dynamic Programming Computational
EE Transactions of Automatic Control, AC-12,

.J. Korsak, 'A Dynamic Programming Successive
hnique with Convergence Proofs'. Automatica,
).
timum Structural Design by Dynamic Programming'.
SCE, 94, ST8 1887-1906, (1968).
.J. Sheppard, 'Optimizing the Shape of
tures'. Proc. of Inst of Civ Eng, 47, 363-376,

A.C. Palmer, 'Optimal Design of Transmission
Programming'. Computers and Structures,
).
namic Programming and Structural Optimization'.
on Computers in Optimization of Structural Design.
posium at University of Wales, Swansea, Jan 1972.

4) DS 519.7 LARS
70/5505

DS 624.1771 KIRS
80/66774

11. P. Packia Raj and S. Olani Durrant, 'Optimum Structural Design by Dynamic Programming'. Jnl of Struct Div., ASCE, ST8, 1575-1589, (1976).
12. W. Weaver and F.W. Patton, 'Automated Design of Space Trusses'. AISC Eng Jnl., 5, 26-36, (1968).
13. B.O. Kuzmanovic, N. Willems and F.M. Thomas, 'Automated Design of Three-legged Steel Transmission Towers'. Computer and Structures, 7, 171-182, (1977)
14. G.C. Howell and W.S. Doyle, 'Dynamic Programming and Direct Iteration for the Optimum Design of Skeletal Towers'. Paper to be presented at 'Symposium on Future Trends in Computerized Structural Analysis and Synthesis'. To be held October-November 1978. Washington, DC, USA.

Additional References

15 V.B. Venkayya, N.S. Khot and V.S. Reddy, 'Optimization of Structures Based on the Study of Energy Distribution'. 2nd Conf. on Matrix Methods in Structural Analysis, Wright Paterson, AFFDL-TR-68-156, October 1968.

16. L. Berke and V.B. Venkayya, 'Review of Optimum Critical Approaches to Structural Optimization'. ASME, AMD, 7, 23-34, (1974).

17. L.A. Schmidt and T.P. Kicher, 'Synthesis of Materialsand Configuration Selection'. ASCE (ST3), p 79, (1962)

18. L.C. Schmidt, 'Minimum Weight Layouts of Elastic Statically Determinate Triangular Frames Under Alternative Load Systems'. J. of Mech and Phys of Solids. 10, 139-151, (1962).

19. R. Gallagher and O. Zienkiewitz, eds, Optimum Structural Design. Wiley, London, 1973.

20. Reinschmidt, Cornell, Brotchie, 'Iterative Design and Structural Optimization'. J. Struct Div., ASCE, ST6, 92, 281-318, (1966)

21. U. Kirsh, M. Reiss and U. Sharwin, 'Optimum Design by Partitioning into Substructures'. J. Struct Div., ASCE ST1, 98, 249-267, (1972)

22. U. Kirsh and Rubinstein, "Structural Reanalysis by Iteration'. Computers and Structures, 2, 497, (1972).

23. Spillers and Al-Bonna, 'Optimization using Iterative Design Techniques'. Computers and Structures, 3, 1263-1271, (1973).

24. D.M. Vanderplaats and F. Moses, 'Automatic Design of Trusses for Optimum Geometry'. J. Struct Div., ASCE ST3, 98, 671-690, (1972)

25. Alspaugh and Kazuo Kunoo, 'Optimum Configuration and Dimensional Design of Truss Structures'. Computers and Structures. 4, 755-770, (1974).

26. D.P. Mondkar and G.H. Powell, 'Towards Optimal In-core Equation Solving'. Computers and Structures, 4, 551-548, (1974).

27. D. Johnson and B.M. Brotten, 'Optimum Design of Redundant Trusses'. J. Struct Div., ASCE (ST12), 95, 2589-2610, (1969)

28. P. Pederson, 'Optimal Joint Positions for Space Trusses'. J Struct Div., ASCE (ST12), 99, 2459-2478, (1973).

29. D. Kavlie and G. Powell, 'Efficient Re-analysis of Modified Structures'. J. Struct Div., ASCE (ST1), 97, 377, (1971).

30. W.S. La Pay and G.G. Goble, 'Optimum Design of Trusses for Ultimate Loads'. J. Struct Div., ASCE (ST1), 97, 157, (1971).

31. A.R. Toakley, 'Optimum Design Using Available Sections'. J. Struct Div., ASCE (ST5), 94, 1219-1241, (1968).

32. R.J. Melesh and R. Luik, 'Multiple Configuration Analysis of Structures'. J. Struct Div., ASCE (ST11), 94, 2581-2595, (1968).

APPENDIX A

A1.1 EQUIVALENT SUBSTRUCTURE LOADS

Three plan shapes of structure have been considered in the formulation of the DPSA method into a viable computer program, each of which requires a specific calculation of equivalent substructure loads, although all use a similar technique.

Consider a general point A on the z-axis and at a cell-face level (Figure A1).

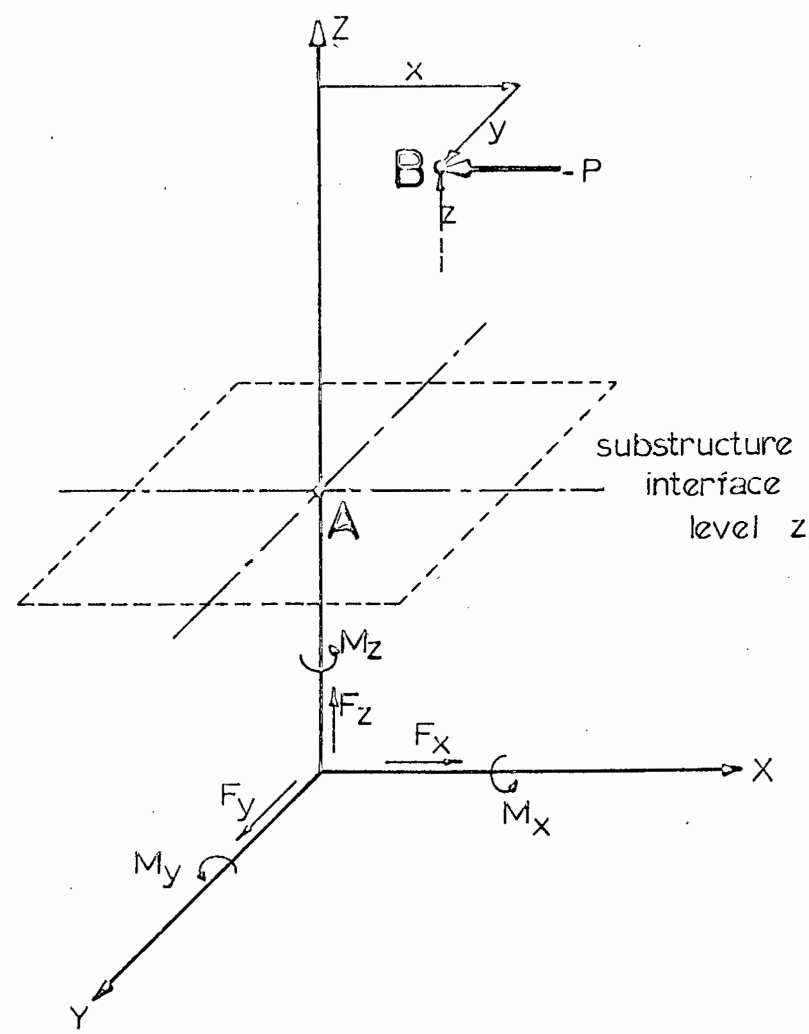


Figure A1 Equivalent Loading System for Point A

Any external load which is applied to the structure at a point of greater elevation (i.e. larger z-coordinate) can be replaced at A by an equivalent set of loads and moments related to the sign convention specified.

Example: Apply a load $-P$ at point B (x,y,z);

Find the equivalent loads and moments at A

Solution:

$$F_x = -P$$
$$F_y = 0$$
$$F_z = 0$$
$$M_x = 0$$
$$M_y = P\,(z-z\ell)$$
$$M_z = -P\,y$$

In general terms, loads P_x, P_y and P_z applied at (x,y,z) produce equivalent cellular loads of:

$$F_x = P_x$$
$$F_y = P_y$$
$$F_z = P_z$$
$$M_x = P_y\,(z-z\ell) - P_z\,y$$
$$M_y = P_z\,x - P_x\,(z-z\ell)$$
$$M_z = P_x\,y - P_y\,x \qquad \text{(A.1)}$$

These loads must now be applied to the substructure interface nodes in some proportion so that their effect is equivalent to that at A. [It should be noted that loads can only be applied at nodes in the displacement method of analysis and as point A is usually not a structural node, it cannot be used as a load point].

2. Space-truss tower - Rectangular plan shape (Figure A3)

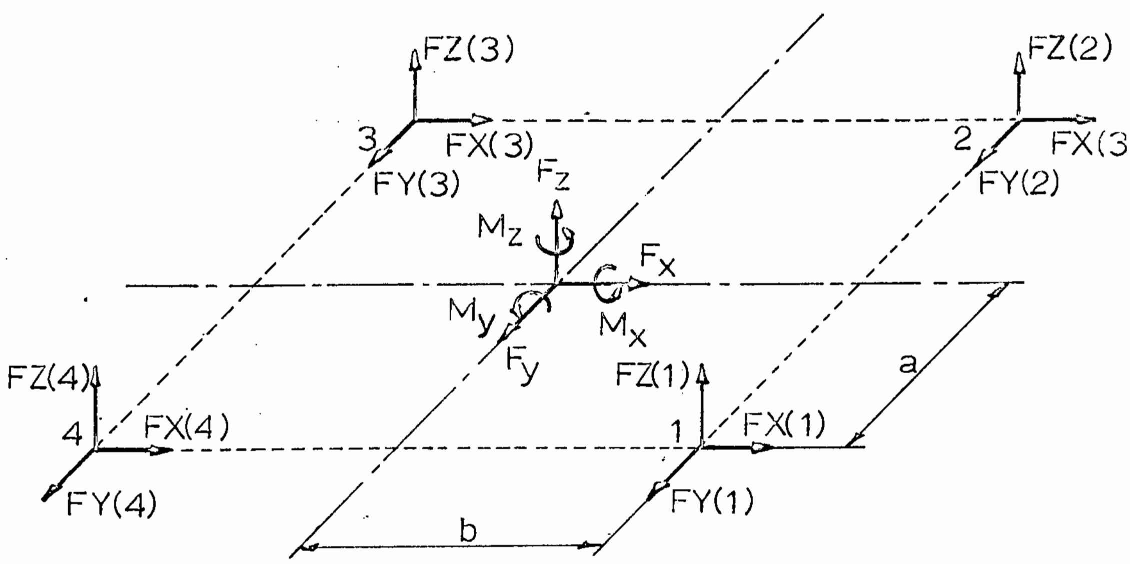


Figure A3 Equivalent Loads for a Rectangular Tower

The equivalent nodal loads are:

$$FX(1) = F_x/4 + M_z/(8\ a)$$

$$FY(1) = F_y/4 - M_z/(8\ b)$$

$$FZ(1) = F_z/4 - M_x/(4\ a) + M_y/(4\ b)$$

$$FX(2) = F_x/4 - M_z/(8\ a)$$

$$FY(2) = F_y/4 - M_z/(8\ b)$$

$$FZ(2) = F_z/4 + M_x/(4\ a) + M_y/(4\ b)$$

$$FX(3) = F_x/4 - M_z/(8\ a)$$

$$FY(3) = F_y/4 + M_z/(8\ b)$$

$$FZ(3) = F_z/4 + M_x/(4\ a) - M_y/(4\ b)$$

$$FX(4) = F_x/4 + M_z/(8\ a)$$

$$FY(4) = F_y/4 + M_z/(8\ b)$$

$$FZ(4) = F_z/4 - M_x/(4\ a) - M_y/(4\ b) \qquad (A.3)$$

3. Space-truss tower - Triangular plan shape (Figure A4)

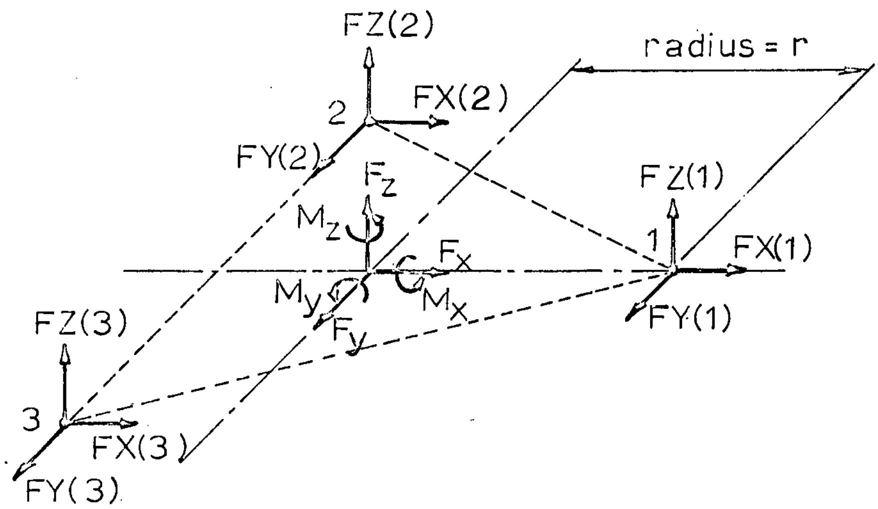


Figure A4 Equivalent Loads for a Triangular Tower

The equivalent nodal loads are:

$$
\begin{aligned}
FX(1) &= F_x/3 \\
FY(1) &= F_y/3 - M_z/(3\,r) \\
FZ(1) &= F_z/3 + M_y/(3\,r) \\
FX(2) &= F_x/3 - M_z/(2\sqrt{3}\,r) \\
FY(2) &= F_y/3 + M_z/(6\,r) \\
FZ(2) &= F_z/3 + M_x/(\sqrt{3}\,r) - M_y/(3\,r) \\
FX(3) &= F_x/3 + M_z/(2\sqrt{3}\,r) \\
FY(3) &= F_y/3 + M_z/(6\,r) \\
FZ(3) &= F_z/3 - M_x/(\sqrt{3}\,r) - M_y/(3\,r)
\end{aligned}
\tag{A.4}
$$

The above formulae are universally applicable to the shape of structure for which they have been derived.

A1.2 EQUIVALENT SUBSTRUCTURE SUPPORT CONSTRAINTS

The need for substructure boundary conditions has arisen due to the degeneration of the structure into individual substructures (Section 3.3.3.1). It is necessary to produce a stable structure which can be analysed using a displacement method. Stabilization of supports is affected by fixing the lower substructure interface nodes rigidly in space, that is, it is assumed that the substructure exists independently of the rest of the structure and is supported on its own rigid foundations.

APPENDIX B

B1.1 A ~~NUMBERICAL~~ NUMERICAL EXAMPLE OF THE DYNAMIC PROGRAMMING TECHNIQUE

The Dynamic Programming technique can best be illustrated by a simple problem representing a network of possible routes between two fixed points.

Consider a network of routes between points A and B as shown in Figure B1 which represent the only possible routes between the two points.

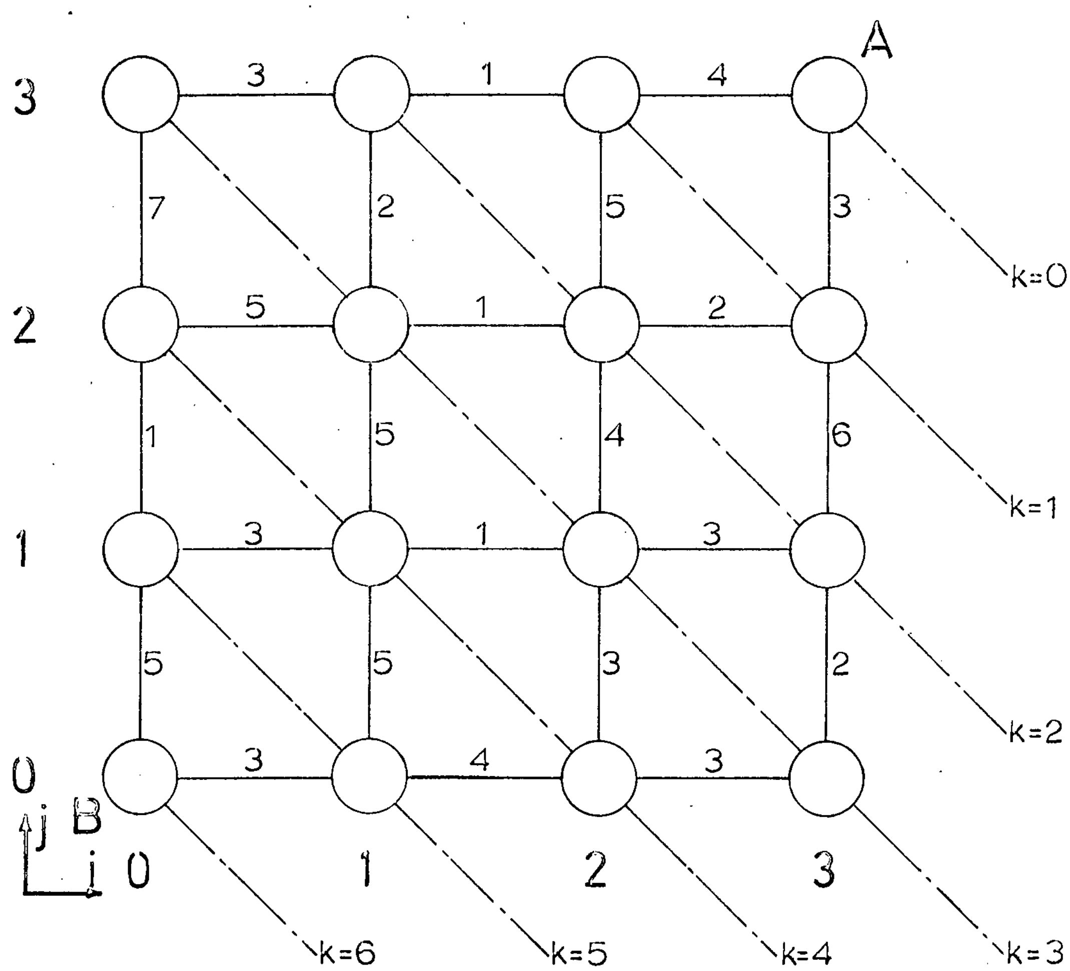


Figure B1 Simple Network Problem

It is required to move from point B to point A by the least 'expensive' route. The cost of travelling between any two adjoining points (i,j) and (k,ℓ) is represented by the numeral attached to the line and is defined as $t(i,j;k,\ell)$. The circles at the node points will be used for the accumulated cost of the optimum route between that point (i,j) and A, and is defined as $f(i,j)$.

The problem, in accordance with the equations in Chapter 2 is therefore:

Find a route denoted by $u(k)$ for $k = 0$ to 6 between A and B so that J is a minimum

where

$$J = \sum_{k=\cancel{\phi}0}^{k=6} \ell(u(k)) \tag{B.1}$$

and ℓ is the cost of the route between two adjoining nodes

The route is restricted to movements 'up' in the positive j-direction or 'across' in the positive i-direction from any point in the network.

The sequence of decisions which must be made to produce an optimum cost route from node (i,j) to A is (in terms of equation 2.1).

$$f(i,j) = \min[t(i,j;k,\ell) + f(k,\ell)] \tag{B.2}$$

At each point (i,j), there are two possible routes to an adjacent node:

1. (i,j) to $(i+1,j)$
2. (i,j) to $(i,j+1)$

Consequently equation (B.2) is:

$$f(i,j) = \min \begin{cases} t(i,j;\ i+1,j) + f(i+1,j) \\ t(i,j;\ i,j+1) + f(i,j+1) \end{cases} \qquad (B.3)$$

The Dynamic programming solution is found by a systematic re-use of equation (B.3).

The demonstration of equation (B.3) can be achieved by considering Figure B2 where, by definition, $f(3,3) = 0$.

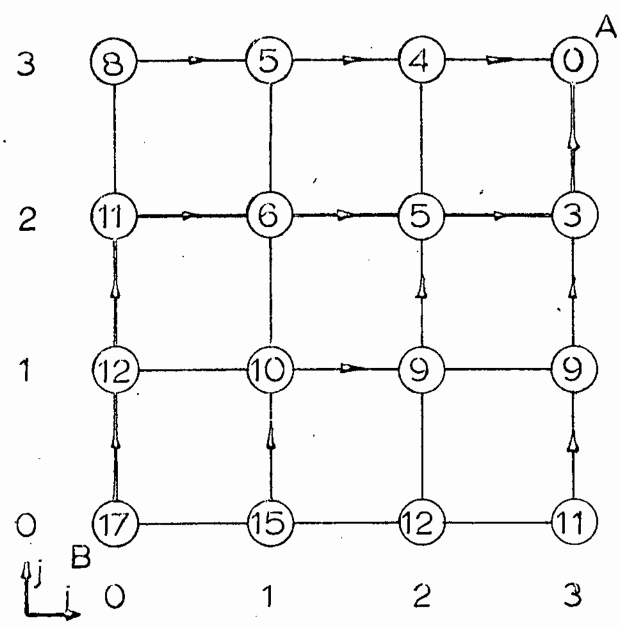


Figure B2 Route from B to A

1. for $k = 1$, two possible routes exist:

(2,3) to (3,3) for which $f(2,3) = 4$

and (3,2) to (3,3) for which $f(3,2) = 3$

As an aid to visualization of the process, these accumulated costs are written in their corresponding circles and an arrow indicates the sense of direction of the best route.

2. for k = 2, four possible routes exist, of which two can immediately be evaluated:

$$f(1,3) = t(1,3;\ 2,3) + f(2,3) = 1 + 4 = 5$$

$$f(3,1) = t(3,1;\ 3,2) + f(3,2) = 6 + 3 = 9$$

The route from (2.2) is found by using equation (B.3)

$$f(2,2) = \min \begin{cases} t(2,2;\ 2,3) + f(2,3) \\ t(2,2;\ 3,2) + f(3,2) \end{cases}$$

$$= \min \begin{cases} 5 + 4 \quad = \ 5. \\ 2 + 3 \end{cases}$$

which corresponds to the route (2,3), (3,2),A as shown in Figure B2.

3. This calculation is continued through stages k = 3,4 and 5 to find the optimum route from (0,1) to A and from (1,0) to A. For k = 6

$$f(0,0) = f(B) = \min \begin{cases} t(0,0;\ 0,1) + f(0,1) \\ t(0,0;\ 1,0) + f(1,0) \end{cases}$$

$$= \min \begin{cases} 5 + 12 \quad = 17 \\ 3 + 15 \end{cases}$$

The final optimum route can now be traced through the network by following the arrows. The optimum route is (0,0),(0,1), (0,2), (1,2) (2,2), (3,2), (3,3). Figure B2 shows the completed diagram and the optimum route from B to A.

APPENDIX C

C1.1 A NUMERICAL EXAMPLE OF THE DYNAMIC PROGRAMMING SUCCESSIVE APPROXIMATION TECHNIQUE

The DPSA technique can best be demonstrated by the use of a two-dimensional dynamic programming problem involving the production of an air conditioning duct transition section of the type shown in Figure C1.

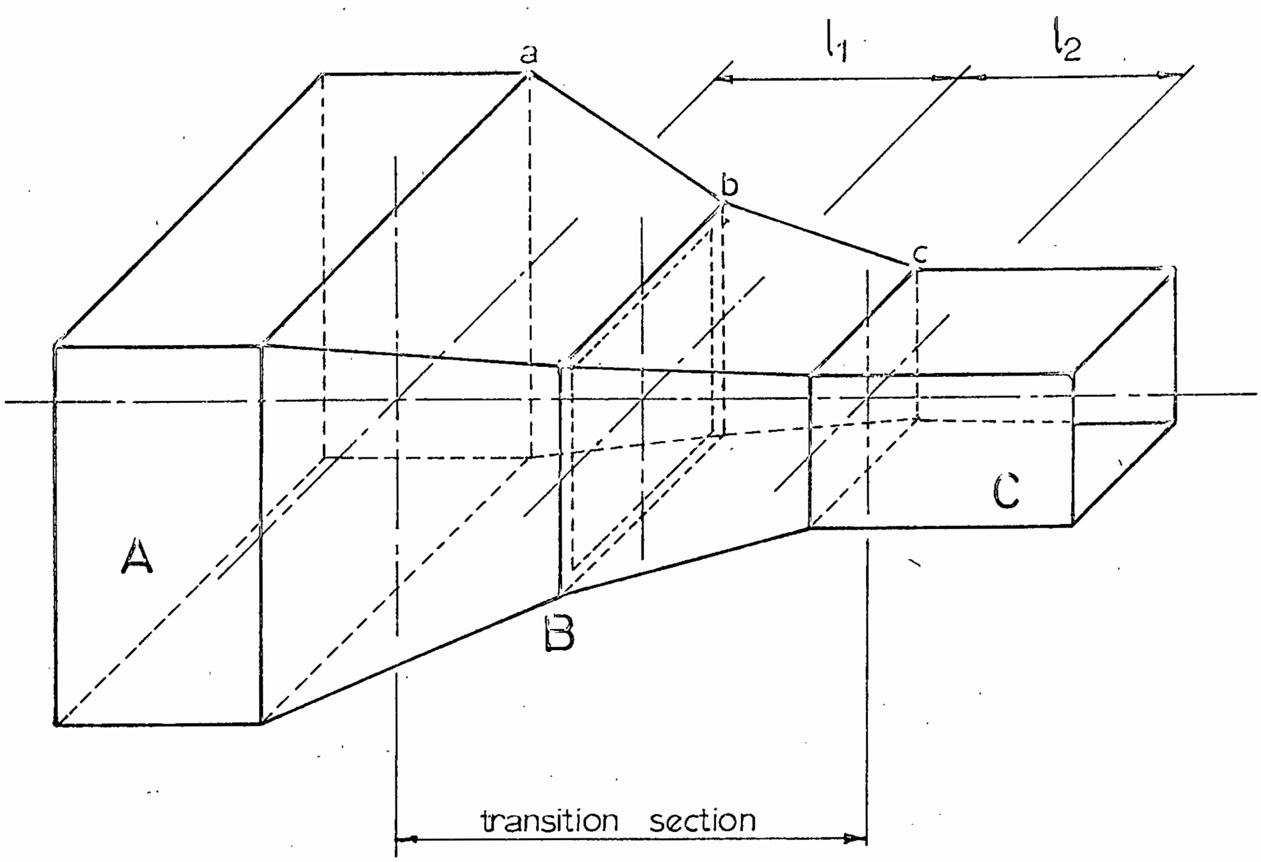


Figure C1 Air Conditioning Duct Transition Section

It is required to design a transition section to connect two air conditioning ducts, A and C so that the area of sheet metal used is a minimum. A is an existing shaft of cross-sectional dimensions 5 x 5 units. The size of C is to be decided from the design of the transition section, but it is known that two sizes of duct are available i.e. 3 x 1 and 2 x 1 where the first value is the vertical dimension. It is assumed that the costs of ducts A and C do not influence the cost of the transition section. In addition, a diaphragm is required at B which is manufactured in varying rectangular or square shapes ranging from 5 x 5 to 1 x 1 units. The spacing ab and bc is 2 and 3 units respectively.

C1.2 SELECTION OF STATE VARIABLES

The transition section can be divided into two substructures. Substructure 1 is defined between stage k = 0 and stage k = 1 and substructure 2 between stage k = 1 and k = 2 (Figure C2).

Let the state variables $x_1(k)$, $x_2(k)$; $x_1(k-1)$, $x_2(k-1)$ define the possible dimensions of the right and left hand ends of substructure k in the coordinate directions p and q respectively. The values of the state variables are therefore:

k = 0		k = 1		k = 2	
$x_1(0)$	$x_2(0)$	$x_1(1)$	$x_2(1)$	$x_1(2)$	$x_2(2)$
5,0	5,0	1,0	1,0	2,0	1,0
		2,0	2,0	3,0	1,0
		3,0	3,0		
		4,0	4,0		
		5,0	5,0		

TABLE C1

STATE VARIABLES x(k)

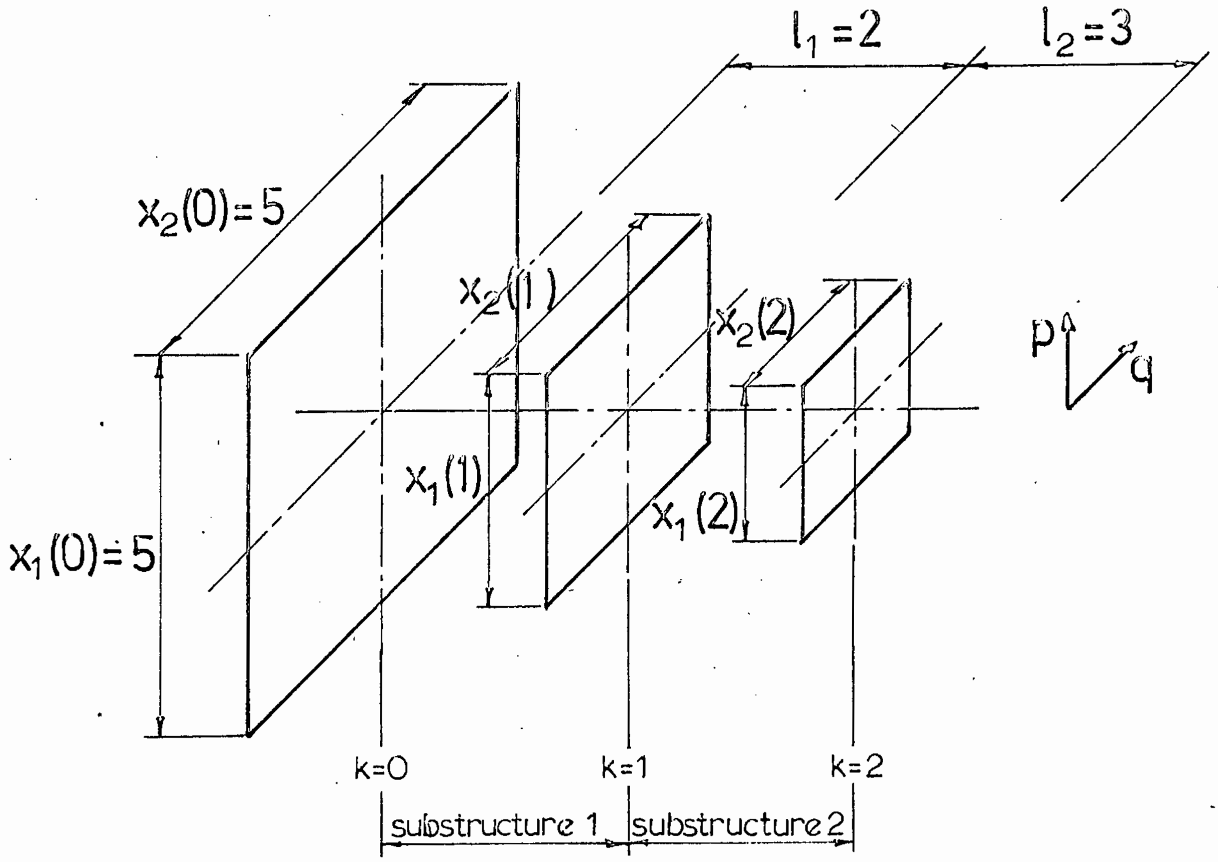


Figure C2 State Variables for Transition Section

C1.3 PERFORMANCE CRITERION

A minimum area of sheet metal is required, hence:

$$J = \min \sum_{k=0}^{k=2} [\text{surface area of substructure } k]$$

$$J = \min \sum_{k=0}^{k=2} [SA(k)] \qquad \text{(c.1)}$$

The surface area of substructure k can be defined in terms of state variables $x_1(k)$, $x_2(k)$, $x_1(k-1)$, $x_2(k-1)$ as follows:

$$SA^{(k)} = [x_1(k-1) + x_1(k)]\sqrt{\left(\frac{x_2(k-1) - x_2(k)}{2}\right)^2 + \ell m^2}$$

$$+ [x_2(k-1) + x_2(k)]\sqrt{\left(\frac{x_1(k-1) - x_1(k)}{2}\right)^2 + \ell m^2} \qquad (C.2)$$

and by definition SA(0) = 0,0

C1.4 INITIAL SOLUTION

In order to begin the calculation, an initial solution is required. For this purpose, let Table C2 represent the initial solution. The total area of the transition section is 86,96 units^2.

k	x_1	x_2
0	5,0	5,0
1	5,0	5,0
2	2,0	1,0

TABLE C2

INITIAL SOLUTION

C1.5 OPTIMIZATION CALCULATIONS FOR STATE VARIABLE x_1

The DPSA technique is commenced by keeping the state variable x_2 constant at the values specified in Table C2. A single dimensional dynamic programming procedure is performed with respect to state variable x_1 using the possible values in Table C3.

k / i	0	1	2
5	5,0	5,0	⊠
4	⊠	4,0	⊠
3	⊠	3,0	3,0
2	⊠	2,0	2,0
1	⊠	1,0	⊠

TABLE C3

VALUES OF x_1

In Table C3 the symbol ⊠ denotes a position which is not to be considered in the solution.

1. $k = 0$

Control begins with $k = 0$, where $x_1(0) = 5$ is the only position to be considered, i.e. it is a fixed boundary state (cf equations (2.4) and (2.9)). By definition $SA(0) = 0,0$ and hence $J = 0$.

2. $k = 1$

Control moves to stage $k = 1$ where 5 possible positions of $x_1(1)$ exist (Table C3). The value of J for each must be calculated:

$$J = \min_{i=0}^{i=5} \left[\sum_{k=0}^{k=1} (SA^i(k)) \right] \quad \text{from equation (C.1)}$$

$$J = \min_{i=0}^{i=5} \left[SA^i(1) + SA(0) \right] \quad \text{for all values of } x(k) \text{ where } k = 1$$

Since $SA(0) = 0,0$

$$\therefore \quad J = \min_{i=0}^{i=5} \left[SA^i(1) \right]$$

Example: for $i = 5$; $k = 1$

The values of the state variables are:

$$\left.\begin{array}{l} x_1(k) = 5 \\ x_1(k-1) = 5 \end{array}\right\} \text{Table C3} \qquad \left.\begin{array}{l} x_2(k) = 5 \\ x_2(k-1) = 5 \end{array}\right\} \text{From initial solution}$$

and
$$SA^5(1) = (5 + 5)\sqrt{\left(\frac{5 - 5}{2}\right)^2 + 2^2}$$

$$+ (5 + 5)\sqrt{\left(\frac{5 - 5}{2}\right)^2 + 2^2}$$

$$= 40 \text{ units}^2$$

A diagram can be used to express the above calculations (Figure C3).

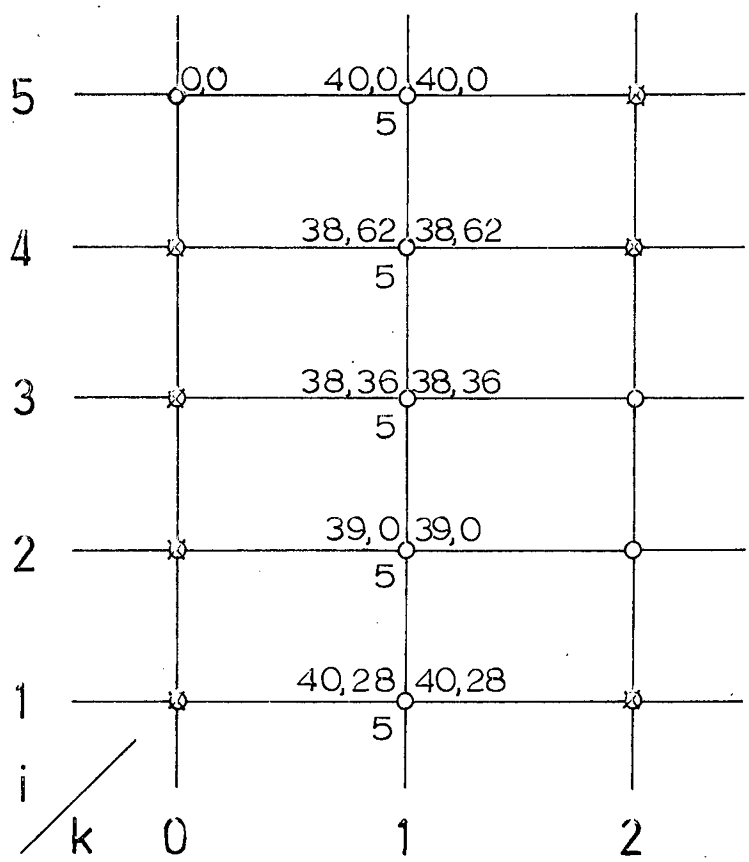
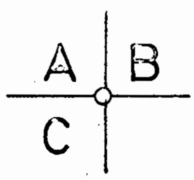


Figure C3 State Variable x_1 for Stage $k = 1$

The following legend is applicable to Figure C3 and the subsequent Figure of this type:



where

A = area of a single substructure for k,k-1

B = accumulated area of all substructures up to stage k

C = position in stage k-1 which gives a minimum value of J

3. k = 2

It remains to complete the process to the k = 2 stage.

From equation (C.1)

$$J = \min_{i=1}^{i=5} \left[\sum_{k=0}^{k=2} [SA^i(k)] \right]$$

$$J = \min_{i=1}^{i=5} \left[SA^i(2) + \min \sum_{k=0}^{k=1} [SA^i(k)] \right]$$

The second part of the expression is known from stage k = 1. This is accomplished by finding $J = \min_{i=1}^{i=5} \left[Area^i(x_1(1); x_1(2)) + \sum_{k=0}^{k=1} SA^i(k) \right]$

for the existing values of $x_1(2)$

i.e. $x_1(2) = 3,0$

$x_1(2) = 2,0$

for $x_1(2) = 3,0$

$$J = \min \begin{cases} \text{Area } (5,0;\ 3,0) + 40,0 = 87,82 \\ \text{Area } (4,0;\ 3,0) + 38,62 = 82,11 \\ \text{Area } (3,0;\ 3,0) + 38,36 = 77,99 \\ \text{Area } (2,0;\ 3,0) + 39,0 = 75,28 \\ \text{Area } (1,0;\ 3,0) + 40,28 = 73,68 \end{cases}$$

and for $x_1(2) = 2,0$

$$J = \min \begin{cases} \text{Area } (5,0;\ 2,0) + 40,0 = 85,36 \\ \text{Area } (4,0;\ 2,0) + 38,62 = 79,23 \\ \text{Area } (3,0;\ 2,0) + 38,36 = 74,64 \\ \text{Area } (2,0;\ 2,0) + 39,0 = 71,42 \\ \text{Area } (1,0;\ 2,0) + 40,28 = 69,34 \end{cases}$$

where Area $(x_1(1);\ x_1(2)) = SA(k)$ equation C.2

In diagrammatical form:

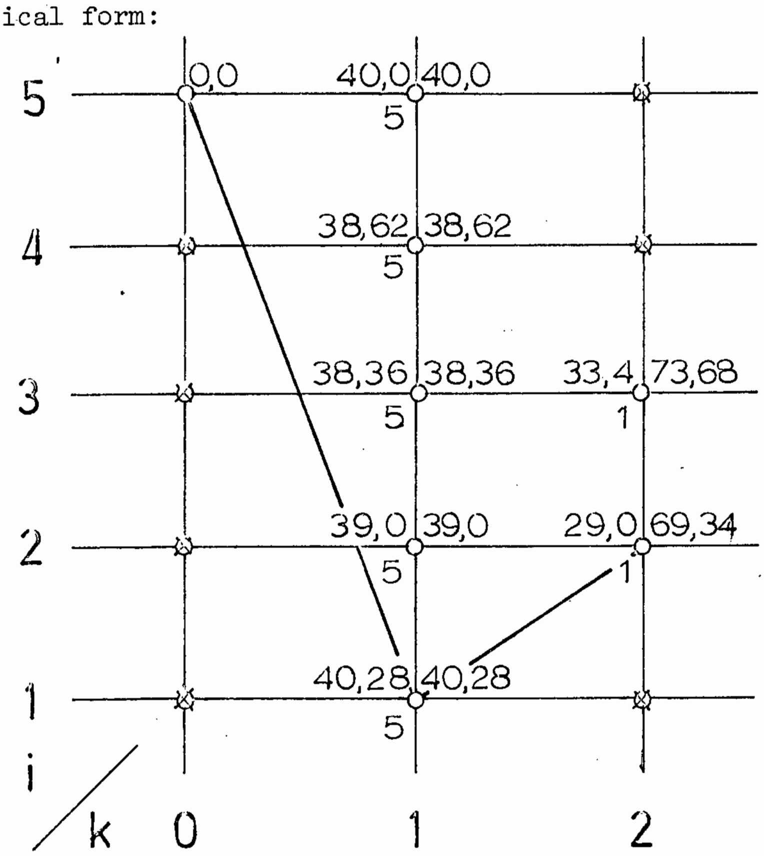


Figure C4 Completed Route for x_1 - Cycle 1

From the diagram, 69,34 is less than 73,68 at k = 2, hence the least material area can be accomplished by using the dimensions (cf Figure C1, and C2)

at	a	$x_1 = 5,0$	$x_2 = 5,0$	initial solution of x_2
	b	$x_1 = 1,0$	$x_2 = 5,0$	initial solution of x_2
	c	$x_1 = 2,0$	$x_2 = 1,0$	initial solution of x_2

with a total area of 69,34 units^2.

C1.6 OPTIMIZATION CALCULATIONS FOR STATE VARIABLE x_2

State variable x_2 is now released. x_1 is held constant at the values arrived at above.

The calculation proceeds precisely as was the case for varying x_1. The arithmetic produces, in diagrammatical form, the following result:

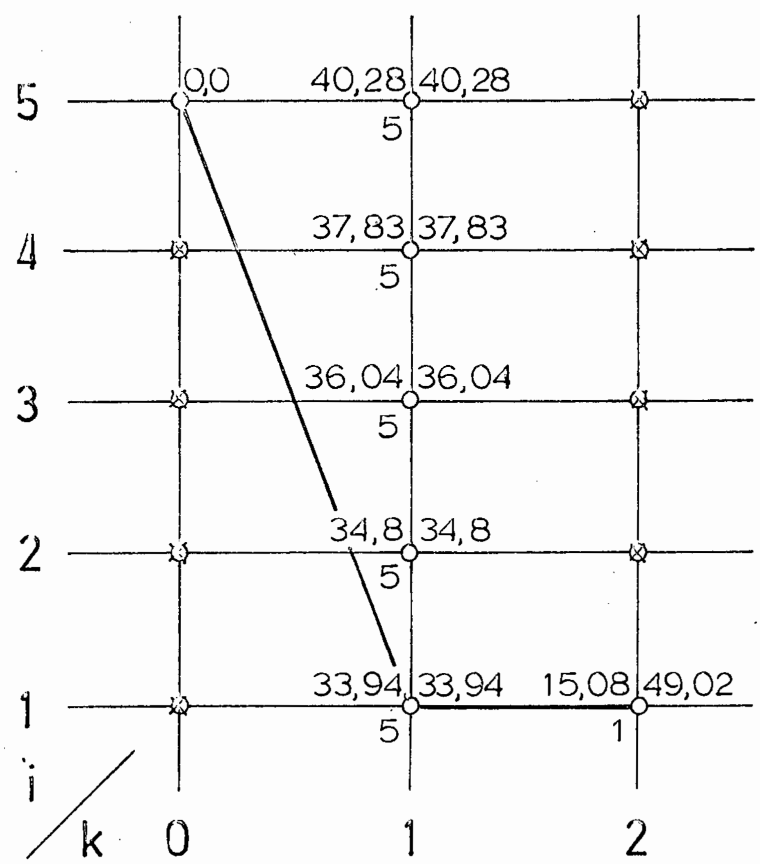


Figure C5 Completed Route for x_2 - Cycle 1

The follwoing values of x_1 and x_2 are produced by this optimization

at	a	$x_1 = 5,0$ from iteration 1	$x_2 = 5,0$ from iteration 2
	b	$x_1 = 1,0$ " " "	$x_2 = 1,0$ " " "
	c	$x_1 = 2,0$ " " "	$x_2 = 1,0$ " " "

With a total area of 49,02 units^2.

The calculation are now performed cyclically with alternate state variables being released. The results produced are:

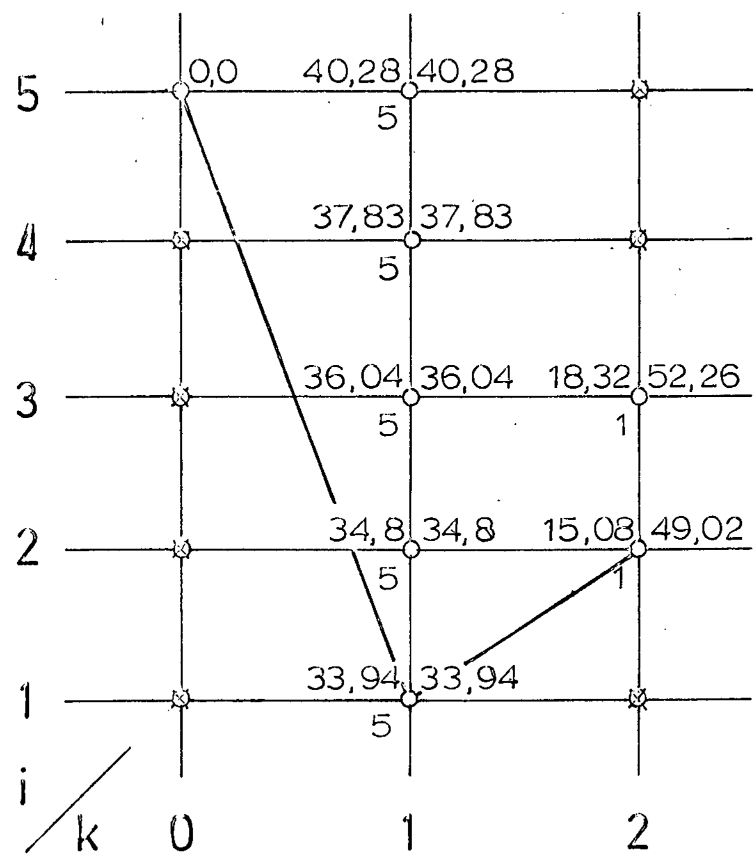


Figure C6 Completed Route for x_1 - Cycle 2

C1.7 FINAL SOLUTION

The shape of Figures C6 and C4 are identical. Therefore the optimum policy for both x_1 and x_2 has been found since there is no improvement in the total surface area between cycles. The final solution can be traced.

The control sequence represented by the values of i which give the optimum configuration are:

k	0	1	2
u_1	5	1	2
u_2	5	1	1

TABLE C3

CONTROL SEQUENCE u(k)

The state variable configuration is:

k	0	1	2
x_1	5,0	1,0	2,0
x_2	5,0	1,0	1,0

TABLE C4

STATE VARIABLE x(k)

The final solution, for which the area is 49,02 units^2 is drawn in Figure C7.

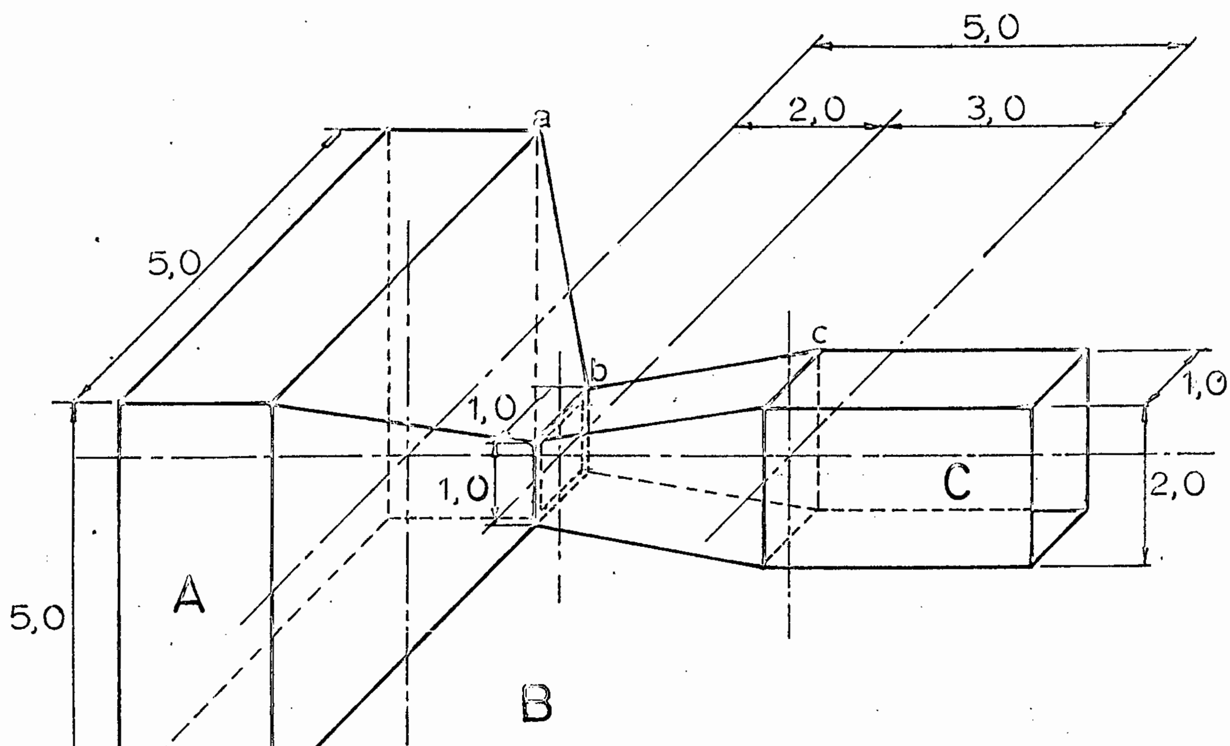


Figure C7 Final Dimensions of Transition Section

A P P E N D I X   D

D1.1 THE DISPLACEMENT METHOD FOR BAR ELEMENTS, AN ENERGY APPROACH

The total potential energy of a structure is

$$\pi = U + W_p$$

where U is the strain energy and W_p is the potential of the applied loads. Since the forces are assumed to remain constants during a variation of the displacements, the relation between the work done by the loads W and of the potential of the loads is

$$\delta W = -\delta W_p.$$

The principle of minimum potential energy then is

$$\delta\pi = \delta U + \delta W_p = \delta U - \delta W = 0$$

The potential energy for a linear elastic body is expressed in terms of the internal work and the potential of the body forces and surface tractions

$$\pi = \int_V dU(u,v,w) - \int_V (\bar{X}u + \bar{Y}v + \bar{Z}w)dV - \iint_{S_i} (\bar{T}_x u + \bar{T}_y v + \bar{T}_z w)dS_i$$

where U = internal energy

$\int_V dV$ = body force

$\int_{S_i} dS_i$ = surface forces over the surface S_i

For a bar structure, the integral can be substituted by a summation, since a bar in the structure contributes a finite definable amount of internal energy to the structure. Hence

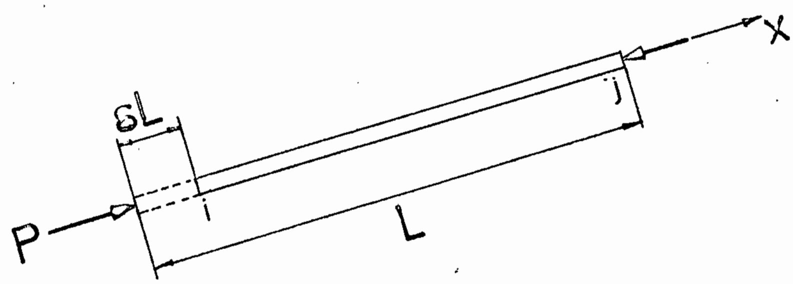
$$\int_V dU(u,v,w) = \sum_{i=1}^{m} U_i \qquad \text{where } U_i = \text{internal energy for bar i}$$

while the potential of the body and surface forces are

$$\int_V (\overline{X}u + \overline{Y}v + \overline{Z}w)dV + \int_{S_i} (\overline{T}_x u + \overline{T}_y v + \overline{T}_z w)dS_i$$

$$= \sum_{i=1}^{m} (\overline{X}_i u_i + \overline{Y}_i v_i + \overline{Z}_i w_i) + \sum_{i=1}^{m} (\overline{T}_{xi} u_i + \overline{T}_{yi} v_i + \overline{T}_{zi} w_i)$$

Consider a bar element in local line coordinates



The internal energy in the bar is the force times the displacement

hence $U_i = P\delta L$

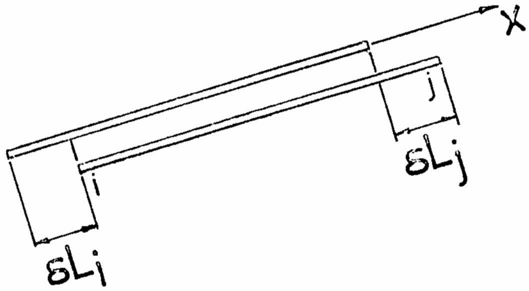
now from Hooke's law $\sigma = \frac{P}{A} = E\varepsilon$

and $\varepsilon = \frac{\delta L}{L}$

$\therefore \quad P = AE\frac{\delta L}{L}$

hence $U_i = \frac{AE}{L}\,\delta L\delta L$

Consider now a bar in local line coordinate subjected to axial force.



Therefore the change in length of the member -

$$\delta L = \delta L_i - \delta L_j$$

$$\therefore \quad \delta L = [1 \; -1] \begin{bmatrix} \delta L_i \\ \delta L_j \end{bmatrix}$$

$$\therefore \quad U_i = \frac{AE}{L} [\delta L_i \;\; \delta L_j] \begin{bmatrix} 1 \\ 1 \end{bmatrix} [1 \; -1] \begin{bmatrix} \delta L_i \\ \delta L_j \end{bmatrix}$$

$$= \frac{AE}{L} [\delta L_i \;\; \delta L_j] \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta L_i \\ \delta L_j \end{bmatrix}$$

From the definition of the transformation of coordinate axes from local to the global set

$$\delta L_i = u_i\lambda + v_i\mu + w_i\upsilon$$

and $$\delta L_j = u_j\lambda + v_j\mu + w_j\upsilon$$

$$\therefore \quad \delta L = u_i\lambda + v_i\mu + w_i\upsilon - u_j\lambda - u_j\mu - w_j\upsilon$$

where $$\lambda = \frac{X_B - X_A}{L}$$

$$\mu = \frac{Y_B - Y_A}{L}$$

$$\upsilon = \frac{Z_B - Z_A}{L}$$

In matrix form

$$\delta L = [\lambda\ \mu\ \upsilon\ -\lambda\ -\mu\ -\upsilon] \begin{bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{bmatrix}$$

and $$\overset{U_1}{\cancel{V_i}} = \frac{AE}{L} [\delta L]^T [\delta L]$$

$$\therefore \quad \overset{U_1}{\cancel{V_i}} = \frac{AE}{L} [u_i\ v_i\ w_i\ u_j\ v_j\ w_j] \begin{bmatrix} \lambda \\ \mu \\ \upsilon \\ -\lambda \\ -\mu \\ -\upsilon \end{bmatrix} [\lambda\ \mu\ \upsilon\ -\lambda\ -\mu\ -\upsilon] \begin{bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{bmatrix}$$

or $$\overset{U_1}{\cancel{V_i}} = \frac{AE}{L} \{u\}^T [k_e] \{u\}$$

and

$$k_e = \begin{bmatrix} \lambda \\ \mu \\ \upsilon \\ -\lambda \\ -\mu \\ -\upsilon \end{bmatrix} [\lambda\ \mu\ \upsilon\ -\lambda\ -\mu\ -\upsilon]$$

Returning to the total potential energy of the structure

$$\pi = \sum_{i=1}^{m} \frac{AE}{L} \{u\}^T [k_e] \{u\} - \sum_{i=1}^{m} \{u\}^T B_i - \sum_{i=1}^{m} \{u\}^T S_i$$

where B_i is the matrix of Body forces for bar i

S_i is the matrix of Surface forces for bar i

Applying the variational principle

$$\{\delta u\}^T \left[\sum_{i=1}^{m} \frac{AE}{L} [k_e]\{u\} - \sum_{i=1}^{m} B_i - \sum_{i=1}^{m} S_i \right] = 0$$

Now since the variation of displacements is arbitrary, the expression in brackets must disappear

$$\therefore \quad \left[\sum_{i=1}^{m} \frac{AE}{L} [k_e] \right] \{u\} = \sum_{i=1}^{m} B_i + \sum_{i=1}^{m} S_i$$

or more generally

$$[K]\{u\} = \{P\}$$

where {P} is the summation of {B} and {S} and [K] is the system stiffness matrix.

APPENDIX E

E1.1 CROUT REDUCTION FOR THE SOLUTION OF LINEAR SIMULTANEOUS EQUATIONS

Consider a system of linear equations to be solved:

$$\{A\}\underline{x} = \underline{b}$$

By Gauss elimination, the algorithm is

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}\, a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$$

for $i = k+1,\ldots,N.$

$j = k,\ldots,N.$

where N is the number of equations

with back substitution

$$b_{i}^{(k)} = b_{i}^{(k-1)} - \frac{a_{ik}^{(k-1)}\, b_{k}^{(k-1)}}{a_{kk}^{(k-1)}}$$

for $i = k+1,\ldots,N$

where i is the row number and

j is the column number in the matrix A

Notable points are:

1. Symmetry has not been taken into account.
2. a_{ik}/a_{kk} is a constant operator while j runs from k to N, and hence can be used as a common factor both in the forward elimination as well as in the back substitution.
3. Where $a_{ik} = a_{kj} = 0$, there is no change in a_{ij} which should be recognized.

The Crout Reduction Process can be defined by its essential features:

1. Re-order the sequence in which the terms of the coefficient matrix A are modified
2. Each term is changed directly from its initial value in the unreduced matrix to its final value in the upper triangular reduced matrix.

The procedure is a direct consequence of the fact that in the Gauss Elimination process, row i of the triangular matrix is obtained by a linear combination of rows 1 to i-1. The Crout algorithm is then:

$$a_{ij}^{(i-1)} = a_{ij} - \sum_{k=1}^{i-1} \frac{a_{ki}^{(k-1)} a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$$

for $j = 2,\ldots,N$

$i = 2,\ldots,j$

and the back substitution

$$b_{ij}^{(i-1)} = b_i - \sum_{k=1}^{i-1} \frac{a_{ki}^{(k-1)} b_k^{(k-1)}}{a_{kk}^{(k-1)}}$$

APPENDIX F

F1.1 MATRIX STORAGE TECHNIQUE

All stiffness matrices of the type required by the Displacement Method are symmetric positive definite; and if the structural numbering system is carefully considered, it is also well bonded. Typically, the matrix A is as follows (Figure F1).

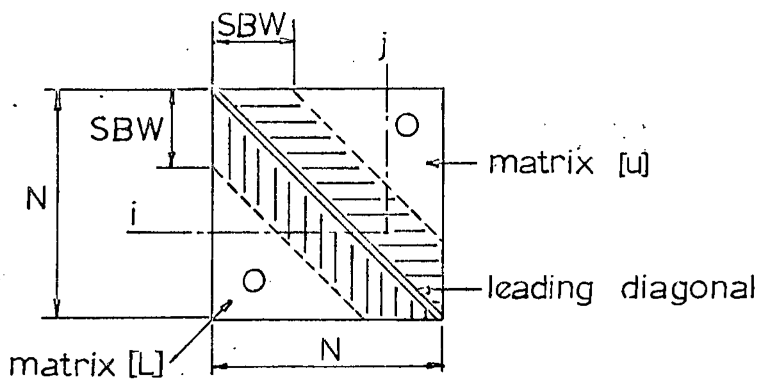


Figure F1 Square Matrix A

For computer storage, it is very wasteful and often impossible to store the whole N x N matrix when only a part of it contains relevant information. Due to symmetry, the matrix is divided into three sections.

(a) the leading diagonal

(b) the upper triangular matrix [u]

(c) the lower triangular matrix [L]

Since [u] is merely the mirror image of [L] only one need be stored (Figure F2).

Now the banded form of the matrix requires that only a portion of [u] be stored, namely SBW x N and this can be condensed into a rectangular matrix as shown in Figure F3.

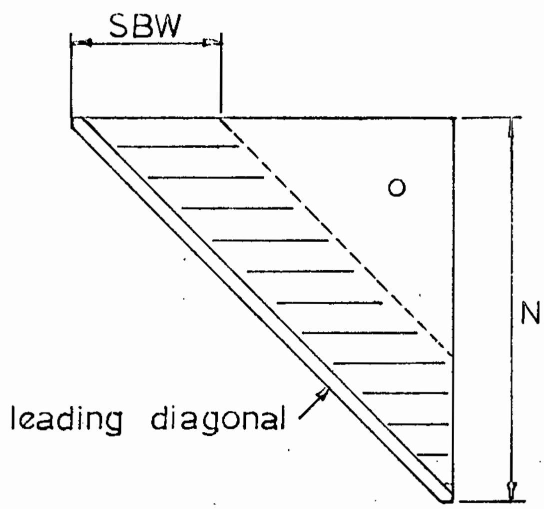


Figure F2 Upper Triangular Matrix Form of A

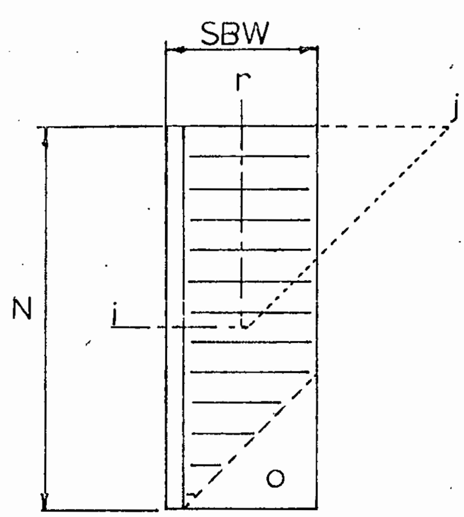


Figure F3 Rectangular Matrix Form of A

where SBW is the semi-band width of the matrix.

Let i and j be the row and column numbers respectively in the square matrix A. The equivalents in the rectangualr matrix are

Square Matrix		Rectangular Matrix	
row number	= i	row number	= i
column number	= i	column number	= $r = j - i + 1$

Problems which are solved by the displacement method require differing sizes of stiffness matrix A. This is dependent on the number of nodes in the structure. An efficient method of storage on the UNIVAC 1106 for varying sized problems is to specify a vector of length SBW x N, where SBW and N are ~~ungiue~~ Unique for different problems. This means that the rectangular matrix (Figure F3) is stored as a single array (Figure F4).

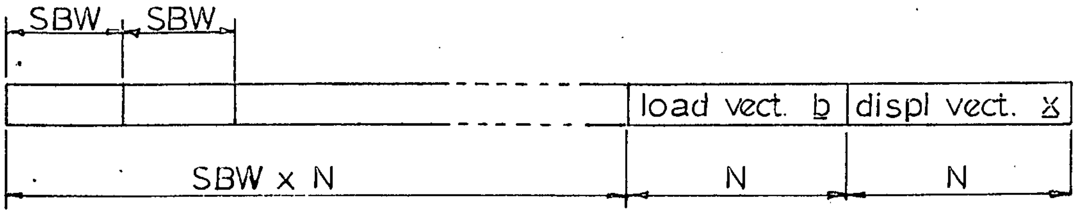


Figure F4 Single Array Matrix A

In addition, a load vector b and displacement vector x, both of length N, are added to the array. The total length is then (SBW + 2) x N.

A UNIVAC systems subroutine is available to change the size of this array depending on the problem.

This technique can be used to store all the variables required, provided they are contained in a single array.

A P P E N D I X   G

G1.1 TEMPERATURE ANALYSIS FOR A BAR STRUCTURE

Method of Analysis

State 1

Prior to the change of temperature, let the members be disconnected from one another. A difference in temperature ΔT will cause the member to change length by δt. The thermal strain Δ_T is:

$$\varepsilon_T = \frac{\delta t}{L} = \frac{\alpha \Delta T L}{L} = \alpha \Delta T$$

where

α is the coefficient of thermal expansion

L is the length of the member

In order to reassemble the structure, a force must be applied in order to return it to its original length. Let this force be
$P^{(1)} = EA\varepsilon_T$

Due to the extension the bar will produce an equal and opposite force on the nodes affected by the bar, i.e. State 2.

State 2

By applying $P^{(2)} = -P^{(1)}$ at the nodes, it is equivalent to applying the force to the whole structure. Consequently differential forces are set up in the various members. i.e. actual force in the member $= F^{(2)}$.

The externally applied forces $P^{(1)}$, $P^{(2)}$ cancel each other, thus leaving the heated structure with no external loading.

State 2 causes the nodal displacements. These define the deformed configuration of the heated structure, while the final self-equilibrating force is the sum of the forces from each state.

i.e. $$F^{*} = P^{(1)} + F^{(2)}$$

which is used to calculate the member stresses.

A P P E N D I X   H

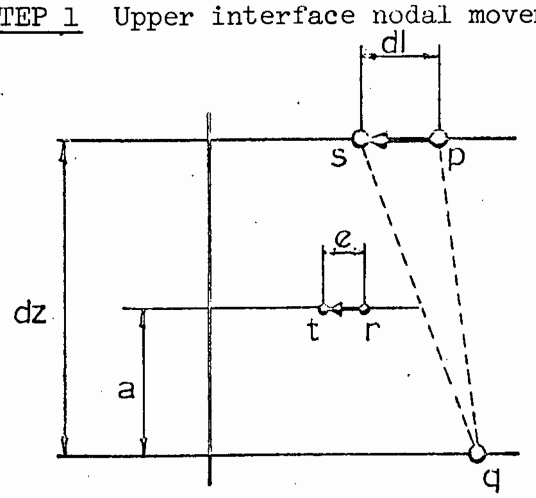
H1.1   NODAL COORDINATE REPOSITIONING

When the substructure interface dimensions are altered in the DPSA technique, the position of a general node in the substructure is also altered. This is done by simple proportion by relating the movement of the interface nodes with the equivalent movement at the general node. The formulae which govern these movements are:

1. For rectangular plan or plane truss towers:

   (a) Dimension change in either the x- or y-direction:

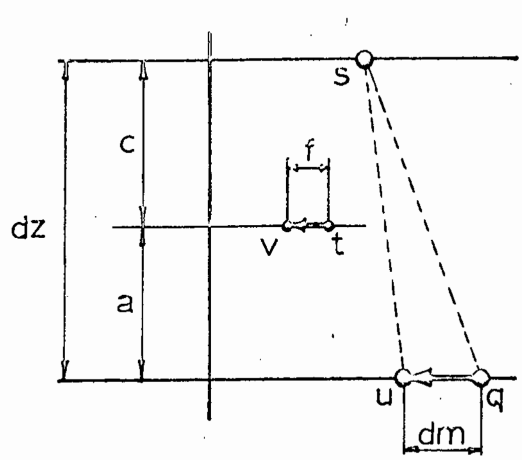
STEP 1   Upper interface nodal movement in x- or y-direction



$$e = \frac{(a \times d\ell \times r)}{(dz*p) + (q - p)(dz - a)}$$

$$t = r + e$$

STEP 2   Lower interface nodal movement in x- or y-direction

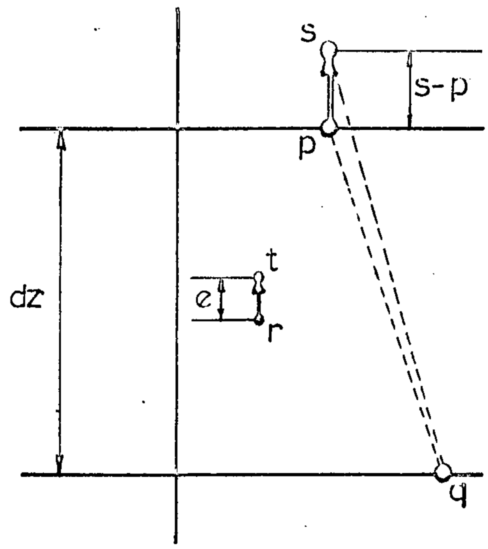


$$f = \frac{(c \times dm \times t)}{(dz \times s) + (c*(q - s))}$$

$$u = t + f$$

(b) Elevation change in z-direction

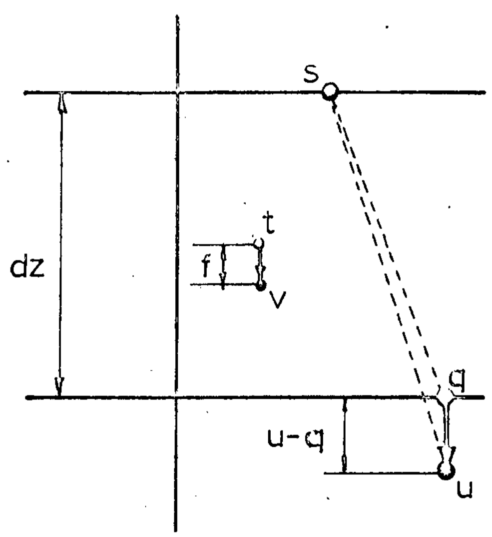
<u>STEP 1</u> Upper interface nodal movement in z-direction



$$e = \frac{(s - p)*(r - q)}{dz}$$

$$t = r + e$$

<u>STEP 2</u> Lower interface nodal movement in z-direction



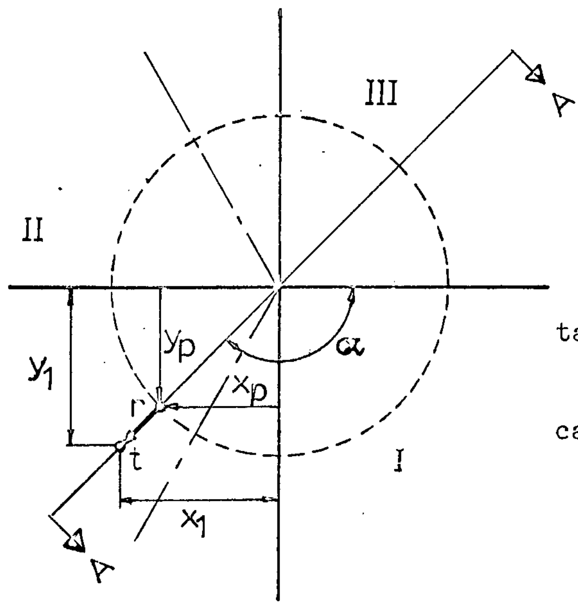
$$f = \frac{(u - q)*(s - t)}{dz}$$

$$v = t + f$$

2. For triangular plan towers;

(a) Radius change:

STEP 1. Upper interface nodal movement - radius increase



Calculate angle from

$$\alpha = \arctan \frac{y_p}{x_p}$$

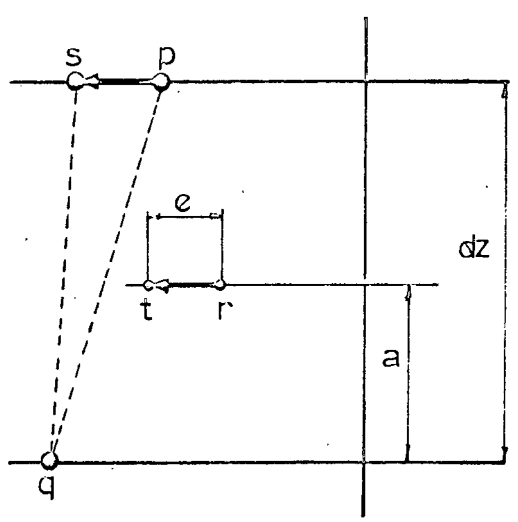
$$\tan 30^\circ = 0{,}57735$$

$$\text{calculate } ru = \frac{p \times 0{,}57735}{(\sin\alpha + 0{,}57735 \cos\alpha)}$$

$$r\ell = \frac{ru \times q}{p}$$

$$\text{then } m = \frac{(s - p)ru}{p}$$

$$e = \frac{ma}{dz}$$



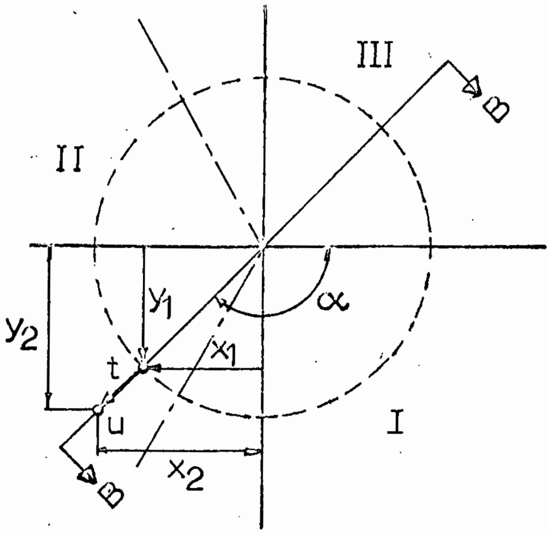
and new coordinates are

$$x_1 = x_p + e \cos\alpha$$

$$y_1 = y_p + e \sin\alpha$$

Section A-A

STEP 2 Lower interface nodal movement - radius increase



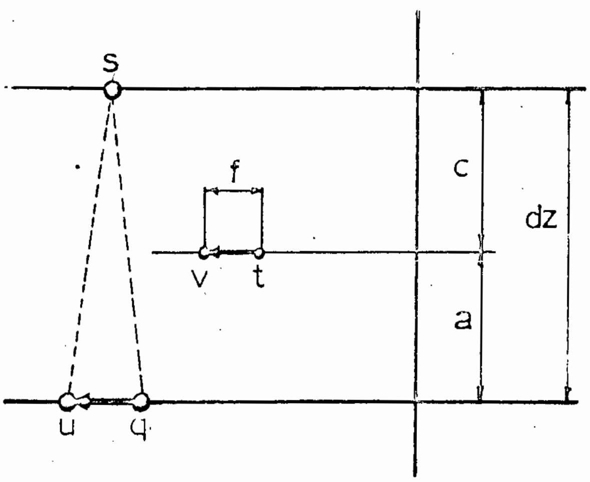
$$n = \frac{(u - q) \times r\ell}{q}$$

$$f = \frac{n \times c}{q}$$

and final coordinates are

$$x_2 = x_1 + f \cos\alpha$$

$$y_2 = y_1 + f \sin\alpha$$



Section B-B

(b) Elevation change:

Same as for 1.(b) above.

Notes: The signs of the above formulae are correct if the equation is kept exactly as shwon, i.e. the formulae are applicable to both positive and negative dimension changes for either interface.

APPENDIX I

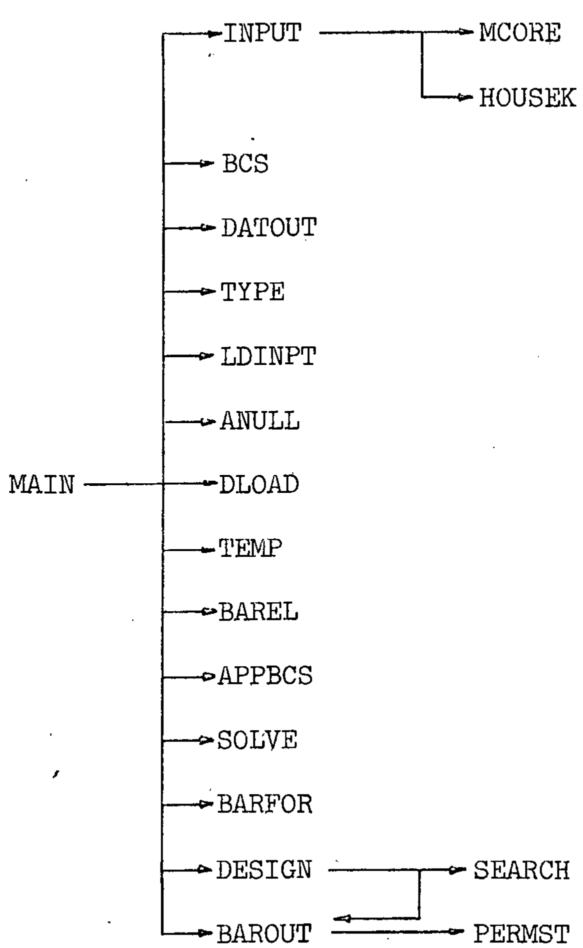
USER'S MANUAL: PROGRAM DYSPAN

I1.1 PROGRAM NAME: CONTRA*DYSPAN

I1.2 DESCRIPTION

Program for the design of space structures with a specified geometry. A direct iteration procedure is involved whereby the efficiency of a member is calculated by considering the ratio between its actual and permissible stresses. Iterations are performed until this ratio is as close to unity as possible.

I1.3 SUBROUTINE LAYOUT



II.4 SUBROUTINE FUNCTIONS

1.	MAIN program	- coordination of calculation steps - input of title and choice of either interactive or batch mode data entry - output of results
2.	Subroutine INPUT	- input of member and nodal data (coordinates, member incidence, member group number)
3.	Subroutine MCORE	- UNIVAC systems routine for the dynamic core storage of a single array (AB)
4.	Subroutine HOUSEK	- housekeeping subroutine for the control of the data stored in array AB
5.	Subroutine BCS	- input of support constraints
6.	Subroutine DATOUT	- output of data, if required
7.	Subroutine TYPE	- input of section classes and effective length coefficients, Young's Moduli, permissible slenderness ratios, yield stress
8.	Subroutine LDINPT	- input of loading data, combination load cases, selfweight and temperature loading
9.	Subroutine ANULL	- initialization of the system stiffness matrix
10.	Subroutine DLOAD	- calculation of the self-weight of the structure
11.	Subroutine TEMP	- calculation of the loading due to temperature changes
12.	Subroutine BAREL	- calculation of the member stiffness matrices and the assembly of the system stiffness matrix
13.	Subroutine APPBCS	- application of the support constraints to the stiffness matrix

14. Subrotuine SOLVE - solution of the simultaneous equations of the stiffness matrix by the Crout Reduction Method

15. Subroutine BARFOR - calculation of member forces from differential nodal displacements

16. Subroutine DESIGN - checks the design of current design variables and predicts 'improved' values

17. Subroutine SEARCH - searches through the list of sections to find a suitable section for the predicted value from DESIGN

18. Subroutine BAROUT - output of results

19. Subroutine PERMST - calculation of permissible stresses and slenderness ratio for output

I1.5 PROGRAM LIMITATION

The following limitations are set on the size of the structure which can be designed with the program DYSPAN:

Number of member groups in the structure

NTYPE = 20

Number of restrainted degrees of freedom (i.e. support reactions)

NRS = 50

Number of load cases

NLC = 5

Number of combination load cases

NLC - 2 = 3

The program has been written with dynamic core storage allocation facilities. Hence, the program sets up enough storage to accommodate the problem as it is needed. An upper limit on the number of nodes and members of a structure can therefore not be fixed categorically, but depends entirely on the structures configuration. It should be noted, however, that the maximum in-core storage capacity of our UNIVAC 1106 is 65 K (65000 words) without using an extended storage facility.

I1.6 STORAGE ARRANGEMENTS

One major array AB is used to store all the relevant information required within the problem. This array is dynamically stored in the following form:

	Array Name	Array Dimensions	Total length
(a)	Stiffness matrix	S(NDOF,ND)	NDOF x ND
(b)	Current load vector	PLD(NDOF)	NDOF
(c)	Load case vectors	5*DPL(NDOF)	5 x NDOF
(d)	Dead load vector	DVL(NDOF)	NDOF
(e)	Displacement vector	U(NDOF)	NDOF
(f)	Coordinate vectors X,Y,Z	X(NNP) Y(NNP) Z(NNP)	NDOF
(g)	First node incidence vector	NPI(NEL)	NEL
(h)	Second node incidence vector	NPJ(NEL)	NEL
(i)	Member group number vector	ITYPE(NEL)	NEL
(j)	Force vector	P(NEL)	NEL

Total length of array AB = NDOF(ND + 9) + 4*NEL

AB is set up as a single array together with the housekeeping array IH as follows:

Point Load Case

S matrix	PLD	DPL vectors	DVL vect	U vect	X	Y	Z	cont
					NNP	NNP	NNP	
NDOF*ND	NDOF	5*NDOF	NDOF	NDOF		NDOF		

Housekeeping array IH (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12)

	NPI	NPJ	ITYPE	P
	NEL	NEL	NEL	NEL

IH(12) (13) (14) (15)

I1.7 NOTES FOR DATA INPUT

(i) Data is in free format throughout, i.e. if more than a single record is required per card, they are separated by commas A,B,C,D

(ii) Data must be input as indicated with every variable being assigned a value, even if it is zero.

(iii) Units should be metric (S.I.) and consistent throughout. For example, if the coordinates are entered in metre units, and the loads in kilonewtons, then the remaining data must also be entered in kN,m units.

(iv) Control statements must be used strictly as indicated. Upper case characters are obligatory, while lower case characters indicate that the user is required to enter his personal information. The symbol 'b̸' indicates a blank character.

(v) For data input, a value is required for each upper case variable mentioned.

(vi) Variables which begin with the letters, I,J,K,L,M,N are integers and must be input as such: e.g. for variable NNP input a value 7 All other variables are real and must be input with a decimal point attached e.g. for variable x(I), input a value 2. or 2.0

I1.8 CONTROL STATEMENTS

Runstream used for batch mode jobs.

```
@RUNϕrunid,accountnumber/userid,CONTRA,time,pages.
@PASSWDϕpassword.
@ASG,AXϕDYSPAN.
@XQTϕDYSPAN.ABS
       data required
@FIN
```

For interactive mode (terminal) jobs, the following runstream must be used:

```
Terminal call number, e.g. U1106H.
userid/password.
@RUNϕrunid,accountnumber,CONTRA
@ASG,AXZϕDYSPAN.
@XQTϕDYSPAN.ABS
enter data as required by the questions asked by the program
@FIN
@@TERM.
```

I1.9 DATA PREPARATION

A list of the required data, in order is as follows:

```
TITLE

IDT

NNP,NEL,MAXDIF

F,X(I),Y(I),Z(I)                    } for I = 1,NNP
 :
 :
J,NPI(J),NPJ(J),ITYPE(J)            } for J = 1,NEL
 :
 :
NRS

NPB(N),NCOND(N)                     } for N = 1,NRS
 :
 :
NLC

IDSEC(K),EFFLEN(K)                  } for K = 1,NTYPE
 :
 :
E(1),E(2),E(3),E(4)

COMLR,HIGLR,PERA,YS

[1]      for data input for load cases 1 and 2 go to [2]

[2]      NPL

         K,LX(I),LY(I),LZ(I)        } for I = 1,NPL
          :
          :
[4]      IDL

         ITL

         if ITL = 1 enter DELTAT,ALPHA.

         return to [1] for input of following load case.

[3]      IA

         if IA = 0, go to [2]

         IB

         LC(1),LC(2),...,LC(IB)

         continue with data at [4]
```

Nomenclature Explanation

The variables in the above list are defined as follows:

TITLE	- an alphanumeric character string (not exceeding 60 characters) for the job title
IDT	- 1 - interactive mode data input 2 - batch mode data input (card image format)
NNP	- number of nodes in the structure
NEL	- number of members in the structure
MAXDIF	- maximum member nodal difference for calculation of the half band width ND = 3*(MAXDIF + 2)
I,X(I),Y(I),Z(I)	- global coordinates in x,y,z directions of node I
J,NPI(J),NPJ(J),ITYPE(J)	- member incidences and group number for member J
NPI(J),NPJ(J)	- node members assocaited with member J
ITYPE(J)	- group reference number associating similar member types in the structure (i.e. the same area and radius of gyration is to be found for all members with similar group reference numbers) total number of groups = NTYPE
NRS	- number of rigidly restrained degrees of freedom for the structure; each node has 3 degrees of freedom to be satisfied, i.e. maximum number of restraints per node is 3

NPB(N)	-	restrained node number
NCOND(N)	-	type of support constraint applied 1 for restraint in x-direction 2 for restraint in y-direction 3 for restraint in z-direction
NLC	-	number of load cases to be processed
IDSEC(K)	-	section class identification number for each group of members, K 1 for pipe sections 2 for angle sections 3 for double angle sections 4 for channel sections
EFFLEN(K)	-	effective length coefficient for member group K
E(1),...,E(4)	-	values of Young's Modulus for the 4 section classes above in the appropriate units
COMLR	-	permissible slenderness ratio for struts (BS 449 clause 33)
HIGLR	-	maximum permissible slenderness ratio for reversal of stress (BS 449 clause 44a)
PERA	-	permissible percentage cross sectional area for the purpose of tensile stress calculations
YS	-	minimum yield stress of the material used (BS 449 appendix B)
NPL	-	number of loaded nodes in the structure
I,LX(I),LY(I),LZ(I)	-	loads in the x,y,z directions respectively at node I

elf weight to be neglected for this
oad case
elf weight to be included for this
oad case
emperature loading to be neglected
or this load case
emperature loading to be included
or this load case
ıge of temperature in °C
ficient of linear expansion
bination load case indicator
ad case is <u>not</u> a combination of others
ad case is a combination of others
er of load cases to be combined
case numbers to be combined

ıs example in Figure I1.1 follows:

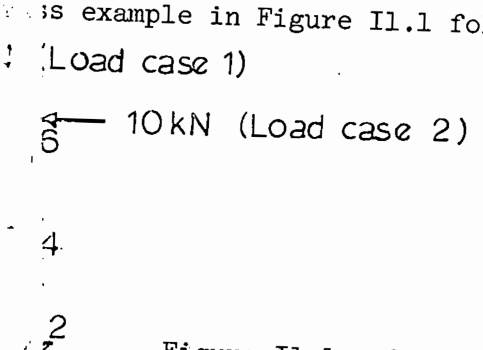


Figure I1.1 Sample Tower

(i) 15 kN at node 5

(ii) 10 kN at node 6

(iii) 15 kN at node 5, 10 kN at node 6 and self weight included

Sample Data Listing

```
CONTRA*DYSPAN(1).SAMPLE
SAMPLE PROBLEM FOR USERS MANUAL              TITLE
2                                            IDT
6,10,3                                       NNP,NEL,MAXDIF
1,-0.5,0.0,0.0
2,0.5,0.0,0.0
3,-0.375,0.0,1.0                             I,X(I),Y(I),Z(I)
4,0.375,0.0,1.0
5,-0.25,0.0,2.0
6,0.25,0.0,2.0
1,1,3,1
2,1,4,2
3,2,3,2
4,2,4,1
5,3,4,3                                      J,NPI(J),NPJ(J),ITYPE(J)
6,3,5,1
7,3,6,2
8,4,5,2
9,4,6,1
10,5,6,3
10                                           NRS
1,1
1,2
1,3
2,1
2,2                                          NPB(N),NCOND(N)
2,3
3,2
4,2
5,2
6,2
3                                            NLC
4,1,2,1,2,1                                  IDSEC(K),EFFLEN(K)
200000000.0,200000000.0,200000000.0,200000000.0   E(1)...E(4)
180.0,350.0,100.0,250.0                      COMLR,HIGLR,PERA,YS
1                                            NPL
5,0,0,-15.0                                  I,LX(I),LY(I),LZ(I)
0                    LOAD CASE 1             IDL
0                                            ITL
1                                            NPL
6,-10.0,0.0,0.0                              I,LX(I),LY(I),LZ(I)
0                    LOAD CASE 2             IDL
0                                            ITL
1                                            IA
2                                            IB
1,2                  LOAD CASE 3             LC(1),LC(2)
1                                            IDL
0                                            ITL
```

I1.11 SAMPLE OUTPUT

The following pages contain the sample output for the structure in Figure I1.1

LOAD CASE 1
****

AXIAL FORCES
****

ELEMENT NUMBER	FORCE (N)	STRESS N/M**2	PERMISSIBLE STRESS
1	-11.404	-13369.747	-89400.543
2	.088	375.051	155000.000
3	-1.573	-6692.867	-30935.787
4	-2.586	-3032.007	-89400.543
5	1.042	7337.519	155000.000
6	-12.883	-15103.128	-89400.543
7	.334	1422.551	155000.000
8	-2.614	-11122.639	-38644.012
9	-.286	-334.926	-89400.543
10	-.213	-1497.281	-73292.856

NODAL DISPLACEMENTS
****

NODE	X-DISP	Y-DISP	Z-DISP
1	.00000000	.00000000	.00000000
2	.00000000	.00000000	.00000000
3	-.00000881	.00000000	-.00006679
4	.00001871	.00000000	-.00001306
5	-.00005667	.00000000	-.00013751
6	-.00006041	.00000000	-.00002465

RECOMMENDED SECTIONS FOR THIS DESIGN
****

NOTE: THE FOLLOWING CODE IS USED TO INDICATE
SECTION CLASS
1 PIPE SECTION
2 ANGLE
3 DOUBLE ANGLE
4 CHANNEL

MEMBER TYPE REF. NUMBER	SECTION CLASS	SECTION SPECIFICATION NUMBER
1	4	1
2	2	3
3	2	1

LOAD CASE 2
****

AXIAL FORCES
****

ELEMENT NUMBER	FORCE (N)	STRESS N/M**2	PERMISSIBLE STRESS
1	-2.586	-3032.007	-89400.543
2	-1.573	-6692.867	-30935.787
3	.088	375.051	155000.000
4	-11.404	-13369.747	-89400.543
5	1.042	7337.519	155000.000
6	-.286	-334.926	-89400.543
7	-2.614	-11122.639	-38644.012
8	.334	1422.551	155000.000
9	-12.883	-15103.128	-89400.543
10	-.213	-1497.281	-73292.856

NODAL DISPLACEMENTS
****

NODE	X-DISP	Y-DISP	Z-DISP
1	.00000000	.00000000	.00000000
2	.00000000	.00000000	.00000000
3	-.00001871	.00000000	-.00001306
4	.00000881	.00000000	-.00006679
5	.00006041	.00000000	-.00002465
6	.00005667	.00000000	-.00013751

RECOMMENDED SECTIONS FOR THIS DESIGN
****

NOTE: THE FOLLOWING CODE IS USED TO INDICATE
SECTION CLASS
1 PIPE SECTION
2 ANGLE
3 DOUBLE ANGLE
4 CHANNEL

MEMBER TYPE REF. NUMBER	SECTION CLASS	SECTION SPECIFICATION NUMBER
1	4	1
2	2	3
3	2	1

LOAD CASE 3
****

THIS LOAD CASE IS A COMBINATION
OF LOAD CASES 1, 2,
DEAD LOADS ARE INCLUDED IN THIS LOAD CASE

AXIAL FORCES
****

ELEMENT NUMBER	FORCE (N)	STRESS N/M**2	PERMISSIBLE STRESS
1	-44.405	-52057.083	-89400.543
2	-8.206	-30848.289	-38631.719
3	-6.069	-22816.091	-38631.719
4	-55.748	-65354.960	-89400.543
5	4.638	32660.584	155000.000
6	-27.318	-32025.633	-89400.543
7	-5.209	-19581.864	-47952.667
8	-5.209	-19581.862	-47952.667
9	-27.318	-32025.633	-89400.543
10	-.628	-4420.553	-73292.856

NODAL DISPLACEMENTS
****

NODE	X-DISP	Y-DISP	Z-DISP
1	.00000000	.00000000	.00000000
2	.00000000	.00000000	.00000000
3	-.00006293	.00000000	-.00025649
4	.00005955	.00000000	-.00032444
5	.00009444	.00000000	-.00043879
6	.00008339	.00000000	-.00048409

RECOMMENDED SECTIONS FOR THIS DESIGN
****

NOTE: THE FOLLOWING CODE IS USED TO INDICATE
SECTION CLASS
1 PIPE SECTION
2 ANGLE
3 DOUBLE ANGLE
4 CHANNEL

MEMBER TYPE REF. NUMBER	SECTION CLASS	SECTION SPECIFICATION NUMBER
1	4	1
2	2	4
3	2	1

APPENDIX J

USER'S MANUAL: PROGRAM DYNGEO AND DYNPRE

J1.1 PROGRAM NAMES:

CONTRA*DYNGEO

CONTRA*DYNPRE

J1.2 DESCRIPTION

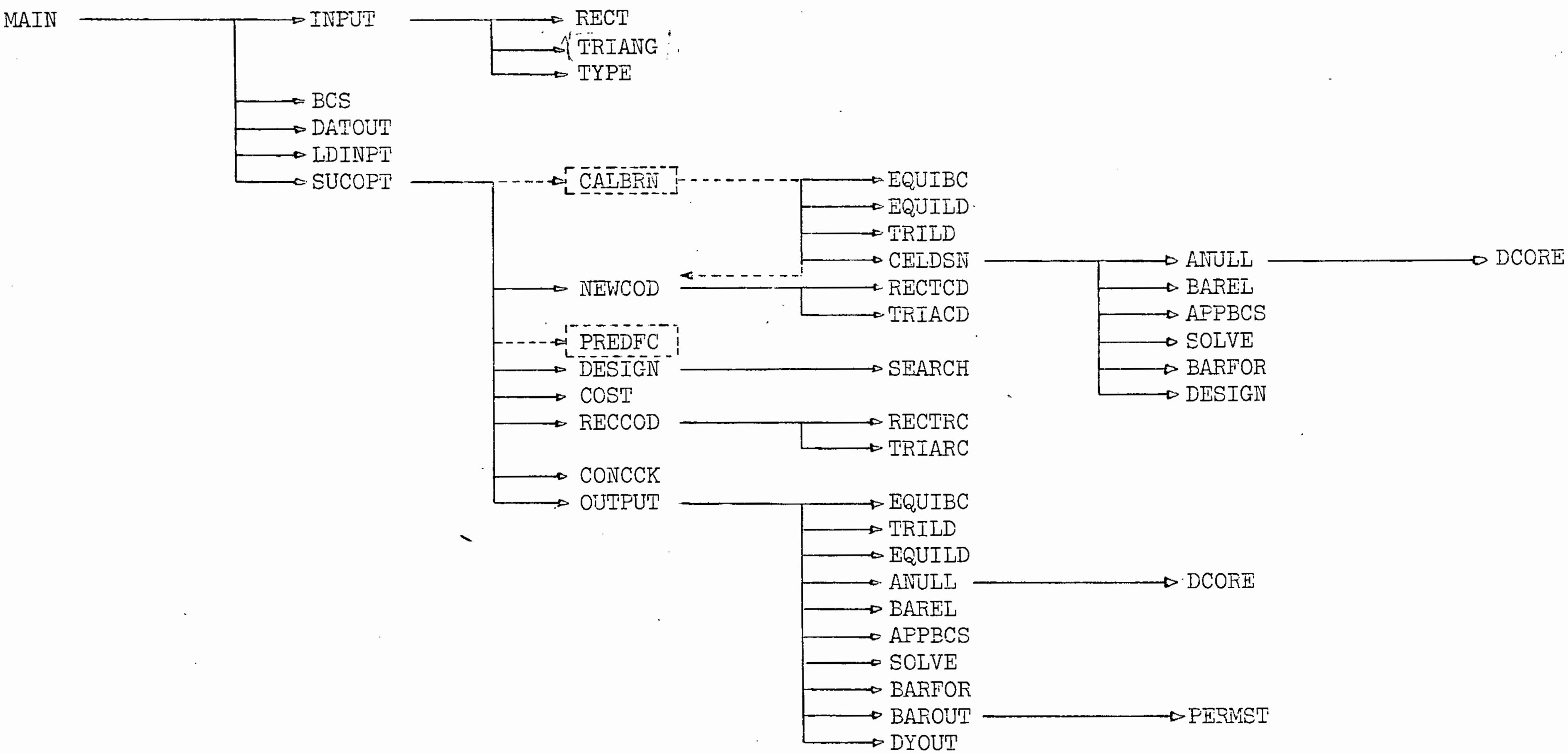
Programs for the structural and geometric configuration designs of tower type structures. A minimum weight design is found by considering all the possible configurations available by the application of Dynamic Programming. The member sizes are calculated by the Direct Iteration method.

J1.4 SUBROUTINE LAYOUT

The following subroutine layout is applicable to both DYNGEO and DYNPRE. Subroutines which are boxed are used only in DYNPRE. Hence, for the purposes of following the layout to DYNGEO, these subroutines should be deleted.

J1.5 SUBROUTINE FUNCTIONS

1.	MAIN program	- coordinate calculation steps - input of title and choice of interactive or batch mode data entry
2.	Subroutine INPUT	- input of member and nodal data (coordinates, member incidences etc)
3.	Subroutine RECT	- input of substructure interface dimensions for rectangular plan or plane truss towers


MAIN
INPUT
RECT
TRIANG
TYPE
BCS
DATOUT
LDINPT
SUCOPT
CALBRN
EQUIBC
EQUILD
TRILD
CELDSN
ANULL
DCORE
BAREL
APPBCS
SOLVE
BARFOR
DESIGN
NEWCOD
RECTCD
TRIACD
PREDFC
DESIGN
SEARCH
COST
RECCOD
RECTRC
TRIARC
CONCCK
OUTPUT
EQUIBC
TRILD
EQUILD
ANULL
DCORE
BAREL
APPBCS
SOLVE
BARFOR
BAROUT
PERMST
DYOUT

4.	Subroutine TRIANG	- input of substructure interface dimensions for triangular plan towers
5.	Subroutine TYPE	- input of section classes and effective length coefficients
6.	Subroutine BCS	- input of systems support constraints
7.	Subroutine DATOUT	- output of data, if required
8.	Subroutine LDINPT	- input of loading data and combination load cases
9.	Subroutine SUCOPT	- organization of the Dynamic Programming calculations and output of results
10.	Subroutine CALBRN	- only used in program DYNPRE - calibration of the forces in a sub-structure for 4 controlling configurations
11.	Subroutine NEWCOD	- coordination of the calculations for the repositioning of node in configuration changes
12.	Subroutine RECTCD	- re-calculation of node positions in plane truss or rectangular plan tower substructures
13.	Subroutine TRIACD	- re-calculation of node positions in triangular plan tower substructures
14.	Subroutine PREDFC	- only used in program DYNPRE - prediction of forces for a general substructure configuration from the four controlling configurations
15.	Subroutine EQUIBC	- setting up of equivalent substructure support constraints

16.	Subroutine EQUILD	- calculation of equivalent substructure loads for plane truss and rectangular plan towers
17.	Subroutine TRILD	- calculation of equivalent substructure loads for triangular plan towers
18	Subroutine CELDSN	- coordination of substructure analysis and design calculations
19.	Subroutine ANULL	- initialization of stiffness matrix and/or calculation of stiffness matrix dimensions
20.	Subroutine DCORE	- UNIVAC systems routine for the dynamic core storage of a single array ((AB)
21.	Subroutine BAREL	- calculation of each member stiffness matrix and assembly of system stiffness matrix
22.	Subroutine APPBCS	- application of systems and equivalent substructure support constraints
23.	Subroutine SOLVE	- solution of the simultaneous equations of the stiffness matrix by the Crout Reduction Method
24.	Subroutine BARFOR	- calculation of member forces from differential nodal displacements
25.	Subroutine DESIGN	- checks the design of current design variables and predicts 'improved' values
26.	Subroutine SEARCH	- searches through the list of sections to find a suitable section for the predicted value from DESIGN
27.	Subroutine COST	- calculation of the 'cost' or weight of the structure

28. Subroutine RECCOD - coordination of the recalculation of the entire structure's nodes as predicted by the DPSA method

29. Subroutine RECTRC - re-calculation of node position in plane truss or rectangular plan tower structures

30. Subroutine TRIARC - re-calculation of node positions in triangular plan tower structures

31. Subroutine CONCCK - checks the similarity between consecutive cycle results in order to terminate the DPSA calculations

32. Subroutine OUTPUT - coordinates the output of the relevant results

33. Subroutine BAROUT - outputs the forces, stress, slenderness ratios and structural designs for each substructure

34. Subroutine PERMST - calculates the permissible stresses and slenderness ratios for output

35. Subroutine DYOUT - prepares a data-file for input to the program DYSPAN

J1.5 PROGRAM LIMITATIONS

The following limitations are set on the size of the structure which can be designed with the programs DYNGEO and DYNPRE using the UCT UNIVAC 1106.

Number of nodes in the structure

NNP = 100

Number of members in the structure

NEL = 250

Number of substructures

NCELL = 10

Number of member groups per substructure

NTYPE = 10

Number of possible substructure interface dimensions

(in each coordinate direction) = 5

Total number of loaded nodes per load case

NPL = 20

Total number of load cases

NLC = 5

Total number of combination load cases

NLC - 2 = 3

Number of restrainted degrees of freedom (i.e. support reactions)

NRS = 50

With the above storage capacity, special arrangements must be made for interactive data input.

J1.6 STORAGE ARRANGEMENTS

All arrays are individually dimensioned, except the array AB, which is dynamically stored. The following arrays are stored in AB.

Array Name	Array Dimensions	Total length
Stiffness matrix	S(NDOF,ND)	NDOF*ND
Load vector	PLD(NDOF)	NDOF
Displacement vector	U(NDOF)	NDOF
Force vector	P(NELC)	NELC

Total length of array AB = NDOF(ND + 2) + NELC

AB is setup as a single array together with the housekeeping array IH as follows:

S matrix	PLD	U	P
NDOF x ND	NDOF	NDOF	NDOF

(1) (2) (3)

Housekeeping Array IH

where NODF = number of degrees of freedom for current substructure

ND = 3 x (MAXDIF(NCEL) + 2) = maximum member node difference for current substructure for calculation of half band width

NELC = number of members in the current substructure

J1.7 NOTES FOR DATA INPUT

i) Notes as for program DYSPAN (Appendix I)

ii) The data for both program DYNGEO AND DYNPRE are identical - i.e. a single set of data can be used to execute both programs.

iii) The axes system to be used in shown in Figure 3.6

iv) The nodes and members must be numbered strictly in consecutive substructure order. Nodes must be numbered from foundation level upward in circular order for each elevation level. All members in one substructure must be numbered before numbering is commensed for the following substructure. Members at the interface levels must be included in the lower substructure (cf Example 1, Chapter 4).

J1.8 CONTROL STATEMENTS

Runstream used for batch node jobs:

```
@RUN␢runid,accountnumber/userid,CONTRA,time,pages
@PASSWD␢password
@ASG,AX␢DYNGEO
```

or

```
@ASG,AX␢DYNPRE
```

```
        @DELETE,C¢DASPAN
        @XQT¢DYNGEO.ABS
or      @XQT¢DYNPRE.ABS
              data required
        @FIN
```

For interactive mode (terminal) jobs, special arrangements with regard to storage must be made due to the limitations on the UNIVAC 1106 system. Segmental MAPping and a considerable decrease in the array sizes is needed. However, the runstream to be used is:

```
        terminal call number,  e.g U1106H
        userid/password
        @RUN¢runid,accountnumber,CONTRA
        @ASG,AXZ¢DYNGEO
or      @ASG,AXZ¢DYNPRE
        @XQT¢DYNGEO.ABS
or      @XQT¢DYNPRE.ABS
        enter data as required by the questions asked by the program
        @FIN
        @@TERM
```

J1.9 DATA PREPARATION

A list of the required data follows. The input instructions are set up in a flow chart and should be followed explicitly. All upper case variables are for data input, while lower case sentences are merely flow chart directions. Sentences and variables in [ ] require no input, but the value calculated is required for future data input.

```
        TITLE

        IDT

        IOUTP

        NNP,NEL

        NT(J),X(J),Y(J),Z(J)          } J  =  1,NNP
        :
        :
        NPI(K),NPJ(K),NCL(K),ITYPE(K)  }    K = 1,NEL
        :
        :
        [find the number of substructures in the structure  = NCELL]

        [find the maximum number of member groups in each substructure, I:

          i.e.   NTYPE(I)   for  I  =  1,NCELL]

        ICS

[1]     MINNO(I),MAXNO(I),MINEL(I),MAXEL(I),MAXDIF(I)

        if [I = 1] enter  MAX1

        if [I = NCELL] enter  MIN1

        if [ICS = 1 or 3] go to [2]

        if [ICS = 2] go to [4]

[2]     if [I ≠ 1] go to [3]

        VV1(1,1),VV1(1,2),VV1(1,3),VV1(1,4),VV1(1,5)

        VV2(1,1),VV2(1,2),VV2(1,3),VV2(1,4),VV2(1,5) [not for ICS = 3]


[3] VV1 W1(I+1,K)   for K = 1 to 5

        VV2(I+1,K)  for K = 1 to 5  [not for ICS = 3]

        VV3(I+1,K)  for K = 1 to 5

        go to [6]

[4]     if [I = 1]  go to [5]

        VV1(1,1),VV1(1,2),VV1(1,3),VV1(1,4),VV1(1,5)
```

[5] VV1(I+1,K) for K = 1 to 5

VV2(I+1,K) for K = 1 to 5

[6] IDSEC(K,I),EFFLEN(K,I) for K = 1 to NTYPE(I)

go to [1] for input of data for next substructure. If I = NCELL cont

COMLR,HIGLR,PERA,YS

NRS

NPB(N),NCOND(N) for N = 1 to NRS

NLC

[7] for data input for load cases 1 and 2 to to [8]

for data input for load cases 3,4 and 5 go to [9]

[8] NPL

I,LX(I),LY(I),LZ(I) for I = 1 to NPL

return to [7] for input of following load case

[9] is this load case to be a combination of preceeding load cases ?

if NO, go to [8]

if YES, enter

IB

LC(1),LC(2),...,LC(IB)

does the designer require data to be prepared for use in program

DYSPAN ? enter YES or NO

does the designer require a graphical plot of the design ?

enter YES or NO

go to [7] or terminate input

Nomenclature Explanation

The variables in the above list are defined as follows:

TITLE	- an alphanumeric character string (not exceeding 60 characters) for the job title
IDT = 1	- interactive mode data input
= 2	- batch mode data input (card image format)
IOUTP = 'YES'	- progressive output reports on the status of the DPSA calculations to be printed
= 'NO'	- only final results to be printed
NNP	- number of nodes in the structure
NEL	- number of members in the structure
NT(J) = 'FX'	- node to be fixed at X(J),Y(J),Z(J) throughout the entire claculation
= 'DX'	- node whose positional coordinates are dependent on the substructure interface dimensions
= 'VX'	- primary interface node - i.e., the nodes which defined the interface dimensions (cf section 3.3.2.2)
X(J),Y(J),Z(J)	- initial coordinates of node J (cf section 3.3.2.3)
NPI(K),NPJ(K)	- node numbers associated with member J
NCL(K)	- substructure number in which member J is situated
ITYPE(K)	- group reference number associated with similar member types in the substructure (i.e. the same area and radius of gyration is to be found for all members with similar

		ITYPE number within a substructure) total no of member groups per substructure = NTYPE(I), where I is the substructure number
ICS = 1	-	rectangular plan tower structure
= 2	-	triangular plan tower structure
= 3	-	plane truss tower structure
MINNO(I)	-	smallest node number in substructure I
MAXNO(I)	-	largest node number in substructure I
MINEL(I)	-	smallest member number in substructure I
MAXEL(I)	-	largest member number in substrucutre I
MAX1	-	largest node number at foundation level
MIN1	-	smallest node number at top interface level
for ICS = 1		
VV1(1,1),...,VV1(1,5)	-	possible dimensions in x-direction of foundation level interface
VV2(1,1),...,VV2(1,5)	-	possible dimensions in y-direction of foundation level interface
VV1(I+1,K) VV2(I+1,K) VV3(I+1,K)	-	possible dimension of the upper interface level of substructure I in the x,y and z-directions (cf section 3.3.2.2)
for ICS = 2		
VV1(1,1),...,VV1(1,5)	-	possible radii of the circles enclosing the equilateral triangles of the foundation level interface
VV1(I+1,K) VV2(I+1,K)	-	possible radii of enclosing circles and elevation of upper interface level of substructure I respectively (cf section 3.3.2.2)

for ICS = 3

VV1(1,1),...,VV1(1,5)	-	possible dimensions in x-direction of foundation level interface
VV1(I+1,K) VV3(I+1,K)	-	possible dimensions of the upper interface level of substructure I in the x and z directions respectively
IDSEC(K,I)	-	section class identification number for each group of member K in substructure I
	=	1 for pipe sections
	=	2 for angle sections
	=	3 for double angle sections
	=	4 for channel sections
EFFLEN(K,I)	-	effective length coefficient for member group K in substurcture I
COMLR	-	permissible slenderness ratio for struts (BS 449 clause 33)
HIGLR	-	maximum permissible slenderness ratio for reversal of stress (BS 449 clause 44a)
PERA	-	permissible percentage cross sectional area for the purposes of tensile stress calculations
YS	-	minimum yield stress of the material used (BS 449 appendix B)
NRS	-	number of rigidly restrained degrees of freedom for the structure; each node has 3 degrees of freedom to be satisfied, i.e. maximum number of restraints per node is 3
NPB(N)	-	restrained node number
NCOND(N)	-	type of support constraint applied

= 1 for restraint in x-direction

= 2 for restraint in y-direction

= 3 for restraint in z-direction

NLC - number of load cases to be processed

NPL - number of loaded nodes in the structure

I,LX(I),LY(I),LZ(I) - loads in the x,y,z directions respectively at node I

IB - number of load cases to be combined

LC(1),..,LC(IB) - load case numbers to be combined

J1.10 SAMPLE INPUT

Sample data for example 2 of Chapter 3 is as follows

```
CONTRA*DYNPRE(1).TRI1
TRIANGULAR 2 CELL TOWER       TITLE
2                             IDT
NO                            IOUTP
15,42                         NNP,NEL
VX,3,0,0
VX,-1.5,-2.5981,0.0
VX,-1.5,2.5981,0.0
DX,0.625,1.082532,1
DX,0.625,-1.082532,1
DX,-1.25,0,1
VX,2,0,2
VX,-1,-1.732,2                NT,X-COORD,Y-COORD,Z-COORD
VX,-1,1.732,2
DX,0.375,0.64952,3
DX,0.375,-0.64952,3
DX,-0.75,0,3
VX,1,0,4
VX,-0.5,-0.866,4
VX,-0.5,0.866,4
1,7,1,1
2,8,1,1
3,9,1,1
7,8,1,2
8,9,1,2
9,7,1,2
4,7,1,3
4,9,1,3
4,3,1,3
4,1,1,3
5,7,1,3
5,8,1,3
5,2,1,3
5,1,1,3
6,9,1,3
6,8,1,3
6,2,1,3
6,3,1,3
4,5,1,4
5,6,1,4
6,4,1,4                       NPI,NPJ,NCL,ITYPE
7,13,2,1
8,14,2,1
9,15,2,1
13,14,2,2
14,15,2,2
15,13,2,2
10,13,2,3
10,15,2,3
10,9,2,3
10,7,2,3
11,13,2,3
11,14,2,3
11,8,2,3
11,7,2,3
12,15,2,3
12,14,2,3
12,8,2,3
12,9,2,3
10,11,2,4
```

60	11,12,2,4	
61	12,10,2,4	
62	2	ICS
63	1,9,1,21,6	MINNO(1),MAXNO(1),MINEL(1),MAXEL(1),MAXDIF(1)
64	3	MAX1
65	0.4,0.45,0.5,0.55,0.6	VV1(1,J) for J=1,5
66	0.3,0.35,0.4,0.45,0.5	VV1(2,J) for J=1,5
67	1.8,1.9,2.0,2.1,2.2	VV2(2,J) for J=1,5
68	1,1,2,1,2,1,2,1	IDSEC(I),EFFLEN(I)....
69	7,15,22,42,6	MINNO(2),MAXNO(2),MINEL(2),MAXEL(2),MAXDIF(2)
70	13	MIN1
71	0.2,0.25,0.3,0.35,0.4	VV1(3,J) for J=1,5
72	3.8,3.9,4.0,4.1,4.2	VV2(3,J) for J=1,5
73	1,1,2,1,2,1,2,1	IDSEC(I),EFFLEN(I)....
74	180.0,350.0,100.0,250.0	COMLR,HIGLR,PERA,YS
75	9	NRS
76	1,1	
77	1,2	
78	1,3	
79	2,1	
80	2,2	NPB(I),NCOND(I)
81	2,3	
82	3,1	
83	3,2	
84	3,3	
85	1	NLC
86	3	NPL
87	13,0,0,10	
88	14,0,15,0	I,PX(I),PY(I),PZ(I)
89	15,-12,0,0	
90	YES	Optional Data for program DYSPAN
91	NO	Optional Data for program DRAW

The following pages contain the Sample Output for the structure in Example 2.

NODAL COORDINATES FOR FINAL STRUCTURE
****

NODE NUMBER	X-COORD	Y-COORD	Z-COORD	NODE TYPE
1	.500	.000	.000	VX
2	-.250	-.433	.000	VX
3	-.250	.433	.000	VX
4	.106	.184	.900	DX
5	.106	-.184	.900	DX
6	-.212	.000	.900	DX
7	.350	.000	1.800	VX
8	-.175	-.303	1.800	VX
9	-.175	.303	1.800	VX
10	.087	.152	2.800	DX
11	.087	-.152	2.800	DX
12	-.175	.000	2.800	DX
13	.350	.000	3.800	VX
14	-.175	-.303	3.800	VX
15	-.175	.303	3.800	VX

WHERE FX=SET OF COORDINATES FOR A FIXED NODE
DX=SET OF COORDINATES FOR A DEPENDANT NODE
VX=SET OF COORDINATES FOR A VARIABLE NODE

CELL CONSTANTS
****

CELL NUMBER	NODE NUMBER MIN	NODE NUMBER MAX	ELEMENT NO. MIN	ELEMENT NO. MAX	MAXNODE DIFFERENCE
1	1	9	1	21	6
2	7	15	22	42	6

BEST SOLUTION CHOSEN BY DYNAMIC PROGRAMMING
****
THE STRUCTURE IS OF TRIANGULAR PLAN SHAPE,
THEREFORE:
VARIABLE 1 REPRESENTS THE RADIUS OF THE
ESCRIBED CIRCLE AT EACH CELL LEVEL
VARIABLE 2 REPRESENTS THE ELEVATION OF
EACH CELL LEVEL ABOVE THE BASE

IN THE DIAGRAMATIC REPRESENTATION,
THE ACTUAL POSITION VARIABLES CHOSEN
ARE INDICATED BY THE USE OF AN >X>
WHILE THE OTHER POSSIBLE POSITIONS
ARE INDICATED BY AN >*>

VARIABLE 1
****

LEVEL POSITION NUMBER VALUE

BASE	3	.500
1	2	.350
2	4	.350

GRAPHIC REPRESENTATION OF VARIABLE 1
****

```
     *       *       *       X       *          LEVEL 2
   .200    .250    .300    .350    .400

     *       X       *       *       *          LEVEL 1
   .300    .350    .400    .450    .500

     *       *       X       *       *          BASE
   .400    .450    .500    .550    .600
```

VARIABLE 2
****

LEVEL	POSITION NUMBER	VALUE
BASE	1	.000
1	1	1.800
2	1	3.800

GRAPHIC REPRESENTATION OF VARIABLE 2
****

```
     X       *       *       *       *          LEVEL 2
   3.800   3.900   4.000   4.100   4.200

     X       *       *       *       *          LEVEL 1
   1.800   1.900   2.000   2.100   2.200

     X                                          BASE
    .000
```

FOUNDATION REACTIONS

****

NODE	REACTION X	REACTION Y	REACTION Z
1	4.000	-5.675	-68.800
2	3.416	-4.663	-36.419
3	4.584	-4.663	95.220

LOADING AND FORCES FOR EACH CELL
****

CELL NUMBER 1
****

LOADING NODE	LOAD X	LOAD Y	LOAD Z
7	-4.000	5.964	55.714
8	-3.165	4.518	26.631
9	-4.835	4.518	-72.346

AXIAL FORCES
****

ELEMENT NUMBER	MEMBER TYPE REF. NUMBER	FORCE (KN)	STRESS KN/M**2	PERMISSIBLE STRESS	L/R RATIO
1	1	54.544	87158.345	155000.000	51.514
2	1	27.456	43873.318	155000.000	51.514
3	1	-75.686	-120942.630	-132152.637	51.514
4	2	.634	4464.351	155000.000	125.770
5	2	-.464	-3267.570	-53492.920	125.768
6	2	-.447	-3150.045	-53490.883	125.770
7	3	-13.289	-56548.205	-56733.003	121.534
8	3	6.582	28009.072	155000.000	121.538
9	3	-10.840	-46126.806	-52016.140	127.810
10	3	3.786	16111.889	155000.000	127.804
11	3	14.724	62654.957	155000.000	121.534
12	3	11.107	47265.506	155000.000	121.538
13	3	14.910	63448.078	155000.000	127.810
14	3	12.254	52144.830	155000.000	127.804
15	3	-3.331	-14174.728	-56731.390	121.536
16	3	-11.879	-50547.354	-56731.390	121.536
17	3	-4.851	-20642.876	-52018.660	127.807
18	3	-11.143	-47418.731	-52018.660	127.807
19	4	-.000	-.548	-106796.833	76.389
20	4	-.000	-.933	-106773.699	76.367
21	4	.000	.933	155000.000	76.367

RECOMMENDED SECTIONS FOR THIS DESIGN
****

NOTE: THE FOLLOWING CODE IS USED TO INDICATE
SECTION CLASS
1 PIPE SECTION

2 ANGLE
3 DOUBLE ANGLE
4 CHANNEL

MEMBER TYPE REF. NUMBER	SECTION CLASS	SECTION SPECIFICATION NUMBER
1	1	14
2	2	1
3	2	3
4	2	1

CELL NUMBER 2
****

LOADING

NODE	LOAD X	LOAD Y	LOAD Z
13	.000	.000	10.000
14	.000	15.000	.000
15	-12.000	.000	.000

AXIAL FORCES
****

ELEMENT NUMBER	MEMBER TYPE REF. NUMBER	FORCE (KN)	STRESS KN/M**2	PERMISSIBLE STRESS	L/R RATIO
22	1	19.880	57290.793	155000.000	102.564
23	1	8.212	23665.972	155000.000	102.564
24	1	-24.625	-70963.998	-74557.809	102.564
25	2	-4.491	-25807.782	-72456.471	104.520
26	2	-8.020	-46090.270	-72458.903	104.517
27	2	7.485	43018.534	155000.000	104.520
28	3	-25.798	-59995.025	-57880.891	120.107
29	3	21.963	51077.770	155000.000	120.106
30	3	-25.800	-60000.859	-57882.250	120.106
31	3	21.967	51085.146	155000.000	120.107
32	3	15.474	35986.052	155000.000	120.107
33	3	15.481	36001.351	155000.000	120.106
34	3	15.477	35992.226	155000.000	120.106
35	3	15.478	35995.174	155000.000	120.107
36	3	3.767	8761.000	155000.000	120.106
37	3	-24.062	-55957.405	-57881.750	120.106
38	3	3.770	8767.672	155000.000	120.106
39	3	-24.065	-55964.078	-57881.750	120.106
40	4	-.001	-9.138	-122200.328	62.905
41	4	-.002	-11.721	-122214.563	62.890
42	4	.002	11.710	155000.000	62.890

RECOMMENDED SECTIONS FOR THIS DESIGN
****

NOTE: THE FOLLOWING CODE IS USED TO INDICATE
SECTION CLASS
1 PIPE SECTION
2 ANGLE

3 DOUBLE ANGLE
4 CHANNEL

MEMBER TYPE REF. NUMBER	SECTION CLASS	SECTION SPECIFICATION NUMBER
1	1	11
2	2	2
3	2	5
4	2	1

TOTAL WEIGHT OF THIS DESIGN
IN KILONEWTONS IS 1.115

TO USE THE DATA CREATED ABOVE, THE CONTROLS
@ASG,AX CONTRA*DYSPAN.
@XQT CONTRA*DYSPAN.ABS
@ADD CONTRA*DASPAN.
ARE REQUIRED

A P P E N D I X   K

USER'S MANUAL: PROGRAM DRAW

K1.1   PROGRAM NAME: CONTRA*DYNPRE.DWSABS

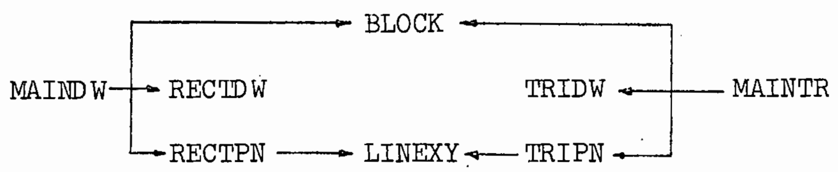
CONTRA*DYNPRE.DWTABS

K1.2   DESCRIPTION

Program for the computer plot of the results produced by DYNPRE and DYNGEO. DWSABS plots the structure shapes for rectangular plan and plane truss towers. DWTABS plots the structure shape for triangular plan towers

K1.3   SUBSTRUCTURE LAYOUT

(1)   DWSABS   (2)   DWTABS



K1.4   SUBROUTINE FUNCTIONS

1.   MAINDW,MAINTR   – coordination of plotting for (a) plane truss and rectangular plane towers; (b) triangular plan towers respectively

2.   BLOCK   – plots a title block and

3.   RECTDW,TRIDW   – plots the elevations of tower types (a) and (b) respectively

4. RECTPN,TRIPN - plots the plans of the interface buds for tower types (a) and (b) respectively

5. LINEXY - plots the axes systems on the plans of the interface levels

K1.5 PROGRAM LIMITATIONS

Since these programs are for use primarily on the Graphic Display Terminal, the following limitations are placed on the program by the interactive core storage capabilities of the UNIVAC 1106.

number of nodes NNP = 50

number of members NEL = 100

Other limitations are similar to those in Appendix J.

K1.6 NOTES

Data for these programs are prepared by programs DYNGEO and DYNPRE; no other data is required.

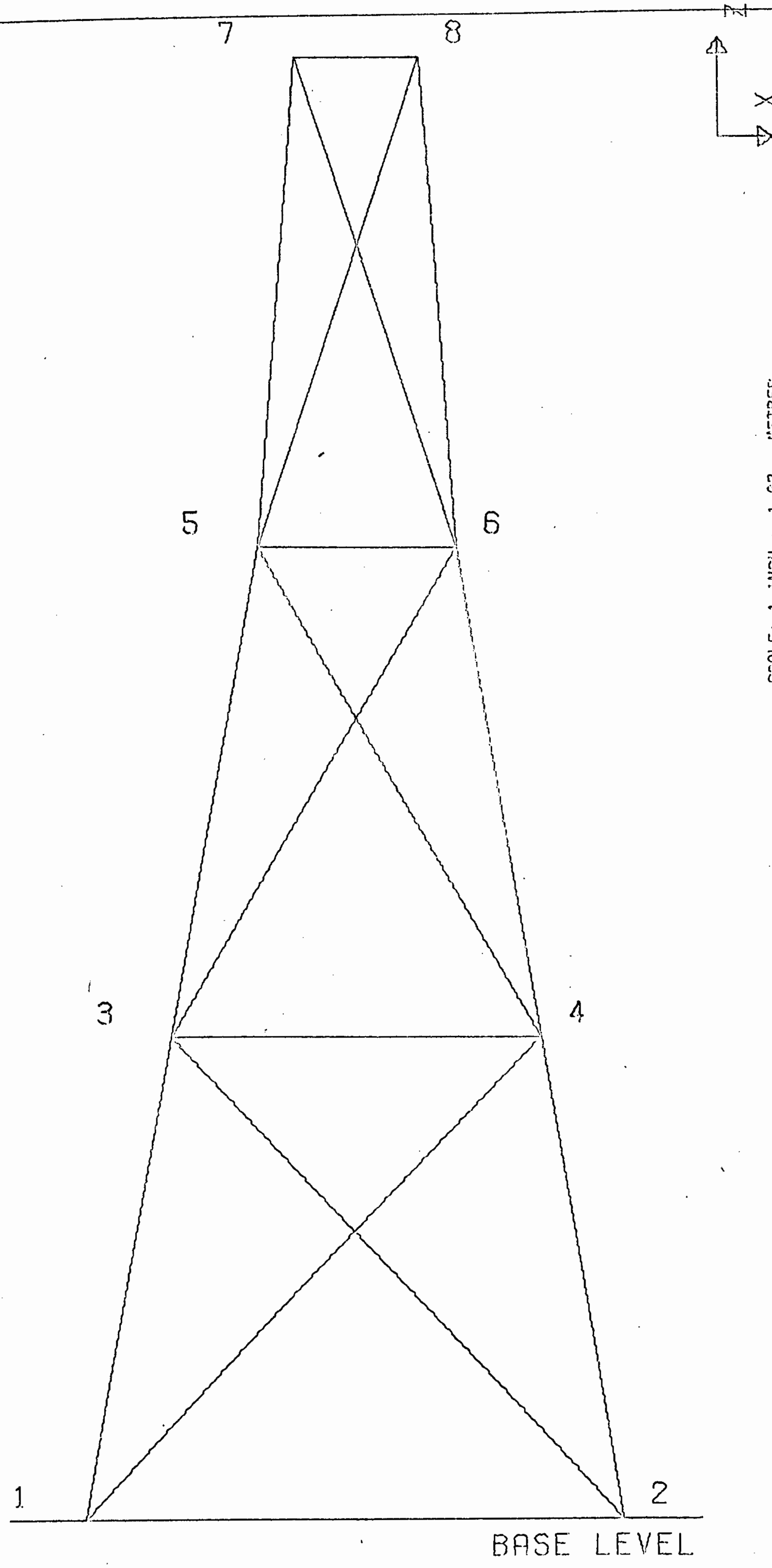
K1.7 CONTROL STATEMENTS

The runstreams for execution for these programs on the Graphics Display Terminal are

```
        terminal call number eg  UT106L
        userid/password
        @RUNbrunid,accountnumber,CONTRA
        @ASG,AXbDYNPRE
        @XQTbDYNPRE.DWSABS
or      @XQTbDYNPRE.DWTABS
        enter data as required by Graphics Terminal
        = drawing produced
```

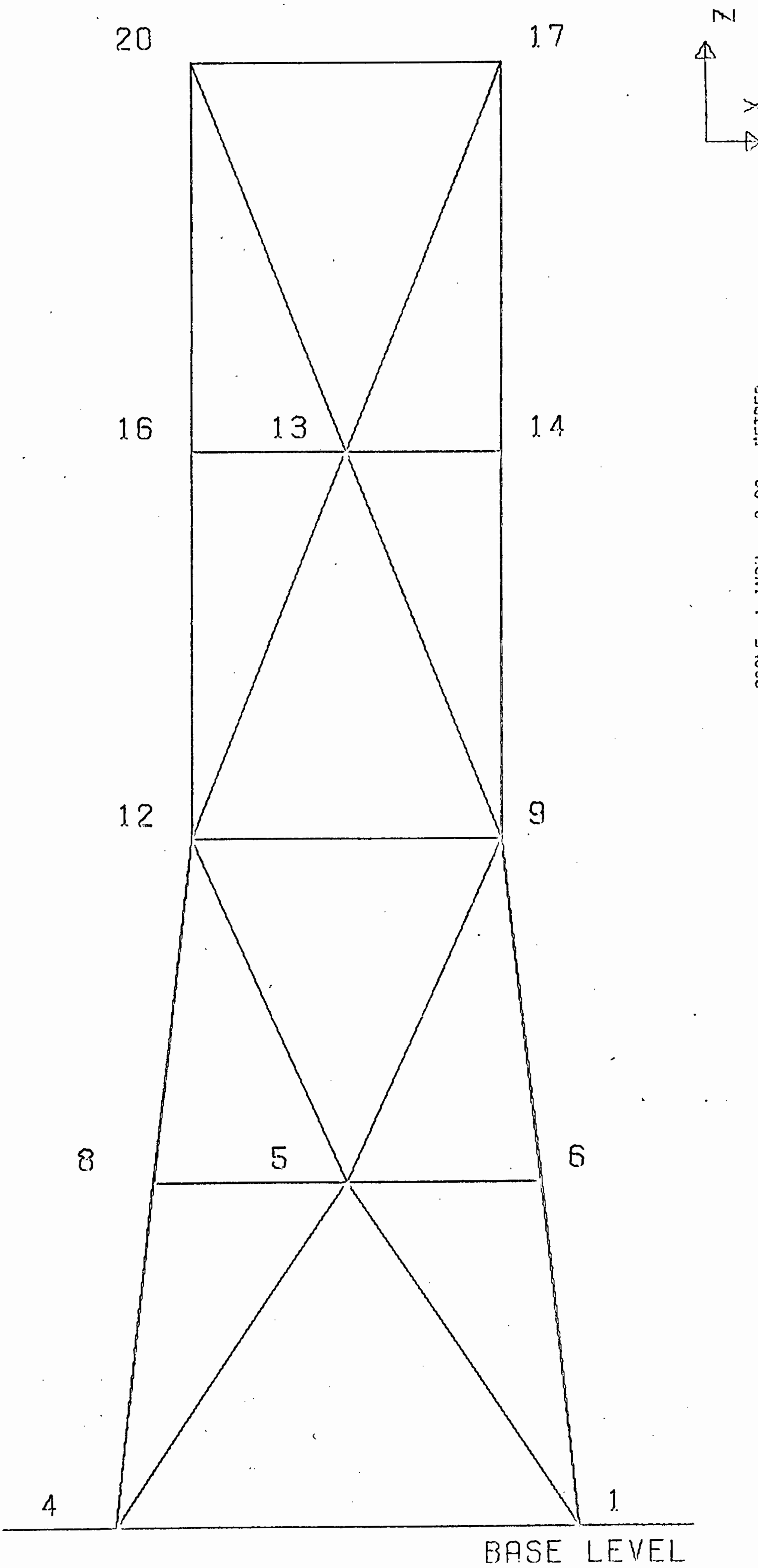
```
if another drawing is required
@XQTbTEKFAST*CALPREV.RELOOK
if plot is to be sent to CALCOMP plotter
@XQTbTEKFAST*CALPREV.TOPLOTTER
@FIN
@@TERM.
```

K1.8 The following pages contain the Sample Output from Program DRAW for Examples 1 and 3.



UNIVERSITY OF CAPE TOWN
DEPARTMENT OF CIVIL ENGINEERING
STRUCTURAL DESIGN PROGRAM

DRAWING TITLE: Plane Truss Tower : Example 1
DRAWING TYPE: ELEVATION, X-Z PLANE
PROGRAMMER: G.C.H.
PAGE 1

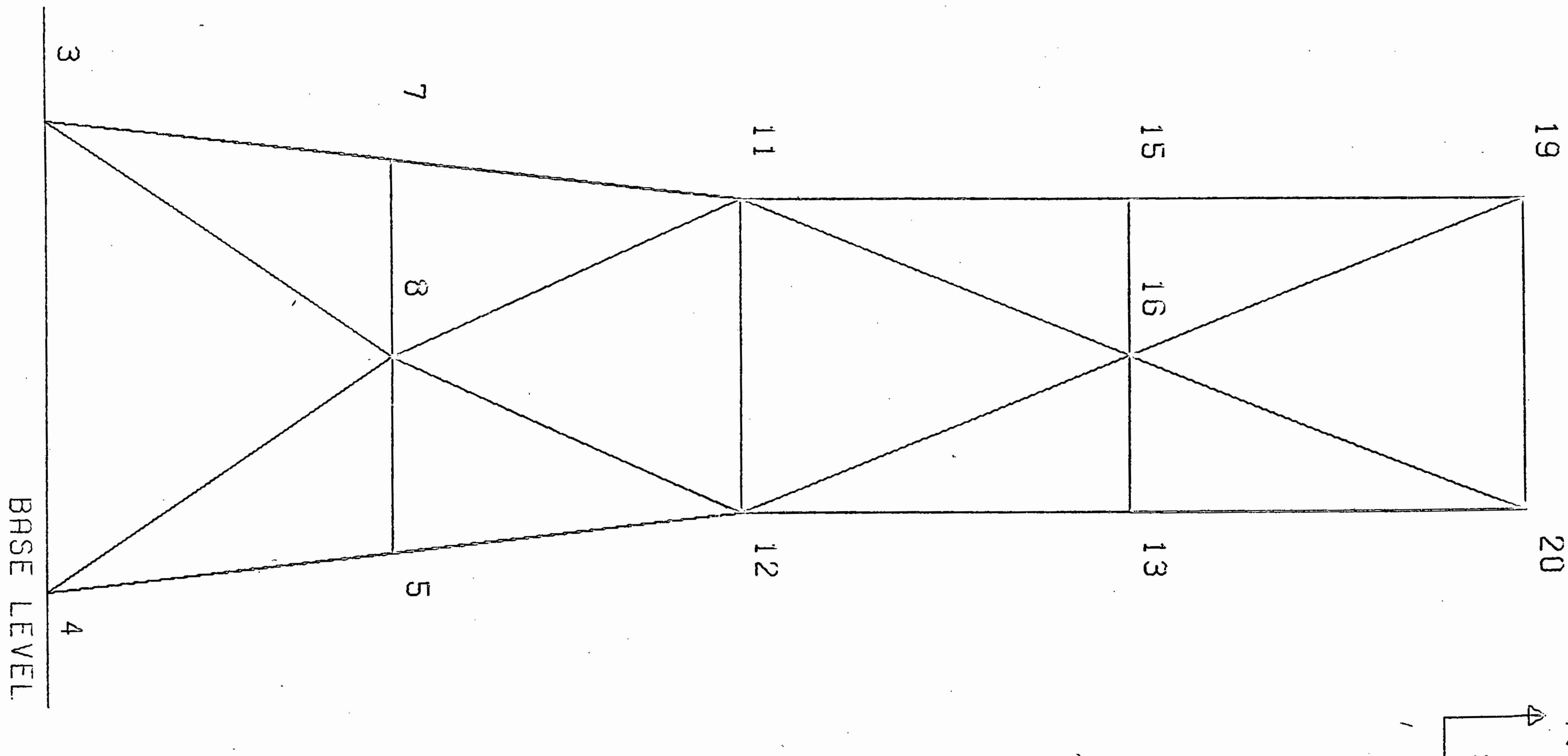


SCALE: 1 INCH = 2.63 METRES

UNIVERSITY OF CAPE TOWN
DEPARTMENT OF CIVIL ENGINEERING
STRUCTURAL DESIGN PROGRAM

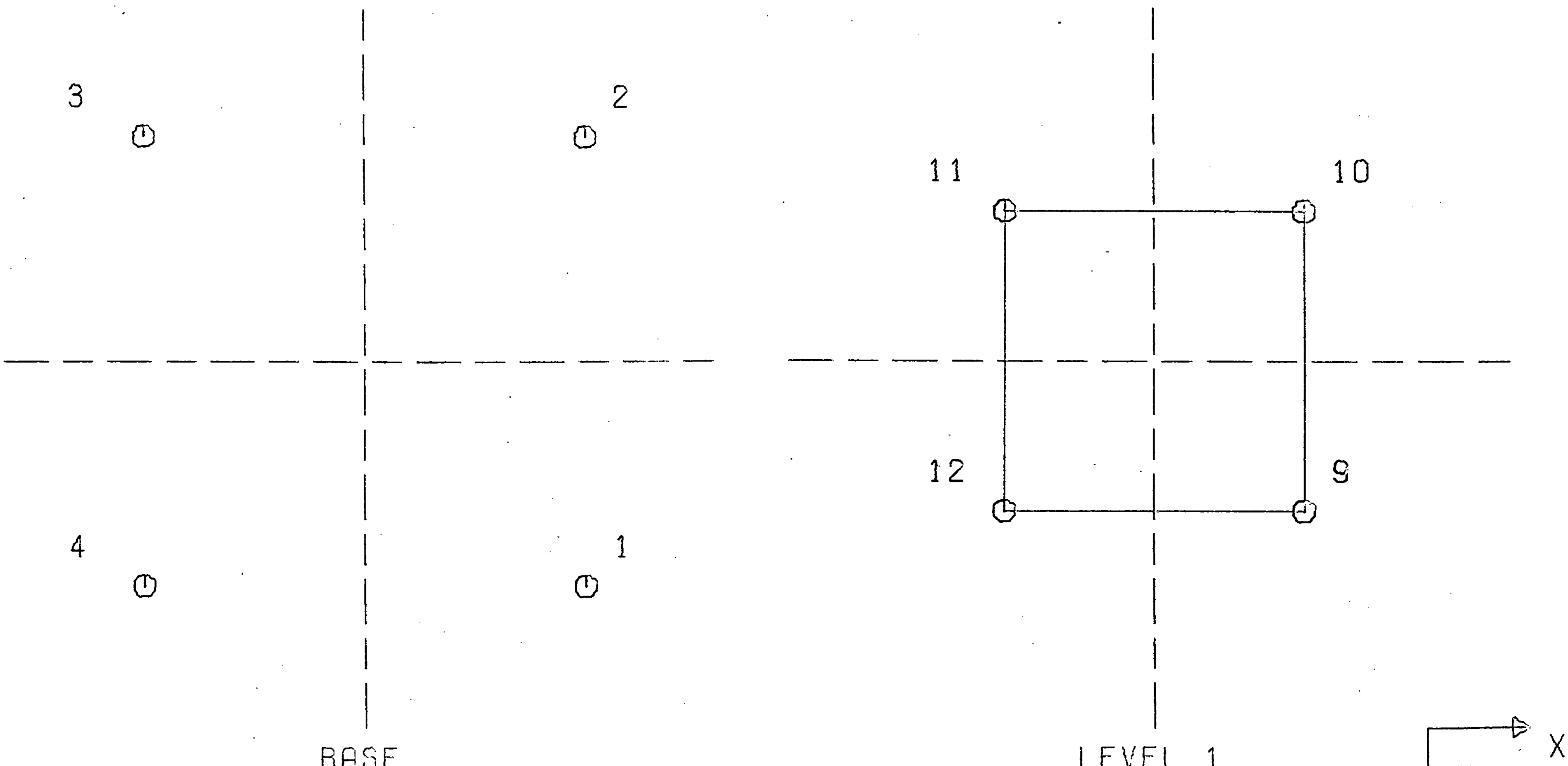
DRAWING TITLE: Rectangular Tower (2 substructures): Examp3
DRAWING TYPE: ELEVATION, X-Z PLANE
PROGRAMMER: G.C.H.

PAGE 1



SCALE: 1 INCH = 2.63 METRES

UNIVERSITY OF CAPE TOWN	DRAWING TITLE: Rectangular Tower (2 substructures) : Example 3
DEPARTMENT OF CIVIL ENGINEERING	DRAWING TYPE: ELEVATION, Y-Z PLANE
STRUCTURAL DESIGN PROGRAM	PROGRAMMER: G.C.A. PAGE 2



SCALE 1 INCH = 2.50 METRES

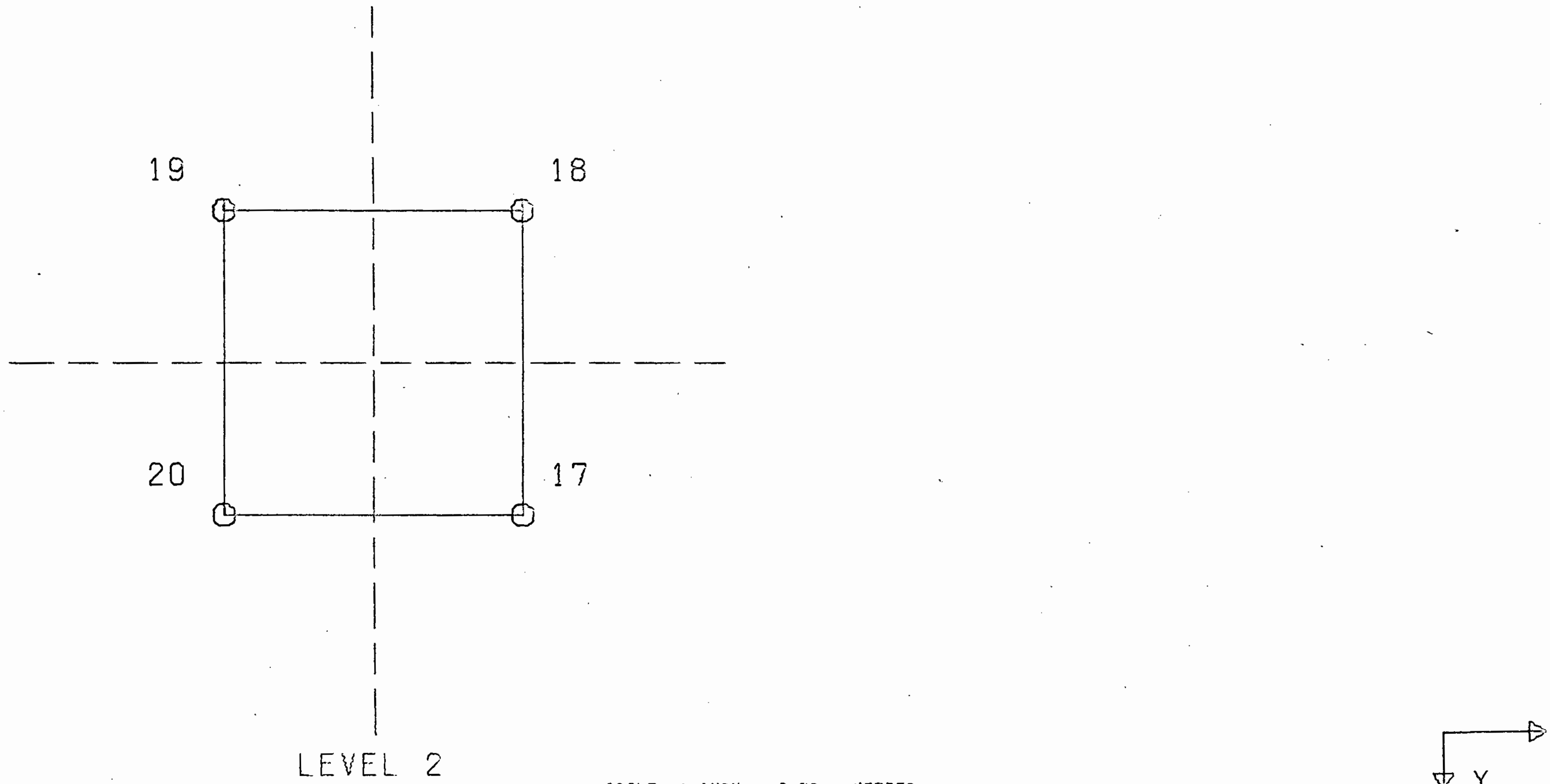
UNIVERSITY OF CAPE TOWN
DEPARTMENT OF CIVIL ENGINEERING
STRUCTURAL DESIGN PROGRAM

DRAWING TITLE: Rectangular Tower (2 substructures): Example 3

DRAWING TYPE: PLAN OF BASE AND LEVEL 1

PROGRAMMER: G. C. H.

PAGE 3



SCALE: 1 INCH = 2.50 METRES

UNIVERSITY OF CAPE TOWN	DRAWING TITLE: Rectangular Tower (2 substructures): Example 3
DEPARTMENT OF CIVIL ENGINEERING	DRAWING TYPE: PLAN OF LEVEL 2
STRUCTURAL DESIGN PROGRAM	PROGRAMMER: G.C.H. — PAGE 4

APPENDIX L

SECTION LISTS

This appendix contains the lists of section used by programs DYSPAN, DYNGEO and DYNPRE

i) Pipe Sections

Member size number	Size	Wall thickness	Area (m^2) x 10^{-6}	Radius of gyration (m) x 10^{-3}
1	12,7	1,63	56,7	3,96
2	15,9	1,63	73,1	5,08
3	19,1	1,63	89,5	6,2
4	21,4	2,0	121,9	6,9
5	25,4	2,0	147,0	8,3
6	27,0	2,0	157,0	8,87
7	31,8	2,0	187,0	10,56
8	34,13	2,0	201,9	11,4
9	42,9	2,0	257,0	14,5
10	50,8	2,0	306,7	17,3
11	57,2	2,0	347,0	19,5
12	60,3	2,0	386,4	21,8
13	76,2	2,0	466,2	26,2
14	101,6	2,0	625,8	35,2
15	114,3	2,0	705,6	39,7
16	88,9	3,25	874,5	30,3
17	101,6	2,9	1007,3	25,4
18	165,1	2,9	1477,7	57,4
19	139,7	4,85	2153,5	38,2
20	165,1	6,35	3167,0	56,2

(ii) Angle Sections

Member size number	Size	kg/m	Area (m^2) x 10^{-6}	Radius of gyration (m) x 10^{-3}
1	25 x 25	1,11	142,0	4,82
2	30 x 30	1,36	174,0	5,8
3	40 x 40	1,84	235,0	7,82
4	45 x 45	2,09	266,0	8,81
5	45 x 45	3,38	430,0	8,7
6	50 x 50	3,77	480,0	9,72
7	60 x 60	4,57	584,0	11,7
8	60 x 60	5,43	691,0	11,7
9	70 x 70	6,38	813,0	13,7
10	80 x 80	7,34	935,0	15,7
11	80 x 80	9,63	1227,0	15,6
12	90 x 90	10,9	1389,0	17,6
13	100 x 100	12,2	1551,0	19,6
14	100 x 100	15,0	1915,0	19,5
15	120 x 120	18,2	2318,0	23,6
16	120 x 120	21,6	2754,0	23,5
17	150 x 150	27,3	3483,0	29,5
18	150 x 150	40,1	5103,0	29,2
19	200 x 200	48,5	6179,0	39,4
20	200 x 200	71,1	9059,0	39,0

(iii) Double Angle Sections

Member size number	Size	kg/m	Area (m^2) x 10^{-6}	Radius of gyration (m) x 10^{-3}
1	65 x 50	5,16	1320,0	19,5
2	75 x 50	5,65	1438,0	18,6
3	80 x 60	6,37	1622,0	22,9
4	65 x 50	6,75	1720,0	19,85
5	90 x 65	7,07	1802,0	24,45
6	75 x 50	7,39	1882,0	19,02
7	100 x 75	8,04	2050,0	28,35
8	80 x 60	6,34	2126,0	23,3
9.	100 x 65	9,94	2534,0	23,95
10	100 x 75	10,6	2694,0	28,73
11	90 x 65	11,5	2914,0	25,27
12	125 x 75	12,2	3098,0	26,81
13	100 x 65	12,3	3122,0	24,32
14	125 x 75	15,0	3836,0	27,2
15	100 x 75	15,4	3934,0	29,5
16	150 x 75	17,0	4326,0	25,61
17	125 x 75	17,8	4538,0	27,56
18	150 x 90	18,2	4630,0	32,36
19	150 x 75	24,8	6326,0	25,56
20	150 x 90	26,6	6780,0	33,22

(iv) Channel Sections

Member size number	Size	kg/m	Area (m^2) $\times 10^{-6}$	Radius of gyration (m) $\times 10^{-3}$
1	76 x 38	6,7	853,0	11,2
2	80 x 45	8,64	1102,0	10,0
3	100 x 50	10,6	1345,0	11,8
4	120 x 55	13,4	1699,0	13,3
5	178 x 54	14,5	1856,0	21,5
6	127 x 64	14,9	1898,0	13,8
7	140 x 60	16,0	2037,0	14,0
8	152 x 76	17,9	2283,0	16,9
9	160 x 65	18,8	2401,0	15,2
10	180 x 70	22,0	2797,0	16,4
11	200 x 75	25,3	3218,0	17,4
12	220 x 80	29,4	3744,0	17,6
13	240 x 85	33,2	4231,0	18,5
14	260 x 90	37,9	4828,0	18,6
15	280 x 95	41,8	5342,0	18,7
16	300 x 100	46,2	5878,0	18,8
17	381 x 102	55,1	7019,0	23,4
18	380 x 102	63,1	8041,0	23,7
19	400 x 110	71,8	9152,0	22,2
20	non existent		100000,0	80,0

Courses Completed in Partial Fulfillment

of the M.Sc.(Eng) Degree at the University of Cape Town

Course		Date Credited	Credit Value
CE 515	Surface Structures	1976	5
CE 506	Properties of Concrete	1976	3
CE 519	Steel Structures	1976	3
AM 308	Numerical Analysis and Computation	1976	10
		Total	21

Total credit requirements for the M.Sc.(Eng) Degree: 40

Course Credits:	21
Half Thesis:	20
Total	41

UNIVERSITY OF CAPE TOWN

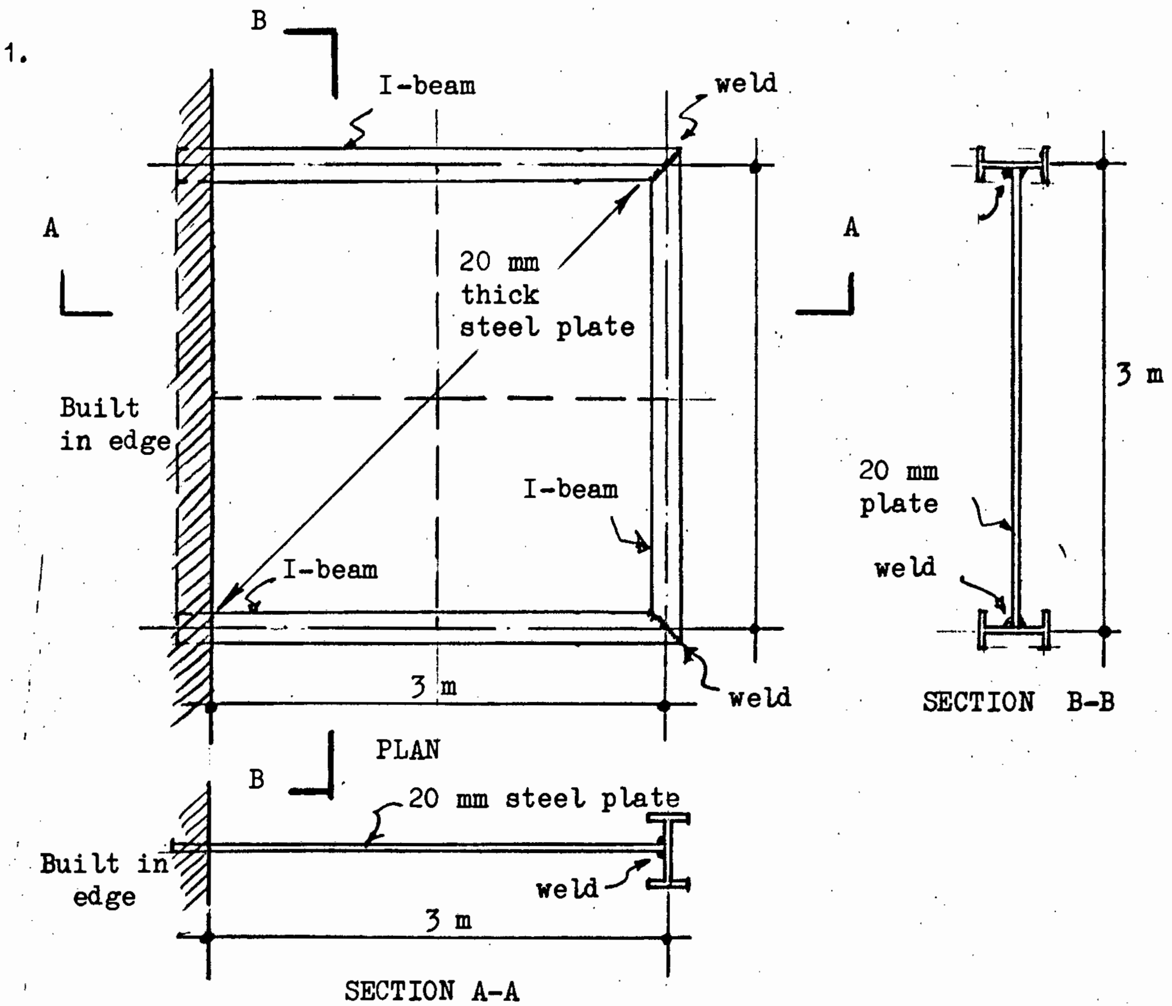
DEPARTMENT OF CIVIL ENGINEERING

UNIVERSITY EXAMINATION - 21st AUGUST, 1976

COURSE CE 515: SURFACE STRUCTURES

Time allowed: 3 hours   Answer both questions

1.



A 20 mm thick plate is welded to a rigid framework on three sides and the plate and beams are built into a substantial concrete wall on the other side as shown.

The properties of the steel beams and the plate are given below.

The plate is subjected to a uniformly distributed load of 10 kN/m^2.

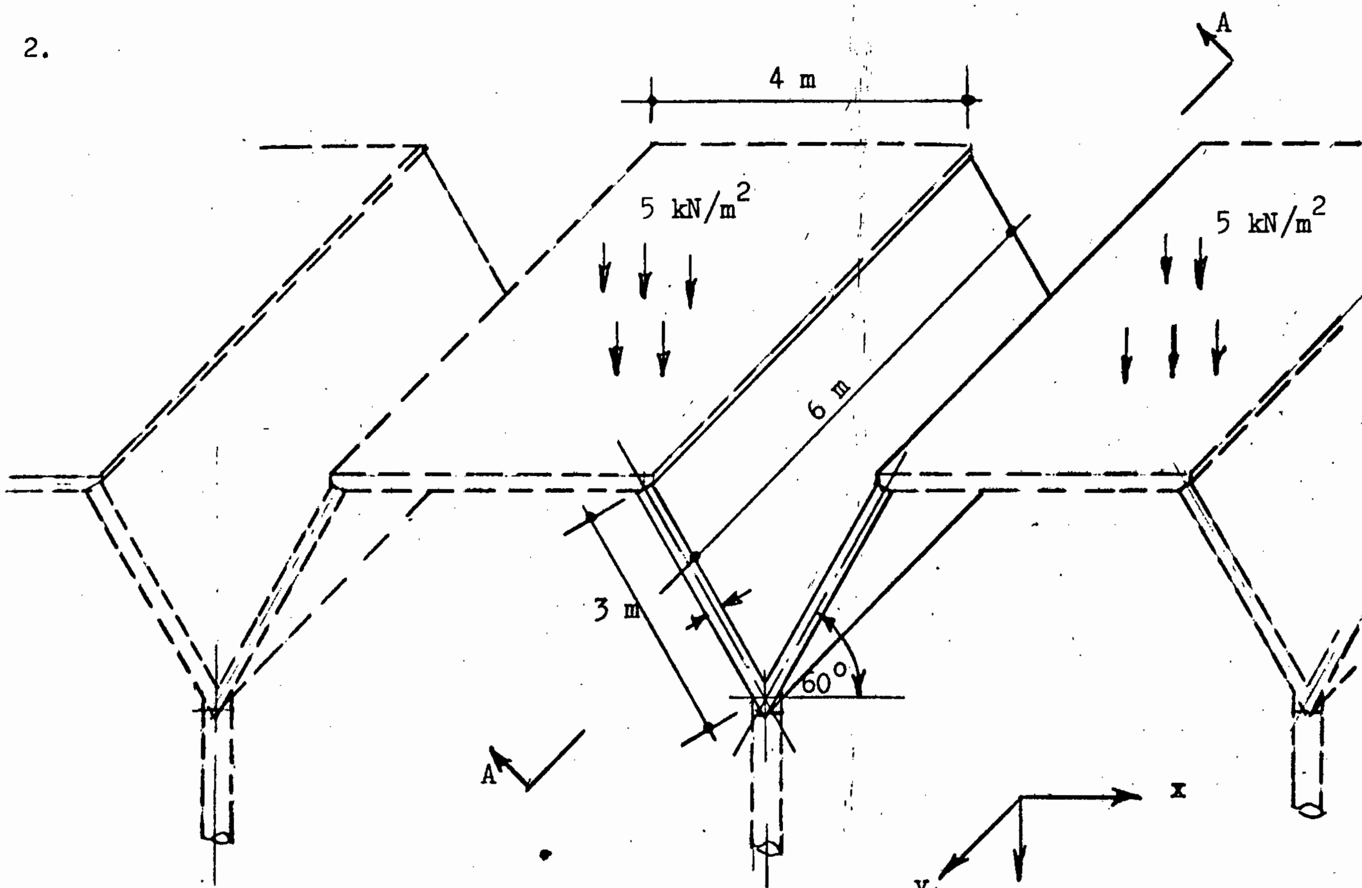
Show all the steps necessary to analyse this structure for displacements and bending moments.

(Hint: Demonstrate the method with a coarse grid as shown).

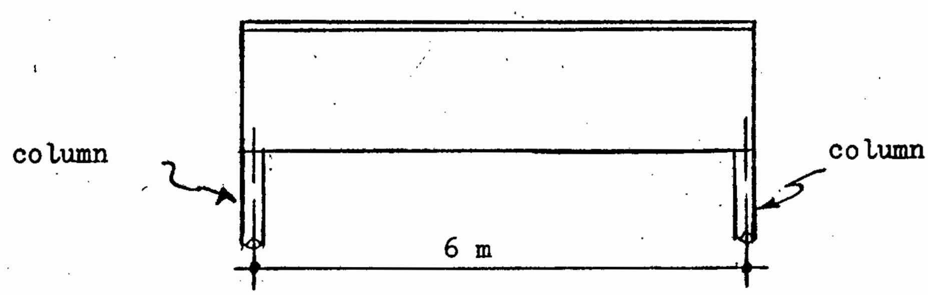
Section Properties:

Beams:	E = 200 GPa	$I = 1,7 \times 10^{-3}\ m^4$	Plate:	E = 200 GPa
	G = 80 GPa	$J = 0,05 \times 10^{-3}\ m^4$		$\nu = 0,3$
		$A = 22 \times 10^{-3}\ m^2$		h = 20 mm

2.



Oblique View



Longitudinal Section A-A

Show what steps are required to determine the displacements, stresses and bending moments in the V-shaped portion only of the roof structure shown.

The horizontal slabs are subjected to a uniformly distributed load of $5\ kN/m^2$.

Note: (1) There is no moment connection between the horizontal slabs and the V-shaped sections.

(2) All slabs are 100 mm thick.

UNIVERSITY OF CAPE TOWN

DEPARTMENT OF CIVIL ENGINEERING

UNIVERSITY EXAMINATION: JUNE 1976

COURSE CE 506 - PROPERTIES OF CONCRETE

Time allowed: 3 hours 5th June, 1976

Part A consists of fifteen multiple-choice questions. Each question is followed by five suggested answers; select the one which is best in each case and circle one of (a), (b), (c), (d) or (e) for each question. This portion of the examination paper must NOT be removed from the Examination Room and must be handed in for marking.
Part B consists of five questions. Answer all questions.

PART A - Multiple-Choice Section (All questions of equal value)

Question A1: In controlling the quality of concrete produced for a project, a test is needed which:

(a) gives the true strength of the material;

(b) gives, for variations in testing procedures, the least variation in results;

(c) gives the true strength of the specimen;

(d) gives a clearly defined stress pattern;

(e) is easy to carry out.

Question A2: In design of concrete mixes according to CP 110 Concrete Structures Code, the target strength chosen is directly related to:

(a) the design strength f_{cu};

(b) the design strength f_{cu} plus 1,65 times the standard deviation 'σ'.

(c) the design strength f_{cu} plus the standard deviation 'σ';

(d) the design strength f_{cu} plus the coefficient of variation 'v';

(e) the design strength f_{cu} plus 1,65 times the coefficient of variation 'v'.

Question A3: The most important aspect of sampling from a pre-mixed concrete truck is to:

(a) protect the sample from wind and sun;

(b) obtain a representative sample in order to carry out further tests;

(c) ensure that the concrete is properly mixed;

(d) check the workability and slump;

(e) obtain a sufficient quantity of concrete to carry out further tests.

/Question A4: ....

Question A4: For a water/cement ratio of 0,6 by weight the use of rounded river gravel in place of crushed aggregate of cubic shape and rough texture will:

(a) show little difference in compressive strength but increase flexural strength;

(b) increase compressive strength by about 10% and also increase flexural strength;

(c) decrease compressive strength by about 10% but increase flexural strength;

(d) increase compressive strength slightly but lower flexural strength;

(e) decrease slightly, both compressive and flexural strengths.

Question A5: The Unit Water Method of Mix Design, described in lectures, suggests that the grading of the combined aggregate be made finer than the recommended grading when:

(a) the maximum aggregate size is larger;

(b) the maximum aggregate size is smaller;

(c) the coarse aggregate is crushed material;

(d) the cement content is higher;

(e) the cement content is lower.

Question A6: An increase in the proportion of aggregate material in the sieve range 2,00 mm to 9,5 mm (No. 8 to 3/8") will tend to:

(a) make the concrete harsh and liable to honeycomb;

(b) make the finishability of the concrete better;

(c) improve the economy of the mix;

(d) increase the amount of water required;

(e) reduce the amount of water required.

Question A7: The addition of an air entraining agent to a concrete mix usually leads to:

(a) a more economical mix;

(b) a stronger concrete;

(c) a decrease in the required sand percentage;

(d) a decrease in cement content;

(e) a denser concrete because of improved workability.

/Question A8: .....

Question A8: In the Unit Water Method of Mix Design, described in lectures, the estimated water content for a particular slump is fixed by:

(a) the maximum size of the aggregate;

(b) the grading of the aggregate;

(c) the shape of the aggregate;

(d) (a) and (b) above;

(e) (a) and (c) above.

Question A9: Capillary water in hydrated cement paste is:

(a) water held in areas of restricted adsorption of the gel structure;

(b) water occupying space beyond the range of surface forces of the solid phase of the gel structure.

(c) water existing in cavities and channels up to 100 times greater than the size of gel pores;

(d) both (b) and (c) above;

(e) water chemically combined such that it is part of the solid matter in the hardened paste.

Question A10: Plastic shrinkage of concrete is caused by:

(a) removal of capillary and gel pore water;

(b) the absorbtion of mixing water by porous or dry aggregates;

(c) sedimentation and settling of solids in the concrete mix;

(d) bleeding of free water to the top surface of the concrete where it is often lost by evaporation or drainage;

(e) all of (b), (c) and (d) above.

Question A11: The secant elastic modulus of concrete is increased by:

(a) increased water:cement ratio and increased paste content;

(b) constant water:cement ratio and increased paste content;

(c) increased water:cement ratio and decreased water content;

(d) constant water:cement ratio and air entrainment;

(e) decreased water:cement ratio and decreased paste content;

Question A13: Decreasing the water/cement ratio influences the ultrasonic pulse velocity because:

(a) poor compaction leads to voids;

(b) a decrease in the density causes the pulse velocity to increase;

(c) an increase in strength (due to a lowering of the water cement ratio) causes the pulse velocity to increase;

(d) an increase in the density causes the pulse velocity to increase;

(e) an excess of paste causes the pulse velocity to decrease.

/Question A14: ....

Question A14: Rapid Hardening Portland cement can be manufactured by:

(a) more finely grinding the Portland cement;

(b) changing the ratio of $C_2S:C_3S$;

(c) intergrinding some high alumina cement with the Portland cement;

(d) both (a) and (b) above;

(e) all of (a), (b) and (c) above.

Question A15: Excessive bleeding of concrete can be corrected by:

(a) adding more cement;

(b) adding crusher dust or other fine material;

(c) by air entrainment;

(d) both of (a) and (b) above;

(e) all of (a), (b) and (c) above

[Total 20 marks]

PART B

Question B1:

(a) A laboratory trial mix of concrete with 30 kg of water, 50 kg of cement, 130 kg of sand and 180 kg of stone gave a 28-day strength which was too low, a slump of 110 mm and real mortar excess of 8%. It is decided that a reduction in water/cement ratio to 0,56 will probably correct the strength requirement. What mix would you suggest for a second trial to give a slump of 60 mm and a real mortar excess of 2% given that the densities of the water, cement, sand and stone are 1000, 3150, 2600 and 2750 kg/m^3 respectively.

(b) The compressive strength of the second trial mix after 28 days' storage at 18°C is 33 MPa. Using Plowman's method, determine how long it would take to reach the same strength at 25°C. What will be the compressive strength after 3 days at 25°C?

[20 marks]

Question B2:

Consider an average structural grade concrete made with 20 mm river gravel aggregate (irregular gravel), normal Portland cement, water/cement ratio (by weight) 0,60 aggregate/cement ratio 6,0, and slump of 75 mm.

(i) Calculate the effect on strength of adding water so as to increase the slump to 150 mm.

(ii) How does this strength change compare with that expected to result from changing from gravel to crushed coarse aggregate but maintaining the aggregate/cement ratio at 6,0 and slump at 75 mm?

(iii) If a graded river gravel with maximum size 80 mm was used in place of the 20 mm gravel, comment on the expected water demand, water/cement ratio and resulting compressive strength of the concrete.

Clearly state the assumptions made in each case.

[15 marks]

Question B3:

(i) Explain briefly how the progressive hydration of cement may lead to self-desiccation of concrete.

(ii) Calculate the gel/space ratio for a concrete with a water/cement ratio of 0,60 at an age of 14 days at which time 60 per cent of the cement had hydrated. Comment on the expected compressive strength corresponding to this gel/space ratio.

(iii) 100 g of cement and 20 g of water are placed in one sealed container and 100 g of cement and 60 g of water are placed in another sealed container. Calculate in both instances the maximum degree of hydration possible, the volume of gel formed, the weight of chemically combined water and the weight of free water in the capillary pores.

[20 marks]

/Question B4: ....

Question B4: "When concrete specimens are loaded axially in compression they always fail in tension". Briefly discuss this statement and go on to discuss the effect of specimen size and shape, and also the effectiveness of capping materials on the apparent ultimate compressive strength of concrete test specimens.

[10 marks]

Question B5: A considerable number of different types of test procedures have been devised to measure "workability" of concrete. Discuss the reasons for the multiplicity of methods used. List ways in which the workability of concrete can be increased without increasing the water content.

[20 marks]

UNIVERSITY OF CAPE TOWN

UNIVERSITY EXAMINATION OCTOBER 1976

NUMERICAL ANALYSIS and COMPUTATION (a)

(NAC(a) ≡ APPLIED MATHEMATICS AM 346)

Time : 3 hours

Not more than FIVE questions to be answered :

1. (a) The iterative procedure

$$x_{r+1} = F(x_r)$$

to find an approximate solution of the equation

$$x = F(x)$$

is known as the 'method of successive substitutions'. Discuss the convergence of this procedure, and illustrate by finding a convergent procedure to solve the equation

$$x + \log_e x = 0 \quad .$$

Taking $x_o = 0{,}5$, find a solution to three decimal places, and check your solution by applying the Newton-Raphson iterative method.

(b) When $f(x)$ has two zeros α_1 and α_2 which are nearly coincident, so that $f'(x)$ is zero at a point β between α_1 and α_2 , show that, if β is found, approximations to α_1 and α_2 are given by

$$\alpha = \beta \pm \left\{ - \frac{2f(\beta)}{f''(\beta)} \right\}^{\frac{1}{2}} \quad .$$

Improved values may then be obtained by the usual methods.

Hence determine initial approximations to two real roots of the equation

$$3x^4 + 8x^3 - 6x^2 - 24x + 18 = 0 \, .$$

(20)

University of Cape Town, University Examination, October 1976
Numerical Analysis and Computation (a) (NAC(a) ≡ Applied Mathematics AM346) continued:

2. Given the tabulated function

x	1	2	3	4	5	6
f(x)	2439	2174	1961	1786	1639	1515

(a) Draw up a difference table

(b) Find $f(4,2)$ using Stirling's interpolation formula

$$f(x) = f_o + \theta\mu\delta f_o + \tfrac{1}{2}!\ \theta^2\delta^2 f_o + \tfrac{1}{3}!\ \theta(\theta^2-1)\mu\delta^3 f_o +$$
$$\tfrac{1}{4}!\ \theta^2(\theta^2-1)\delta^4 f_o + \ldots\ldots$$

(c) Find $f'(3)$ by differentiating Stirling's formula

(d) Find $\int_1^5 f(x)dx$ by integrating Stirling's formula.

Show the relation of the quadrature formula obtained to Simpson's rule.

(20)

3. A two-point Gaussian Quadrature formula has form:

$$\int_{-h}^{h} f(x)dx = h\{a\ f(\alpha h) + b\ f(\beta h)\} + E$$

Evaluate the weighting constants a and b and the position parameters α and β so that the formula is exact $(E = 0)$ for $f(x)$ any polynomial of degree ≤ 3.

Find E when $f(x) = x^4$.

Calculate $\int_{-0,5}^{0,5} x^4\,dx$ (i) using the Gaussian Quadrature formula
(ii) analytically
(iii) using Simpson's rule

$$\int_{x_o}^{x_o+2h} f(x)dx \simeq \frac{h}{3}\{f(x_o) + 4f(x_o+h) + f(x_o+2h)\}$$

Show that if $f(x)$ has a Taylor series expansion about $x = 0$ then

$$E \simeq \frac{h^5}{135} f^{iv}(0).$$

(20)

4. Describe the Taylor series method for the approximate solution of the differential equation

$$\frac{dy}{dx} = f(x,y) \quad ; \quad y(0) = y_o$$

Show that the Simple Runge-Kutta formula

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$
$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf(x_n+h\ ,\ y_n+k_1)$$

provides an approximation to the Taylor series method and find the order of the term in h to which there is agreement.

Compare the approximate solutions obtained by the Taylor series, simple Runge-Kutta and Euler methods for two steps ($h = 0,1$) in the problem:

$$\frac{dy}{dx} = x^2 + y^2 \qquad y(0) = 1 .$$

(20)

5. (a) For the set of equations

$$2x_1 + 5x_2 - x_3 = 7$$
$$4x_1 + x_2 + x_3 = -1$$
$$-x_1 - x_2 + 3x_3 = 2$$

(i) Describe the variations of the Gaussian elimination method, and solve by the complete pivoting method.

(ii) Show that, for one re-arrangement of these equations, the Gauss-Seidel iterative method will work, but for others it will not. Perform two iterations in each of two cases, and compare the results with the exact solution (-1; 2; 1).

(b) For the set of equations

$$1{,}7\ x_1 + 2{,}3\ x_2 - 1{,}5\ x_3 = 2{,}35$$
$$1{,}1\ x_1 + 1{,}6\ x_2 - 1{,}9\ x_3 = -0{,}94$$
$$2{,}7\ x_1 - 2{,}2\ x_2 + 1{,}5\ x_3 = 2{,}70$$

it has been found that there is a solution near (1; 2; 3).

Find an improved solution.

(23)

University of Cape Town, University Examination, October 1976
Numerical Analysis and Computation (a) (NAC(a) ≡ Applied Mathematics AM346) continued:

6.(a) The Apeiron Manufacturing Company has to decide how much finishing to perform on their products prior to sales. They may sell the products as rough casting, semi-finished product, or finished product. Each category requires a different amount of work:

	Casting	Machining	Plating	
Rough Casting	5	1	1	
Semi-finished	5	4	1	
Finished	5	4	2	(in hours)

The cost per hour of production in each of the departments is the same. The profit per unit of sales is:

rough casting R3 ; semi-finished R5 ; finished R6 .

The company can sell all the items which they can produce. The casting department has a daily production capacity of 130 hours, the machining department 86 hours and plating 40 hours.

How many products in each of the three categories should be manufactured to maximise daily profit?

Set up this problem as an LPP and solve it using the Simplex Method.

(b) A manufacturing company has factories in Alberton, Brackenfell and Camperdown which produce 80 , 60 and 30 television sets per week. The sets are to be transported to main warehouses in Parow, Queenstown and Reddersburg, which have weekly demands of 50 , 50 and 70 sets. The cost of transporting one set from each factory to each of the warehouses is given by the table (rand/set)

	P	Q	R
A	9	5	3
B	2	6	7
C	8	4	5

Using the method of iterative improvement of simple solutions, find the least cost of transportation.

(28)

UNIVERSITY OF CAPE TOWN

UNIVERSITY EXAMINATION NOVEMBER 1976

NUMERICAL ANALYSIS and COMPUTATION (b)

Time : 3 hours

Not more than FIVE questions to be answered :

1. For suitable values of the physical constants the heat conduction equation can be written in the form

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} .$$

Find the way in which the temperature distribution across a wall $x = 0$ to $x = 2$ varies with time, given:

the initial condition $u(x,0) = 1000 \sin \frac{\pi x}{4}$

the boundary condition $u(0,t) = 1000 \sin \frac{8\pi t}{3}$

and perfect insulation at $x = 2$.

Consider a mesh with h , the increment in x , $= \frac{1}{4}$ and show that

(i) a particular choice of k , the increment in t , leads to a simple difference equation

(ii) the derivative boundary condition can be dealt with by introducing the points $u_{9,j}$. Show that $u_{9,j} = u_{7,j}$

Calculate some values of u to 3 significant figures.

(20)

Question 2 on Page 2.

University of Cape Town, University Examination, November 1976
Numerical Analysis and Computation (b) (continued):

4. (a) Comment on the mixed congruential method for producing pseudo-random numbers

$$x_{n+1} = \lambda x_n + \mu \quad \text{modulo} \quad P$$

where P is a large integer and x_n, λ, μ are positive integers less than P.

If a computer with word length 36 has a FORTRAN compiler which takes no action on integer overflow, comment on the function

(which has $P = 2^{35} - 1$, $\lambda = 2^{16} + 3$)

```
      FUNCTION DRAND(I)
      I=I*262147
    1 IF(I.LT.O)I=I+34359738367+1
      DRAND=I*1.164153219E-10
      RETURN
      END
```

Explain how you would use a pseudo-random number generator in a program to estimate

$$\int_0^1 e^{x^2}\, dx$$

For what type of definite integral is this approach more useful?

(b) Explain (not in great detail) how the least-squares polynomial fitting technique (as developed in question 3) is extended to produce the following formulae for five-point quadratic smoothing of data:

$$Y_{-2} = \frac{1}{35}(31y_{-2} + 9y_{-1} - 3y_0 - 5y_1 + 3y_2)$$
$$Y_{-1} = \frac{1}{35}(9y_{-2} + 13y_{-1} + 12y_0 + 6y_1 - 5y_2)$$
$$Y_0 = \frac{1}{35}(-3y_{-2} + 12y_{-1} + 17y_0 + 12y_1 - 3y_2)$$
$$Y_1 = \frac{1}{35}(-5y_{-2} + 6y_{-1} \,\ldots\ldots\ldots$$
$$Y_2 = \ldots\ldots\ldots$$

How is this set of formulae applied to a set of 10 experimental values? Illustrate graphically.

(20)

5. (a) Explain how you would find the eigenvalue of maximum modulus of a real, symmetric matrix, and why the process converges.

(b) For the eigenvalue problem $(\underline{A} - \lambda\underline{I})\bar{x} = 0$

given that the matrix $\underline{A} = \begin{pmatrix} 7 & 3 & 3 & 0 \\ -3 & -2 & 2 & -3 \\ 5 & 3 & 1 & -2 \\ 5 & 3 & -3 & 2 \end{pmatrix}$

has eigenvalue = 4 with associated eigenvector

(1 ; -1 ; 0 ; 1)

find, using the orthogonality property $\bar{x}_i^T \bar{x}_j = 0$, a 3×3 matrix $\underline{B}$ from which the other eigenvalues of $\underline{A}$ could be determined.

What would be an alternative method of finding another eigenvalue?

(22)

6. The function u satisfies Laplace's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

at all points within the region $0 \leqslant x \leqslant 4$; $0 \leqslant y \leqslant 4$ and has boundary values

$$u(x, 0) = x^2 + x \quad ; \quad u(x, 4) = 20 - 5x$$

$$u(0, y) = y^2 + y \quad ; \quad u(4, y) = 20 - 5y$$

Using Liebmann's iterative method (and the diagonal symmetry), calculate values for u at unit grid points, correct to the nearest integer.

When is Liebmann's method to be preferred to successive over-relaxation?

(18)

UNIVERSITY OF CAPE TOWN

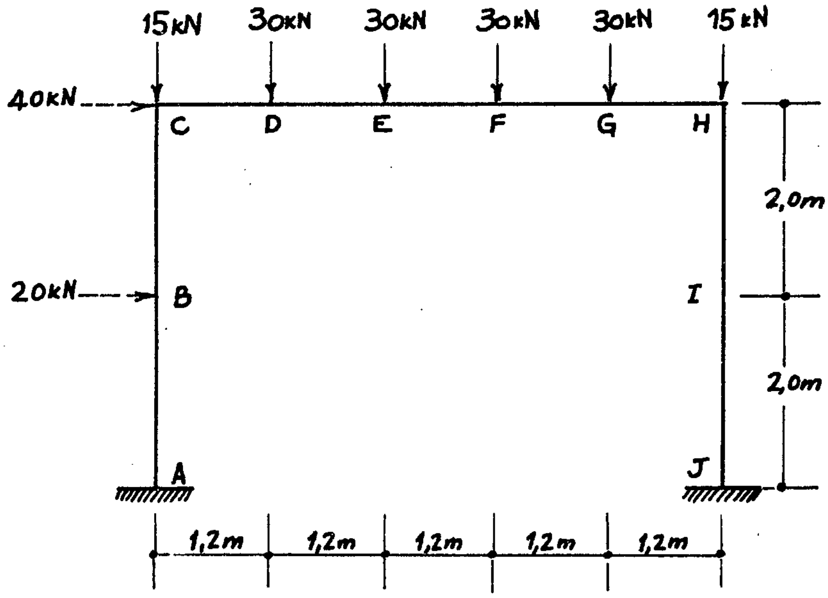
DEPARTMENT OF CIVIL ENGINEERING

UNIVERSITY EXAMINATION: 13th NOVEMBER, 1976

CE 519 - STEEL STRUCTURES

Time allowed: 3 hours

1.



The rectangular frame shown above is to be designed by plastic methods. The loads shown are working loads: the vertical loads represent dead plus superimposed loads and the horizontal loads represent wind loads. The wind loads may act from left to right (at B and C as shown) or from right to left (at H and I).

1. Use limit analysis to determine the least value of M_p for which the frame can equilibrate all factored load combinations using the following assumptions:

   (a) the frame is designed with a uniform section,

   (b) the load factor for dead plus superimposed load alone is 1,75, and for dead plus superimposed load plus wind load 1,4.

   Draw the bending moment and shear force diagrams, and determine the axial loads in the members, for the collapse conditions.

   [40 marks]

2. Using the Abridged Version of the Handbook on Hot Rolled Structural Steel Sections, and the Design Recommendations issued, select an appropriate parallel flange I-section for this design. The yield stress is to be taken as 250 MPa.

   Choose your section/sections with respect to the collapse bending moments, shear forces and axial loads. Consider

   (a) whether the section or sections chosen is/are compact,

   (b) whether shear stiffeners are required,

   (c) lateral stability and the points at which lateral bracing is required,

   (d) in-plane buckling.

   [35 marks]

3. The moments computed from an elastic analysis with uniform E.I. over the entire frame are given below. Tension on the inside of the frame is taken as positive.

Section	Moment due to vertical loads only (kNm)	Moment due to wind only acting from left to right (kNm)
A	+ 27,00	- 70,91
B	- 13,50	+ 1,93
C	- 54,00	+ 34,77
D	+ 18,00	+ 20,36
E	+ 54,00	+ 5,95
F	+ 54,00	- 8,45
G	+ 18,00	- 22,86
H	- 54,00	- 37,27
I	- 13,50	+ 9,95
J	+ 27,00	+ 57,16

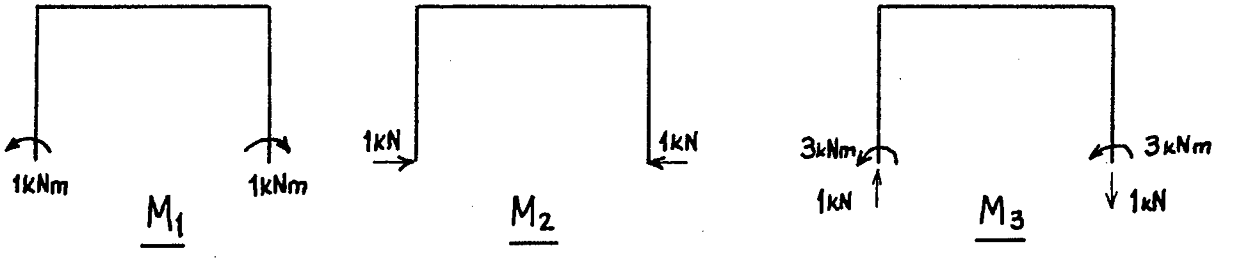
Assume for simplicity that the dead and superimposed load together may or may not act. The wind may act from left to right or from right to left.

For the M_p value calculated in part (a), determine the load factor against failure by alternating plastic deformation at any section.

Do you consider this result to be significant in determining member sizes?

[10 marks]

4.



Using the three independent self-stress systems associated with the force systems shown above, write down the compatibility equations for the structure analysed in 1. above at the point of collapse. Assume plastic hinge rotations at each of the hinges in the mechanism.

You may take the following values for the integrals below:

$$\int M_1 \frac{M}{EI} ds = -0,0076 \text{ kNm}$$

4. (Continued)

$$\int M_2 \frac{M}{EI} ds = - 0,0491 \text{ kNm}$$

$$\int M_3 \frac{M}{EI} ds = + 0,0240 \text{ kNm}$$

M is the collapse bending moment diagram, and moments causing tension on the inside are positive. The integrals extend over the whole structure.

Hence determine which is the last hinge to form.

[15 marks]

14 AUG 1978