39046

National Library
of Canada

Bibliothèque nationale
du Canada

CANADIAN THESES
ON MICROFICHE

THÈSES CANADIENNES
SUR MICROFICHE

NAME OF AUTHOR/NOM DE L'AUTEUR JALAL UDDIN KHANDOKER

TITLE OF THESIS/TITRE DE LA THÈSE "A Computer Program for Three Dimensional Analysis of Buildings"

UNIVERSITY/UNIVERSITÉ Carleton University

DEGREE FOR WHICH THESIS WAS PRESENTED/
GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE M. Eng.

YEAR THIS DEGREE CONFERRED/ANNÉE D'OBTENTION DE CE DEGRÉ 1978

NAME OF SUPERVISOR/NOM DU DIRECTEUR DE THÈSE Professor J. L. Humar

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies
of the film.

The author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission.

*L'autorisation est, par la présente, accordée à la BIBLIOTHÈ-
QUE NATIONALE DU CANADA de microfilmer cette thèse et
de prêter ou de vendre des exemplaires du film.*

*L'auteur se réserve les autres droits de publication; ni la
thèse ni de longs extraits de celle-ci ne doivent être imprimés
ou autrement reproduits sans l'autorisation écrite de l'auteur.*

DATED/DATE Aug 1978 SIGNED/SIGNÉ _____

PERMANENT ADDRESS/RÉSIDENCE FIXE _____

National Library of Canada

Cataloguing Branch
Canadian Theses Division

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada

Direction du catalogage
Division des thèses canadiennes

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

Si il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

A COMPUTER PROGRAM FOR
THREE DIMENSIONAL ANALYSIS OF BUILDINGS

by
JALAL UDDIN KHANDOKER
B.Sc. Engg.

A THESIS
Presented to the
Faculty of Graduate Studies
of Carleton University
in partial fulfillment of
the degree of
MASTER OF ENGINEERING IN CIVIL ENGINEERING

Carleton University

Ottawa, Canada

August 1978

© 1978 Jalal Uddin Khandoker

The undersigned recommend to the Faculty of Graduate Studies
acceptance of the thesis:

"A COMPUTER PROGRAM FOR THREE DIMENSIONAL ANALYSIS OF BUILDINGS"

submitted by Jalal Uddin Khandoker, B.Sc.Engg., in partial fulfillment
of the requirements for the degree of

MASTER OF ENGINEERING

Thesis Supervisor

Chairman, Department of
Civil Engineering

Carleton University

ABSTRACT

A set of computer programs are developed for a linear three-dimensional analysis of building structures consisting of frames and shearwalls for gravity, earthquake and lateral loads. The building is resolved into an orthogonal set of frame or frame-shearwall substructures connected by the floor diaphragms which are considered rigid in their own planes. A frame substructure is composed of horizontal beams and vertical columns or shearwalls. Beams must be prismatic. A column may be prismatic or circular. Flexural and shear deformations are accounted for in all members; for columns, axial deformations are also included. Joints of finite sizes having infinite rigidity are considered in the formulation. The mass of the building is assumed to be lumped at floor levels. The columns and shearwalls are considered to rest on rigid foundations located at the same level.

Two different computer programs are developed for the purpose. The first program, called AIDS, is a conversational computer program designed to act as a 'front end' processor to the main analysis program and reads all necessary input data. The data can be input in 'free format' mode with a high speed input option. The program then rearranges the data and stores it in disc files to be used finally by the second program.

The second program, called ANGELS, is designed to perform the analysis. A maximum of four independent gravity load cases, three differ-

ent lateral load cases and a design response spectrum loading may be considered in the analysis. The above loads can be weighted and combined at the user's option for the purpose of analysis. Any number of such combinations are possible. Also, at the user's option, the structural mode shapes and frequencies are evaluated and printed out. Design story shears for spectrum loading may also be obtained.

The programs are written in standard FORTRAN IV language and designed to run on any computer accepting FORTRAN IV. During the process of development, the program AIDS was tested on an IM-70 minicomputer and the created data files were electronically transferred to a HONEYWELL Sigma-9 computer. The program ANGELS designed to run in a batch mode was tested on a Sigma-9 computer using the input data generated by the program AIDS.

ACKNOWLEDGEMENTS

From his supervisor, Professor J.L. Humar, the author has received invaluable assistance in the form of encouragement and advice. This he acknowledges most gratefully.

The author is also grateful to Dr. E.W. Wright for the fruitful discussions and advice during this work.

The author is thankful to the Public Works, Canada, for allowing him to use their subroutine REED and a portion of the computer program SLAM.

The author wishes to express thanks to his wife, Mrs. Nasreen Khandoker, for her help and inspiration during the work.

The fellowship and the graduate assistantship awarded to the author by Carleton University are gratefully acknowledged.

Thanks are also extended to Mrs. Sandra Ryan for her care in typing the manuscript.

TABLE OF CONTENTS

	<u>PAGE</u>
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF SYMBOLS	ix
CHAPTER 1. INTRODUCTION	1
1.1 General	1
1.2 Computer Programs for Structural Analysis	2
1.3 Conversational Computer Programs	6
1.3.1 Free Format Data Input	7
1.3.2 Timesharing on a Computer	9
1.3.3 Development of Conversational Programs	10
1.4 Earthquake Analysis of Buildings	12
1.5 The Objectives and Scope of Present Work	13
1.6 The Format of the Thesis	18
CHAPTER 2. STRUCTURAL IDEALIZATION	20
2.1 Basic Assumptions	20
2.2 Structural Modelling	21
2.3 Frame Substructure	24
2.4 Member Stiffness	28
2.4.1 Beam Stiffness	28
2.4.2 Column Stiffness	32
2.5 Frame Stiffness	35
2.6 Frame Sidesway Stiffness	38
2.7 Assembly of Building Stiffness Matrix	40
2.8 Solution Static Loads	45
2.8.1 Gravity Loads	45
a. Beam Fixed End Forces	45
b. Frame Joint Force Vector	47
2.8.2 Lateral Loads	48
2.8.3 Evaluation of Member Forces	51

	<u>PAGE</u>
CHAPTER 3. ANALYSIS FOR EARTHQUAKE MOTION	54
3.1 Formulation of Equations of Motion	54
3.2 Mode Shapes and Frequencies	58
3.2.1 Matrix Iteration Method	59
3.2.2 Inverse Power Method with Shifts	66
3.3 Transformation of the Equations of Motion	69
3.4 Spectrum Analysis	71
3.5 Design Response Spectrum	73
CHAPTER 4. DESCRIPTION OF PROGRAMS	75
4.1 General	75
4.2 Mathematical Description of the Building	76
4.3 Input Data Program - AIDS	78
4.4 Analysis Program - ANGELS	85
4.4.1 Basic Function	85
4.4.2 Organization of the Program	87
4.5 Efficiency of Algorithms	91
4.5.1 Generation of Frame Stiffness Matrices, Subroutine STIFF	94
4.5.2 Solution of Simultaneous Equations	97
4.5.3 Condensation of Frame Stiffness Matrices	100
4.5.4 Solution of Eigenvalue Problem	104
4.5.5 Optimization of Computer In-Core Storage Space	106
CHAPTER 5. SUMMARY AND CONCLUSIONS	109
5.1 Summary	109
5.2 Conclusions	112
5.3 Recommendations for Further Development	113
REFERENCES	117
APPENDICES	119
A.	
Chapter A1. INTRODUCTION	1
A1.1 General	1
A1.2 Assumptions and Limitations	2
A1.3 Numerical Description of Building	4
A1.4 Sign Conventions and Units	9
A1.5 Job Control Commands	11
Chapter A2. PROGRAM FOR INPUT DATA - AIDS	18
A2.1 Description of Program	18
A2.2 Instructions for Data Input	22
A2.3 Definition of FORTRAN Symbols	26
A2.4 Example of Data Input	32
A2.5 Data Output	52

	<u>PAGE</u>
CHAPTER A3. ANALYSIS PROGRAM - ANGELS	60
A3.1 Function	60
A3.2 Basic Operations	61
A3.3 Definition of FORTRAN Symbols	66
A3.4 Input Data and Formats	81
A3.5 Output from the Program	90
A3.6 Example of Analysis	90
B.	
CHAPTER B1. FORTRAN LISTING OF PROGRAM-AIDS	1
CHAPTER B2. FORTRAN LISTING OF PROGRAM-ANGELS	51

LIST OF FIGURES

<u>FIGURE</u>	<u>TITLE</u>	<u>PAGE</u>
2.1	Typical frame and shearwall building.	22
2.2	Elevation of a setback frame.	25
2.3	Mathematical model of a frame.	27
2.4	Beam model with rigid end stubs.	29
2.5	Flexural beam deformation.	29
2.6	Transformation relation between beam deformations and frame displacements.	29
2.7	Flexural column deformations	33
2.8	Transformation relation between column deformations and frame displacements.	33
2.9	Frame degrees of freedom.	36
2.10	Frame stiffness matrix.	36
2.11	Global and local coordinates of a typical floor.	44
2.12	Fixed end actions for gravity load.	46
2.13	Joint forces for gravity load.	46
2.14	Plan of nth floor of a building with lateral loads.	49
2.15	Beam end forces for joint deformations.	49
3.1	Ground and structure displacements	56
3.2	Measurement of eigenvalues from shifted origin.	68
3.3	Elastic average response spectrum for 1.09 maximum ground acceleration.	68

<u>FIGURE</u>	<u>TITLE</u>	<u>PAGE</u>
4.1	Reinforced concrete building under gravity, earthquake and lateral loads.	80
4.2	Outline flow diagram for program AIDS.	84
4.3	Outline flow diagram for operation of program ANGELS.	92
4.4	Program organization and file management for program ANGELS.	93
4.5	Storage of stiffness matrix of a multistoried frame.	96
4.6	Condensation of stiffness matrix	101

LIST OF SYMBOLS

a	length of rigid zone at the left end of a beam or at the bottom of a column
$\{a\}$	eigenvector
A	area of cross-section of a member
$[A]$	coefficient matrix
A_i	value of the largest element of the i th trial vector in the iteration of a mode shape
$[A^m]$	transformation matrix for the m th frame
b	length of rigid zone at the right end of a beam or at the top of a column
$\{b\}$	vector of known components (forces)
$[B]$	matrix of mode shapes
$[C]$	damping matrix
C_n	multiplicative constants in the expansion of trial vector
d_n^m	perpendicular distance to the m th frame from global origin at level n
$[D]$	dynamic matrix
$[D]_i$	dynamic matrix modified so that the iteration converges in the i th mode shape
$[\bar{D}]$	dynamic matrix formed when origin is shifted
DX_n	x-distance of point of action of lateral load at level n from global reference point
DY_n	y-distance of point of action of lateral load at level n from global reference point
E	modulus of elasticity

$[f]$	member flexibility matrix
F	member end actions due to joint displacement only
F_1^k	moment at joint k due to gravity load
F_2^k	vertical load at joint k due to gravity load
FM	final end actions of a member
FX	fixed end forces on a beam due to gravity load
$\{F\}$	frame joint load vector
$\{F_B\}$	vector of frame story forces
$\{F_g\}$	global load vector for lateral loads
$\{FH_n\}$	lateral load vector at level n in global origin
G	modulus of rigidity
HX_n	x-component of lateral load at level n
HY_n	y-component of lateral load at level n
I	moment of inertia
$[I]$	identity matrix
k_{AA}	submatrix of the frame stiffness matrix with elements not corresponding to sidesway
k_{BB}	submatrix of the frame stiffness matrix with elements corresponding to sidesway
$[k_b]$	beam stiffness matrix considering bending degrees of freedom only
$[k_c]$	column stiffness matrix considering bending and axial distortions
$[k^m]$	condensed stiffness matrix of mth frame in the local coordinates
$[k^m]$	condensed stiffness matrix of mth frame in global coordinate system

$[k_{ij}]$	a submatrix of $[k^m]$
K_*	frame sidesway stiffness matrix
$[K]$	stiffness matrix
$[K_B]$	beam stiffness matrix
$[K_C]$	column stiffness matrix
$[K_F]$	frame stiffness matrix
$[K_g]$	global stiffness matrix
L	clear length of a member between rigid zone ends
L_b	row number of first non-zero entry in the load vector
L_j	row number of first non-zero entry in column j of the coefficient matrix
$[M]$	global mass matrix
N	number of stories in the building
NC	number of columns in a frame
NE	number of elements of a frame stiffness matrix required to be stored
NF	number of frames in the building
NM	number of modes to be considered
NP	number of period coordinates required to define the response spectrum
NT	number of towers in the building
NB_i	number of bays in tower i
NS_i	number of stories in tower i
P	concentrated load acting on beam
$P(t)$	time dependent forcing function

Q_i	modal participation factor for i th mode
$\{r\}$	vector of pseudostatic displacements along the global coordinates due to a unit displacement of the base in the direction of the earthquake motion
S_a	spectral acceleration
S_D	spectral displacement
S_v	spectral velocity
$[S]_i$	matrix for sweeping mode shapes up to the i th mode shape from a trial vector
T	period
$[T_B]$	beam transformation matrix
$[T_C]$	column transformation matrix
T_D	period defining the boundary between the velocity bound and the acceleration bound on the design response spectrum
$\{u\}$	mode shape
$\ddot{v}_g$	time dependent acceleration of the ground motion
v_n	displacement vector at n th floor in global coordinates
v_n^m	displacement in local coordinates of the m th frame at level n
$\{\dot{v}\}$	displacement vector
$\{v_a\}$	vector of absolute displacement along global coordinates with reference to a fixed frame in space
$\{\dot{v}\}$	velocity vector
$\{\ddot{v}\}$	acceleration vector
$\{v_g\}$	vector of pseudostatic displacements along the global coordinates resulting from displacement of the base due to the earthquake motion
w	load distributed uniformly on a beam

$\{x\}$	vector of unknown displacements
x_n	displacement of floor at level n in global x -direction
$\bar{x}_n$	x -distance of global origin or center of mass from reference axis at level n
y_n	displacement of floor at level n in global y -direction
$\bar{y}_n$	y -distance of global origin or center of mass at level n from reference axis
z_i	i th normal mode coordinate
$\{z\}$	normal displacement vector
β	angle of incidence of earthquake with global x -axis
ϕ	member deformation
$\{\phi_i\}$	i th mode shape
ξ_i	damping ratio in the i th mode
λ_0	shifted origin in the extradiation of eigenvalue
λ	inverse of the square of frequency
μ	ductility factor
μ_i	square of the i th residual eigenvalue measured from a shifted origin
ω_i	i th natural frequency
ψ	phase angle
θ	joint deformation
θ_n	rotation about the global origin at level n
θ_A	displacement vector not corresponding to sidesway
θ_B	displacement vector corresponding to sidesway
$\{\theta\}$	frame displacement vector
$\{\theta_g\}$	global displacement vector

CHAPTER 1INTRODUCTION1.1 GENERAL

A real building is a three-dimensional continuous system with complex structural components and distribution of loading. However, for the purpose of analysis, it is a common practice to resolve the building into a series of plane frames so that the problem is reduced to one in two dimensions. These approximations may give satisfactory results only in cases where the building is symmetrical in nature and has uniform distributions of mass and stiffnesses. For practical reasons, real buildings are very often unsymmetrical and have some irregularities in their structural shapes and uneven distributions of loads and structural properties. Analysis of such buildings by the plane frame approach is likely to give inaccurate results, which may lead to irrational design. Therefore, for many practical buildings, a three-dimensional analysis in which the structure as a whole is considered to take part in resisting the forces is essential.

On the other hand, a very rigorous approach to three-dimensional analysis would involve a large number of unknown variables and thus would necessitate the solution of a great many number of equations. Only a few buildings, which are very irregular and have an extremely complex distribution of loading, may need such a sophisticated analysis. Most

buildings, even when they are complex, usually have a comparatively simple geometry and consist of right angle frames or frame-shearwall systems. With the aid of some basic reasonable assumptions, it is possible to reduce the number of unknowns considerably, still considering the building as a whole for the analysis. The results obtained by such an approach are quite accurate for most practical purposes. This makes the analysis simpler and suitable for normal design office use. The task is however still too formidable for manual computations and suitable computer programs must be developed for the analysis.

1.2 COMPUTER PROGRAMS FOR STRUCTURAL ANALYSIS

There are several general purpose computer programs that can carry out two and three-dimensional linear analysis of complex structures. Most of these programs can be used for the analysis of buildings. However, there are several disadvantages in the use of a general structural analysis program for a building system. Some of the major difficulties are as follows:

1. Most buildings are of simple geometry with horizontal and vertical members. Also, many frames and shearwalls are typical and repeated in the structural layout. Most general purpose programs do not recognize these facts. Therefore, on one hand they are inefficient for the purpose and on the other they need an input data which is often unnecessarily complex and voluminous.

2. The general purpose analysis programs will also not recognize that the floor diaphragms are very rigid in their own planes (this is true of most well designed buildings). If this assumption is not made, the resulting number of equilibrium equations may be very large. This may lead to a very large increase in the computational effort. Also, numerical errors may be introduced because the inplane floor stiffnesses are several orders of magnitude greater than the story to story stiffnesses. Since these two stiffnesses are added in a direct stiffness approach, double precision may be required in the solution, and even then the numerical difficulties may not be completely eliminated.
3. In the dynamic analysis of buildings, a reasonable approximation can be made whereby the mass of the structure is lumped at the floor levels. With this assumption, the eigenvalue problem is reduced to a reasonable size and results obtained are sufficiently accurate for all practical purposes. A general purpose analysis program, not specifically designed to solve a building structure, may not utilize the above approximation.
4. In a building, the loading is of a regular form and certain loads are applied only at a limited number of locations. The simplifications resulting from this may not be exploited in a general purpose program. Also, in many programs, there may be no simple and direct method of satisfying various loading

requirements such as, for example, the various combinations of gravity, wind and earthquake loadings and such requirements may be satisfied only at the cost of increased computational effort.

5. It is desirable to have the computer output categorized as pertaining to a frame, story, column, or a beam and presented accordingly. Most general purpose programs can not supply the output in a convenient format.

The above drawbacks cause difficulty for the user. Also, general purpose programs usually entail a high running cost. To overcome these difficulties, programs which are specially dedicated to the analysis of building structures need to be designed. There are, of course, some special purpose programs available for analysis of only building structures. However, the majority of them are problem oriented; for example, some may be able to analyze the structure for static loads only, some others may be able to perform only a dynamic analysis of the building⁽⁸⁾ and so on. Some programs have other limitations. Many of them can not directly handle setback frames; they usually put zero properties for members above the setback levels, increasing computation cost and computer storage requirements. Also, many such programs do not take into account shear deformations in structural members which may be significant in some cases. Thus, the need for a more sophisticated special building analysis program is apparent.

+ Numbers in round parentheses refer to the references listed at the end of this thesis.

A few special purpose programs which can effectively perform a three-dimensional analysis of buildings for various loading conditions have been developed in recent past. Notable among them is TABS⁽²⁰⁾. Program TABS performs a three-dimensional analysis of buildings on the basis of assumptions similar to those discussed above and hence is not beset by many of the difficulties experienced in the use of general purpose programs.

The program reported in this thesis has more or less the same capabilities as TABS. In its internal organization, however, it is completely different. This program uses computational techniques based on the up-to-date state-of-the-art. The techniques will be detailed in the subsequent sections of this thesis; they relate mainly to the storage of stiffness matrix, solution of equations, static condensation of matrices and the solution of eigenvalue problems. To the user, however, the major difference is in data input. Unlike TABS, the present program affords the facility of completely free format data input through an interactive conversational 'front-end' processor.

1.3 CONVERSATIONAL COMPUTER PROGRAMS

To date, most computer programs that handle civil engineering problems have been designed for processing in the batch mode. Most of them are written in FORTRAN IV and in some of them (e.g. ICES STRUDL, AMECO, etc.), the communication between the user and the program is in a special problem-oriented language. In order to work with them, the user must put substantial effort in learning the language of communication or in mastering the data input format. Also, one of the troublesome features of standard FORTRAN is the rigid nature of input format. If numbers or 'alphanumeric' variables are positioned incorrectly on data cards or in the input data files, the computer interprets the information incorrectly. The user is thus forced to spend time to make certain that the data on his punched cards are exactly in the specified position. If the structure to be solved is big, the number of such data cards are equally large and the time spent on scrutinizing them may be considerable. After preparation of the data, the user submits it to the data center for 'batch execution' and has to wait for several minutes to several hours before he receives the printed output.

The difficulties just cited in using engineering programs are obviously a cause of frustration to the user-engineer and of reluctance in using these programs frequently.

Computer programs have been developed which overcome at least some of the above difficulties. These programs have one or more of the following attributes:

1. Data input is made in a 'free format' mode.
2. A remote 'time sharing' terminal is used to run the programs instead of resorting to batch execution.
3. An interactive conversational mode of communication is established between the user and the computer during the execution of a program.

The above points are discussed below in some detail.

1.3.1 Free Format Data Input

It is possible to devise an interfacing FORTRAN subroutine which will accept data in essentially 'free format', and convert it into integers, real numbers, or alphanumeric variables as appropriate. This makes it possible to avoid punching data in the correct position on data cards or on input data files. Instead, one only needs to separate individual data items by one or more blanks. One such general purpose subroutine, called 'REED' has been developed by Wright⁽¹⁸⁾ and is in extensive use. The subroutine operates in the following steps:

1. A line (or card) of data input is read in A1 format and stored in an array.
2. The array of data just read is processed character by character. Strings of consecutive non-blank alphanumeric characters are recognized as discrete data items, that is, a group of one or more consecutive blanks is recognized as a data item separator.

3. If a particular discrete data item, so defined, contains only digits and possibly plus, minus, or decimal signs in appropriate locations, it is recognized as a potential numeric data item and its value as a real number is computed. If it contains any other character, it is considered as not being a numeric data item.
4. From the unprocessed alphanumeric input array, the subroutine constructs the real numeric input array, the integer input array, and the processed alphanumeric input array. The program then selects the desired data items from the desired array.

The program can accept numeric data in a very easy and natural way. The input is truly free format; no decimal is required for an integer; also, when real numbers are whole numbers, as is often the case, they can be supplied as just the numbers, the decimal being optional. By contrast, if the user by mistake follows an integer by a decimal, the program is not disturbed thereby. Also, if by mistake, a user types in the letter O as the digit zero and the letter I as the digit one, the subroutine can automatically process them as the desired numerals. A more detailed description of the subroutine is available in References (18) and (19).

Since the subroutine is written in standard FORTRAN, it is machine independent and can be used on any computer accepting FORTRAN.

1.3.2 Time-Sharing on a Computer

To paraphrase Wright⁽¹⁷⁾, the idea of time-sharing on a computer is to commute computer resources in time (time-sharing) from one user to another. In contrast, a batch mode of operation forces a user to commute from one computing activity to another between computer runs. The concept of time-sharing arises from the fact that the human thinking process and responses are rather slow relative to the logical and arithmetic capability of the computer creating a time lag between the two systems (the human and the computer). This time lag has been properly utilized to switch the computing resources from one user to another in such a way that the user can interact from a terminal on line to the computer and think he has the sole access.

The intention of time-sharing, however, is not the reduction of the programmer's frustration but a new mode of machine-aided problem solving. Of course, in this mode, a user can sit down at a console and in one or more continuous sessions write, debug, and run his programs interactively. In this way, one feels more imaginative and can usually maintain a better contact with his problem during its execution. Also, using the system file storage facilities, information can be updated and retrieved on demand.

The very common features of time-sharing are as follows:

1. The creation, editing and accessing of data files.
2. The development, editing and debugging of programs.
3. The execution of small and medium jobs.

A more detailed account of time-sharing mode of operation can be found in References (21) and (17).

With the development of time-sharing systems, new types of computer programs which utilize this facility provided by the computer have been written and are in successful use.

1.3.3 Development of Conversational Programs

With the development of a time-sharing system in computers, one of the areas that has received considerable attention has been that of conversational computer programs. The role of such a program may be described as follows.

With a conversational computer program, the user in effect carries on a dialogue (a conversation) with a host computer via an appropriate time-sharing terminal. To accomplish this, the user at first establishes contact with the computer and requests it to execute a desired program to solve a particular problem. The computer replies via the terminal first by announcing that the desired program is about to be executed. Then it starts asking a series of auto-explanatory questions one by one, via the terminal and invites the user to supply an immediate answer at the terminal. Finally, when all relevant information (i.e. the input data) has been supplied, the computer provides a solution to the problem on the user's terminal.

The important features of a conversational program are at once obvious. First, it requires from the user only a rudimentary knowledge

of computer use. Second, the user does not have the need to go through a program user's manual or to learn a complicated set of instructions for the execution of the program.

Again, since the conversation takes place via a time-sharing terminal, an essential aspect of a conversational program is a free-format data input. For a structural problem of some size, a large volume of data input is required by the program. However, for a real building, many members are typical and repetitive over the whole structure, and may carry loads that are similar. Also, many of the dimensions are repetitive. It is possible to write an efficient conversational program which takes advantage of the above features and makes it possible for the user to supply data at a high speed using the free-format facility of the program.

In addition to providing engineers and users with a range of computer-based tools for quick and direct problem solving, conversational programs can be usefully employed to prepare the input data for many of the larger compute-bound programs which must be run in a remote batch mode.

For many other applications, conversational programs can offer improved effectiveness and very substantial savings in time even to the engineer experienced in computer use. The above advantages of the conversational computer programs appear to be immediately obvious and make them very effective for use in the engineering offices.

The role of conversational programs is described in more detail in Reference (19).

1.4 EARTHQUAKE ANALYSIS OF BUILDINGS

At the present time, most buildings in regions of Canada with a possible seismic hazard are designed based on the equivalent static lateral loads specified in the National Building Code⁽¹⁴⁾. In this approach, the magnitude of the design base shear depends on two basic parameters: the fundamental period of the building and the seismic zone in which it is located. An approximate formula may be used to estimate the fundamental period. The suggested distribution of the lateral loads over the height of the building is based on the assumption that the predominant response is in the fundamental mode, though an approximate adjustment is made to account for the effect of higher modes. While this approach may be satisfactory for buildings that are regular in distribution of mass and stiffness, it has serious limitations when applied to the design of buildings with complex structural configurations. Thus, for buildings that may have unusual mass or stiffness distributions, major setbacks, or large torsional eccentricities, a more comprehensive dynamic analysis should be carried out to obtain reasonable estimates of the earthquake induced forces. The commentary to the National Building Code⁽³⁾ recommends a dynamic analysis approach to the design of such buildings. In the recommended procedure, the structure is analyzed for a design earthquake base motion represented by an average response spectrum,

which is appropriate to the estimated peak ground acceleration at the site and the damping in the structure. The calculated response is then modified for plastic behaviour in a manner which takes into account the capability of the structure for inelastic deformation.

The computer program described in this report has the ability to perform a dynamic analysis of the building following the approach given in the commentary to the code.

1.5 OBJECTIVES AND SCOPE OF PRESENT WORK

The work reported in this thesis involves the development of computer programs for linear three-dimensional analysis of building systems for the effect of gravity, earthquake and lateral loads. Effort has been made to incorporate into these programs many of the desirable attributes described earlier in this Chapter. The following two computer programs have been developed:

1. The first program called AIDS has been developed to read all relevant information and the input data required for the analysis of the building in a conversational mode and with a 'high speed' data input option.
2. The second program called ANGELS is designed to accept the input data prepared by AIDS and to carry out a linear three-dimensional analysis of the building structure for gravity, earthquake and lateral load systems.

The scope and limitations of the programs mentioned above are summarized here. More detailed descriptions of some of these appear in the subsequent sections.

1.5.1 The Input Data Program - AIDS

This program written in standard FORTRAN IV is essentially a 'front end' processor for the other program ANGELS. It is designed to read all input data in a conversational mode and to prepare it for use by ANGELS.

The important features of this program are as follows:

1. With the aid of general purpose subroutine 'REED'⁽¹⁸⁾, the program reads all data in a true free-format mode.
2. For a big building, the volume of the actual input data is large. However, in most cases, various dimensions, member properties and sizes and loadings etc. are typical and repeated over the structure. The program has a rapid data input facility which makes it possible to greatly speed up the process of reading repetitive data.
3. While the data is being read from the terminal in a conversational mode, a copy of the input supplied by the user is written on a disc file for possible subsequent corrections of input data.
4. The data read with a rapid input option is interpreted by the program rearranged and written on three separate disc files - the first containing the general data and the seismic

loading data, if any, the second containing the member and frame properties data, and the third containing the loading data frame by frame.

As stated earlier, the program acts only as a 'front end' processor for reading in data and does not carry out any other computations.

1.5.2 The Analysis Program - ANGELS

This program is designed to read input data from the three disc data files generated by program AIDS, perform a linear three-dimensional analysis of the building and print the results in terms of member forces. The program is written in standard FORTRAN IV and runs in a remote batch mode. The scope and limitations of this program are as follows:

1. The building to be analyzed should consist of right angle frame or frame-shearwall substructures, each containing horizontal beams and vertical columns or shearwalls. Frames with identical properties are grouped together, each group being called a 'Frame Group'.
2. The joints in a frame are considered rigid and the base columns and shearwalls are considered to be attached to rigid foundations at the same level.
3. The beams are considered axially rigid and the floor diaphragms infinitely rigid in their own planes. Bending distortions are accounted for in both beams and columns; in addition, for

columns, the axial deformations are also included. Shearing distortions and finite joint regions of infinite rigidity may be included at the user's option.

4. For three-dimensional analysis, since the floor diaphragms are considered infinitely rigid in their planes, there are only three degrees of freedom - two translations in the x and y directions and a rotation at each floor of the building. However, for symmetric buildings with symmetric distribution of mass and lateral loads, the analysis can be accomplished by a superposition of a couple of two-dimensional analyses, one in each principal direction. In appropriate cases, if the user so wishes, the program will carry out such two-dimensional analyses.
5. The mass of the building is assumed to be lumped at the floor levels and only translational inertia forces are considered.
6. The program performs a dynamic analysis using an earthquake design response spectrum according to the method discussed in Section 1.4. The three-dimensional mode shapes, frequencies and story forces are evaluated and printed out at the option of the user. Also, if required, the root-sum-square forces are computed at the member level.
7. The program can carry out an analysis for a maximum of four different gravity load cases, three different lateral load cases and a seismic load case for a specified design response

spectrum. The individual load cases can be weighted and superimposed in any number of load combinations and the member forces can be obtained for each such combination. A load combination is referred to as a 'Load System'. For each load system, the multipliers for individual component load cases must be specified by the user.

8. Any of the gravity, lateral or seismic load cases may be suppressed from analysis. The three-dimensional mode shapes and frequencies can be evaluated at the user's option, even when the seismic load case is not considered.

Briefly, the following steps are taken to improve the efficiency of the program as a whole:

1. The frame stiffness matrices are stored in a compacted column string storage to minimize storage space requirements.
2. An efficient algorithm is used⁽²²⁾ to solve the stiffness equations. It works on a stiffness matrix stored as described in step one above and uses the Crout reduction technique. An adaptation of the same technique is used in the static condensation of the stiffness matrix, required for obtaining the frame sidesway stiffness matrix.
3. Because of the lumped mass idealization, the mass matrix obtained is diagonal. An efficient eigenvalue solution algorithm, which takes into account the special character of the mass matrix, is used.

4. A dynamic storage allocation scheme is used for major variable arrays. In this method, storage is allocated at run time and because many arrays can share a common storage, the minimum amount of computer storage is used.

During program development, the input data program AIDS was run on a dedicated IM-70 minicomputer with a terminal capable of communicating with the computer at the rate of 960 characters per second. The three disc data files produced by program AIDS were transferred to a HONEYWELL Sigma-9 computer as three 'unkeyed' disc files. These files formed the input data files for program ANGELS which was run in remote batch mode.

1.6 THE FORMAT OF THE THESIS

Details of the present work are reported in this thesis according to the following format.

Chapter 1 of the thesis has presented a brief discussion of the concepts related to the work and has specified the objectives of the project.

The assumptions made and the limitations inherent in the method of analysis used here are presented in Chapter 2. The theoretical background to the method of analysis and the mathematical modelling of the structure are also presented there.

The theory used in the dynamic analysis of the structure is developed in Chapter 3.

Chapter 4 presents a description of the computer programs developed for this project. The logic and the organization of the programs are briefly discussed and outline flow diagrams are given.

Chapter 5 presents a summary of the work done and discusses the various findings. Possible further developments on the subject are also outlined.

The program user's manual is contained in Appendix A.

The FORTRAN listings of the computer programs are presented in Appendix B.

CHAPTER 2

STRUCTURAL IDEALIZATION

2.1 BASIC ASSUMPTIONS

An exact three-dimensional structural analysis, that is, a space frame analysis, is required for only a limited number of buildings. For a majority of buildings, the following two basic assumptions can be made in formulating the mathematical model of the structure. These assumptions significantly reduce the computational effort and greatly simplify the preparation of input data.

1. It is assumed that the floor diaphragms are rigid in their own planes. This is a realistic approximation for a large number of common buildings. As a result, the structure has three degrees of freedom at each floor level - two translational in the x and y directions and a rotational about the vertical axis of the floor.

The other degrees of freedom include: the bending deformations in the horizontal beams and floor slabs, and a vertical displacement and a rotation at each column or shearwall at each floor level.

2. It is further assumed that the horizontal lateral loads act at the floor levels and are transferred to the columns through rigid floor diaphragms. Also, the mass of the building is

assumed to be lumped at the floor levels. As a consequence, the active dynamic degrees of freedom at each floor are three, corresponding to the three displacement degrees of freedom as mentioned in 1 above.

2.2 STRUCTURAL MODELLING

The elevation and plan views of a typical building are shown in Fig. 2.1. The complete building structure is modelled as consisting of two sets of planar frames or frame-shearwall systems which are orthogonal to each other in plan view. Isolated shearwalls within a set are considered to be frames consisting of a single continuous column line. Each frame is treated as an independent substructure. A substructure is permitted to have two degrees of freedom per joint (vertical displacement and in-plane rotation) and one lateral translation per story. However, of all the above degrees of freedom, the only active dynamic degrees of freedom are those associated with lateral displacements at the floor levels. It is possible to eliminate the inactive joint degrees of freedom by a static condensation of the stiffness matrix before the Global stiffness matrix is assembled. Thus, a typical constituent frame-wall system of story N will bring forth a condensed or sidesway stiffness matrix of size N by N . The substructure stiffness matrix is assembled into the global matrix on the assumption that the structure has three active degrees of freedom per floor: two translational, and one rotational.

(a) PLAN

(b) ELEVATION

FIG.2.1 TYPICAL FRAME AND SHEARWALL BUILDING

The increase in the stiffness of a shearwall due to the presence of an intersecting shearwall can be accounted for in any reasonable manner. However, in the present structural modelling, it is assumed that a frame-wall substructure does not have any stiffness in a direction perpendicular to its own plane, nor does it have any torsional stiffness about its own axis.

The following additional approximations are also inherent in the present approach:

1. Complete compatibility with regard to joint displacements is not enforced at joints which are common to two intersecting frames. Thus, in a column which is common to two intersecting frames, the axial deformations will not be the same. However, for design purposes, a reasonable approximation to the axial load in such a column can be obtained by adding those calculated for the component frames. As for joint rotations, since the intersecting frames are assumed to be perpendicular in plan view, the in-plane joint rotations of the frames are orthogonal to each other and are hence uncoupled.
2. As the floor diaphragms are assumed to be rigid in their respective planes, it is apparent that the axial deformation is not permitted in the beams. However, bending stiffness of the floors may be included approximately in modelling of individual frames. All beam center lines in a floor are assumed to be coincident with the central plane of the floor diaphragm.

Similarly, columns for all stories, located in one vertical stack must have the same center line.

3. The torsional resistance of any member section is neglected.

2.3 FRAME SUBSTRUCTURE

A frame substructure may be a beam and column system, a frame-shearwall system or an isolated shearwall. Any of the above substructures may be regular or setback. The cross-section of a typical setback frame-wall substructure is shown in Fig. 2.2. A frame in which the number of bays in the base floor is larger than those in some upper floors is called a setback frame. The part of the frame height having the same number of bays is defined as a tower; the lowest being called the base tower. In modelling the frame, it is assumed that the center lines of all columns in a stack lie in the same vertical. The assumption will, however, result in some errors when an upper column center line is offset with respect to the center line of the column below it, but such errors will be reasonably small and can be neglected. The above assumption also implies that the bay widths are constant throughout the height of the building and are measured between the center lines of the adjacent column stacks.

It is further assumed that the center lines of all beams at a floor lie in one horizontal plane. This assumption will result in errors in the analysis when the beams in a floor are of different depths, but such errors are likely to be small. The story height is measured from the common center line of beams in one floor to the same in the next floor.

- Tower : part of frame height with same no. of bays
 NST : no. of stories in a tower
 NBA : no. of bays in a tower
 NBS : no. of bay-setback between two towers at left

FIG. 2.2 ELEVATION OF A SETBACK FRAME

Fig. 2.3 compares a real frame and its mathematical model on the basis of the assumption stated above.

Deformations within the joint may be neglected by specifying finite beam and column widths. When finite joint sizes are used, the effective length of a beam is reduced at either end by half the width of the columns below. Also, the effective height of a column is reduced by half the maximum depth of girders on either side, at both ends of the column.

Beams and columns may be prismatic or circular. However, when finite joint sizes are taken into account, a rigid zone of very large moment of inertia is assumed at each end of such members. Shear deformation is included and for columns, axial distortions are also considered.

In general, each frame is thought of as a setback frame. Hence, the number of towers (blocks with the same number of bays) must be specified. A regular frame is treated as one having a single tower. Members may be omitted from any position simply by specifying zero properties. However, if two adjacent beams are omitted, the lateral displacement of the common joint is still constrained to be the same as that of other joints at that level. Also, at least one of the members framing into a joint must have non-zero properties.

It is assumed that the foundations of all columns and shearwalls are rigid and lie at the same level.

(a) ACTUAL FRAME

(b) IDEALIZED FRAME

FIG.2.3 MATHEMATICAL MODAL OF A FRAME

2.4 MEMBER STIFFNESS

2.4.1 Beam Stiffness

A beam member, being a part of the frame, can be modelled as shown in Fig. 2.4. The total length of a beam is defined as the distance between the two end column center lines. When finite joint sizes are considered, the effective length of the beam is its total length less the lengths of the two rigid end stubs. A rigid end stub is taken as half the width of the column below the beam end. The beams are considered to have two degrees of freedom at each end and in developing their stiffness both flexure and shear deformations are taken into account.

For a prismatic beam (Fig. 2.4), the member-flexibility matrix corresponding to the rotational degrees of freedom and including shear and bending distortions can be written as^(13,20):

$$[f] = \frac{L}{6EI} \begin{bmatrix} 2+\alpha & \alpha-1 \\ \alpha-1 & 2+\alpha \end{bmatrix} \quad (2.1)$$

where

$$\alpha = \frac{6EI}{k'AGL^2} \quad (2.1a)$$

A = area of cross-section

k' = effective shear coefficient < 1 (2.1b)

L = length of beam

I = moment of inertia

E = modulus of elasticity

G = modulus of rigidity

By inverting the above flexibility matrix, the basic stiffness matrix for end rotations is obtained as:

$$[k_b] = \frac{2EI}{L(1+2\alpha)} \begin{bmatrix} 2+\alpha & 1-\alpha \\ 1-\alpha & 2+\alpha \end{bmatrix} \quad (2.2)$$

In line with the direct stiffness technique, two rigid end stubs are now attached to the effective beam length of Fig. 2.5 and a transformation relation developed between the member deformations and frame displacements. Referring to Fig. 2.6, the transformation relationship is derived as follows.

$$Y = a\theta_1 + b\theta_2 + \theta_3 - \theta_4 \quad (2.3a)$$

Since ϕ_1 and ϕ_2 are small

$$\beta = \frac{Y}{L} \quad (2.3b)$$

$$\phi_1 = \theta_1 + \beta \quad (2.3c)$$

and

$$\phi_2 = \theta_2 + \beta \quad (2.3d)$$

where L is the effective length and a and b are lengths of rigid end stubs.

Expressing Eqs. (2.3a) through (2.3d) in a matrix form:

$$\begin{Bmatrix} \phi_1 \\ \phi_2 \end{Bmatrix} = \begin{bmatrix} 1+a/L & b/L & 1/L & -1/L \\ a/L & 1+b/L & 1/L & -1/L \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{Bmatrix} \quad (2.4)$$

Equations (2.4) can be symbolically written as:

$$[\phi_B] = [T_B] [\theta_B] \quad (2.4a)$$

where $[T_B]$ represents a transformation matrix and $[\theta_B]$ and $[\phi_B]$ denote member deformation and frame displacement vectors respectively.

Now, following standard theory, the stiffness matrix, K_B , of a beam in the frame coordinates, is given by:

$$[K_B] = [T_B]^T [k_b] [T_B] \quad (2.5)$$

Substitution of Eqs. (2.2) and (2.4) into Eq. (2.5) gives:

$$[K_B] = \frac{2EI}{L(1+2\alpha)} \begin{bmatrix} 1+a/L & a/L \\ b/L & 1+b/L \\ 1/L & 1/L \\ -1/L & -1/L \end{bmatrix} \begin{bmatrix} 2+\alpha & 1-\alpha \\ 1-\alpha & 2+\alpha \end{bmatrix} \begin{bmatrix} 1+a/L & b/L & 1/L & -1/L \\ a/L & 1+b/L & 1/L & -1/L \end{bmatrix} \quad (2.6)$$

Equation (2.6) then gives the following stiffness matrix for an individual beam, when finite rigid joints are considered and shearing deformation is included:

$$[K_B] = \gamma \times \begin{bmatrix} (2+\alpha + \frac{6a}{L} + \frac{6a^2}{L^2}) & (1-\alpha + \frac{3}{L}(a+b) + \frac{6ab}{L^2}) & (\frac{3}{L} + \frac{6a}{L^2}) & -(\frac{3}{L} + \frac{6a}{L^2}) \\ (1-\alpha + \frac{3}{L}(a+b) + \frac{6ab}{L^2}) & (2+\alpha + \frac{6b}{L} + \frac{6b^2}{L^2}) & (\frac{3}{L} + \frac{6b}{L^2}) & -(\frac{3}{L} + \frac{6b}{L^2}) \\ (\frac{3}{L} + \frac{6a}{L^2}) & (\frac{3}{L} + \frac{6b}{L^2}) & (\frac{6}{L^2}) & -(\frac{6}{L^2}) \\ -(\frac{3}{L} + \frac{6a}{L^2}) & -(\frac{3}{L} + \frac{6b}{L^2}) & -(\frac{6}{L^2}) & (\frac{6}{L^2}) \end{bmatrix} \quad (2.7)$$

where

$$\gamma = \frac{2EI}{L(1+2\alpha)} \quad (2.8)$$

2.4.2 Column Stiffness

Column stiffness matrix is developed in a very similar manner except that here the axial deformation is also taken into account.

Thus, in terms of the basic deformation coordinates shown in Fig.

2.7, the column stiffness inclusive of shear distortion can be written as:

$$[k_C] = \gamma \times \begin{bmatrix} 2+\alpha & 1-\alpha & 0 \\ 1-\alpha & 2+\alpha & 0 \\ 0 & 0 & \frac{AE}{\gamma L} \end{bmatrix} \quad (2.9)$$

Referring to Fig. 2.8, the transformation relation is derived as follows:

$$x = a\theta_1 + b\theta_2 - \theta_3 + \theta_4 \quad (2.10a)$$

Since ϕ_1 and ϕ_2 are small,

FIG. 2.7 FLEXURAL COLUMN DEFORMATIONS

FIG. 2.8 TRANSFORMATION RELATION BETWEEN COLUMN DEFORMATIONS AND FRAME DISPLACEMENTS

$$\beta = \frac{x}{L} \quad (2.10b)$$

$$\phi_1 = \theta_1 + \beta \quad (2.10c)$$

$$\phi_2 = \theta_2 + \beta \quad (2.10d)$$

$$\phi_3 = \theta_6 - \theta_5 \quad (2.10e)$$

Expressing Eqs. (2.10a) through (2.10e) in a matrix form:

$$\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix} = \begin{bmatrix} 1+a/L & b/L & -1/L & 1/L & 0 & 0 \\ a/L & 1+b/L & -1/L & 1/L & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{Bmatrix} \quad (2.11)$$

or

$$[\phi_c] = [T_c][\theta_c] \quad (2.12)$$

where $[T_c]$ denotes the column transformation matrix.

From Eqs. (2.9) and (2.12), the column stiffness matrix $[K_c]$ is given by:

$$[K_c] = [T_c]^T [K_c] [T_c] \quad (2.13)$$

which leads to the following final form:

$$[K_c] = \gamma \begin{bmatrix} (2+\alpha + \frac{6a}{L} + \frac{6a^2}{L^2}) & (1-\alpha + \frac{3}{L}(a+b) + \frac{6ab}{L^2}) & -(\frac{3}{L} + \frac{6a}{L^2}) & (\frac{3}{L} + \frac{6a}{L^2}) & 0 & 0 \\ (1-\alpha + \frac{3}{L}(a+b) + \frac{6ab}{L^2}) & (2+\alpha + \frac{6b}{L} + \frac{6b^2}{L^2}) & -(\frac{3}{L} + \frac{6b}{L^2}) & (\frac{3}{L} + \frac{6b}{L^2}) & 0 & 0 \\ -(\frac{3}{L} + \frac{6a}{L^2}) & -(\frac{3}{L} + \frac{6b}{L^2}) & (\frac{6}{L^2}) & -(\frac{6}{L^2}) & 0 & 0 \\ (\frac{3}{L} + \frac{6a}{L^2}) & (\frac{3}{L} + \frac{6b}{L^2}) & -(\frac{6}{L^2}) & (\frac{6}{L^2}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{AE}{\gamma L} & -\frac{AE}{\gamma L} \\ 0 & 0 & 0 & 0 & -\frac{AE}{\gamma L} & \frac{AE}{\gamma L} \end{bmatrix} \quad (2.14)$$

The definition of member stiffness matrices in this form makes it a simple matter to model a shearwall. Thus, a shearwall is treated like any other column, except that the former has a large width. The beams framing into the shearwall will therefore have wide rigid end stubs.

2.5 FRAME STIFFNESS

The complete stiffness matrix for each frame is assembled using the direct stiffness technique. In this approach, the frame degrees of freedom must be numbered first. An efficient scheme is used as illustrated in Fig. 2.9 in which the frame degrees of freedom are numbered in such a manner that all non-zero elements of the stiffness matrix lie in a minimum bandwidth around the leading diagonal. Starting from the

FIG. 2.9 FRAME DEGREES OF FREEDOM

1	2		6	10	15		27		51									
	3	4	7	11	16	21	28	35	52									
		5	8	12	17	22	29	36	43	53								
			9	13	18	23	30	37	44	54								
				14	19	24	31	38	45	55	62							
					20	25	32	39	46	56	63	70						
						26	33	40	47	57	64	71	78					
							34	41	48	58	65	72	79					
								42	49	59	66	73	80	86	100			
									50	60	67	74	81	87	93	101		
										61	68	75	82	88	94	102		
											69	76	83	89	95	103		
												77	84	90	96	104	109	
													85	91	97	105	110	115
														92	98	106	111	116
															99	107	112	117
																108	113	118
																	114	119
																		120

FIG. 2.10 FRAME STIFFNESS MATRIX

lowest floor, the rotational degrees of freedom at a floor are numbered first; the story sideways degree of freedom is numbered next, being followed by the vertical joint displacement degrees of freedom. With this sequence of numbering, the maximum half-bandwidth of the stiffness matrix works out to $[3 \times (NBA + 1) + 2]$, where NBA is the number of bays in the base story.

The stiffness matrix is always symmetric. It is necessary to make the best use of this symmetry and bandedness of the stiffness matrix to conserve computer storage space and to avoid repetition of computations. In assembling the stiffness matrix, the method of direct joint stiffness is employed. In this method, a unit value is assigned to one joint displacement, the other displacement values being held to zero, and the forces generated at all affected degrees of freedom are calculated. Because only one symmetric half of the stiffness matrix need be generated, when a unit displacement is assigned to a particular degree of freedom, joint stiffness forces need be determined only in the affected degrees of freedom with a lower number. Of course, the joint force along the degree of freedom to which the displacement is assigned must also be determined. The stiffness matrix for the frame of Fig. 2.9 is shown schematically in Fig. 2.10.

The details of the technique employed are described in Section 4.5.1.

2.6 FRAME SIDESWAY STIFFNESS

The approximations described in Section 2.5 allow each frame to be treated as a separate substructure. The only connection is through the common displacements at the floor levels. Hence, the first step in the development of the stiffness matrix for the complete structure is to develop the sidesway stiffness matrices of the individual frames. A sidesway stiffness matrix contains only the degrees of freedom corresponding to the story sidesways. Since there is no straightforward method of determining the sidesway stiffness matrix directly, the complete frame stiffness matrix is developed first, as described in Section 2.5. Static condensation of the stiffness matrix is then used to obtain the sidesway stiffness matrix as follows.

A general frame stiffness equation for a given set of loads can be written as:

$$[K_F]\{\theta\} = \{F\} \quad (2.15)$$

where

$[K_F]$ = frame stiffness matrix

$\{\theta\}$ = frame displacement vector

$\{F\}$ = frame joint force vector.

Let the displacement vector be rearranged and partitioned into two sub-vectors such that vector θ_B represents displacements along sidesway degrees of freedom and θ_A denotes displacements along the remaining degrees of freedom. If the stiffness matrix and the load vector are also

rearranged and partitioned correspondingly, Eq. (2.15) can be rewritten

as:

$$\begin{bmatrix} k_{AA} & k_{AB} \\ k_{BA} & k_{BB} \end{bmatrix} \begin{bmatrix} \theta_A \\ \theta_B \end{bmatrix} = \begin{bmatrix} F_A \\ F_B \end{bmatrix} \quad (2.16)$$

or as

$$k_{AA}\theta_A + k_{AB}\theta_B = F_A \quad (2.17)$$

$$k_{BA}\theta_A + k_{BB}\theta_B = F_B \quad (2.18)$$

If the only loads acting on the frame are horizontal loads at floor levels, vector F_A is zero, and Eq. (2.17) gives:

$$\theta_A = - (k_{AA}^{-1} k_{AB} \theta_B) \quad (2.19)$$

Substitution of Eq. (2.19) into Eq. (2.18) gives:

$$(k_{BB} - k_{BA} k_{AA}^{-1} k_{AB}) \theta_B = F_B \quad (2.20)$$

or

$$K_* \theta_B = F_B \quad (2.21)$$

where

$$K_* = k_{BB} - k_{BA} k_{AA}^{-1} k_{AB} \quad (2.22)$$

The stiffness matrix K_* defined in Eq. (2.22) relates the story sidesway forces, F_B , to the story displacements degrees of freedom θ_B and is known as the condensed stiffness matrix or the Frame sidesway stiffness matrix..

It is important to note that in Eq. (2.21) the lateral loads F_B are not known until the complete building is solved. Thus, the individual frame sidesway stiffness matrices are essential for assembling the complete structure stiffness matrix or the global stiffness matrix and in distributing the in-plane lateral story forces, acting on the building, to its component frames (i.e. to obtain the force vector F_B in Eq. (2.21)).

The sidesway stiffness matrix can be obtained directly from Eq. (2.22). An alternative approach is to use the Gaussian elimination and back substitution method with a suitable adjustment technique as described in Section 4.5.3.

2.7 ASSEMBLY OF BUILDING STIFFNESS MATRIX

In order to assemble the complete building stiffness matrix from the component frame sidesway stiffness matrices, the latter should be transformed from the local coordinates of the frame to a common displacement coordinate system, which will be referred to as the global system. The local coordinates correspond to the lateral displacement of the frame at each floor level. The global coordinate system comprises three coordinates at each floor - two translations and one rotation. The location of these coordinates is taken at the centers of mass of different floors. To specify the positions of these centers of mass, a system of global reference axes are used. The coordinates of the centers of mass as measured from the global reference axes may vary from floor to floor.

With structure coordinates centered at the mass centers, the mass matrix required for the dynamic analysis will be a diagonal matrix, thus simplifying the eigenvalue solution. However, in cases where an analysis is required for lateral loads other than due to earthquake, the position of the global origin at each floor is arbitrary and can be set on the reference point.

The first step in the global stiffness matrix assembly is to develop a transformation relation between the local and global coordinate systems mentioned above. Fig. 2.11 shows the coordinate systems, at say, the n th floor level. For a frame oriented in the global Y -direction,

$$v_n^m = y_n + d_n^m \theta_n \quad (2.23)$$

where d_n^m is the perpendicular distance from the global origin to the frame axis, positive in the positive direction of global x -axis.

In matrix form:

$$v_n^m = \begin{bmatrix} 0 & 1 & d_n^m \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} \quad (2.24)$$

or

$$v_n^m = \begin{bmatrix} a_n^m \end{bmatrix} \{v_n\} \quad (2.25)$$

where $\begin{bmatrix} a_n^m \end{bmatrix}$ is the transformation matrix for floor n . For a frame oriented in the global X -direction, it can be shown that

$$\begin{bmatrix} a_n^m \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_n^m \end{bmatrix} \quad (2.26)$$

where d_n^m is the perpendicular distance measured parallel to the global Y-axis and is positive in the positive Y-direction. The complete transformation matrix for frame m is given by:

$$\begin{bmatrix} A^m \end{bmatrix} = \begin{bmatrix} a_1^m & & & \\ & a_2^m & & \\ & & \ddots & \\ & & & a_n^m \\ & & & & \ddots \\ & & & & & a_N^m \end{bmatrix} \quad (2.27)$$

where N is the number of stories in the frame.

If the lateral stiffness of the frame in the local coordinate system is represented by $\begin{bmatrix} k^m \end{bmatrix}$ and in global system by $\begin{bmatrix} \underline{k}^m \end{bmatrix}$, then

$$\begin{bmatrix} \underline{k}^m \end{bmatrix} = \begin{bmatrix} A^m \end{bmatrix}^T \begin{bmatrix} k^m \end{bmatrix} \begin{bmatrix} A^m \end{bmatrix} \quad (2.28)$$

The transformation matrix A^m is very sparse; the matrix operations in Eq. (2.28) are therefore greatly simplified, and it can be shown that

$$\begin{bmatrix} \underline{k}_{ij} \end{bmatrix} = \begin{bmatrix} a_i^m \end{bmatrix}^T k_{ij} \begin{bmatrix} a_j^m \end{bmatrix} \quad (2.29)$$

in which $\begin{bmatrix} \underline{k}_{ij} \end{bmatrix}$ is a 3x3 submatrix.

The global stiffness matrix K_g is now obtained by the direct summation of the transformed frame lateral stiffness matrices. Thus

$$\begin{bmatrix} K_g \end{bmatrix} = \sum_m \begin{bmatrix} \underline{k}^m \end{bmatrix} \quad (2.30)$$

It may be noted that the global stiffness $[K_g]$ is a full matrix but its size is relatively small compared to the total number of degrees of freedom associated with all the frames in the structure. In buildings with a symmetric distribution of mass and stiffness, the translational and rotational modes of vibration are uncoupled, and a lateral motion at the base of the building does not excite its torsional modes of vibration. In such a situation, the total response (or displacements) of the building can be obtained by the superposition of its responses (or displacements) along the two mutually perpendicular principal directions. A similar situation arises when the lateral loads on the building act through the centers of rigidity. The analysis is thus reduced from a single three-dimensional problem to a couple of two-dimensional ones resulting in a considerable saving in computing time. To analyze the building for its response in any one principal direction, the global stiffness matrix of the building for motion in that direction must be obtained. This is achieved in a simple manner by a direct superposition of the lateral stiffness matrices of all the frames oriented in the direction being considered. Thus,

$$[K_g] = \sum_m [k^m] \quad (2.30a)$$

where summation extends over all the frames in the direction considered.

FIG. 2.11 GLOBAL AND LOCAL COORDINATES OF A TYPICAL FLOOR

2.8 SOLUTION FOR STATIC LOADS

2.8.1 Gravity Loads

Gravity loads are supplied as uniformly distributed loads and/or concentrated loads on individual beams. For these loads, the joint load vector is prepared for each frame substructure and the system is analyzed frame by frame by the usual methods of two dimensional analysis. The general frame stiffness, Eqs. (2.15), are used and solved by one of the elimination methods which give the unknown joint displacements vector $\{\theta\}$. These joint displacements along with the member fixed end forces, which were derived earlier during preparation of the joint load vector, are applied to the individual frame members giving the member forces at the face of the joints in a standard fashion.

a) Beam Fixed End Forces

For a prismatic beam, the fixed end force vector for concentrated load including shear effect can be expressed as ⁽²⁾:

$$\begin{Bmatrix} FX_1 \\ FX_2 \\ FX_3 \\ FX_4 \end{Bmatrix} = \frac{P}{1+2\alpha} \begin{Bmatrix} \frac{cd^2}{L^2} \left(1 + \frac{\alpha L}{d}\right) \\ -\frac{c^2d}{L^2} \left(1 + \frac{\alpha L}{c}\right) \\ \frac{d^2}{L^3} \left(3c+d + \frac{2\alpha L^2}{d}\right) \\ \frac{c^2}{L^3} \left(c+3d + \frac{2\alpha L^2}{c}\right) \end{Bmatrix} \quad (2.31)$$

where FX_1, FX_2, FX_3, FX_4 are the fixed end actions and the other notations are explained in Fig. 2.12.

FIG.2.12 FIXED END ACTIONS FOR GRAVITY LOADS

FIG. 2.13 JOINT FORCES FOR GRAVITY LOADS

For a load distributed uniformly over the entire length of the beam, the shear deformation has no effect on the fixed end force vector and referring to Fig. 2.12(b), the fixed end force vector is given by:

$$\begin{Bmatrix} FX_1 \\ FX_2 \\ FX_3 \\ FX_4 \end{Bmatrix} = \begin{Bmatrix} wL^2/12 \\ -wL^2/12 \\ wL/2 \\ wL/2 \end{Bmatrix} \quad (2.32)$$

where w = intensity of load per unit length of the beam.

b) Frame Joint Load Vector

Since the columns do not carry any loads along their length, the joint loads are obtained as follows. The expressions derived below include the effect of finite joint sizes.

Let k be a typical joint where two beams i and j meet as shown in Fig. 2.13a. Using the free body diagram of Fig. 2.13b, the joint forces F_1^k and F_2^k are given by:

$$F_1^k = -FX_2^i - FX_1^j - (-FX_4^i)b_i + (-FX_3^j)a_j + \frac{w_i(b_i)^2}{2} - \frac{w_j(a_j)^2}{2}$$

or

$$F_1^k = -[FX_2^i + FX_1^j + a_j FX_3^j - b_i FX_4^i] + \left[\frac{w_i(b_i)^2}{2} - \frac{w_j(a_j)^2}{2} \right] \quad (2.33)$$

and

$$F_2^k = -[FX_4^i + FX_3^j + w_i b_i + w_j b_j] \quad (2.34)$$

where F_x represents the sum of fixed end actions for concentrated and distributed loads, and other notations are described in Fig. 2.13(b).

Finally, the frame joint force vector $\{F\}$ is obtained by computing F_1^k and F_2^k for all joints in a frame and putting them in appropriate slots corresponding to the degrees of freedom in the frame.

2.8.2 Lateral Loads

In this case, the global X and Y components of lateral forces and their points of action are supplied for each floor of the structure.

The global load vector $\{F_g\}$ is obtained as follows:

Referring to Fig. 2.14,

$$FH_n^1 = HX_n \quad (2.36)$$

$$FH_n^2 = HY_n \quad (2.37)$$

$$FH_n^3 = -HX_n(DY_n - \bar{Y}_n) + HY_n(DX_n - \bar{X}_n) \quad (2.38)$$

and

$$\{FH_n\} = \begin{bmatrix} FH_n^1 \\ FH_n^2 \\ FH_n^3 \end{bmatrix} \quad (2.39)$$

For a building where stiffness and lateral loads are symmetrically distributed, FH^3 may be taken as zero and the translational and rotational displacements are uncoupled which greatly simplifies the computations as discussed in Section 2.7. The global load vector $\{F_g\}$ is given by:

FIG. 2.14 PLAN OF n^{th} FLOOR OF A BUILDING WITH LATERAL LOADS

FIG. 2.15 BEAM END FORCES FOR JOINT DEFORMATIONS

$$\{F_g\} = \begin{Bmatrix} FH_1 \\ FH_2 \\ \vdots \\ FH_n \\ \vdots \\ FH_N \end{Bmatrix} \quad (2.40)$$

The above derivation is based on the assumption that the global origin is coincident with the center of mass. As stated earlier, when analysis for earthquake forces is not required, global origin can be moved to the reference origin. In that case, $\bar{X}_n, \bar{Y}_n$ will each be equal to zero and the global stiffness equations can be written as:

$$[K_g]\{\theta_g\} = \{F_g\} \quad (2.41)$$

Equations (2.41) are solved by the method of matrix inversion⁽¹²⁾ to give the global displacements $\{\theta_g\}$. The global displacements can be transformed to individual frame story displacements using the transformation matrix given by Eq. (2.27). Substitution of these story displacements in Eq. (2.21) gives the frame story forces $\{F_B\}$. A frame load vector is prepared from these story forces and the frame analyzed by the general method of Section 2.8.1 to obtain the member forces. It should be noted that the frame load vector in this case has zero values in all positions other than those corresponding to the sideways degrees of freedom and that there are no fixed end forces acting on the members.

2.8.3 Evaluation of Member Forces

After the frame stiffness Eqn. (2.15) is solved for any load vector, the frame displacement vector $\{\theta\}$ can be obtained. The transformation relation (2.4) then gives:

$$\phi_1 = (1 + \frac{a}{L})\theta_1 + \frac{b}{L}\theta_2 + \frac{1}{L}(\theta_3 - \theta_4)$$

or

$$\phi_1 = \theta_1 + (a\theta_1 + b\theta_2 + \theta_3 - \theta_4)/L = \theta_1 + \delta/L \quad (2.42)$$

Similarly,

$$\phi_2 = \theta_2 + (a\theta_1 + b\theta_2 + \theta_3 - \theta_4)/L = \theta_2 + \delta/L \quad (2.43)$$

where ϕ and θ are member deformations and the corresponding frame displacements respectively as shown in Fig. 2.6 and $\delta = (a\theta_1 + b\theta_2 + \theta_3 - \theta_4)$.

Using beam stiffness matrix of Eq. (2.2), the member stiffness relation for the clear length of the beam can be expressed as:

$$\gamma \begin{bmatrix} 2+\alpha & 1-\alpha \\ 1-\alpha & 2+\alpha \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} FX_1 \\ FX_2 \end{bmatrix} = \begin{bmatrix} FM_1 \\ FM_2 \end{bmatrix} \quad (2.44)$$

where

FX_1 = fixed end moment at the left end of a beam

FX_2 = fixed end moment at the right end of a beam

FM_1 = final member end moment at the left end

FM_2 = final member end moment at the right end.

Eqs. (2.44) give:

$$FM_1 = \gamma[(2+\alpha)\phi_1 + (1-\alpha)\phi_2] + FX_1 = F_1 + FX_1 \quad (2.45)$$

$$FM_2 = \gamma[(1-\alpha)\phi_1 + (2+\alpha)\phi_2] + FX_2 = F_2 + FX_2 \quad (2.46)$$

where F_1 and F_2 are the beam end moments caused by the corresponding end rotation ϕ_1 and ϕ_2 only.

Let F_3 and F_4 be the beam end reactions associated with the end moments F_1 and F_2 as shown in Fig. 2.15. Reactions F_3 and F_4 are then given by:

$$F_3 = (F_1 + F_2)/L \quad (2.47)$$

and

$$F_4 = -(F_1 + F_2)/L = -F_3 \quad (2.48)$$

If FX_3 and FX_4 are the fixed end reactions due to the applied loads, the final beam end reactions may be obtained as:

$$FM_3 = F_3 + FX_3 \quad (2.49)$$

and

$$FM_4 = F_4 + FX_4 = -F_3 + FX_4 \quad (2.50)$$

Substitutions of Eq. (2.42) into Eq. (2.45) and Eq. (2.43) into Eq. (2.46) give:

$$FM_1 = \gamma[(2+\alpha)(\theta_1 + \delta/L) + (1-\alpha)(\theta_2 + \delta/L)] + FX_1$$

or

$$FM_1 = \gamma[(2+\alpha)\theta_1 + (1-\alpha)\theta_2 + 3\delta/L] + FX_1 \quad (2.51)$$

Similarly,

$$FM_2 = \gamma[(1-\alpha)\theta_1 + (2+\alpha)\theta_2 + 3\delta/L] + FX_2 \quad (2.52)$$

Thus, Eqs. (2.49) and (2.50) with the aid of Eqs. (2.42) through (2.48) give:

$$FM_3 = \gamma[3(\theta_1 + \theta_2)/L + 6\delta/L^2] + FX_3 \quad (2.53)$$

and

$$FM_4 = -\gamma[3(\theta_1 + \theta_2)/L + 6\delta/L^2] + FX_4 \quad (2.54)$$

Thus, using the computed values of frame displacements, final beam end forces at the face of joints can be obtained by using Eqs. (2.51) through (2.54). A similar set of expressions can be derived for column end forces; however, the fixed end forces for columns are all zero.

CHAPTER 3

ANALYSIS FOR EARTHQUAKE MOTION

3.1 FORMULATION OF EQUATIONS OF MOTION

An actual building is a continuous system with a complex distribution of mass and stiffness. The exact formulation of the dynamic response of such a structure involves an infinite number of degrees of freedom. For most structures, however, a reasonable approximation to the true dynamic response may be obtained by defining the motion along a limited number of degrees of freedom within the system. In the buildings considered here, the response may be described by the lateral motions of each floor level, as described earlier in Section 2.7. Correspondingly, the mass of the building is lumped at each floor level, so that the system response is completely defined by specifying three displacements at each floor which has an assigned mass: two orthogonal translations in the plane of the floor and a rotation about the center of mass at the floor. With this lumped mass idealization, the dynamic equilibrium of the building in the above coordinate system is described as:

$$[M]\{\ddot{v}\} + [C]\{\dot{v}\} + [K]\{v\} = \{P(t)\} \quad (3.1)$$

in which M is the mass matrix, C the damping matrix, K the global stiffness matrix and $P(t)$ the time dependent forcing function; v being the displacement vector.

A structure subjected to a general base motion can be schematically represented by Fig. 3.1(a). In this case, as there is no externally applied load on the system, i.e. $P(t)=0$, Eq. (3.1) becomes:

$$[M]\{\ddot{v}_a\} + [C]\{\dot{v}\} + [K]\{v\} = 0 \quad (3.2)$$

in which $\{v_a\}$ is the vector of absolute displacements along the structure degrees of freedom measured with reference to a fixed frame in space and $\{v\}$ is the vector of displacements relative to the base. Since

$$\{v_a\} = \{v\} + \{v_g\} \quad (3.3)$$

where $\{v_g\}$ (Fig. 3.1(b)) is the vector of pseudo-static displacements due to support movement, it follows that

$$\{\ddot{v}_a\} = \{\ddot{v}\} + \{\ddot{v}_g\} \quad (3.4)$$

For the n th floor of a building with the displacement coordinates as shown in Fig. 3.1, relations (3.3) and (3.4) take the form:

$$\begin{Bmatrix} v_{aX} \\ v_{aY} \\ v_{a\theta} \end{Bmatrix}_n = \begin{Bmatrix} v_X \\ v_Y \\ v_\theta \end{Bmatrix}_n + \begin{Bmatrix} v_{gX} \\ v_{gY} \\ v_{g\theta} \end{Bmatrix} = \begin{Bmatrix} v_X \\ v_Y \\ v_\theta \end{Bmatrix}_n + \begin{Bmatrix} \cos\beta \\ \sin\beta \\ 0 \end{Bmatrix} \dot{v}_g \quad (3.4a)$$

and

$$\begin{Bmatrix} \ddot{v}_{aX} \\ \ddot{v}_{aY} \\ \ddot{v}_{a\theta} \end{Bmatrix}_n = \begin{Bmatrix} \ddot{v}_X \\ \ddot{v}_Y \\ \ddot{v}_\theta \end{Bmatrix}_n + \begin{Bmatrix} \cos\beta \\ \sin\beta \\ 0 \end{Bmatrix} \ddot{v}_g \quad (3.4b)$$

FIG. 3.1 GROUND AND STRUCTURE DISPLACEMENTS

i.e.

$$\{\ddot{v}_a\}_n = \{\ddot{v}\}_n + \{r\}_n \ddot{v}_g \quad (3.4c)$$

Hence, for all the floors,

$$\{\ddot{v}_a\} = \{\ddot{v}\} + \{r\} \ddot{v}_g \quad (3.4d)$$

where

$$\{r\} = \begin{Bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \\ \vdots \\ r_N \end{Bmatrix}; \quad r_1 = r_2 = \dots = r_n = \begin{cases} \cos \beta \\ \sin \beta \\ 0 \end{cases} \dots \text{etc.} \quad (3.4e)$$

and

β = angle of incidence of earthquake with the global x-axis

N = total number of stories in the building.

With the aid of Eq. (3.4), Eq. (3.2) can be rearranged as:

$$[M][\{\ddot{v}\} + \{r\} \ddot{v}_g] + [C]\{\dot{v}\} + [K]\{v\} = 0 \quad (3.5)$$

or

$$[M]\{\ddot{v}\} + [C]\{\dot{v}\} + [K]\{v\} = -\ddot{v}_g [M] \{r\} \quad (3.6)$$

The equations in (3.6) are coupled because, although $[M]$ is a diagonal matrix, $[K]$ and $[C]$ are not. These equations must normally be solved as $3N$ simultaneous differential equations. However, it is possible to develop a transformation which uncouples these equations so that they may be solved as a set of $3N$ independent equations. The transformation

matrix which consists of $3N$ columns is generated via the mode shapes or eigenvectors of the system as described in the following sections.

3.2 MODE SHAPES AND FREQUENCIES

The mode shapes and frequencies of the building are obtained by solving the undamped free vibration equations given by:

$$[M]\{\ddot{v}\} + [K]\{v\} = 0 \quad (3.7)$$

Eqs. (3.7) have a solution of the form:

$$\{v\} = \{a\}\sin(\omega t + \psi) \quad (3.8)$$

so that

$$\{\ddot{v}\} = -\omega^2 \{a\}\sin(\omega t + \psi) \quad (3.9)$$

where ω = frequency of vibration and ψ = phase angle.

Substitution of Eqs. (3.8) and (3.9) into Eqs. (3.7) gives:

$$-\omega^2 [M]\{a\} + [K]\{a\} = 0 \quad (3.10)$$

or

$$([I] - \omega^2 [K]^{-1} [M])\{a\} = 0$$

or

$$([I] - \omega^2 [D])\{a\} = 0 \quad (3.11)$$

in which $[I]$ is an identity matrix and $[D]$ called the dynamic matrix is equal to $[K]^{-1}[M]$. Equations (3.11) have a non-trivial solution only when the determinant of the matrix $([I] - \omega^2 [D])$ is zero. This condition leads to a polynomial equation of order $3N$ in ω^2 and gives $3N$ different

values of the frequency ω . Substitution of these values of ω into Eqs. (3.11) gives $3N$ different solutions for the vector $\{a\}$. These solution vectors are not, however, unique. One of the elements of such a vector can be assigned any arbitrary value; the rest are then uniquely determined. The $3N$ solution vectors are called the mode shapes, and it is usual to normalize them so that if $\{\phi_i\}$ represents the i th mode shape

$$\{\phi_i\}^T [M] \{\phi_i\} = 1 \quad (3.12)$$

In the dynamic analysis of a structure, it is usual to calculate only the first few modes, since contribution of the higher modes to the response of the system is negligible. The method for determining mode shapes and frequencies, which is most suitable in these cases, is the so-called matrix iteration method or the method of inverse power^(5,6). A brief description of the method is given below.

3.2.1 The Matrix Iteration Method

An alternative form of Eq. (3.11) is:

$$\left(\frac{1}{\omega^2} [I] - [D]\right) \{a\} = 0$$

or

$$\frac{1}{\omega^2} \{a\} = [D] \{a\} \quad (3.13)$$

This implies that if a vector $\{a\}$ representing any one of the mode shapes is multiplied by the dynamic matrix the resulting vector is equal to $\{a\}$ multiplied by a certain constant. In general, if an arbitrary vector

$\{u_i\}$ is multiplied by $[D]$, the resulting vector $\{u_{i+1}\}$ will not be similar to $\{u_i\}$. However, it can be shown that if the multiplication by $[D]$ is repeated several times the resulting vector will converge to the mode shape with the lowest frequency, that is, the fundamental mode shape. This can be expressed as follows.

If u_i and u_{i+1} represent two successive trial vectors so that

$$\frac{1}{A_{i+1}} \{u\}_{i+1} = [D] \{u\}_i \quad (3.14)$$

where A_{i+1} is that element of u_{i+1} which has the largest absolute value, then u_{i+1} will converge to the first mode shape and A_{i+1} will converge to the square of the fundamental frequency. This can be proved as follows.

Let $\omega_1, \omega_2, \omega_3, \dots, \omega_N$ represent the N frequencies of an N degree of freedom system so that

$$\omega_1 < \omega_2 < \omega_3 < \dots < \omega_N$$

and let the corresponding mode shapes be denoted by the vectors $\phi_1, \phi_2, \dots, \phi_N$. Then any arbitrary trial vector, say $\{u\}$ can be represented as:

$$\{u\} = c_1 \{\phi_1\} + c_2 \{\phi_2\} + \dots + c_N \{\phi_N\} \quad (3.15)$$

where $c_1, c_2, \dots, c_N$ are constants. Premultiplication of $\{u\}$ by $[D]$ gives:

$$\begin{aligned} \{u\}_2 &= [D] \{u\}_1 = c_1 [D] \{\phi_1\} + c_2 [D] \{\phi_2\} + \dots + c_N [D] \{\phi_N\} \\ &= c_1 \frac{1}{\omega_1^2} \{\phi_1\} + c_2 \frac{1}{\omega_2^2} \{\phi_2\} + \dots + c_N \frac{1}{\omega_N^2} \{\phi_N\} \\ &= \frac{1}{\omega_1^2} \left[c_1 \{\phi_1\} + c_2 \frac{\omega_1^2}{\omega_2^2} \{\phi_2\} + \dots + c_N \frac{\omega_1^2}{\omega_N^2} \{\phi_N\} \right] \end{aligned} \quad (3.16a)$$

Repeated multiplication by $[D]$ will give:

$$\{u\}_3 = [D]\{u\}_2 = \frac{1}{\omega_1^4} \left[c_1 \{\phi_1\} + c_2 \frac{\omega_1^4}{\omega_2^4} \{\phi_2\} + \dots + c_N \frac{\omega_1^4}{\omega_N^4} \{\phi_N\} \right] \quad (3.16b)$$

$$\begin{aligned} \{u\}_p = [D]\{u\}_{p-1} &= \frac{1}{\omega_1^{2(p-1)}} \left[c_1 \{\phi_1\} + c_2 \frac{\omega_1^{2(p-1)}}{\omega_2^{2(p-1)}} \{\phi_2\} + \dots \right. \\ &\quad \left. + c_N \frac{\omega_1^{2(p-1)}}{\omega_N^{2(p-1)}} \{\phi_N\} \right] \end{aligned} \quad (3.16c)$$

$$\begin{aligned} \{u\}_{p+1} = [D]\{u\}_p &= \frac{1}{\omega_1^{2p}} \left[c_1 \{\phi_1\} + c_2 \frac{\omega_1^{2p}}{\omega_2^{2p}} \{\phi_2\} + \dots \right. \\ &\quad \left. + c_N \frac{\omega_1^{2p}}{\omega_N^{2p}} \{\phi_N\} \right] \end{aligned} \quad (3.16d)$$

Since $\omega_1 < \omega_2 < \omega_3 \dots < \omega_N$, then if p is sufficiently large,

$$\frac{1}{\omega_1^{2p}} \gg \frac{1}{\omega_2^{2p}} \gg \dots \gg \frac{1}{\omega_N^{2p}} \quad (3.17)$$

Therefore, in the limit,

$$\lim_{p \rightarrow \infty} \{u\}_p = \frac{1}{\omega_1^{2(p-1)}} c_1 \{\phi_1\} \quad (3.18a)$$

$$\lim_{p \rightarrow \infty} \{u\}_{p+1} = \frac{1}{\omega_1^{2p}} c_1 \{\phi_1\} \quad (3.18b)$$

The ratio of any two corresponding elements of $\{u\}_p$ and $\{u\}_{p+1}$ will approach the value

$$\frac{u_{j,p+1}}{u_{j,p}} = \frac{1}{\omega_1^2} \quad (3.19)$$

and the vector $\{u\}_p$ will approach $\{\phi_1\}$.

To obtain the higher modes by the inverse power method, the trial vector must be independent of the previous modes. If it is possible to make c_1 zero in Eq. (3.15), then after iteration, vector $\{u\}$ will converge to the second mode, $\{\phi_2\}$. This gives the condition:

$$c_1 = \frac{[u]^T [M] \{\phi_1\}}{\{\phi_1\}^T [M] \{\phi_1\}} = 0$$

or

$$\{u\}^T [M] \{\phi_1\} = 0 \quad (3.21)$$

On expanding,

$$u_1 m_1^1 + u_2 m_2^1 + u_3 m_3^1 + \dots + u_N m_N^1 = 0 \quad (3.22)$$

where

$$\{m^1\} = [M] \{\phi_1\} \quad (3.23)$$

In Eq. (3.22), elements $u_2, u_3, \dots, u_N$ can be chosen arbitrarily and a new value of u_1 determined as follows:

$$\bar{u}_1 = - \frac{1}{m_1^1} [u_2 m_2^1 + u_3 m_3^1 + \dots + u_N m_N^1] \quad (3.24)$$

Also,

$$\begin{aligned} \bar{u}_2 &= u_2 \\ \bar{u}_3 &= u_3 \\ &\vdots \\ \bar{u}_N &= u_N \end{aligned} \quad (3.25)$$

In matrix form, these relationships can be expressed as:

$$\{\bar{u}\} = \begin{bmatrix} 0 & -\frac{m_2^1}{m_1^1} & -\frac{m_3^1}{m_1^1} & \dots & -\frac{m_N^1}{m_1^1} \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_N \end{bmatrix} \quad (3.26)$$

or

$$\{\bar{u}\} = [S]_1 \{u\} \quad (3.27)$$

Matrix $[S]$ is called the sweeping matrix because it sweeps out the previous mode and $\{\bar{u}\}$ is called a purified trial vector. Since the process of iteration may introduce contributions of the swept mode in the new trial vector, the process of sweeping should be repeated at each step. Recognizing that

$$\begin{aligned} \{\bar{u}\}_p &= [D]\{\bar{u}\}_{p-1} \\ &= [D][S]_1 \{u\}_{p-1} \\ &= [D]_2 \{u\}_{p-1} \end{aligned} \quad (3.28)$$

the sweeping process will be automatically accomplished if $[D]$ is replaced by $[D]_2$ in the iteration process. Matrix $[D]_2 = [D][S]_1$ is called the modified dynamic matrix.

A similar process is used to obtain the higher modes. Thus, for the third mode, the trial vector should be independent of the first and the second modes and the following orthogonality relationships should be satisfied:

$$\{u\}^T [M] \{\phi_1\} = 0 \quad (3.29)$$

$$\{u\}^T [M] \{\phi_2\} = 0 \quad (3.30)$$

Expansion of Eqs. (3.29) and (3.30) leads to

$$u_1 m_1^1 + u_2 m_2^1 + \dots + u_N m_N^1 = 0 \quad (3.31)$$

$$u_1 m_1^2 + u_2 m_2^2 + \dots + u_N m_N^2 = 0 \quad (3.32)$$

When Eqs. (3.31) and (3.32) are used to define new elements $\bar{u}_1$ and $\bar{u}_2$, these two elements of the trial vector are constrained. The remaining elements may be chosen arbitrarily; thus,

$$\left. \begin{array}{l} \bar{u}_3 = u_3 \\ \bar{u}_4 = u_4 \\ \dots \\ \bar{u}_N = u_N \end{array} \right\} \quad (3.33)$$

Equations (3.31), (3.32) and (3.33) may be combined to form a new sweeping matrix $[S]_2$ which sweeps out the first and second modes. The new sweeping matrix is similar to $[S]_1$ defined in Eq. (3.26), except that here the first two rows are determined by Eqs. (3.31) and (3.32). The second

sweeping is used, as was the first, to modify the dynamic matrix as:

$$[D]_3 = [D]_2[S]_2 \quad (3.34)$$

Because of the orthogonality relationships,

$$\{\phi_i\}^T [M] \{\phi_j\} = 0, \quad i \neq j \quad (3.38)$$

and Eq. (3.37) reduces to

$$\begin{aligned} [D]_2 \{u\} &= \frac{c_1}{\omega_1} \{\phi_1\} + \frac{c_2}{\omega_2} \{\phi_2\} + \dots + \frac{c_N}{\omega_N} \{\phi_N\} - \frac{c_1}{\omega_1} \{\phi_1\} \\ &= \frac{c_2}{\omega_2} \{\phi_2\} + \dots + \frac{c_N}{\omega_N} \{\phi_N\} \end{aligned} \quad (3.39)$$

Obviously, the first mode has been swept off and the iteration process will therefore converge to the second mode.

3.2.2 Inverse Power Method with Shifts

The method of sweeping, used with the iteration process to obtain the higher modes, may introduce cumulatively increasing numerical inaccuracies in those mode shapes. After a certain number of modes have been obtained, the loss in accuracies may become significant. The inverse power method can be slightly modified⁽⁵⁾ to overcome this drawback.

Consider again the eigenvalue Eq. (3.10) which can be rewritten as:

$$K - \lambda M \{u\} = 0 \quad (3.40)$$

where $\lambda = \omega^2$ and let a shift be introduced in the origin from where the eigenvalues are measured so that

$$\lambda = \lambda_0 + \mu \quad (3.41)$$

Then

$$[K - (\lambda_0 + \mu)M] \{u\} = 0$$

or

$$\frac{1}{\mu} \{u\} = [K - \lambda_0 M]^{-1} [M] \{u\} \quad (3.42)$$

It is obvious that if a revised dynamic matrix $\bar{D} = [K - \lambda_0 M]^{-1} [M]$ is used, the iteration will converge to the smallest value of μ or to the mode whose frequency is closest to the shift point.

Fig. 3.2 shows a schematic representation of the shift of origin. Supposing that the first four mode shapes and frequencies have been obtained with $\lambda_0 = 0$. At this stage, it is decided that the origin be shifted to a point close to the fifth mode frequency.

In order to estimate a value of the new shift, the iteration process is continued beyond λ_4 with the previous origin but terminated after only a few cycles giving a new value of λ which will be close to λ_5 and may be used as λ_0 for the new dynamic matrix. However, numerical instability may arise if λ_0 is too close to λ_5 . The shifted origin is therefore located so that λ_0 is the average of λ_4 and the estimated value of λ_5 . The following new dynamic matrix is now formed:

$$\bar{D} = [K - \lambda_0 M]^{-1} [M] \quad (3.43)$$

If the fifth mode frequency is closer to the shifted origin than any other frequency, an iteration using $[\bar{D}]$ will converge to the fifth mode. However, because only an approximate estimate of the fifth mode frequency can be made at this stage in the iteration and also because the

FIG. 3.2 MEASUREMENT OF EIGENVALUES FROM SHIFTED ORIGIN(ref. 8)

FIG. 3.3 ELASTIC AVERAGE RESPONSE SPECTRUM FOR 1.09 MAX GROUND ACCELERATION (ref. 8)

shifted origin will probably be used to find the sixth and a few more modes, the dynamic matrix is revised so that the mode shapes previously determined are swept off giving:

$$[\bar{D}]_2 = [\bar{D}] - \sum_{i=1}^4 \frac{1}{\nu_i} (\{\phi_i\} \{\phi_i\}^T [M]) \quad (3.44)$$

The iteration carried out with $[\bar{D}]_2$ will converge to the fifth mode. In general, about six modes can be extracted from each origin without a significant loss of accuracy. Each time the origin is shifted, a new matrix inversion is required. However, in exchange, not only the accuracy of the subsequent modes is improved, but also the iteration process is speeded up.

3.3 TRANSFORMATION OF THE EQUATIONS OF MOTION

The mode shapes of any structural system satisfy the orthogonality relationships:

$$\{\phi_i\}^T [M] \{\phi_j\} = 0, \quad i \neq j \quad (3.45)$$

$$\{\phi_i\}^T [K] \{\phi_j\} = 0, \quad i \neq j \quad (3.46)$$

Further, if the mode shapes have been properly normalized, the following relationships hold:

$$\{\phi_i\}^T [M] \{\phi_i\} = 1 \quad (3.47)$$

and

$$\{\phi_i\}^T [K] \{\phi_i\} = \omega_i^2 \quad (3.48)$$

Now, let the following transformation of coordinates be introduced in Eqs. (3.6).

$$\{v\} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_i & \dots & \phi_{3N} \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \\ \vdots \\ z_{3N} \end{Bmatrix} \quad (3.49)$$

$$\text{or } \{v\} = [B]\{z\} \quad (3.50)$$

so that Eq. (3.6) reduces to:

$$[M][B]\{\ddot{z}\} + [C][B]\{\dot{z}\} + [K][B]\{z\} = -\ddot{v}_g[M]\{r\} \quad (3.51)$$

Premultiplication of Eqs. (3.51) by $[B]^T$ yields:

$$[B]^T[M][B]\{\ddot{z}\} + [B]^T[C][B]\{\dot{z}\} + [B]^T[K][B]\{z\} = -\ddot{v}_g[B]^T[M]\{r\} \quad (3.52)$$

Now because of orthogonality relationships, Eqs. (3.45) and (3.46), matrices $[B]^T[M][B]$ and $[B]^T[K][B]$ will be diagonal. This is not in general true of $[B]^T[C][B]$. However, let it be assumed that the damping matrix is specially chosen so that the transformation diagonalizes it and let

$$[\phi_i]^T[C]\phi_i = 2\omega_i\xi_i \quad (3.53)$$

where ξ_i represents damping in the i th mode. Then Eqs. (3.52) will give $3N$ uncoupled single degree of freedom equations; the i th equation being

$$\ddot{z}_i + 2\omega_i\xi_i\dot{z}_i + \omega_i^2z_i = -Q_i\ddot{v}_g \quad (3.54)$$

where Q_i is called the modal participation factor for the i th mode and is given by:

$$Q_i = \{\phi_i\}^T [M] \{r\} \quad (3.55)$$

The $3N$ single degrees of freedom equations can be solved to obtain the vector $\{z\}$, which on substitution into transformation Eq. (3.50) will give the displacement vector $\{v\}$.

3.4 SPECTRUM ANALYSIS

For earthquake analysis of a structure, when the maximum values of the displacements and forces are of interest, a realistic approach is via the response spectrum. The response of a single degree of freedom system subjected to a ground-motion is defined by the equation:

$$m\ddot{v} + c\dot{v} + kv = -m\ddot{v}_g$$

or

$$\ddot{v} + 2\omega\xi\dot{v} + \omega^2 v = -\ddot{v}_g \quad (3.56)$$

The maximum absolute value that v attains relative to the ground during the entire history is called the spectral displacement and is denoted by S_D , so that

$$S_D = v_{\max} \quad (3.57)$$

Spectral velocity and spectral acceleration are two other quantities of interest, and it is customary to define them as follows:

$$\text{Spectral velocity, } S_v = \omega v_{\max} \quad (3.58)$$

$$\text{Spectral acceleration, } S_a = \omega^2 v_{\max} \quad (3.59)$$

Any one of the three spectral quantities may be evaluated for certain values of damping and plotted against frequency or period. Such a plot is called a response spectrum.

In terms of the spectral acceleration, the maximum response of a structure in its i th mode is given by:

$$z_i(\max) = \frac{Q_i S_a(\omega_i, \xi_i)}{\omega_i} \quad (3.60)$$

If it is assumed that the maximum response in each mode occurs at the same instance of time, the above implies a set of actual displacements.

$$\{v_i\} = \{\phi_i\} z_i(\max) \quad (3.61)$$

and a corresponding set of member forces.

The maximums of different modes will generally occur at different times. The response is therefore less than the direct sum of the absolute values in different modes. A reasonable estimate of the maximum displacements and member forces may be made by calculating the root-sum-square of the modal values. However, since this may sometimes underestimate the true maxima, the National Building Code of Canada recommends that a second estimate of true response be obtained by taking the sum of the two absolute modal values and that the greater of the values so obtained and the one obtained by root-sum-square be taken as the true response.

3.5 DESIGN RESPONSE SPECTRUM

The dynamic analysis procedure specified in the commentary to the National Building Code recommends the use of design response spectra to evaluate the maximum response of a structure. The code provides design response spectra for several values of damping and a design acceleration of $1g$. Response spectra for other values of ground acceleration are obtained by a simple linear scaling. Also, spectra for the values of damping other than those given in the code can be obtained by linear interpolation. The design spectra given in the commentary consist of line segments on a double log graph as shown in Fig. 3.3. The damping in the structure is automatically provided for by selecting an appropriate response spectrum. The effect of inelasticity can be accounted for by a procedure given in the Code⁽¹⁴⁾. The commentary to the Code recommends that the elastic spectral values be adjusted for the inelasticity in the following manner.

The spectral acceleration should be divided by the ductility factor μ , that can be realized in the structure, for modal periods falling in the range of the velocity and displacement bounds on the response spectrum (i.e. for periods greater than T_D shown in Fig. 3.3) and by $\sqrt{2\mu-1}$ for modal periods along the acceleration bound (i.e. for periods less than T_D).

The spectral displacement should be left unchanged for periods greater than T_D , but should be multiplied by $\mu/\sqrt{2\mu-1}$ for periods less than T_D .

Appropriate values of μ are given in the commentary for different structural layouts and materials.

CHAPTER 4

DESCRIPTION OF PROGRAMS

4.1 GENERAL

This Chapter presents a description of the program developed for the three-dimensional analysis of building structures. The program is written in standard FORTRAN IV language and is designed to run on any computer installation which accepts FORTRAN IV.

The program is divided into two sections. The first section named AIDS⁺ reads all necessary input data in a conversational mode with rapid input options. This program section then translates the data, rearranges it and writes it in rigid format on disc files so that it is in a form compatible with the main program of the second part. The program has been tested on an IM-70 minicomputer and should run on other minicomputers with similar attributes.

The second section named ANGELS^{*} is designed to carry out a complete elastic analysis of the building based on the assumptions described in Chapter 2 and using the data supplied by program AIDS. The program has been tested on a HONEYWELL Sigma-9 computer and can be adapted to run on a minicomputer. This program can also work independently of program AIDS. In that case, however, data has to be supplied through cards or

+ AIDS - Automatic Input Data Supplier

* ANGELS - ANalysis of building for Gravity, Earthquake and Lateral loads

data files in rigid format described in Appendix A. Program ANGELS finally produces all necessary output requested through user options. The programs are designed to accept any set of consistent units for length, force and displacements. Appropriate set of units must be selected by the user while supplying input data.

4.2 MATHEMATICAL DESCRIPTION OF THE BUILDING

For the purpose of preparing the numerical input data, the building to be analyzed is at first resolved into two orthogonal sets of planar frame or frame-shearwall systems. Each frame or frame-wall system must be assigned two identifier numbers. The first identifier is a frame group number; frames with similar geometric and structural properties being collected into one group. A typical building, such as the one shown in Fig. 4.1, may have several frame groups. Each group is assigned a number starting from one and increasing sequentially. The second identifier is a unique number assigned to each frame, called the frame number. This number also starts from one and increases sequentially. It is, however, a requirement that the frames in Group I are numbered first, those in Group II are numbered next and so on, till all the frame groups are exhausted. Also, within a group, frames carrying exactly the same gravity loads must be numbered so that they form a cluster uninterrupted by frames with different loading patterns.

Gravity loads are to be specified in terms of uniformly distributed loads and concentrated loads on each beam in a frame. A maximum of four

concentrated loads can be specified on any one beam. Lateral loads are applied at the floor levels and act on the complete structure. The values of their X and Y components along with the coordinates of their points of action with reference to the global reference point should be supplied to the program.

The program also needs the orientations and locations of the frames to assemble the global stiffness matrix. Further, it requires the location of mass center and the translational and rotational masses at each floor level to assemble the global mass matrix. The frame locations and mass centers must be specified with respect to the global reference points at each floor level. The choice of a reference point is arbitrary, but all reference points must lie on the same vertical line. Also, it is to be noted that the line of action of the earthquake force resultant acts through the center of mass at each floor level.

Each frame or shearwall element is assumed to have stiffness only in the in-plane direction. Elements which have stiffness in another direction too must be defined by an additional element. For example, the properties of columns which are common to two different frames must be supplied twice. As stated earlier, the axial deformations in these doubly defined columns will not be the same. However, this incompatibility should not cause any serious error in the analysis.

The properties of a frame or shearwall are specified with respect to its local stiffness direction. Floor beam center lines and column center lines are the basic reference lines used in the frame description.

The height from one floor beam center line to the next is assumed to be the same for all frames and is equal to the story height. A member may be omitted from any position by specifying zero properties. However, a shearwall must be continuous all through to the base of the frame. Beams must be prismatic and columns may be prismatic or circular. Both flexural and shearing deformations are considered in the analysis. Beams are considered axially rigid but axial deformation of columns are accounted for. Also, finite joint sizes of infinite rigidity may be considered. A frame, in general, is considered as a setback one. A part of the frame height containing the same number of bays is named a tower; a frame containing more than one tower is called a setback frame as illustrated in Fig. 2.2. A regular frame is considered as one consisting of a single tower.

4.3 INPUT DATA PROGRAM - AIDS

The program is designed to act as a front end program to the main analysis program - ANGELS. It reads all input data required for the main program in a conversational mode. The conversational program is designed to accept a machine independent free-format input. To achieve this, a special subroutine called REED has been used in this program⁽¹⁸⁾. The input of data is supplied via a time sharing terminal and is truly free format. One or more data items can be read from a line of input, consecutive data items being separated by one or more blanks. The data

can be numeric or alphanumeric. For numeric data, no decimal is required for an integer; also, a real whole number can be supplied as an integer, the decimal being optional.

If the building to be analyzed is large in size with a great many frames, a large amount of data will be required by the program ANGELS. However, very often, several dimensions, sizes and loads in a structure are repetitive. For example, most story heights may be equal; also, member sizes may be the same over a number of stories and bays. It is possible to speed up the data input considerably by taking advantage of these repetitions. In program AIDS, a very careful attention has been given to this aspect and ways have been devised for speeding up the input process without making it too complex.

The essential features of data input for the program AIDS are illustrated with reference to the building of Fig. 4.1. The conversational data input to a minicomputer is shown in Appendix A. Those portions of the dialogue which are keyed in by the user are underlined; the remaining portions are prompted by the computer.

The first question that the program asks the user is whether he wishes to make a copy of the input data on a disc file. If the data is currently being supplied through a terminal, it is a good practice to make a copy. The copy can be later used to make any corrections that may be required in the data supplied during the current session on the terminal. The user can express his wish to create such a file by typing a Y. With this option, the program will make a copy of the currently

2

FIG. 4.1 REINFORCED CONCRETE BUILDINGS UNDER GRAVITY, EARTHQUAKE AND LATERAL LOADS.

FIG. 4.1 (contd) Gravity Loads On The Building

FIG. 4.1 (contd) Lateral Loads on The Building

FIG. 4.1 (contd) Centers of Masses

supplied input data on a disc file on unit 6; the name of the file being predetermined during assignment of units.

The option to read the data from a previously created disc file instead of from a terminal, is exercised at the question immediately following.

The next few questions which are self explanatory, deal with structure and frame description. A subsequent question prompts the user to supply the story heights. The story heights are supplied starting from the lowest story and proceeding upwards. For the present example, the user types in 120, in reply to a request for story height between levels 0 and 1. Because the height of the next story is also 120, the user can speed up the data input at this stage by typing in 120 TO 2. If he does so, the program automatically assigns a value of 120 to both of the first and the second stories and then prompts the user to supply the story height between levels 2 and 3; the user then types in 100.

The next question asked by the computer relates to the frame information and left identification. The left identification is a symbol as put on the drawing to identify the left side of a frame in a building.

The program then asks for frame group properties starting from Group I. The program first prompts the user to supply the lengths of bays for frame Group I, starting from the left, and proceeding to the right. A procedure similar to the one described for the story heights can be used in supplying this data..

Beam cross-section properties for frame Group I are input next. The computer asks whether all beams at Story No. 1 are of rectangular concrete sections or other (i.e. steel sections, etc.). If they are all concrete, the user types in C. The program then asks for the width and the depth of the

beam in Bay No. 1 (the leftmost bay). To speed up the input, the user types in 10 20 TO 2. With this, the program automatically assigns a size of 10 by 20 to the first and second beams of Story No. 1. Since all the beams at Story No. 1 have been input, the computer asks whether the beams in the next story are exactly like those in any preceding story (here the first story only). In fact, the beams in Story No. 2 are exactly of the same size as those in Story No. 1. The user can, therefore, speed up the input by typing in L(IKE) 1. The computer automatically assigns appropriate sizes to beams in story 2 so that these beams are exactly the same as the corresponding beams in story 1.

It is to be noted that when asked for the type of section, the user has the option to type in C or O. For type C, the program asks for the beam width and depth. Program ANGELS uses these properties to calculate the area and moment of inertia for the gross section. The computed values are then used in the analysis. For cross-sections other than rectangular or for the case where the user does not want to use the gross section properties, he should type in O. In that case, the user would be asked to supply the moment of inertia and depth in lieu of width and depth of the beams. Again, the speed of input can be enhanced by using the procedure described above.

Column cross-section properties are supplied in a manner similar to that followed for the beam cross-section properties.

Very often, several frames in a structure carry exactly the same gravity loads. When frames carrying similar gravity loads also belong to the same frame group, program AIDS can enhance the speed of data input. For example, with reference to the building of Fig. 4.1, in which

frames 1 and 2 belong to the frame Group I, the beams in frames 1 and 2 carry exactly the same gravity loads (in each of the gravity load cases). After the gravity load data has been supplied for frame 1, the computer asks whether the next frame (i.e. frame 2) carries exactly the same gravity load as frame 1. The user now types in Y. The program makes a note of this on the disc file and asks for the load data for frame 3. If frame 3 were also in frame Group I and the gravity loads on it were similar to those in frame 1, the user could further speed up input by typing in Y TO 3.

The dialogue between the computer and the user is self-explanatory and the data can be readily input in the usual manner.

An outline flow diagram of program AIDS is shown in Fig. 4.2. It is clear from the flow diagram that the program reads all input data in conversational mode, rearranges it and writes it on three disc files on units 3, 4 and 5 respectively. Also, the program is designed to keep a duplicate copy of all the user's input on a disc file assigned to unit 6.

Fig. 4.2. Outline flow diagram for Program AIDS.

Program AIDS also provides the facility for correcting errors in the input data. This is achieved by creating a duplicate input file on unit 6. For major errors, the user can edit this file using the conventional edit system of the computer. He can then rerun program AIDS, this time specifying that the input data be read from the disc file instead of from the terminal. However, for minor numerical errors made during the conversational input, the user can appropriately edit the file written on units 3, 4 or 5. For example, suppose that when asked for modulus of elasticity in Story No. 5 and Bay No. 3, the user has input a value of 2500 instead of 3000. After program AIDS has been run, the user can edit the file on unit 4 (file for member properties) substituting the correct value of modulus of elasticity for the wrong value.

The files written on units 3, 4 and 5 are used to supply the input data to the main analysis program ANGELS. During the development of programs reported here, the front end program AIDS was run on an IM-70 minicomputer using a 9600 bauds-terminal, and the data files created by the program were transferred to a HONEYWELL Sigma-9 computer as 'unkeyed' input data files. The main program ANGELS was run in a batch mode on Sigma-9 which read the data from the unkeyed input data files.

4.4 ANALYSIS PROGRAM - ANGELS

4.4.1 Basic Function

The program is designed to perform a complete three-dimensional analysis of multistoried buildings consisting of rigid frame and/or

frame-shearwall systems based on the assumptions described in Chapter 2. Any substructure within the building may be a regular or a setback frame or frame-wall. The analysis is based on the assumption that beams are axially rigid but columns can deform axially. Flexural deformations are, of course, considered. In addition, the following options are available:

- i) Shear deformations in the beams and columns may be accounted for provided the modulus of rigidity and the shear coefficients are specified. Such deformations will be disregarded if any of these items is assigned a value of zero. In fact, program AIDS asks the user whether he wishes to consider shear deformations and if he replies with an N, the computer will automatically set the shear coefficient to zero.
- ii) Finite joint sizes may be considered in the analysis.

Program ANGELS can analyze a building for:

- i) a maximum of 4 gravity load cases (GL1, GL2, GL3 and GL4)
- ii) a maximum of 3 lateral load cases (WL1, WL2, WL3)
- iii) one lateral earthquake load case supplied through a design spectrum (EL).

The above-stated load cases can be combined to form any number of load combinations. A load combination is defined by specifying the multiplication factors applicable to its component loads. One such load combination is called a 'Load System'. It may be formed, for example, from the expression $aGL2 + bWL1 + cEL$, where a, b and c are called load case multipliers.

In addition, if required, the program can evaluate three-dimensional mode shapes and frequencies of the building. Also, at the user's option, the modal story forces and displacements of the building (adjusted for ductility) for a specified design elastic spectrum may be calculated and printed out.

4.4.2 Organization of the Program

An outline flow diagram showing the program organization is shown in Fig. 4.3. The basic operations can be divided into the following major steps.

1. The first operation performed by the main program is to read and print out the basic control information and the global data associated with the building. This data is read from the disc file on unit 1.
2. The program next reads the properties of all substructures of the building from the disc file on unit 2. This data is then adapted for use in the calculations of stiffnesses and member forces. The program produces an echo print of the input data and writes its adapted version on a file on unit 4. The operations on this step are performed by subroutine PROP.
3. The next operation involves the formation of the frame stiffness matrices for each different frame or frame-shearwall in the building using frame property data written on unit 4. The program keeps a record of the diagonal addresses of the stiffness matrices and assigns suitable indices to the sidesway

degrees of freedom. The stiffness matrices and the associated records are stored sequentially on a disc file or a tape on unit 5. The operations are performed by subroutine STIFF.

4. The above stiffness matrices are then reduced one by one using Crout reduction method⁽⁶⁾ and stored sequentially on a disc file (or a tape) on unit 6 (subroutine REDUCE).
5. The gravity load data is then read from the disc file on unit 3, load vectors are prepared and an echo print of the input load data is produced (subroutine LOAD). Gravity load analysis is then carried out for each frame for all gravity load cases. The operations are performed by subroutine GRAVT which calls subroutine BSUB. The resulting joint displacement vectors are sequentially written on a disc file on unit 8 for use in Step 15.
6. The next operation involves the formation of a sidesway stiffness matrix for each individual frame and shearwall in the building by static condensation of the frame stiffness matrix which is retrieved from unit 5. The sidesway stiffness matrices are then sequentially written on the disc file on unit 2. The operation is performed by subroutine COND which calls subroutines CONDEN, REDUCE and BSUB.
7. Story mass data is then read from the disc file on unit 1 and printed out. This data includes the location of mass center at each floor and the story translation and rotational masses.

The operations are carried out by subroutine STORY which also assembles the global mass matrix.

8. The program next assembles the global stiffness matrix using frame sidesway stiffness matrices retrieved from unit 2, and the frame locations data read in step 1. Subroutine FRAME performs these operations. The global stiffness matrix is written on a file on unit 5 for use in step 10.
9. The program next reads the lateral load data for the building from the file at unit 3, for a maximum of 3 different lateral load cases and prints it out. The structure is then analyzed for these lateral loads. Frame lateral story forces are calculated and written on a file on unit 4 for use in step 13. These operations are accomplished by subroutine HLOAD which calls subroutines RECIP and SFORCE.
10. The mode shapes and frequencies of the building are then evaluated by subroutine EIGVAL, which is called through subroutine SPCTRM. The procedure described in Section 3.2 is followed in the evaluation.
11. The modal response is next calculated using the specifications of the design elastic spectrum read by the main program in step 1. The seismic design forces and displacements along the global coordinates are worked out and the values obtained are adjusted for ductility before being printed out at the user's

option. The operations are performed by the subroutine SPCTRM which calls subroutine RESP.

12. From the adjusted elastic modal displacements worked out in Step 11 and the frame location data read earlier, the modal floor story forces are worked out for each frame and stored sequentially on a disc file on unit 1 for subsequent use. The operations described in this step are carried out by subroutine SFORCE.
13. The program next retrieves from unit 1 the frame lateral story forces calculated in step 9, forms joint load vectors and solves for joint displacements (subroutine ROOT) for each lateral load case. The joint displacements are written sequentially on a file on unit 3 for use in step 15.
14. Using the modal frame story forces evaluated in step 12, the program then computes the root-sum-square forces for the members in each frame of the building and writes them down on a tape or disc file on unit 5 for use in step 15. These operations also are performed by subroutine ROOT which calls subroutines BSUB and FORCE.
15. Finally, the program reads for each frame the displacement vectors for gravity load cases from unit 8 and for lateral load cases from unit 3. The following steps are carried out for each load system. The displacement vectors for gravity and lateral load cases are multiplied by appropriate multiplying

factors defined earlier by the user (read in step 1) to form a final displacement vector. The member forces are then determined from this displacement vector. If the load system also includes earthquake loads, the root-sum-square forces are read from unit 5, multiplied by the proper factor and superimposed on the member forces calculated above to evaluate the final member forces which are then printed out. When combining earthquake forces with other loads, two sets of values are calculated to give both maximum and minimum member forces. In one set, the earthquake forces are added to other forces, while in the second set, they are subtracted from the combination of other forces. The entire procedure as described above is repeated for each frame of the building. The above operations are performed by subroutine SYSTEM which calls subroutine FORCE.

The program organization and file management for the program ANGELS is shown in Fig. 4.4.

4.5 EFFICIENCY OF ALGORITHMS

In the development of program ANGELS, effort has been made to achieve optimum efficiency in computer storage, solution of equations, condensation of frame stiffness matrices and the solution of eigenvalue problems. Various subroutine algorithms which effectively handle the above problems are discussed below.

FIG. 4.3. Outline flow diagram for operation of the program ANGELS.

FIG. 4.4 PROGRAM ORGANIZATION AND FILE MANAGEMENT FOR PROGRAM-ANGELS

4.5.1 Generation of Frame Stiffness Matrices, Subroutine STIFF

This subroutine is designed to formulate the frame stiffness matrices exploiting their symmetry and bandedness. With the algorithm used, only the non-zero elements within the upper half of the diagonal of a matrix are computed and stored. The subroutine internally generates an efficient numbering scheme as described in Section 2.5, which gives a very compact bandwidth for the matrix.

The stiffness matrix for the frame of Fig. 2.9 is schematically shown in Fig. 2.10 where only the non-zero elements and embedded zeros have been identified. It is obvious that because of symmetry and sparsity of the stiffness matrix it is not necessary to store all the elements of the matrix, and with special storage schemes, a great economy can be exercised in the storage requirement. Two such schemes have been in general use. They are;

- 1) Skew banded rectangular storage and
- 2) Compacted column storage with the non-zero part of each column above the diagonal of the matrix, strung compactly into a vector.

Thus, for the frame of Fig. 2.9, the storage of the entire stiffness matrix requires a 19×19 array, i.e. 361 storage spaces. A skew banded storage scheme, in which only the upper half band of the matrix is stored, with the diagonal elements positioned in the first column, will require a 19×11 array, i.e. 209 storage spaces. The compacted column storage will

require only 120 storage spaces. Obviously, the last scheme which omits the storage of almost all zero elements is the most efficient. It should be noted that the embedded zeros must still be stored; their storage is, in any case, essential because they are modified to non-zero quantities during a subsequent Gaussian elimination.

Subroutine STIFF assembles the frame stiffness matrices from the known values of member stiffness elements generated by the function routines CST and BST. The assembly of a stiffness matrix begins at the lowest story and proceeds upwards. During one pass of a DO-loop, the column vectors of the matrix pertaining to all degrees of freedom at a particular story are generated and stored in compacted column strings. It should be noted that because of the symmetry of the stiffness matrix, only those elements of a column including and above the diagonal need be generated. Concurrently with the assembly of the matrix, the program also generates the addresses of the diagonal positions in the storage string and assigns appropriate indices to the columns for identifying those degrees of freedom which will have to be eliminated subsequently during the static condensation of the matrix. The array that stores the stiffness matrix of Fig. 2.10 by the compacted column storage scheme is shown in Fig. 4.5. Fig. 4.5 also shows the diagonal address array and the array containing the indices assigned to various degrees of freedom.

1	2	3	4	5	6	7	8	110	111	112	113	114	115	116	117	118	119	120
---	---	---	---	---	---	---	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

(a) String Storage of Stiffness Matrix

1	3	5	9	14	20	26	34	42	50	61	69	77	85	92	99	108	114	120
---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----

(b) Diagonal Address Array

0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(c) Array of Indices to Identify Sidesway Degrees of Freedom

FIG. 4.5 STORAGE OF STIFFNESS MATRIX OF A MULTISTORY FRAME

4.5.2 Solution of Simultaneous Equations

Any structural analysis problem eventually reduces to the solution of a set of simultaneous equations. For a fairly large building, the number of such equations can be very large. The efficiency of an analysis program is therefore greatly dependent on the efficiency of the equation solving routine that it uses. Researchers have therefore devoted considerable attention to this aspect and solution algorithms of almost optimal efficiency have been developed. Most of them use some variation of the Gaussian elimination technique. For sparsely populated coefficient matrices, the Crout reduction procedure, which is a modified form of Gauss elimination scheme, has been found to be ideally suitable for computer application.

An algorithm based on the Crout reduction method has been developed⁽²²⁾ which omits all unnecessary computations and may be briefly described as follows.

Consider the solution of a symmetrical set of equations:

$$[A]\{x\} = \{b\} \quad (4.1)$$

where

$[A] = [a_{ij}] = N \times N$ non-singular symmetric matrix

$\{x\} = \{x_i\} = N \times 1$ vector of unknown, and

$\{b\} = \{b_i\} = N \times 1$ vector of known elements.

The required solution formulations may then be expressed as follows.

i) Reduction of coefficient matrix A

$$a_{ij}^{(i-1)} = a_{ij} - \sum_{k=k_0}^{i-1} \bar{a}_{ki}^{(k-1)} a_{kj}^{(k-1)} \quad (4.20)$$

and

$$a_{jj}^{(j-1)} = a_{jj} - \sum_{k=L_j}^{j-1} \bar{a}_{kj}^{(k-1)} a_{kj}^{(k-1)} \quad (4.2b)$$

in which

$$j = 2, \dots, N$$

$$i = L_j + 1, \dots, j-1$$

$$k_0 = \max(L_i, L_j)$$

$$\bar{a}_{ki}^{(k-1)} = a_{ki}^{(k-1)} / a_{kk}^{(k-1)}$$

L_j = row number of the first non-zero entry in column j of the matrix [A]

Superscript (k-1) indicates the value after elimination of the (k-1)th equation.

ii) Reduction of load vector {b}

$$b_i^{(i-1)} = b_i - \sum_{k=k_0}^{i-1} \bar{a}_{ki}^{(k-1)} b_k^{(k-1)} \quad (4.3a)$$

in which

$$i = L_b + 1, \dots, N$$

$$k_0 = \max(L_i, L_b)$$

L_b = row number of the first non-zero entry in vector {b}

and

$$b_i^{(i-1)} = b_i^{(i-1)} / a_{ii}^{(i-1)} \quad (4.3b)$$

where

$$i = L_b, \dots, N$$

Equation (4.3b) is applied after completing the operations in Eq. (4.3a).

iii). Back substitution

$$x_k = b_k^{(k-1)}, \quad k = 1, \dots, N \quad (4.4)$$

and

$$x_k = x_k - \bar{a}_{ki}^{(k-1)} x_i$$

in which

$$i = N, N-1, \dots, 2$$

$$k = L_i, L_{i+1}, \dots, i-1$$

The solution vector is generally overwritten on {b}. It should be noted that $a_{kj}^{(k-1)}$ is replaced by $\bar{a}_{kj}^{(k-1)} = a_{kj}^{(k-1)} / a_{kk}^{(k-1)}$ when the operations in Eqs. (4.2) are performed. Equations (4.3) and (4.4) may be repeated for any number of load cases.

Subroutine REDUCE is designed to reduce the frame stiffness matrices using the procedure of Eqs. (4.2). Subroutine BSUB carries out the reduction of load vectors and back substitution based on Eqs. (4.3) and (4.4). After subroutines REDUCE and BSUB have been called, vector {b} contains the solution vector of joint displacements.

4.5.3 Condensation of Frame Stiffness Matrices

The frame stiffness matrix is required to be condensed to obtain the frame sidesway matrix which consists of elements relating to the story sidesway degrees of freedom only.

For a symmetric matrix like the one shown in Fig. 4.6a which has been suitably rearranged and partitioned so that all sidesway degrees of freedom are grouped together, the condensed stiffness matrix can be expressed as:

$$K_* = k_{BB} - k_{BA} k_{AA}^{-1} k_{AB} \quad (2.22)$$

where k_{AA} is the submatrix which relates to themselves the degrees of freedom other than those corresponding to sidesway. k_{BB} is a submatrix which relates the sidesway degrees of freedom to each other, and k_{AB} and k_{BA} are submatrices which describe the cross-relation between the sidesway degrees of freedom and the others. Condensed matrix K_* is of the same size as the submatrix k_{BB} .

The condensation can thus be performed by inverting the submatrix k_{AA} and solving Eqs. (2.22). However, when the degrees of freedom are numbered in such a manner as to give the minimum bandwidth, the columns associated with the sidesway degrees of freedom are scattered within the matrix (as indicated in Fig. 4.6(b), rather than being grouped together. In such a case, Eqs. (2.22) can be applied only if the matrix of Fig. 4.6(b) is rearranged to attain the form shown in Fig. 4.6(a). But any such rearrangement will greatly enlarge the bandwidth of the matrix causing a large increase in the required storage space.

(a) Regular Matrix With Sidesway Degrees of Freedom Partitioned

(b) Banded Matrix With Sidesway Degrees of Freedom in Between

FIG. 4.6 CONDENSATION OF STIFFNESS MATRIX

An alternative method which uses the Gaussian elimination or the Crout reduction technique can be conveniently employed to remedy the above situation. The method, as used in this program (subroutine CONDENSE), is described below.

Referring to Fig. 4.6(a), let the submatrix k_{AB} be of size n by m , and the i th column vector be referred to as k_{AB}^i . Submatrix k_{BA} will contain m row vectors of length n , so that

$$\begin{bmatrix} k_{BA}^i \end{bmatrix} = \{k_{AB}^i\}^T \quad (4.5)$$

Now introduce a fictitious unknown displacement vector $\{x^i\}$ of length n so that, by treating the vector $\{k_{AB}^i\}$ as a load vector, a stiffness equation may be written as

$$\begin{bmatrix} k_{AA} \end{bmatrix} \{x^i\} = \{k_{AB}^i\} \quad (4.6)$$

Eq. (4.6) can be solved for $\{x^i\}$ as

$$\{x^i\} = \begin{bmatrix} k_{AA} \end{bmatrix}^{-1} \{k_{AB}^i\} \quad (4.7)$$

However, Eqs. (4.6) contain a set of simultaneous equations which can be solved by Gaussian elimination or the Crout reduction technique to obtain $\{x^i\}$. In fact, once $\begin{bmatrix} k_{AA} \end{bmatrix}$ has been reduced, only a load vector reduction and a back substitution are required to find $\{x^i\}$ for any particular value of i . The process can be repeated m times to find all $\{x^i\}$ s. Now

$$k_{*}^{ij} = k_{BB}^{ij} - \begin{bmatrix} k_{BA}^i \end{bmatrix} \{x^j\}$$

or

$$k_{*}^{ij} = k_{BB}^{ij} - \{k_{AB}^i\}^T \{x^j\} \quad (4.8)$$

Because $[K_*]$ can overwrite $[k_{BB}]$, Eq. (4.3) can be written as

$$k_{BB}^{ij} \Leftarrow k_{BB}^{ij} - (k_{AB}^i)^T \{x^j\} \quad (4.9)$$

Equation (4.9) is solved for all values of i and j , finally giving rise to the condensed stiffness matrix K_* .

This method can still be used when the sidesway vectors are scattered in the stiffness matrix and not grouped together. If the rows and columns corresponding to the sidesway degrees of freedom, shown shaded in Fig. 4.6(b) are deleted, the remaining matrix is k_{AA} . During forward reduction of the stiffness matrix therefore, rows and columns corresponding to the sidesway degrees are skipped. The resulting matrix contains a reduced k_{AA} matrix. To continue the operations described in Eqs. (4.7) through (4.9), the vectors contained in the L-shaped shaded areas of Fig. 4.6(b) are taken one at a time and treated as the load vectors. It should be noted that the portions shaded solid are elements of matrix k_{BB} ; they should therefore be omitted from the load vectors. These omitted elements are finally used in Eq. (4.9) to give the elements of the condensed matrix. The operations do involve some book keeping but are otherwise straightforward.

Subroutine CONDEN effectively uses the above approach with the aid of subroutines REDUCE and BSUB as described earlier. Provision is made within the subroutines to recognize the sidesway vectors and their rows in the matrix and skip or carry out operations on them as required in the procedure described above.

4.5.4 Solution of Eigenvalue Problem

The determination of mode shapes and frequencies is important for the dynamic analysis of a building, for several reasons. One of these is to develop a transformation relation by which it is possible to reduce a large number of coupled equations of motion into an equivalent set of uncoupled single degree of freedom equations.

Subroutine EIGVAL is designed to evaluate the mode shapes and frequencies of the building based on the inverse power method with shifts as described in Section 3.2. The subroutine uses the method of partitioning⁽¹²⁾ to invert a matrix. The algorithm takes advantage of the symmetry and sparsity of the matrix to be inverted and treats it as a one-dimensional array to reduce the address arithmetic (subroutine RECIP). The inverted matrix occupies the same space as the original stiffness matrix. Also, the latter is transferred to the auxiliary storage before inversion is carried out. The method as a whole is iterative and the iteration for evaluating a mode shape is continued till a convergence has been achieved⁽⁷⁾. This is supposed to happen when the ratio of the maximum difference between the absolute values of the corresponding elements of two successive trial vectors to the current value of that element is less than a preset allowable value which is 0.000001 in the standard version of the program. This method of testing for convergence is likely to create trouble when one of the elements is zero or close to zero. In such a case, the difference between the successive trial values of the zero

(very small) element may only be a random noise of very small magnitude. It will, however, be greatly magnified when divided by the current value of the element, creating an erroneous impression that the iteration has not converged, even though the convergence criterion may be satisfied for other elements of the vector. This problem is overcome by skipping the convergence test for an element which is very small as compared to the other elements of the vector. In the standard version of the program, an element with a value less than or equal to 10^{-6} in a vector normalized so that the largest element is equal to 1 is considered very small. The iteration process is also terminated if the number of iterations has reached a preset maximum, even though the convergence might not have been achieved. When this occurs, the program prints a warning message, giving the mode number being calculated and the maximum difference between the corresponding elements of the last two trial vectors. If this difference is too large and therefore unacceptable, either of the following two corrective actions can be taken and the job run again:

A smaller number may be specified in the input data for the number of modes to be extracted from one origin. Alternatively, the limit on the number of iterations may be increased. This can be achieved by changing the number in the following FORTRAN statement of the main program:

```
MI = 300
```

It should be noted here that the inverse power method is very efficient when only the first few modes are to be extracted. However, using the algorithm of the subroutine EIGVAL, the accuracy and the

computational efficiency is further improved by periodically shifting the origin on the eigenvalue axis.

4.5.5 Optimization of Computer In-Core Storage Space

Since program ANGELS is a fairly large analysis program, a considerable amount of computer storage is required to accommodate the instructions. Also, the analysis of a large building requires a fair amount of core space. To minimize the core-space requirement, therefore, a dynamic storage allocation scheme is devised within the program for major variable arrays by storing them in a blank COMMON block. Because dynamic allocation of storage requires that the variables sharing the COMMON space be transferred between subroutines and between subroutines and the main program as arguments in the calling statements, this results in some loss of efficiency. To optimize the situation, one major array (of frame properties), which is repeatedly required by the program, is kept in labelled COMMON block called/COM1/ instead of being placed in the blank COMMON.

In the blank COMMON block, the amount of high speed storage required for a particular problem may be changed by altering the following two FORTRAN statements at the beginning of the main program:

```
COMMON SK(n)
```

```
MCORE = n
```

where n is the storage requirement for the major arrays.

For a given building, the value of n is computed as follows:

Let

N = total number of stories in the building

NF = total number of frames in the building

NQ = size of the global stiffness matrix

= $3 \times N$ when a three-dimensional analysis is required

= N when two-dimensional analysis is required, as would be the case, for example, symmetrical buildings.

NM = number of modes to be considered.

For the complete building, check

$$1. \quad N_1 = NQ(NQ+1) + (N+NF)(N+2) + NF.$$

If dynamic analysis is required, check

$$2. \quad N_2 = NQ(NQ+2NM+1) + NF(N+2) + NM + MP$$

where

$$MP = NQ * NM / N$$

but not less than NQ .

Also, for the largest frame,

NC = total number of columns in the frame

and

$$3. \quad N_3 = 14 \times NC + NE$$

where

NE = total number of elements in the frame stiffness matrix, that are required to be stored.

NEL can be obtained by running the 'front end' program AIDS, or can be

approximately calculated as follows.

If NT = number of towers in the frame, for a setback frame (for a regular frame $NT=1$)

NB_i = number of bays in tower i

NS_i = number of stories in tower i

and if

$$NY_i = 4(NB_i)^2 + 15 * NB_i + 13$$

then NEL can be approximately obtained (within 10% above) by the integer value of

$$NE = \{ (NS_1 - 0.5)NY_1 + \sum_{i=2}^{NT} (NS_i + 0.5)NY_i \}$$

The frame which leads to the largest value of N_3 may not be readily recognized. In that case, two or three frames may have to be checked out.

The value of n is then the maximum of N_1 , N_2 and N_3 .

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 SUMMARY

This work was concerned primarily with the development of a set of two computer programs for the three-dimensional linear analysis of multistoried building structures.

The first package in this set called AIDS is written in FORTRAN IV and is an interactive program designed to serve as a 'Front-End' processor to the second part. It reads all input data, required for the three-dimensional analysis of a building, in a conversational mode with a rapid data input option. The program then rearranges the supplied data in proper format to be compatible with the main analysis program and writes it into three disc files or tapes. The first of these files contains the general control and information data and the data required for the seismic analysis, if any; the second file contains the frame and member properties data stored sequentially, and member by member within a frame group. Data is stored for only one frame in a group of frames having similar properties (called a frame group). The third file contains the gravity load data for each frame and for each gravity load case followed by the data for the lateral load cases. The program keeps a duplicate version of user's input and stores it sequentially along with appropriate identifying comments in a fourth file. This copy is used

for any correction to the user supplied input which may be required when the user makes one or more mistakes in typing in the data during execution of the program. This can be achieved by editing the duplicate file after the program has run and then rerunning the program, this time specifying the option that the data will be read from the duplicate file instead of from the terminal. During execution of the program, the user can correct only those errors in a line of supplied data which he detects before hitting the return key at the end of that line. The questions asked for by the computer are designed to be self-explanatory so that the user has to put in only a minimum of effort in learning to use the conversational data input facility. During execution, the program prompts the user to type in the required data, which is accepted in a free format, relieving the user of the rather inconvenient task of learning the rigid data input formats of standard FORTRAN.

The second program, called ANGELS, also is written in FORTRAN IV language and is designed to perform a three-dimensional linear analysis of buildings.

For the purpose of analysis, the program assumes that the floor diaphragms of the building are infinitely rigid in their own planes and the building rests on a foundation which is axially and flexurally rigid and lies in one horizontal plane. Assumption of rigid diaphragm implies that the beams are axially rigid. However, columns are considered axially deformable. Flexural deformations are taken into account

for all beams and columns. At the user's option, shear deformations and finite joint sizes can also be accounted for.

Since the floor diaphragms are axially rigid and the columns are inter-connected through these floors, there are only three displacement coordinates at each floor level - two translations and one rotation about the vertical axis. The mass of the building is also assumed to be lumped at the floor levels.

The program can analyze a building for a maximum of four different gravity load cases, three different lateral load cases and for one design earthquake response spectrum. If desired, the earthquake forces can be modified to take account of the ductility in the structure. Individual load cases can be weighted and superimposed to obtain any number of load combinations. Member forces are worked out for each such load case. The results of the analysis are printed out in the form of member end actions at the faces of the joints, or at the center of joints when finite joint sizes are not considered. Also, at the option of the user, the program evaluates and prints out the three-dimensional mode shapes and frequencies of the building and 'Absolute Sum' and 'Root-Sum-Square' values of story earthquake forces and displacements. The program also provides an echoprint of all input data.

Because of the lengthy computations and the large volume of output, the program is designed to run in a remote batch mode, taking in all data from the three disc files generated by the 'front end' program

AIDS. However, program ANGELS can also work independently of program AIDS. In that case, the input data must be prepared on data cards in appropriate format and then copied to three unkeyed disc files.

5.2 CONCLUSIONS

1. With the aid of an interactive conversational program, which is executed in a time-sharing mode, data can be read at the user's terminal in a free format. Such an interactive program can also be designed to speed up the data input considerably when advantage is taken of the repetitive nature of building dimensions, member properties and loading. By making use of a duplicate file, corrections can be efficiently made in the input data supplied through the conversational input data program. The above advantages can be realized by writing a front-end processor to the main analysis program. Such a processor written in the conversational mode is thus a very convenient means for the preparation and correction of input data for a fairly large or very large analysis program. It can save a substantial amount of engineer's time which he would otherwise have to devote in learning a complex set of instructions for the use, requirements and data input formats of a program. An additional advantage of the front-end processor is that it can be run on an in-house mini-computer. Data preparation and checking thus becomes simpler and more economical. The present study confirmed the above advantages of a front-end processor written in conversational mode.

2. For linear three-dimensional analysis of a building, consideration of floors as rigid diaphragms, an assumption which is reasonable for most well designed buildings, reduces the number of unknown variables considerably. The resulting structural (global) stiffness matrix is comparatively small in size and leads to substantial savings in the computational effort. However, the results obtained on the basis of this assumption are sufficiently accurate for all practical purposes. Similarly, the assumption that the mass of the building is lumped at the floor levels is also quite reasonable. With this assumption, the mass matrix generated by the program is diagonal and of the same size as the global stiffness matrix. This greatly simplifies the eigenvalue problem, leading to saving the computational effort in the iterative solution for the mode shapes and frequencies of the building.
3. The use of compacted column storage scheme for the frame stiffness matrices is most economical in terms of computer storage requirements. Also, dynamic storage allocation in which major variable arrays are assigned storage space during run time is an effective means of reducing computer core space requirements.

5.3 RECOMMENDATIONS FOR FURTHER DEVELOPMENTS

The study has indicated the need for further development in the following areas related to computer programming for linear three-dimensional analysis of building structures.

1. Study of the Presence of Shear Panels and Infilled Shearwalls in a Building

Shear panels (a shearwall divided at the base into two supporting columns) and infilled shearwalls are sometimes used in multistory buildings. At the present stage, the program does not have the ability to take account of substructures of the above type.

Inclusion of some members in analysis by modifying the subroutines for the assembly of frame stiffness matrices and calculation of member forces will make the program more general.

2. Study of the Effect of Cantilever Spans Beyond the Exterior Spans

In a practical design situation, the building structures might have cantilever spans beyond the exterior spans. The programs developed for this project can not handle exterior cantilever spans. The generality of the present program can be extended by accounting for cantilever spans in the three-dimensional analysis of the buildings.

3. Study of the Effect of Non-Prismatic Beams and Columns

For practical reasons, a building may sometimes have beams with haunches at their ends or columns with capitals as in flat slab structures. The program, in its present form, is not able to handle such practical situations. The program may be extended to include the presence of such non-prismatic members.

4. Modification of Reference Axes for Beams and Columns

The program described in this thesis considers the floor beam center lines and column center lines as basic reference axes for the frames. It assumes that all beam center lines lie in a common plane and that all columns in the vertical stack have a common center line. But, in practical buildings, the top surfaces of beams rather than their center lines lie in one plane. Also, for architectural reasons, the exterior columns are generally flush on their outer faces; some interior columns also may not have their center lines coincident. An analysis based on the assumption of coplanar center lines may therefore lead to some errors. The magnitude of these errors needs some study.

The program can be suitably modified to take into account the actual rather than the idealized locations of the beam-column center lines.

5. Study of the Inclusion of Semi-Rigid and Hinged Joints and Foundations

The present study is concerned with the analysis of buildings which have rigid beam-column joints and are attached to a flexurally rigid foundation. But, a practical building, particularly one in steel, may have hinged or semi-rigid joints. Also, the foundation of the building may not be essentially rigid. Inclusion of semi-rigid and hinged joints and supports in the present analysis program will increase its generality.

6. Study of Effect of Actual Earthquake Acceleration in the Analysis

In the present study, the seismic analysis of buildings is based on a design earthquake response spectrum. The present work may be extended by incorporating into it the ability to carry out a time history analysis for a given earthquake acceleration record.

7. Study of the Presence of Braced Frames in the Building

Some buildings particularly those of steel may contain frames with inclined bracings. The present program may be extended to take into account the effect of bracings in the building frames.

8. Study to Run the Present Program in a Minicomputer

The present program is designed to run in batch mode on a computer with a fair size of memory. Since in-house minicomputers are finding increasing use in engineering design offices, it will be worthwhile adapting this program to run on a minicomputer. It should be possible to accomplish this without too much effort.

REFERENCES

1. Humar, J.L., "A Computer Program for the Analysis of Reinforced Concrete Frames", Department of Civil Engineering, Carleton University, Ottawa, Canada, January 1978.
2. Gere, J.M. and W. Weaver, "Analysis of Framed Structures", D. Van Nostrand Co. Inc., Princeton, N.J., 1965.
3. "Commentaries on Part 4 of the National Building Code of Canada 1975", Supplement No. 4 to the NBC, National Research Council, 1975.
4. Firmin, A., M.J. Gilmour and R.A. Collins, "Computer Programs for Structural Analysis and Design - A Canadian View", Canadian Journal of Civil Engineering, Vol. 4, 1977.
5. Clough, R.W. and J. Penzien, "Dynamics of Structures", McGraw-Hill.
6. Hurty, W.C. and M.F. Rubinstein, "Dynamics of Structures", Prentice Hall.
7. Humar, J.L., "Eigenvalue Programs for Building Structures", Computers and Structures, Vol. 8, pp. 75-91, Pergamon Press, 1978.
8. Boyer, L.T. and J.C. Carson, "Experience with Interactive Computing System", ASCE, Journal of Structural Engineering Division, Vol. 97, No. ST1, January 1971, pp. 289-299.
9. Hall, A.S., "Joint Deformation in Building Frames", The Institute of Engineers, Civil Engineering Transactions, April, 1969, Australia.
10. Gluck, J., "Lateral Load Analysis of Irregular Shearwall Multistory Structures", Journal ACI, July 1970, pp. 548-552.
11. MacLeod, I.A., "Lateral Stiffness of Shearwalls with Openings", Tall Building Conference, Proceedings, University of Southampton, April, 1966, pp. 231-243.
12. Willems, N. and W.M. Lucas, "Matrix Analysis for Structural Engineers", Prentice Hall, 1968.
13. Meek, J.L., "Matrix Structural Analysis", McGraw-Hill.

14. "National Building Code of Canada", 1975.
15. Humar, J.L., "Seismic Analysis of Multistory Buildings", Department of Civil Engineering, Carleton University, Ottawa, Canada, 1976.
16. Wilson, E.L., "SOLID SAP - A Static Analysis Program for Three Dimensional Solid Structures", University of California, Structural Engineering Laboratory Report No. UC-SESM 71-19, March 1970.
17. Wright, W., "Structural Design by Computer", 1975.
18. Wright, W., "SUBROUTINE REED", The CAD Project, Technological Research and Development Branch, Public Works, Canada, May 2, 1978.
19. Wright, W., "The Use of Conversational Computer Programs in Structural Engineering Offices", Canadian Journal of Civil Engineering, Vol 4, 1977, pp. 417-435.
20. Wilson, E.L. and H.H. Dovey, "Three Dimensional Analysis of Building Systems - TABS", Earthquake Engineering Center, University of California, Berkley, California, December 1972.
21. Watson, R.W., "Timesharing System Design Concepts", McGraw-Hill, 1970.
22. Mondkar, D.P. and G.H. Powell, "Towards Optimal In-Core Equation Solving", Computers and Structures, Vol. 4, pp. 531-548, Pergamon Press, 1974.

APPENDIX A

LINEAR THREE-DIMENSIONAL ANALYSIS
OF BUILDING STRUCTURES

PROGRAM USER'S MANUAL

August 1978

DEPARTMENT OF CIVIL ENGINEERING

CARLETON UNIVERSITY

OTTAWA, CANADA

DISCLAIMER

The program package described here consists of a set of computer programs designed to perform a three dimensional linear analysis of building structures with rigid frames and shearwalls for gravity, earthquake and lateral load cases. While every effort has been made to ensure the accuracy of the information presented in this manual, and the reliability of the programs described therein, use of this material can be made only with the understanding that neither the programmer nor the manual writer will assume any liability of any kind arising from such use.

DISCLAIMER

The National Library of Canada also
does not assume any liability of
anykind arising from the use of
information presented in this manual.

CHAPTER A1

INTRODUCTION

A1.1 GENERAL

The programs described in this user's manual constitute a part of a thesis submitted by the author to Carleton University, Ottawa, Canada, in partial fulfillment of the requirements of an M.Eng. degree in Civil Engineering. The programs were developed for a linear three-dimensional analysis of building structures.

The first program is a conversational program, called AIDS⁺, designed to read the input data required by the second program which is the main analysis program. The analysis program named ANGELS^{*} is designed to perform a linear three-dimensional analysis of building structures for gravity, earthquake and lateral loads. The program calculates and prints out the member forces for different load cases. At the user's option, it can also calculate and output the mode shapes and frequencies of the building. The building to be analyzed should consist of two orthogonal sets of rectangular frame or frame-shearwall substructures. The joints and the foundations are considered rigid and the stiffness method of analysis is used.

+ AIDS - Automatic Input Data Supplier

* ANGELS - Analysis of buildings for Gravity, Earthquake and Lateral loads

To make this manual self-contained, a brief description of the algorithms and program operations is included. For a more detailed presentation of these topics, the original thesis may be referred to. This manual also presents outline flow charts, a list of FORTRAN variables used, instructions for the use of the programs, the FORTRAN source listings and an example of a three-dimensional analysis of a building.

A1.2 ASSUMPTIONS AND LIMITATIONS

An exact three-dimensional analysis is required for only a limited number of buildings. For a majority of buildings, the following two basic approximations can be made which greatly simplify the preparation of input data and significantly reduce the computational effort.

1. It is assumed that the floor diaphragms are rigid in their own planes. This is a realistic approximation for a large number of common buildings. As a result, the structure has three degrees of freedom at each floor level - two translational, in the X and Y directions, and a rotational about the vertical axis of the floor.

The other degrees of freedom include: the bending deformation in the horizontal beams and floor slabs, and a vertical displacement and a rotation at each column or shearwall at each floor level.

2. It is further assumed that the horizontal lateral loads act at the floor levels and are transferred to the columns

through the rigid floor diaphragms. Also, the mass of the building is assumed to be lumped at the floor levels. As a consequence, the active dynamic degrees of freedom at each floor are three, corresponding to the three displacement degrees of freedom mentioned in 1 above.

Also, in modelling the complete structure, the building is assumed to consist of two sets of planar frames or frame-shearwall systems which are orthogonal to each other in plan view. Isolated shearwalls within a set are considered to be frames consisting of a single continuous column line. Each frame is treated as an independent substructure. A substructure is permitted to have two degrees of freedom per joint (vertical displacement and inplane rotation) and one lateral translation per story. However, of all the above degrees of freedom, the only active dynamic degrees of freedom are those associated with the frame lateral displacements at the floor levels. The substructure stiffness matrix is assembled into the structural (global) stiffness matrix on the assumption that the structure has three active degrees of freedom per floor: two translational and one rotational.

The following additional approximations are also inherent in the present approach:

1. Complete compatibility with regard to joint displacements is not enforced at joints which are common to two intersecting frames. Thus, in a column which is common to two intersecting frames, the axial deformations will not be the same (compatible).

However, for design purposes, a reasonable approximation to the axial load in such a column may be obtained by adding those calculated for the component frames. As for joint rotations, since the intersecting frames are assumed to be perpendicular in plan view, the in-plane joint rotations of the frames are orthogonal to each other and are hence uncoupled.

2. As the floor diaphragms are assumed to be rigid in their respective planes, it is apparent that the axial deformation is not permitted in the beams. However, bending stiffness of the floor slabs may be included approximately in modelling of the individual frames. All beam centerlines in a floor are assumed to be coincident with the central plane of the floor diaphragm. Similarly, columns for all stories, located in one vertical stack are assumed to have the same center line.
3. The torsional resistance of any member section is neglected.

A1.3 IDENTIFICATION OF STRUCTURE GEOMETRY AND LOADING

For the purpose of preparing the numerical input data, the building to be analyzed is at first resolved into two orthogonal sets of planar frame or frame-shearwall systems. Each frame or frame-wall system must be assigned two identifier numbers. The first identifier is a frame group number; frames with similar geometric and structural properties being collected into one group. A typical building, such as

the one shown in Fig. A.1, may have several frame groups. Each group is assigned a number starting from one and increasing sequentially. The second identifier is a unique number assigned to each frame, called the frame number. This number also starts from one and increases sequentially. It is, however, a requirement that the frames in Group I are numbered first, those in Group II are numbered next, and so on, till all the frame groups are exhausted. Also, within a group, frames carrying exactly the same gravity loads must be numbered so that they form a cluster uninterrupted by frames with different loading patterns.

Gravity loads are to be specified in terms of uniformly distributed loads and concentrated loads on each beam in a frame. A maximum of four concentrated loads can be specified on any one beam. Lateral loads are applied at the floor levels and act on the complete structure. The values of their X and Y components along with the coordinates of their points of action with reference to the global reference point should be supplied to the program.

The program also needs the orientations and locations of the frames to assemble the global stiffness matrix. Further, it requires the location of mass center and the translational and rotational masses at each floor level to assemble the global mass matrix. The frame locations and mass centers must be specified with respect to the global reference points at each floor level. The choice of a reference point is arbitrary, but all reference points must lie on the same vertical line. Also, it is to be noted that the line of action of the earthquake force resultant acts through the center of mass at each floor level.

Each frame or shearwall element is assumed to have stiffness only in the in-plane direction. Elements which have stiffness in another direction too must be defined by an additional element. For example, the properties of columns which are common to two different frames must be supplied twice. As stated earlier, the axial deformations in these doubly defined columns will not be the same. However, this incompatibility should not cause any serious error in the analysis.

The properties of a frame or shearwall are specified with respect to its local stiffness direction. Floor beam center lines and column center lines are the basic reference lines used in the frame description. The height from one floor beam center line to the next is assumed to be the same for all frames and is equal to the story height. A member may be omitted by specifying zero properties. However, a shearwall must be continuous all through to the base of the frame. Beams must be prismatic and columns may be prismatic or circular. Both flexural and shearing deformations are considered in the analysis. Beams are considered axially rigid but axial deformations of columns are accounted for. Also, finite joint sizes of infinite rigidity may be considered. In general, each frame is considered a setback frame. A part of the frame height containing the same number of bays is named a tower; a frame containing more than one tower is called a setback frame as illustrated in Fig. A.2. A uniform frame is considered as a setback frame with a single tower.

- Tower : part of frame height with same no. of bays
 NST : no. of stories in a tower
 NBA : no. of bays in a tower
 NBS : no. of bay-setback between two towers at left

FIG. A.2 ELEVATION OF A SETBACK FRAME

In the analysis, a maximum of four different gravity load cases, a maximum of three different lateral load cases and a seismic design spectrum case may be considered. Any or all of the above load cases can be combined to form load combinations. There is no limit on the number of combinations. Each such combination, called a 'load system' is fully specified once the multiplying factors for its component load cases have been supplied.

AL.4 SIGN CONVENTIONS AND UNITS

The positive directions of global axes are indicated in Fig. A.3a. The local axis of a frame will be in the plane of the frame and will be parallel to the global X-axis or global Y-axis. The positive direction of a local axis is the same as the positive direction of the global axis to which it is parallel. Also, for forces and displacements, the following set of conventions is used:

1. Forces and Displacements
 - They are positive when directed upwards or to the right.
2. Moments and Rotations
 - They are positive when counter clockwise.
3. Column Shears
 - A column shear is considered positive when produced by a top reaction directed to the right.
4. Column Axial Force
 - A tensile force is positive.

(a) Forces and Displacements

(b) Moments and Rotations

(d) Column Shears

(c) Loads

(e) Column Axial Loads

FIG. A.3b Sign Convention For Forces And Displacements

5. Gravity Loads on Beams

- They are positive when downward.

The sign conventions are illustrated in Fig. A.3b.

The program accepts any consistent set of units; no transformation of units is carried out within the program. A typical set of consistent units is shown below.

Length, displacement	feet
Force, weight	kips
Time	seconds
Mass	kip-sec ² /ft
Moment	kip-ft
Acceleration due to gravity (g)	ft/sec ²
Spectral acceleration	in units of g
Frequency	rad/sec
Period	seconds

A1.5 JOB CONTROL COMMANDS

During development of the programs, the input data program AIDS was compiled and run on an IM-70 minicomputer. The main analysis program - ANGELS was compiled and run on a HONEYWELL Sigma-9 computer. The job control statements (or cards) required to compile and run the programs are indicated below.

A1.5.1 Compilation and Running of Program AIDS

In an IM-70 minicomputer, the System Utility Codes are as follows:

LØ	means the list output file
LI	means the list input file
SI	means the source input file
SØ	means the source output file
BI	means the binary input file
BØ	means the binary output file

The physical units are:

AØ	console terminal
A1	line printer
F0	floppy drive 0
F1	floppy drive 1
F2	floppy drive 2
F3	floppy drive 3

The meaning of some essential commands are:

ØS	operating system
AS	assign
SF	system file,
SS	scratch source
FØRT:4	FORTAN IV compilation
LNK	link
LDR	load and run

EL	error listing only
LL	
UL	unconditional link
TE	terminate operations
EX	execute

Let the program AIDS (source program) be on a floppy disc in drive F1 and the system file be on a disc in drive 0. Suppose the program is to be compiled and the object module (the binary output after compilation) is to be stored on a disc in drive F2 on a file named BAIDS. The commands necessary for compilation are:

```

>/AS  LØ = A1          (CR)
>/AS  SI = F1.AIDS     (CR)
>/AS  BØ = F2.BAIDS    (CR)
>/AS  SS = F3          (CR)
>/EX  FORT:4,EL        (CR)

```

where (CR) means press 'carriage return'.

After the above commands have been issued, the minicomputer will start compiling the program AIDS; the binary object module will be created and stored in file BAIDS on disc in drive F2.

A free format read subroutine REED is also required to compile and link with BAIDS during creation of the load module in the next step. Supposing that the object code of REED is on file REEDB, the load module of the program can be obtained as follows.

Let the object file BAIDS be in disc F1 and the object file REEDB in F2. Suppose that the load module is to be directed to file *AIDS on disc F3. Then the following control commands will be required to obtain the load module:

>/AS LØ = A1 (CR)

>/AS B1 = F1.BAIDS (CR)

>/AS BØ = F3.*AIDS (CR)

>/AS LI = F2.REEDB (CR)

>/EX ØS:LNK,LL (CR)

When these commands have been issued, the computer will display on the terminal an informative summary and will prompt the user with a set of questions to which the user should respond as follows:

?/AS LI = F0.F:ØSLB (CR)

? LL,TE (CR)

The load module will then be created and stored on file *AIDS on disc F3. This load module can be run whenever input data is to be supplied in a conversational mode. The commands necessary to run a job are as follows:

>/AS 1 = A0 (CR)

>/AS 2 = A0 (CR)

>/AS 3 = F1.RID1 (CR)

>/AS 4 = F1.RID2 (CR)

>/AS 5 = F1.RID3 (CR)


```
>/AS 6 = F2.DUPLI (CR)
```

```
>/AS BI = F3.*AIDS (CR)
```

```
>/EX ØS:LDR,EX (CR)
```

After these commands are typed in, the computer will display on the terminal a message stating that program AIDS is ready to read the input data for program ANGELS. The computer will then prompt the user with a series of questions and will invite immediate answers to them. After the program has run the input data will be stored on three files RID1, RID2 and RID3 on the disc in drive F1. Also, a duplicate version of the user's input will be written on file DUPLI on the disc in drive F2.

Al.5.2 Compilation and Running of Program ANGELS

As a first step in running a job, files RID1, RID2 and RID3 obtained from program AIDS are transferred from the minicomputer to the HONEYWELL computer using appropriate commands. These data files are stored in the user's account on HONEYWELL on three unkeyed files. The following job commands may then be used to run program ANGELS, a source of which exists in the user's account:

1. Compilation and running:

```
!JOB ACCOUNT NO, ACCOUNT NAME, 5,
!LIMIT (TIME,4), (CORE,40), (LØ,100)
!ASSIGN F:1, (FILE,RID1), (INOUT), (REL)
!ASSIGN F:2, (FILE,RID2), (INOUT), (REL)
!ASSIGN F:3, (FILE,RID3), (INOUT), (REL)
!ASSIGN F:4, (FILE,RID4), (OUTIN), (REL)
!ASSIGN F:5, (FILE,RID5), (OUTIN), (REL)
!ASSIGN F:6, (FILE,RID6), (OUTIN), (REL)
!ASSIGN F:7, (DEVICE,LP)
!ASSIGN F:8, (FILE,RID3), (OUTIN), (REL)
!ASSIGN M-61, (FILE,ANGELS), (IN), (SAVE)
!FORTRAN BC,GO
!LOAD (GO)
!RUN
```

2. Compilation and making a load module

```
!JOB ACCOUNT NO,ACCOUNT NAME,5,
!LIMIT (TIME,5), (CORE,40) ,
!ASSIGN F:1, (FILE,RID1), (INOUT), (REL)
!ASSIGN F:2, (FILE,RID2), (INOUT), (REL)
!ASSIGN F:3, (FILE,RID3), (INOUT), (REL)
!ASSIGN F:4, (FILE,RID4), (OUTIN), (REL)
!ASSIGN F:5, (FILE,RID5), (OUTIN), (REL)
!ASSIGN F:6, (FILE,RID6), (OUTIN), (REL)
!ASSIGN F:7, (DEVICE,LP)
!ASSIGN F:8, (FILE,RID8), (OUTIN), (REL)
!ASSIGN M:SI, (FILE,ANGELS), (IN), (SAVE)
!FORTRAN BC,GO
!LOAD (GO), (LMN,*ANGELS), (PERM)
```

If the user wants to run the load module *ANGELS at the same time, he should add the following command to the above run:

```
!RUN (LMN,*ANGELS)
```

This command will initiate a run and the results will be printed out on the line printer. Also, the load module will be saved in the user's account on file named *ANGELS.

3. Running a job with a previously created load module

```
!JOB ACCOUNT NO,ACCOUNT NAME,5,
!LIMIT (TIME,5), (CORE,40)
!ASSIGN F:1, (FILE,RID1), (INOUT), (REL)
!ASSIGN F:2, (FILE,RID2), (INOUT), (REL)
!ASSIGN F:3, (FILE,RID3), (INOUT), (REL)
!ASSIGN F:4, (FILE,RID4), (OUTIN), (REL)
!ASSIGN F:5, (FILE,RID5), (OUTIN), (REL)
!ASSIGN F:6, (FILE,RID6), (OUTIN), (REL)
!ASSIGN F:7, (DEVICE,LP)
!ASSIGN F:8, (FILE,RID8), (OUTIN), (REL)
!RUN (LMN,*ANGELS)
```

4. For small size problems, the program can be run on line and the result directed to the user's terminal. The following on-line commands should be used:

!SET F:1/RID1; INOUT; REL (CR)
!SET F:2/RID2; INOUT; REL (CR)
!SET F:3/RID3; INOUT; REL (CR)
!SET F:4/RID4; OUTIN; REL (CR)
!SET F:5/RID5; OUTIN; REL (CR)
!SET F:6/RID6; OUTIN; REL (CR)
!SET F:7 ME (CR)
!SET F:8/RID8; OUTIN; REL (CR)
!*ANGELS.

CHAPTER A2PROGRAM FOR INPUT DATA - AIDSA2.1 DESCRIPTION OF PROGRAM

Program AIDS, written in standard FORTRAN IV, is designed to act as a 'front end' processor to the main analysis program ANGELS. It reads all input data required for the analysis, in a conversational mode. The data input is in free format. To achieve free format inputs, a special subroutine, called REED⁺, has been used in this program. The input is supplied via a time-sharing terminal. One or more data items can be read from a line, consecutive data items being separated by one or more blanks. The data can be numeric or alphanumeric. For numeric data, no decimal is required for an integer, also a real whole number can be supplied as an integer, the decimal being optional.

If the building to be analyzed is large in size, with a great number of frames, a large amount of data will be required by the program ANGELS. However, very often, several dimensions, sizes and loads in a structure are repetitive. For example, most story heights may be equal; also, member sizes may be the same over a number of bays and stories. It is possible to speed up the data input considerably by taking advantage

+ The general purpose 'free format' subroutine has been developed by Dr. W. Wright of Public Works, Canada.

of these repetitions. In program AIDS, a very careful attention has been given to this aspect and ways have been devised for speeding up the input process without making it too complex.

An outline flow diagram of program AIDS is shown in Fig. A.4. It is clear from the flow diagram that the program reads all input data in a conversational mode, rearranges it and writes it on three disc files on units 3, 4 and 5 respectively. Also, the program is designed to keep a copy of the user's input on a disc file assigned to unit 6.

The program also provides the facility for correcting errors in the input data. This is achieved by creating a duplicate input file on unit 6. For major errors, the user can edit this file after the program has been run, using the conventional edit system of the computer. He can then rerun program AIDS, this time specifying that the input data be read from the disc file instead of from the terminal. However, for minor numerical errors made during the conversational input, the user can appropriately edit the files written on unit 3, 4 or 5. For example, suppose that when asked for modulus of elasticity for beam in Story No. 2 and Bay No. 2, of the frame in Fig. A.1, the user has input a value of 2500 instead of 3000. After program AIDS has been run, the user can edit the file on unit 4 (file for member properties) substituting the correct value of modulus of elasticity for the wrong value.

Fig. A.4. Outline flow diagram for program AIDS.

FIG. A.5 REINFORCED CONCRETE BUILDING UNDER GRAVITY, EARTHQUAKE AND LATERAL LOADS.

(e) Frames 1 & 2

(f) Frames 3 & 4

(g) Frame 5

FIG. A.5(contd) Gravity Loads On The Building

(h) Story 1

(i) Story 2

(j) Story 3

FIG. A.5(contd) Lateral Loads on The Building

(k) Story 1

(l) Story 2

(m) Story 3

FIG. A.5(contd) Centers of Masses

A2.2 INSTRUCTIONS FOR DATA INPUT

For the purpose of analysis, the building must first be modelled in a manner described in Section A1.3. The building is resolved into two orthogonal sets of frames. Each frame must be given an individual number and also a frame group number. In preparing the input data, the following points must also be noted:

1. A fixed reference point must be selected (usually a corner of the building) from which the distances of the frames and their X- or Y-orientations are to be specified.
2. Member sizes must be specified as follows: for rectangular sections, the section dimensions; and for standard steel sections, moment of inertia, area, and width (or depth) in the plane of the frame to which the member belongs.
3. The gravity loads must be specified in terms of a uniformly distributed load and concentrated loads if any on each beam. A maximum of four concentrated loads can be specified on a beam. A maximum of four independent sets of gravity load cases may be considered.
4. For lateral loads, the x- and y-components of loads and the distances of their lines of action from the reference axes must be computed and specified for each floor of the building. A maximum of three different lateral load cases may be considered.

5. For dynamic analysis, the mass of the building assigned to each floor level must be input. For an unsymmetrical building where the lateral and torsional vibrations are coupled, the mass moment of inertia should also be specified. The centers of these masses as measured from the reference point are also to be supplied. When a spectral analysis is required, the period and acceleration coordinates of the spectrum should be specified. In addition, ductility factor, a scale factor for the design acceleration and acceleration due to gravity must also be supplied.

The essential features of data input for the program AIDS are illustrated with reference to the building of Fig. A.5. The conversational data input to a minicomputer is shown in Section A2.5. Those portions of the dialogue which are keyed in by the user are underlined; the remaining portions are prompted by the computer.

The first question that the program asks the user is whether he wishes to make a copy of the input data on a disc file. If the data is currently being supplied through a terminal, it is a good practice to make a copy. The copy can be later used to make any corrections that may be required in the data supplied during the current session on the terminal. The user can express his wish to create such a file by typing a Y. With this option, the program will make a copy of the currently supplied input data on a disc file on unit 6; the name of the file being predetermined during assignment of units.

The option to read the data from a previously created disc file instead of from a terminal, is exercised at the question immediately following.

The next few questions which are self explanatory, deal with structure and frame description. A subsequent question prompts the user to supply the story heights. The story heights are supplied starting from the lowest story and proceeding upwards. For the present example, the user types in 120, in reply to a request for story height between levels 0 and 1. Because the height of the next story is also 120, the user can speed up the data input at this stage by typing in 120 TO 2. If he does so, the program automatically assigns a value of 120 to both of the first and the second stories and then prompts the user to supply the story height between levels 2 and 3; the user then types in 100.

The next question asked by the computer relates to the frame information and left identification. The left identification is a symbol as put on the drawing to identify the left side of a frame in a building.

The program then asks for frame group properties starting from Group I. The program first prompts the user to supply the lengths of bays for frame Group I, starting from the left, and proceeding to the right. A procedure similar to the one described for the story heights can be used in supplying this data.

Beam cross-section properties for frame Group I are input next. The computer asks whether all beams at Story No. 1 are of rectangular concrete sections or other (i.e. steel sections, etc.). If they are all concrete, the user types in C. The program then asks for the width and the depth of the

beam in Bay No. 1 (the leftmost bay). To speed up the input, the user types in 10 20 TO 2. With this, the program automatically assigns a size of 10 by 20 to the first and second beams of Story No. 1. Since all the beams at Story No. 1 have been input, the computer asks whether the beams in the next story are exactly like those in any preceding story (here the first story only). In fact, the beams in Story No. 2 are exactly of the same size as those in Story No. 1. The user can, therefore, speed up the input by typing in L(IKE) 1. The computer automatically assigns appropriate sizes to beams in story 2 so that these beams are exactly the same as the corresponding beams in story 1.

It is to be noted that when asked for the type of section, the user has the option to type in C or O. For type C, the program asks for the beam width and depth. Program ANGELS uses these properties to calculate the area and moment of inertia for the gross section. The computed values are then used in the analysis. For cross-sections other than rectangular or for the case where the user does not want to use the gross section properties, he should type in O. In that case, the user would be asked to supply the moment of inertia and depth in lieu of width and depth of the beams. Again, the speed of input can be enhanced by using the procedure described above.

Column cross-section properties are supplied in a manner similar to that followed for the beam cross-section properties.

Very often, several frames in a structure carry exactly the same gravity loads. When frames carrying similar gravity loads also belong to the same frame group, program AIDS can enhance the speed of data input. For example, with reference to the building of Fig. A.5, in which

frames 1 and 2 belong to the frame Group I, the beams in frames 1 and 2 carry exactly the same gravity loads (in each of the gravity load cases). After the gravity load data has been supplied for frame 1, the computer asks whether the next frame (i.e. frame 2) carries exactly the same gravity load as frame 1. The user now types in Y. The program makes a note of this on the disc file and asks for the load data for frame 3. If frame 3 were also in frame Group I and the gravity loads on it were similar to those in frame 1, the user could further speed up input by typing in Y TO 3.

A2.3 DEFINITION OF FORTRAN SYMBOLS

Variables appearing in program AIDS are listed here. The local variables within subroutine REED are not listed. Indices, counters, and variables providing temporary locations have been grouped together. In the following, type A stands for an array and type S for a scalar.

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
AB	A	areas of cross-section for beams
AC	A	areas of cross-section for columns
AFLX	A	array to store the floating point numbers read by subroutine REED
AG	S	acceleration due to gravity
B	A	bay lengths for a frame group

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
BETA	S	angle of incidence of earthquake with respect to the global x-axis
BI	A	moments of inertia for beams
BK	A	shear coefficients for beam sections
CI	A	moments of inertia of columns in a frame group
CK	A	shear coefficients of column cross-sections in a frame group
CNL	A	array containing concentrated loads on different beams in a frame
DB	A	depths of beams in a frame group
DBX	A	widths of beams in a frame group
DC	A	widths of columns in the plane of the frame
DCX	A	widths of columns perpendicular to the plane of the frame
DDN	S	perpendicular distance of a frame from the reference point
DHX	A	x-coordinates of the points of action of story lateral loads
DHY	A	y-coordinates of the points of action of story lateral loads
DIS	A _n	distances of the concentrated loads on beams from the respective left column center lines
EB	A	modulii of elasticity of beams
EC	A	modulii of elasticity of columns
FBND	A	array containing real input data bounds
GB	A	modulii of rigidity of beams

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
H	A	story heights
HLX	A	x-components of story lateral loads
HLY	A	y-components of story lateral loads
IAL	A	array of single character alphanumeric variables read and processed by subroutine REED
IAX	A	array containing the name of the project
IBC	A	indices for the type of beam cross-sections at different stories
IBND	A	array containing integer input data bounds
ICC	A	array of indices indicating the type of column cross-sections in a column line in a frame group
IDW	S	flag to indicate whether or not an input data file is to be built
IDR	S	flag to indicate whether the data is to be supplied from a disc file or terminal
IFD	S	index to define whether a frame is parallel to global x-axis or y-axis
IFIX	A	array to store the fixed point numbers read by subroutine REED
II	S	frame number; also acts as a counter
ILFT	A	array of symbol to identify the left side of a frame
INDL	S	index to check whether or not the gravity loads in a frame are the same as those in the preceding frame
INDX	A	numeric indicator array to indicate whether or not a discrete data item read by subroutine REED is a number

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
INST	S	flag to indicate whether or not instruction is required by the program AIDS
INX	S	variable indicating the number of characters to be processed by subroutine REED from one line of data input
IOUT	S	index for special output i.e. mode shapes, frequencies and story shears for seismic loading
IR	S	unit number from which input data is read
ISTB	S	flag to indicate whether or not finite joint sizes are to be considered in the analysis
ITOT	S	maximum number of discrete data items expected to be read at each subsequent call to subroutine REED
IW	S	unit number to which output is directed
IW1	S	unit number on which general data and seismic load data is to be written
IW2	S	unit number on which the frame properties data is to be written
IW3	S	unit number on which the gravity and lateral load data are to be written
IWF	S	unit number on which the user's input data is to be copied
IYD	S	flag to indicate whether or not all beams in a frame group have the same modulus of elasticity
IYDN	S	flag to indicate whether or not dynamic analysis is to be carried out
IYS	S	flag to indicate whether or not shear deformations are to be considered

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
JAX	A	array of unprocessed alphanumeric data read by subroutine REED
JEND	A	check array to indicate whether or not the input data is to be checked for validity
IM	S	number of modes to be extracted from one origin
M	S	group number to which frame belongs
NBAY	A	number of bays in a tower of a frame group
NBS	A	number of bays - setback at left between two adjacent towers of a frame group
NCL	A	number of concentrated loads on a beam in a frame
NDYN	S	index to define whether or not seismic load is to be considered in the analysis
NFG		total number of frame groups in the building
NFIG	A	number of frames in a frame group
NFT	S	total number of frames in the building
NGR	S	number of different gravity load cases
NM	S	number of modes to be considered
NNS	S	total number of stories in the building
NOPT	S	story translation code, indicating whether the story displacements for lateral loads are only translational (x- or y-translations) or coupled with rotations (for unsymmetric building)
NP	S	number of period coordinates defining the design spectrum

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
NSS	S	number of stories in a frame group
NST	A	number of stories in a tower of a frame group
NSTB	S	index to check whether or not finite joint sizes are considered
NSYS	S	number of load systems (load combinations) to be considered
NTF	A	number of towers in different frame groups
NWL	S	number of different lateral load cases
SAC	S	acceleration coordinates of the design spectrum
SF	S	scale factor to be applied to the design spectrum
SMULP	A	multipliers for different load cases in a load system
TC	S	period coordinates on the design response spectrum
UDL	A	values of uniformly distributed loads on the beams in a frame
XB	A	x-coordinates of story mass centers from the reference point
XM	A	story masses of the building
XMI	A	story mass moments of inertia of the building
XMU	S	ductility factor
YB	A	Y-coordinates of story mass centers from the reference points
IFR,IMB,INC, IND,INF,IS, NB,NBS1,NBS2, NC,NCASE,NFALT, NI,NIL,NJ,NNF,NNG, NSB,NSI,NT	S	indices, counters, and temporary integer locations

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
IIO, IIT, IOF, IOO, ITF, ITI, IYF	S	alphanumeric variables, temporary
EG, EY	S	temporary real locations
IB, IBL, IC, IL, IN, IO, IT, IY	S	literal constants

A2.4 EXAMPLE OF DATA INPUT

For the building shown in Fig. A.5, the data is prepared in a manner as discussed in Section A2.3. Any consistent sets of units may be used. In this example, a "in-kip-sec" units are used. The conversational data input is shown below. The output from program AIDS as stored in four disc files is shown in Section A2.5.

```

*****
*
* -----
* ***** A I D S *****
* -----
*
* READS DATA BY CONVERSATIONAL
* MODE WITH RAPID INPUT OPTION
* AND WRITES IN THREE FILES
* FOR THE THREE DIMENSIONAL
* BUILDING ANALYSIS PROGRAM.
*
*****

```

DO YOU WISH TO CREATE A NEW DATA FILE CONTAINING
A COPY OF THE PRESENT INPUT DATA FOR FUTURE USE ? Y/N
Y

THE INPUT DATA WILL BE SAVED ON
A FILE NAMED - DUPLI
DO YOU WISH TO INPUT DATA BY MEANS OF THE
TERMINAL OR BY MEANS OF AN EXSITING DATA FILE ? T/F
T

DO YOU WANT INSTRUCTION FOR THIS
RAPID DATA INPUT SYSTEM ? Y/N
Y

RAPID DATA INPUT EXAMPLE :

BEAM CROSS SECTION PROPERTIES :

.... ..
STORY LEVEL 4
ARE ALL THE BEAMS IN THIS STORY LEVEL OF
CONCRETE RECTANGULAR OR OTHER ? C/O
7C (CR)

BAY NO. 1 :
PUT BEAM WIDTH AND THEN BEAM DEPTH, SEPARATED BY A BLANK
715 20 TO 6 (CR)

THIS DATA SUPPLIED BY THE USER WILL SET A
SIZE OF 15 BY 20 TO BEAM 1 TO BEAM 6 AT STORY 4
BAY NO. 7 :
715 24 (CR)

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 4
NOW, ARE THE BEAMS IN THE NEXT STORY EXACTLY LIKE
THOSE IN ANY PRECEDING STORY ? IF NOT--PUT NO
IF YES--LIKE WHICH ?
7LTKE 2 TO 8 (CR)

WITH THIS DATA THE PROGRAM WILL SET ALL BEAM SIZES IN STORIES 5 TO 8, EXACTLY SAME AS THE CORRESPONDING ONES IN STORY 2

A SIMILAR RAPID DATA INPUT PROCEDURE CAN BE USED FOR STORY HEIGHTS, BAY LENGTHS, COLUMN CROSS SECTIONS, LOAD DATA ETC.

DURING THE DATA INPUT FOR GRAVITY LOADS, WHEN DATA FOR ONE FRAME (SAY FRAME NO. 3) IS COMPLETE, THE COMPUTER PROMPTS WITH —
GRAVITY LOAD DATA IS COMPLETE UPTO FRAME NO. 3
NOW, ARE THESE DATA IN THE NEXT FRAME EXACTLY SAME (FOR ALL GR. LOAD CASES) AS THOSE IN FRAME NO. 3? Y/N
?Y TO 6 (CR)

WITH THIS THE COMPUTER WILL PUT GRAVITY LOADS IN FRAMES 4 TO 6 SAME AS THOSE IN FRAME 3

ALL OTHER DIALOGUES ARE SELF EXPLANATORY

PUT A CASE TITLE FOR THIS PROJECT

? EXAMPLE BUILDING

DO YOU WANT FINITE JOINT SIZES TO BE
TO BE CONSIDERED FOR THIS ANALYSIS ? Y/N
?Y

DO YOU WANT SHEAR DEFORMATIONS TO BE
TO BE CONSIDERED FOR THIS ANALYSIS ? Y/N
?Y

** THE UNITS TO BE USED IN THIS PROGRAM MAY BE
OF ANY TYPE, BUT THEY MUST BE CONSISTENT **

STRUCTURE HISTORY :

TOTAL NO. OF FRAMES IN THE THE BUILDING
? 5

NO. OF FRAME GROUPS IN THE BUILDING
? 3

TOTAL NO. OF STORIES IN THE BUILDING
? 3

NO. OF BAYS IN THE WIDEST FRAME OF THE BUILDING?
? 2

STORY HEIGHTS, C/C INTERSECTING FLOOR BEAMS

BETWEEN STORY LEVELS 0 AND 1
? 120 TO 2

BETWEEN STORY LEVELS 2 AND 3
? 100

FRAME INFORMATION :

FOR FRAME NO. 1 :

FRAME GROUP NO.

?1

PARALLEL TO GLOBAL X-DIRECTION OR Y-DIRECTION ? X/Y

?XPERPENDICULAR DISTANCE FROM
THE REFERENCE POINT TO THE FRAME?0SYMBOL(MAX. 10 CHARACTERS) TO IDENTIFY
LEFT SIDE OF THE FRAME?WEST-S

FOR FRAME NO. 2 :

FRAME GROUP NO.

?1

PARALLEL TO GLOBAL X-DIRECTION OR Y-DIRECTION ? X/Y

?XPERPENDICULAR DISTANCE FROM
THE REFERENCE POINT TO THE FRAME?240SYMBOL(MAX. 10 CHARACTERS) TO IDENTIFY
LEFT SIDE OF THE FRAME?WEST-N

FOR FRAME NO. 3 :

FRAME GROUP NO.

?2

PARALLEL TO GLOBAL X-DIRECTION OR Y-DIRECTION ? X/Y

?YPERPENDICULAR DISTANCE FROM
THE REFERENCE POINT TO THE FRAME?240SYMBOL(MAX. 10 CHARACTERS) TO IDENTIFY
LEFT SIDE OF THE FRAME?SOUTH-MID

FOR FRAME NO. 4 :

FRAME GROUP NO.

?2

PARALLEL TO GLOBAL X-DIRECTION OR Y-DIRECTION ? X/Y

?Y

PERPENDICULAR DISTANCE FROM
THE REFERENCE POINT TO THE FRAME
?360

SYMBOL(MAX. 10 CHARACTERS) TO IDENTIFY
LEFT SIDE OF THE FRAME
?SOUTH-E

FOR FRAME NO. 5 :
FRAME GROUP NO.
?3

PARALLEL TO GLOBAL X-DIRECTION OR Y-DIRECTION ? X/Y
?Y

PERPENDICULAR DISTANCE FROM
THE REFERENCE POINT TO THE FRAME
?0

SYMBOL(MAX. 10 CHARACTERS) TO IDENTIFY
LEFT SIDE OF THE FRAME
?SOUTH-W

FRAME GROUP NO. 1

HOW MANY TOWER DOES THIS FRAME HAVE
INCLUDING THE BASE TOWER
?2

IN TOWER 1 (BASE TOWER), NO. OF STORIES
?2

NO. OF BAYS
?2

LENGTHS OF BAYS, C/C OF COLUMNS
STARTING FROM LEFT :

BAY NO. 1 :
?240

BAY NO. 2 :
?120

IN TOWER 2, NO. OF STORIES
?1

NO. OF BAYS
?1

NO. OF BAY-SLIPS BETWEEN
TOWERS 1 AND 2 AT LEFT
?1

BEAM CROSS SECTION PROPERTIES :

THIS SET OF DATA IS TO BE INPUT BEAM BY BEAM
IN EACH FLOOR AND THIS WAY FLOOR BY FLOOR
STORY LEVEL 1 :

ARE ALL THE BEAMS IN THIS STORY LEVEL OF
CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O

?C

BAY NO. 1 :

PUT BEAM WIDTH AND THEN BEAM DEPTH, SEPARATED BY A BLANK

?10 20 TO 2

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 1

NOW, ARE THE BEAMS IN THE NEXT STORY (AND MAX. UPTO STORY 2)

EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?

IF NOT---PUT NO, IF YES---LIKE WHICH ?

EXAMPLE : LIKE M TO N, THIS MEANS THE BEAMS IN
THE NEXT STORY AND TO THE NTH STORY, EXACTLY
LIKE THOSE IN THE MTH STORY.

?LIKE 1

STORY LEVEL 3 :

ARE ALL THE BEAMS IN THIS STORY LEVEL OF

CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O

?C

BAY NO. 2 :

PUT BEAM WIDTH AND THEN BEAM DEPTH, SEPARATED BY A BLANK

?10 15

BEAM MOD. OF ELASTICITY AND MOD. OF RIGIDITY :

ARE THE MODULI OF ELASTICITY SAME FOR ALL THE BEAMS
IN THIS FRAME GROUP ? Y/N

?Y

PUT THE MOD. OF ELASTICITY AND THEN

THE MOD. OF RIGIDITY, SEPARATED BY A BLANK

?3000 1200

SHEAR COEFFICIENTS FOR BEAMS :

STORY LEVEL 1 :

FOR BEAM AT BAY NO. 1

? .833 TO 2

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 1

NOW, ARE THE SHEAR COEFFICIENTS FOR THE NEXT FLOOR BEAMS

(AND MAX. UPTO STORY LEVEL 2) EXACTLY LIKE THOSE

IN ANY PRECEDING STORY? IF NOT---PUT NO, IF YES---LIKE WHICH?

?LIKE 1

STORY LEVEL 3 :
 FOR BEAM AT BAY NO. 2
?833

COLUMN CROSS SECTION PROPERTIES :

THIS SET OF DATA IS TO BE PUT VERTICALLY COLUMN BY COLUMN IN
 EACH COL.LINE AND THIS WAY COL.LINE BY COL.LINE, FROM LEFT
 COLUMN LINE 1 :

ARE ALL THE COLUMNS IN THIS COLUMN LINE OF
 CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O
?C

STORY NO. 1 :
 PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN,
 THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK
?12 16

STORY NO. 2 :
 PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN
 THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK
?12 14

COLUMN LINE 2 :
 ARE ALL THE COLUMNS IN THIS COLUMN LINE OF
 CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O
?C

STORY NO. 1 :
 PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN
 THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK
?16 18

STORY NO. 2 :
 PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN
 THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK
?12 16

STORY NO. 3 :
 PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN
 THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK
?12 12

THIS SET OF DATA IS COMPLETE UPTO COL.LINE 2
 NOW, ARE THE COLUMNS IN THE NEXT COL.LINE EXACTLY
 LIKE THOSE IN ANY PRECEDING COLUMN LINE ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?
?LIKE 2

COLUMN MOD. OF ELASTICITY AND MOD. OF ROGIDITY :
 ARE THE MODULII OF ELASTICITY SAME FOR ALL THE COLUMNS
 IN THIS FRAME GROUP ? Y/N
?Y

PUT THE MOD. OF ELASTICITY AND THEN
THE MOD. OF RIGIDITY, SEPARATED BY A BLANK
73000 1150

SHEAR COEFFICIENTS FOR COLUMNS :

COLUMN LINE 1 :

FOR COLUMN AT STORY NO. 1
7.833 TO 2

COLUMN LINE 2 :

FOR COLUMN AT STORY NO. 1
7.833 TO 3

THIS SET OF DATA IS COMPLETE UPTO COL.LINE 2 ,
NOW, ARE THE SHEAR COEFFICIENTS OF THE COLUMNS IN THE
NEXT COL.LINE EXACTLY LIKE THOSE IN ANY PRECEDING COL.LINE?
IF NOT---PUT NO, IF YES---LIKE WHICH ?
7L 2

FRAME GROUP NO. 2

HOW MANY TOWER DOES THIS FRAME HAVE
INCLUDING THE BASE TOWER
71

IN TOWER 1 (BASE TOWER), NO. OF STORIES
73

NO. OF BAYS
71

LENGTHS OF BAYS, C/C OF COLUMNS
STARTING FROM LEFT :

BAY NO. 1 :
7240

BEAM CROSS SECTION PROPERTIES :
THIS SET OF DATA IS TO BE INPUT BEAM BY BEAM
IN EACH FLOOR AND THIS WAY FLOOR BY FLOOR
STORY LEVEL 1 :
ARE ALL THE BEAMS IN THIS STORY LEVEL OF
CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O
7C

BAY NO. 1 :
 PUT BEAM WIDTH AND THEN BEAM DEPTH, SEPARATED BY A BLANK
?10 18

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 1
 NOW, ARE THE BEAMS IN THE NEXT STORY (AND MAX. UPTO STORY 3)
 EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?
 EXAMPLE : LIKE M TO N , THIS MEANS THE BEAMS IN
 THE NEXT STORY AND TO THE NTH STORY, EXACTLY
 LIKE THOSE IN THE MTH STORY.

?L 1

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 2
 NOW, ARE THE BEAMS IN THE NEXT STORY (AND MAX. UPTO STORY 3)
 EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?
 EXAMPLE : LIKE M TO N , THIS MEANS THE BEAMS IN
 THE NEXT STORY AND TO THE NTH STORY, EXACTLY
 LIKE THOSE IN THE MTH STORY.

?N

STORY LEVEL 3 :
 ARE ALL THE BEAMS IN THIS STORY LEVEL OF
 CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O

?C

BAY NO. 1 :
 PUT BEAM WIDTH AND THEN BEAM DEPTH, SEPARATED BY A BLANK
?10 15

BEAM MOD. OF ELASTICITY AND MOD. OF RIGIDITY :

ARE THE MODULI OF ELASTICITY SAME FOR ALL THE BEAMS
 IN THIS FRAME GROUP ? Y/N

?Y

PUT THE MOD. OF ELASTICITY AND THEN
 THE MOD. OF RIGIDITY, SEPARATED BY A BLANK
?3000 1200

SHEAR COEFFICIENTS FOR BEAMS :

STORY LEVEL 1 :
 FOR BEAM AT BAY NO. 1
? .833

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 1
 NOW, ARE THE SHEAR COEFFICIENTS FOR THE NEXT FLOOR BEAMS
 (AND MAX. UPTO STORY LEVEL 3) EXACTLY LIKE THOSE
 IN ANY PRECEDING STORY? IF NOT---PUT NO, IF YES---LIKE WHICH?
?L 1 TO 3

COLUMN CROSS SECTION PROPERTIES :

THIS SET OF DATA IS TO BE PUT VERTICALLY COLUMN BY COLUMN IN EACH COL.LINE AND THIS WAY COL.LINE BY COL.LINE, FROM LEFT

COLUMN LINE 1 :

ARE ALL THE COLUMNS IN THIS COLUMN LINE OF CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O

?C

STORY NO. 1 :

PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK

?18 16

STORY NO. 2 :

PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK

?16 12

STORY NO. 3 :

PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK

?12 12

THIS SET OF DATA IS COMPLETE UPTO COL.LINE 1

NOW, ARE THE COLUMNS IN THE NEXT COL.LINE EXACTLY LIKE THOSE IN ANY PRECEDING COLUMN LINE ?

IF NOT---PUT NO, IF YES---LIKE WHICH ?

?L 1

COLUMN MOD. OF ELASTICITY AND MOD. OF RIGIDITY :

ARE THE MODULII OF ELASTICITY SAME FOR ALL THE COLUMNS IN THIS FRAME GROUP ? Y/N

?Y

PUT THE MOD. OF ELASTICITY AND THEN

THE MOD. OF RIGIDITY, SEPARATED BY A BLANK

?3000 1150

SHEAR COEFFICIENTS FOR COLUMNS :

COLUMN LINE 1 :

FOR COLUMN AT STORY NO. 1

? .833 TO 3

THIS SET OF DATA IS COMPLETE UPTO COL.LINE 1

NOW, ARE THE SHEAR COEFFICIENTS OF THE COLUMNS IN THE NEXT COL.LINE EXACTLY LIKE THOSE IN ANY PRECEDING COL.LINE?

IF NOT---PUT NO, IF YES---LIKE WHICH ?

?L 1

 FRAME GROUP NO. 3

HOW MANY TOWER DOES THIS FRAME HAVE
 INCLUDING THE BASE TOWER

?1

IN TOWER 1 (BASE TOWER), NO. OF STORIES

?2

NO. OF BAYS

?1

LENGTHS OF BAYS, C/C OF COLUMNS
 STARTING FROM LEFT :

BAY NO. 1 :

?240

BEAM CROSS SECTION PROPERTIES :

THIS SET OF DATA IS TO BE INPUT BEAM BY BEAM
 IN EACH FLOOR AND THIS WAY FLOOR BY FLOOR
 STORY LEVEL 1 :

ARE ALL THE BEAMS IN THIS STORY LEVEL OF
 CONCRETERECTANGULAR SECTIONS OR OTHER ? C/O

?C

BAY NO. 1 :

PUT BEAM WIDTH AND THEN BEAM DEPTH, SEPARATED BY A BLANK

?10 15

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 1
 NOW, ARE THE BEAMS IN THE NEXT STORY (AND MAX. UPTO STORY 2)
 EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?

EXAMPLE : LIKE M TO N, THIS MEANS THE BEAMS IN
 THE NEXT STORY AND TO THE NTH STORY, EXACTLY
 LIKE THOSE IN THE MTH STORY.

?L 1

BEAM MOD. OF ELASTICITY AND MOD. OF RIGIDITY :

ARE THE MODULII OF ELASTICITY SAME FOR ALL THE BEAMS
 IN THIS FRAME GROUP ? Y/N

?Y

PUT THE MOD. OF ELASTICITY AND THEN
 THE MOD. OF RIGIDITY, SEPARATED BY A BLANK

?3000 1200

SHEAR COEFFICIENTS FOR BEAMS :

STORY LEVEL 1 :
FOR BEAM AT BAY NO. 1

?L 833

THIS SET OF DATA IS COMPLETE UPTO STORY LEVEL 1
NOW, ARE THE SHEAR COEFFICIENTS FOR THE NEXT FLOOR BEAMS
(AND MAX. UPTO STORY LEVEL 2) EXACTLY LIKE THOSE
IN ANY PRECEDING STORY? IF NOT--PUT NO, IF YES--LIKE WHICH?
?L 1

COLUMN CROSS SECTION PROPERTIES :

THIS SET OF DATA IS TO BE PUT VERTICALLY COLUMN BY COLUMN IN
EACH COL.LINE AND THIS WAY COL.LINE BY COL.LINE, FROM LEFT

COLUMN LINE 1 :

ARE ALL THE COLUMNS IN THIS COLUMN LINE OF
CONCRETE RECTANGULAR SECTIONS OR OTHER ? C/O

?C

STORY NO. 1 :

PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN
THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK

?16 12

STORY NO. 2 :

PUT THE COLUMN WIDTH ACCROSS THE PLANE OF FRAME AND THEN
THE WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK

?14 12

THIS SET OF DATA IS COMPLETE UPTO COL.LINE 1
NOW, ARE THE COLUMNS IN THE NEXT COL.LINE EXACTLY
LIKE THOSE IN ANY PRECEDING COLUMN LINE ?
IF NOT--PUT NO, IF YES--LIKE WHICH ?

?L 1

COLUMN MOD. OF ELASTICITY AND MOD. OF RIGIDITY :

ARE THE MODULI OF ELASTICITY SAME FOR ALL THE COLUMNS
IN THIS FRAME GROUP ? Y/N

?Y

PUT THE MOD. OF ELASTICITY AND THEN
THE MOD. OF RIGIDITY, SEPARATED BY A BLANK

?3000 1150

SHEAR COEFFICIENTS FOR COLUMNS :

COLUMN LINE 1 :

FOR COLUMN AT STORY NO. 1

?.833 TO 2

THIS SET OF DATA IS COMPLETE UPTO COL.LINE 1 ,
 NOW, ARE THE SHEAR COEFFICIENTS OF THE COLUMNS IN THE
 NEXT COL.LINE EXACTLY LIKE THOSE IN ANY PRECEDING COL.LINE?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?

?L 1

LOAD CASE INFORMATION DATA :

HOW MANY GRAVITY LOAD CASES (MAX. 4)

YOU WANT TO INCLUDE IN YOUR ANALYSIS

?1

HOW MANY LATERAL LOAD CASES (MAX. 3)

YOU WANT TO INCLUDE IN YOUR ANALYSIS

?1

DO YOU WANT TO INCLUDE EARTHQUAKE LOADS
 IN ANALYSIS ? Y/N

?Y

HOW MANY LOAD SYSTEMS (LOAD COMBINATIONS)

DO YOU WANT FOR THE ABOVE LOAD CASES

?4

DO YOU WANT OUTPUT FOR THE STRUCTURAL
 MODE SHAPES AND FREQUENCIES ? Y/N

?Y

DO YOU WANT OUTPUT FOR STORY SHEARS AND
 OTHER FORCES AND MOMENTS FOR DYNAMIC ANALYSIS? Y/N

?Y

FOR THREE DIMENSIONAL ANALYSIS :

NO. OF DEGREES OF FREEDOM PER STORY ? 1/2/3

1 FOR X-TRANSLATION ONLY

2 FOR Y-TRANSLATION ONLY

3 FOR X, Y-TRANSLATION AND ROTATION

?3

LOAD SYSTEMS INFORMATION DATA :

LOAD SYSTEMS MUST BE ARRANGED (NUMBERED) SEQUENTIALLY
ACCORDING TO THE FOLLOWING PREFERENCE LIST.....

LOAD SYSTEMS CONTAINING --

1. GRAVITY LOAD CASES ONLY.
2. GRAVITY LOAD, AND/OR LATERAL LOAD CASES
3. GRAVITY LOAD, AND/OR LATERAL LOAD, AND/OR SEISMIC LOADS

FOR LOAD SYSTEM 1 :

PUT MULT. FACTOR FOR THE GRAVITY LOAD CASE
?1

PUT MULT. FACTOR FOR THE LATERAL LOAD CASE
?0

PUT THE MULT. FACTOR FOR THE EARTHQUAKE LOAD (IF ANY)
?0

FOR LOAD SYSTEM 2 :

PUT MULT. FACTOR FOR THE GRAVITY LOAD CASE
?0

PUT MULT. FACTOR FOR THE LATERAL LOAD CASE
?1

PUT THE MULT. FACTOR FOR THE EARTHQUAKE LOAD (IF ANY)
?0

FOR LOAD SYSTEM 3 :

PUT MULT. FACTOR FOR THE GRAVITY LOAD CASE
?0

PUT MULT. FACTOR FOR THE LATERAL LOAD CASE
?0

PUT THE MULT. FACTOR FOR THE EARTHQUAKE LOAD (IF ANY)
?1

FOR LOAD SYSTEM 4 :

PUT MULT. FACTOR FOR THE GRAVITY LOAD CASE
?1

PUT MULT. FACTOR FOR THE LATERAL LOAD CASE
? .75

PUT THE MULT. FACTOR FOR THE EARTHQUAKE LOAD (IF ANY)
? .8

DATA FOR GRAVITY LOAD CASES :

 FOR FRAME NO. 1

GRAVITY LOAD CASE 1 :

DATA FOR UDL GRAVITY LOADS :

PUT THE DISTRIBUTED LOAD VALUE
 PER LINEAR UNIT IN EACH BEAM

STORY LEVEL 1 :
 FOR BEAM AT BAY NO. 1

? .07 TO 2

UDL DATA IS COMPLETE UPTO STORY LEVEL 1 :
 NOW, ARE THE UDL IN THE NEXT STORY BEAMS(AND MAX. UPTO
 STORY 2) EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?

? L 1

STORY LEVEL 3 :

FOR BEAM AT BAY NO. 2

? .03

DATA FOR CONCENTRATED GRAVITY LOADS :
 (MAX. 4 LOADS PER BEAM)

STORY LEVEL 1 :
 NO. OF CONC. LOADS ON THE BEAM AT BAY NO. 1

? 0

NO. OF CONC. LOADS ON THE BEAM AT BAY NO. 2

? 1

LOAD NO. 1 :
 PUT THE LOAD VALUE AND THEN ITS DISTANCE FROM
 THE LEFT COLUMN CENTRE, SEPARATED BY A BLANK

? 4 60

CONC. LOAD DATA IS COMPLETE UPTO STORY LEVEL 1
 NOW, ARE THE LOADS AND THEIR POSITIONS IN THE NEXT STORY
 (AND MAX. UPTO STORY LEVEL 2) EXACTLY LIKE THOSE
 IN ANY PRECEDING STORY ? IF NOT---PUT NO, IF YES---LIKE WHICH?

? L 1

STORY LEVEL 3 :

NO. OF CONC. LOADS ON THE BEAM AT BAY NO. 2

? 0

GRAVITY LOAD DATA IS COMPLETE UPTO FRAME NO. 1
 NOW, ARE THESE DATA IN THE NEXT FRAME (AND UPTO FRAME NO. 2)
 EXACTLY SAME (FOR ALL GR. LOAD CASES) AS THOSE IN
 FRAME NO. 1 ? Y/N

?Y

 FOR FRAME NO. 3

GRAVITY LOAD CASE 1 :

DATA FOR UDL GRAVITY LOADS :
 PUT THE DISTRIBUTED LOAD VALUE
 PER LINEAR UNIT IN EACH BEAM
 STORY LEVEL 1 :

FOR BEAM AT BAY NO. 1

? .12

UDL DATA IS COMPLETE UPTO STORY LEVEL 1 :
 NOW, ARE THE UDL IN THE NEXT STORY BEAMS (AND MAX. UPTO
 STORY 3) EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?

?L 1

UDL DATA IS COMPLETE UPTO STORY LEVEL 2 :
 NOW, ARE THE UDL IN THE NEXT STORY BEAMS (AND MAX. UPTO
 STORY 3) EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?

?N

STORY LEVEL 3 :

FOR BEAM AT BAY NO. 1

? .03

DATA FOR CONCENTRATED GRAVITY LOADS :
 (MAX. 4 LOADS PER BEAM)
 STORY LEVEL 1 :

NO. OF CONC. LOADS ON THE BEAM AT BAY NO. 1

?1

LOAD NO. 1 :

PUT THE LOAD VALUE AND THEN ITS DISTANCE FROM
 THE LEFT COLUMN CENTRE, SEPARATED BY A BLANK

?4 120

CONC. LOAD DATA IS COMPLETE UPTO STORY LEVEL 1
 NOW, ARE THE LOADS AND THEIR POSITIONS IN THE NEXT STORY
 (AND MAX. UPTO STORY LEVEL 3) EXACTLY LIKE THOSE
 IN ANY PRECEDING STORY ? IF NOT---PUT NO, IF YES---LIKE WHICH?

?L 1

CONC. LOAD DATA IS COMPLETE UPTO STORY LEVEL 2
 NOW, ARE THE LOADS AND THEIR POSITIONS IN THE NEXT STORY
 (AND MAX. UPTO STORY LEVEL 3) EXACTLY LIKE THOSE
 IN ANY PRECEDING STORY ? IF NOT---PUT NO, IF YES---LIKE WHICH?
?N

STORY LEVEL 3 :
 NO. OF CONC. LOADS ON THE BEAM AT BAY NO. 1
?0

GRAVITY LOAD DATA IS COMPLETE UPTO FRAME NO. 3
 NOW, ARE THESE DATA IN THE NEXT FRAME(AND UPTO FRAME NO 4)
 EXACTLY SAME (FOR ALL GR. LOAD CASES) AS THOSE IN
 FRAME NO. 3 ? Y/N
?Y

 FOR FRAME NO. 5

GRAVITY LOAD CASE 1 :
 DATA FOR UDL GRAVITY LOADS :

PUT THE DISTRIBUTED LOAD VALUE
 PER LINEAR UNIT IN EACH BEAM
 STORY LEVEL 1 :
 FOR BEAM AT BAY NO. 1

? .07

UDL DATA IS COMPLETE UPTO STORY LEVEL 1 :
 NOW, ARE THE UDL IN THE NEXT STORY BEAMS(AND MAX. UPTO
 STORY 2) EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?
 IF NOT---PUT NO, IF YES---LIKE WHICH ?
?L 1

DATA FOR CONCENTRATED GRAVITY LOADS :
 (MAX. LOADS PER BEAM)
 STORY LEVEL 1 :

NO. OF CONC. LOADS ON THE BEAM AT BAY NO. 1
?0

CONC. LOAD DATA IS COMPLETE UPTO STORY LEVEL 1.
 NOW, ARE THE LOADS AND THEIR POSITIONS IN THE NEXT STORY
 (AND MAX. UPTO STORY LEVEL 2) EXACTLY LIKE THOSE
 IN ANY PRECEDING STORY ? IF NOT---PUT NO, IF YES---LIKE WHICH?
?LIKE 1

DATA FOR LATERAL LOAD CASES :

LATERAL LOAD CASE 1 :

STORY NO. 1 :

PUT HOR. LOAD COMPONENT IN GLOBAL X-DIRECTION AND THEN
COMPONENT IN THE Y-DIRECTION, SEPARATED BY A BLANK

24 6

PUT THE X-DISTANCE AND THEN THE Y-DISTANCE OF THE
POINT OF ACTION FROM GLOBAL ORIGIN

2180 120

LOAD AND DISTANCE DATA IS COMPLETE UPTO STORY 1
NOW, ARE THESE DATA IN THE NEXT STORY EXACTLY LIKE THOSE
IN ANY PRECEDING STORY? IF NOT—PUT NO, IF YES—LIKE WHICH?
2N

STORY NO. 2 :

PUT HOR. LOAD COMPONENT IN GLOBAL X-DIRECTION AND THEN
COMPONENT IN THE Y-DIRECTION, SEPARATED BY A BLANK

24 4

PUT THE X-DISTANCE AND THEN THE Y-DISTANCE OF THE
POINT OF ACTION FROM GLOBAL ORIGIN

2204 120

LOAD AND DISTANCE DATA IS COMPLETE UPTO STORY 2
NOW, ARE THESE DATA IN THE NEXT STORY EXACTLY LIKE THOSE
IN ANY PRECEDING STORY? IF NOT—PUT NO, IF YES—LIKE WHICH?
2N

STORY NO. 3 :

PUT HOR. LOAD COMPONENT IN GLOBAL X-DIRECTION AND THEN
COMPONENT IN THE Y-DIRECTION, SEPARATED BY A BLANK

21.6 1

PUT THE X-DISTANCE AND THEN THE Y-DISTANCE OF THE
POINT OF ACTION FROM GLOBAL ORIGIN

2300 120

DYNAMIC ANALYSIS :

NO. OF MODES TO BE CONSIDERED

29

NO. OF MODES TO BE EXTRACTED FROM ONE ORIGIN

25

DUCTILITY FACTOR

24

SCALE FACTOR FOR DESIGN SPECTRUM

2.08

ANGLE OF INCIDENCE OF EARTHQUAKE
WITH THE GLOBAL X-AXIS, DEGREES

730

NO. OF PERIOD COORDINATES FOR RESPONSE SPECTRUM

74

FOR PERIOD COORD. NO. 1 :
PUT THE PERIOD VALUE AND THEN THE CORRESPONDING
SPECTRAL ACCELERATION, SEPARATED BY A BLANK

7.1 3

FOR PERIOD COORD. NO. 2 :
PUT THE PERIOD VALUE AND THEN THE CORRESPONDING
SPECTRAL ACCELERATION, SEPARATED BY A BLANK

7.42 3

FOR PERIOD COORD. NO. 3 :
PUT THE PERIOD VALUE AND THEN THE CORRESPONDING
SPECTRAL ACCELERATION, SEPARATED BY A BLANK

75 .26

FOR PERIOD COORD. NO. 4 :
PUT THE PERIOD VALUE AND THEN THE CORRESPONDING
SPECTRAL ACCELERATION, SEPARATED BY A BLANK

715 .029

PERIOD WHICH SEPARATES THE ACCL. BOUND FROM
THE VEL. BOUND ON THE RESPONSE SPECTRUM

7.5

ACCELERATION DUE TO GRAVITY

7386.4

• MASS DATA :

AT STORY LEVEL 1 , TRANSLATIONAL FLOOR MASS

7.2

ROTATIONAL MASS MOM. OF INERTIA

72900.

PUT X-DISTANCE AND THEN Y-DISTANCE OF THE CENTRE OF
MASS FROM THE REFERENCE POINT, SEPARATED BY A BLANK

7180 120

THE MASS AND DISTANCE DATA IS COMPLETE UPTO STORY 1
NOW, ARE THESE DATA IN THE NEXT STORY EXACTLY LIKE THOSE
IN ANY PRECEDING STORY ? IF NOT—PUT NO, IF YES—LIKE WHICH?

7L 1

THE MASS AND DISTANCE DATA IS COMPLETE UPTO STORY 2
NOW, ARE THESE DATA IN THE NEXT STORY EXACTLY LIKE THOSE
IN ANY PRECEDING STORY ? IF NOT—PUT NO, IF YES—LIKE WHICH?

7N

AT STORY LEVEL 3 , TRANSLATIONAL FLOOR MASS
7.06

ROTATIONAL MASS MOM. OF INERTIA
7400

PUT X-DISTANCE AND THEN Y-DISTANCE OF THE CENTRE OF
MASS FROM THE REFERENCE POINT, SEPARATED BY A BLANK
7300 120

** MAX. NO. OF FRAME STIFFNESS ELEMENTS
** REQUIRED TO BE STORED IS, NEL = 120
STOP 0

A2.5 DATA OUTPUT

As mentioned earlier, program AIDS rearranges the user's input and writes in three disc files on units 3, 4 and 5, the names of the files being RID1, RID2 and RID3. File RID1 contains the general structure data and earthquake spectrum loading data. File RID2 contains the frame properties data stored sequentially. File RID3 carries the static load data. Also a copy of the user's input is kept in the file named 'DUPLI' on unit 6. A printout of the data files and the input copy file is given below.

1. General Data File - RID1

EXAMPLE BUILDING

5	3	3	2		
120.000		120.000		100.000	
1	1	X		.00000	WEST-S
2	1	X		240.00000	WEST-N
3	2	Y		240.00000	SOUTH-MID
4	2	Y		360.00000	SOUTH-E
5	3	Y		.00000	SOUTH-W
1	1	1	4	2	
3					
1.000		.000		.000	
.000		1.000		.000	
.000		.000		1.000	
1.000		.750		.800	
9	5				
4					
.100		3.000			
.420		3.000			
5.000		.260			
15.000		.029			
4.000		.080		30.000	.500
.200	2900.000		180.000		120.000
.200	2900.000		180.000		120.000
.060	400.000		300.000		120.000

386.400

3

2. Frame Properties File - RID2

```

2      3      2
2      1
2      1
1
240.000  120.000
1
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .20000E 02
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .20000E 02
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .20000E 02
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .20000E 02
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .15000E 02
C
2
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .16000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .14000E 02
C
3
.30000E 04 .11500E 04 .83300E 00 .16000E 02 .18000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .16000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .12000E 02
C
3
.30000E 04 .11500E 04 .83300E 00 .16000E 02 .18000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .16000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .12000E 02
1      3      2
3
1
240.000
1
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .18000E 02
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .18000E 02
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .15000E 02
C
3
.30000E 04 .11500E 04 .83300E 00 .18000E 02 .16000E 02
.30000E 04 .11500E 04 .83300E 00 .16000E 02 .12000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .12000E 02
C
3
.30000E 04 .11500E 04 .83300E 00 .18000E 02 .16000E 02
.30000E 04 .11500E 04 .83300E 00 .16000E 02 .12000E 02
.30000E 04 .11500E 04 .83300E 00 .12000E 02 .12000E 02
1      2      1
2
1
240.000
1
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .15000E 02
C
.30000E 04 .12000E 04 .83300E 00 .10000E 02 .15000E 02
C
2
.30000E 04 .11500E 04 .83300E 00 .16000E 02 .12000E 02
.30000E 04 .11500E 04 .83300E 00 .14000E 02 .12000E 02
C
2
.30000E 04 .11500E 04 .83300E 00 .16000E 02 .12000E 02
.30000E 04 .11500E 04 .83300E 00 .14000E 02 .12000E 02

```

3. Load Data File - RID3

```

2
0
  .070      .070
  .070      .070
  .030
0
1
  4.000      60.000
0
1
  4.000      60.000
0
1
2
0
  .120
  .120
  .030
1
  4.000      120.000
1
  4.000      120.000
0
1
1
0
  .070
  .070
0
0
  4.00000    6.00000    180.00000    120.00000
  4.00000    4.00000    204.00000    120.00000
  1.60000    1.00000    300.00000    120.00000

```

4. Copy of User's Input, File - Dupli

EXAMPLE BUILDING

Y	- FINITE JOINT SIZES CONSIDERED ?
Y	- SHEAR DEFORMATIONS CONSIDERED ?
5	- NO. OF FRAMES
3	- NO. OF FRAMES GROUPS
3	- NO. OF STORIES
2	- NO. OF BAYS IN THE WIDEST FRAME
120 TO 2	- STORY HT. BETWN. LEVELS 0 AND 1
100	- STORY HT. BETWN. LEVELS 2 AND 3
1	- FOR FRAME 1, FRAME GR. NO.
X	- ORIENTATION OF THE FRAME
0	- DIST. FROM REF. POINT
WEST-S	- LEFT IDENTIFICATION
1	- FOR FRAME 2, FRAME GR. NO.
X	- ORIENTATION OF THE FRAME
240	- DIST. FROM REF. POINT
WEST-N	- LEFT IDENTIFICATION
2	- FOR FRAME 3, FRAME GR. NO.
Y	- ORIENTATION OF THE FRAME
240	- DIST. FROM REF. POINT
SOUTH-MID	- LEFT IDENTIFICATION
2	- FOR FRAME 4, FRAME GR. NO.
Y	- ORIENTATION OF THE FRAME
360	- DIST. FROM REF. POINT
SOUTH-E	- LEFT IDENTIFICATION
3	- FOR FRAME 5, FRAME GR. NO.
Y	- ORIENTATION OF THE FRAME
0	- DIST. FROM REF. POINT
SOUTH-W	- LEFT IDENTIFICATION
2	- NO. OF TOWERS IN FRAME GR. 1
2	- IN TOWER 1, NO. OF STORIES
2	- NO. OF BAYS
240	- BAY LENGTH FOR BAY NO. 1
120	- BAY LENGTH FOR BAY NO. 2
1	- IN TOWER 2, NO. OF STORIES
1	- NO. OF BAYS
1	- BAY-SETBACK AT LEFT BETWN TOWR 1 & 2
BEAM CROSS SECTION PROPERTIES	
C	- SECTION TYPE, STORY 1
10 20 TO 2	- BEAM AT BAY 1
LIKE 1	- COMPLETE UPTO STORY 1, NEXT?
C	- SECTION TYPE, STORY 3
10 15	- BEAM AT BAY 2
Y	- ARE MOD. OF ELAST. SAME FOR ALL BEAMS?
3000 1200	- VALUES OF MOD. OF ELAST & RIGIDITY
SHEAR COEFFICIENTS FOR BEAMS	
.833 TO 2	- BEAM AT BAY 1
LIKE 1	- COMPLETE UPTO STORY 1, NEXT?
.833 -	- BEAM AT BAY 2

COLUMN CROSS SECTION PROPERTIES

C - X-SECTION TYPE AT COL. LINE 1
 12 16 - COLUMN AT STORY 1
 12 14 - COLUMN AT STORY 2
 C - X-SECTION TYPE AT COL. LINE 2
 16 18 - COLUMN AT STORY 1
 12 16 - COLUMN AT STORY 2
 12 12 - COLUMN AT STORY 3
 LIKE 2 - COMPLETE UPTO COL. LINE 2, NEXT?
 Y - ARE MOD. OF ELAST. SAME FOR ALL COLMNS?
 3000 1150 - VALUES OF MOD. OF ELAST & RIGIDITY
 SHEAR COEFFICIENTS FOR COLUMNS
 .833 TO 2 - COLUMN AT STORY 1
 .833 TO 3 - COLUMN AT STORY 1
 L 2 - COMPLETE UPTO COL. LINE 2, NEXT?
 1 - NO. OF TOWERS IN FRAME GR. 2
 3 - IN TOWER 1, NO. OF STORIES
 1 - NO. OF BAYS
 240 - BAY LENGTH FOR BAY NO. 1

BEAM CROSS SECTION PROPERTIES

C - SECTION TYPE, STORY 1
 10 18 - BEAM AT BAY 1
 L 1 - COMPLETE UPTO STORY 1, NEXT?
 N - COMPLETE UPTO STORY 2, NEXT?
 C - SECTION TYPE, STORY 3
 10 15 - BEAM AT BAY 1
 Y - ARE MOD. OF ELAST. SAME FOR ALL BEAMS?
 3000 1200 - VALUES OF MOD. OF ELAST & RIGIDITY
 SHEAR COEFFICIENTS FOR BEAMS
 .833 - BEAM AT BAY 1
 L 1 TO 3 - COMPLETE UPTO STORY 1, NEXT?

COLUMN CROSS SECTION PROPERTIES

C - X-SECTION TYPE AT COL. LINE 1
 18 16 - COLUMN AT STORY 1
 16 12 - COLUMN AT STORY 2
 12 12 - COLUMN AT STORY 3
 L 1 - COMPLETE UPTO COL. LINE 1, NEXT?
 Y - ARE MOD. OF ELAST. SAME FOR ALL COLMNS?
 3000 1150 - VALUES OF MOD. OF ELAST & RIGIDITY
 SHEAR COEFFICIENTS FOR COLUMNS
 .833 TO 3 - COLUMN AT STORY 1
 L 1 - COMPLETE UPTO COL. LINE 1, NEXT?
 1 - NO. OF TOWERS IN FRAME GR. 3
 2 - IN TOWER 1, NO. OF STORIES
 1 - NO. OF BAYS
 240 - BAY LENGTH FOR BAY NO. 1

BEAM CROSS SECTION PROPERTIES

C - SECTION TYPE, STORY 1
 10 15 - BEAM AT BAY 1
 L 1 - COMPLETE UPTO STORY 1, NEXT?
 Y - ARE MOD. OF ELAST. SAME FOR ALL BEAMS?
 3000 1200 - VALUES OF MOD. OF ELAST & RIGIDITY

SHEAR COEFFICIENTS FOR BEAMS

.833

L 1

COLUMN CROSS SECTION PROPERTIES

C

16 12

14 12

L 1

Y

3000 1150

SHEAR COEFFICIENTS FOR COLUMNS

.833 TO 2

L 1

1

1

Y

4

Y

Y

3

1

0

0

0

0

0

0

0

1

1

.75

.8

GRAVITY LOAD CASE DATA

UDL - FRAME 1, GRVT LOAD CASE 1

.07 TO 2

L 1

.03

0

1

4 60

L 1

0

Y

UDL - FRAME 3, GRVT LOAD CASE 1

.12

L 1

N

.03

1

4 120

L 1

N

0

Y

- BEAM AT BAY 1
- COMPLETE UPTO STORY 1, NEXT?

- X-SECTION TYPE AT COL. LINE 1
- COLUMN AT STORY 1
- COLUMN AT STORY 2

- COMPLETE UPTO COL. LINE 1, NEXT?
- ARE MOD. OF ELAST SAME FOR ALL COLUMNS?
- VALUES OF MOD. OF ELAST & RIGIDITY

- COLUMN AT STORY 1
- COMPLETE UPTO COL. LINE 1, NEXT?
- NO. OF GRAVT LOAD CASES
- NO. OF LATERAL LOAD CASES
- SEISMIC LOADS CONSIDERED?
- NO. OF LOAD SYSTEMS CONSIDERED
- OUTPUT FOR MODE SHAPES REQD.?
- OUTPUT FOR DYN. STORY FORCES REQD.?
- CODE FOR STORY DEGREES OF FREEDOM
- IN LOAD SYSTEM 1, FACTRS FOR GRVT LOADS
- IN LOAD SYSTEM 1, FACTRS FOR LATR LOADS
- FACTR FOR SEISMIC LOADS, LOAD SYSTEM 1
- IN LOAD SYSTEM 2, FACTRS FOR GRVT LOADS
- IN LOAD SYSTEM 2, FACTRS FOR LATR LOADS
- FACTR FOR SEISMIC LOADS, LOAD SYSTEM 2
- IN LOAD SYSTEM 3, FACTRS FOR GRVT LOADS
- IN LOAD SYSTEM 3, FACTRS FOR LATR LOADS
- FACTR FOR SEISMIC LOADS, LOAD SYSTEM 3
- IN LOAD SYSTEM 4, FACTRS FOR GRVT LOADS
- IN LOAD SYSTEM 4, FACTRS FOR LATR LOADS
- FACTR FOR SEISMIC LOADS, LOAD SYSTEM 4

- BEAM AT BAY 1
- COMPLETE UPTO STORY 1, NEXT?
- BEAM AT BAY 2
- NO. OF CONCTD. LOADS, BEAM 1, STORY 1
- NO. OF CONCTD. LOADS, BEAM 2, STORY 1
- LOAD - 1 & DIST.
- COMPLETE UPTO STORY 1, NEXT?
- NO. OF CONCTD. LOADS, BEAM 2, STORY 3
- COMPLETE UPTO FRAME 1, NEXT-SAME?

- BEAM AT BAY 1
- COMPLETE UPTO STORY 1, NEXT?
- COMPLETE UPTO STORY 2, NEXT?
- BEAM AT BAY 1
- NO. OF CONCTD. LOADS, BEAM 1, STORY 1
- LOAD - 1 & DIST.
- COMPLETE UPTO STORY 1, NEXT?
- COMPLETE UPTO STORY 2, NEXT?
- NO. OF CONCTD. LOADS, BEAM 1, STORY 3
- COMPLETE UPTO FRAME 3, NEXT-SAME?

UDL - FRAME 5, GRVT LOAD CASE 1

.07

L 1

0

LIKE 1

4 6

180 120

N

4 4

204 120

N

1.6 1

300 120

DYNAMIC ANALYSIS :

9

5

4

.08

30

4

.1 3

.42 3

5 .26

15 .029

.5

386.4

MASS DATA :

.2

2900

180 120

L 1

N

.06

400

300 120

- BEAM AT BAY 1

- COMPLETE UPTO STORY 1, NEXT?

- NO. OF CONCD. LOADS, BEAM 1, STORY 1

- COMPLETE UPTO STORY 1, NEXT?

- LAT. LOAD X-, Y-COMPNTS FOR CASE 1

- X-DIST & Y-DIST

- COMPLETE UPTO STORY 1, NEXT?

- LAT. LOAD X-, Y-COMPNTS FOR CASE 1

- X-DIST & Y-DIST

- COMPLETE UPTO STORY 2, NEXT?

- LAT. LOAD X-, Y-COMPNTS FOR CASE 1

- X-DIST & Y-DIST

- NO. OF MODES CONSIDERED

- MODES EXTRACTED FROM ONE ORIGIN

- DUCTILITY FACTOR

- SCALE FACTOR

- ANGLE OF INCIDENCE

- NO. OF PERIOD COORDS.

- PERIOD & ACCLN. OF COORD. 1

- PERIOD & ACCLN. OF COORD. 2

- PERIOD & ACCLN. OF COORD. 3

- PERIOD & ACCLN. OF COORD. 4

- PERIOD BOUND

- ACCLN. DUE TO GRAVT

- MASS AT STORY 1

- MASS MOM. OF INERTIA

- X-DIST & Y-DIST FROM REF POINT

- COMPLETE UPTO STORY 1, NEXT?

- COMPLETE UPTO STORY 2, NEXT?

- MASS AT STORY 3

- MASS MOM. OF INERTIA

- X-DIST & Y-DIST FROM REF POINT

CHAPTER A3

ANALYSIS PROGRAM ANGELS

A3.1 FUNCTION

The program is designed to perform a complete three-dimensional analysis of multistoried buildings consisting of rigid frame and/or frame-shearwall systems based on the assumptions described in Chapter A1. Any substructure within the building may be a regular or a setback frame or frame-wall. The analysis is based on the assumption that beams are axially rigid but columns can deform axially. Flexural deformations are, of course, considered. In addition, the following options are available:

- i) Shear deformations in the beams and columns may be accounted for provided the modulus of rigidity and the shear coefficients are specified. Such deformations will be disregarded if any of these items is assigned a value of zero. In fact, program AIDS asks the user whether he wishes to consider shear deformations and if he replies with an N, the computer will automatically set the shear coefficients to zero.
- ii) Finite joint sizes may be considered in the analysis.

Program ANGELS can analyze a building for:

- 1) a maximum of 4 gravity load cases (GL1, GL2, GL3, GL4)
- 2) a maximum of 3 lateral load cases (WL1, WL2, WL3)
- 3) one lateral earthquake load case supplied through a design spectrum (EL).

The above-stated load cases can be combined to form any number of load combinations. A load combination is defined by specifying the multiplication factors applicable to its component loads. One such load combination is called a 'load system'. It may be formed, for example, from the expression $aGL2+bWL1+cEL$, where a, b and c are called load case multipliers.

In addition, if required, the program can evaluate three-dimensional mode shapes and frequencies of the building. Also, at the user's option, the modal story forces and displacements of the building (adjusted for ductility) for a specified design elastic spectrum may be calculated and printed out.

A3.2 BASIC OPERATION

An outline flow diagram showing the program operation is shown in Fig. A.6. The basic operations can be divided into the following major steps:

1. The first operation performed by the main program is to read and print out the basic control information and the global data associated with the building. This data is read from the disc file on unit 1.
2. The program next reads the properties of all substructures of the building from the disc file on unit 2. This data is then adapted for use in the calculations of stiffnesses and member forces. The program produces an echo print of the input data

and writes its adapted version on a file on unit 4. The operations in this step are performed by subroutine PROP.

3. The next operation involves the formation of the frame stiffness matrices for each different frame or frame-shearwall in the building using frame property data written on unit 4. The program keeps a record of the diagonal addresses of the stiffness matrices and assigns suitable indices to the sidesway degrees of freedom. The stiffness matrices and the associated records are ~~stored~~ sequentially on a disc file or a tape on unit 5. The operations are performed by subroutine STIFF.
4. The above stiffness matrices are then reduced one by one using Crout reduction method and stored sequentially on a disc file (or a tape) on unit 6 (subroutine REDUCE).
5. The gravity load data is then read from the disc file on unit 3, load vectors are prepared and an echo print of the input load data is produced (subroutine LOAD). Gravity load analysis is then carried out for each frame for all gravity load cases. The operations are performed by subroutine GRAVT which calls subroutine BSUB. The resulting joint displacement vectors are sequentially written on a disc file on unit 8 for use in step 15.
6. The next operation involves the formation of a sidesway stiffness matrix for each individual frame and shearwall in the building by static condensation of the frame stiffness matrix which is retrieved from unit 5. The sidesway stiffness matrices

are then sequentially written on the disc file on unit 2. The operation is performed by subroutine COND which calls subroutine CONDEN, REDUCE and BSUB.

7. Story mass data is then read from the disc file on unit 1 and printed out. This data includes the location of mass centers, at each floor and the story translation and rotational masses. The operations are carried out by subroutine STORY which also assembles the global mass matrix.
8. The program next assembles the global stiffness matrix using frame sidesway stiffness matrix retrieved from unit 2, and the frame location data read in step 1. Subroutine FRAME performs these operations. The global stiffness matrix is written on a file on unit 5 for use in step 10.
9. The program next reads the lateral load data for the building from the file at unit 3, for a maximum of 3 different lateral load cases and prints it out. The structure is then analyzed for these lateral loads. Frame lateral story forces are calculated and written on a file on unit 1 for use in step 13. These operations are accomplished by subroutine HLOAD which calls subroutines RECIP and SFORCE.
10. The mode shapes and frequencies of the building are then evaluated by subroutine EIGVAL, which is called through subroutine SPCTRM.

11. The modal response is next calculated using the specifications of the design elastic spectrum read by the main program in step 1. The seismic design forces and displacements along the global coordinates are worked out and the values obtained are adjusted for ductility before being printed out at the user's option. The operations are performed by the subroutine SPCTRM which calls subroutine RESP.
12. From the adjusted elastic modal displacements worked out in step 11 and the frame location data read earlier, the modal floor story forces are worked out for each frame and stored sequentially on a disc file on unit 1 for subsequent use. The operations described in this step are carried out by subroutine SFORCE.
13. The program next retrieves from unit 1 the frame lateral story forces calculated in step 9, forms joint load vectors and solves for joint displacements (subroutine ROOT) for each lateral load case. The joint displacements are written sequentially on a file on unit 3 for use in step 15.
14. Using the modal frame story forces evaluated in step 12, the program then computes the root-sum-square forces for the members in each frame of the building and writes them down on a tape or disc file on unit 5 for use in step 15. These operations also are performed by subroutine ROOT which calls subroutines BSUB and FORCE.

LEAF 65 OMITTED IN PAGE NUMBERING.

FIG. A.6 Outline flow diagram for operation of the program ANGELS.

FIG. A.7 PROGRAM ORGANIZATION AND FILE MANAGEMENT FOR PROGRAM-ANGELS

15. Finally, the program reads for each frame the displacement vectors for gravity load cases from unit 8 and for lateral load cases from unit 3. The following steps are carried out for each load system. The displacement vectors for gravity and lateral load cases are multiplied by appropriately multiplying factors defined earlier by the user (read in step 1) to form a final displacement vector. The member forces are then determined from this displacement vector. If the load system also includes earthquake loads, the root-sum-square forces are read from unit 5, multiplied by the proper factor and added to (and subtracted from) the member forces calculated above to evaluate the final member forces which are then printed out. The entire procedure described above is repeated for each frame of the building. The above operations are performed by subroutine SYSTEM which calls subroutine FORCE.

The program organization and file management for program ANGELS is shown in Fig. A.7.

A3.3 DEFINITION OF FORTRAN SYMBOLS

The FORTRAN symbols for variables have been listed here main program or subprogram wise. Variables appearing more than once in the main program and one or more subprograms have been listed only once at the time of their first appearance. Also, indices, counters and variables providing temporary locations have been omitted. Type A stands for array and type S for a scalar.

Main Program

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
AG	S	acceleration due to gravity
AI	A	transformation matrix required in formulating global stiffness matrix
AJ	A	transformation matrix required in formulating global stiffness matrix
ALLOW	S	value used in convergence test for extraction of mode shapes
BETA	S	angle of incidence of earthquake with respect to the global x-axis
BMCL	A	dummy variable array
DDA	A	current values of perpendicular distances
DSP	A	story displacement array for a frame
FXBC	A	dummy array
H	A	story heights
IFD	S	flag for the frame orientation (x or y) as read from input data file
IOUT	S	index for special output
IR	S	input unit number
IW	S	output unit number
IM	S	number of modes to be extracted from one origin
M	A	frame group number
MOORE	S	dimension of array in blank common
MI	S	maximum number of iterations to be used in extracting the eigenvalue
MINCR	S	array increment in the blank common

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
NA	A	diagonal addresses for frame stiffness matrix
NAME	A	array containing the project title
NBAY	A	number of frames in different towers of a frame
NBS	A	number of bays setback between adjacent towers of a frame
NDIR	A	orientation of the frame (x or y)
NDYN	S	number of seismic load cases (0 or 1)
NE	S	total number of stiffness elements to be stored for a frame
NEQ	S	number of degrees of freedom in a frame
NFG	S	number of frame groups in the building
NFT	S	number of frames in the building
NGR	S	number of different gravity load cases
NH	S	temporary index
NM	S	number of modes to be considered
NMULP	S	ratio of the number of global degrees of freedom to the total number of stories in the building
NN	S	number of stories in a frame
NNS	S	total number of stories in the building
NOPT	S	story translation code to specify whether pure translation (x or y) or coupled (x,y, θ)
NP	S	number of period coordinates defining the design response spectrum
NQ	S	total global degrees of freedom

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
NR	A	array of indices to identify frame sideways degrees of freedom
NSS	A	number of stories in the tower of a frame
NSYS	S	number of load combinations to be considered
NT	S	number of towers in a frame
NWL	S	number of different lateral load cases
SAC	A	acceleration coordinates in the response spectrum
SF	S	scale factor for design spectrum
SK	S	array in blank common containing the major variables referred to in the program
SLD	A	load vector for a frame
SMULP	A	weight factors for different load cases to be used for the load combinations
TC	A	period coordinates in the response spectrum
TD	S	period defining the boundary between the acceleration bound and velocity bound in the spectrum
XMU	S	ductility factor

Subroutine PROP

AB	A	beams cross-section area
AC	A	columns cross-section area
B	A	bay lengths

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
BA	A	length of rigid end stub at left end of a beam
BB	A	length of rigid end stub at right end of a beam
BE	A	moduli of elasticity of beams
BEF	A	$\gamma = 2EI/L(1+2\alpha)$ for beams
BG	A	$\alpha = 6EI/K'AGL^2$ for beams
BI	A	beams moment of inertia
BK	A	shear coefficient for beams
BL	A	clear beam length
BMCL	A	dummy array
BMG	A	beams modulus of rigidity
CA	A	length of rigid end stub at lower end of a column
CAL	A	a column E/L for columns
CB	A	length of rigid end stub at top end of a column
CE	A	moduli of elasticity of columns
CEF	A	$\gamma = 2EI/L(1+2\alpha)$ for columns
CG	A	$\alpha = 6EI/K'AGL^2$ for columns
CI	A	column moments of inertia
CK	A	column shear coefficients
CL	A	column clear lengths
CLG	A	column moduli of rigidity
DB	A	beam depths

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
DEX	A	beam widths
DC	A	column widths in plane of frame
DCX	A	column widths perpendicular to plane of frame
IBC	S	index for cross-section type for beams in a story
ICC	S	index for cross-section type for columns in a column line
IRI	S	input unit number
IW	S	output unit number
IWI	S	unit number on which the modified frame properties are written
NCL	S	number of columns in a frame
NFR	S	number of frames in a frame group
NOQ	S	number of degrees of freedom in a frame
NSC	A	number of stories in different column lines
OSTB	S	index to check whether or not finite joint sizes are considered

Subroutine STIFF

AK	A	frame stiffness elements within the upper half band in a compacted column string storage
ICD	A	indices to identify the sidesway degree of freedom
IR	S	unit from which modified frame properties are read

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
IW	S	unit on which stiffness matrix is written for subsequent uses
NBM	S	total number of beams in the frame
NDA	A	diagonal addresses for the frame stiffness in a string storage
NS	S	total number of stories in the frame
<u>Subroutine GRAVT</u>		
A	A	reduced stiffness matrix of a frame
B	A	load or displacement vector for a frame
FC	A	dummy array
FX1	A	fixed end moments at left end of beams
FX2	A	fixed end moments at right end of beams
FX3	A	fixed end reactions at left end of beams
FX4	A	fixed end reaction at right end of beams
IFL	S	index to check whether the gravity loads on a frame are the same as those in the preceding frames
IR	S	unit from which gravity load data is read
NEL	S	total number of elements required to be stored in a string for the frame stiffness matrix

Subroutine LOAD

<u>SYSTEM</u>	<u>TYPE</u>	<u>DEFINITION</u>
A	S	distance of concentrated load from left column face
B	S	distance of concentrated load from right column face
CN	A	concentrated loads on a beam
DIS	A	distance of concentrated load from left column center line
FL1	S	left end beam fixed end moments for gravity loads
FL2	S	left end beam fixed end reaction for gravity loads
FR1	S	right end beam fixed end moments for gravity loads
FR2	S	right end beam fixed end reactions for gravity loads
FUD1	S	fixed end moments for UDL
FUD2	S	fixed end reactions for UDL
IS	S	story number
NCL	A	number of concentrated loads on beams
NI	S	bay number
UDL	A	uniformly distributed loads on beams

Subroutine FORCE

BETA	A	displacement vector of a frame
FA	S	final column axial load
Fl	S	final beam left end moment for static loads

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
F2	S	final beam right end moment for static loads
F3	S	final beam left end reaction for static loads
F4	S	final beam right end reaction for static loads
FC1	A	root-sum-square value of column top moment from dynamic analysis
FC2	A	root-sum-square value of column bottom moment from dynamic analysis
FC3	A	root-sum-square value of column axial load
FC4	A	root-sum-square value of column shear force from dynamic analysis
FMB	S	final column bottom moment
FMT	S	final column top moment
FS	S	final column shear
FX1	S	beam left end moment
FX2	S	beam right end moment
FX3	A	beam left end reaction
FX4	A	beam right end reaction
NBA	A	number of bays in the different towers of a frame
OPT	S	a control index
SQB1	A	root-sum-square value of beam left end moment in a frame
SQB2	A	root-sum-square value of beam right end moment in a frame

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
SQB3	A	root-sum-square value of beam left end reaction in a frame
SQB4	A	root-sum-square value of beam right end reaction in a frame

Subroutine COND

IR	S	input unit number
IW	S	output unit number
MM	S	size of the condensed stiffness matrix
NL	S	number of elements to be stored for frame stiffness
SC	A	frame sidesway stiffness matrix
TA	A	temporary storage array used in condensation
TX	A	temporary storage array
XX	A	dummy array

Subroutine CONDEN

JOPT	S	control index
NF	A	number of degrees of freedom other than sidesway degrees of freedom
NC	A	number of sidesway degrees of freedom

Subroutine DOTPRD

C,D	S	temporary array to perform matrix multiplication
L	S	number of degrees of freedom on a frame

Subroutine REDUCE

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
IW	S	output unit number
LI	S	integer indicating the row number of the first non-zero element in column I of the matrix
LJ	S	integer indicating the row number of the first non-zero element in column J of a matrix

Subroutine BSUB

B	A	load/displacement vector
LI	S	integer indicating the row number of the first non-zero element in column I of the matrix
NB	S	row number of the first non-zero element in the load vector

Subroutine ROOT

FB1,FB2 FB3,FB4	A	root-sum-square forces for beams
FC1,FC2 FC3,FC4	A	root-sum-square forces for columns
IR,IRR, IRI	S	input unit numbers
IW	S	output unit number
NFG	S	total number of frame groups in a building

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
<u>Subroutine STORY</u>		
SM	A	global mass vector
XBAR	S	X distance of the story mass centre from the reference point
XM	S	story mass
XMI	S	story mass moment of inertia
YBAR	S	y distance of the story mass centre from the reference point
<u>Subroutine FRAME</u>		
A	A	global stiffness matrix
DDA	A	perpendicular distance to a frame from the reference point
DDN	S	current value of perpendicular distance to a frame from the global reference point
DN	A	perpendicular distance to a frame from the mass centre at a particular story
IFP	S	current value of the frame-group-number
M	A	group number to which a frame belongs
MST	S	number of stories in a frame
NDIR	A	orientation of a frame (X or Y)

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
NFC	S	number of the frame-group to which the frame belongs
BB	A	condensed frame stiffness matrix
BB	A	temporary storage array used by matrix inversion routine
CO	A	trial vector in the matrix iteration for extracting eigenvalues and eigenvectors
DA	A	spectral displacement in a particular mode at a certain story level
DD	S	current value of the root-sum-square modal displacement
FORCE1	S	current value of the story force or torque derived from root-sum-square shears
FORCE2	S	current value of the story force or torque derived from shears obtained by taking the sum of the two absolute maximum modal shears
G	S	acceleration due to gravity
NA	A	array containing the row-numbers of the first non-zero terms in the columns of the global stiffness matrix
NP	S	number of period coordinates defining the design response spectrum
SAC	A	acceleration coordinates on the response spectrum
SHEAR1	S	root-sum-square of the modal shears
SHEAR2	S	sum of the two absolute maximum modal shears
T	S	current value of the modal period
TC	A	period coordinate on the design response spectrum
TD	S	period defining the boundary between the acceleration bound and velocity bound on the design spectrum
VAL	A	modal frequency
VEC	A	mode shape
XMULD	S	factor for modifying the spectral displacement for ductility
XMULP	S	factor for modifying the spectral force for ductility

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
<u>Subroutine EIGVAL</u>		
ALFA	S	modal mass obtained from a mode shape normalized so that the largest element is unity
BETAA	S	square-root of ALFA
CMAX	S	variable containing the current value of the absolute largest element in a trial vector
ERMAX	S	maximum value of ERR
ERR	S	the fractional difference between the values of corresponding elements of two successive trial vectors
NT	S	location of the absolute largest element in a trial vector
NVT	S	variable to test whether the origin for extraction of eigenvalues should be shifted to a new location
R1	S	test value to decide whether the convergence test may be skipped for an element of the trial vector
SH	S	shifted origin for extracting the eigenvalues
VT	S	square of the estimated value of the frequency being currently determined by iteration
VVT	S	estimated value of the frequency being currently determined
<u>Subroutine RESP</u>		
SI	S	modal participation factor
F	A	seismic story force in a frame
X	A	seismic story displacement in a frame

Subroutine HLOAD

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
DA	A	global displacement vectors for lateral loads
DN	S	frame story lateral displacements
DX	A	x-coordinates of the point of action of the story lateral forces from the reference point
DY	A	y-coordinates of the point of action of the story lateral forces from the reference point
F	A	frame story lateral forces
FH	A	lateral load vector in global coordinates
HX	A	x-components of story lateral loads
HY	A	y-components of story lateral loads
IR	S	input unit number
NST	S	total number of stories in the building
S	A	condensed stiffness matrix
X	A	array size equal to the number of stories

Subroutine SYSTEM

B	A	final frame displacement vector
BTA	A	unmodified frame displacement vector for different load cases
FS1,FS2 FS3,FS4	A	final fixed end load vectors
FX1,FX2 FX3,FX4	A	fixed end load vectors for different gravity load cases

<u>SYMBOL</u>	<u>TYPE</u>	<u>DEFINITION</u>
SQB1, SQB2 SQB3, SQB4	A	final root-sum-square forces for beams
SQC1, SQC2 SQC3, SQC4	A	final root-sum-square forces for columns

A3.4 INPUT DATA AND FORMATS

The input data required for the program ANGELS is obtained from those generated by program AIDS and stored in three data files as discussed in Section A2.5. However, the program ANGELS is designed to run also independent of program AIDS. In that case, the data is to be prepared by the user in rigid formats and kept in three unkeyed files in a manner as stored by the program AIDS. To achieve this, the necessary formats required for data input are described below.

1. MAIN FILE-RID1

CARD GROUP	NO. OF CARDS	FORTTRAN VARIABLES	FORMAT	REMARKS
A	1	NAME - case title	40A1	title used to identify the problem
B	1	NFT - total no. of frames NFG - no. of frame groups NNS - no. of stories NBA - max. no. of bays in the widest frame	I5 I5 I5 I5	
C	NNS/8 increased to the next higher integer	(H(I), - story height c/c of floor beams I=1,NNS)	8F10.3	building is idealized so that center lines of all beams in a floor lie on the same horizontal plane
D	NFT	II - frame no. M - frame group no. IFD - frame origination DDA - perp. distance from reference point ILFT - left identification	I5 I5 4X,A1 F15.5 3X,10A1	parallel to global X or Y direction (X or Y) to identify on the drawing the left of a frame as used in this analysis
E	1	NGR - no. of gravity load cases NWL - no. of lateral load cases NDYN - no. of seismic load cases (=0, or 1 only) NSYS - no. of load systems considered IOUT - index for special output	I5 I5 I5 I5 I5	=0, if output for mode shapes and frequencies is not required =1, if output for mode shapes and frequencies is required =2, if story shears are required in addition to mode shapes

CARD GROUP	NO. OF CARDS	FORTTRAN VARIABLES	FORMAT	REMARKS
F	1	NOPT - no. of degrees of freedom per story =1 for X translation only =2 for Y translation only =3 for X,Y translations and rotations	I5	-a value of 1 or 2 may be used for symmetrical buildings only and will result in a substantial reduction in the computation time - omit this card if all of NWL, NDYN and IOUT are equal to zero
G	NSYS	SMULP(I,J) - multipliers for load cases J=1,NK, NK=NGR+NWL+NDYN I=1,NSYS	8F10.3	- for each load system put first the multiplier for gravity load cases, then those for the lateral load cases and then for spectrum cases - omit this card if NSYS=0
H	1	NM - no. of modes considered IM - no. of modes to be extracted from one origin	I5 I5	- omit this card if both IOUT and NDYN are equal to zero
I		NP - no. of period coordinates	I5	- omit if NDYN=0
J	NP	TC - period value SAC - corresponding spectral acceleration	F10.3 F10.3	- start from the first period coordinate - spectral acceleration values are to be specified in units of g, the acceleration due to gravity - omit these if NDYN=0
K	1	XMU - ductility factor SF - scale factor for design acceleration BETA - angle of incidence of earthquake with global X-axis, degrees TD - period which separates the acceleration bound and velocity bound AG - acceleration due to gravity (g)	F10.3 F10.3 F10.3 F10.3 F10.3	- omit this card if NDYN=0

CARD GROUP	NO. OF CARDS	FORTTRAN VARIABLES	FORMAT	REMARKS
L	NNS	XM - translational floor mass	F10.0	- include one card per story
		XMI - rotational mass moment of ineratia of the floor	F10.0	- omit these cards if both NDYN and IOU are zero
		XB - X-distance of the center of mass from the reference point	F10.0	
		YB - Y-distance of the center of mass from the reference point	F10.0	

2. DATA FILE FOR FRAME PROPERTIES - RID2

CARD GROUP	NO. OF CARDS	FORTRAN VARIABLES	FORMAT	REMARKS
A	1	NT - no. of towers in a frame group NNS - no. of stories in the frame group NFR - no. of frames in the frame group	I5 I5 I5	- for a regular frame (not setback) NT=1
B	NT/5 increased to the next integral value	NSS(J), - no. of stories in each tower starting from the lowest tower	5I5	- part of the frame height containing the same number of bays is called a tower
C	NT/5 increased to next integer	NBAY(J), - no. of bays in each tower	5I5	- start from the base tower
D	(NT-1)/5 increased to next integer	NBS(J), - no. of bay-set-back at left, between two towers, starting from the lowest	5I5	- for a regular frame (i.e. if NT=1), omit this card
E	1	B(K) - lengths of bays in the base tower, starting from left,	8F10.3	
F	1	OSTB - index to identify whether or not the finite joint sizes are considered	I5	- OSTB=0 if finite joint sizes are not considered - OSTB=1 if finite joint sizes are considered in the analysis
G	1	IBC - index for beam cross-section type = C, if the cross-section is rectangular = 0, if the section is other than rectangular	1X,A1	- include one card per story followed by the card group H or I

CARD GROUP	NO. OF CARDS	FORTTRAN VARIABLES	FORMAT	REMARKS
H	no. of beams in a story	BE - beam modulus of elasticity	E12.6	- if IBC=0, include one card per beam, starting from left and complete for all beams in the story - omit this group of cards if IBC=0
		BMG - modulus of rigidity of beam (put zero if shear deformation is not considered)	E12.6	
		BK - shear coefficient (<1) of beam (put zero if shear deformation is not considered)	E12.6	
		DBX - width of the beam DB - depth of beam	E12.6	
I	no. of beams in a story	BE - same as H	E12.6	- omit this group of cards if IBC=C - if IBC=0, include one card per beam starting from left; complete for all beams in the story - repeat card groups G through I for all the stories of the frame group
		BMG - same as H	E12.6	
		BK - same as H	E12.6	
		DB - same as H	E12.6	
		AB - area of crosssection of the beam	E12.6	
		BI - moment of inertia of the beam	E12.6	
J	1	ICC - index for column cross-section type =C if the cross-section is rectangular =0 if the cross-section is other than rectangular	IX,A1	- include one card per column line followed by the card group K or L
		NSC - no. of columns in the column line	I5	
K	no. of columns in a column line	CE - column modulus of elasticity	E12.5	- omit this group of cards if ICC=0 - if ICC=C, include one card per column, starting from the lowest and complete for all columns in the column line
		CLG - modulus of rigidity for the column (put zero if shear deformation is not considered)	E12.5	
		CK - coefficient of shear (<1) for the column (put zero if shear deformation is not considered)	E12.5	

CARD GROUP	NO. OF CARDS	FORTRAN VARIABLES	FORMAT	REMARKS
K (cont)		DCX - width of column across E12.5 the plane of frame		
		DC - width of column in the E12.5 plane of frame		
L	no. of columns in a column line	CE - same as K CLG - same as K CK - same as K DC - same as K AC - area of cross-section of the column CI - moment of inertia of the column	E12.5 E12.5 E12.5 E12.5 E12.5 E12.5	- omit this group of cards if ICC=C - if ICC=0 include one card per column starting from the lowest and complete for all columns in the column line
				- repeat the card groups J through L for all the column lines in the frame group

3. DATA FILE FOR STATIC LOADS - RID3

CARD GROUP	NO. OF CARDS	FORTRAN VARIABLES	FORMAT	REMARKS
A	1	NFR - no. of frames in a group	I5	
B	1	IFL - index for gravity loads repetitive over different frames =0 if the gravity load in the present frame is not similar to that of the preceding frame =1 if the load in the present frame is similar to that of the preceding frame for all gravity load cases	I5	if IFL=1, omit card groups C through E
C	no. of stories in the frame	UDL(I,J) - value of load J=1,NBAY distributed uniformly over the beam	8F10.3	- include one card per story for beams in that story starting from the lowest story
D	1	NC - no. of concentrated loads on a beam, starting from the left of a story		- put one card per beam followed by the card group E
E	1	CNL(I) - value of concentrated load DIS(I) - distance of the I=1,NC load from left column center line.	F10.3 F10.3	- omit this card if NC=0 - if NC≠0, include one card per beam and put the load value followed by its distance; continue this way for all the concentrated loads on the beam
				- repeat card groups D and E for all beams in a story and this way for all the stories in the frame

CARD GROUP	NO. OF CARDS	FORTRAN VARIABLES	FORMAT	REMARKS
				- then repeat card groups C through E for all gravity load cases (only if IFL=0)
				- then repeat card groups B through E for all frames in a frame group; then this way continue for all frame groups in the building
F	no. of stories in the building	HX - global x-components of the story lateral load HY - global y-components of the story lateral load DX - x-coordinate of the point application DY - y-coordinate of the point application	F15.5 F15.5 F15.5 F15.5	- omit this group of cards if NWL=0
				- repeat card group F for all lateral load cases

A3.5 OUTPUT FROM THE PROGRAM

The program echo prints all input data and options as supplied by the user. In addition, the following outputs may be obtained:

1. Member forces in terms of end moments and shears and column axial loads may be obtained for any number of weighted combinations of a maximum of four gravity load cases, a maximum of three lateral load cases and a design response spectrum case. If finite joint sizes are specified, the above forces will be computed at the faces of the joints; otherwise, at the center's
2. Structural mode shapes and frequencies (optional).
3. For seismic response spectrum, the design values of story shears and cumulative torsional moments, story forces and torsional moments, story displacements and rotations for the structure as a whole. The design values are obtained by two different methods, one by taking the root-sum-square of the modal values, and the other by taking the sum of two absolute maximum modal values. The modal values are themselves adjusted to account for ductility. These items will be evaluated and printed out at the user's options.

A3.6 EXAMPLE OF ANALYSIS

The building shown in Fig. A.5 is analyzed. The data generated by program AIDS (as shown in Section A2.5) is input through three 'unkeyed' disc files. The previously created 'load module' of the program ANGELS is then run and the output given below is obtained.

***** ANALYSIS OF BUILDINGS FOR GRAVITY, EARTHQUAKE AND LATERAL LOADS *****

```
AAAAAA NNN NNN GGGGGG PEEEEEEEE LLL SSSSSS
AAAAAAA NNN GGGGGGGGG PEEEEEEEE LLL SSSSSSSSS
AAA AAA NNN GGG GGG PEEEEEEEE LLL SSS
AAA AAA NNN GGG GGG PEEEEEEEE LLL SSS
AAAAAAA NNN GGG GGG PEEEEEEEE LLL SSSSSSS
AAAAAAA NNN GGG GGG PEEEEEEEE LLL SSSSSSS
AAA AAA NNN GGG GGG PEEEEEEEE LLL SSS
AAA AAA NNN GGG GGG PEEEEEEEE LLL SSS
AAA AAA NNN GGG GGG PEEEEEEEE LLL SSSSSSS
AAA AAA NNN GGG GGG PEEEEEEEE LLL SSSSSSS
```

***** ANALYSIS OF BUILDINGS FOR GRAVITY, EARTHQUAKE AND LATERAL LOADS *****

PROGRAMMED BY --- J. U. KHANJOKER ---
DEPARTMENT OF CIVIL ENGINEERING, CARLETON UNIVERSITY, OTTAWA, CANADA ---

NAME OF PROJECT :: EXAMPLE BUILDING

TOTAL NO. OF FRAMES IN THIS BUILDING = 5
TOTAL NO. OF FRAME GROUPS = 3
TOTAL NO. OF STORIES IN THE BUILDING = 3
NO. OF BAYS IN THE WIDEST FRAME = 2

*** UNITS ARE CONSISTENT

FRAME POSITION DATA

FRAME	FRAME GROUP	DIRECTION	DIST FROM ORIGIN	LEFT SYMBOL
1	1	X	.00	WEST-S
2	1	X	240.00	WEST-N
3	2	Y	240.00	SOUTH-MID
4	2	Y	360.00	SOUTH-E
5	3	Y	.00	SOUTH-N

NO. OF GRAVITY LOAD CASES = 1
 NO. OF LATERAL LOAD CASES = 1
 NO. OF SEISMIC LOAD CASES = 1
 NO. OF LOAD SYSTEMS (LOAD COMBINATIONS) = 4

 STORY TRANSLATION CODE = 3
 NUMBER OF MODES TO BE CONSIDERED = 9
 NUMBER OF MODES TO BE EXTRACTED FROM ONE ORIGIN = 5
 DUCTILITY FACTOR = 4.00
 SCALE FACTOR FOR DESIGN GROUND ACCELERATION = .08
 ANGLE OF INCIDENCE OF EARTHQUAKE = 30.00

 ACCELERATION SPECTRUM

NUMBER OF PERIOD COORDS. = 4
 PERIOD BOUND. = .50
 ACCELERATION DUE TO GRAVITY ... = 386.40

PERIOD ACCELERATION

.100 3.000
 .420 3.000
 5.000 .260
 15.000 .029

 STORY HEIGHTS, C/C OF
 INTERSECTING FLOOR BEAMS

BETWEEN STORY
 FLOOR LEVELS HEIGHTS

0 AND 1 120.000
 1 AND 2 120.000
 2 AND 3 100.000

 FRAME PROPERTIES

 FRAME GROUP NO. 1

NO. OF TOWERS = 2
 NO. OF STORIES = 3

TOWER NO.	NO. OF BAYS	NO. OF STORIES
1	2	2
2	1	1

NO. OF BAYS-SETBACK AT LEFT

BETWN TOWERS 1 AND 2 = 1

LENGTHS OF BAYS C/C OF
 COLS. STARTING AT LEFT

BAY NO.	LENGTH
1	240.00
2	120.00

 BEAM PROPERTIES

STORY	BAY TYPE	WIDTH	DEPTH	AREA	INERTIA	ELASTICITY	RIGIDITY	SH. COEF
1	1 C	10.00	20.00			.30000E 04	.12000E 04	.833
	2 C	10.00	20.00			.30000E 04	.12000E 04	.833
2	1 C	10.00	20.00			.30000E 04	.12000E 04	.833
	2 C	10.00	20.00			.30000E 04	.12000E 04	.833
3	2 C	10.00	15.00			.30000E 04	.12000E 04	.833

COLUMN PROPERTIES

COLM LINE	STORY TYPE	CROSS WIDTH	WIDTH	AREA	INERTIA	ELASTICITY	RIGIDITY	SH.COEF
1	1	C	12.00	16.00		.30000E 04	.11500E 04	.833
	2	C	12.00	14.00		.30000E 04	.11500E 04	.833
2	1	C	16.00	18.00		.30000E 04	.11500E 04	.833
	2	C	12.00	16.00		.30000E 04	.11500E 04	.833
	3	C	12.00	12.00		.30000E 04	.11500E 04	.833
3	1	C	16.00	18.00		.30000E 04	.11500E 04	.833
	2	C	12.00	16.00		.30000E 04	.11500E 04	.833
	3	C	12.00	12.00		.30000E 04	.11500E 04	.833

** FINITE JOINT SIZES ARE CONSIDERED **

** SHEAR DEFORMATIONS ARE CONSIDERED **

FRAME GROUP NO. 2

NO. OF TOWERS = 1
NO. OF STORIES = 3

TOWER NO. NO. OF BAYS NO. OF STORIES
1 1 3

LENGTHS OF BAYS C/C OF
COLS. STARTING AT LEFT

BAY NO. LENGTH

1 240.00

 BEAM PROPERTIES

STORY	BAY	TYPE	WIDTH	DEPTH	AREA	INERTIA	ELASTICITY	RIGIDITY	SH.COEF
1	1	C	10.00	18.00			.30000E 04	.12000E 04	.833
2	1	C	10.00	18.00			.30000E 04	.12000E 04	.833
3	1	C	10.00	15.00			.30000E 04	.12000E 04	.833

 COLUMN PROPERTIES

COLM	STORY	TYPE	CROSS WIDTH	WIDTH	AREA	INERTIA	ELASTICITY	RIGIDITY	SH.COEF
1	1	C	18.00	16.00			.30000E 04	.11500E 04	.833
	2	C	16.00	12.00			.30000E 04	.11500E 04	.833
	3	C	12.00	12.00			.30000E 04	.11500E 04	.833
2	1	C	18.00	16.00			.30000E 04	.11500E 04	.833
	2	C	16.00	12.00			.30000E 04	.11500E 04	.833
	3	C	12.00	12.00			.30000E 04	.11500E 04	.833

** FINITE JOINT SIZES ARE CONSIDERED **

** SHEAR DEFORMATIONS ARE CONSIDERED **

 FRAME GROUP NO. 3

NO. OF TOWERS = 1
 NO. OF STORIES = 2

TOWER NO.	NO. OF BAYS	NO. OF STORIES
1	1	2

LENGTHS OF BAYS C/C OF
COLS. STARTING AT LEFT

BAY NO.	LENGTH
1	240.00

BEAM PROPERTIES

STORY	BAY TYPE	WIDTH	DEPTH	AREA	INERTIA	ELASTICITY	RIGIDITY	SH. COEF
1	1 C	10.00	15.00			.30000E 04	.12000E 04	.833
2	1 C	10.00	15.00			.30000E 04	.12000E 04	.833

COLUMN PROPERTIES

COLM. LINE	STORY TYPE	CROSS WIDTH	WIDTH	AREA	INERTIA	ELASTICITY	RIGIDITY	SH. COEF
1	1 C	16.00	12.00			.30000E 04	.11500E 04	.833
	2 C	14.00	12.00			.30000E 04	.11500E 04	.833
2	1 C	16.00	12.00			.30000E 04	.11500E 04	.833
	2 C	14.00	12.00			.30000E 04	.11500E 04	.833

** FINITE JOINT SIZES ARE CONSIDERED **

** SHEAR DEFORMATIONS ARE CONSIDERED **

NOTE:- BEAM/COL TYPE:: C=CONC. SECTION, O=NOT A CONC. RECT. SECTION

 DATA FOR GRAVITY LOAD CASES

 FRAME NO. 1

 GRAVITY LOAD CASE 1 :

 SUPPLIED LOAD DATA

STORY	BAY	UDL	CONCENTRATED LOAD DATA			
			LOAD-1 DIST-1	LOAD-2 DIST-2	LOAD-3 DIST-3	LOAD-4 DIST-4
1	1	.070				
	2	.070	4.00	60.00		
2	1	.070				
	2	.070	4.00	60.00		
3	2	.030				

 GRAVITY LOADS (FOR ALL LOAD CASES) IN FRAME 2
 ARE EXACTLY SAME AS THOSE IN FRAME 1

FRAME NO. 3

GRAVITY LOAD CASE 1 :

SUPPLIED LOAD DATA

		CONCENTRATED LOAD DATA			
		LOAD-1, DIST-1	LOAD-2 DIST-2	LOAD-3 DIST-3	LOAD-4 DIST-4
STORY	BAY	UDL			
1	1	.120	4.00	120.00	
2	1	.120	4.00	120.00	
3	1	.030			

GRAVITY LOADS (FOR ALL LOAD CASES) IN FRAME 4
ARE EXACTLY SAME AS THOSE IN FRAME 3

FRAME NO. 5

GRAVITY LOAD CASE 1 :

SUPPLIED LOAD DATA

STORY BAY UDL LOAD-1 DIST-1 LOAD-2 DIST-2 LOAD-3 DIST-3 LOAD-4 DIST-4

1 1 .070
2 1 .070

STORY MASS DATA

LEVEL	MASS	MASS INERTIA	XBAR	YBAR
1	.200	2900.000	180.00	120.00
2	.200	2900.000	180.00	120.00
3	.060	400.000	300.00	120.00

LATERAL LOAD DATA

LATERAL LOAD CASE 1:

LEVEL	X-FORCE	Y-FORCE	X-DIST	Y-DIST
1	4.000	6.000	180.000	120.020
2	4.000	4.000	204.000	120.000
3	1.000	1.000	300.000	120.000

 DYNAMIC ANALYSIS

MODE NUMBER	FREQUENCY
1	.17152E 02
2	.25657E 02
3	.30442E 02
4	.60302E 02
5	.70373E 02
6	.78203E 02
7	.10136E 03
8	.10537E 03
9	.14247E 03

 MODE SHAPES

LEVEL	DIRN	1	2	3	4
3	X TRAN	.29002E-06	.22365E 01	-.20657E-05	.14541E-05
	Y TRAN	-.20855E 01	.19563E-05	.20593E 01	.19359E 01
	ROTN	.13137E-02	.10001E-07	.15824E-01	-.12427E-01
2	X TRAN	.22196E-06	.17116E 01	-.15809E-05	-.30044E-06
	Y TRAN	-.17724E 01	.25645E-06	-.39298E 03	-.96467E-01
	ROTN	.11982E-02	.02076E-08	.13239E-01	-.70433E-02
1	X TRAN	.97897E-07	.75491E 00	-.69726E-00	-.59071E-00
	Y TRAN	-.72250E 03	-.29924E-07	-.36514E 03	-.10755E 01
	ROTN	.72367E-03	.35412E-08	.57172E-02	-.14572E-02

LEVEL	DIRN	5	6	7	8
3	X TRAN	.31301E-01	.53393E-05	.13666E-01	-.70070E-05
	Y TRAN	.22201E-05	.20213E-01	.58567E-06	.13421E-00
	ROTN	.49020E-09	-.43251E-02	-.18255E-06	-.41820E-01
2	X TRAN	-.66613E-00	-.11363E-05	-.12754E-01	.65396E-05
	Y TRAN	-.20685E-05	-.12495E-01	-.79926E-06	-.18311E-00
	ROTN	-.80473E-09	-.30866E-02	.12872E-07	.29487E-02
1	X TRAN	-.12716E-01	-.21691E-05	.16772E-01	-.85998E-05
	Y TRAN	.29652E-05	.11178E-01	.16736E-05	.38338E-00
	ROTN	-.11643E-07	-.73721E-02	.39550E-07	.90602E-02

LEVEL	DIRN	9
3	X TRAN	.31111E-11
	Y TRAN	.47449E-00
	ROTN	.18053E-01
2	X TRAN	-.73890E-11
	Y TRAN	-.31650E-00
	ROTN	-.10273E-01
1	X TRAN	-.58334E-12
	Y TRAN	.37521E-00
	ROTN	.13155E-01

GLOBAL SEISMIC STORY FORCES AND DISPLACEMENTS

STORY LEVEL	DIRN	SHEAR OR CUMULATIVE TOR. MOMENT		FORCE OR TORSIONAL MOMENT		DISP.
		CODE 1	CODE 2	CODE 1	CODE 2	
3	X	2.827	3.695	2.827	3.695	.2593E 00
	Y	1.467	1.854	1.467	1.854	.3103E 00
	THETA	26.580	34.857	26.580	34.857	.2067E-03
2	X	9.122	9.929	6.295	6.235	.1981E 00
	Y	5.266	5.650	3.798	3.797	.2636E 00
	THETA	119.356	149.789	92.776	114.933	.1832E-03
1	X	12.040	13.163	2.918	3.234	.8765E-01
	Y	6.900	7.821	1.635	2.171	.1077E 00
	THETA	154.394	190.388	35.038	40.599	.1089E-03

**CODE 1 - ROOT SUM SQUARE

**CODE 2 - SUM OF TWO MAXIMUM ABSOLUTE NODAL VALUES

 * ANALYSIS FOR LOAD SYSTEMS *

LOAD CASE MULTIPLYING FACTORS :

LOAD SYSTEM	1	2	3	4
GRAVITY LOAD CASE 1	1.000	.000	.000	1.000
LATERAL LOAD CASE 1	.000	1.000	.000	.750
SEISMIC LOAD CASE	.000	.000	1.000	.800

 FRAME NO. 1

FOR LOAD SYSTEM 1

BEAM END ACTIONS

STORY BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1				
1	.22914E 03	-.30429E 03	.74680E 01	.81420E 01
2	.14911E 03	-.78763E 02	.62596E 01	.48804E 01
2				
1	.16747E 03	-.31956E 03	.71990E 01	.85510E 01
2	.20367E 03	-.55482E 02	.70649E 01	.42151E 01
3				
2	.16414E 02	-.10202E 02	.16760E 01	.15632E 01

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	-.97354E 02	-.52009E 02	-.15717E 02	-.13578E 01
	2	.53536E 02	.29086E 02	-.34254E 02	.75111E 00
	3	.43168E 02	.23573E 02	-.12029E 02	.60674E 00
2	1	-.18626E 03	-.14688E 03	-.76890E 01	-.33314E 01
	2	.11124E 03	.90865E 02	-.18593E 02	.20210E 01
	3	.67857E 02	.63182E 02	-.65183E 01	.13104E 01
3	2	-.24866E 02	-.60070E 00	-.18568E 01	-.30683E 00
	3	.17900E 02	.74135E 01	-.17432E 01	.30683E 00

STORY LATERAL DISPLT

ST.LEVEL	DISPLT
1	.12372E-03
2	.30374E-02
3	-.45996E-02

FOR LOAD SYSTEM 2

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	-.54331E 02	-.48018E 02	-.45896E 00	.45896E 00
	2	-.90822E 02	-.99312E 02	-.18641E 01	.18641E 01
2	1	-.35953E 02	-.32719E 02	-.30521E 00	.30521E 00
	2	-.57887E 02	-.66264E 02	-.11938E 01	.11938E 01
3	2	-.18971E 02	-.17694E 02	-.33950E 00	.33950E 00

COLUMN END ACTIONS

STORY	COLUMN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.20282E 02	.44273E 02	.76417E 00	.58686E 00
	2	.69338E 02	.10000E 03	.26332E 01	.15399E 01
	3	.49779E 02	.89578E 02	-.33974E 01	.12669E 01
2	1	.32262E 02	.26025E 02	.30522E 00	.58286E 00
	2	.67176E 02	.62044E 02	.12281E 01	.12922E 01
	3	.49374E 02	.44275E 02	-.15333E 01	.93650E 00
3	2	.17963E 02	.18150E 02	.33952E 00	.43510E 00
	3	.16972E 02	.13396E 02	-.33952E 00	.36610E 00

STORY LATERAL DISPLT

ST. LEVEL	DISPLT
1	.12628E-01
2	.29503E-01
3	.38967E-01

FOR LOAD SYSTEM 3

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.92482E 02	.81757E 02	.78134E 00	.78134E 00 MAX
	2	-.92482E 02	-.81757E 02	-.78134E 00	-.78134E 00 MIN
	3	.15528E 03	.16992E 03	.31802E 01	.31802E 01 MAX
	4	.15528E 03	-.16992E 03	-.31802E 01	-.31802E 01 MIN
2	1	.50053E 02	.53483E 02	.49831E 00	.49831E 00 MAX
	2	-.50053E 02	-.53483E 02	-.49831E 00	-.49831E 00 MIN
	3	.94612E 02	.10815E 03	.19496E 01	.19496E 01 MAX
	4	-.94612E 02	-.10815E 03	-.19496E 01	-.19496E 01 MIN
3	2	.32745E 02	.30808E 02	.50841E 00	.50841E 00 MAX
	3	-.32745E 02	-.30808E 02	-.50841E 00	-.50841E 00 MIN

COLUMN END ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	1	.38910E 02	.79289E 02	.12772E 01	.10731E 01 MAX
		2	-.38910E 02	-.79289E 02	-.12772E 01	-.10731E 01 MIN
	2	1	.12545E 03	.17735E 03	.43915E 01	.27523E 01 MAX
		2	-.12545E 03	-.17735E 03	-.43915E 01	-.27523E 01 MIN
	3	1	.91834E 02	.15941E 03	.56666E 01	.22830E 01 MAX
		2	-.91834E 02	-.15941E 03	-.56666E 01	-.22830E 01 MIN
2	1	1	.52721E 02	.41624E 02	.49833E 00	.94260E 00 MAX
		2	-.52721E 02	-.41624E 02	-.49833E 00	-.94260E 00 MIN
	2	1	.11023E 03	.10077E 03	.20239E 01	.21096E 01 MAX
		2	-.11023E 03	-.10077E 03	-.20239E 01	-.21096E 01 MIN
	3	1	.81039E 02	.71172E 02	.25174E 01	.15208E 01 MAX
		2	-.81039E 02	-.71172E 02	-.25174E 01	-.15208E 01 MIN
3	2	1	.30964E 02	.32087E 02	.58845E 00	.75920E 00 MAX
		2	-.30964E 02	-.32087E 02	-.58845E 00	-.75920E 00 MIN
	3	1	.29416E 02	.20064E 02	.58845E 00	.65613E 00 MAX
		2	-.29416E 02	-.20064E 02	-.58845E 00	-.65613E 00 MIN

FOR LOAD SYSTEM 4

BEAM END ACTIONS

STORY	BAY-NO	LEFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.26238E 03	-.27490E 03	.77489E 01	.91113E 01 MAX
	2	.11441E 03	-.40571E 03	.64987E 01	.78611E 01 MIN
	3	.20522E 03	-.17313E 02	.74122E 01	.88298E 01 MAX
2	1	.43238E 02	-.28918E 03	.23110E 01	.37278E 01 MIN
	2	.18743E 03	-.30131E 03	.73688E 01	.91785E 01 MAX
	3	.93583E 02	-.38689E 03	.65715E 01	.83812E 01 MIN
	4	.23595E 03	-.18659E 02	.77293E 01	.66701E 01 MAX
	5	.84568E 02	-.19170E 03	.46059E 01	.35527E 01 MIN
	6	.26301E 02	.10930E 01	.18929E 01	.22886E 01 MAX
	7	.24311E 02	-.48199E 02	.95142E 00	.13471E 01 MIN

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE	
1	1	-.51015E 02	.44626E 02	-.14122E 02	-.59226E-01	MAX
	2	-.11327E 03	-.82235E 02	-.16166E 02	-.17762E 01	MIN
	3	.20593E 03	.24597E 03	-.28766E 02	.41079E 01	MAX
	1	.52205E 01	-.37795E 02	-.35792E 02	-.29581E 00	MIN
	2	.15397E 03	.21829E 03	-.10043E 02	.33833E 01	MAX
	3	.70357E 01	-.36773E 02	-.19110E 02	-.26953E 00	MIN
2	1	-.11989E 03	-.94060E 02	-.70614E 01	-.21402E 01	MAX
	2	-.20424E 03	-.16066E 03	-.78588E 01	-.36483E 01	MIN
	3	.24980E 03	.21802E 03	-.16053E 02	.46779E 01	MAX
	1	.73433E 02	.56779E 02	-.19291E 02	.13025E 01	MIN
	2	.16972E 03	.15333E 03	-.56543E 01	.32294E 01	MAX
	3	.40056E 02	.39451E 02	-.96822E 01	.79608E 00	MIN
3	2	.13377E 02	.38681E 02	-.11314E 01	.62685E 00	MAX
	3	-.36165E 02	-.12658E 02	-.20729E 01	-.58787E 00	MIN
	1	.54162E 02	.37512E 02	-.15271E 01	.11094E 01	MAX
	2	.70967E 01	-.25906E 01	-.24686E 01	.56404E-01	MIN

FRAME NO. 2

FOR LOAD SYSTEM 1

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.22914E 03	-.30429E 03	.74680E 01	.81420E 01
	2	.14911E 03	-.78763E 02	.62596E 01	.48804E 01
2	1	.16747E 03	-.31956E 03	.71990E 01	.85510E 01
	2	.20367E 03	-.55482E 02	.70649E 01	.42151E 01
3	2	.16414E 02	-.10282E 02	.16768E 01	.15632E 01

COLUMN END ACTIONS

STORY	COL-LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	-.97354E 02	-.52009E 02	-.15717E 02	-.13578E 01
	2	.53536E 02	.29086E 02	-.34254E 02	.75111E 00
	3	.43168E 02	.23573E 02	-.12029E 02	.60674E 03
2	1	-.18626E 03	-.14688E 03	-.76890E 01	-.33314E 01
	2	.11124E 03	.90865E 02	-.18593E 02	.20210E 01
	3	.67857E 02	.63182E 02	-.65183E 01	.13104E 01
3	2	-.24866E 02	-.60070E 00	-.18568E 01	-.30683E 00
	3	.17900E 02	.74135E 01	-.17432E 01	.30683E 00

STORY LATERAL DISPLT

ST.LEVEL	DISPLT
1	.12372E-03
2	.30374E-02
3	-.45998E-02

FOR LOAD SYSTEM 2

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	-.77462E 02	-.68515E 02	-.65460E 00	.65460E 00
	2	-.13203E 03	-.14491E 03	-.27151E 01	.27151E 01
2	1	-.37483E 02	-.34823E 02	-.32136E 00	.32136E 00
	2	-.60711E 02	-.68506E 02	-.12425E 01	.12425E 01
3	2	-.18904E 02	-.17726E 02	-.33917E 00	.33917E 00

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.45176E 02	.79390E 02	.97596E 00	.11324E 01
	2	.12903E 03	.17282E 03	.33209E 01	.27441E 01
	3	.99280E 02	.15700E 03	-.42968E 01	.23298E 01
2	1	.34239E 02	.20706E 02	.32137E 00	.54945E 00
	2	.72662E 02	.61044E 02	.12603E 01	.13371E 01
	3	.52457E 02	.37749E 02	-.15816E 01	.90206E 00
3	2	.17933E 02	.17717E 02	.33919E 00	.42952E 00
	3	.17008E 02	.13297E 02	-.33919E 00	.36733E 00

STORY LATERAL DISPLT

ST. LEVEL	DISPLT
1	.20853E-01
2	.40201E-01
3	.49920E-01

FOR LOAD SYSTEM 3

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.92482E 02	.81757E 02	.78134E 00	.78134E 00 MAX
	2	-.92482E 02	-.81757E 02	-.78134E 00	-.78134E 00 MIN
	3	.15528E 03	.16992E 03	.31882E 01	.31882E 01 MAX
	1	-.15528E 03	-.16992E 03	-.31882E 01	-.31882E 01 MIN
	2	.50653E 02	.53483E 02	.49831E 00	.49831E 00 MAX
	3	-.50653E 02	-.53483E 02	-.49831E 00	-.49831E 00 MIN
	1	.94612E 02	.10815E 03	.19496E 01	.19496E 01 MAX
	2	-.94612E 02	-.10815E 03	-.19496E 01	-.19496E 01 MIN
	3	.32745E 02	.30808E 02	.50841E 00	.50841E 00 MAX
	1	-.32745E 02	-.30808E 02	-.50841E 00	-.50841E 00 MIN

COLUMN END ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	1	.38910E 02	.79289E 02	.12772E 01	.10731E 01 MAX
		2	-.38910E 02	-.79289E 02	-.12772E 01	-.10731E 01 MIN
		3	.12545E 03	.17736E 03	.43915E 01	.27523E 01 MAX
		3	-.12545E 03	-.17736E 03	-.43915E 01	-.27523E 01 MIN
		3	.91834E 02	.15941E 03	.56666E 01	.22830E 01 MAX
		3	-.91834E 02	-.15941E 03	-.56666E 01	-.22830E 01 MIN
2	1	1	.52721E 02	.41624E 02	.49833E 00	.94260E 00 MAX
		2	-.52721E 02	-.41624E 02	-.49833E 00	-.94260E 00 MIN
		2	.11023E 03	.10077E 03	.20239E 01	.21096E 01 MAX
		3	-.11023E 03	-.10077E 03	-.20239E 01	-.21096E 01 MIN
		3	.81039E 02	.71171E 02	.25174E 01	.15208E 01 MAX
		3	-.81039E 02	-.71171E 02	-.25174E 01	-.15208E 01 MIN
3	2	1	.30964E 02	.32087E 02	.58846E 00	.75920E 00 MAX
		2	-.30964E 02	-.32087E 02	-.58846E 00	-.75920E 00 MIN
		3	.29416E 02	.25064E 02	.58845E 00	.65813E 00 MAX
		3	-.29416E 02	-.25064E 02	-.58845E 00	-.65813E 00 MIN

FOR LOAD SYSTEM 4

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.24503E 03	-.29027E 03	.76021E 01	.92580E 01 MAX
	2	.97058E 02	-.42108E 03	.63520E 01	.80079E 01 MIN
	2	.17431E 03	-.51513E 02	.67739E 01	.94673E 01 MAX
	2	-.74146E 02	-.32338E 03	.16727E 01	.43661E 01 MIN
	3	.18628E 03	-.30289E 03	.73567E 01	.91906E 01 MAX
	3	.92435E 02	-.38847E 03	.65594E 01	.83933E 01 MIN
	3	.23383E 03	-.20340E 02	.76928E 01	.67067E 01 MAX
	3	.82450E 02	-.19338E 03	.45733E 01	.35872E 01 MIN
	3	.28432E 02	.10690E 01	.18931E 01	.22883E 01 MAX
	3	-.23960E 02	-.48223E 02	.95167E 00	.13469E 01 MIN

COLUMN END ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	1	-.32344E 02	.70964E 02	-.13963E 02	.34994E 00 MAX
	2	2	-.94600E 02	-.55898E 02	-.16007E 02	-.13670E 01 MIN
	3	3	.25066E 03	.30058E 03	-.28250E 02	.50110E 01 MAX
	1	1	.49950E 02	.16817E 02	-.35277E 02	.60729E 00 MIN
	2	2	.19110E 03	.26885E 03	-.10718E 02	.41805E 01 MAX
	3	3	.44161E 02	.13795E 02	-.19785E 02	.52768E 00 MIN
	1	1	-.11841E 03	-.90049E 02	-.70493E 01	-.21652E 01 MAX
	2	2	-.20276E 03	-.16465E 03	-.78466E 01	-.36734E 01 MIN
	3	3	.25392E 03	.21727E 03	-.16028E 02	.47115E 01 MAX
	1	1	.77546E 02	.50029E 02	-.19267E 02	.13361E 01 MIN
	2	2	.17203E 03	.14843E 03	-.56906E 01	.32036E 01 MAX
	3	3	.42368E 02	.34506E 02	-.97185E 01	.77025E 00 MIN
	1	1	.13355E 02	.38357E 02	-.11316E 01	.62267E 00 MAX
	2	2	-.36188E 02	-.12982E 02	-.20732E 01	-.59205E 00 MIN
	3	3	.54189E 02	.37437E 02	-.15269E 01	.11088E 01 MAX
	1	1	.71237E 01	.26651E 01	-.24684E 01	.55829E 01 MIN

FRAME NO. 3

FOR LOAD SYSTEM 1

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.55444E 03	-.55440E 03	.15440E 02	.15440E 02
2	1	.53000E 03	-.52906E 03	.15682E 02	.15678E 02
3	1	.12509E 03	-.13039E 03	.34170E 01	.34222E 01

COLUMN END ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1		-.32347E 03	-.17210E 03	-.36400E 02	-.44646E 01
	2		.32349E 03	.17208E 03	-.36400E 02	.44647E 01
2	1		-.31521E 03	-.26679E 03	-.20000E 02	-.57060E 01
	2		.31520E 03	.26673E 03	-.20000E 02	.57060E 01
3	1		-.12220E 03	-.22272E 03	-.35979E 01	-.41062E 01
	2		.12067E 03	.22220E 03	-.36022E 01	.41062E 01

STORY LATERAL DISPLT

ST.LEVEL	DISPLT
1	-.35309E-05
2	-.39708E-04
3	-.26776E-03

FOR LOAD SYSTEM 2

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	-.13093E 03	-.13093E 03	-.11690E 01	.11690E 01
2	1	-.67667E 02	-.67671E 02	-.59359E 00	.59359E 00
3	1	-.17361E 02	-.17444E 02	-.15265E 00	.15265E 00

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.75160E 02	.18178E 03	.19153E 01	.23147E 01
	2	.75164E 02	.18178E 03	-.19152E 01	.23148E 01
2	1	.56100E 02	.36148E 02	.74631E 00	.90439E 00
	2	.56091E 02	.36144E 02	-.74627E 00	.90426E 00
3	1	.16509E 02	.47154E 01	.15270E 00	.25267E 00
	2	.16457E 02	.47283E 01	-.15266E 00	.25371E 00

STORY LATERAL DISPLT

ST. LEVEL	DISPLT
-----------	--------

1	.35950E-01
2	.77713E-01
3	.92974E-01

FOR LOAD SYSTEM 3

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.99067E 02	.99066E 02	.88452E 00	.88452E 00 MAX
		-.99067E 02	-.99066E 02	-.88452E 00	-.88452E 00 MIN
2	1	.70349E 02	.70350E 02	.61713E 00	.61713E 00 MAX
		-.70349E 02	-.70356E 02	-.61713E 00	-.61713E 00 MIN
3	1	.21970E 02	.22073E 02	.19317E 00	.19317E 00 MAX
		-.21970E 02	-.22073E 02	-.19317E 00	-.19317E 00 MIN

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOY-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.41465E 02	.12176E 03	.16838E 01	.14684E 01 MAX
	2	-.41465E 02	-.12176E 03	-.16838E 01	-.14684E 01 MIN
	3	.41470E 02	.12176E 03	.16838E 01	.14685E 01 MAX
2	1	-.41470E 02	-.12176E 03	-.16838E 01	-.14685E 01 MIN
	2	.52714E 02	.44398E 02	.80800E 00	.95084E 00 MAX
	3	-.52714E 02	-.44398E 02	-.80800E 00	-.95084E 00 MIN
	4	.52702E 02	.44393E 02	.80797E 00	.95067E 00 MAX
	5	-.52702E 02	-.44393E 02	-.80797E 00	-.95067E 00 MIN
	6	.20529E 02	.11307E 02	.19321E 00	.37402E 00 MAX
	7	-.20529E 02	-.11307E 02	-.19321E 00	-.37402E 00 MIN
	8	.20436E 02	.11321E 02	.19318E 00	.37530E 00 MAX
	9	-.20436E 02	-.11321E 02	-.19318E 00	-.37530E 00 MIN

FOR LOAD SYSTEM 4

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.53550E 03	-.57334E 03	.15271E 02	.17024E 02 MAX
	2	.37699E 03	-.73105E 03	.13856E 02	.15609E 02 MIN
	3	.53553E 03	-.52403E 03	.15730E 02	.16617E 02 MAX
	4	.42297E 03	-.63660E 03	.14743E 02	.15630E 02 MIN
	5	.13445E 03	-.12582E 03	.34579E 01	.36912E 01 MAX
	6	.99297E 02	-.16113E 03	.31488E 01	.33821E 01 MIN

COLUMN END-ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	1	-.23393E 03	.61635E 02	-.33616E 02	-.15539E 01 MAX
	2	2	-.30028E 03	-.13317E 03	-.36310E 02	-.39033E 01 MIN
	2	1	.41304E 03	.40582E 03	-.36489E 02	.73755E 01 MAX
	2	2	.34669E 03	.21101E 03	-.39183E 02	.50260E 01 MIN
	2	1	-.23097E 03	-.20416E 03	-.18794E 02	-.42670E 01 MAX
	2	2	-.31531E 03	-.27520E 03	-.20086E 02	-.57883E 01 MIN
	3	1	.39951E 03	.32935E 03	-.19914E 02	.71447E 01 MAX
	3	2	.31519E 03	.25832E 03	-.21206E 02	.56236E 01 MIN
	3	1	-.93393E 02	-.21014E 03	-.33288E 01	-.36175E 01 MAX
	3	2	-.12624E 03	-.22623E 03	-.36379E 01	-.42159E 01 MIN
	3	1	.14936E 03	.23480E 03	-.35622E 01	.45967E 01 MAX
	3	2	.11666E 03	.21069E 03	-.38713E 01	.39962E 01 MIN

FRAME NO. 4

FOR LOAD SYSTEM 1

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.55444E 03	-.55440E 03	.15440E 02	.15440E 02
2	1	.53000E 03	-.52956E 03	.15682E 02	.15678E 02
3	1	.12989E 03	-.13039E 03	.34178E 01	.34222E 01

COLUMN END ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	1	-.32347E 03	-.17210E 03	-.36400E 02	-.44646E 01
	2	2	.32349E 03	.17208E 03	-.36400E 02	.44647E 01
2	1	1	-.31521E 03	-.26679E 03	-.20000E 02	-.57060E 01
	2	2	.31528E 03	.26673E 03	-.20000E 02	.57060E 01
3	1	1	-.12220E 03	-.22272E 03	-.35979E 01	-.41062E 01
	2	2	.12067E 03	.22220E 03	-.36022E 01	.41062E 01

STORY LATERAL DISPLT

ST.LEVEL	DISPLT
1	-.35309E-05
2	-.39708E-04
3	-.26776E-03

FOR LOAD SYSTEM 2

BEAM END ACTIONS

STORY	BAY-NO	LEFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	-.11903E 03	-.11903E 03	-.10627E 01	.10627E 01
2	1	-.67655E 02	-.67659E 02	-.59348E 00	.59348E 00
3	1	-.17124E 02	-.17206E 02	-.15057E 00	.15057E 00

COLUMN END ACTIONS

STORY COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1 1	.61163E 02	.15811E 03	.18068E 01	.19755E 01
2 2	.61168E 02	.15811E 03	-.18068E 01	.19755E 01
2 1	.56122E 02	.40095E 02	.74411E 00	.94330E 00
2 2	.56113E 02	.40091E 02	-.74407E 00	.94318E 00
3 1	.16304E 02	.43874E 01	.15061E 00	.24632E 00
2 2	.16254E 02	.44001E 01	-.15057E 00	.24736E 00

STORY LATERAL DISPLT

ST. LEVEL	DISPLT
1	.31830E-01
2	.72365E-01
3	.87497E-01

FOR LOAD SYSTEM 3

BEAM END ACTIONS

STORY BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE	
1 1	.88801E 02	.88800E 02	.79286E 00	.79286E 00	MAX
2 1	-.88801E 02	-.88800E 02	-.79286E 00	-.79286E 00	MIN
3 1	.67798E 02	.67804E 02	.59475E 00	.59475E 00	MAX
2 1	-.67798E 02	-.67804E 02	-.59475E 00	-.59475E 00	MIN
3 1	.21156E 02	.21255E 02	.18601E 00	.18601E 00	MAX
2 1	-.21156E 02	-.21255E 02	-.18601E 00	-.18601E 00	MIN

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.32449E 02	.10433E 03	.15645E 01	.12297E 01 MAX
	2	-.32449E 02	-.10433E 03	-.15645E 01	-.12297E 01 MIN
		.32454E 02	.10433E 03	.15645E 01	.12298E 01 MAX
		-.32454E 02	-.10433E 03	-.15645E 01	-.12298E 01 MIN
2	1	.50522E 02	.44678E 02	.77897E 00	.93238E 00 MAX
	2	-.50522E 02	-.44678E 02	-.77897E 00	-.93238E 00 MIN
		.50510E 02	.44673E 02	.77894E 00	.93221E 00 MAX
		-.50510E 02	-.44673E 02	-.77894E 00	-.93221E 00 MIN
3	1	.19773E 02	.10711E 02	.18605E 00	.35893E 00 MAX
	2	-.19773E 02	-.10711E 02	-.18605E 00	-.35893E 00 MIN
		.19684E 02	.10726E 02	.18601E 00	.36017E 00 MAX
		-.19684E 02	-.10726E 02	-.18601E 00	-.36017E 00 MIN

FOR LOAD SYSTEM 4

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.53621E 03	-.57263E 03	.15277E 02	.16871E 02 MAX
		.39413E 03	-.71471E 03	.14009E 02	.15603E 02 MIN
2	1	.53349E 03	-.52000E 03	.15713E 02	.16599E 02 MAX
		.42502E 03	-.03455E 03	.14761E 02	.15647E 02 MIN
3	1	.13390E 03	-.12629E 03	.34537E 01	.30039E 01 MAX
		.10013E 03	-.16030E 03	.31561E 01	.33003E 01 MIN

 COLUMN END ACTIONS

STORY	COL.	LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	1	-.25164E 03	.29943E 02	-.33793E 02	-.19993E 01 MAX
		2	-.30356E 03	-.13698E 03	-.36296E 02	-.39668E 01 MIN
	2	1	.39533E 03	.37413E 03	-.36503E 02	.69301E 01 MAX
		2	.34341E 03	.20721E 03	-.39007E 02	.49625E 01 MIN
2	1	1	-.23271E 03	-.20098E 03	-.18819E 02	-.42526E 01 MAX
		2	-.31354E 03	-.27246E 03	-.20065E 02	-.57444E 01 MIN
	2	1	.39777E 03	.33253E 03	-.19935E 02	.71591E 01 MAX
		2	.31696E 03	.26106E 03	-.21181E 02	.56676E 01 MIN
3	1	1	-.94152E 02	-.21086E 03	-.33361E 01	-.36343E 01 MAX
		2	-.12579E 03	-.22800E 03	-.36338E 01	-.42086E 01 MIN
	2	1	.14861E 03	.23408E 03	-.35664E 01	.45799E 01 MAX
		2	.11711E 03	.21692E 03	-.38640E 01	.40036E 01 MIN

 FRAME NO. 5

 FOR LOAD SYSTEM 1

 BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.27781E 03	-.27792E 03	.79795E 01	.79805E 01
2	1	.23906E 03	-.23931E 03	.79789E 01	.79811E 01

 COLUMN END ACTIONS

STORY	COL.LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	-.89867E 02	-.47898E 02	-.16798E 02	-.12192E 01
	2	.89633E 02	.47522E 02	-.16802E 02	.12192E 01
2	1	-.25805E 03	-.19840E 03	-.83989E 01	-.43061E 01
	2	.25616E 03	.19599E 03	-.84011E 01	.43061E 01

 STORY LATERAL DISPLT

ST.LEVEL	DISPLT
1	-.17303E-03
2	-.11406E-02

 FOR LOAD SYSTEM 2

 BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	-.81196E 02	-.81376E 02	-.71303E 00	.71303E 00
2	1	-.45493E 02	-.45662E 02	-.39980E 00	.39980E 00

 COLUMN END ACTIONS

STORY	COL.LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.46096E 02	.89600E 02	.11128E 01	.12085E 01
	2	.46269E 02	.89932E 02	-.11128E 01	.12167E 01
2	1	.43332E 02	.25759E 02	.39979E 00	.65100E 00
	2	.43164E 02	.25408E 02	-.39981E 00	.65306E 00

STORY LATERAL DISPLT

ST.LEVEL DISPLT

1	.44176E-01
2	.88410E-01

FOR LOAD SYSTEM 3

BEAM END ACTIONS

STORY	BAY NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.66153E 02	.66295E 02	.58091E 00	.58091E 00 MAX
2	1	-.66153E 02	-.66295E 02	-.58091E 00	-.58091E 00 MIN
		.47100E 02	.47313E 02	.41409E 00	.41409E 00 MAX
		-.47100E 02	-.47313E 02	-.41409E 00	-.41409E 00 MIN

COLUMN END ACTIONS

STORY	COL.LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	.25302E 02	.60049E 02	.99292E 00	.75371E 00 MAX
	2	-.25302E 02	-.60049E 02	-.99392E 00	-.75371E 00 MIN
		.24878E 02	.59999E 02	.99390E 00	.75273E 00 MAX
2	1	-.24878E 02	-.59999E 02	-.99390E 00	-.75273E 00 MIN
		.44340E 02	.35231E 02	.41409E 00	.74997E 00 MAX
	2	-.44340E 02	-.35231E 02	-.41409E 00	-.74997E 00 MIN
		.44150E 02	.35070E 02	.41410E 00	.75373E 00 MAX
		-.44150E 02	-.35070E 02	-.41410E 00	-.75373E 00 MIN

FOR LOAD SYSTEM 4

BEAM END ACTIONS

STORY	BAY-NO	LFT-MOMENT	RT-MOMENT	LFT-FORCE	RT-FORCE
1	1	.26983E 03	.28591E 03	.79095E 01	.89000E 01 MAX
		.16399E 03	-.39199E 03	.69800E 01	.80505E 01 MIN
2	1	.24262E 03	-.23570E 03	.80103E 01	.86122E 01 MAX
		.16726E 03	-.31140E 03	.73478E 01	.79496E 01 MIN

COLUMN END ACTIONS

STORY	COL. LN	TOP-MOMENT	BOT-MOMENT	AXIAL-FORCE	SHEAR-FORCE
1	1	-.34603E 02	.67542E 02	-.15169E 02	.29020E 00 MAX
	2	-.75086E 02	-.28537E 02	-.16759E 02	-.91573E 00 MIN
		.14424E 03	.16297E 03	-.16841E 02	.27293E 01 MAX
2	1	.10443E 03	.66972E 02	-.18431E 02	.15250E 01 MIN
		-.19008E 03	-.15089E 03	-.77678E 01	-.32173E 01 MAX
	2	-.26103E 03	-.20726E 03	-.84304E 01	-.44173E 01 MIN
		.32385E 03	.24310E 03	-.83697E 01	.53989E 01 MAX
		.25321E 03	.18699E 03	-.90322E 01	.41930E 01 MIN

*** SIGN USED ***
 MOM/TORNE: CNTR.CLCK +VE ; FORCE/DISP: UP +VE, RHT +VE
 *** NOTE :
 MAX = OTHER FORCES + SEISMIC ROOT SUM SQ. FORCES
 MIN = OTHER FORCES - SEISMIC ROOT SUM SQ. FORCES

APPENDIX B

CHAPTER B1

FORTRAN LISTING OF PROGRAM - AIDS

```

0001 C
0002 C
0003 C
0004 C
0005 C
0006 C
0007 C
0008 C
0009 C
0010 C
0011 C
0012 C
0013 C
0014 C
0015 C

```

ASSIGNMENT OF LOGICAL INPUT AND OUTPUT DEVICES

CONVERSATIONAL INPUT	...	...	...	...	...	...	...	...	1
CONVERSATIONAL OUTPUT	...	...	...	...	...	...	...	...	2
RID1	...	OUTPUT OF GENERAL STRUCTURE DATA							...
		AND DATA FOR DYNAMIC ANALYSIS	...	...	...	...	...	...	3
RID2	...	OUTPUT OF FRAME PROPERTIES DATA	...	...	...	...	...	...	4
RID3	...	OUTPUT OF GRAVITY AND LATERAL LOAD DATA	...	...	...	...	...	...	5
DUPW	...	NEW FILE CREATING COPY OF INPUT DATA	...	...	...	...	...	...	6
DUPR	...	PREVIOUSLY CREATED INPUT DATA FILE	...	...	...	...	...	...	7
SCRA	...	SCRATCH FILE REQD. FOR SECOND PART	...	...	...	...	...	...	8

```

COMMON/BLOK1/IR,IW,INX,ITOT,JAX(80),IAL(48),IFIX(6),AFLX(6),
+INDX(6),JBND(8,6),IBND(4,6),FBND(4,6)
COMMON/MAIN/NIF(10),NST(10,5),NBAY(10,5),NBS(10,5),NSC(8),
+NFIG(10),NFG,NFT,NNS,MBA,H(30),M(30),IFD(30),DDN(30),IAX(40),
+IW1,IW3,IWF,IWS,IY,IT,IL,IBL,IO,IDR,IDW,JTOT
DIMENSION B(8),EB(30,8),AR(30,8),GB(30,8),BK(30,8),BI(30,8),
+DB(30,8),DBX(30,8),IBC(30),ICC(9)
DIMENSION EC(30,8),AC(30,8),GC(30,8),CK(30,8),CI(30,8),DC(30,8),
+DCX(30,8)
EQUIVALENCE (EB(1,1),EC(1,1)),(AB(1,1),AC(1,1)),(GB(1,1),GC(1,1)),
+(BK(1,1),CK(1,1)),(BI(1,1),CI(1,1)),(DB(1,1),DC(1,1)),
+(DBX(1,1),DCX(1,1))

```

0001 C
0002 C
0003 C
0004 C
0005 C
0006 C
0007 C
0008 C
0009 C
0010 C
0011 C
0012 C
0013 C
0014 C
0015 C
0016 C
0017 C
0018 C
0019 C
0020 C
0021 C
0022 C
0023 C
0024 C
0025 C
0026 C
0027 C
0028 C
0029 C
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042 C

DATA IB, IC, IN, IT, IY, IL, IHB, IHC, IHN, IHT, IHY, IHL/
DATA IBL, IO, JR, IH, IHO, IHR/

INITIALIZE READ AND WRITE PARAMETERS

IR=1
IW=2
IW1=3
IW2=4
IW3=5
IWS=6
IWF=8
INX=40
ITOT=1
JTOT=0

INITIALIZE DATA BOUND PARAMETERS

D0 50 I=1,4
D0 50 J=1,6
JBND(I,J)=0
JBND(I+4,J)=0
FBND(I,J)=0
50 CONTINUE

START RAPID DATA INPUT SYSTEM

INITIATE AN INPUT DATA FILE FOR FUTURE USE

WRITE(IW,1000)
CALL REED
IDW=IAL(1)
IF(IDW.EQ.IY) WRITE(IW,1100)

ESTABLISH MANNER OF DATA INPUT

WRITE(IW,1200)
CALL REED
IDR=IAL(1)
IF(IDR.NE.IT) IR=7

0043
0044
0045 C
0046 C
0047 C
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058 C
0059 C
0060 C
0061
0062
0063
0064
0065
0066
0067 C
0068 C
0069 C
0070 C
0071 C
0072
0073
0074
0075
0076 C
0077 C
0078 C
0079
0080
0081
0082

```

0083 C
0084 C
0085 C
0086 IF(IDR.EQ.IT) WRITE(IW,1405)
0087 READ(IR,7050) (IAX(KK),KK=1,40)
0088 IF(IDW.EQ.IY) WRITE(IWF,7050) (IAX(KK),KK=1,40)
0089 JTOT=JTOT+1
0090 ISTB=IBL
0091 IYS=IBL
0092 IF(IDR.EQ.IT) WRITE(IW,1440)
0093 CALL REED
0094 ISTB=IAL(1)
0095 IF(IDW.EQ.IY) WRITE(IWF,8000) ISTB
0096 JTOT=JTOT+1
0097 IF(IDR.EQ.IT) WRITE(IW,1450)
0098 CALL REED
0099 IYS=IAL(1)
0100 IF(IDW.EQ.IY) WRITE(IWF,8010) IYS
0101 JTOT=JTOT+1
0102 IF(IDR.EQ.IT.AND.IYS.NE.IY) WRITE(IW,1460)

0103 C
0104 C
0105 C
0106 IF(IDR.EQ.IT) WRITE(IW,1500)
0107 CALL REED
0108 NFI=IFIX(1)
0109 IF(IDW.EQ.IY) WRITE(IWF,8020) NFI
0110 JTOT=JTOT+1
0111 IF(IDR.EQ.IT) WRITE(IW,1600)
0112 CALL REED
0113 NFG=IFIX(1)
0114 IF(IDW.EQ.IY) WRITE(IWF,8030) NFG
0115 JTOT=JTOT+1
0116 IF(IDR.EQ.IT) WRITE(IW,1700)
0117 CALL REED
0118 NNS=IFIX(1)
0119 IF(IDW.EQ.IY) WRITE(IWF,8040) NNS
0120 JTOT=JTOT+1
0121 IF(IDR.EQ.IT) WRITE(IW,1750)
0122 CALL REED

```

STRUCTURE HISTORY :

```

0123 MBA=IFIX(1)
0124 IF(IDW.EQ.IY) WRITE(IWF,8045) MBA
0125 JTOT=JTOT+1
0126 C
0127 C READ STORY HEIGHTS
0128 C
0129 IF(IDR.EQ.IT) WRITE(IW,2900)
0130 IIT=IBL
0131 IIO=IBL
0132 INC=0
0133 DO 70 I=1,NNS
0134 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 60
0135 IF(I.GT.INC) GO TO 60
0136 H(I)=H(I-1)
0137 GO TO 70
0138 JJ=I-1
0139 IF(IDR.EQ.IT) WRITE(IW,2950) JJ,I
0140 ITOT=3
0141 CALL REED
0142 IF(IDW.EQ.IY) WRITE(IWF,8050) (JAX(KK),KK=1,24),JJ,I
0143 JTOT=JTOT+1
0144 H(I)=AFLX(1)
0145 IIT=IAL(9)
0146 IIO=IAL(10)
0147 INC=IFIX(3)
0148 70 CONTINUE
0149 C
0150 C FRAME INFORMATION
0151 C
0152 IF(IDR.EQ.IT) WRITE(IW,1900)
0153 K=1
0154 MP=1
0155 DO 80 II=1,NFT
0156 IF(IDR.EQ.IT) WRITE(IW,1950) II
0157 CALL REED
0158 MM=IFIX(1)
0159 M(II)=MM
0160 IF(IDW.EQ.IY) WRITE(IWF,8060) MM,II
0161 JTOT=JTOT+1
0162 IF(IDR.EQ.IT) WRITE(IW,2100)

```

```

0163 CALL REED
0164 IFD(II)=IAL(1)
0165 IF(IDW.EQ.IY) WRITE(IWF,8070) IFD(II)
0166 JTOT=JTOT+1
0167 IF(IDR.EQ.IT) WRITE(IW,2200)
0168 CALL REED
0169 DDN(II)=AFLX(1)
0170 IF(IDW.EQ.IY) WRITE(IWF,8080) (JAX(KK),KK=1,20)
0171 JTOT=JTOT+1
0172 IF(MM.LE.MP) GO TO 80
0173 NFIG(MP)=II-K
0174 K=II
0175 MP=MM
0176 80 CONTINUE
0177 NFIG(MM)=NFT-K+1
0178 C
0179 IF(IDR.EQ.IT) WRITE(IW,2400)
0180 DO 400 IJ=1,NFG
0181 IF(IDR.EQ.IT) WRITE(IW,2500) IJ
0182 ITOT=1
0183 CALL REED
0184 NT=IFIX(1)
0185 NTF(IJ)=NT
0186 IF(IDW.EQ.IY) WRITE(IWF,8100) NT,IJ
0187 JTOT=JTOT+1
0188 NSS=0
0189 DO 160 IK=1,NT
0190 IF(IK.NE.1) GO TO 110
0191 IF(IDR.EQ.IT) WRITE(IW,2600)
0192 GO TO 120
0193 IF(IDR.EQ.IT) WRITE(IW,2650) IK
0194 110 CALL REED
0195 NST(IJ,IK)=IFIX(1)
0196 IF(IDW.EQ.IY) WRITE(IWF,8110) NST(IJ,IK),IK
0197 JTOT=JTOT+1
0198 IF(IDR.EQ.IT) WRITE(IW,2700)
0199 CALL REED
0200 NBAY(IJ,IK)=IFIX(1)
0201 IF(IDW.EQ.IY) WRITE(IWF,8120) NBAY(IJ,IK)
0202 JTOT=JTOT+1

```

```

0203 IF(IK.EQ.1) GO TO 122
0204 KK=IK-1
0205 IF(IDR.EQ.IT) WRITE(IW,2750) KK,IK
0206 CALL REED
0207 NBS(IJ,KK)=IFIX(1)
0208 IF(IDW.EQ.IY) WRITE(IWF,8130) NBS(IJ,KK),KK,IK
0209 JTOT=JTOT+1
0210 C
0211 C READ BEAM SPANS
0212 C
0213 IF(IK.NE.1) GO TO 140
0214 IF(IDR.EQ.IT) WRITE(IW,2800)
0215 NB=NBAY(IJ,1)
0216 IIT=IBL
0217 IIO=IBL
0218 INC=0
0219 IF(NB.EQ.0) GO TO 130
0220 DO 130 I=1,NB
0221 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 125
0222 IF(I.GT.INC) GO TO 125
0223 B(I)=B(I-1)
0224 GO TO 130
0225 125 IF(IDR.EQ.IT) WRITE(IW,2850) I
0226 ITOI=3
0227 CALL REED
0228 IF(IDW.EQ.IY) WRITE(IWF,8140) (JAX(KK),KK=1,24),I
0229 JTOT=JTOT+1
0230 B(I)=AFLX(1)
0231 IIT=IAL(9)
0232 IIO=IAL(10)
0233 INC=IFIX(3)
0234 130 CONTINUE
0235 140 NSS=NSS+NST(IJ,IK)
0236 160 CONTINUE
0237 C
0238 C
0239 WRITE(IW2,7000) NT,NSS,NFIG(IJ)
0240 WRITE(IW2,7000) (NST(IJ,IK),IK=1,NT)
0241 WRITE(IW2,7000) (NBAY(IJ,IK),IK=1,NT)
0242 IF(NT.EQ.1) GO TO 161

```



```

0243 NT1=NT-1
0244 WRITE(IW2,7000) (NBS(IJ,IK),IK=1,NT1)
0245 161 WRITE(IW2,7030) (B(IK),IK=1,NB)
0246 C
0247 C
0248 C
0249 READ BEAM CROSS SECTION PROPERTIES
0250 IF(IDR.EQ.IT) WRITE(IW,3000)
0251 IF(IDW.EQ.IY) WRITE(IWF,8150)
0252 JT=JT+1
0253 IF(IDR.NE.IT) READ(IR,7040)
0254 IS=0
0255 NN=0
0256 DO 190 I=1,NT
0257 NSS=NST(IJ,I)
0258 NB=NBAY(IJ,I)
0259 IF(I.NE.1) GO TO 162
0260 NSB=0
0261 GO TO 163
0262 162 NSB=NSB+NB
0263 163 NBS1=NSB+1
0264 NBS2=NSB+NB
0265 NN=NN+NSS
0266 ILL=IBL
0267 I00=IBL
0268 IMB=0
0269 IND=0
0270 DO 180 J=1,NSS
0271 IS=IS+1
0272 IIT=IBL
0273 IIO=IBL
0274 INC=0
0275 IF(ILL.NE.IL) GO TO 170
0276 IBC(IS)=IBC(IMB)
0277 IF(IBC(IS).NE.JR) GO TO 165
0278 DO 164 II=NBS1,NBS2
0279 DB(IS,II)=DB(IMB,II)
0280 DBX(IS,II)=DBX(IMB,II)
0281 164 CONTINUE
0282 GO TO 166
0283 165 DO 166 II=NBS1,NBS2

```

```

0283 BI(1S,11)=BI(1MB,11)
0284 AB(1S,11)=AB(1MB,11)
0285 DB(1S,11)=DB(1MB,11)
0286 188 CONTINUE
0287 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 178
0288 IF(1S.EQ.IND) GO TO 178
0289 GO TO 180
0290 170 IF(IDR.EQ.IT) WRITE(1W,3050) IS
0291 CALL REED
0292 IIC=IAL(1)
0293 IBC(1S)=IIC
0294 IF(IDW.EQ.IY) WRITE(1WF,8155) IIC,1S
0295 JTOT=JTOT+1
0296 IF(IIC.NE.JR) GO TO 175
0297 IF(IDR.EQ.IT) WRITE(1W,3070)
0298 DO 174 JK=NBS1,NBS2
0299 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 172
0300 IF(JK.GT.INC) GO TO 172
0301 DB(1S,JK)=DB(1S,JK-1)
0302 DBX(1S,JK)=DBX(1S,JK-1)
0303 GO TO 174
0304 172 IF(IDR.EQ.IT) WRITE(1W,2850) JK
0305 ITOT=4
0306 CALL REED
0307 IF(IDW.EQ.IY) WRITE(1WF,8160) (JAX(KK),KK=1,40),JK
0308 JTOT=JTOT+1
0309 DBX(1S,JK)=AFLX(1)
0310 DB(1S,JK)=AFLX(2)
0311 IIT=IAL(17)
0312 IIO=IAL(18)
0313 INC=IFIX(4)
0314 174 CONTINUE
0315 GO TO 178
0316 175 IF(IDR.EQ.IT) WRITE(1W,3080)
0317 DO 177 JK=NBS1,NBS2
0318 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 176
0319 IF(JK.GT.INC) GO TO 176
0320 BI(1S,JK)=BI(1S,JK-1)
0321 AB(1S,JK)=AB(1S,JK-1)
0322 DB(1S,JK)=DB(1S,JK-1)

```

```

0323 GO TO 177
0324 176 IF(IDR.EQ.IT) WRITE(IW,2850) JK
0325 ITOT=5
0326 CALL REED
0327 IF(IDW.EQ.IY) WRITE(IWF,8160) (JAX(KK),KK=1,40),JK
0328 JTOT=JTOT+1
0329 BI(IS,JK)=AFLX(1)
0330 AB(IS,JK)=AFLX(2)
0331 DB(IS,JK)=AFLX(3)
0332 IIT=IAL(25)
0333 IIO=IAL(26)
0334 INC=IFIX(5)
0335 177 CONTINUE
0336 178 IF(IS.GE.NN) GO TO 180
0337 179 IF(IDR.EQ.IT) WRITE(IW,3200) IS,NN
0338 ITOT=4
0339 CALL REED
0340 IF(IDW.EQ.IY) WRITE(IWF,8170) (JAX(KK),KK=1,32),IS
0341 JTOT=JTOT+1
0342 ILL=IAL(1)
0343 IMB=IFIX(2)
0344 ITI=IAL(17)
0345 IOO=IAL(18)
0346 IND=IFIX(4)
0347 180 CONTINUE
0348 190 CONTINUE
0349 C
0350 C
0351 NSI=0
0352 NI=0
0353 DO 200 J=1,NT
0354 NB=NBAY(IJ,J)+1
0355 NSI=NSI+NST(IJ,J)
0356 NJ=NI
0357 IF(J.EQ.NT) GO TO 192
0358 NI=NI+NBS(IJ,J)
0359 NI1=NI+NBAY(IJ,J+1)+1
0360 192 DO 196 I=1,NB
0361 IK=I+NJ
0362 IF(J.EQ.NT) GO TO 194

```

```

0363 IF(IK.GT.NI.AND.IK.LE.NI) GO TO 196
0364 NSC(IK)=NSI
0365 GO TO 196
0366 194 NSC(IK)=NSI
0367 196 CONTINUE
0368 200 CONTINUE
0369 C
0370 C
0371 C
0372 READ 'MODULII OF ELASTICITY AND MODULII OF RIGIDITY OF BEAMS
IYD=IBL
0373 IF(IDR.EQ.IT) WRITE(IW,3600)
0374 CALL REED
0375 IYD=IA(1)
0376 IF(IDW.EQ.IY) WRITE(IWF,8190) IYD
0377 JTOT=JTOT+1
0378 IF(IYD.NE.IY) GO TO 210
0379 IF(IDR.EQ.IT) WRITE(IW,3650)
0380 ITOT=2
0381 CALL REED
0382 IF(IDW.EQ.IY) WRITE(IWF,8200) (JAX(KK),KK=1,20)
0383 JTOT=JTOT+1
0384 EY=AFLX(1)
0385 EG=AFLX(2)
0386 IS=0
0387 NJ=0
0388 DO 206 I=1,NT
0389 NSS=NST(IJ,I)
0390 NB=NBAY(IJ,I)
0391 DO 204 J=1,NSS
0392 IS=IS+1
0393 DO 203 K=1,NB
0394 NI=NJ+K
0395 EB(JS,NI)=EY
0396 GB(JS,NI)=EG
0397 203 CONTINUE
0398 204 CONTINUE
0399 IF(I.EQ.NT) GO TO 208
0400 NJ=NJ+NBS(IJ,I)
0401 206 CONTINUE

```

```

0402 GO TO 230
0403 210 IF(IDR.EQ.IT) WRITE(IW,3650)
0404 NN=0
0405 IS=0
0406 DO 228 I=1,NT
0407 NSS=NSI(IJ,I)
0408 NN=NN+NSS
0409 NB=NBY(IJ,I)
0410 IF(I.NE.1) GO TO 212
0411 NSB=0
0412 GO TO 214
0413 212 NSB=NSB+NBS(IJ,I-1)
0414 214 NBS1=NSB+1
0415 NBS2=NSB+NB
0416 ILL=IBL
0417 IIT=IBL
0418 I(X)=IBL
0419 IMB=0
0420 IND=0
0421 DO 226 J=1,NSS
0422 IS=IS+1
0423 IIT=IBL
0424 IIO=IBL
0425 INC=0
0426 IF(ILL.NE.IL) GO TO 218
0427 DO 216 II=NBS1,NBS2
0428 EB(IS,II)=EB(IMB,II)
0429 GB(IS,II)=GB(IMB,II)
0430 216 CONTINUE
0431 IF(IIT.NE.IT.OR.IOO.NE.IO) GO TO 224
0432 IF(IS.EQ.IND) GO TO 224
0433 GO TO 226
0434 218 IF(IDR.EQ.IT) WRITE(IW,4050) IS
0435 DO 223 K=NBS1,NBS2
0436 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 220
0437 IF(K.GT.INC) GO TO 22Q
0438 EB(IS,K)=EB(IS,K-1)
0439 GB(IS,K)=GB(IS,K-1)
0440 GO TO 223
0441 220 IF(IDR.EQ.IT) WRITE(IW,2850) K
0442 ITOT=4

```

```

0443 CALL REED
0444 IF(IDW.EQ.IY) WRITE(IWF,8160) (JAX(KK),KK=1,40),K
0445 JTOT=JTOT+1
0446 EB(IS,K)=AFLX(1)
0447 GB(IS,K)=AFLX(2)
0448 IIT=IAL(17)
0449 IIO=IAL(18)
0450 INC=IFIX(4)
0451 223 CONTINUE
0452 224 IF(IS.GE.NN) GO TO 226
0453 225 IF(IDR.EQ.IT) WRITE(IW,3900) IS,NN
0454 ITOT=4
0455 CALL REED
0456 IF(IDW.EQ.IY) WRITE(IWF,8170) (JAX(KK),KK=1,32),IS
0457 JTOT=JTOT+1
0458 ILL=IAL(1)
0459 IMB=IFIX(2)
0460 IIT=IAL(17)
0461 IOO=IAL(18)
0462 IND=IFIX(4)
0463 226 CONTINUE
0464 228 CONTINUE
0465 C
0466 C
0467 C
0468 READ SHEAR COEFFICIENTS FOR BEAMS
0469 230 IF(IYS.NE.IY) GO TO 250
0470 IF(IDR.NE.IT) GO TO 231
0471 WRITE(IW,4000)
0472 GO TO 232
0473 231 READ(IR,7040)
0474 232 IF(IDW.EQ.IY) WRITE(IWF,8205)
0475 JTOT=JTOT+1
0476 NN=0
0477 IS=0
0478 DO 248 I=1,NT
0479 NSS=NST(IJ,I)
0480 NN=NN+NSS
0481 NB=NBAY(IJ,I)
0482 IF(I.NE.1) GO TO 233
    NSB=0

```

```

0483      GO TO 234
0484      233 NSB=NSB+NBS(IJ,I-1)
0485      234 NBS1=NSB+1
0486      NBS2=NSB+NB
0487      ILL=IBL
0488      IIT=IBL
0489      IOX=IBL
0490      IMB=0
0491      IND=0
0492      DO 246 J=1,NSS
0493      IS=IS+1
0494      IIT=IBL
0495      IO=IBL
0496      INC=0
0497      IF(ILL.NE.IL) GO TO 238
0498      DO 236 II=NBS1,NBS2
0499      BK(IS,II)=BK(IMB,II)
0500      236 CONTINUE
0501      IF(IIT.NE.IT.OR.IOX.NE.IO) GO TO 244
0502      IF(IS.EQ.IND) GO TO 244
0503      GO TO 246
0504      238 IF(IDR.EQ.IT) WRITE(IW,4050) IS
0505      DO 243 K=NBS1,NBS2
0506      IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 240
0507      IF(K.GT.INC) GO TO 240
0508      BK(IS,K)=BK(IS,K-1)
0509      GO TO 243
0510      240 IF(IDR.EQ.IT) WRITE(IW,4100) K
0511      ITOT=3
0512      CALL REED
0513      IF(IDW.EQ.IY) WRITE(IMF,8160) (JAX(KK),KK=1,40),K
0514      JTOT=JTOT+1
0515      BK(IS,K)=AFLX(1)
0516      IIT=IAL(9)
0517      IIO=IAL(10)
0518      INC=IFIX(3)
0519      243 CONTINUE
0520      244 IF(IS.GE.NN) GO TO 246
0521      245 IF(IDR.EQ.IT) WRITE(IW,4200) IS,NN
0522      ITOT=4

```

```

0523 CALL REED
0524 IF(IDW.EQ.IY) WRITE(IWF,8170) (JAX(KK),KK=1,32),IS
0525 JTOT=JTOT+1
0526 ILL=IAL(1)
0527 IMB=IFIX(2)
0528 ITT=IAL(17)
0529 IX=IAL(18)
0530 IND=IFIX(4)
0531 CONTINUE
0532 CONTINUE
0533 CONTINUE
0534 C
0535 C
0536 C
0537 WRITE BEAM PROPERTIES
0538 IS=0
0539 NJ=0
0540 NSIB=1
0541 IF(ISTB.NE.IY) NSTB=0
0542 WRITE(IW2,7000) NSIB
0543 C
0544 DO 254 I=1,NT
0545 NSS=NST(IJ,I)
0546 NB=NBAY(IJ,I)
0547 DO 253 J=1,NSS
0548 IS=IS+1
0549 WRITE(IW2,7040) IBC(IS)
0550 DO 252 K=1,NB
0551 NI=NJ+K
0552 IF(IYS.NE.IY) BK(IS,NI)=0.0
0553 IF(IBC(IS).NE.JR) GO TO 251
0554 WRITE(IW2,7010) EB(IS,NI),GB(IS,NI),BK(IS,NI),DB(IS,NI)
0555 GO TO 252
0556 251 WRITE(IW2,7010) EB(IS,NI),GB(IS,NI),BK(IS,NI),DB(IS,NI),
0557 +BI(IS,NI)
0558 252 CONTINUE
0559 253 CONTINUE
0560 NJ=NJ+NBS(IJ,I)
0561 CONTINUE
0562 C
0563 C
0564 C
0565 C
0566 C
0567 C
0568 C
0569 C
0570 C
0571 C
0572 C
0573 C
0574 C
0575 C
0576 C
0577 C
0578 C
0579 C
0580 C
0581 C
0582 C
0583 C
0584 C
0585 C
0586 C
0587 C
0588 C
0589 C
0590 C
0591 C
0592 C
0593 C
0594 C
0595 C
0596 C
0597 C
0598 C
0599 C
0600 C
0601 C
0602 C
0603 C
0604 C
0605 C
0606 C
0607 C
0608 C
0609 C
0610 C
0611 C
0612 C
0613 C
0614 C
0615 C
0616 C
0617 C
0618 C
0619 C
0620 C
0621 C
0622 C
0623 C
0624 C
0625 C
0626 C
0627 C
0628 C
0629 C
0630 C
0631 C
0632 C
0633 C
0634 C
0635 C
0636 C
0637 C
0638 C
0639 C
0640 C
0641 C
0642 C
0643 C
0644 C
0645 C
0646 C
0647 C
0648 C
0649 C
0650 C
0651 C
0652 C
0653 C
0654 C
0655 C
0656 C
0657 C
0658 C
0659 C
0660 C
0661 C
0662 C
0663 C
0664 C
0665 C
0666 C
0667 C
0668 C
0669 C
0670 C
0671 C
0672 C
0673 C
0674 C
0675 C
0676 C
0677 C
0678 C
0679 C
0680 C
0681 C
0682 C
0683 C
0684 C
0685 C
0686 C
0687 C
0688 C
0689 C
0690 C
0691 C
0692 C
0693 C
0694 C
0695 C
0696 C
0697 C
0698 C
0699 C
0700 C
0701 C
0702 C
0703 C
0704 C
0705 C
0706 C
0707 C
0708 C
0709 C
0710 C
0711 C
0712 C
0713 C
0714 C
0715 C
0716 C
0717 C
0718 C
0719 C
0720 C
0721 C
0722 C
0723 C
0724 C
0725 C
0726 C
0727 C
0728 C
0729 C
0730 C
0731 C
0732 C
0733 C
0734 C
0735 C
0736 C
0737 C
0738 C
0739 C
0740 C
0741 C
0742 C
0743 C
0744 C
0745 C
0746 C
0747 C
0748 C
0749 C
0750 C
0751 C
0752 C
0753 C
0754 C
0755 C
0756 C
0757 C
0758 C
0759 C
0760 C
0761 C
0762 C
0763 C
0764 C
0765 C
0766 C
0767 C
0768 C
0769 C
0770 C
0771 C
0772 C
0773 C
0774 C
0775 C
0776 C
0777 C
0778 C
0779 C
0780 C
0781 C
0782 C
0783 C
0784 C
0785 C
0786 C
0787 C
0788 C
0789 C
0790 C
0791 C
0792 C
0793 C
0794 C
0795 C
0796 C
0797 C
0798 C
0799 C
0800 C
0801 C
0802 C
0803 C
0804 C
0805 C
0806 C
0807 C
0808 C
0809 C
0810 C
0811 C
0812 C
0813 C
0814 C
0815 C
0816 C
0817 C
0818 C
0819 C
0820 C
0821 C
0822 C
0823 C
0824 C
0825 C
0826 C
0827 C
0828 C
0829 C
0830 C
0831 C
0832 C
0833 C
0834 C
0835 C
0836 C
0837 C
0838 C
0839 C
0840 C
0841 C
0842 C
0843 C
0844 C
0845 C
0846 C
0847 C
0848 C
0849 C
0850 C
0851 C
0852 C
0853 C
0854 C
0855 C
0856 C
0857 C
0858 C
0859 C
0860 C
0861 C
0862 C
0863 C
0864 C
0865 C
0866 C
0867 C
0868 C
0869 C
0870 C
0871 C
0872 C
0873 C
0874 C
0875 C
0876 C
0877 C
0878 C
0879 C
0880 C
0881 C
0882 C
0883 C
0884 C
0885 C
0886 C
0887 C
0888 C
0889 C
0890 C
0891 C
0892 C
0893 C
0894 C
0895 C
0896 C
0897 C
0898 C
0899 C
0900 C
0901 C
0902 C
0903 C
0904 C
0905 C
0906 C
0907 C
0908 C
0909 C
0910 C
0911 C
0912 C
0913 C
0914 C
0915 C
0916 C
0917 C
0918 C
0919 C
0920 C
0921 C
0922 C
0923 C
0924 C
0925 C
0926 C
0927 C
0928 C
0929 C
0930 C
0931 C
0932 C
0933 C
0934 C
0935 C
0936 C
0937 C
0938 C
0939 C
0940 C
0941 C
0942 C
0943 C
0944 C
0945 C
0946 C
0947 C
0948 C
0949 C
0950 C
0951 C
0952 C
0953 C
0954 C
0955 C
0956 C
0957 C
0958 C
0959 C
0960 C
0961 C
0962 C
0963 C
0964 C
0965 C
0966 C
0967 C
0968 C
0969 C
0970 C
0971 C
0972 C
0973 C
0974 C
0975 C
0976 C
0977 C
0978 C
0979 C
0980 C
0981 C
0982 C
0983 C
0984 C
0985 C
0986 C
0987 C
0988 C
0989 C
0990 C
0991 C
0992 C
0993 C
0994 C
0995 C
0996 C
0997 C
0998 C
0999 C
1000 C

```

READ COLUMN CROSS SECTION PROPERTIES

0563 C
 0564 IF(IDR.EQ.IT) WRITE(IW,3300)
 0565 IF(IDW.EQ.IY) WRITE(IWF,8210)
 0566 JTOT=JTOT+1
 0567 IF(IDR.NE.IT) READ(IR,7040)
 0568 N9=NBAY(IJ,1)+1
 0569 ILL=IBL
 0570 IOO=IBL
 0571 IMB=0
 0572 IND=0
 0573 NNF=NSC(1)
 0574 DO 280 I=1,NB
 0575 IIT=IBL
 0576 IIO=IBL
 0577 INC=0
 0578 NN=NNF
 0579 IF(ILL.NE.IL) GO TO 259
 0580 ICC(I)=IOC(IMB)
 0581 IF(ICC(I).NE.JR) GO TO 256
 0582 DO 255 II=1,NN
 0583 DC(II,I)=DC(II,IMB)
 0584 DCX(II,I)=DCX(II,IMB)
 0585 255 CONTINUE
 0586 GO TO 258
 0587 256 DO 257 II=1,NN
 0588 CI(II,I)=CI(II,IMB)
 0589 AC(II,I)=AC(II,IMB)
 0590 DC(II,I)=DC(II,IMB)
 0591 257 CONTINUE
 0592 258 CONTINUE
 0593 IF(ITT.NE.IT.OR.IOO.NE.IO) GO TO 278
 0594 IF(1.EQ.IND) GO TO 278
 0595 GO TO 280
 0596 259 IF(IDR.EQ.IT) WRITE(IW,3350) I
 0597 CALL REED
 0598 IIC=IAL(1)
 0599 ICC(I)=IIC
 0600 IF(IDW.EQ.IY) WRITE(IWF,8220) IIC,I
 0601 JTOT=JTOT+1
 0602 IF(IIC.NE.JR) GO TO 270

```

0603 IF(IDR.EQ.IT) WRITE(IW,3360)
0604 DO 265 J=1,NN
0605 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 260
0606 IF(J.GT.INC) GO TO 260
0607 DC(J,I)=DC(J-1,I)
0608 DCX(J,I)=DCX(J-1,I)
0609 GO TO 265
0610 260 IF(IDR.EQ.IT) WRITE(IW,3400) J
0611 ITOT=4
0612 CALL REED
0613 IF(IDW.EQ.IY) WRITE(IWF,8230) (JAX(KK),KK=1,40),J
0614 JTOT=JTOT+1
0615 DCX(J,I)=AFLX(1)
0616 DC(J,I)=AFLX(2)
0617 IIT=IAL(17)
0618 IIO=IAL(18)
0619 INC=IFIX(4)
0620 265 CONTINUE
0621 GO TO 278
0622 270 IF(IDR.EQ.IT) WRITE(IW,3370)
0623 DO 277 J=1,NN
0624 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 275
0625 IF(J.GT.INC) GO TO 275
0626 CI(J,I)=CI(J-1,I)
0627 AC(J,I)=AC(J-1,I)
0628 DC(J,I)=DC(J-1,I)
0629 GO TO 277
0630 275 IF(IDR.EQ.IT) WRITE(IW,3400) J
0631 ITOT=5
0632 CALL REED
0633 IF(IDW.EQ.IY) WRITE(IWF,8230) (JAX(KK),KK=1,40),J
0634 JTOT=JTOT+1
0635 CI(J,I)=AFLX(1)
0636 AC(J,I)=AFLX(2)
0637 DC(J,I)=AFLX(3)
0638 IIT=IAL(25)
0639 IIO=IAL(26)
0640 INC=IFIX(5)
0641 277 CONTINUE
0642 278 IF(I.GE.NB) GO TO 280

```

```

0643 NNF=NSC(I+1)
0644 IF(NNF.LE.NN) GO TO 279
0645 ILL=IBL
0646 GO TO 280
0647 279 IF(IDR.EQ.IT) WRITE(IW,3500) I
0648 ITOT=4
0649 CALL REED
0650 IF(IDW.EQ.IY) WRITE(IWF,8240) (JAX(KK),KK=1,32), I
0651 JTOT=JTOT+1
0652 ILL=IAL(I)
0653 IMB=IFIX(2)
0654 ITT=IAL(17)
0655 IOO=IAL(18)
0656 IND=IFIX(4)
0657 280 CONTINUE
0658 C
0659 C
0660 C
0661 IYD=IBL
0662 IF(IDR.EQ.IT) WRITE(IW,4300)
0663 CALL REED
0664 IYD=IAL(1)
0665 IF(IDW.EQ.IY) WRITE(IWF,8260) IYD
0666 JTOT=JTOT+1
0667 IF(IYD.NE.IY) GO TO 290
0668 IF(IDR.EQ.IT) WRITE(IW,3650)
0669 ITOT=2
0670 CALL REED
0671 IF(IDW.EQ.IY) WRITE(IWF,8200) (JAX(KK),KK=1,20)
0672 JTOT=JTOT+1
0673 EY=AFLX(1)
0674 EG=AFLX(2)
0675 NB=NBAY(IJ,1)+1
0676 DO 284 I=1,NB
0677 NN=NSC(I)
0678 DO 282 J=1,NN
0679 EC(J,I)=EY
0680 GC(J,I)=EG
0681 282 CONTINUE
0682 284 CONTINUE

```

```

0683      GO TO 310
0684 290 IF(IDR.EQ.IT) WRITE(IW,3650)
0685      NB=NBAY(IJ,I)+1
0686      ILL=IBL
0687      IOO=IBL
0688      IMB=O
0689      IND=O
0690      NNF=NSC(1)
0691      DO 308 I=1,NB
0692      IIT=IBL
0693      IIO=IBL
0694      INC=O
0695      NN=NNF
0696      IF(ILL.NE.IL) GO TO 294
0697      DO 292 J=1,NN
0698      EC(J,I)=EC(J,IMB)
0699      GC(J,I)=GC(J,IMB)
0700 292 CONTINUE
0701      IF(IIT.NE.IT.OR.IOO.NE.IO) GO TO 306
0702      IF(I.EQ.IND) GO TO 306
0703      GO TO 308
0704 294 IF(IDR.EQ.IT) WRITE(IW,4750) I
0705      DO 305 J=1,NN
0706      IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 296
0707      IF(J.GT.INC) GO TO 296
0708      EC(J,I)=EC(J-1,I)
0709      GC(J,I)=GC(J-1,I)
0710      GO TO 305
0711 296 IF(IDR.EQ.IT) WRITE(IW,4050) J
0712      ITOI=4
0713      CALL REED
0714      IF(IDW.EQ.IY) WRITE(IWF,8230) (JAX(KK),KK=1,40),J
0715      JTOT=JTOT+1
0716      EC(J,I)=AFLX(1)
0717      GC(J,I)=AFLX(2)
0718      IIT=IAL(17)
0719      IIO=IAL(18)
0720      INC=IFIX(4)
0721 305 CONTINUE
0722 306 IF(I.GE.NB) GO TO 308

```

```

0723 NNF=NSC(I+1)
0724 IF(NNF.LE.NN) GO TO 307
0725 ILL=IBL
0726 GO TO 308
0727 IF(IDR.EQ.IT) WRITE(IW,4600) I
0728 ITOT=4
0729 CALL REED
0730 IF(IDW.EQ.IY) WRITE(IWF,8240) (JAX(KK),KK=1,32),I
0731 JTOT=JTOT+1
0732 ILL=IAL(1)
0733 IMB=IFIX(2)
0734 ITT=IAL(17)
0735 IOO=IAL(18)
0736 IND=IFIX(4)
0737 308 CONTINUE
0738 C
0739 C READ SHEAR COEFFICIENTS FOR COLUMNS
0740 C
0741 310 IF(IYS.NE.IY) GO TO 340
0742 IF(IDR.NE.IT) GO TO 311
0743 WRITE(IW,4700)
0744 GO TO 312
0745 311-READ(IR,7040)
0746 312 IF(IDW.EQ.IY) WRITE(IWF,8265)
0747 JTOT=JTOT+1
0748 ILL=IBL
0749 ITT=IBL
0750 IOO=IBL
0751 IMB=0
0752 IND=0
0753 NNF=NSC(1)
0754 DO 330 I=1,NB
0755 IIT=IBL
0756 IIO=IBL
0757 INC=0
0758 NN=NNF
0759 IF(ILL.NE.IL) GO TO 314
0760 DO 313 J=1,NN
0761 CK(J,I)=CK(J,IMB)
0762 313 CONTINUE

```

```

0763 IF(IIT,NE.IT,OR,I00,NE,I0) GO TO 327
0764 IF(I,EQ,IND) GO TO 327
0765 GO TO 330
0766 314 IF(IDR,EQ,IT) WRITE(IW,4750) I
0767 DO 326 J=1,NN
0768 IF(IIT,NE.IT,OR,I10,NE,I0) GO TO 329
0769 IF(J,GT,INC) GO TO 320
0770 CK(J,I)=CK(J-1,I)
0771 GO TO 326
0772 320 IF(IDR,EQ,IT) WRITE(IW,4800) J
0773 ITOT=3
0774 CALL REED
0775 IF(IDW,EQ,IY) WRITE(IWF,8230) (JAX(KK),KK=1,40),J
0776 JTOT=JTOT+1
0777 CK(J,I)=AFLX(1)
0778 IIT=IAL(9)
0779 I10=IAL(10)
0780 INC=IFIX(3)
0781 326 CONTINUE
0782 327 IF(I,GE,NB) GO TO 330
0783 NNF=NSC(I+1)
0784 IF(NNF,LE,NN) GO TO 328
0785 ILL=IBL
0786 GO TO 330
0787 328 IF(IDR,EQ,IT) WRITE(IW,4900) I
0788 ITOT=4
0789 CALL REED
0790 IF(IDW,EQ,IY) WRITE(IWF,8240) (JAX(KK),KK=1,32),I
0791 JTOT=JTOT+1
0792 ILL=IAL(1)
0793 IMB=IFIX(2)
0794 IIT=IAL(17)
0795 I00=IAL(18)
0796 IND=IFIX(4)
0797 330 CONTINUE
0798 340 CONTINUE
0799 C
0800 C WRITE COLUMN PROPERTIES DATA
0801 C
0802 NB=NBAY(IJ,1)+1

```

```
0803 DO 390 I=1,NB  
0804 NN=NSC(I)  
0805 WRITE(IW2,7O45) ICC(I),NN  
0806 DO 380 J=1,NN  
0807 IF(IYS.NE.IY) CK(J,I)=0.0  
0808 IF(ICC(I).NE.JR) GO TO 375  
0809 WRITE(IW2,7O10) EC(J,I),GC(J,I),CK(J,I),DCX(J,I),DC(J,I)  
0810 GO TO 380  
0811 375 WRITE(IW2,7O10) EC(J,I),GC(J,I),CK(J,I),DC(J,I),AC(J,I),CI(J,I)  
0812 380 CONTINUE  
0813 390 CONTINUE  
0814 C  
0815 400 CONTINUE  
0816 C  
0817 STOP  
0818 C  
0819 1000 FORMAT(//  
0820 + * * * * *  
0821 + * * * * *  
0822 + * * * * *  
0823 + * * * * * A I D S *****  
0824 + * * * * *  
0825 + * * * * *  
0826 + * * * * * READS DATA BY CONVERSATIONAL  
0827 + * * * * * MODE WITH RAPID INPUT OPTION  
0828 + * * * * * AND WRITES IN THREE FILES  
0829 + * * * * * FOR THE THREE DIMENSIONAL  
0830 + * * * * * BUILDING ANALYSIS PROGRAM.  
0831 + * * * * *  
0832 + * * * * *  
0833 + * * * * * DO YOU WISH TO CREATE A NEW DATA FILE CONTAINING//  
0834 + * * * * * A COPY OF THE PRESENT INPUT DATA FOR FUTURE USE ? Y/N//  
0835 1100 FORMAT(// THE INPUT DATA WILL BE SAVED ON//  
0836 + * * * * * A FILE NAMED - DUPLI//  
0837 1200 FORMAT(// DO YOU WISH TO INPUT DATA BY MEANS OF THE//  
0838 + * * * * * TERMINAL OR BY MEANS OF AN EXISTING DATA FILE ? T/F//)  
0839 1405 FORMAT(/// CASE TITLE (MAX. 40 CHARACTERS)//)  
0840 1440 FORMAT(// DO YOU WANT FINITE JOINT SIZES TO BE CONSIDERED ? Y/N//)  
0841 1450 FORMAT(// DO YOU WANT SHEAR DEFORMATIONS TO BE CONSIDERED ? Y/N//)  
0842 1460 FORMAT(// WHEN ASKED FOR, PUT VALUES OF//
```

```

0843  +- BEAM/COLUMN MOD. OF RIGIDITY EQUAL TO 0.0.
0844  1500 FORMAT(//
0845  +- ** USE CONSISTENT UNITS ***//
0846  +- STRUCTURE HISTORY ://
0847  +- TOTAL NO. OF FRAMES IN THE BUILDING//
0848  1600 FORMAT(// NO. OF FRAME GROUPS IN THE BUILDING//
0849  1700 FORMAT(// TOTAL NO. OF STORIES IN THE BUILDING//
0850  1750 FORMAT(// MAX. NO. OF BAYS IN THE WIDEST FRAME OF THE BUILDING//
0851  1900 FORMAT(// FRAME INFORMATION ://
0852  1950 FORMAT(// FRAME NO., I3, , ://
0853  +- FRAME GROUP NO.//
0854  2100 FORMAT(// PARALLEL TO GLOBAL X-DIRECTION OR Y-DIRECTION ? X/Y//
0855  2200 FORMAT(// PERPENDICULAR DISTANCE FROM//
0856  +- THE REFERENCE POINT TO THE FRAME//
0857  2400 FORMAT(// FRAME GROUP PROPERTIES ://
0858  2500 FORMAT(//
0859  +- -----//
0860  +- FRAME GROUP NO., I3, //
0861  +- -----//
0862  +- NO. OF TOWERS, INCLUDING THE BASE TOWER//
0863  2600 FORMAT(// IN TOWER 1 (BASE TOWER), NO. OF STORIES//
0864  2650 FORMAT(// IN TOWER, I3, , NO. OF STORIES//
0865  2700 FORMAT(// NO. OF BAYS//
0866  2750 FORMAT(// NO. OF BAYS-SETBACK AT LEFT//
0867  +- BETWEEN TOWERS, I3, , AND, I3//
0868  2800 FORMAT(// LENGTHS OF BAYS, C/C OF COLUMNS//
0869  +- STARTING FROM LEFT ://
0870  2850 FORMAT(// BAY NO., I3, , ://
0871  2900 FORMAT(// STORY HEIGHTS, C/C FLOOR BEAMS ://
0872  2950 FORMAT(// BETWEEN STORY LEVELS, I3, , AND, I3//
0873  3000 FORMAT(// BEAM CROSS SECTION PROPERTIES ://
0874  +- THIS SET OF DATA IS TO BE INPUT BEAM BY BEAM//
0875  +- STARTING AT LEFT AT EACH FLOOR AND THUS FLOOR BY FLOOR//
0876  3050 FORMAT(// STORY LEVEL, I3, , ://
0877  +- ARE ALL THE BEAMS IN THIS STORY OF//
0878  +- RECTANGULAR SECTION OR OTHER ? R/O//
0879  3070 FORMAT(// INPUT BEAM WIDTH AND THEN DEPTH SEPARATED BY A BLANK//
0880  3080 FORMAT(// INPUT BEAM MOMENT OF INERTIA, AREA AND DEPTH//
0881  +- SEPARATED BY BLANKS//
0882  3200 FORMAT(// THIS SET OF DATA IS COMPLETE UPTO STORY , I3, , //

```



```

0883  +- ARE THE BEAMS IN THE NEXT STORY(AND POSSIBLY MAX.//
0884  +- UPTO STORY',I3,') EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?//
0885  +- IF NOT---PUT NO, IF YES---LIKE WHICH ?//
0886  3300 FORMAT(// COLUMN CROSS SECTION PROPERTIES ://
0887  +- THIS SET OF DATA IS TO BE INPUT VERTICALLY COLUMN BY COLUMN IN//
0888  +- EACH COL. LINE AND THUS COL. LINE BY COL.LINE, FROM LEFT//
0889  3350 FORMAT(// COLUMN LINE',I3,'://
0890  +- ARE ALL THE COLUMNS IN THIS COLUMN LINE OF//
0891  +- RECTANGULAR SECTION OR OTHER ? R/O//
0892  3360 FORMAT(// INPUT COLUMN WIDTH ACROSS THE PLANE OF FRAME AND//
0893  +- THEN WIDTH IN THE PLANE OF FRAME, SEPARATED BY A BLANK//
0894  3370 FORMAT(// INPUT COLUMN MOM. OF INERTIA, AREA AND THEN//
0895  +- COLUMN WIDTH IN PLANE OF FRAME, SEPARATED BY BLANKS//
0896  3400 FORMAT(// STORY NO.',I3,'://
0897  3500 FORMAT(// THIS SET OF DATA IS COMPLETE UPTO COL.LINE',I3,//
0898  +- ARE THE COLUMNS IN THE NEXT COL. LINE EXACTLY//
0899  +- LIKE THOSE IN ANY PRECEDING COLUMN LINE ?//
0900  +- IF NOT---PUT NO, IF YES---LIKE WHICH ?//
0901  3600 FORMAT(// BEAM MOD. OF ELASTICITY AND MOD. OF RIGIDITY ://
0902  +- ARE THE MOD. OF ELASTICITY AND MOD. OF RIGIDITY SAME//
0903  +- FOR ALL THE BEAMS IN THIS FRAME GROUP ? Y/N//
0904  3650 FORMAT(// INPUT THE MOD. OF ELASTICITY AND THEN//
0905  +- THE MOD. OF RIGIDITY, SEPARATED BY A BLANK//
0906  3900 FORMAT(// THIS SET OF DATA IS COMPLETE UPTO STORY ',I3,'://
0907  +- ARE THE MODULII FOR BEAMS IN THE NEXT STORY(AND MAX.//
0908  +- UPTO STORY',I3,') EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?//
0909  +- IF NOT---PUT NO, IF YES---LIKE WHICH ?//
0910  4000 FORMAT(// SHEAR COEFFICIENTS FOR BEAMS ://
0911  4050 FORMAT(// STORY NO.',I3,'://
0912  4100 FORMAT(// FOR BEAM IN BAY NO.',I3,'://
0913  4200 FORMAT(// THIS SET OF DATA IS COMPLETE UPTO STORY',I3,'://
0914  +- ARE THE SHEAR COEFFICIENTS FOR THE BEAMS IN THE NEXT STORY//
0915  +- (AND MAX. UPTO STORY',I3,') EXACTLY LIKE THOSE//
0916  +- IN ANY PRECEDING STORY ? IF NOT---PUT NO, IF YES---LIKE WHICH ?//
0917  4300 FORMAT(// COLUMN MOD. OF ELASTICITY AND MOD. OF RIGIDITY SAME//
0918  +- ARE THE MOD. OF ELASTICITY AND MOD. OF RIGIDITY SAME//
0919  +- FOR ALL THE COLUMNS IN THIS FRAME GROUP ? Y/N//
0920  4600 FORMAT(// THIS SET OF DATA IS COMPLETE UPTO COL.LINE',I3,'://
0921  +- NOW, ARE THE MOD. OF ELASTICITY OF THE COLUMNS IN THE//
0922  +- NEXT COL.LINE EXACTLY LIKE THOSE IN ANY PRECEDING COL.LINE?//

```

```

0923      IF NOT---PUT NO, IF YES---LIKE WHICH ??)
0924      4700 FORMAT(/// SHEAR COEFFICIENTS FOR COLUMNS ://)
0925      4750 FORMAT(/// COLUMN LINE', I3, ' ://)
0926      4800 FORMAT(/// FOR COLUMN IN STORY NO.', I3)
0927      4900 FORMAT(/// THIS SET OF DATA IS COMPLETE UPTO COL. LINE', I3, ' ,//
0928      ++ ARE THE SHEAR COEFFICIENTS FOR COLUMNS IN THE NEXT//
0929      ++ COL. LINE EXACTLY LIKE THOSE IN ANY PRECEDING COL. LINE ??//
0930      ++ IF NOT---PUT NO, IF YES---LIKE WHICH ??)
0931      7000 FORMAT(5I5)
0932      7010 FORMAT(6E12.5)
0933      7020 FORMAT(5F15.5)
0934      7030 FORMAT(8F10.3)
0935      7040 FORMAT(IX, IAI)
0936      7045 FORMAT(IX, IAI, I5)
0937      7050 FORMAT(40A1)
0938      7060 FORMAT(IX, 40A1)
0939      7080 FORMAT(2I5, 4X, AI, F15.5)
0940      C
0941      8000 FORMAT(A1, 3IX, -- FINITE JOINT SIZES CONSIDERED ??)
0942      8010 FORMAT(A1, 3IX, -- SHEAR DEFORMATIONS CONSIDERED ??)
0943      8020 FORMAT(I3, 29X, -- NO. OF FRAMES')
0944      8030 FORMAT(I3, 29X, -- NO. OF FRAMES GROUPS')
0945      8040 FORMAT(I3, 29X, -- NO. OF STORIES')
0946      8045 FORMAT(I3, 29X, -- NO. OF BAYS IN THE WIDEST FRAME')
0947      8050 FORMAT(24A1, 8X, -- STORY HT. BETWN. LEVELS', I3, ' AND', I3)
0948      8060 FORMAT(I3, 29X, -- FOR FRAME', I3, ' FRAME GR. NO.')
0949      8070 FORMAT(A1, 3IX, -- ORIENTATION OF THE FRAME')
0950      8080 FORMAT(20A1, 12X, -- DIST. FROM REF. POINT')
0951      8100 FORMAT(I3, 29X, -- NO. OF TOWERS IN FRAME GR.', I3)
0952      8110 FORMAT(I3, 29X, -- IN TOWER', I3, ' NO. OF STORIES')
0953      8120 FORMAT(I3, 29X, -- NO. OF BAYS')
0954      8130 FORMAT(I3, 2IX, -- BAY-SETBACK AT LEFT BETWN TOWR', I2, ' &', I2)
0955      8140 FORMAT(24A1, 8X, -- BAY LENGTH FOR BAY NO.', I3)
0956      8150 FORMAT(// BEAM CROSS SECTION PROPERTIES')
0957      8155 FORMAT(A1, 3IX, -- SECTION TYPE, STORY', I3)
0958      8160 FORMAT(40A1, -- BEAM AT BAY', I3)
0959      8170 FORMAT(32A1, -- COMPLETE UPTO STORY', I3, ' , NEXT??)
0960      8190 FORMAT(A1, 2IX, -- ARE MOD. OF ELAST. SAME FOR ALL BEAMS??)
0961      8200 FORMAT(20A1, 12X, -- VALUES OF MOD. OF ELAST & RIGIDITY')
0962      8205 FORMAT(// SHEAR COEFFICIENTS FOR BEAMS')

```

4 OF 4

```

0963 8210 FORMAT(' COLUMN CROSS SECTION PROPERTIES')
0964 8220 FORMAT(A1,31X,'- X-SECTION TYPE AT COL. LINE',I3)
0965 8230 FORMAT(40A1,'- COLUMN AT STORY',I3)
0966 8240 FORMAT(32A1,'- COMPLETE UPTO COL. LINE',I3,' NEXT?')
0967 8260 FORMAT(A1,23X,'-ARE MOD. OF ELAST SAME FOR ALL COLMNS?')
0968 8265 FORMAT(' SHEAR COEFFICIENTS FOR COLUMNS')
0969 C
0970 END

```

.....
 SECOND PART OF PROGRAM - AIDS


```

0001 C
0002 C
0003 C
0004 C
0005 C
0006 C
0007 C
0008 COMMON/BLOCK1/IR,IW,INX,ITOT,JAX(80),IAL(48),IFIX(6),AFLX(6),
0009 +INDX(6),JBND(8,6),IBND(4,6),FBND(4,6)
0010 COMMON/MAIN/NIF(10),NST(10,5),NBAY(10,5),NBS(10,5),NSC(8),
0011 +NFIG(10),NFG,NFT,NNS,MBA,H(30),M(30),IFD(30),DDN(30),IAX(40),
0012 +IW1,IW3,IWS,IWF,IY,II,IL,IBL,IO,IDR,IDW,JTOT
0013 DIMENSION UDL(30,7),CNL(30,7,4),DIS(30,7,4),SMULP(15,8)
0014 DIMENSION NCL(30,7),HLX(30),HLY(30),XM(30),XMI(30),XB(30),YB(30)
0015 +,DHX(30),DHY(30)
0016 EQUIVALENCE (UDL(1,1),HLX(1)),(UDL(1,2),HLY(1)),(UDL(1,3),DHX(1))
0017 +,(UDL(1,4),DHY(1)),(UDL(1,5),XM(1)),(UDL(1,6),XMI(1)),
0018 +(UDL(1,7),XB(1)),(UDL(1,1),YB(1))
0019 C
0020 WRITE(IW1,7060) (IAX(KK),KK=1,40)
0021 WRITE(IW1,7000) NFT,NFG,NNS,MBA
0022 WRITE(IW1,7030) (H(I),I=1,NNS)
0023 DO 10 I=1,NFT
0024 10 WRITE(IW1,7080) I,M(I),IFD(I),DDN(I)
0025 C
0026 IF(IDW,NE,IY) GO TO 30
0027 DO 20 I=1,JTOT
0028 READ(IWS,7070) (JAX(KK),KK=1,70)
0029 WRITE(IWF,7070) (JAX(KK),KK=1,70)
0030 20 CONTINUE
0031 30 IF(IDR,EQ,II) GO TO 50
0032 DO 40 I=1,JTOT
0033 40 READ(IR,7040)
0034 50 CONTINUE
0035 C
0036 READ LOAD SYSTEMS INFORMATION
0037 C
0038 NGR=0
0039 NWL=0
0040 NDYN=0
0041 NSYS=0
0042 IMS=IBL

```

```

0043 ISS=IBL
0044 ITOT=1
0045 IDYN=IBL
0046 IF(IDR.EQ.IT) WRITE(IW,1410)
0047 CALL REED
0048 NGR=IFIX(1)
0049 IF(IDW.EQ.IY) WRITE(IWF,8270) NGR
0050 IF(IDR.EQ.IT) WRITE(IW,1420)
0051 CALL REED
0052 NWL=IFIX(1)
0053 IF(IDW.EQ.IY) WRITE(IWF,8280) NWL
0054 IF(IDR.EQ.IT) WRITE(IW,1430)
0055 CALL REED
0056 IYDN=IAL(1)
0057 IF(IDW.EQ.IY) WRITE(IWF,8290) IYDN
0058 IF(IDR.EQ.IT) WRITE(IW,1435)
0059 CALL REED
0060 NSYS=IFIX(1)
0061 IF(IDW.EQ.IY) WRITE(IWF,8300) NSYS
0062 IF(IYDN.EQ.IY) NDYN=1
0063 IOUT=0
0064 IF(IDR.EQ.IT) WRITE(IW,1436)
0065 CALL REED
0066 IMS=IAL(1)
0067 IF(IMS.EQ.IY) IOUT=1
0068 IF(IDW.EQ.IY) WRITE(IWF,8310) IMS
0069 IF(NDYN.EQ.O) GO TO 401
0070 IF(IDR.EQ.IT) WRITE(IW,1438)
0071 CALL REED
0072 ISS=IAL(1)
0073 IF(IDW.EQ.IY) WRITE(IWF,8320) ISS
0074 IF(ISS.EQ.IY) IOUT=2
0075 401 NK=NWL+NDYN
0076 WRITE(IW,7000) NGR,NWL,NDYN,NSYS,IOUT
0077 IF(NK.EQ.O.AND.IOUT.EQ.O) GO TO 402
0078 IF(IDR.EQ.IT) WRITE(IW,5700)
0079 CALL REED
0080 NOPT=IFIX(1)
0081 IF(IDW.EQ.IY) WRITE(IWF,8325) NOPT
0082 WRITE(IW,7000) NOPT

```

```

0083 402' IF(NSYS.EQ.O) GO TO 426
0084 C
0085 NCASE=NK+NGR
0086 IF(IDR.EQ.IT) WRITE(IW,4910)
0087 DO 403 I=1,NSYS
0088 DO 403 J=1,NCASE
0089 403 SMULP(I,J)=O.O
0090 C
0091 DO 425 I=1,NSYS
0092 J=0
0093 IF(IDR.EQ.IT) WRITE(IW,4920) I
0094 IF(NGR-I) 408,404,405
0095 404 ITOT=I
0096 IF(IDR.EQ.IT) WRITE(IW,4925)
0097 CALL REED
0098 J=J+1
0099 SMULP(I,J)=AFLX(I)
0100 GO TO 407
0101 405 ITOT=NGR
0102 IF(IDR.EQ.IT) WRITE(IW,4930) NGR
0103 CALL REED
0104 DO 406 K=1,NGR
0105 J=J+1
0106 SMULP(I,J)=AFLX(K)
0107 406 IF(IDW.EQ.IY) WRITE(IWF,8330) (JAX(KK),KK=1,32),I
0108 408 IF(NWL-I) 416,410,412
0109 410 ITOT=I
0110 J=J+1
0111 IF(IDR.EQ.IT) WRITE(IW,4935)
0112 CALL REED
0113 SMULP(I,J)=AFLX(I)
0114 GO TO 415
0115 412 ITOT=NWL
0116 IF(IDR.EQ.IT) WRITE(IW,4940) NWL
0117 CALL REED
0118 DO 414 K=1,NWL
0119 J=J+1
0120 414 SMULP(I,J)=AFLX(K)
0121 415 IF(IDW.EQ.IY) WRITE(IWF,8340) (JAX(KK),KK=1,32),I
0122 416 IF(NDYN.EQ.O) GO TO 418

```

```

0123 ITOT=1
0124 IF(IDR.EQ.IT) WRITE(IW,4945)
0125 CALL REED
0126 IF(IDW.EQ.IY) WRITE(IWF,8350) (JAX(KK),KK=1,8),I
0127 J=J+1
0128 SMULP(I,J)=AFLX(1)
0129 418 WRITE(IW1,7030) (SMULP(I,K),K=1,NCASE)
0130 425 CONTINUE
0131 C
0132 426 IF(NGR.EQ.O.AND.NWL.EQ.O) GO TO 560
0133 C
0134 C READ DATA FOR GRAVITY LOAD CASES
0135 C DATA FOR DISTRIBUTED LOADS :
0136 C
0137 IF(NGR.EQ.O) GO TO 510
0138 IFR=0
0139 NFR=0
0140 IF(IDR.NE.IT) GO TO 427
0141 WRITE(IW,4950)
0142 GO TO 428
0143 427 READ(IR,7040)
0144 428 IF(IDW.EQ.IY) WRITE(IWF,8360)
0145 DO 500 IJ=1,NFG
0146 NNG=NFIG(IJ)
0147 WRITE(IW3,7000) NNG
0148 IYF=IBL
0149 ITF=IBL
0150 IOF=IBL
0151 INF=0
0152 NFR=NFR+NNG
0153 DO 490 JJ=1,NNG
0154 IFR=IFR+1
0155 IF(IYF.NE.IY) GO TO 431
0156 INDL=1
0157 WRITE(IW3,7000) INDL
0158 IF(ITF.NE.IT.OR.IOF.NE.IO) IYF=IBL
0159 IF(IFR.EQ.INF) IYF=IBL
0160 GO TO 490
0161 431 INDL=0
0162 WRITE(IW3,7000) INDL

```



```

0163 IF(IDR.EQ.IT) WRITE(IW,4980) IFR
0164 C
0165 NT=NTF(IJ)
0166 DO 475 IK=1,NGR
0167 IF(IDR.NE.IT) GO TO 432
0168 WRITE(IW,4990) IK
0169 WRITE(IW,5000)
0170 GO TO 433
0171 432 READ(IR,7040)
0172 433 IF(IDW.EQ.IY) WRITE(IMF,8370) IFR,IK
0173 IS=0
0174 NN=0
0175 DO 450 I=1,NT
0176 NSS=NST(IJ,I)
0177 NB=NBAY(IJ,I)
0178 NN=NN+NSS
0179 IF(I.NE.1) GO TO 434
0180 NSB=0
0181 GO TO 435
0182 434 NSB=NSB+NBS(IJ,I-1)
0183 435 NBS1=NSB+1
0184 NBS2=NSB+NB
0185 ILL=IBL
0186 IIT=IBL
0187 IOO=IBL
0188 IMB=0
0189 IND=0
0190 DO 448 J=1,NSS
0191 IS=IS+1
0192 IIT=IBL
0193 IOO=IBL
0194 INC=0
0195 IF(ILL.NE.IL) GO TO 438
0196 DO 436 II=NBS1,NBS2
0197 UDL(IS,II)=UDL(IMB,II)
0198 436 CONTINUE
0199 WRITE(IW3,7030) (UDL(IS,II),II=NBS1,NBS2)
0200 IF(IIT.NE.IT.OR.IOO.NE.IO) GO TO 445
0201 IF(IS.EQ.IND) GO TO 445
0202 GO TO 448

```

```

0203 438 IF(IDR.EQ.IT) WRITE(IW,4050) IS
0204 DO 444 K=NBS1,NBS2
0205 IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 440
0206 IF(I.GT.INC) GO TO 440
0207 UDL(IS,K)=UDL(IS,K-1)
0208 GO TO 444
0209 440 IF(IDR.EQ.IT) WRITE(IW,4100) K
0210 ITOI=3
0211 CALL REED
0212 IF(IDW.EQ.IY) WRITE(IWF,8160) (JAX(KK),KK=1,40),K
0213 UDL(IS,K)=AFLX(1)
0214 IIT=IAL(9)
0215 IIO=IAL(10)
0216 INC=IFIX(3)
0217 444 CONTINUE
0218 WRITE(IW3,7030) (UDL(IS,K),K=NBS1,NBS2)
0219 IF(IS.GE.NN) GO TO 448
0220 446 IF(IDR.EQ.IT) WRITE(IW,5200) IS,NN
0221 ITOI=4
0222 CALL REED
0223 IF(IDW.EQ.IY) WRITE(IWF,8170) (JAX(KK),KK=1,32),IS
0224 ILL=IAL(1)
0225 IMB=IFIX(2)
0226 IIT=IAL(17)
0227 IIO=IAL(18)
0228 IND=IFIX(4)
0229 448 CONTINUE
0230 450 CONTINUE
0231 IF(IDR.EQ.IT) WRITE(IW,5300)
0232 C
0233 C
0234 C
0235 C
0236 IS=0
0237 NN=0
0238 DO 470 I=1,NT
0239 NSS=NST(IJ,I)
0240 NN=NN+NSS
0241 NB=NBAY(IJ,I)
0242 IF(I.NE.1) GO TO 452
    NSB=0

```

READ FORCE DATA FOR CONCENTRATED LOADS

```

0243      GO TO 454
0244      452 NSB=NSB+NBS(IJ,I-1)
0245      454 NBS1=NSB+1
0246      NBS2=NSB+NB
0247      ILL=IBL
0248      IIT=IBL
0249      IOO=IBL
0250      IMB=0
0251      IND=0
0252      DO 468 J=1,NSS
0253      IS=IS+1
0254      IIT=IBL
0255      IOO=IBL
0256      INC=0
0257      IF(ILL.NE.IL) GO TO 460
0258      DO 458 II=NBS1,NBS2
0259      NC=NCL(IMB,II)
0260      NCL(IS,II)=NC
0261      WRITE(IW3,7000) NC
0262      IF(NC.EQ.0) GO TO 458
0263      DO 455 JK=1,NC
0264      CNL(IS,II,JK)=CNL(IMB,II,JK)
0265      DIS(IS,II,JK)=DIS(IMB,II,JK)
0266      455 CONTINUE
0267      WRITE(IW3,7030) (CNL(IS,II,JK),DIS(IS,II,JK),JK=1,NC)
0268      458 CONTINUE
0269      IF(IIT.NE.IT.OR.IOO.NE.IO) GO TO 465
0270      IF(IS.EQ.IND) GO TO 465
0271      GO TO 468
0272      460 IF(IDR.EQ.IT) WRITE(IW,4050) IS
0273      DO 464 K=NBS1,NBS2
0274      IF(IIT.NE.IT.OR.IIO.NE.IO) GO TO 461
0275      IF(K.GT.INC) GO TO 461
0276      NCL(IS,K)=NCL(IS,K-1)
0277      WRITE(IW3,7000) NCL(IS,K)
0278      GO TO 464
0279      461 IF(IDR.EQ.IT) WRITE(IW,5350) K
0280      ITOT=3
0281      CALL REED
0282      NCL(IS,K)=IFIX(1)

```

```

0283 IIT=IAL(9)
0284 IIO=IAL(10)
0285 INC=IFIX(3)
0286 NC=NCL(IS,K)
0287 IF(NC.LE.4) GO TO 462
0288 IF(IDR.EQ.IT) WRITE(IW,5370)
0289 GO TO 461
0290 462 WRITE(IW3,7000) NC
0291 IF(IDW.EQ.IY) WRITE(IWF,8380) (JAX(KK),KK=1,24),K,IS
0292 IF(NC.EQ.0) GO TO 464
0293 IF(IDR.EQ.IT) WRITE(IW,5380)
0294 DO 463 JK=1,NC
0295 IF(IDR.EQ.IT) WRITE(IW,5400) JK
0296 ITOT=2
0297 CALL REED
0298 IF(IDW.EQ.IY) WRITE(IWF,8390) (JAX(KK),KK=1,16),JK
0299 CNL(IS,K,JK)=AFLX(1)
0300 DIS(IS,K,JK)=AFLX(2)
0301 463 CONTINUE
0302 WRITE(IW3,7030) (CNL(IS,K,JK),DIS(IS,K,JK),JK=1,NC)
0303 464 CONTINUE
0304 465 IF(IS.GE.NN) GO TO 468
0305 466 IF(IDR.EQ.IT) WRITE(IW,5450) IS,NN
0306 ITOT=4
0307 CALL REED
0308 IF(IDW.EQ.IY) WRITE(IWF,8170) (JAX(KK),KK=1,32),IS
0309 ILL=IAL(1)
0310 IMB=IFIX(2)
0311 IIT=IAL(17)
0312 IOO=IAL(18)
0313 IND=IFIX(4)
0314 468 CONTINUE
0315 470 CONTINUE
0316 475 CONTINUE
0317 IF(JJ.EQ.NNG) GO TO 490
0318 IF(IDR.EQ.IT) WRITE(IW,5480) IFR,NFR
0319 ITOT=3
0320 CALL REED
0321 IF(IDW.EQ.IY) WRITE(IWF,8400) (JAX(KK),KK=1,24),IFR
0322 IYF=IAL(1)

```

```

0323 ITF=IAL(9)
0324 IOF=IAL(10)
0325 INF=IFIX(3)
0326 490 CONTINUE
0327 500 CONTINUE
0328 C
0329 C READ DATA FOR LATERAL LOADS
0330 C
0331 510 IF(NWL.EQ.O) RETURN
0332 IF(IDR.EQ.IT) WRITE(IW,5500)
0333 DO 555 J=1,NWL
0334 IF(IDR.EQ.IT) WRITE(IW,5520) J
0335 ILL=IBL
0336 ITT=IBL
0337 IOO=IBL
0338 IMB=O
0339 IND=O
0340 DO 550 I=1,NNS
0341 IF(ILL.NE.IL) GO TO 520
0342 HLX(I)=HLX(IMB)
0343 HLY(I)=HLY(IMB)
0344 DHX(I)=DHX(IMB)
0345 DHY(I)=DHY(IMB)
0346 WRITE(IW3,7020) HLX(I),HLY(I),DHX(I),DHY(I)
0347 IF(ITT.NE.IT,OR.IOO.NE.IO) GO TO 530
0348 IF(I.EQ.IND) GO TO 530
0349 GO TO 550
0350 520 IF(IDR.EQ.IT) WRITE(IW,5550) I
0351 ITOT=2
0352 CALL REED
0353 IF(IDW.EQ.IY) WRITE(IWF,8410) (JAX(KK),KK=1,16),J
0354 HLX(I)=AFLX(1)
0355 HLY(I)=AFLX(2)
0356 IF(IDR.EQ.IT) WRITE(IW,5600)
0357 CALL REED
0358 IF(IDW.EQ.IY) WRITE(IWF,8420) (JAX(KK),KK=1,16)
0359 DHX(I)=AFLX(1)
0360 DHY(I)=AFLX(2)
0361 WRITE(IW3,7020) HLX(I),HLY(I),DHX(I),DHY(I)
0362 530 IF(1.GE.NNS) GO TO 550

```

```

0363 540 IF(IDR.EQ.IT) WRITE(IW,5650) I
0364 ITOT=4
0365 CALL REED
0366 IF(IDW.EQ.IY) WRITE(IWF,8170) (JAX(KK),KK=1,32),I
0367 ILL=IAL(1)
0368 IMB=IFIX(2)
0369 ITT=IAL(17)
0370 IOO=IAL(18)
0371 IND=IFIX(4)
0372 550 CONTINUE
0373 555 CONTINUE
0374 C
0375 C READ DATA FOR DYNAMIC ANALYSIS
0376 C
0377 560 IF(IYDN.NE.IY.AND.IOUT.EQ.0) GO TO 650
0378 IF(IDR.NE.IT) GO TO 562
0379 WRITE(IW,5800)
0380 GO TO 564
0381 562 READ(IR,7040)
0382 564 IF(IDW.EQ.IY) WRITE(IWF,8430)
0383 CALL REED
0384 NM=IFIX(1)
0385 IF(IDW.EQ.IY) WRITE(IWF,8440) NM
0386 IF(IDR.EQ.IT) WRITE(IW,5850)
0387 CALL REED
0388 LM=IFIX(1)
0389 IF(IDW.EQ.IY) WRITE(IWF,8450) LM
0390 WRITE(IW,7000) NM,LM
0391 IF(NDYN.EQ.0) GO TO 566
0392 IF(IDR.EQ.IT) WRITE(IW,5900)
0393 CALL REED
0394 XMU=AFLX(1)
0395 IF(IDW.EQ.IY) WRITE(IWF,8460) (JAX(KK),KK=1,10)
0396 IF(IDR.EQ.IT) WRITE(IW,6000)
0397 CALL REED
0398 SF=AFLX(1)
0399 IF(IDW.EQ.IY) WRITE(IWF,8470) (JAX(KK),KK=1,10)
0400 IF(IDR.EQ.IT) WRITE(IW,6050)
0401 CALL REED
0402 BETA=AFLX(1)

```

```

0403 IF(IDW.EQ.IY) WRITE(IWF,8480) (JAX(KK),KK=1,10)
0404 IF(IDR.EQ.IT) WRITE(IW,6100)
0405 CALL REED
0406 NP=FIX(1)
0407 IF(IDW.EQ.IY) WRITE(IWF,8490) NP
0408 WRITE(IWI,7000) NP
0409 IF(IDR.EQ.IT) WRITE(IW,6110)
0410 DO 565 IP=1,NP
0411 IF(IDR.EQ.IT) WRITE(IW,6120) IP
0412 CALL REED
0413 TC=AFLX(1)
0414 SAC=AFLX(2)
0415 IF(IDW.EQ.IY) WRITE(IWF,8500) (JAX(KK),KK=1,16),IP
0416 WRITE(IWI,7030) TC,SAC
0417 CONTINUE
0418 IF(IDR.EQ.IT) WRITE(IW,6150)
0419 CALL REED
0420 TD=AFLX(1)
0421 IF(IDW.EQ.IY) WRITE(IWF,8510) (JAX(KK),KK=1,10)
0422 IF(IDR.EQ.IT) WRITE(IW,6200)
0423 CALL REED
0424 AG=AFLX(1)
0425 IF(IDW.EQ.IY) WRITE(IWF,8520) (JAX(KK),KK=1,10)
0426 WRITE(IWI,7030) XMU,SF,BETA,TD,AG
0427 CONTINUE
0428 IF(IDR.NE.IT) GO TO 567
0429 WRITE(IW,6300)
0430 GO TO 568
0431 567 READ(IR,7040)
0432 568 IF(IDW.EQ.IY) WRITE(IWF,8530)
0433 ILL=IBL
0434 IIT=IBL
0435 IOK=IBL
0436 IMB=O
0437 IND=O
0438 DO 600 I=1,NNS
0439 IF(ILL.NE.IL) GO TO 570
0440 XM(I)=XM(IMB)
0441 XMI(I)=XMI(IMB)
0442 XB(I)=XB(IMB)

```

```

0443 YB(I)=YB(IMB)
0444 WRITE(IW1,7030) XM(I),XMI(I),XB(I),YB(I)
0445 IF(ITT.NE.IT.OR.I00.NE.I0) GO TO 580
0446 IF(I.EQ.IND) GO TO 580
0447 GO TO 600
0448 570 IF(IDR.EQ.IT) WRITE(IW,6350) I
0449 CALL REED
0450 XM(I)=AFLX(I)
0451 IF(IDW.EQ.IY) WRITE(IMF,8540) (JAX(KK),KK=1,10),I
0452 IF(NOPT.EQ.3) GO TO 575
0453 XMI(I)=0.0
0454 GO TO 576
0455 575 IF(IDR.EQ.IT) WRITE(IW,6400)
0456 CALL REED
0457 XMI(I)=AFLX(I)
0458 IF(IDW.EQ.IY) WRITE(IMF,8550) (JAX(KK),KK=1,10)
0459 576 IF(IDR.EQ.IT) WRITE(IW,6450)
0460 ITOI=2
0461 CALL REED
0462 XB(I)=AFLX(I)
0463 YB(I)=AFLX(2)
0464 IF(IDW.EQ.IY) WRITE(IMF,8560) (JAX(KK),KK=1,16)
0465 WRITE(IW1,7030) XM(I),XMI(I),XB(I),YB(I)
0466 IF(I.GE.NNS) GO TO 600
0467 580 IF(IDR.EQ.IT) WRITE(IW,6500) I
0468 ITOI=4
0469 CALL REED
0470 IF(IDW.EQ.IY) WRITE(IMF,8170) (JAX(KK),KK=1,32),I
0471 ILL=IAL(I)
0472 IMB=IFIX(2)
0473 ITI=IAL(17)
0474 I00=IAL(18)
0475 IND=IFIX(4)
0476 600 CONTINUE
0477 C
0478 C
0479 C
0480 FIND NO.OF ELEMENTS OF STIFFNESS MATRIX RECD TO STORE
0481
0482 650 NEI=0
0483 DO 700 IJ=1,NFG
0484 NT=NTF(IJ)

```



```

0483 JTN=0
0484 JTR=0
0485 DO 695 I=1,NT
0486 NB=NBAY(IJ,I)
0487 NSS=NST(IJ,I)
0488 JA=2*NB+4
0489 JB=3*NB+5
0490 JC=NB+1
0491 JY=2*JA*JC+JB
0492 JTN=JTN+JY*(NSS-1)
0493 IF(I.EQ.1) GO TO 692
0494 NSB=NBS(IJ,I-1)
0495 JP=JA1-NSB
0496 JQ=JP+JC
0497 JR=NB+NBI-NSB+4
0498 JT=JC*(JP+JR)+JQ
0499 JTR=JTR+JT
0500 GO TO 694
0501 692 JX=(JA-NB)*NB+2*(JA-1)
0502 694 JA1=JA
0503 NBI=NB
0504 695 CONTINUE
0505 NEL=JX+JTN+JTR
0506 IF(NEL.GT.NEL) NEL=NEL
0507 NEL=NEL
0508 C
0509 700 CONTINUE
0510 WRITE(IW,7090) NEL
0511 C
0512 STOP
0513 C
0514 1410 FORMAT(/// LOAD CASE INFORMATION DATA :///
0515 + NUMBER OF GRAVITY LOAD CASES (MAX. 4)')
0516 1420 FORMAT(// NUMBER OF LATERAL LOAD CASES (MAX. 3)')
0517 1430 FORMAT(// DO YOU WANT TO INCLUDE EARTHQUAKE LOADS',/
0518 + IN ANALYSIS ? Y/N')
0519 1435 FORMAT(// NUMBER OF LOAD SYSTEMS (LOAD COMBINATIONS)')
0520 1436 FORMAT(// DO YOU WANT THE STRUCTURAL'//
0521 + MODE SHAPES AND FREQUENCIES ? Y/N')
0522 1438 FORMAT(// DO YOU WANT EARTHQUAKE STORY SHEAR',/

```

```

0523  + FORCES AND DISPLACEMENTS ? Y/N )
0524  4050 FORMAT(// STORY NO., I3, //
0525  4100 FORMAT(// FOR BEAM IN BAY NO., I3, //
0526  4910 FORMAT(// LOAD SYSTEMS INFORMATION DATA ://
0527  + LOAD SYSTEMS MUST BE ARRANGED(NUMBERED) SEQUENTIALLY//
0528  + ACCORDING TO THE FOLLOWING PREFERENCE LIST.....//
0529  + LOAD SYSTEMS CONTAINING ---//
0530  + 1 GRAVITY LOAD CASES ONLY//
0531  + 2 GRAVITY LOAD, AND/OR LATERAL LOAD CASES//
0532  + 3 GRAVITY LOAD, AND/OR LATERAL LOAD, AND/OR SEISMIC LOADS//
0533  4920 FORMAT(// LOAD SYSTEM, I3, //
0534  4925 FORMAT(// INPUT MULT. FACTOR FOR THE GRAVITY LOAD CASE//
0535  4930 FORMAT(// INPUT MULT. FACTORS FOR GRAVITY LOAD CASES 1 THRU, I3, //
0536  + SEPARATED BY BLANKS//
0537  4935 FORMAT(// INPUT MULT. FACTOR FOR THE LATERAL LOAD CASE//
0538  4940 FORMAT(// INPUT MULT. FACTORS FOR LATERAL LOAD CASES 1 THRU, I3, //
0539  + SEPARATED BY BLANKS//
0540  4945 FORMAT(// INPUT MULT. FACTOR FOR THE EARTHQUAKE LOAD(IF ANY)//
0541  4950 FORMAT(// DATA FOR GRAVITY LOAD CASES ://
0542  4980 FORMAT(//
0543  + -----//
0544  + FRAME NO., I3, //
0545  + -----//
0546  4990 FORMAT(// GRAVITY LOAD CASE, I3, //
0547  5000 FORMAT(// UNIFORMLY DISTRIBUTED LOADS(UDL) ://
0548  + INPUT THE DISTRIBUTED LOAD PER LINEAR UNIT FOR EACH BEAM//
0549  5200 FORMAT(// UDL DATA IS COMPLETE UPTO STORY, I3, //
0550  + ARE THE UDLs ON THE BEAMS IN THE NEXT STORY(AND MAX. UPTO//
0551  + STORY, I3, //) EXACTLY LIKE THOSE IN ANY PRECEDING STORY ?//
0552  + IF NOT,---PUT NO, IF YES---LIKE WHICH ?//
0553  5300 FORMAT(// DATA FOR CONCENTRATED LOADS ://
0554  + (MAX. 4 LOADS PER BEAM)//
0555  5350 FORMAT(// NO. OF CONC. LOADS ON THE BEAM AT BAY NO., I3)
0556  5370 FORMAT(// THE NO. OF CONC. LOADS JUST SUPPLIED//
0557  + IS MORE THAN 4, TRY AGAIN.....//
0558  5380 FORMAT(// INPUT THE LOAD VALUE AND THEN ITS DISTANCE FROM//
0559  + THE LEFT COLUMN CENTER, SEPARATED BY A BLANK//
0560  5400 FORMAT(// LOAD NO., I3, //
0561  5450 FORMAT(// CONC. LOAD DATA IS COMPLETE UPTO STORY , I3, //
0562  + ARE THE LOADS AND THEIR POSITIONS IN THE NEXT STORY//

```

```

0563      + (AND MAX. UPTO STORY',I3,' ) EXACTLY LIKE THOSE'//
0564      + IN ANY PRECEDING STORY ? IF NOT--PUT NO, IF YES--LIKE WHICH ??'//
0565 5480 FORMAT(/// GRAVITY LOAD DATA IS COMPLETE FOR FRAME NO.',I3,',//
0566      + IS THIS DATA EXACTLY THE SAME( FOR ALL GR. LOAD CASES)??'//
0567      + IN THE NEXT FRAME( AND POSSIBLY UPTO FRAME NO.',I3,', ) ? Y/N' )
0568 5500 FORMAT(/// DATA FOR LATERAL LOAD CASES ://)
0569 5520 FORMAT(/// LATERAL LOAD CASE',I3,', ://)
0570 5550 FORMAT(/// STORY NO.',I3,', ://)
0571      + INPUT HOR. LOAD COMPONENT IN GLOBAL X-DIRECTION AND THEN //
0572      + IN THE Y-DIRECTION, SEPARATED BY A BLANK' )
0573 5600 FORMAT(/// INPUT THE X-DISTANCE AND THEN THE Y-DISTANCE OF //
0574      + THE POINT OF ACTION FROM GLOBAL ORIGIN' )
0575 5650 'FORMAT(/// LOAD AND DISTANCE DATA IS COMPLETE UPTO STORY',I3,',, //
0576      + IS THIS DATA FOR THE NEXT STORY EXACTLY LIKE THAT'//
0577      + IN ANY PRECEDING STORY? IF NOT--PUT NO, IF YES--LIKE WHICH??' )
0578 5700 'FORMAT(/// FOR THREE DIMENSIONAL ANALYSIS ://)
0579      + CODE FOR NO. OF DEGREES OF FREEDOM AT EACH STORY ? 1/2/3'//
0580      + 1 FOR X-TRANSLATION ONLY'//
0581      + 2 FOR Y-TRANSLATION ONLY'//
0582      + 3 FOR X, Y-TRANSLATION AND ROTATION'//)
0583 5800 FORMAT(/// DYNAMIC ANALYSIS ://)
0584      + NO. OF MODES TO BE CONSIDERED' )
0585 5850 FORMAT(/// NO. OF MODES TO BE EXTRACTED FROM ONE ORIGIN'//
0586      + (USUALLY NOT MORE THAN 5)' )
0587 5900 FORMAT(/// DUCTILITY FACTOR' )
0588 6000 FORMAT(/// SCALE FACTOR FOR DESIGN SPECTRUM' )
0589 6050 FORMAT(/// ANGLE OF INCIDENCE OF EARTHQUAKE'//
0590      + WITH THE GLOBAL X-AXIS, DEGREES' )
0591 6100 FORMAT(/// NO. OF PERIOD COORDINATES FOR RESPONSE SPECTRUM' )
0592 6110 FORMAT(/// FOR EACH PERIOD COORDINATE, INPUT THE PERIOD VALUE'//
0593      + AND THEN THE CORRESPONDING SPECTRAL ACCELERATION',//
0594      + SEPARATED BY A BLANK' )
0595 6120 FORMAT(/// PERIOD COORD. NO.',I3,', ://)
0596 6150 FORMAT(/// PERIOD WHICH SEPARATES THE ACCL. BOUND FROM'//
0597      + THE VEL. BOUND ON THE RESPONSE SPECTRUM' )
0598 6200 FORMAT(/// ACCELERATION DUE TO GRAVITY' )
0599 6300 FORMAT(/// MASS DATA ://)
0600 6350 FORMAT(/// AT STORY LEVEL',I3,', ; TRANSLATIONAL FLOOR MASS' )
0601 6400 FORMAT(/// ROTATIONAL MASS MOM. OF INERTIA' )
0602 6450 FORMAT(/// PUT X-DISTANCE AND THEN Y-DISTANCE OF THE CENTRE OF'//

```

```

0603      * MASS FROM THE REFERENCE POINT, SEPARATED BY A BLANK *
0604      FORMAT(// THE MASS AND DISTANCE DATA IS COMPLETE UPTO STORY',I3, /
0605      * IS THIS DATA IN THE NEXT STORY EXACTLY LIKE THAT *
0606      * IN ANY PRECEDING STORY ? IF NOT--PUT NO, IF YES--LIKE WHICH??)
0607      FORMAT(5I5)
0608      FORMAT(5F15.5)
0609      FORMAT(8F10.3)
0610      FORMAT(1X,1A1)
0611      FORMAT(40A1)
0612      FORMAT(1X,40A1)
0613      FORMAT(70A1)
0614      FORMAT(2I5,4X,A1,F15.5)
0615      FORMAT(// ** MAX. NO. OF FRAME STIFFNESS ELEMENTS //
0616      * ** REQUIRED TO BE STORED IS, NEL =',I6)
0617      C
0618      8160 FORMAT(40A1, '- BEAM AT BAY',I3)
0619      8170 FORMAT(32A1, '- COMPLETE UPTO STORY',I3, ', NEXT??)
0620      8270 FORMAT(I3,29X, '- NO. OF GRAVIT LOAD CASES')
0621      8280 FORMAT(I3,29X, '- NO. OF LATERAL LOAD CASES')
0622      8290 FORMAT(A1,31X, '- SEISMIC LOADS CONSIDERED??)
0623      8300 FORMAT(I3,29X, '- NO. OF LOAD SYSTEMS CONSIDERED')
0624      8310 FORMAT(A1,31X, '- OUTPUT FOR MODE SHAPES REQD.??)
0625      8320 FORMAT(A1,31X, '- OUTPUT FOR DYN. STORY FORCES REQD.??)
0626      8325 FORMAT(I3,29X, '- CODE FOR STORY DEGREES OF FREEDOM')
0627      8330 FORMAT(32A1, '- IN LOAD SYSTM',I3, ', FACTRS FOR GRVT LOADS')
0628      8340 FORMAT(32A1, '- IN LOAD SYSTM',I3, ', FACTRS FOR LATR LOADS')
0629      8350 FORMAT(8A1,24X, '- FACTR FOR SEISMIC LOADS, LOAD SYSTM',I3)
0630      8360 FORMAT(' GRAVITY LOAD CASE DATA')
0631      8370 FORMAT(' UDL - FRAME',I3, ', GRVT LOAD CASE',I3)
0632      8380 FORMAT(24A1,8X, '- NO. OF CONCTD. LOADS, BEAM',I2, ', STORY',I3)
0633      8390 FORMAT(16A1,16X, '- LOAD -',I3, ' & DIST.')
0634      8400 FORMAT(24A1, '- COMPLETE UPTO FRAME',I3, ', NEXT-SAME??)
0635      8410 FORMAT(16A1,16X, '- LAT. LOAD X-Y-COMPNTS FOR CASE',I2)
0636      8420 FORMAT(16A1,16X, '- X-DIST & Y-DIST')
0637      8430 FORMAT(' DYNAMIC ANALYSIS :')
0638      8440 FORMAT(I3,29X, '- NO. OF MODES CONSIDERED')
0639      8450 FORMAT(I3,29X, '- MODES EXTRACTED FROM ONE ORIGIN')
0640      8460 FORMAT(10A1,22X, '- DUCTILITY FACTOR')
0641      8470 FORMAT(10A1,22X, '- SCALE FACTOR')
0642      8480 FORMAT(10A1,22X, '- ANGLE OF INCIDENCE')

```

```

0643      FORMAT(I3,29X,'- NO. OF PERIOD COORDS.')
```

```

0644      FORMAT(16A1,16X,'- PERIOD & ACCLN. OF COORD.',I3)
```

```

0645      FORMAT(10A1,22X,'- PERIOD BOUND')
```

```

0646      FORMAT(10A1,22X,'- ACCLN. DUE TO GRAVITY')
```

```

0647      FORMAT(' MASS DATA :')
```

```

0648      FORMAT(10A1,22X,'- MASS AT STORY',I3)
```

```

0649      FORMAT(10A1,22X,'- MASS MOM. OF INERTIA')
```

```

0650      FORMAT(16A1,16X,'- X-DIST & Y-DIST FROM REF POINT')
```

```

0651 C
```

```

0652      END
```

```

1.000 C SUBROUTINE REED
2.000 C SUBROUTINE FOR FREE FORMAT READ
3.000 C
4.000 C VERSION OF UTILITY SUBROUTINE REED: MAY 4, 1978
5.000 C
6.000 C COMMON/BLOK1/IR,IW,IN,ITOT,JA(80),IA(48),IFI(6),AFL(6),IND(6),
7.000 C +JBND(8,6),IBND(4,6),FBND(4,6)
8.000 C DIMENSION JBNTT(6)
9.000 C
10.000 C DATA IO,I1,I2,I3,I4,I5,I6,I7,I8,I9/
11.000 C +IH0,IH1,IH2,IH3,IH4,IH5,IH6,IH7,IH8,IH9/
12.000 C DATA IBLANK,IPLUS,IMINUS,IDOT/IH,IH+,IH-,IH.,/
13.000 C DATA II,I0,IQ,IT,IU,IY/IHI,IHO,IHT,IHU,IHY/
14.000 C DATA IF,INN/1HF,1HN/
15.000 C
16.000 C INITIALIZE INPUT DATA ARRAYS TO BLANK OR ZERO
17.000 C
18.000 C 112 DO 100 I=1,48
19.000 C 100 IA(I)=IBLANK
20.000 C DO 101 I=1,6
21.000 C IFI(I)=0
22.000 C IND(I)=0
23.000 C JBNTT(I)=0
24.000 C 101 AFL(I)=0.
25.000 C
26.000 C DO 603 J=1,6
27.000 C DO 603 K=1,4
28.000 C 603 JBNTT(J)=JBNTT(J)+JBND(K,J)+JBND(K+4,J)
29.000 C
30.000 C READ UNPROCESSED ALPHANUMERIC INPUT
31.000 C M70$
32.000 C 110 READ(IR,1,END=108)(JA(I),I=1,IN)
33.000 C M70$
34.000 C GO TO 109
35.000 C MALS
36.000 C 110 READ(IR,1)(JA(I),I=1,IN)
37.000 C MALS
38.000 C IF(ENDFILE IR)108,109
39.000 C 108 WRITE(IW,3)
40.000 C GO TO 110

```

```

41.000 C
42.000 C SET COUNTS, ETC TO INITIAL VALUES
43.000 109 KOUNT=1
44.000 IOUNT=1
45.000 INUMB=1
46.000 NUMB=0
47.000 ISIGN=1
48.000 IDES=0
49.000 IDET=1
50.000 IWORD=0
51.000 ISIZE=0
52.000 IBIG=0
53.000 IZERO=0
54.000 C
55.000 C PROCESS AS MUCH AS REQUESTED OF INPUT ARRAY UNLESS TERMINATED
56.000 C BY UPPER LIMITS ON NO. OF DATA UNITS TO BE GENERATED
57.000 DO 102 I=1, IN
58.000 C RECOGNIZE BLANK AND TRANSFER CONTROL FOR PROCESSING OF BLANK
59.000 IF(JA(I).NE.IBLANK)GOTO 103
60.000 IF(IWORD.EQ.0)GOTO 102
61.000 IF(INUMB.EQ.0)GOTO 107
62.000 GOTO 113
63.000 C
64.000 103 IWORD=1
65.000 IF(ISIZE.GT.8)GOTO 106
66.000 C LOAD A1 DATA INTO ALPHANUMERIC ARRAY IF DATA UNIT LENGTH DOES NOT
67.000 C EXCEED EIGHT
68.000 IA(IOUNT)=JA(I)
69.000 IOUNT=IOUNT+1
70.000 ISIZE=ISIZE+1
71.000 106 IF(INUMB.EQ.0)GOTO 102
72.000 C
73.000 C CONVERT ALPHANUMERIC DIGIT TO NUMERICAL EQUIVALENT
74.000 IF(JA(I).EQ.10.OR.JA(I).EQ.10)GOTO 500
75.000 IF(JA(I).EQ.11.OR.JA(I).EQ.11)GOTO 501
76.000 IF(JA(I).EQ.12)GOTO 502
77.000 IF(JA(I).EQ.13)GOTO 503
78.000 IF(JA(I).EQ.14)GOTO 504
79.000 IF(JA(I).EQ.15)GOTO 505
80.000 IF(JA(I).EQ.16)GOTO 506

```

```

81.000 IF(JA(I).EQ.I7)GOTO 507
82.000 IF(JA(I).EQ.I8)GOTO 508
83.000 IF(JA(I).EQ.I9)GOTO 509
84.000 IF(JA(I).EQ.IDOT)GOTO 150
85.000 IF(JA(I).EQ.IMINUS)GOTO 122
86.000 INUMB=0
87.000 ISIGN=1
88.000 IDES=0
89.000 IBIG=0
90.000 IFI(KOUNT)=0
91.000 AFL(KOUNT)=0.
92.000 IND(KOUNT)=0
93.000 GOTO 102
94.000 C
95.000 150 IF(IDES.EQ.1)INUMB=0
96.000 IDES=1
97.000 C IDES=1 MEANS THAT A DECIMAL POINT HAS BEEN ENCOUNTERED
98.000 GOTO 102
99.000 C
100.000 122 IF((ISIGN.EQ.-1).OR.(NUMB.NE.0).OR.(IZERO.NE.0)
101.000 +.OR.(IDES.NE.0))INUMB=0
102.000 ISIGN=-1
103.000 GOTO 102
104.000 C
105.000 500 NUM=0
106.000 IF(NUMB.EQ.0)IZERO=1
107.000 GOTO 104
108.000 501 NUM=1
109.000 GOTO 104
110.000 502 NUM=2
111.000 GOTO 104
112.000 503 NUM=3
113.000 GOTO 104
114.000 504 NUM=4
115.000 GOTO 104
116.000 505 NUM=5
117.000 GOTO 104
118.000 506 NUM=6
119.000 GOTO 104
120.000 507 NUM=7

```



```

121.000      GOTO 104
122.000      NUM=8
123.000      GOTO 104
124.000      NUM=9
125.000      104 CONTINUE
126.000      IF(NUMB.GT.3275)GOTO 105
127.000      114 IF(IDET.EQ.10000)GOTO 119
128.000      NUMB=10*NUMB+NUM
129.000      IF(IDES.EQ.1)IDET=10*IDET
130.000      GOTO 102
131.000 C
132.000      119 ANUM=NUM
133.000      IF(1BIG.EQ.2)GOTO 117
134.000      ANUMB=NUMB
135.000      GOTO 115
136.000 C
137.000      105 ANUM=NUM
138.000      IF(1BIG.EQ.1)GOTO 116
139.000      IF(1BIG.EQ.2)GOTO 117
140.000      IF(NUMB.EQ.3276.AND.NUM.LE.7)GOTO 114
141.000      ANUMB=NUMB
142.000      116 IF(IDES.EQ.1)GOTO 115
143.000      ANUMB=10.*ANUMB+ANUM
144.000      1BIG=1
145.000      GOTO 102
146.000 C
147.000      115 ADET=IDET
148.000      1BIG=2
149.000      ANUMB=ANUMB/ADET
150.000      117 ADET=10.*ADET
151.000      ANUMB=ANUMB+ANUM/ADET
152.000      GOTO 102
153.000 C
154.000      118 IND(KOUNT)=1
155.000      IF(1SIGN.LT.1)ANUMB=-ANUMB
156.000      AFL(KOUNT)=ANUMB
157.000      GOTO 121
158.000 C
159.000 C CARRY OUT FINAL COMPUTATION AND STORAGE OF NUMERICAL DATA
160.000      113 IF(1BIG.GE.1)GOTO 118

```

```

161.000      NUMB=NUMB*ISIGN
162.000 C D0 FINAL COMPUTATION AND STORAGE OF FIXED POINT NUMERICAL DATA
163.000      IF1(KOUNT)=NUMB/IDET
164.000      AIDET=IDET
165.000      ANUMB=NUMB
166.000 C D0 FINAL COMPUTATION AND STORAGE OF FLOATING POINT NUMERICAL DATA
167.000      AFL(KOUNT)=ANUMB/AIDET
168.000      IND(KOUNT)=2
169.000 C
170.000 C PERFORM RANGE CHECKS IF REQUIRED
171.000      121 JBNT=JBND(1,KOUNT)+JBND(2,KOUNT)+JBND(3,KOUNT)+JBND(4,KOUNT)
172.000      IF(JBNT.EQ.0)GO TO 524
173.000      IF(JBND(1,KOUNT).EQ.0)GO TO 521
174.000      IF(IF1(KOUNT).LE.IBND(1,KOUNT))GO TO 521
175.000      WRITE(IW,5)IF1(KOUNT)
176.000      GO TO 112
177.000      521 IF(JBND(2,KOUNT).EQ.0)GO TO 522
178.000      IF(IF1(KOUNT).GE.IBND(2,KOUNT))GO TO 522
179.000      WRITE(IW,6)IF1(KOUNT)
180.000      GO TO 112
181.000      522 IF(JBND(3,KOUNT).EQ.0)GO TO 523
182.000      IF(IF1(KOUNT).LE.IBND(3,KOUNT))GO TO 523
183.000      WRITE(IW,7)IF1(KOUNT)
184.000      READ(IR,1)IYN
185.000      IF((IYN.NE.IY).AND.(IYN.NE.IO).AND.(IYN.NE.IO))GO TO 112
186.000      523 IF(JBND(4,KOUNT).EQ.0)GO TO 524
187.000      IF(IF1(KOUNT).GE.IBND(4,KOUNT))GO TO 524
188.000      WRITE(IW,8)IF1(KOUNT)
189.000      READ(IR,1)IYN
190.000      IF((IYN.NE.IY).AND.(IYN.NE.IO).AND.(IYN.NE.IO))GO TO 112
191.000      524 JBNT=JBND(5,KOUNT)+JBND(6,KOUNT)+JBND(7,KOUNT)+JBND(8,KOUNT)
192.000      IF(JBNT.EQ.0)GO TO 528
193.000      IF(JBND(5,KOUNT).EQ.0)GO TO 525
194.000      IF(AFL(KOUNT).LE.FBND(1,KOUNT))GO TO 525
195.000      WRITE(IW,9)AFL(KOUNT)
196.000      GO TO 112
197.000      525 IF(JBND(6,KOUNT).EQ.0)GO TO 526
198.000      IF(AFL(KOUNT).GE.FBND(2,KOUNT))GO TO 526
199.000      WRITE(IW,10)AFL(KOUNT)
200.000      GO TO 112

```

```

201.000
202.000
203.000
204.000
205.000
206.000
207.000
208.000
209.000
210.000
211.000
212.000
213.000
214.000
215.000
216.000
217.000
218.000
219.000
220.000
221.000
222.000
223.000
224.000
225.000
226.000
227.000
228.000
229.000
230.000
231.000
232.000
233.000
234.000
235.000
236.000
237.000
238.000
239.000
240.000

526 IF(JBND(7,KOUNT).EQ.0)GO TO 527
   IF(AFL(KOUNT).LE.FBND(3,KOUNT))GO TO 527
   WRITE(IW,11)AFL(KOUNT)
   READ(IR,1)IYN
   IF((IYN.NE.IY).AND.(IYN.NE.IO).AND.(IYN.NE.IO))GOTO 112
527 IF(JBND(8,KOUNT).EQ.0)GO TO 528
   IF(AFL(KOUNT).GE.FBND(4,KOUNT))GO TO 528
   WRITE(IW,12)AFL(KOUNT)
   READ(IR,1)IYN
   IF((IYN.NE.IY).AND.(IYN.NE.IO).AND.(IYN.NE.IO))GOTO 112
528 CONTINUE

      NUMB=0
      ISIGN=1
      IDES=0
      IDET=1
      IBIG=0

      C
218.000
219.000
220.000
221.000
222.000
223.000
224.000
225.000
226.000
227.000
228.000
229.000
230.000
231.000
232.000
233.000
234.000
235.000
236.000
237.000
238.000
239.000
240.000

C DATA ITEM COUNT FOR BOTH NUMERIC AND ALPHANUMERIC DATA
107 KOUNT=KOUNT+1
   IF(KOUNT.GT.ITOT.OR.KOUNT.GT.6)GOTO 510
   IOUNT=8*KOUNT-7
   ISIZE=0
   IWORD=0
   INUMB=1
   IDES=0
   NUMB=0
   ISIGN=1
   IBIG=0
   IDET=1
   IZERO=0
102 CONTINUE

510 CONTINUE
   IF(IA(1).EQ.IQ.AND. IA(2).EQ.IU.AND. IA(3).EQ.II.AND. IA(4).EQ.IT)
     + GOTO 333
   IF(IA(1).EQ.IF.AND. IA(2).EQ.II.AND. IA(3).EQ.INN) GOTO 333
   GO TO 111
333 WRITE(IW,4)
   STOP

```

```

241.000 C
242.000
243.000
244.000
245.000
246.000
247.000
248.000
249.000
250.000
251.000
252.000
253.000
254.000
255.000 C
256.000
257.000
258.000
259.000
260.000
261.000
262.000
263.000
264.000
265.000
266.000
267.000
268.000
269.000
270.000
271.000
272.000
273.000
274.000
275.000
276.000
277.000
278.000

601 WRITE(IW,13)KKKK
    GOTO 112
111 CONTINUE
    DO 602 J=1,ITOT
        KKK=J
        IF(IND(J).EQ.O.AND.JBNTI(J).GT.O)GOTO 601
602 CONTINUE
    DO 120 I=1,8
    DO 120 J=1,ITOT
120 JBND(I,J)=0
831 CONTINUE
    RETURN

1 FORMAT(13OAI)
3 FORMAT(' NO DATA SUPPLIED. TRY AGAIN')
4 FORMAT(' PROGRAM HAS BEEN INTERRUPTED BY USER')
5 FORMAT(' DATA VALUE JUST SUPPLIED (',I7,') IS TOO LARGE.',/,
+ ', TRY AGAIN')
6 FORMAT(' DATA VALUE JUST SUPPLIED (',I7,') IS TOO SMALL.',/,
+ ', TRY AGAIN')
7 FORMAT(' DATA VALUE JUST SUPPLIED (',I7,') IS UNUSUALLY LARGE.',/,
+ ', DO YOU WISH TO USE IT? Y/N')
8 FORMAT(' DATA VALUE JUST SUPPLIED (',I7,') IS UNUSUALLY SMALL.',/,
+ ', DO YOU WISH TO USE IT? Y/N')
9 FORMAT(' DATA VALUE JUST SUPPLIED (',G12.4,') IS TOO LARGE.',/,
+ ', TRY AGAIN')
10 FORMAT(' DATA VALUE JUST SUPPLIED (',G12.4,') IS TOO SMALL.',/,
+ ', TRY AGAIN')
11 FORMAT(' DATA VALUE JUST SUPPLIED (',G12.4,') IS UNUSUALLY',
+ ', LARGE.',/, ' DO YOU WISH TO USE IT? Y/N')
12 FORMAT(' DATA VALUE JUST SUPPLIED (',G12.4,') IS UNUSUALLY',
+ ', SMALL.',/, ' DO YOU WISH TO USE IT? Y/N')
13 FORMAT(
+ ' THERE IS AN ERROR IN TERM NO.',I3, ' OF INPUT DATA JUST SUPPLIED.'
+ ', PLEASE TRY AGAIN')
    END

```

CHAPTER B2

FORTRAN LISTING OF PROGRAM - ANGELS

1.000 C
2.000 C
3.000 C
4.000 C
5.000 C
6.000 C
7.000 C
8.000 C
9.000 C
10.000 C
11.000 C
12.000 C
13.000 C
14.000 C
15.000 C
16.000 C
17.000 C
18.000
19.000
20.000
21.000
22.000
23.000
24.000
25.000 C
26.000
27.000
28.000
29.000
30.000
31.000
32.000
33.000
34.000
35.000
36.000
37.000
38.000
39.000
40.000

***** ANGELS *****

THREE DIMENSIONAL ANALYSIS OF
BUILDING STRUCTURES
PROGRAMMED BY --- J.U.KHANDOKER
DEPARTMENT OF CIVIL ENGINEERING
CARLETON UNIVERSITY, OTTAWA, ONT
VERSION : 0078-AUG010-AJNAM

COMMON/COM1/BMCL(2490),NT,NN,NSS(10),NBAY(9),NBS(9),H(30)
COMMON/CON/NNS,NFT,NOPT,NQ,NMULP
COMMON/SPEC/NM,LM,XMU,SE,BETA,M1,ALLOW,TD,AG,NP,TC(10),SAC(10)
COMMON/COMSYS/NGR,NWL,NDYN,NSYS,IOUT,SMULP(12,8)
COMMON/MISC/IR,IW,AI(3),AJ(3),IDN,NH
COMMON_SK(12000)
DIMENSION M(20),NDIR(20),DDA(20),NAME(40)
DATA IX,IY,IHX,IHY/

IR=1
IW=7
MCORE=12000
M1=300
ALLOW=0.000001
CALL NAM(IW)
READ(IR,600) (NAME(I),I=1,40)
WRITE(IW,650) (NAME(I),I=1,40)
READ(IR,100) NFT,NFG,NNS,NBA
WRITE(IW,520) NFT,NFG,NNS,NBA
WRITE(IW,135)
READ(IR,130) (H(IK),IK=1,NNS)
WRITE(IW,301)
WRITE(IW,302)

```

41.000 C
42.000 C
43.000 C
44.000
45.000
46.000
47.000
48.000
49.000
50.000
51.000
52.000
53.000
54.000
55.000
56.000
57.000
58.000
59.000
60.000
61.000
62.000
63.000
64.000
65.000
66.000
67.000
68.000
69.000
70.000
71.000
72.000
73.000
74.000
75.000
76.000
77.000
78.000
79.000
80.000

READ AND PRINT FRAME LOCATIONS

DO 10 K=1,NFT
  READ(IR,110) II,M(K),IFD,DDA(K)
  ND=1
  IF(IFD.EQ.IY) ND=2
  NDIR(K)=ND
  WRITE(IW,303) II,M(K),IFD,DDA(K)
10 CONTINUE
  READ(IR,100) NGR,NWL,NDYN,NSYS,IOUT
  WRITE(IW,652) NGR,NWL,NDYN,NSYS
  NJ=NWL+NDYN
  IF(IOUT.EQ.0.AND.NJ.EQ.0) GO TO 11
  READ(IR,100) NOPT
  WRITE(IW,307) NOPT
11 NK=NJ+NGR
  IF(NSYS.EQ.0) GO TO 13
  DO 12 I=1,NSYS
12 READ(IR,130) (SMULP(I,J),J=1,NK)
13 IF(NDYN.EQ.0.AND.IOUT.EQ.0) GO TO 15
  READ(IR,100) NM,LM
  WRITE(IW,312) NM
  WRITE(IW,314) LM
  IF(NDYN.EQ.0) GO TO 15
  READ(IR,100) NP
  DO 14 IP=1,NP
14 READ(IR,130) TC(IP),SAC(IP)
  READ(IR,130) XMU,SF,BETA,TD,AG
  WRITE(IW,304) XMU
  WRITE(IW,315) SF
  WRITE(IW,316) BETA
  BETA=BETA*3.1415962/180.
  CALL SP(IW,2)
  WRITE(IW,331)
  WRITE(IW,318)
  WRITE(IW,331)
  CALL SP(IW,2)
  WRITE(IW,319) NP
  WRITE(IW,332) TD

```

```

81.000
82.000
83.000
84.000
85.000
86.000
87.000
88.000
89.000 C
90.000
91.000
92.000
93.000
94.000
95.000
96.000
97.000
98.000
99.000
100.000
101.000
102.000
103.000
104.000
105.000
106.000
107.000
108.000
109.000
110.000
111.000
112.000
113.000 C
114.000
115.000
116.000
117.000
118.000 C
119.000
120.000 C

WRITE(IW,333) AG
CALL SP(IW,1)
WRITE(IW,321)
WRITE(IW,322) (TC(I),SAC(I),I=1,NP)
15 WRITE(IW,140)
DO 16 IK=1,NNS
  IJ=IK-1
16 WRITE(IW,150) IJ,IK,H(IK)

NBA1=NBA+1
M1=NNS*NBA
M2=M1+NNS
N1=NBA+2
N2=N1+NNS
N3=N2+NBA1
N4=N3+NBA
N5=N4+M1
N6=N5+M1
N7=N6+M1
N8=N7+M1
N9=N8+M1
N10=N9+M1
N11=N10+M1
N12=N11+M2
N13=N12+M2
N14=N13+M2
N15=N14+M2
N16=N15+M2
N17=N16+M2
N18=N17+M2-1
MINCR=N18-MCORE
IF(N18.GT.MCORE) CALL CORE(MINCR)

CALL PROP(NFG,NBA,NBA1,NNS,NQO,NEL,MBM,MCL,SK(1),
+SK(N1),SK(N2),SK(N3),SK(N4),SK(N5),SK(N6),SK(N7),
+SK(N8),SK(N9),SK(N10),SK(N11),SK(N12),SK(N13),SK(N14),
+SK(N15),SK(N16),SK(N17))

IF(NGR.NE.O) WRITE(IW,200)

```



```

121.000
122.000
123.000
124.000
125.000
126.000
127.000
128.000
129.000
130.000
131.000
132.000
133.000
134.000
135.000
136.000
137.000
138.000
139.000
140.000
141.000
142.000
143.000
144.000
145.000
146.000
147.000
148.000
149.000
150.000
151.000
152.000
153.000
154.000
155.000
156.000
157.000
158.000
159.000
160.000

M1=1+NEL
M2=M1+NQQ
M3=M2+NQQ
M4=M3+NQQ
M5=M4+MBM
M6=M5+MBM
M7=M6+MBM
M8=M7+MBM
M9=M8+NNS+NBA-1
MINCR=M9-MCORE
IF(M9.GT.MCORE) CALL CORE(MINCR)

IJ=0
DO 17 I=1,NFG
CALL STIFF(NEL,NQQ,SK(1),SK(M1),SK(M2),NEQ,NE)
JOPT=0
CALL REDUCE(JOPT,NEL,NQQ,SK(1),SK(M1),SK(M2),NEQ,NE)
IF(NGR.EQ.0) GO TO 17
CALL GRAVT(IJ,NGR,NEL,NQQ,NE,NEQ,MBM,NBA,NNS,SK(1),SK(M1),
+SK(M2),SK(M3),SK(M4),SK(M5),SK(M6),SK(M7),SK(M8))
17 CONTINUE
IF(NJ.EQ.0.AND.IOUT.EQ.0) GO TO 90
CALL COND(NFG,SK(1),SK(M1),SK(M2),SK(M3),NQQ,NEL,NEQ,NE)

IF(NOPT.EQ.3) GO TO 18
NQ=NNS
GO TO 20
18 NQ=3*NNS
20 N1=NQ*NQ+1
N2=N1+NNS*NFT
N3=N2+NQ
N4=N3+NNS
N5=N4+NNS
N6=N5+NNS*NNS-1
MINCR=N6-MCORE
IF(N6.GT.MCORE) CALL CORE(MINCR)
IDN=1

CALL STORY(SK(N2),SK(N3),SK(N4))

```

```

161.000 CALL FRAME(SK(1),SK(N1),M,NDIR,SK(N3),SK(N4),SK(N5),DDA)
162.000 C
163.000 REWIND 1
164.000 IF(NWL.NE.O) CALL HLOADY(SK(1),SK(N1),M,NDIR,
165.000 +SK(N3),SK(N4))
166.000 C
167.000 IF(NDYN.EQ.O.AND.IOUT.EQ.O) GO TO 80
168.000 WRITE(IW,300)
169.000 NMULP=1
170.000 IF(NOPT.EQ.3) NMULP=3
171.000 N4=N3+NQ*NM
172.000 N5=N4+NM
173.000 N6=N5+NQ*NM
174.000 N71=N6+NM*NMULP
175.000 N72=N6+NQ
176.000 N7=MAXO(N71,N72)
177.000 MINCR=N7-MCORE-1
178.000 IF(N7.GT.MCORE) CALL CORE(MINCR)
179.000 C
180.000 NH=NM
181.000 CALL SPCTRM(SK(1),SK(N2),SK(N3),SK(N4),SK(N5),SK(N3),
182.000 +SK(N6),SK(N6))
183.000 N5=N4+NNS
184.000 C
185.000 CALL SFORCE(SK(1),SK(N1),M,NDIR,SK(N3),SK(N4),SK(N5))
186.000 C
187.000 80 REWIND 1
188.000 M9=M8+MCL
189.000 M10=M9+MCL
190.000 M11=M10+MCL
191.000 M12=M11+MCL
192.000 MINCR=M12+NNS-1-MCORE
193.000 IF(NWL.EQ.O) GO TO 85
194.000 IF(MINCR.GT.O) CALL CORE(MINCR)
195.000 REWIND 3
196.000 CALL ROOT(NFG,NDIR,NEL,NQQ,MCL,MBM,O,1,SK(1),SK(M1),
197.000 +SK(M2),SK(M3),SK(M4),SK(M5),SK(M6),SK(M7),
198.000 +SK(M8),SK(M9),SK(M10),SK(M11),SK(M12))
199.000 85 IF(NDYN.EQ.O) GO TO 90
200.000 IF(MINCR.GT.O) CALL CORE(MINCR)

```

```

201.000      REWIND 5
202.000      CALL ROOT(NFG,NDIR,NEL,NQQ,MCL,MBM,1,1,SK(1),SK(M1),
203.000      +SK(M2),SK(M3),SK(M4),SK(M5),SK(M6),SK(M7),
204.000      +SK(M8),SK(M9),SK(M10),SK(M11),SK(M12))
205.000 C
206.000      90 IF(NSYS.EQ.0) GO TO 99
207.000      M1=4*MBM
208.000      N1=NQQ+1
209.000      N2=N1+MBM
210.000      N3=N2+MBM
211.000      N4=N3+MBM
212.000      N5=N4+MBM
213.000      N6=N5+MBM
214.000      N7=N6+MBM
215.000      N8=N7+MBM
216.000      N9=N8+MBM
217.000      N10=N9+MCL
218.000      N11=N10+MCL
219.000      N12=N11+MCL
220.000      N13=N12+MCL
221.000      N14=N13+M1
222.000      N15=N14+M1
223.000      N16=N15+M1
224.000      N17=N16+M1
225.000      N18=N17+7*NQQ
226.000      N19=N18+NNS-1
227.000      MINCR=N19-MCORE
228.000      IF(MINCR.GT.0) CALL CORE(MINCR)
229.000 C
230.000      CALL SYSTEM(NFG,NQQ,MBM,MCL,NNS,SK(1),SK(N1),SK(N2),
231.000      +SK(N3),SK(N4),SK(N5),SK(N6),SK(N7),SK(N8),SK(N9),
232.000      +SK(N10),SK(N11),SK(N12),SK(N13),SK(N14),SK(N15),SK(N16),
233.000      +SK(N17),SK(N18))
234.000 C
235.000      99 STOP
236.000 C
237.000      100 FORMAT(5I5)
238.000      110 FORMAT(2I5,4X,A1,F15.5)
239.000      120 FORMAT(5F15.5)
240.000      130 FORMAT(8F10.3)

```

```

241.000 135 FORMAT(/// *** UNITS ARE CONSISTENT//)
242.000 140 FORMAT(///
243.000 + STORY HEIGHTS, C/C OF//
244.000 + INTERSECTING FLOOR BEAMS//
245.000 +
246.000 + BETWEEN STORY//
247.000 + STORY LEVELS HEIGHTS//)
248.000 150 FORMAT(2X,I3, AND,I3,F12.3)
249.000 200 FORMAT(IH1,/
250.000 +
251.000 + DATA FOR GRAVITY LOAD CASES//
252.000 +
253.000 300 FORMAT(IH1,/
254.000 +
255.000 + DYNAMIC ANALYSIS//
256.000 +
257.000 301 FORMAT(/2X,
258.000 +2X, FRAME POSITION DATA//
259.000 +2X,
260.000 302 FORMAT(2X, FRAME, 4X, FRAME GROUP, 4X, DIRECTION, 4X,
261.000 + DIST FROM ORIGIN//)
262.000 303 FORMAT(1X, I4, I12, I3X, A1, F18.2)
263.000 304 FORMAT(2X, DUCTILITY FACTOR ... ..
264.000 + ... =, F6.2)
265.000 307 FORMAT(2X, STORY TRANSLATION CODE . ... ..
266.000 + ... =, I3)
267.000 312 FORMAT(2X, NUMBER OF MODES TO BE CONSIDERED ... ..
268.000 + ... =, I3)
269.000 314 FORMAT(2X, NUMBER OF MODES TO BE EXTRACTED FROM ONE ORI,
270.000 + GIN =, I3)
271.000 315 FORMAT(2X, SCALE FACTOR FOR DESIGN GROUND ACCELERATION,
272.000 + ... =, F6.2)
273.000 316 FORMAT(2X, ANGLE OF INCIDENCE OF EARTHQUAKE ... ..
274.000 + ... =, F6.2)
275.000 318 FORMAT(1X, ACCELERATION SPECTRUM)
276.000 319 FORMAT(1X, NUMBER OF PERIOD COORDS .....=, I8)
277.000 321 FORMAT(1X, PERIOD, 6X, ACCELERATION//)
278.000 322 FORMAT(1X, F8.3, F13.3)
279.000 331 FORMAT(1X,
280.000 332 FORMAT(1X, PERIOD BOUND.....=, F10.2)

```

```

281.000 333 FORMAT(1X, ' ACCELERATION DUE TO GRAVITY...', F10.2)
282.000 520 FORMAT(2X, ' TOTAL NO. OF FRAMES IN THE BUILDING ...',
283.000 +, ' ... =', I3, /
284.000 +2X, ' TOTAL NO. OF FRAME GROUPS .. ... ..',
285.000 +, ' ... =', I3, /
286.000 +2X, ' TOTAL NO. OF STORIES IN THE BUILDING ....',
287.000 +, ' ... =', I3, /
288.000 +2X, ' MAX. NO. OF BAYS IN THE WIDEST FRAME ...',
289.000 +, ' ... =', I3, /
290.000 600 FORMAT(40A1)
291.000 650 FORMAT(//19X, ' NAME OF PROJECT :: ', 40A1//)
292.000 652 FORMAT(//
293.000 +, ' NO. OF GRAVITY LOAD CASES ... .. =', I3, /
294.000 +, ' NO. OF LATERAL LOAD CASES ... .. =', I3, /
295.000 +, ' NO. OF SEISMIC LOAD CASES ... .. =', I3, /
296.000 +, ' NO. OF LOAD SYSTEMS (LOAD COMBINATIONS) .. =', I3, /
297.000 END
298.000 C
299.000 SUBROUTINE NAM(IW)
300.000 C
301.000 C WRITES TITLE OF THE PROGRAM ANGELS
302.000 C
303.000 WRITE(IW, 50)
304.000 WRITE(IW, 100)
305.000 WRITE(IW, 200)
306.000 WRITE(IW, 60)
307.000 WRITE(IW, 300)
308.000 RETURN
309.000 C
310.000 50 FORMAT(1H1//, ' ,79(1H=), // ,7(1H*)', ' ANALYSIS OF BUILDINGS',
311.000 +, ' FOR GRAVITY, EARTHQUAKE AND LATERAL LOADS ', 7(1H*), /
312.000 +, ' ,79(1H=))
313.000 60 FORMAT(//, ' ,79(1H=), // ,7(1H*)', ' ANALYSIS OF BUILDINGS',
314.000 +, ' FOR GRAVITY, EARTHQUAKE AND LATERAL LOADS ', 7(1H*), /
315.000 +, ' ,79(1H=))
316.000 100 FORMAT(//4X6(1HA)4X, 3(1HN)6X3(1HN)5X6(1HG)5X12(1HE),
317.000 +1X3(1HL)13X6(1HS)/3X8(1HA)3X4(1HN)5X3(1HN)3X
318.000 +10(1HG)3X12(1HE)1X3(1HL)11X10(1HS)1X/2X3(1HA),
319.000 +4X3(1HA)2X5(1HN)4X3(1HN)2X3(1HG)1X4X1X3(1HG)2X
320.000 +3(1HE)10X3(1HL)10X3(1HS)6X3(1HS)/1X3(1HA)6X3(1HA),

```

```

321.000 +IX6(IHN)3X3(IHN)1X3(1HG)8X3(1HG)1X3(1HE)10X3(1HL)10X3(1HS)1X
322.000 200 FORMAT(1X3(1HA)6X3(1HA)2(1X3(1HN))2X3(1HN)1X3(1HG)12X9(1HE),
323.000 +4X3(1HL)11X8(1HS)1X/1X12(1HA)1X3(1HN)1X2(1X3(1HN))1X
324.000 +3(1HG)4X,7(1HG)1X9(1HE)4X3(1HL)13X8(1HS)1X/1X12(1HA),
325.000 +1X3(1HN)3X6(1HN)1X3(1HG)4X7(1HG)1X3(1HE)10X3(1HL)19X
326.000 +3(1HS)/1X3(1HA)6X3(1HA)1X3(1HN)4X5(1HN)2X3(1HG)
327.000 +6X3(1HG)1X1X3(1HE)10X3(1HL)10X3(1HS)1X4X1X3(1HS)/
328.000 +1X3(1HA)6X3(1HA)1X3(1HN)5X4(1HN)3X10(1HG)3X12(1HE),
329.000 +1X12(1HL)2X10(1HS)1X/1X3(1HA)6X3(1HA)1X3(1HN)6X3(1HN),
330.000 +5X6(1HG)1X4X12(1HE)1X12(1HL)4X6(1HS)1X/
331.000 300 FORMAT(/,----- PROGRAMMED BY --- J. U. KHANDOKER,
332.000 + AND J. L. HUMAR -----/,
333.000 +----- DEPARTMENT OF CIVIL ENGINEERING, CARLETON,
334.000 + UNIVERSITY, OTTAWA, CANADA -----/)
335.000 END
336.000 C
337.000 SUBROUTINE PROP(NFG,NBA,NBA1,NNN,NQQ,NEL,MBM,
338.000 +MCL,NSC,IBC,ICC,B,AB,BE,BMG,BK,BI,DB,DBX,AC,CE,CLG,
339.000 +CK,CI,DC,DCX)
340.000 C
341.000 C
342.000 C
343.000 COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
344.000 +CL(240),CA(240),CB(240),CG(240),CEF(240),CAL(240),NT,
345.000 +NNS,NSS(10),NBAY(9),NBS(9),H(30)
346.000 C
347.000 DIMENSION NSC(NBA1),IBC(NNN),ICC(NBA1),B(NBA),AB(NNN,NBA),
348.000 +BE(NNN,NBA),BMG(NNN,NBA),BK(NNN,NBA),BI(NNN,NBA),
349.000 +DB(NNN,NBA),DBX(NNN,NBA),AC(NNN,NBA1),CE(NNN,NBA1),
350.000 +CLG(NNN,NBA1),CK(NNN,NBA1),CI(NNN,NBA1),DC(NNN,NBA1),
351.000 +DCX(NNN,NBA1)
352.000 C
353.000 DATA IRR/1HR/
354.000 IRI=2
355.000 IW1=4
356.000 IW=7
357.000 WRITE(IW,501)
358.000 IBI=0
359.000 NEI=0
360.000 DO 200 II=1,NFG

```

```

361.000 WRITE(IW,502) II
362.000 READ(IR1,503) NT,NNS,NFR
363.000 WRITE(IW1,503) NT,NNS,NFR
364.000 READ(IR1,503) (NSS(J),J=1,NT)
365.000 WRITE(IW1,503) (NSS(J),J=1,NT)
366.000 READ(IR1,503) (NBAY(J),J=1,NT)
367.000 WRITE(IW1,503) (NBAY(J),J=1,NT)
368.000 WRITE(IW,505) NT,NNS
369.000 WRITE(IW,506)
370.000 WRITE(IW,507) (I,NBAY(I),NSS(I),I=1,NT)
371.000 IF(NT.EQ.1) GO TO 8
372.000 NT=NT-1
373.000 READ(IR1,503) (NBS(J),J=1,NT1)
374.000 WRITE(IW1,503) (NBS(J),J=1,NT1)
375.000 WRITE(IW,508)
376.000 DO 5 I=1,NT1
377.000 J=I+1
378.000 5 WRITE(IW,509) I,J,NBS(I)
379.000 8 NB=NBAY(I)
380.000 READ(IR1,504) (B(K),K=1,NB)
381.000 WRITE(IW,510)
382.000 WRITE(IW,511) (J,B(J),J=1,NB)
383.000 READ(IR1,503) OSTB
384.000 WRITE(IW,512)
385.000 IS=0
386.000 IB=0
387.000 NJ=0
388.000 DO 50 I=1,NT
389.000 NN=NSS(I)
390.000 NB=NBAY(I)
391.000 DO 40 J=1,NN
392.000 IS=IS+1
393.000 READ(IR1,513) IBC(IS)
394.000 DO 30 K=1,NB
395.000 IB=IB+1
396.000 NI=NJ+K
397.000 IF(IBC(IS),NE.IRR) GO TO 20
398.000 READ(IR1,514) BE(IS,NI),BMG(IS,NI),BK(IS,NI),DBX(IS,NI),DB(IS,NI)
399.000 IF(K.NE.1) GO TO 18
400.000 WRITE(IW,515) IS,NI,IBC(IS),DBX(IS,NI),DB(IS,NI),BE(IS,NI),

```

```

401.000      +BMG(IS,NI),BK(IS,NI)
402.000      GO TO 30
403.000  18  WRITE(IW,522) NI,IBC(IS),DBX(IS,NI),DB(IS,NI),BE(IS,NI),
404.000      +BMG(IS,NI),BK(IS,NI)
405.000      GO TO 30
406.000  20  READ(IRI,514) BE(IS,NI),BMG(IS,NI),BK(IS,NI),DB(IS,NI),
407.000      +AB(IS,NI),BI(IS,NI)
408.000      IF(K.NE.1) GO TO 25
409.000      WRITE(IW,516) IS,NI,IBC(IS),DB(IS,NI),AB(IS,NI),BI(IS,NI),
410.000      +BE(IS,NI),BMG(IS,NI),BK(IS,NI)
411.000      GO TO 30
412.000  25  WRITE(IW,523) NI,IBC(IS),DB(IS,NI),AB(IS,NI),BI(IS,NI),
413.000      +BE(IS,NI),BMG(IS,NI),BK(IS,NI)
414.000  30  CONTINUE
415.000  40  CONTINUE
416.000      NJ=NJ+NBS(I)
417.000  50  CONTINUE
418.000      NBM=IB
419.000      NCL=IB+NNS
420.000      WRITE(IW,517)
421.000      NB=NBAY(I)+1
422.000      DO 75 I=1,NB
423.000      READ(IRI,518) ICC(I),NSC(I)
424.000      NN=NSC(I)
425.000      DO 70 J=1,NN
426.000      IF(ICC(I).NE.IRR) GO TO 60
427.000      READ(IRI,514) CE(J,I),CLG(J,I),CK(J,I),DCX(J,I),DC(J,I)
428.000      IF(J.NE.1) GO TO 55
429.000      WRITE(IW,515) I,J,ICC(I),DCX(J,I),DC(J,I),CE(J,I),CLG(J,I),CK(J,I)
430.000      GO TO 70
431.000  55  WRITE(IW,522) J,ICC(I),DCX(J,I),DC(J,I),CE(J,I),
432.000      +CLG(J,I),CK(J,I)
433.000      GO TO 70
434.000  60  READ(IRI,514) CE(J,I),CLG(J,I),CK(J,I),DCX(J,I),DC(J,I),AC(J,I),CI(J,I)
435.000      IF(J.NE.1) GO TO 65
436.000      WRITE(IW,516) I,J,ICC(I),DCX(J,I),AC(J,I),CI(J,I),CE(J,I),
437.000      +CLG(J,I),CK(J,I)
438.000      GO TO 70
439.000  65  WRITE(IW,523) J,ICC(J),DC(J,I),AC(J,I),CI(J,I),CE(J,I),
440.000      +CLG(J,I),CK(J,I)

```



```

441.000
442.000
443.000
444.000
445.000
446.000
447.000
448.000
449.000
450.000
451.000
452.000
453.000
454.000 C
455.000 C
456.000 C
457.000 C
458.000
459.000
460.000
461.000
462.000
463.000
464.000
465.000
466.000
467.000
468.000
469.000
470.000
471.000
472.000
473.000
474.000
475.000
476.000
477.000
478.000
479.000
480.000

70 CONTINUE
75 CONTINUE
  NEQ=2*NCL+NNS
  WRITE(IW,503) NBM,NCL,NEQ
  IF(OSTB.EQ.0) GO TO 76
  WRITE(IW,600)
  GO TO 77
76 WRITE(IW,610)
77 IF(BMG(1,1).EQ.0.0.OR.BK(1,1).EQ.0.0) GO TO 78
  WRITE(IW,620)
  GO TO 79
78 WRITE(IW,630)
79 CONTINUE

  CALCULATE BEAM AND COLUMN LENGTHS AND END STUBS
  OSTB=0 MEANS FINITE JOINT SIZES ARE NOT CONSIDERED
  IF(OSTB.EQ.0) GO TO 100
  IB=0
  IC=0
  NJ=0
  JS=1
  NB=NBAY(I)+1
  DO 80 I=1,NB
    CA(I)=0.0
    DO 98 I=1,NT
      NS=NSS(I)
      NB=NBAY(I)+1
      NJ1=NB*(I)
      NJ2=NJ1+NBAY(I+1)+1
      IF(I.EQ.NT) NJ1=NB
      DO 95 J=1,NS
        D2=0.0
        IS=JS
        JS=JS+1
        JC=IC+NB
        IF(J.EQ.NS) JC=JC-NJ1
        DO 90 K=1,NB
          NI=NJ+K
          NK=NI-1

```

```

481.000      JB=IB
482.000      IB=IB+1
483.000      IC=IC+1
484.000      JC=JC+1
485.000      IF(K.NE.NB) GO TO 81
486.000      BB(JB)=DC(IS,NI)/2.
487.000      CB(IC)=DB(IS,NK)/2.
488.000      GO TO 85
489.000      81 BA(IB)=DC(IS,NI)/2.
490.000      D1=DB(IS,NI)/2.
491.000      IF(K.EQ.1) GO TO 82
492.000      D2=DB(IS,NK)/2.
493.000      BB(JB)=BA(IB)
494.000      CB(IC)=MAX1(D1,D2)
495.000      82 IF(J.NE.NS) GO TO 86
496.000      85 IF(K.LE.NJ1.OR.K.GT.NJ2) GO TO 87
497.000      86 CA(JC)=CB(IC)
498.000      87 IF(K.EQ.1) GO TO 88
499.000      BL(JB)=B(NK)-BA(JB)-BB(JB)
500.000      88 CL(IC)=H(IS)-CA(IC)-CB(IC)
501.000      90 CONTINUE
502.000      IB=JB
503.000      95 CONTINUE
504.000      NJ=NJ+NJ1
505.000      98 CONTINUE
506.000      C
507.000      C
508.000      C
509.000      100 IS=0
510.000      IB=0
511.000      NJ=0
512.000      DO 120 I=1,NI
513.000      NS=NSS(I)
514.000      NB=NBAY(I)
515.000      DO 110 J=1,NS
516.000      IS=IS+1
517.000      DO 105 K=1,NB
518.000      NI=NJ+K
519.000      IB=IB+1
520.000      IF(OSTB.NE.O) GO TO 101

```

PREPARE DATA FOR PROGRAMS STIFF AND LOAD

```

521.000      BA(IB)=0.
522.000      BB(IB)=0.
523.000      BL(IB)=B(NI)
524.000 101 IF(IBC(IS).NE.IRR) GO TO 102
525.000      AB(IS,NI)=DBX(IS,NI)*DBX(IS,NI)
526.000      BI(IS,NI)=DBX(IS,NI)*(DB(IS,NI)**3/12.0
527.000 102 IF(BMG(IS,NI).EQ.0.0.OR.BK(IS,NI).EQ.0.) GO TO 103
528.000      BG(IB)=6.*BE(IS,NI)*BI(IS,NI)/(BK(IS,NI)*AB(IS,NI)*
529.000      +BMG(IS,NI)*(BL(IB)**2)
530.000      BEF(IB)=2.*BE(IS,NI)*BI(IS,NI)/(BL(IB)*(1.+2.*BG(IB)))
531.000      GO TO 104
532.000 103 BG(IB)=0.0
533.000      BEF(IB)=2.*BE(IS,NI)*BI(IS,NI)/BL(IB)
534.000 104 WRITE(IW1,520) BL(IB),BA(IB),BB(IB),BG(IB),BEF(IB)
535.000 105 CONTINUE
536.000 110 CONTINUE
537.000      NJ=NJ+NBS(I)
538.000 120 CONTINUE
539.000      C
540.000      IS=0
541.000      IC=0
542.000      NJ=0
543.000      DO 150 I=1,NT
544.000      NS=NSS(I)
545.000      NB=NBAY(I)+1
546.000      DO 140 J=1,NS
547.000      IS=IS+1
548.000      DO 130 K=1,NB
549.000      NI=NJ+K
550.000      IC=IC+1
551.000      IF(OSTB.NE.0) GO TO 122
552.000      CA(IC)=0.
553.000      CB(IC)=0.
554.000      CL(IC)=H(IS)
555.000 122 IF(ICC(NI).NE.IRR) GO TO 125
556.000      AC(IS,NI)=DCX(IS,NI)*DCX(IS,NI)
557.000      CI(IS,NI)=DCX(IS,NI)*(DC(IS,NI)**3/12.
558.000      -CAL(IC)=AC(IS,NI)*CE(IS,NI)/CL(IC)
559.000      IF(CLG(IS,NI).EQ.0.0.OR.CK(IS,NI).EQ.0.) GO TO 127
560.000      CG(IC)=6.*CE(IS,NI)*CI(IS,NI)/

```

```

561.000      +(CK(IS,NI)*AC(IS,NI)*CLG(IS,NI)*(CL(IC))**2)
562.000      CEF(IC)=2.*CE(IS,NI)*CI(IS,NI)/(CL(IC))*(1.+2.*CG(IC)))
563.000      GO TO 128
564.000  127 CG(IC)=0.0
565.000      CEF(IC)=2.*CE(IS,NI)*CI(IS,NI)/CL(IC)
566.000  128 WRITE(IWI,520) CL(IC),CA(IC),CB(IC),CG(IC),CEF(IC),CAL(IC)
567.000  130 CONTINUE
568.000  140 CONTINUE
569.000      NJ=NJ+NBS(I)
570.000  150 CONTINUE
571.000      C
572.000  151 FIND MAX. NO. OF BEAMS(MBM), MAX. NO. OF COLUMNS(MCL),
573.000  152 MAX NO.OF EQNS.(NQQ) AND MAX NO.OF STIFFNESS ELEMENTS(NEL)
574.000  153 C
575.000      IF(NBM.GT.IB1) GO TO 155
576.000      NBM=IB1
577.000      GO TO 160
578.000  155 MBM=NBMM
579.000      MCL=NCL
580.000      NQQ=NEQ
581.000      IB1=NBMM
582.000  160 JTN=0
583.000      JTR=0
584.000      DO 180 I=1,NT
585.000      NB=NBAY(I)
586.000      NN=NSS(I)
587.000      JA=2*NB+4
588.000      JB=3*NB+5
589.000      JC=NB+1
590.000      JY=2*JA*JC+JB
591.000      JTN=JTN+JY*(NN-1)
592.000      IF(I.EQ.1) GO TO 165
593.000      NSB=NBS(I-1)
594.000      JP=JAI-NSB
595.000      JQ=JP+JC
596.000      JR=NB+NB1-NSB+4
597.000      JT=JC*(JP+JR)+JQ
598.000      JTR=JTR+JT
599.000      GO TO 170
600.000

```

```

601.000 165 JX=(JA-NB)*NB+2*(JA-1)
602.000 170 JA1=JA
603.000 NBI=NB
604.000 180 CONTINUE
605.000 NEL=JX+JTN+JTR
606.000 IF(NEL.GT.NEL) NEL=NEL
607.000 NE1=NEL
608.000 C
609.000 200 CONTINUE
610.000 WRITE(IW,525)
611.000 REWIND IWI
612.000 RETURN
613.000 C
614.000 501 FORMAT(//2X,-----,/)
615.000 +,2X,FRAME PROPERTIES',/
616.000 +2X,-----,/)
617.000 502 FORMAT(//
618.000 +,-----,/)
619.000 +, FRAME GROUP NO.,I3,/,
620.000 +,-----,/)
621.000 503 FORMAT(5I5)
622.000 504 FORMAT(8F10.3)
623.000 505 FORMAT(//2X,NO. OF TOWERS =,I3,/,
624.000 +2X,NO.OF STORIES =,I3)
625.000 506 FORMAT(/2X,TOWER NO. OF NO.OF,/,
626.000 +3X,NO. BAYS STORIES,/)
627.000 507 FORMAT(3X,I2,8X,I2,10X,I2)
628.000 508 FORMAT(/2X,NO. OF BAYS-SETBACK AT LEFT,/,
629.000 +,-----,/)
630.000 509 FORMAT(2X,BETWN TOWERS,I3, AND,I3, =,I3)
631.000 510 FORMAT(/2X,LENGTHS OF BAYS C/C OF,/,
632.000 +2X,COLS. STARTING AT LEFT,/,
633.000 +2X,-----,/)
634.000 +4X,BAY NO.,5X,LENGTH,/)
635.000 511 FORMAT(18,4X,F10.2)
636.000 512 FORMAT(//2X,BEAM PROPERTIES',/
637.000 +2X,-----,/)
638.000 +, STORY BAY TYPE WIDTH DEPTH AREA INERTIA',
639.000 +, ELASTICITY RIGIDITY SH.COEFF,
640.000 513 FORMAT(1X,A1)

```

```

641.000 514 FORMAT(6E12.5)
642.000 515 FORMAT(/2I5,4X,A1,F9.2,F8.2,18X,2E11.5,F8.3)
643.000 516 FORMAT(/2I5,4X,A1,F9.2,8X,F8.2,F10.2,2E11.5,F8.3)
644.000 517 FORMAT(/2X, COLUMN PROPERTIES, /
645.000 +2X, -----//
646.000 +4 COLM, 13X, CROSS, /
647.000 +4 LINE STORY TYPE WIDTH WIDTH AREA INERTIA,
648.000 +4 ELASTICITY RIGIDITY SH.COEF, )
649.000 518 FORMAT(1X,A1,J5)
650.000 520 FORMAT(6E12.5)
651.000 522 FORMAT(110,4X,A1,F9.2,F8.2,18X,2E11.5,F8.3)
652.000 523 FORMAT(110,4X,A1,F9.2,8X,F8.2,F10.2,2E11.5,F8.3)
653.000 525 FORMAT(/, NOTE:- BEAM/COL TYPE: R=RECT. SECTION, ,
654.000 +4 0= NOT A RECT. SECTION, /)
655.000 600 FORMAT(/, ** FINITE JOINT SIZES ARE CONSIDERED **, )
656.000 610 FORMAT(/, ** FINITE JOINT SIZES ARE NOT CONSIDERED **, )
657.000 620 FORMAT(/, ** SHEAR DEFORMATIONS ARE CONSIDERED **, )
658.000 630 FORMAT(/, ** SHEAR DEFORMATIONS ARE NOT CONSIDERED **, )
659.000 END
660.000 C
661.000 C
662.000 C
663.000 C
664.000 C
665.000 C
666.000 C
667.000 C
668.000 C
669.000 C
670.000 C
671.000 C
672.000 C
673.000 C
674.000 C
675.000 C
676.000 C
677.000 C
678.000 C
679.000 C
680.000 C

SUBROUTINE STIFF(NEL,NQQ,AK,NDA,ICD,NEQ,NE)

THIS PROGRAM ASSEMBLES THE FRAME STIFFNESS MATRICES

COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
+CL(240),CA(240),CB(240),CG(240),CEF(240),CAL(240),NT,NS,
+NST(10),NBAY(9),NBS(9),HC(30)
DIMENSION AK(NEL),NDA(NQQ),ICD(NQQ)

READS FRAME PROPERTIES

IR=4
IW=5.
READ(IR,520) NT,NS
READ(IR,520) (NST(I),I=1,NT)
READ(IR,520) (NBAY(I),I=1,NT)
NT2=NT-1
IF(NT.GT.1) READ(IR,520) (NBS(I),I=1,NT2)
READ(IR,520) NBM,NCL
READ(IR,515) (BL(I),BA(I),BB(I),BG(I),BEF(I),I=1,NBM)

```

```

681.000 C
682.000 C
683.000 C
684.000 C
685.000
686.000
687.000
688.000 C
689.000 C
690.000 C
691.000
692.000
693.000
694.000
695.000
696.000
697.000
698.000
699.000
700.000
701.000
702.000
703.000
704.000
705.000
706.000
707.000
708.000
709.000
710.000
711.000
712.000
713.000
714.000
715.000
716.000
717.000
718.000
719.000
720.000

READ(IR,510) (CL(I),CA(I),CB(I),CG(I),CEF(I),CAL(I),I=1,NCL)
INITIALIZES STIFFNESS ELEMENTS
DO 6 I=1,NEL
  AK(I)=0.
6 CONTINUE

ASSEMBLES STIFFNESS FOR THE FIRST FLOOR
IB=0
ID=0
K=0.
IC=0
NB=NBAY(1)
NB2=NB+1
NBS1=NBS(1)
NB1=NBAY(2)
NB3=NB1+1
NBS2=NBS1+NB3
JC=NB2
IF(NST(1).EQ.1) JC=JC-NBS1
KH=2*NB2-1
KV=KH+NB2+1
IDV=NB2+1
SUM=0.
DO 50 IN=1,NB2
  JB=IB
  IB=IB+1
  IC=IC+1
  JC=JC+1
  K=K+1
  KH=KH+1
  KV=KV+1
  KVI=KV+NB+3
  IF(IN.EQ.1) KVI=KVI-1
  KV2=KVI-1
  X2=0.
  X3=0.
  X4=0.

```

```

721.000
722.000
723.000
724.000
725.000
726.000
727.000
728.000
729.000
730.000
731.000
732.000
733.000
734.000
735.000
736.000
737.000
738.000
739.000
740.000
741.000
742.000
743.000
744.000
745.000
746.000
747.000
748.000
749.000
750.000
751.000
752.000
753.000
754.000
755.000
756.000
757.000
758.000
759.000
760.000

X5=0.
Y=0.
Y1=0.
Z1=0.
Z2=0.
Z3=0.
Z4=0.
Z5=0.
Z6=0.
Z8=0.
X1=CST(IC,3)
Y2=CST(IC,5)
Y3=CST(IC,6)
Z7=CAL(IC)
IF(NS.EQ.1) GO TO 20
IF(NST(1).NE.1) GO TO 10
IF(IN.LE.NBS1.OR.IN.GT.NBS2) GO TO 20
10 X2=CST(JC,1)
Y=CST(JC,4)
Y1=CST(JC,6)
Z8=CAL(JC)
20 IF(IN.EQ.1) GO TO 30
X3=BST(JB,2)
X4=BST(JB,3)
AK(K)=X3
K=K+1
Z1=BST(JB,4)
Z2=BST(JB,6)
Z3=BST(JB,5)
AK(KV)=-Z1
KV=KV+1
IF(IN.EQ.NB2) GO TO 40
30 X5=BST(IB,1)
Z4=BST(IB,4)
Z5=BST(IB,6)
Z6=BST(IB,5)
40 AK(K)=X1+X2+X4+X5
AK(KH)=Y2-Y
SUM=SUM+Y1+Y3
AK(KV)=-Z3+Z4

```



```

761.000      KV=KV+1
762.000      AK(KV)=Z6
763.000      AK(KV2)=-Z2
764.000      AK(KV1)=Z2+Z5+Z7+Z8
765.000      ID=ID+1
766.000      IDV=IDV+1
767.000      NDA(ID)=K
768.000      NDA(IDV)=KV1
769.000      ICD(ID)=0
770.000      ICD(IDV)=0
771.000      KV=KV1
772.000      50 CONTINUE
773.000      KH=KH+1
774.000      AK(KH)=SUM
775.000      ID=ID+1
776.000      NDA(ID)=KH
777.000      ICD(ID)=1
778.000      ID=IDV
779.000      K=KV1
780.000      IF(NS.EQ.1) GO TO 499
781.000      IB=JB
782.000      NS1=2
783.000      C
784.000      DO 200 IT=1,NT
785.000      NST1=NS1(IT)
786.000      NB2=NB+1
787.000      NK=2*NB+3
788.000      IF(IT.NE.1) GO TO 104
789.000      IF(NST1.EQ.1) GO TO 160
790.000      104 IF(IT.EQ.NT) GO TO 105
791.000      NB1=NBAY(IT+1)
792.000      NB3=NB1+1
793.000      NBS1=NBS(IT)
794.000      NBS2=NB'S1+NB3
795.000      GO TO 106
796.000      105 NBS1=NB2
797.000      106 NN=NK+1
798.000      NNH=NN*NB2
799.000      NNV=NNH+NN+NB2
800.000      C

```

ASSEMBLES STIFFNESSES FOR TYPICAL FLOORS

```

801.000 C
802.000 C
803.000
804.000
805.000
806.000
807.000
808.000
809.000
810.000
811.000
812.000
813.000
814.000
815.000
816.000
817.000
818.000
819.000
820.000 C
821.000
822.000
823.000
824.000
825.000
826.000
827.000
828.000
829.000
830.000
831.000
832.000
833.000
834.000
835.000
836.000
837.000
838.000
839.000
840.000

D0 155 J=NS1,NST1
IF(J.NE.2) GO TO 110
N1=NK
N2=NB+2
N3=NK
N4=NB2
NH=NNH
NV=NNV
IH=O
110 JC=IC+NB2
IF(J.EQ.NST1) JC=JC-NBS1
IDV=ID+NB2+1
KH=K+NH
KV=K+NV
KH1=KH+NI
SUM=O.
SUM1=O.

D0 150 IN=1,NB2
NI=NI+1
J8=IB
IB=IB+1
IC=IC+1
JC=JC+1
K=K+1
KH=KH+1
KH1=KH1+1
KV=KV+1
KI=K+NI
K2=K+N2-IN
KV1=KV+N3
KV2=KV+N4
K3=K1-1
KV3=KV2-1
KV4=KV2+1
KV5=KV1-1
X4=O.
X5=O.

```

```

841.000      X6=0.
842.000      X7=0.
843.000      Y3=0.
844.000      Y4=0.
845.000      Z2=0.
846.000      Z3=0.
847.000      Z4=0.
848.000      Z5=0.
849.000      Z6=0.
850.000      Z7=0.
351.000      Z8=0.
852.000      X1=CST(IC,2)
853.000      X2=CST(IC,3)
854.000      X3=CST(IC,5)
855.000      Y1=CST(IC,4)
856.000      Y2=CST(IC,6)
857.000      Z1=CAL(IC)
858.000      IF(J.NE.NST1) GO TO 115
859.000      IF(IN.LE.NBS1.OR.IN.GT.NBS2) GO TO 120
860.000      115 X4=CST(JC,1)
861.000          Y3=CST(JC,4)
862.000          Y4=CST(JC,6)
863.000          Z2=CAL(JC)
864.000      120 IF(IN.EQ.1) GO TO 130
865.000          X5=BSI(JB,2)
866.000          X6=BSI(JB,3)
867.000          Z3=BSI(JB,4)
868.000          Z4=BSI(JB,6)
869.000          Z5=BSI(JB,5)
870.000      IF(IN.EQ.NB2) GO TO 140
871.000      130 X7=BSI(IB,1)
872.000          Z6=BSI(IB,4)
873.000          Z7=BSI(IB,6)
874.000          Z8=BSI(IB,5)
875.000      140 AK(K)=X1
876.000          AK(K2)=-X3
877.000          AK(K3)=X5
878.000          AK(K1)=X2+X4+X6+X7
879.000          AK(KH)=Y1
880.000          AK(KH1)=X3-Y3

```

```

881.000 SUM=SUM-Y2
882.000 SUM1=SUM1+Y2+Y4
883.000 AK(KV)=-Z1
884.000 AK(KV2)=-Z5+Z6
885.000 AK(KV3)=-Z3
886.000 AK(KV4)=Z8
887.000 AK(KV5)=-Z4
888.000 AK(KV1)=Z1+Z2+Z4+Z7
889.000 ID=ID+1
890.000 IDV=IDV+1
891.000 NDA(ID)=K1
892.000 NDA(IDV)=KV1
893.000 ICD(ID)=0
894.000 ICD(IDV)=0
895.000 K=K1
896.000 KV=KV1
150 CONTINUE
897.000 KH=KH+IH+1
898.000 KH1=KH1+1
899.000 AK(KH)=SUM
900.000 AK(KH1)=SUM1
901.000 ID=ID+1
902.000 NDA(ID)=KH1
903.000 ICD(ID)=1
904.000 ID=IDV
905.000 K=KV1
906.000 IB=JB
907.000
908.000 155 CONTINUE
909.000 IF(IT.EQ.NT) GO TO 200
910.000 C
911.000 160 N1=NK-NBS1
912.000 N2=NB-NBS1+2
913.000 N3=NB+NB1+3-NBS1
914.000 N4=NB2-NBS1
915.000 NN=NK-NBS1+1
916.000 NH=NN+NB3
917.000 NV=NH+NN+NB3
918.000 IH=NB-NB1-NBS1
919.000 NS1=1
920.000 NB=NB1

```

```

921.000      200 CONTINUE
922.000      C
923.000      499 NEQ=ID
924.000      NE=K
925.000      WRITE(IW,500) NEQ,NE
926.000      WRITE(IW,500) (NDA(II),ICD(II),II=1,NEQ)
927.000      WRITE(IW,510) (AK(IK),IK=1,NE)
928.000      RETURN
929.000      C
930.000      500 FORMAT(2I8)
931.000      510 FORMAT(6E12.5)
932.000      515 FORMAT(5E12.5)
933.000      520 FORMAT(5I5)
934.000      END
935.000      C
936.000      FUNCTION CST(I,IND)
937.000      COMMON/COM1/BEAM(1050),CL(240),CA(240),CB(240),CG(240),
938.000      +CEF(240),CAL(240),NT,NS,NST(10),NBAY(9),NBS(9),H(30)
939.000      FG=CG(I)
940.000      CO=CEF(I),
941.000      GO TO (3,4,5,6,7,8), IND
942.000      3 CST=CO*(2.+FG+6.*CA(I)/CL(I)+6.*(CA(I)/CL(I))**2)
943.000      RETURN
944.000      4 CST=CO*(1.-FG+3.*(CA(I)+CB(I))/CL(I)+6.*CA(I)*CB(I)/CL(I)**2)
945.000      RETURN
946.000      5 CST=CO*(2.+FG+6.*CB(I)/CL(I)+6.*(CB(I)/CL(I))**2)
947.000      RETURN
948.000      6 CST=CO*(3./CL(I)+6.*CA(I)/CL(I)**2)
949.000      RETURN
950.000      7 CST=CO*(3./CL(I)+6.*CB(I)/CL(I)**2)
951.000      RETURN
952.000      8 CST=CO*6./CL(I)**2
953.000      RETURN
954.000      END
955.000      C
956.000      FUNCTION BST(I,IMB)
957.000      COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
958.000      +COLM(1440),NT,NS,NST(10),NBAY(9),NBS(9),H(30)
959.000      FG=BG(I)
960.000      CO=BEF(I)

```

```

961.000      GO TO(1,2,3,4,5,6), IMB
962.000      BSI=CO*(2.+FG+6.*BA(I)/BL(I)+6.*(BA(I)/BL(I))**2)
963.000      RETURN
964.000      BSI=CO*(1.-FG+3.*(BA(I)+BB(I))/BL(I)+6.*BA(I)*BB(I)/BL(I)**2)
965.000      RETURN
966.000      BSI=CO*(2.+FG+6.*BB(I)/BL(I)+6.*(BB(I)/BL(I))**2)
967.000      RETURN
968.000      BSI=CO*(3./BL(I)+6.*BA(I)/BL(I)**2)
969.000      RETURN
970.000      BSI=CO*(3./BL(I)+6.*BB(I)/BL(I)**2)
971.000      RETURN
972.000      BSI=CO*6./BL(I)**2
973.000      RETURN
974.000      END
975.000      C
976.000      SUBROUTINE GRAVT(IJ,NGR,NE,NQQ,NEL,NEQ,MBM,MBA,NNN,
977.000      +A,NA,NR,SLD,FX1,FX2,FX3,FX4,UDL)
978.000      C
979.000      C
980.000      C
981.000      COMMON/COM1/BMCL(2490),NT,NNS,NSS(10),NBAY(9),NBS(9),H(30)
982.000      DIMENSION A(NE),NA(NQQ),NR(NQQ),SLD(NQQ),FX1(MBM),FX2(MBM),
983.000      +FX3(MBM),FX4(MBM),UDL(NNN,MBA)
984.000      IR=3
985.000      IW=7
986.000      READ(IR,500) NFR
987.000      DO 100 I=1,NFR
988.000      IJ=IJ+1
989.000      READ(IR,500) IFL
990.000      WRITE(8) IFL
991.000      IF(IFL.NE.0) GO TO 50
992.000      WRITE(IW,600) IJ
993.000      DO 40 J=1,NGR
994.000      WRITE(IW,630) J
995.000      CALL LOAD(NQQ,MBA,NNN,SLD,MBM,NBM,FX1,FX2,FX3,FX4,UDL)
996.000      JOPT=0
997.000      CALL BSUB(JOPT,NE,NQQ,A,NA,NR,SLD,NEQ,NEL)
998.000      WRITE(8) (SLD(JJ),JJ=1,NEQ)
999.000      WRITE(8) (FX1(JJ),FX2(JJ),FX3(JJ),FX4(JJ),JJ=1,NBM)
1000.000      40 CONTINUE

```

```

1001.000
1002.000
1003.000
1004.000
1005.000
1006.000
1007.000
1008.000
1009.000
1010.000
1011.000
1012.000
1013.000
1014.000
1015.000
1016.000
1017.000
1018.000
1019.000
1020.000
1021.000
1022.000
1023.000
1024.000
1025.000
1026.000
1027.000
1028.000
1029.000
1030.000
1031.000
1032.000
1033.000
1034.000
1035.000
1036.000
1037.000
1038.000
1039.000
1040.000

      C
      K=IJ
      GO TO 100
      50 WRITE(IW,620) IJ,K
      100 CONTINUE
      RETURN

      500 FORMAT(5I5)
      600 FORMAT(//
+-----//
+  FRAME NO.,I3, /
+-----//
      620 FORMAT(//  *** //
+  GRAVITY LOADS(FOR ALL LOAD CASES) IN FRAME, I3, /
+  ARE EXACTLY SAME AS THOSE IN FRAME, I3, /
+  *** //
      630 FORMAT(//
+  GRAVITY LOAD CASE, I3, /
+-----//
      END

      SUBROUTINE LOAD(NQQ,MBA,NNN,SLD,MBM,NBM,FX1,FX2,FX3,FX4,UDL)
      PREPARE THE LOAD MATRIX

      COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
+CL(240),CA(240),CB(240),CG(240),CEF(240),CAL(240),
+NT,NNS,NSS(10),NBA(9),NBS(9),CNL(15),DIS(15)
      DIMENSION SLD(NQQ),FX1(MBM),FX2(MBM),FX3(MBM),FX4(MBM),
+UDL(NNN,MBA)

      IR=3
      IW=7
      IK=0
      IH=0
      IS=0
      NJ=0
      IB=0
      DO 6 I=1,NT
      NN=NSS(I)
      NB=NBA(I)

```

```

1041.000
1042.000
1043.000
1044.000
1045.000
1046.000
1047.000
1048.000
1049.000
1050.000
1051.000
1052.000
1053.000
1054.000
1055.000
1056.000
1057.000
1058.000
1059.000
1060.000
1061.000
1062.000
1063.000
1064.000
1065.000
1066.000
1067.000
1068.000
1069.000
1070.000
1071.000
1072.000
1073.000
1074.000
1075.000
1076.000
1077.000
1078.000
1079.000
1080.000

DO 5 J=1,NN
IH=IH+1
READ(IR,2000) (UDL(IH,K),K=1,NB)
5 CONTINUE
6 CONTINUE
WRITE(IW,4500)
DO 40 I=1,NT
NN=NSS(I)
NB=NBA(I)
DO 30 J=1,NN
IS=IS+1
FN1=0.0
FN2=0.0
DO 25 K=1,NB
IB=IB+1
IK=IK+1
IK1=IK+NB+2
NI=NJ+K
UD=UDL(IS,K)
FUD2=UD*BL(IB)/2.
FUD1=FUD2*BL(IB)/6.
FST2=UD*BA(IB)
FST1=FST2*BA(IB)/2.
FST4=UD*BB(IB)
FST3=-FST4*BB(IB)/2.
READ(IR,1000) NC
FC1=0.0
FC2=0.0
FC3=0.0
FC4=0.0
BGX=BG(IB)
IF(NC.EQ.0) GO TO 21
15 READ(IR,2000) (CNL(KK),DIS(KK),KK=1,NC)
DO 20 KK=1,NC
CN=CNL(KK)
A=DIS(KK)-BA(IB)
B=BL(IB)-A
PG=CN/(1.+2.*BGX)
FI=(1.+BGX*BL(IB)/B)*PG*A*B*B/BL(IB)**2
F2=(3.*A*B+2.*BGX*BL(IB)**2/B)*PG*B*B/BL(IB)**3

```



```

1081.000 F3=-(1.+BGX*BL(IB)/A)*PG*B*A*A/BL(IB)**2
1082.000 F4=CN-F2
1083.000 FC1=FC1+F1
1084.000 FC2=FC2+F2
1085.000 FC3=FC3+F3
1086.000 FC4=FC4+F4
1087.000
1088.000 20 CONTINUE
1089.000 21 FL1=FUD1+FC1
1090.000 FL2=FUD2+FC2
1091.000 SLD(IK)=- (FL1+FN1+FST1+FL2*BA(IB))
1092.000 SLD(IK1)=- (FL2+FN2+FST2)
1093.000 FR1=-FUD1+FC3
1094.000 FR2=FUD2+FC4
1095.000 FN1=FR1+FST3-FR2*BB(IB)
1096.000 FN2=FR2+FST4
1097.000 FX1(IB)=FL1
1098.000 FX2(IB)=FR1
1099.000 FX3(IB)=FL2
1100.000 FX4(IB)=FR2
1101.000 IF(NC.NE.0) GO TO 23
1102.000 IF(K.NE.1) GO TO 22
1103.000 WRITE(IW,5000) (IS,NI,UD)
1104.000 GO TO 25
1105.000 22 WRITE(IW,5100) NI,UD
1106.000 GO TO 25
1107.000 23 IF(K.NE.1) GO TO 24
1108.000 WRITE(IW,5000) IS,NI,UD,(CNL(KK),DIS(KK),KK=1,NC)
1109.000 GO TO 25
1110.000 24 WRITE(IW,5100) NI,UD,(CNL(KK),DIS(KK),KK=1,NC)
1111.000 25 CONTINUE
1112.000 IK=IK+1
1113.000 IK1=IK1+1
1114.000 IK2=IK+1
1115.000 SLD(IK)=-FN1
1116.000 SLD(IK1)=-FN2
1117.000 SLD(IK2)=0.0
1118.000 IK=IK1
1119.000 30 CONTINUE
1120.000 IF(1.EQ.NT) GO TO 40
NJ=NJ+NBS(I)

```

```

1121.000      40 CONTINUE
1122.000      NBM=IB
1123.000      RETURN
1124.000      C
1125.000      1000 FORMAT(5I5)
1126.000      2000 FORMAT(8F10.3)
1127.000      4500 FORMAT(//2X,'SUPPLIED LOAD DATA',/
1128.000      +2X,'-----',//
1129.000      +38X,'CONCENTRATED LOAD DATA',/
1130.000      + '  STORY BAY   UDL   LOAD-1 DIST-1   LOAD-2 DIST-2',
1131.000      + '  LOAD-3 DIST-3   LOAD-4 DIST-4')
1132.000      5000 FORMAT(/215,F7.3,2X,4(F8.2,F7.2))
1133.000      5100 FORMAT(I10,F7.3,2X,4(F8.2,F7.2))
1134.000      END
1135.000      C
1136.000      SUBROUTINE FORCE(OPT,MBM,MCL,NQQ,NNN,FX1,FX2,FX3,FX4,
1137.000      +FC1,FC2,FC3,FC4,SOB1,SOB2,SOB3,SOB4,BETA,DSP)
1138.000      C
1139.000      PROGRAM TO CALCULATE MEMBER FORCES
1140.000      C
1141.000      COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
1142.000      +CL(240),CA(240),CB(240),CG(240),CEF(240),CAL(240),NT,NNS,
1143.000      +NSS(10),NBA(9),NBS(9),DISQ(30)
1144.000      DIMENSION FX1(MBM),FX2(MBM),FX3(MBM),FX4(MBM),FC1(MCL),
1145.000      +FC2(MCL),FC3(MCL),FC4(MCL),SOB1(MBM),SOB2(MBM),
1146.000      +SOB3(MBM),SOB4(MBM),BETA(NQQ),DSP(NNN)
1147.000      C
1148.000      OPT=0 MEANS THERE IS NO GRAVITY LOAD ON THE BEAMS
1149.000      OPT=1 MEANS BEAMS HAVE GRAVITY LOADS
1150.000      OPT=2 MEANS CALCULATE ROOI SUM SQR. OF FORCES
1151.000      OPT=3 MEANS OPT=0 & OPT=2 ; OPT=4 MEANS OPT=1 & OPT=2
1152.000      C
1153.000      IW=7
1154.000      IF(OPT.NE.2) WRITE(IW,600)
1155.000      IS=0
1156.000      IK1=1
1157.000      NJ=0
1158.000      IB=0
1159.000      DO 30 I=1,NT
1160.000      NN=NSS(I)

```

1161.000
 1162.000
 1163.000
 1164.000
 1165.000
 1166.000
 1167.000
 1168.000
 1169.000
 1170.000
 1171.000
 1172.000
 1173.000
 1174.000
 1175.000
 1176.000
 1177.000
 1178.000
 1179.000
 1180.000
 1181.000
 1182.000
 1183.000
 1184.000
 1185.000
 1186.000
 1187.000
 1188.000
 1189.000
 1190.000
 1191.000
 1192.000
 1193.000
 1194.000
 1195.000
 1196.000
 1197.000
 1198.000
 1199.000
 1200.000

NB=NBA(I)
 DO 25 J=1,NN
 IS=IS+1
 IJ1=IK1+NB+2
 DO 20 K=1,NB
 IB=IB+1
 NI=NJ+K
 IK=IK1
 IK1=IK1+1
 IJ=IJ1
 IJ1=IJ1+1
 DL1=BETA(IK)
 DL2=BETA(IK1)
 DEL=BA(IB)*BETA(IK)+BB(IB)*BETA(IK1)+BETA(IJ)-BETA(IJ1)
 DBL=3.*DEL/BL(IB)
 AA=BEF(IB)
 B=(1.-BG(IB))
 C=(2.+BG(IB))
 F1=AA*(C*DL1+B*DL2+DBL)
 F2=AA*(B*DL1+C*DL2+DBL)
 F3=AA*(3.*(DL1+DL2)+2.*DBL)/BL(IB)
 F4=-F3
 KOPT=OPT+1
 GO TO(16,10,19,12,10), KOPT
 10 F1=F1+FX1(IB)
 F2=F2+FX2(IB)
 F3=F3+FX3(IB)
 F4=F4+FX4(IB)
 IF(OPT.EQ.1) GO TO 16
 12 FS1=F1+SQB1(IB)
 FS2=F2+SQB2(IB)
 FS3=F3+SQB3(IB)
 FS4=F4+SQB4(IB)
 F1=F1-SQB1(IB)
 F2=F2-SQB2(IB)
 F3=F3-SQB3(IB)
 F4=F4-SQB4(IB)
 IF(K.NE.1) GO TO 14
 WRITE(IW,530) IS,NI,FS1,FS2,FS3,FS4,F1,F2,F3,F4
 GO TO 20

```

1201.000
1202.000
1203.000
1204.000
1205.000
1206.000
1207.000
1208.000
1209.000
1210.000
1211.000
1212.000
1213.000
1214.000
1215.000
1216.000
1217.000
1218.000
1219.000
1220.000
1221.000
1222.000
1223.000
1224.000
1225.000
1226.000
1227.000
1228.000
1229.000
1230.000
1231.000
1232.000
1233.000
1234.000
1235.000
1236.000
1237.000
1238.000
1239.000
1240.000

14 WRITE(IW,535) NI,FS1,FS2,FS3,FS4,F1,F2,F3,F4
   GO TO 20
16 IF(K.NE.1) GO TO 17
   WRITE(IW,520) IS,NI,F1,F2,F3,F4
   GO TO 20
17 WRITE(IW,525) NI,F1,F2,F3,F4
   GO TO 20
19 FX1(IB)=FX1(IB)+F1*F1
   FX2(IB)=FX2(IB)+F2*F2
   FX3(IB)=FX3(IB)+F3*F3
   FX4(IB)=FX3(IB)
20 CONTINUE
   IKI=IKI+1
25 CONTINUE
   IF(I.EQ.NT) GO TO 30
   NJ=NJ+NBS(I)
30 CONTINUE

IS=0
IK=0
NJ=0
IC=0
IF(OPT.NE.2) WRITE(IW,620)
DO 90 I=1,NT
  NN=NSS(I)
  NB=NBA(I)+1
  DO 80 J=1,NN
    IS=IS+1
    IJ=IK+NB+1
    IV=IJ
    DP=BETA(IJ)
    IF(OPT.NE.2) GO TO 35
    DSP(IS)=DSP(IS)+DP*DP
    GO TO 38
35 DSP(IS)=DP
38 CONTINUE
    DO 60 K=1,NB
      IC=IC+1
      NI=NJ+K
      IK=IK+1

```

```

1241.000 IV=IV+1
1242.000 AA=CEF(IC)
1243.000 B=(1.-CG(IC))
1244.000 C=(2.+CG(IC))
1245.000 IF(IS.EQ.1) GO TO 40
1246.000 IK1=IK1+1
1247.000 IV1=IV1+1
1248.000 DL1=BETA(IK)
1249.000 DL2=BETA(IK1)
1250.000 DEL=CB(IC)*BETA(IK)+CA(IC)*BETA(IK1)+BETA(IJ)-BETA(IJ1)
1251.000 DELV=BETA(IV)-BETA(IV1)
1252.000 GO TO 50
1253.000 40 DL2=0.0
1254.000 DL1=BETA(IK)
1255.000 DEL=BETA(IK)*CB(IC)+BETA(IJ)
1256.000 DELV=BETA(IV)
1257.000 50 DBL=3.*DEL/CL(IC)
1258.000 FMT=AA*(C*DL1+B*DL2+DBL)
1259.000 FMB=AA*(B*DL1+C*DL2+DBL)
1260.000 FS=AA*(3.*(DL1+DL2)+2.*DBL)/CL(IC)
1261.000 FA=DELV*CAL(IC)
1262.000 IF(OPT-2) 53,55,51
1263.000 51 FS1=FMT+FC1(IC)
1264.000 FS2=FMB+FC2(IC)
1265.000 FS3=FA+FC3(IC)
1266.000 FS4=FS+FC4(IC)
1267.000 FMT=FMT-FC1(IC)
1268.000 FMB=FMB-FC2(IC)
1269.000 FA=FA-FC3(IC)
1270.000 FS=FS-FC4(IC)
1271.000 IF(K.NE.1) GO TO 52
1272.000 WRITE(IW,530) IS,NI,FS1,FS2,FS3,FS4,FMT,FMB,FA,FS
1273.000 GO TO 60
1274.000 52 WRITE(IW,535) NI,FS1,FS2,FS3,FS4,FMT,FMB,FA,FS
1275.000 GO TO 60
1276.000 53 IF(K.NE.1) GO TO 54
1277.000 WRITE(IW,520) IS,NI,FMT,FMB,FA,FS
1278.000 GO TO 60
1279.000 54 WRITE(IW,525) NI,FMT,FMB,FA,FS
1280.000 GO TO 60

```

```

1281.000
1282.000
1283.000
1284.000
1285.000
1286.000
1287.000
1288.000
1289.000
1290.000
1291.000
1292.000
1293.000
1294.000
1295.000
1296.000
1297.000
1298.000
1299.000
1300.000
1301.000
1302.000
1303.000
1304.000
1305.000
1306.000 C
1307.000
1308.000
1309.000
1310.000
1311.000
1312.000
1313.000
1314.000
1315.000
1316.000
1317.000
1318.000
1319.000
1320.000

55 FC1(IC)=FC1(IC)+FMT*FMT
   FC2(IC)=FC2(IC)+FMB*FMB
   FC3(IC)=FC3(IC)+FA*FA
   FC4(IC)=FC4(IC)+FS*FS
60 CONTINUE
   IK1=IK-NB
   IJ1=IJ
   IK=IV
   IV1=IJ
80 CONTINUE
   IF(I.EQ.NT) GO TO 90
   NJ=NJ+NBS(I)
   IK1=IK1+NBS(I)
   IV1=IV1+NBS(I)
90 CONTINUE
   IF(OPT-2) 92,99,94
92 WRITE(IW,640)
   WRITE(IW,650) (IS,DSP(IS),IS=1,NNS)
   GO TO 99
94 WRITE(IW,640)
   DO 96 IS=1,NNS
   DP=DSP(IS)+DISQ(IS)
   DPI=DSP(IS)-DISQ(IS)
96 WRITE(IW,540) IS,DP,DPI
99 RETURN

520 FORMAT(/I5,I6,4(3X,E12.5))
525 FORMAT(I11,4(3X,E12.5))
530 FORMAT(I5,I6,4(3X,E12.5),/ MAX',/
      +11X,4(3X,E12.5),/ MIN')
535 FORMAT(I11,4(3X,E12.5),/ MAX',/11X,4(3X,E12.5),/ MIN')
540 FORMAT(I7,E15.5,/ MAX',/7X,E15.5,/ MIN')
600 FORMAT(/2X,'BEAM END ACTIONS',/
      +2X,'-----',/
      +,' STORY BAY-NO',2X,'LFT-MOMENT',6X,'RT-MOMENT',6X,'LFT-FORCE',
      +7X,'RT-FORCE')
620 FORMAT(/2X,'COLUMN END ACTIONS',/
      +2X,'-----',/
      +,' STORY COL.LN TOP-MOMENT BOT-MOMENT AXIAL-FORCE',
      +,' SHEAR-FORCE')

```

```

1321.000
1322.000
1323.000
1324.000
1325.000
1326.000 C
1327.000
1328.000 C
1329.000 C
1330.000 C
1331.000 C
1332.000
1333.000
1334.000
1335.000
1336.000
1337.000
1338.000
1339.000
1340.000
1341.000
1342.000
1343.000
1344.000
1345.000
1346.000
1347.000
1348.000
1349.000
1350.000
1351.000
1352.000 C
1353.000
1354.000 C
1355.000
1356.000
1357.000
1358.000
1359.000
1360.000

640 FORMAT(/// STORY LATERAL DISPLT//
+-----+
+ ST.LEVEL DISPLT//
650 FORMAT(I7,E15.5)
END

SUBROUTINE COND(NFG,A,NA,NR,TX,NQQ,NEL,NEQ,NL)
THIS READS THE FRAME STIFFNESS AND CONDENSES TO
THE STORY DEGREES OF FREEDOM AND WRITES
COMMON/COM1/SC(35,35),TA(660),NF(625),NC(40)
DIMENSION A(NEL),NA(NQQ),NR(NQQ),TX(NQQ)
IR=5
IW=2
REWIND IW
REWIND IR
DO 20 I=1,NFG
READ(IR,302) NEQ,NL
READ(IR,302) (NA(I),NR(I),I=1,NEQ)
READ(IR,303) (A(I),I=1,NL)
CALL CONDEN(MM,A,NA,NR,NQQ,NEL,NEQ,NL,TX)
WRITE(IW,302) II,MM
DO 10 I=1,MM
WRITE(IW,303) (SC(I,J),J=1,MM)
10 CONTINUE
20 CONTINUE
RETURN
302 FORMAT(2I8)
303 FORMAT(6E12.5)
END

SUBROUTINE CONDEN(MM,A,NA,NR,NQQ,NEL,NEQ,NL,TX)
COMMON/COM1/SC(35,35),TA(660),NF(625),NC(40)
DIMENSION A(NEL),NA(NQQ),NR(NQQ),TX(NQQ)
JOPT=1
CALL REDUCE(JOPT,NEL,NQQ,A,NA,NR,NEQ,NL)
LL=0
MM=0

```

```

1361.000      DO 50 I=1,NEQ
1362.000      NN=NR(I)+1
1363.000      GO TO(30,40,50),NN
1364.000      30 LL=LL+1
1365.000      NF(LL)=I
1366.000      GO TO 50
1367.000      40 MM=MM+1
1368.000      NC(MM)=I
1369.000      50 CONTINUE
1370.000      DO 55 I=1,MM
1371.000      DO 55 J=1,MM
1372.000      55 SC(I,J)=0.
1373.000      DO 60 J=2,MM
1374.000      KJ=NC(J)
1375.000      NAJ=NA(KJ)
1376.000      JJ=KJ-1
1377.000      NAPJ=NA(JJ)
1378.000      IJ=NAJ-KJ
1379.000      IL=J-1
1380.000      DO 60 I=1,IL
1381.000      KI=NC(I)
1382.000      NADR=IJ+KI
1383.000      IF(NADR.LE.NAPJ) GO TO 60
1384.000      SC(I,J)=A(NADR)
1385.000      SC(J,I)=A(NADR)
1386.000      60 CONTINUE
1387.000      DO 70 I=1,MM
1388.000      KI=NC(I)
1389.000      NAI=NA(KI)
1390.000      SC(I,I)=A(NAI)
1391.000      70 CONTINUE
1392.000      DO 130 J=1,MM
1393.000      DO 75 I=1,NEQ
1394.000      75 TX(I)=0.
1395.000      NTEST=0
1396.000      JADR=NC(J)
1397.000      DO 90 I=1,LL
1398.000      IADR=NF(I)
1399.000      IF(IADR.LT.JADR) GO TO 80
1400.000      IA=JADR

```



```

1401.000 JA=IADR
1402.000 NADR=NA(JA)-JA+IA
1403.000 JJ=JA-1
1404.000 GO TO 85
1405.000 80 NADR=NA(JADR)-JADR+IADR
1406.000 JJ=JADR-1
1407.000 85 NAJP=NA(JJ)
1408.000 IF(NADR-LE.NAJP) GO TO 90
1409.000 NTEST=1
1410.000 TX(IADR)=A(NADR)
1411.000 90 CONTINUE
1412.000 IF(NTEST.EQ.0) GO TO 130
1413.000 DO 100 I=1,NEQ
1414.000 TA(I)=TX(I)
1415.000 100 CONTINUE
1416.000 CALL BSUB(JOPT,NEL,NQQ,A,NA,NR,TX,NEQ)
1417.000 CALL DOTPRD(TX,NEQ,S,NQQ)
1418.000 SC(J,J)=SC(J,J)-S
1419.000 IF(J.EQ.1) GO TO 130
1420.000 DO 105 I=1,NEQ
1421.000 105 TA(I)=0.
1422.000 IL=J-1
1423.000 DO 130 I=1,IL
1424.000 N2T=0
1425.000 IADR=NC(I)
1426.000 DO 120 K=1,IL
1427.000 KADR=NF(K)
1428.000 IF(KADR.LT.IADR) GO TO 110
1429.000 KA=IADR
1430.000 IA=KADR
1431.000 NADR=NA(IA)-IA+KA
1432.000 II=IA-1
1433.000 GO TO 115
1434.000 110 NADR=NA(IADR)-IADR+KADR
1435.000 II=IADR-1
1436.000 115 NAIP=NA(II)
1437.000 IF(NADR.LE.NAIP) GO TO 120
1438.000 N2T=1
1439.000 TA(KADR)=A(NADR)
1440.000 120 CONTINUE

```

```

1441.000
1442.000
1443.000
1444.000
1445.000
1446.000
1447.000
1448.000 C
1449.000
1450.000
1451.000
1452.000
1453.000
1454.000
1455.000
1456.000
1457.000
1458.000 C
1459.000
1460.000
1461.000
1462.000
1463.000
1464.000
1465.000
1466.000
1467.000
1468.000
1469.000
1470.000
1471.000
1472.000
1473.000
1474.000
1475.000
1476.000
1477.000
1478.000
1479.000
1480.000

IF(N2T.EQ.0) GO TO 130
CALL DOTPRD(TX,NEQ,S,NQQ)
SC(I,J)=SC(I,J)-S
SC(J,I)=SC(I,J)
130 CONTINUE
RETURN
END

SUBROUTINE DOTPRD(D,L,S,NQQ)
COMMON/COM1/SC(35,35),C(660),NF(625),NC(40)
DIMENSION D(NQQ)
S=0.
DO 10 I=1,L
S=S+C(I)*D(I)
10 CONTINUE
RETURN
END

SUBROUTINE REDUCE(JOPT,NEL,NQQ,A,NA,NR,NEQ,NL)
DIMENSION A(NEL),NA(NQQ),NR(NQQ)
IW=6
NAJP=1
DO 90 J=2,NEQ
NAJ=NA(J)
IF(JOPT.EQ.1.AND.NR(J).NE.0) GO TO 90
IL=J-1
LJ=J+NAJP-NAJ+1.
IF(LJ.GT.IL) GO TO 90
IN=LJ+1
JK=NAJ-J
IF(IN.GT.IL) GO TO 60
NAIP=NA(LJ)
DO 50 I=IN,IL
NAI=NA(I)
IF(JOPT.EQ.1.AND.NR(I).NE.0) GO TO 50
KL=I-1
LI=I+NAIP-NAI+1
KI=MAXO(LI,LJ)
IF(KI.GT.KL) GO TO 50
IK=NAI-I

```

```

1481.000 SUM=0.
1482.000 DO 30 K=KI,KL
1483.000 IF(JOPT.EQ.1.AND.NR(K).NE.O) GO TO 30
1484.000 IKA=IK+K
1485.000 JKA=JK+K
1486.000 SUM=SUM+A(IKA)*A(JKA)
1487.000 30 CONTINUE
1488.000 IJ=JK+I
1489.000 A(IJ)=A(IJ)-SUM
1490.000 50 NAIP=NAI
1491.000 60 SUM=0.
1492.000 DO 70 K=LJ,IL
1493.000 IF(JOPT.EQ.1.AND.NR(K).NE.O) GO TO 70
1494.000 JKA=JK+K
1495.000 KKA=NA(K)
1496.000 CC=A(JKA)/A(KKA)
1497.000 SUM=SUM+CC*A(JKA)
1498.000 A(JKA)=CC
1499.000 70 CONTINUE
1500.000 A(NAJ)=A(NAJ)+SUM
1501.000 90 NAJP=NAJ
1502.000 IF(JOPT.EQ.1) GO TO 99
1503.000 WRITE(IW,540) NEQ,NL
1504.000 WRITE(IW,540) (NA(I),NR(I),I=1,NEQ)
1505.000 WRITE(IW,550) (A(I),I=1,NL)
1506.000 99 RETURN
1507.000 540 FORMAT(2I8)
1508.000 550 FORMAT(6E12.5)
1509.000 END
1510.000 C
1511.000 SUBROUTINE BSUB(JOPT,NEL,NQQ,A,NA,NR,B,NEQ)
1512.000 DIMENSION A(NEL),NA(NQQ),NR(NQQ),B(NQQ)
1513.000 N1=NEQ-1
1514.000 DO 100 I=1,N1
1515.000 IF(B(I).NE.(O.)) GO TO 110
1516.000 100 CONTINUE
1517.000 I=N1
1518.000 110 NB=I
1519.000 IN=NB+1
1520.000 NAIP=NA(NB)

```

```

1521.000      DO 150 I=IN,NEQ
1522.000      NAI=NA(I)
1523.000      IF(JOPT.EQ.1.AND.NR(I).NE.O) GO TO 150
1524.000      LI=I-NAI+NAIP+1
1525.000      KI=MAXO(LI,NB)
1526.000      KL=I-1
1527.000      IF(KI.GT.KL)GO TO 150
1528.000      IK=NAI-I
1529.000      SUM=O.
1530.000      DO 120 K=KI,KL
1531.000      IF(JOPT.EQ.1.AND.NR(K).NE.O) GO TO 120
1532.000      IKA=IK+K
1533.000      SUM=SUM+A(IKA)*B(K)
1534.000      120 CONTINUE
1535.000      B(I)=B(I)-SUM
1536.000      150 NAIP=NAI
1537.000      DO 160 I=NB,NEQ
1538.000      IF(JOPT.EQ.1.AND.NR(I).NE.O) GO TO 160
1539.000      NAI=NA(I)
1540.000      B(I)=B(I)/A(NAI)
1541.000      160 CONTINUE
1542.000      I=NEQ
1543.000      NAN=NA(I)
1544.000      165 KL=(I-1)
1545.000      NAI=NA(KL)
1546.000      IF(JOPT.EQ.1.AND.NR(I).NE.O) GO TO 175
1547.000      KI=I-NAN+NAI+1
1548.000      IF(KI.GT.KL)GO TO 175
1549.000      IK=NAN-I
1550.000      DO 170 K=KI,KL
1551.000      IF(JOPT.EQ.1.AND.NR(K).NE.O) GO TO 170
1552.000      IKA=IK+K
1553.000      B(K)=B(K)-A(IKA)*B(I)
1554.000      170 CONTINUE
1555.000      175 I=I-1
1556.000      IF(I.EQ.1) GO TO 180
1557.000      NAN=NAI
1558.000      GO TO 165
1559.000      180 RETURN
1560.000      END

```

```

1561.000 C
1562.000
1563.000 C
1564.000 C
1565.000 C
1566.000
1567.000
1568.000
1569.000
1570.000
1571.000
1572.000
1573.000
1574.000
1575.000
1576.000
1577.000
1578.000
1579.000
1580.000
1581.000
1582.000
1583.000
1584.000
1585.000
1586.000
1587.000
1588.000
1589.000
1590.000
1591.000
1592.000
1593.000
1594.000
1595.000
1596.000
1597.000
1598.000
1599.000
1600.000

SUBROUTINE STORY(SM,XBAR,YBAR)
  READS AND PRINTS STORY DATA
  COMMON/CON/NS1,NF1,NOPT,NEQ,NMULP
  COMMON/COMSYS/NGR,NWL,NDYN,NSYS,IOUT,SMULP(12,8)
  COMMON/MISC/IR,IW,AI(3),AJ(3),IDN,NH
  DIMENSION SM(NEQ),XBAR(NS1),YBAR(NS1)
  IF(NDYN.NE.O.OR.IOUT.NE.O) GO TO 5
  DO 3 I=1,NS1
    XBAR(I)=O.O
    YBAR(I)=O.O
  3 CONTINUE
  RETURN
  5 CALL SP(IW,2)
    WRITE(IW,301)
    WRITE(IW,302)
    WRITE(IW,301)
    CALL SP(IW,1)
    WRITE(IW,303)
    DO 20 I=1,NS1
      READ(IR,304) XM,XMI,XBAR(I),YBAR(I)
      WRITE(IW,305) I,XM,XMI,XBAR(I),YBAR(I)
      IF(NOPT.EQ.3) GO TO 10
      SM(I)=XM
      GO TO 20
    10 L=3*I
      SM(L)=XMI
      L=L-1
      SM(L)=XM
      L=L-1
      SM(L)=XM
    20 CONTINUE
    RETURN
    301 FORMAT(1X,'-----')
    302 FORMAT(1X,' STORY MASS DATA')
    303 FORMAT(1X,' LEVEL',6X,'MASS',6X,'MASS INERTIA',6X,
      1,'XBAR',6X,'YBAR')
    304 FORMAT(8F10.0)

```

```

1601.000
1602.000
1603.000
1604.000
1605.000
1606.000
1607.000
1608.000
1609.000
1610.000 C
1611.000 C
1612.000 C
1613.000 C
1614.000
1615.000
1616.000
1617.000
1618.000
1619.000
1620.000
1621.000
1622.000
1623.000
1624.000 C
1625.000 C
1626.000 C
1627.000
1628.000
1629.000 C
1630.000 C
1631.000 C
1632.000 C
1633.000
1634.000
1635.000
1636.000
1637.000
1638.000
1639.000
1640.000

305 FORMAT(IX,I4,F13.3,F15.3,F13.2,F10.2)
END
SUBROUTINE CORE(M)
COMMON/MISC/IR,IW,AI(3),AJ(3),IDN,NH
WRITE(IW,301) M
STOP
301 FORMAT(5X,'** INCREASE MCORE BY',I7,'**')
END
SUBROUTINE FRAME(A,DN,M,NDIR,XBAR,YBAR,S,DDA)
READS FRAME LOCATIONS AND FRAME STIFFNESS MATRICES AND
ASSEMBLES BUILDING STIFFNESS MATRIX
COMMON/CON/NST,NFT,NOPT,NEQ,NMULP
COMMON/MISC/IR,IW,AI(3),AJ(3),IDN,NH
DIMENSION A(NEQ,NEQ),S(NST,NST),XBAR(NST),YBAR(NST)
DIMENSION DN(NST,NFT),M(NFT),NDIR(NFT),DDA(NFT)
IFP=0
DO 5 I=1,NEQ
DO 5 J=1,NEQ
5 A(I,J)=0.0
REWIND 2
GO TO(10,40,70),NOPT
HERE WHEN NOPT=1
10 DO 30 NF=1,NFT
IF(M(NF).EQ.IFP)GO TO 20
READ CONDENSED FRAME STIFFNESS FROM UNIT 2, TRANSFORM IT
TO GLOBAL COORDINATES AND ADD TO BUILDING STIFFNESS MATRIX
READ(2,306) NFG,MST
DO 12 LL=1,MST
READ(2,307)(S(LL,J),J=1,MST)
12 CONTINUE
IFP=M(NF)
20 IF(NDIR(NF).EQ.2)GO TO 30
DO 30 I=1,MST
DO 30 J=1,MST

```

```

1641.000
1642.000
1643.000
1644.000 C
1645.000 C
1646.000 C
1647.000
1648.000
1649.000 C
1650.000 C
1651.000 C
1652.000 C
1653.000
1654.000
1655.000
1656.000
1657.000
1658.000
1659.000
1660.000
1661.000
1662.000
1663.000
1664.000 C
1665.000 C
1666.000 C
1667.000
1668.000
1669.000
1670.000 C
1671.000 C
1672.000 C
1673.000 C
1674.000
1675.000
1676.000
1677.000
1678.000
1679.000
1680.000

      A(I,J)=A(I,J)+S(I,J)
30  CONTINUE
      GO TO 155

      HERE WHEN NOPT=2

40  DO 60 NF=1,NFT
      IF(M(NF).EQ.IFP)GO TO 50

      READ CONDENSED FRAME STIFFNESS FROM UNIT 2, TRANSFORM IT
      TO GLOBAL COORDINATES AND ADD TO BUILDING STIFFNESS MATRIX

      READ(2,306)NFG,MST
      DO 42 LL=1,MST
      READ(2,307)(S(LL,J),J=1,MST)
42  CONTINUE
      IFP=M(NF)
50  IF(NDIR(NF).EQ.1)GO TO 60
      DO 60 I=1,MST
      DO 60 J=1,MST
      A(I,J)=A(I,J)+S(I,J)
60  CONTINUE
      GO TO 155

      HERE WHEN NOPT=3

70  DO 140 I=1,NF1
      DDN=DDA(I)
      IF(M(I).EQ.IFP) GO TO 80

      READ CONDENSED FRAME STIFFNESS FROM UNIT 2, TRANSFORM IT
      TO GLOBAL COORDINATES AND ADD TO BUILDING STIFFNESS MATRIX

      READ(2,306)NFG,MST
      DO 72 LL=1,MST
      READ(2,307)(S(LL,J),J=1,MST)
72  CONTINUE
      IFP=M(I)
80  IF(NDIR(I).EQ.2) GO TO 100
      DO 90 K=1,MST

```

```

1681.000
1682.000
1683.000
1684.000
1685.000
1686.000
1687.000
1688.000
1689.000
1690.000
1691.000
1692.000
1693.000
1694.000
1695.000
1696.000
1697.000
1698.000
1699.000
1700.000
1701.000
1702.000
1703.000
1704.000
1705.000
1706.000
1707.000
1708.000
1709.000
1710.000
1711.000
1712.000
1713.000 C
1714.000 C
1715.000 C
1716.000
1717.000
1718.000
1719.000
1720.000

          DN(K,I)=-DDN+YBAR(K)
90  CONTINUE
      AI(1)=1.
      AI(2)=0.
      AJ(1)=1.
      AJ(2)=0.
      GO TO 120
100  DO 110 K=1,MST
      DN(K,I)=DDN-XBAR(K)
110  CONTINUE
      AI(1)=0.
      AI(2)=1.
      AJ(1)=0.
      AJ(2)=1.
      GO TO 120
120  DO 130 II=1,MST
      IAD=(II-1)*3
      AI(3)=DN(II,I)
      DO 130 JJ=II,MST
      JAD=(JJ-I)*3
      AJ(3)=DN(JJ,I)
      DO 130 KK=1,3
      DO 130 MM=1,3
      NI=IAD+KK
      NJ=JAD+MM
      A(NI,NJ)=A(NI,NJ)+S(II,JJ)*AI(KK)*AJ(MM)
130  CONTINUE
140  CONTINUE
      DO 150 I=2,NEQ
      II=I-1
      DO 150 J=1,II
      A(I,J)=A(J,I)
150  CONTINUE

      WRITE GLOBAL STIFFNESS MATRIX ON UNIT 5
155  REWIND 5
      WRITE(5) ((A(I,J),I=1,NEQ),J=1,NEQ)
      RETURN
300  FORMAT(3I5,6F10.0)
306  FORMAT(2I8)

```



```

1721.000 307 FORMAT(6E12.5)
1722.000 END
1723.000 SUBROUTINE SPCIRM(A,SM,DA,VAL,VEC,CO,BB,TEMP,NA)
1724.000 C
1725.000 C CALCULATES AND PRINTS MODE SHAPES AND FREQUENCIES
1726.000 C CALCULATES DESIGN STORY SHEARS AND FORCES
1727.000 C
1728.000 COMMON/CON/NSI,NFI,NOPT,NEQ,NMULP
1729.000 COMMON/SPEC/NM,LM,XMU,SF,BEFA,MI,ALLOW,TD,G,NP,TC(10),SAC(10)
1730.000 COMMON/COMSYS/NGR,NWL,NDYN,NSYS,IOUT,SMULP(12,8)
1731.000 COMMON/MISC/IR,IW,AI(3),AJ(3),IDN,NH
1732.000 EQUIVALENCE (SHIP,AI),(SH2P,AJ)
1733.000 DIMENSION A(NEQ,NEQ),SM(NEQ),DA(NEQ,NM),VAL(NM)
1734.000 DIMENSION VEC(NEQ,NM),CO(NEQ),NA(NEQ),BB(NEQ)
1735.000 DIMENSION TEMP(NM,NMULP)
1736.000 DIMENSION SHIP(3),SH2P(3)
1737.000 C
1738.000 C CALCULATE MODE SHAPES AND FREQUENCIES
1739.000 C
1740.000 CALL EIGVAL(A,SM,VAL,VEC,CO,NA,BB)
1741.000 IF(IOUT.EQ.0) GO TO 41
1742.000 WRITE(IW,307)
1743.000 DO 10 I=1,NM
1744.000 10 WRITE(IW,308) I,VAL(I)
1745.000 CALL SP(IW,2)
1746.000 WRITE(IW,330)
1747.000 WRITE(IW,309)
1748.000 WRITE(IW,330)
1749.000 XNM=FLOAT(NM)/4.
1750.000 NM1=NM/4
1751.000 XNM1=FLOAT(NM1)
1752.000 IF(XNM.GT.XNM1) NM1=NM1+1
1753.000 DO 40 INM=1,NM1
1754.000 KI=(INM-1)*4+1
1755.000 LAST=INM*4
1756.000 IF(LAST.GT.NM) LAST=NM
1757.000 CALL SP(IW,2)
1758.000 WRITE(IW,310) (I,I=K1,LAST)
1759.000 CALL SP(IW,2)
1760.000 GO TO(20,30,12),NOPT

```

```

1761.000
1762.000
1763.000
1764.000
1765.000
1766.000
1767.000
1768.000
1769.000
1770.000
1771.000
1772.000
1773.000
1774.000
1775.000
1776.000
1777.000
1778.000
1779.000
1780.000
1781.000
1782.000
1783.000 C
1784.000 C
1785.000 C
1786.000
1787.000
1788.000
1789.000
1790.000
1791.000
1792.000
1793.000
1794.000
1795.000
1796.000
1797.000
1798.000
1799.000
1800.000

12 DO 15 IKK=1,NST
   I=NST-IKK+1
   II=(I-1)*3+1
   WRITE(IW,317) I, (VEC(II,KK),KK=K1, LAST)
   II=II+1
   WRITE(IW,323) (VEC(II,KK),KK=K1, LAST)
   II=II+1
   WRITE(IW,324) (VEC(II,KK),KK=K1, LAST)
   CALL SP(IW,2)
15 CONTINUE
   GO TO 40
20 DO 25 I=1,NST
   WRITE(IW,317) I, (VEC(I,J),J=K1, LAST)
   CALL SP(IW,1)
25 CONTINUE
   GO TO 40
30 DO 35 I=1,NST
   WRITE(IW,326) I, (VEC(I,J),J=K1, LAST)
   CALL SP(IW,1)
35 CONTINUE
40 CONTINUE
41 IF (NDYN.EQ.0) RETURN

FOR EACH MODE, CALCULATE DESIGN SPECTRAL ACCELERATION

DO 90 I=1,NM
T=2.*3.14159/VAL(I)
IF (T.GT.1C(1)) GO TO 45
J=1
JJ=2
GO TO 60
45 NPP=NP-1
DO 50 J=1,NPP
   JJ=J+1
   IF (I.GT.1C(J).AND.I.LI.1C(JJ)) GO TO 60
   IF (I.EQ.1C(J)) GO TO 70
50 CONTINUE
   J=NPP
   JJ=NP
60 CT=ALOG10(SAC(JJ)/SAC(J))/(ALOG10(1C(JJ)/1C(J)))

```

```

1801.000 SA=SAC(J)*(T/TC(J))*CT*G*SF
1802.000 GO TO 80
1803.000 70 SA=SAC(J)*G*SF
1804.000 C CALCULATE MODAL STORY DISPLACEMENT
1805.000 C
1806.000 C
1807.000 80 CALL RESP(SM,DA,VEC,SA,I)
1808.000 90 CONTINUE
1809.000 CALL SP(IW,3)
1810.000 WRITE(IW,334)
1811.000 WRITE(IW,335)
1812.000 WRITE(IW,334)
1813.000 CALL SP(IW,3)
1814.000 WRITE(IW,313)
1815.000 DO 95 I=1,NM
1816.000 DO 95 J=1,NMULP
1817.000 95 T=AMP(I,J)/=0.0
1818.000 DO 97 J=1,NMULP
1819.000 SHIP(J)=0.0
1820.000 97 SH2P(J)=0.0
1821.000 DO 160 JJ=1,NST
1822.000 J=NST+1-JJ
1823.000 JX=(J-1)*3
1824.000 DO 150 KI=1,NMULP
1825.000 JY=J
1826.000 IF(NOPT.EQ.3) JY=JX+KI
1827.000 SUMA=0.
1828.000 SUMB=0.
1829.000 XMULP=1.
1830.000 XMULD=1.
1831.000 XMAXA=0.
1832.000 XMAXB=0.
1833.000 C
1834.000 C
1835.000 C
1836.000 C
1837.000 C
1838.000 C
1839.000 C
1840.000 C
MODIFY DISPLACEMENTS AND FORCES FOR THE DESIGN DUCTILITY
SUPERIMPOSE MODAL STORY FORCES AND DISPLACEMENTS AND
PRINT OUT DESIGN VALUES
DO 140 I=1,NM
T=2.*3.14159/VAL(I)
IF(T.GE.10) GO TO 100

```

```

1841.000
1842.000
1843.000
1844.000
1845.000
1846.000
1847.000
1848.000
1849.000
1850.000
1851.000
1852.000
1853.000
1854.000
1855.000
1856.000
1857.000
1858.000
1859.000
1860.000
1861.000
1862.000
1863.000
1864.000
1865.000
1866.000
1867.000
1868.000
1869.000
1870.000
1871.000
1872.000
1873.000
1874.000
1875.000
1876.000
1877.000
1878.000
1879.000
1880.000

XMULTP=1./SQRT(2.*XMU-1.)
XMULD=XMU/SQRT(2.*XMU-1.)
GO TO 110
100 XMULTP=1./XMU
110 IF(IOUT.NE.2) GO TO 135
TEMP(I,K1)=TEMP(I,K1)+DA(JY,I)*SM(JY)*XMULTP
IF(XMAXA.LT.ABS(TEMP(I,K1))) GO TO 120
IF(XMAXB.LT.ABS(TEMP(I,K1))) XMAXB=ABS(TEMP(I,K1))
GO TO 130
120 XMAXB=XMAXA
XMAXA=ABS(TEMP(I,K1))
130 SUMA=SUMA+TEMP(I,K1)**2
DA(JY,I)=DA(JY,I)/VAL(I)**2
SUMB=SUMB+(DA(JY,I)*XMULD)**2
135 DA(JY,I)=DA(JY,I)*XMULTP
140 CONTINUE
IF(IOUT.NE.2) GO TO 150
SHEAR1=SQRT(SUMA)
SHEAR2=XMAXA+XMAXB
FORCE1=SHEAR1-SHIP(K1)
FORCE2=SHEAR2-SH2P(K1)
SHIP(K1)=SHEAR1
SH2P(K1)=SHEAR2
DD=SQRT(SUMB)
GO TO(142,144,146),K1
142 IF(NOPT.EQ.2) GO TO 148
CALL SP(IW,1)
WRITE(IW,328) J,SHEAR1,SHEAR2,FORCE1,FORCE2,DD
GO TO 150
144 WRITE(IW,327) SHEAR1,SHEAR2,FORCE1,FORCE2,DD
GO TO 150
146 WRITE(IW,314) SHEAR1,SHEAR2,FORCE1,FORCE2,DD
GO TO 150
148 CALL SP(IW,1)
WRITE(IW,329) J,SHEAR1,SHEAR2,FORCE1,FORCE2,DD
150 CONTINUE
160 CONTINUE
CALL SP(IW,3)
WRITE(IW,315)
WRITE(IW,316)

```

```

1881.000 C
1882.000
1883.000
1884.000
1885.000
1886.000
1887.000
1888.000
1889.000
1890.000
1891.000
1892.000
1893.000
1894.000
1895.000
1896.000
1897.000
1898.000
1899.000
1900.000
1901.000
1902.000
1903.000
1904.000
1905.000
1906.000
1907.000
1908.000
1909.000 C
1910.000 C
1911.000 C
1912.000
1913.000
1914.000
1915.000
1916.000
1917.000 C
1918.000 C
1919.000 C
1920.000

      RETURN
307 FORMAT(1X, 'MODE NUMBER', 11X, 'FREQUENCY'//)
308 FORMAT(1X, 17, E25.5)
309 FORMAT(1X, 'MODE SHAPES')
310 FORMAT(1X, 'LEVEL', 3X, 'DIRN', 18, 3I12)
311 FORMAT(15, 7F10.0)
312 FORMAT(4F10.0)
313 FORMAT(21X, 'SHEAR OR', 12X, 'FORCE OR', 20X, 'CUMULATIVE',
111X, 'TORSIONAL', 1X, 'STORY', 2X, 'DIRN', 8X, 'TOR. MOMENT',
211X, 'MOMENT', 10X, 'DISP.', 1X, 'LEVEL', 11X, 'CODE 1', 4X,
3'CODE 2', 4X, 'CODE 1', 4X, 'CODE 2')
314 FORMAT(8X, 'THETA', 4F10.3, E11.4)
315 FORMAT(7X, '**CODE 1 - ROOT SUM SQUARE')
316 FORMAT(7X, '**CODE 2 - SUM OF TWO MAXIMUM ABSOLUTE',
1' MODAL VALUES')
317 FORMAT(1X, 14, 2X, 'X TRAN', 4E12.5)
323 FORMAT(7X, 'Y TRAN', 4E12.5)
324 FORMAT(9X, 'ROTN', 4E12.5)
326 FORMAT(1X, 14, 2X, 'Y TRAN', 4E12.5)
327 FORMAT(8X, 'Y', 4F10.3, E11.4)
328 FORMAT(14, 4X, 'X', 4F10.3, E11.4)
329 FORMAT(14, 4X, 'Y', 4F10.3, E11.4)
330 FORMAT(1X, '-----')
334 FORMAT(1X, '-----')
335 FORMAT(1X, 'GLOBAL SEISMIC STORY FORCES AND DISPLACEMENTS')
      END
SUBROUTINE EIGVAL(A, SM, VAL, VEC, CO, NA, BB)
      CALCULATES MODE SHAPES AND FREQUENCIES
COMMON/CON/NST, NFI, NOPT, NEQ, NMULP
COMMON/SPEC/NM, LM, XMU, SF, BETA, MI, ALLOW, TD, G, NP, TC(10), SAC(10)
COMMON/MISC/IR, IW, AI(3), AJ(3), IDN, NH
DIMENSION A(NEQ, NEQ), SM(NEQ), VEC(NEQ, NM), VAL(NM)
DIMENSION CO(NEQ), NA(NEQ), BB(NEQ)
      FORM DYNAMIC MATRIX
      CALL DYNO(A, SM, NA, BB)

```

```

1921.000 SH=0.
1922.000 K=1.
1923.000 LL=0
1924.000 RI=1.E-06
1925.000 NVT=0
1926.000 C
1927.000 C
1928.000 C
1929.000 C
1930.000 C
1931.000 C
1932.000 C
1933.000 C
1934.000 C
1935.000 C
1936.000 C
1937.000 C
1938.000 C
1939.000 C
1940.000 C
1941.000 C
1942.000 C
1943.000 C
1944.000 C
1945.000 C
1946.000 C
1947.000 C
1948.000 C
1949.000 C
1950.000 C
1951.000 C
1952.000 C
1953.000 C
1954.000 C
1955.000 C
1956.000 C
1957.000 C
1958.000 C
1959.000 C
1960.000 C

      SH=0.
      K=1.
      LL=0
      RI=1.E-06
      NVT=0

      SET INITIAL VALUE OF THE TRIAL VECTORS
10 DO 20 I=1,NEQ
   CO(I)=1.
20 CONTINUE
   NI=1

      OBTAIN NEW TRIAL VECTORS
30 DO 40 I=1,NEQ
   VEC(I,K)=0.
   DO 40 J=1,NEQ
     VEC(I,K)=VEC(I,K)+A(I,J)*CO(J)
40 CONTINUE

      INITIALIZE NEW TRIAL VECTOR BY DIVIDING IT BY THE
      LARGEST ELEMENT
      NT=1
      CMAX=ABS(VEC(1,K))
      DO 50 I=2,NEQ
        IF(ABS(VEC(I,K)).LT.CMAX)GO TO 50
      CMAX=ABS(VEC(I,K))
      NT=I
50 CONTINUE
      IT=CO(NT)/VEC(NT,K)
      TTS=TT/ABS(IT)
      CMAX=CMAX*TTS
      DO 60 I=1,NEQ
        VEC(I,K)=VEC(I,K)/CMAX
60 CONTINUE

      TEST IF THE ORIGIN REQUIRES SHIFTING

```

```

1961.000 C
1962.000
1963.000
1964.000
1965.000
1966.000
1967.000
1968.000
1969.000
1970.000 C
1971.000 C
1972.000 C
1973.000
1974.000
1975.000
1976.000
1977.000
1978.000
1979.000
1980.000
1981.000
1982.000
1983.000
1984.000
1985.000
1986.000
1987.000 C
1988.000 C
1989.000 C
1990.000
1991.000
1992.000 C
1993.000 C
1994.000 C
1995.000
1996.000
1997.000
1998.000
1999.000 C
2000.000 C

IF(LL.LT.LM) GO TO 65
IF(NVT.EQ.1)GO TO 65
IF(NI.LT.10)GO TO 65
VT=SH+ABS(1./CMAX)
KT=K-1
VVT=SQRT(VT)
IF(ABS(VVT-VAL(KT)).GT..05*VAL(KT))GO TO 100
NVT=1

TEST FOR CONVERGENCE

65 ERMAX=0.
DO 70 I=1,NEQ
IF(ABS(VEC(I,K)).LT.R1)GO TO 70
ERR=(CO(I)-VEC(I,K))/VEC(I,K)
ERR=ABS(ERR)
IF(ERR.GT.ERMAX) ERMAX=ERR
70 CONTINUE

IF(ERMAX.LE.ALLOW) GO TO 120
IF(NI.GT.MI) GO TO 118
DO 80 I=1,NEQ
CO(I)=VEC(I,K)
80 CONTINUE
NI=NI+1
GO TO 30

HERE TO CALCULATE THE NEW ORIGIN

100 SH=(VT+VAL(KT)*VAL(KT))/2.
REWIND 5

RETRIEVE BUILDING STIFFNESS MATRIX FROM UNIT 5

READ(5)((A(I,J),I=1,NEQ),J=1,NEQ)
DO 110 I=1,NEQ
A(I,I)=A(I,I)-SH*SM(I)
110 CONTINUE

FORM DYNAMIC MATRIX AND MODIFY IT TO SWEEP THE MODE

```

```

2001.000 C      SHAPES ALREADY FOUND
2002.000 C
2003.000 IDN=1
2004.000 CALL DYN0(A,SM,NA,BB)
2005.000 DO 112 I=1,NEQ
2006.000 CO(I)=VEC(I,K)
2007.000 112 CONTINUE
2008.000 KK=K-1
2009.000 DO 115 N=1,KK
2010.000 HAT=1./((VAL(N)*VAL(N)-SH)
2011.000 DO 115 I=1,NEQ
2012.000 DO 115 J=1,NEQ
2013.000 A(I,J)=A(I,J)-HAT*VEC(J,N)*VEC(I,N)*SM(J)
2014.000 115 CONTINUE
2015.000 LL=0
2016.000 NI=NI+1
2017.000 GO TO 30
2018.000 C
2019.000 C      PRINT OUT WARNING MESSAGE
2020.000 C
2021.000 118 WRITE(IW,301)
2022.000 WRITE(IW,302) K
2023.000 WRITE(IW,303) NI
2024.000 WRITE(IW,304) ERMAS
2025.000 C
2026.000 C      CALCULATE DIMENSIONLESS FREQUENCY
2027.000 C      AND NORMALISED MODE SHAPE
2028.000 C
2029.000 120 VAL(K)=SQRT(SH+1/CMAX)
2030.000 ALFA=0.
2031.000 DO 130 I=1,NEQ
2032.000 ALFA=ALFA+VEC(I,K)*VEC(I,K)*SM(I)
2033.000 130 CONTINUE
2034.000 BETAA=SQRT(ALFA)
2035.000 DO 140 I=1,NEQ
2036.000 VEC(I,K)=VEC(I,K)/BETAA
2037.000 140 CONTINUE
2038.000 IF(K-NM) 150,200,200
2039.000 150 CONTINUE
2040.000 C

```



```

2041.000 C
2042.000 C
2043.000
2044.000
2045.000
2046.000
2047.000
2048.000
2049.000
2050.000
2051.000
2052.000
2053.000
2054.000
2055.000
2056.000
2057.000
2058.000
2059.000
2060.000 C
2061.000 C
2062.000 C
2063.000
2064.000
2065.000
2066.000 C
2067.000 C
2068.000 C
2069.000
2070.000
2071.000
2072.000
2073.000
2074.000
2075.000
2076.000
2077.000
2078.000
2079.000 C
2080.000 C

      MODIFY DYNAMIC MATRIX TO SWEEP THE MODE SHAPE LAST FOUND

      DO 160 I=1,NEQ
      DO 160 J=1,NEQ
      A(I,J)=A(I,J)-CMAX*VEC(I,K)*VEC(J,K)*SM(J)
160 CONTINUE
      K=K+1
      LL=LL+1
      NVT=0
      GO TO 10
200 RETURN
301 FORMAT(//6X,'** WARNING - ITERATION COUNTER LIMIT',
1, ' REACHED')
302 FORMAT(5X,' MODE NUMBER.....=,I4)
303 FORMAT(5X,' NUMBER OF ITERATION.....=,I4)
304 FORMAT(5X,' MAXIMUM DIFFERENCE BETWEEN SUCCESSIVE',
1, ' TRIAL VECTORS.....=,F12.8)
      END
      SUBROUTINE DYNO(A,SM,NA,BB)
      FORMS DYNAMIC MATRIX; D=INV(A)*SM
      COMMON/CON/NST,NFT,NOPT,NEQ,NMULP
      COMMON/MISC/IR,IW,AI(3),AJ(3),IDN,NH
      DIMENSION A(1),NA(1),SM(1),BB(1)
      INVERT MATRIX A
      IF(IDN.EQ.1) CALL RECIP(A,NA,BB)
      DO 10 I=2,NEQ
      I1=I-1
      IAD=(I-1)*NEQ
      JADR=I
      DO 10 J=1,I1
      IADR=IAD+J
      A(JADR)=A(IADR)
      JADR=JADR+NEQ
10 CONTINUE
      FORM THE PRODUCT OF A INVERSE AND THE MASS MATRIX SM

```

```

2081.000 C
2082.000
2083.000
2084.000
2085.000
2086.000
2087.000
2088.000
2089.000
2090.000
2091.000 C
2092.000 C
2093.000 C
2094.000 C
2095.000
2096.000
2097.000
2098.000
2099.000
2100.000
2101.000
2102.000
2103.000
2104.000
2105.000
2106.000
2107.000
2108.000
2109.000
2110.000
2111.000
2112.000
2113.000
2114.000
2115.000
2116.000
2117.000
2118.000
2119.000
2120.000

      DO 20 J=1,NEQ
      JAD=(J-1)*NEQ
      DO 20 I=1,NEQ
      JADR=JAD+I
      A(JADR)=A(JADR)*SM(J)
20  CONTINUE
      RETURN
      END
      SUBROUTINE RECIP(A,NA,BB)
      INVERTS SYMMETRIC MATRIX A AND STORES INVERTED SYMMETRIC
      HALF IN THE DIAGONAL AND UPPER TRIANGLE OF A
      COMMON/CON/NST,NFT,NOPT,NEQ,NMULP
      DIMENSION A(1),BB(1),NA(1)
      NN=NEQ-1
      NADR=1
      DO 40 J=2,NEQ
      NADR=NADR+NEQ
      LIMIT=NADR+J-1
      N1=1
      DO 30 I=NADR,LIMIT
      IF(A(I).NE.O)GO TO 40
      N1=N1+1
30  CONTINUE
40  NA(J)=N1
      NA(1)=1./A(1)
      DO 80 M=1,NN
      K=M+1
      KAD=(K-1)*NEQ
      KADK=KAD+K
      IN=NA(K)
      IF(IN.LE.M)GO TO 45
      A(KADK)=1./A(KADK)
      GO TO 80
45  DO 60 I=1,M
      IAD=(I-1)*NEQ
      BB(I)=O.

```

```

2121.000 IF(IN.GT.I)GO TO 55
2122.000 DO 50 J=IN,I
2123.000 IADR=IAD+J
2124.000 KADR=KAD+J
2125.000 BB(I)=BB(I)+A(IADR)*A(KADR)
2126.000 50 CONTINUE
2127.000 55 IF(I.EQ.M)GO TO 65
2128.000 II=I+1
2129.000 JF=MAXO(II,IN)
2130.000 JADR=(JF-1)*NEQ+I
2131.000 DO 60 J=JF,M
2132.000 KADR=KAD+J
2133.000 BB(I)=BB(I)+A(JADR)*A(KADR)
2134.000 JADR=JADR+NEQ
2135.000 60 CONTINUE
2136.000 65 D=0.
2137.000 DO 70 I=IN,M
2138.000 KADR=KAD+I
2139.000 D=D+A(KADR)*BB(I)
2140.000 D=A(KADK)-D
2141.000 A(KADK)=1./D
2142.000 DO 75 I=1,M
2143.000 IADR=KAD+I
2144.000 75 A(IADR)=-BB(I)*A(KADK)
2145.000 DO 80 I=1,M
2146.000 JADR=(I-1)*NEQ+I
2147.000 DO 80 J=1,M
2148.000 KADR=KAD+J
2149.000 A(JADR)=A(JADR)-BB(I)*A(KADR)
2150.000 JADR=JADR+NEQ
2151.000 80 CONTINUE
2152.000 RETURN
2153.000 END
2154.000 SUBROUTINE SP(IW,ISP)
2155.000 C
2156.000 C SKIPS A LINE IN THE OUTPUT
2157.000 C
2158.000 DO 10 I=1,ISP
2159.000 10 WRITE(IW,I)
2160.000 1 FORMAT(' ')

```

```

2161.000 RETURN
2162.000 END
2163.000 SUBROUTINE RESP(SM,DA,VEC,SA,I)
2164.000 C
2165.000 C CALCULATES MODAL STORY DISPLACEMENTS
2166.000 C
2167.000 COMMON/CON/NSI,NFT,NOPT,NEQ,NMULP
2168.000 COMMON/SPEC/NM,LM,XMU,SF,BETA,MI,ALLOW,TD,G,NP,TC(10),SAC(10)
2169.000 DIMENSION VEC(NEQ,NM),SM(NEQ),DA(NEQ,NM)
2170.000 SUM=0.
2171.000 IF(NOPT.EQ.3) GO TO 20
2172.000 C
2173.000 C HERE WHEN NOPT=1 OR 2, CALCULATE MODAL PARTICIPATION FACTOR
2174.000 C
2175.000 DO 10 J=1,NEQ
2176.000 SUM=SUM+VEC(J,I)*SM(J)
2177.000 10 CONTINUE
2178.000 SI=SUM
2179.000 GO TO 40
2180.000 C
2181.000 C HERE WHEN NOPT=3, CALCULATE MODAL PARTICIPATION FACTOR
2182.000 C
2183.000 20 LAST=NEQ-2
2184.000 DO 30 J=1, LAST, 3
2185.000 SUM=SUM+VEC(J,I)*SM(J)*COS(BETA)
2186.000 30 CONTINUE
2187.000 LAST=NEQ-1
2188.000 DO 35 J=2, LAST, 3
2189.000 SUM=SUM+VEC(J,I)*SM(J)*SIN(BETA)
2190.000 35 CONTINUE
2191.000 SI=SUM
2192.000 40 DO 50 J=1,NEQ
2193.000 DA(J,I)=VEC(J,I)*SI*SA
2194.000 50 CONTINUE
2195.000 RETURN
2196.000 END
2197.000 SUBROUTINE SFORCE(S,DN,M,NDIR,DA,F,X)
2198.000 C
2199.000 C CALCULATES AND PRINTS MODAL STORY DISPLACEMENTS AND
2200.000 C FORCES IN INDIVIDUAL FRAME

```

```

2201.000 C
2202.000
2203.000
2204.000
2205.000
2206.000
2207.000
2208.000
2209.000
2210.000
2211.000
2212.000 C
2213.000 C
2214.000 C
2215.000
2216.000
2217.000 C
2218.000 C
2219.000 C
2220.000
2221.000
2222.000
2223.000
2224.000
2225.000
2226.000
2227.000
2228.000
2229.000
2230.000
2231.000
2232.000
2233.000
2234.000 C
2235.000 C
2236.000 C
2237.000
2238.000
2239.000
2240.000

COMMON/CON/NST,NFT,NOPT,NEQ,NMULP
COMMON/MISC/IWI,IW,AI(3),AJ(3),IDN,NH
DIMENSION S(NST,NST),DN(NST,NFT),M(NFT),NDIR(NFT)
DIMENSION DA(NEQ,NH),F(NST),X(NST)
IFP=0
IWI=1
REWIND 2
NM=NH
CALL SP(IW,3)
GO TO(10,60,110),NOPT

HERE FOR NOPT=1
10 DO 50 NF=1,NFT
IF(M(NF).EQ.IFP) GO TO 20
READ(2,306)NFG,MST
DO 12 LL=1,MST
READ(2,307)(S(LL,J),J=1,MST)
12 CONTINUE
IFP=M(NF)
20 IF(NDIR(NF).EQ.2) GO TO 50
WRITE(IWI) NF,MST,NM
CALL SP(IW,3)
DO 40 K=1,NM
DO 30 I=1,MST
F(I)=0.
DO 30 J=1,MST
F(I)=F(I)+S(I,J)*DA(J,K)
30 CONTINUE
WRITE MODAL STORY FORCES ON UNIT IWI
WRITE(IWI) (F(I),I=1,MST)
40 CONTINUE
50 CONTINUE
RETURN

```

```

2241.000 C
2242.000 C
2243.000 C
2244.000
2245.000
2246.000 C
2247.000 C
2248.000 C
2249.000
2250.000
2251.000
2252.000
2253.000
2254.000
2255.000
2256.000
2257.000
2258.000
2259.000
2260.000
2261.000
2262.000
2263.000 C
2264.000 C
2265.000 C
2266.000
2267.000
2268.000
2269.000
2270.000 C
2271.000 C
2272.000 C
2273.000
2274.000 C
2275.000 C
2276.000 C
2277.000 C
2278.000
2279.000
2280.000

      HERE FOR NOPT=2
60 DO 100 NF=1,NFT
   IF(M(NF).EQ.IFP) GO TO 70

      READ CONDENSED FRAME STIFFNESS MATRIX FROM UNIT 2

      READ(2,306)NFG,MST
      DO 62 LL=1,MST
      READ(2,307)(S(LL,J),J=1,MST)
62 CONTINUE
      IFP=M(NF)
70 IF(NDIR(NF).EQ.1) GO TO 100
   WRITE(IW1) NF,MST,NM
   CALL SP(IW,3)
   DO 90 K=1,NM
   DO 80 I=1,MST
      F(I)=0.
      DO 80 J=1,MST
      F(I)=F(I)+S(I,J)*DA(J,K)
80 CONTINUE

      WRITE STORY FORCES ON UNIT IW1

      WRITE(IW1) (F(I),I=1,MST)
90 CONTINUE
100 CONTINUE
      RETURN

      HERE FOR NOPT=3
110 DO 180 NF=1,NFT
   IF(M(NF).EQ.IFP) GO TO 120

      READ CONDENSED FRAME STIFFNESS MATRIX FROM UNIT 2

      READ(2,306)NFG,MST
      DO 112 LL=1,MST
      READ(2,307)(S(LL,J),J=1,MST)

```

```

2281.000 112 CONTINUE
2282.000 IFP=M(NF)
2283.000 120 CALL SP(IW,3)
2284.000 WRITE(IWI),NF,MST,NM
2285.000 IF(NDIR(NF).EQ.2) GO TO 130
2286.000 AI(1)=1.
2287.000 AI(2)=0.
2288.000 GO TO 140
2289.000 130 AI(1)=0.
2290.000 AI(2)=1.
2291.000 140 DO 180 K=1,NM
2292.000 DO 150 I=1,MST
2293.000 X(I)=0.
2294.000 II=(I-1)*3
2295.000 AI(3)=DN(I,NF)
2296.000 DO 150 J=1,3
2297.000 ID=II+J
2298.000 X(I)=X(I)+AI(J)*DA(ID,K)
2299.000 150 CONTINUE
2300.000 DO 160 I=1,MST
2301.000 F(I)=0.
2302.000 DO 160 J=1,MST
2303.000 F(I)=F(I)+S(I,J)*X(J)
2304.000 160 CONTINUE
2305.000 C
2306.000 C WRITE STORY FORCES ON WNIT IWI
2307.000 C
2308.000 WRITE(IWI) (F(I),I=1,MST)
2309.000 180 CONTINUE
2310.000 RETURN
2311.000 C
2312.000 306 FORMAT(2I8)
2313.000 307 FORMAT(6E12.5)
2314.000 C
2315.000 C
2316.000 C SUBROUTINE ROOT(NF,G,NDIR,NEL,NQ,MCL,MBM,IDN,IRR,
2317.000 +A,NA,SD,SLD,FB1,FB2,FB3,FB4,FC1,FC2,FC3,FC4,DSP)
2318.000 COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
2319.000 +CL(240),CA(240),CB(240),CG(240),CEF(240),N1,NS,
2320.000 +NST(10),NBAY(9),NBS(9),DISQ(30)

```

```

2321.000
2322.000
2323.000
2324.000
2325.000 C
2326.000
2327.000
2328.000
2329.000
2330.000
2331.000
2332.000
2333.000
2334.000
2335.000
2336.000
2337.000
2338.000
2339.000
2340.000
2341.000
2342.000
2343.000
2344.000
2345.000
2346.000
2347.000
2348.000
2349.000
2350.000
2351.000
2352.000
2353.000
2354.000
2355.000
2356.000
2357.000
2358.000
2359.000
2360.000

COMMON/CON/NNS,NFT,NOPT,NQ,NMULP
DIMENSION A(NEL),NA(NQ),SD(NQ),SLD(NQ),FB1(MBM),
+FB2(MBM),FB3(MBM),FB4(MBM),FC1(MCL),FC2(MCL),FC3(MCL),
+FC4(MCL),DSP(NNS),NDIR(NFT)

REWIND 4
REWIND 6
IR=4
IRI=6
IW=7
JOPT=0
IFR=0
OPT=2
20 DO 100 II=1,NFG
  READ(IR,500) NT,NS,NFR
  READ(IR,500) (NST(I),I=1,NT)
  READ(IR,500) (NBAY(I),I=1,NT)
  NT2=NT-1
  IF(NT.GT.1) READ(IR,500) (NBS(I),I=1,NT2)
  READ(IR,500) NBM,NCL
  READ(IR,525) (BL(I),BA(I),BB(I),BEF(I),I=1,NBM)
  READ(IR,520) (CL(I),CA(I),CB(I),CG(I),CEF(I),CAL(I),I=1,NCL)
  READ(IRI,510) NEQ,NL
  READ(IRI,515) (NA(I),I=1,NEQ)
  READ(IRI,520) (A(I),I=1,NL)
  DO 90 I=1,NFR
    IFR=IFR+1
    IF(NOPT.LE.2.AND.NOPT.NE.NDIR(IFR)) GO TO 90
    READ(IRH,NF,MST,NM
  DO 28 IL=1,NBM
    FB1(IL)=0.
    FB2(IL)=0.
    FB3(IL)=0.
    FB4(IL)=0.
  28 CONTINUE
  DO 30 IL=1,NCL
    FC1(IL)=0.
    FC2(IL)=0.
    FC3(IL)=0.
    FC4(IL)=0.

```



```

2361.000
2362.000
2363.000
2364.000
2365.000
2366.000
2367.000
2368.000
2369.000
2370.000
2371.000
2372.000
2373.000
2374.000
2375.000
2376.000
2377.000
2378.000
2379.000
2380.000
2381.000
2382.000
2383.000
2384.000
2385.000
2386.000
2387.000
2388.000
2389.000
2390.000 C
2391.000
2392.000
2393.000
2394.000
2395.000
2396.000
2397.000
2398.000
2399.000
2400.000

30 CONTINUE
DO 35 IL=1,MST
35 DSP(IL)=O.O
DO 70 J=1,NM
READ(IRR) (SD(JJ),JJ=1,MST)
DO 40 KK=1,NEQ
40 SLD(KK)=O.
K=O
IS=O
DO 60 IJ=1,NT
NN=NST(IJ)
NB=NBAY(IJ)+1
NB1=NB+1
DO 50 IK=1,NN
IS=IS+1
K=K+NB1
SLD(K)=SD(IS)
K=K+NB
50 CONTINUE
60 CONTINUE
CALL BSUB(JOPT,NEL,NQO,A,NA,SD,SLD,NEQ)
IF(IDN.NE.O) GO TO 65
WRITE(3) (SLD(J),J=1,NEQ)
GO TO 70
65 MB=MBA+1
CALL FORCE(OPT,MBM,MCL,NQO,NNS,FB1,FB2,FB3,FB4,FC1,FC2,
+FC3,FC4,NA(1),NA(MB),SD(1),SD(MB),SLD,DSP)
70 CONTINUE
IF(IDN.EQ.O) GO TO 90
DO 75 IB=1,NBM
FB1(IB)=SQRT(FB1(IB))
FB2(IB)=SQRT(FB2(IB))
FB3(IB)=SQRT(FB3(IB))
FB4(IB)=SQRT(FB4(IB))
75 CONTINUE
DO 80 IC=1,NCL
FC1(IC)=SQRT(FC1(IC))
FC2(IC)=SQRT(FC2(IC))
FC3(IC)=SQRT(FC3(IC))

```

```

2401.000 FC4(IC)=SQRT(FC4(IC))
2402.000 80 CONTINUE
2403.000 DO 85 IL=1,NS
2404.000 85 DSP(IL)=SQRT(DSP(IL))
2405.000 WRITE(5) (FB1(IB),FB2(IB),FB3(IB),FB4(IB),IB=1,NBM)
2406.000 WRITE(5) (FC1(IC),FC2(IC),FC3(IC),FC4(IC),IC=1,NCL)
2407.000 WRITE(5) (DSP(IL),IL=1,NS)
2408.000 90 CONTINUE
2409.000 100 CONTINUE
2410.000 RETURN
2411.000 C
2412.000 500 FORMAT(5I5)
2413.000 510 FORMAT(2I8)
2414.000 515 FORMAT(I8)
2415.000 520 FORMAT(6E12.5)
2416.000 525 FORMAT(5E12.5)
2417.000 END
2418.000 C
2419.000 SUBROUTINE HLOAD(A,DN,M,NDIR,XBAR,YBAR)
2420.000 C
2421.000 C PREPARE LATERAL LOAD VECTOR
2422.000 C AND FIND FRAME STORY FORCES
2423.000 C
2424.000 COMMON/CON/NSI,NFI,NPT,NEQ,NMULP
2425.000 COMMON/COM1/HX(40),HY(40),DX(40),DY(40),NA(120),FH(120),
2426.000 +DA(120,3),F(40),X(40),S(1710)
2427.000 COMMON/CMSC/NGR,NWL,NDYN,NSYS,IOUT,SMULP(12,8)
2428.000 COMMON/MISC/IW1,IW,AI(3),AJ(3),IDN,NH
2429.000 DIMENSION A(NEQ,NEQ),XBAR(NSI),YBAR(NSI),M(NFI),
2430.000 +NDIR(NFI),DN(NSI,NFI)
2431.000 IR=3
2432.000 IW=7
2433.000 WRITE(IW,110)
2434.000 CALL RECIP(A,NA,FH)
2435.000 IDN=0
2436.000 DO 99 IH=1,NWL
2437.000 WRITE(IW,150) IH
2438.000 DO 60 I=1,NSI
2439.000 READ(IR,100) HX(I),HY(I),DX(I),DY(I)
2440.000 GO TO (30,40,50),NPT

```

```

2441.000      30 FH(I)=HX(I)
2442.000      GO TO 60
2443.000      40 FH(I)=HY(I)
2444.000      GO TO 60
2445.000      50 XB=XBAR(I)
2446.000      YB=YBAR(I)
2447.000      K=3*I
2448.000      FH(K)=HY(I)*(DX(I)-XB)-HX(I)*(DY(I)-YB)
2449.000      FH(K-2)=HX(I)
2450.000      FH(K-1)=HY(I)
2451.000      60 CONTINUE
2452.000      WRITE(IW,200) (I,HX(I),HY(I),DX(I),DY(I),I=1,NST)
2453.000      C
2454.000      DO 90 I=1,NEQ
2455.000      SB=0.
2456.000      DO 70 J=1,I
2457.000      70 SB=SB+A(J,I)*FH(J)
2458.000      IF(I.EQ.NEQ) GO TO 90
2459.000      II=I+1
2460.000      DO 80 K=II,NEQ
2461.000      80 SB=SB+A(I,K)*FH(K)
2462.000      90 DA(I,IH)=SB
2463.000      C
2464.000      99 CONTINUE
2465.000      NH=NWL
2466.000      CALL SFORCE(S,DN,M,NDIR,DA,F,X)
2467.000      RETURN
2468.000      100 FORMAT(5F15.5)
2469.000      110 FORMAT(//
2470.000      +//-----//
2471.000      +// LATERAL LOAD DATA//
2472.000      +//-----//)
2473.000      150 FORMAT(// LATERAL LOAD CASE',I3,' :',//
2474.000      +//-----//
2475.000      +// LEVEL X-FORCE Y-FORCE X-DIST
2476.000      200 FORMAT(I5,2X,4F10.3)
2477.000      END
2478.000      C
2479.000      SUBROUTINE SYSTEM(NFG,NQ,MBM,MCL,NNS,B,FS1,FS2,FS3,
2480.000      +FS4,SQB1,SQB2,SQB3,SQB4,SQC1,SQC2,SQC3,SQC4,FX1,FX2,

```

```

2481.000      +FX3,FX4,BTA,DSP)
2482.000      C
2483.000      C
2484.000      C
2485.000      C
2486.000      C
2487.000      C
2488.000      C
2489.000      C
2490.000      C
2491.000      C
2492.000      C
2493.000      C
2494.000      C
2495.000      C
2496.000      C
2497.000      C
2498.000      C
2499.000      C
2500.000      C
2501.000      C
2502.000      C
2503.000      C
2504.000      C
2505.000      C
2506.000      C
2507.000      C
2508.000      C
2509.000      C
2510.000      C
2511.000      C
2512.000      C
2513.000      C
2514.000      C
2515.000      C
2516.000      C
2517.000      C
2518.000      C
2519.000      C
2520.000      C

      +FX3,FX4,BTA,DSP)
      FINDS AND WRITES MEMBER FORCES FOR LOAD SYSTEMS

      COMMON/COM1/BL(210),BA(210),BB(210),BG(210),BEF(210),
      +CL(240),CA(240),CB(240),CG(240),CEF(240),CAL(240),NT,NS,
      +NST(10),NBAY(9),NBS(9),DISQ(30)
      DIMENSION B(NQQ),FS1(MBM),FS2(MBM),SQB4(MBM),SQC1(MCL),
      +SQB1(MBM),SQB2(MBM),SQB3(MBM),FS3(MBM),FS4(MBM),
      +SQC2(MCL),SQC3(MCL),SQC4(MCL),FX1(4,MBM),FX2(4,MBM),
      +FX3(4,MBM),FX4(4,MBM),BTA(7,NQQ),DSP(NNS)
      COMMON/COMSYS/NGR,NWL,NDYN,NSYS,IOUT,SMULP(12,8)

      REWIND 4
      IR=4
      IW=7
      K=NGR
      WRITE(IW,600)
      N1=NGR+1
      N2=NGR+NWL
      NCASE=N2+NDYN
      WRITE(IW,100) (I,I=1,NSYS)
      IF(NGR.NE.0) GO TO 2
      WRITE(IW,150)
      GO TO 4
2  DO 3 J=1,NGR
      WRITE(IW,200) J,(SMULP(I,J),I=1,NSYS)
3  CONTINUE
      REWIND 8
4  IF(NWL.NE.0) GO TO 5
      WRITE(IW,250)
      GO TO 7
5  CONTINUE
      REWIND 3
      DO 6 J=1,NWL
      K=K+1
      WRITE(IW,300) J,(SMULP(I,K),I=1,NSYS)
6  CONTINUE
7  IF(NDYN.NE.0) GO TO 8
      WRITE(IW,350)

```

```

2521.000      GO TO 9
2522.000      8 K=K+1
2523.000      REWIND 5
2524.000      WRITE(IW,400) (SMULP(I,K),I=1,NSYS)
2525.000      9 IFR=0
2526.000      DO 60 I=1,NFG
2527.000      READ(IR,500) NT,NS,NFR
2528.000      READ(IR,500) (NST(I),I=1,NT)
2529.000      READ(IR,500) (NBAY(I),I=1,NT)
2530.000      NT2=NT-1
2531.000      IF(NT.GT.1) READ(IR,500) (NBS(I),I=1,NT2)
2532.000      READ(IR,500) NBM,NCL,NEQ
2533.000      READ(IR,525) (BL(I),BA(I),BB(I),BG(I),BEF(I),I=1,NBM)
2534.000      READ(IR,520) (CL(I),CA(I),CB(I),CG(I),CEF(I),CAL(I),I=1,NCL)
2535.000      DO 55 IJ=1,NFR
2536.000      IFR=IFR+1
2537.000      NK=NCASE
2538.000      WRITE(IW,550) IFR
2539.000      IF(NGR.EQ.0) GO TO 15
2540.000      READ(8) INDL
2541.000      IF(INDL.NE.0) GO TO 15
2542.000      DO 10 I=1,NGR
2543.000      READ(8) (BTA(I,J),J=1,NEQ)
2544.000      READ(8) (FX1(I,J),FX2(I,J),FX3(I,J),FX4(I,J),J=1,NBM)
2545.000      10 CONTINUE
2546.000      15 IF(NWL.EQ.0) GO TO 20
2547.000      DO 18 I=N1,N2
2548.000      18 READ(3) (BTA(I,J),J=1,NEQ)
2549.000      20 IF(NDYN.EQ.0) GO TO 25
2550.000      READ(5) (SQB1(K),SQB2(K),SQB3(K),SQB4(K),K=1,NBM)
2551.000      READ(5) (SOC1(K),SOC2(K),SOC3(K),SOC4(K),K=1,NCL)
2552.000      READ(5) (DISQ(K),K=1,NS)
2553.000      NK=NK+1
2554.000      SMLT=1
2555.000      25 DO 50 I=1,NSYS
2556.000      WRITE(IW,560) I
2557.000      DO 26 IB=1,NEQ
2558.000      26 B(IB)=0.0
2559.000      DO 28 IK=1,NBM
2560.000      FSI(IK)=0.0

```

```

2561.000 FS2(IK)=0.0
2562.000 FS3(IK)=0.0
2563.000 FS4(IK)=0.0
2564.000 28 CONTINUE
2565.000 L=0
2566.000 DO 40 J=1,NK
2567.000 XMUL=SMULP(I,J)
2568.000 IF(XMUL.EQ.0) GO TO 40
2569.000 DO 30 K=1,NEQ
2570.000 30 B(K)=B(K)+XMUL*BTA(J,K)
2571.000 IF(J.GT.NGR) GO TO 40
2572.000 L=L+1
2573.000 DO 35 K=1,NBM
2574.000 FS1(K)=FS1(K)+XMUL*FX1(J,K)
2575.000 FS2(K)=FS2(K)+XMUL*FX2(J,K)
2576.000 FS3(K)=FS3(K)+XMUL*FX3(J,K)
2577.000 FS4(K)=FS4(K)+XMUL*FX4(J,K)
2578.000 35 CONTINUE
2579.000 40 CONTINUE
2580.000 OPT=0
2581.000 IF(L.GT.0) OPT=1
2582.000 IF(NDYN.EQ.0) GO TO 45
2583.000 XMUL=SMULP(I,NCASE)
2584.000 IF(XMUL.EQ.0) GO TO 45
2585.000 OPT=OPT+3
2586.000 SMLT=XMUL/SMLT
2587.000 DO 42 K=1,NBM
2588.000 SQB1(K)=SMLT*SQB1(K)
2589.000 SQB2(K)=SMLT*SQB2(K)
2590.000 SQB3(K)=SMLT*SQB3(K)
2591.000 SQB4(K)=SMLT*SQB4(K)
2592.000 42 CONTINUE
2593.000 DO 43 K=1,NCL
2594.000 SQC1(K)=SMLT*SQC1(K)
2595.000 SQC2(K)=SMLT*SQC2(K)
2596.000 SQC3(K)=SMLT*SQC3(K)
2597.000 SQC4(K)=SMLT*SQC4(K)
2598.000 43 CONTINUE
2599.000 DO 44 K=1,NS
2600.000 44 DISO(K)=SMLT*DISO(K)

```

```

2601.000 SMLT=XMUL
2602.000 45 CALL FORCE(OPT,MBM,MCL,NQ,NNS,FS1,FS2,FS3,FS4,SQC1,
2603.000 +SQC2,SQC3,SQC4,SQB1,SQB2,SQB3,SQB4,B,DSP)
2604.000 50 CONTINUE
2605.000 55 CONTINUE
2606.000 60 CONTINUE
2607.000 WRITE(IW,700)
2608.000 IF(NDYN.NE.0) WRITE(IW,675)
2609.000 RETURN
2610.000 100 FORMAT(/20X,'LOAD CASE MULTIPLYING FACTORS :',//
2611.000 + ' LOAD SYSTEM',6X,12I7)
2612.000 150 FORMAT(/20X,'*** GRAVITY LOAD CASES ARE NOT CONSIDERED ***')
2613.000 200 FORMAT(/ ' GRAVITY LOAD CASE',I2,I1,12F7.3)
2614.000 250 FORMAT(/20X,'*** LATERAL LOAD CASES ARE NOT CONSIDERED ***')
2615.000 300 FORMAT(/ ' LATERAL LOAD CASE',I2,I1,12F7.3)
2616.000 350 FORMAT(/20X,'*** SEISMIC LOAD CASE IS NOT CONSIDERED ***')
2617.000 400 FORMAT(/ ' SEISMIC LOAD CASE ',I2F7.3)
2618.000 500 FORMAT(5I5)
2619.000 520 FORMAT(6E12.5)
2620.000 525 FORMAT(5E12.5)
2621.000 550 FORMAT(/
2622.000 + ' -----//
2623.000 + ' FRAME NO.',I3, /
2624.000 + ' -----//)
2625.000 560 FORMAT(/
2626.000 + ' FOR LOAD SYSTEM',I3, /
2627.000 + ' -----//)
2628.000 600 FORMAT(1H, /
2629.000 +27X,'*****')
2630.000 +27X,'* ANALYSIS FOR LOAD SYSTEMS *', /
2631.000 +27X,'*****')
2632.000 675 FORMAT( ' *** NOTE ://
2633.000 + ' MAX = STATIC FORCES OR DISPLACEMENTS + SEISMIC//
2634.000 + ' ROOT SUM SQUARE FORCES OR DISPLACEMENTS//
2635.000 + ' MIN = STATIC FORCES OR DISPLACEMENTS - SEISMIC//
2636.000 + ' ROOT SUM SQUARE FORCES OR DISPLACEMENTS//
2637.000 700 FORMAT(/ ' *** SIGN USED ***', /
2638.000 + ' MOM/TORN: CNTR.CLCK +VE ; FORCE/DISP: UP +VE, RHT +VE')
2639.000 END

```

END

1 2 3 4 5 6 7 8 9

FIN