Structural Engineering Report No. 116

REINFORCED CONCRETE COLUMN DESIGN PROGRAM

by

CHI-KUN LEUNG

and

S. H. SIMMONDS

Department of Civil Engineering
The University of Alberta
Edmonton, Alberta, Canada

APRIL, 1984

REINFORCED CONCRETE COLUMN DESIGN PROGRAM

by

Chi-Kun Leung

and

S. H. Simmonds

Structural Engineering Report No. 116

Department of Civil Engineering
The University of Alberta
Edmonton, Alberta, Canada

April 1984

ABSTRACT

The development of a computer program, **COLUMN1**, for the design of reinforced concrete columns is presented. The columns may be either tied or spiral, long or short, braced or unbraced, and have either circular or rectangular cross-sections. Loading may be defined as either factored or unfactored with bending about one or both axes. For a given loading and material properties the column will select a satisfactory cross-section including longitudinal and lateral reinforcement. Provision is made for the designer to specify all or any portion of the cross-section, and to place limitations on bar size and bar numbers. All designs conform to either building standard CSA3-A23.3-M77 using S.I. units or ACI 318-77 using Imperial units. The program may be run in either batch or interactive mode with a hard copy output option if the latter mode is selected.

The design philosophy and engineering logic of the program are discussed and a number of examples to demonstrate most of the features of the program are included.

ACKNOWLEDGEMENT

This report describes a computer program for the design of reinforced concrete written by Mr. Chi-Kun Leung under the direction of Dr. S.H. Simmonds. The authors acknowledge the financial assistance for this study provided by the National Science and Engineering Research Council of Canada through operating grant NSERC A-1691.

TABLE OF CONTENTS

CHAPTER		PAGE

LIST OF TABLES

LIST OF FIGURES

NOTATIONS

Roman letters, capital

Ast	total area of longitudinal reinforcement in a column cross section
Asx	total area of reinforcement distributed along x-faces in a column
Asy	total area of reinforcement distributed along y-faces in a column
Cm	moment gradient correction factor
Cx	size of rectangular column measured in x-direction for which moment about y-axis are being determined
Cy	size of rectangular column measured in y-direction for which moment about x-axis are being determined
Ds	distance from centroid of extreme compression reinforcement to centroid of extreme tension reinforcement in opposite half of member
Ec	modulus of elasticity of concrete
EI	flexural stiffness of a column
Es	modulus of elasticity of steel
Ig	moment of inertia of gross concrete section about the centroidal axis, neglecting the reinforcement
Ise	moment of inertia of reinforcement about the centroidal axis of the member cross-section
Mc	magnified factored moment to be used for design of compression member
Mn	nominal moment strength at section
Mu	factored moment at section
Mux	factored moment about x-axis which is primarily resisted by the reinforcement in y-faces

Muy	factored moment about y-axis which is primarily resisted by the reinforcement in x-faces
M_1	value of smaller factored end moment on compression member calculated by conventional elastic frame analysis, positive if member bent in single curvature, negative if bent in double curvature
M_2	value of larger factored end moment on compression member calculated by conventional elastic frame analysis, always positive
Nx	number of bars in x-face only including corner bars
Ny	number of bars in y-face only
Pc	critical axial load
Pn	nominal axial load strength at given eccentricity
Pu	factored axial load strength at given eccentricity
Sb	the actual clear bar spacing in x-faces or y-faces
Sbx	the actual clear bar spacing in x-faces
Sby	the actual clear bar spacing in y-faces

Roman letters, lower case

a	depth of equivalent rectangular stress block
c	distance from extreme compression fiber to neutral axis
d	distance from extreme compression fiber to centroid of tension reinforcement
d_b	nominal diameter of bars
dc	thickness of clear concrete cover
d_s	lateral reinforcement diameter
d'	distance from extreme compression fibre to the center of the bar located closest thereto

e	eccentricity of design load parallel to axis measured from the centroid of the section
f'c	specified compressive strength of concrete
f_y	specified yield strength of reinforcement
h	diameter of a round column or side of a rectangular column
k	effective length factor for compression members
lu	unsupported length of compression member
x-face	lateral faces or edges in y-direction with column dimension Cy
y-face	end faces in x-direction with column dimension Cx

Greek letters

β_1	a coefficient relating the depth of the equivalent rectangular stress block to the depth from the compression face to the neutral axis
β_d	ratio of maximum factored dead load moment to maximum factored total load moment, always positive between zero and one
Ξcu	maximum usable strain at the extreme concrete compression fiber assumed equal to 0.003
δ	moment magnification factor for columns
ρ_g	maximum permissible ratio of longitudinal reinforcement in a column
ρ_t	total area of reinforcement divided by total area of cross-section in a column
ϕ	capacity reduction factor
θ	angle between longitudinal bars distributed along circular core

Subscripts

$_x$ about x-axis

$_y$ about y-axis

1. INTRODUCTION

1.1 Column Design

The design of a reinforced concrete column is essentially one of trial and error. Design consists of finding a cross-section that will support satisfactorily both an axial load and a bending moment. However, since the capacity of a section to carry axial load is dependent on the magnitude of the moment that is acting (in some cases the moment is a function of the section dimensions), there are no closed form solutions for determining a section uniquely. The problem is compounded when moments act about both axes of the section.

Traditionally, to reduce the required number of trials to obtain a satisfactory design, use was made of non-dimensional 'interaction diagrams'. These are plots of axial load vs. moment capacity relationships for specific material properties and section geometry. Such plots are practical for bending about one axis only, so that, in practice, it is common to consider bending only about one axis and ignore the moment in the other direction, a practice that is sometimes unrealistic and unsafe. Thus, there is a need for a computer program that will select an economical and adequate section for reinforced concrete columns with bending about both axes.

Reinforced concrete columns are classified as either tied or spiral. Both columns have longitudinal and lateral reinforcement. With a tied column the lateral reinforcement prevents premature buckling of the longitudinal reinforcement but does not confine the concrete. With a spiral column the longitudinal reinforcement and concrete is enclosed in a spiral of reinforcement of sufficient strength and spacing that the concrete cannot fail before a significant triaxial stress condition is established in the concrete. The difference in the two types is the mode of failure. Tied columns will fail shortly after reaching maximum axial load with little ductility whereas spiral columns will demonstrate great ductility with little or no loss of load carrying capacity. The different behavior modes can be predicted and to take advantage of the enhanced characteristics of spiral columns it is necessary to meet a number of requirements that have been determined by experiment. These are specified in building codes with the result that a computer program for column design in practice must conform to many limitations imposed by the building codes.

Reinforced concrete columns must also be constructed manually. This imposes certain restrictions on dimension tolerances, number and arrangement of reinforcement bars, bar diameters, etc. For use in a design office, a computer program must permit the designer to impose on the design a number of restraints that are user dependent, that is they can vary from run to run depending on the requirements of a particular construction project.

A number of programs have been developed for design of reinforced concrete columns, such as RC.Column by Grant T. Halvorsen (15), and the column program, COLS, by Portland Cement Association (7). These programs are essentially 'checking' programs, that is the user enters concrete dimensions and reinforcement size and pattern and the program checks to see if the section will satisfy a given loading. If input section is not satisfactory, only minor alterations to the section will be made for further trials. If after these minor alterations the section is not satisfactory, the user must begin again by entering a new section which hopefully will be closer to a satisfactory section. While such programs are useful in the hands of those designers who from past experience, can predict closely the size of section that should be entered, they are of little use to the inexperienced designer. The purpose of this project is to develop a program that can be used by students to obtain quickly designs for reinforced concrete columns so that they can obtain such experience. For this reason the program must be capable of selecting a column section, including reinforcement, for any given loading, column length and bracing that conforms to specified building code requirements and any user imposed limitations, when only the loading is given. Such a program is, of course, even more useful in the hands of an experienced designer and can be incorporated into a integrated CAD system.

1.2 Organization of Thesis

The underlying philosophy as to the requirements, scope and limitations of the program are given in Chapter 2. Chapter 3 contains a discussion of the engineering decisions that are required in order to obtain complete designs. Design examples are discussed in Chapter 4. Chapter 5 contains a brief summary and conclusions. A Users' Manual for the program and a listing are given in the Appendices.

2. DESIGN PHILOSOPHY

2.1 Scope of Columns Considered

The purpose of the program, *COLUMN1* , is to provide engineers with a tool that will design most columns that will be encountered in practice in accordance with either the CSA3-A23.3-M77 (8) or the ACI318-77 (1,2) building codes. For this reason, both long and short columns subject to either uni-axial or bi-axial loading are considered but cross-sections are restricted to either rectangular or circular shapes. Futhermore, columns may be braced or unbraced. For long columns the moment magnifier method is used. For long braced columns the moment magnifier is computed from the input geometry and loading. However, for long unbraced columns the sums for the axial load and critical load for all columns in that floor are required and these values must be entered manually.

In keeping with the generality of solution but to limit the possible longitudinal bar arrangements, the following restrictions on reinforcement geometry are imposed. For circular columns the reinforcement consist of an even number of bars which must be located around the perimeter in one layer for both tied and spiral columns. The program checks for minimum number of bars and adequate bar spacing to satisfy building code requirements. Square column sections but not rectangular sections may be used for spiral columns in which case the reinforcement must be placed in a circular pattern. For tied rectangular columns the reinforcing is restricted to being parallel to the faces, but may be located on two opposite faces only or along all faces providing that double symmetry of the cross-section is maintained. Individual bars may be specified so that the usual restriction placed on interaction diagrams in most handbooks which require reinforcement along all faces to have equal areas in each face is not required.

In keeping with the concept of designer control, the user has, for each problem, the option of specifying either 'design' or 'check'. For a check problem, the user enters both the concrete dimensions and the complete reinforcement pattern and the program determines whether the section entered is adequate for the specified loadings. For a design problem, the user specifies only as little or as much of the cross-section details as he wishes and the computer completes the design of the section to safely carry the loading.

Prior to checking a given cross-section for adequacy to resist the loading, the program checks the reinforcement for limitation of percentage of reinforcement, bar size and bar spacing. If the longitudinal reinforcement was entered by the user and was found to violate any of the imposed limitations, a message indicating the violation is printed and execution of the program terminates. Once a satisfactory section is determined the reinforcement pattern is checked to determine whether a tangential lap splice is possible and if so an appropriate message is printed.

For tied columns the program will select the tie diameter and required tie spacing. The actual choice of tie pattern is generally standard and is left to the designer. For spiral columns the program selects the spiral diameter and pitch and the minimum number of vertical spacers to satisfy codes requirements.

2.2 Operational Modes

A single column design or multiple column designs can be solved during a single run. The user has the option of running the program in either batch or interactive mode, however, the mode selected applies to all designs that are to be solved during that run.

In batch mode operation, the user prepares a data input file, and obtains output after each run. Unlike the interactive mode, the user cannot interact with the design solution during the run.

In interactive mode, the user enters the required input line by line after being prompted by the computer. The output is displayed on the terminal screen. This output includes a summary of the column design. At this point, the user has the option of either revising the input and rerunning the design or proceeding to the next design.

In interactive mode all output during the run appears on the terminal screen. At the end of each design, the final solution is stored in a file from which a hard copy output may be listed, if requested, at the end of the run.

2.3 Control for Design

The first line of input for each design is a user defined heading by which the design and output are identified.

When the appropriate building code is entered, the corresponding units and bar designations are assigned. In addition all data for which the code gives specific values are assigned default values. Some of these parameters established by default, such as concrete cover, maximum percentage of reinforcement, etc. can be overridden by entering desired values. Other parameters, such as strength reduction factors for tied and spiral columns or bar designations, can only be altered by editing the program and, therefore cannot be done during a program run. The entry of data that is not code dependent, such as the magnitude of loading, column length, reinforcement pattern and material strength properties cannot be assigned default values and must be entered.

The final line of input for each design is a coded number that indicates whether another design follows or whether the run is to be terminated. For batch mode, when another design follows the coded number indicates at which stage following reading of the new heading new data is to be read since input not read is taken to be the same as for the previous design. In interactive mode, the user has a number of options depending on whether he wants to revise the design or proceed to the next design.

For each column design a number of different loading conditions may be considered. With initial loading case the user indicates whether it is a 'design' loading or a 'check' loading. However for multiple loading cases, subsequent loadings have to be a check at which case the program checks the cross-section which was either entered or designed for the initial loading for adequacy. In batch mode, the total number of load cases must be entered. In interactive mode, the total number of load cases is not required as the user has an option to revise the loading conditions as he wishes.

For detail input procedure and data assigned by default, the reader is referred to the USER'S MANUAL in Appendix A.

2.4 Output

The user has an option of either obtaining a short output or a long output. The short output provides sufficient information to evaluate the design and to construct the column and would be the usual output in design practice. The long output provides additional detail information, including an echo check of default values for all input data and results of certain intermediate calculations that permit monitoring of the design for debugging purposes or for evaluating a peculiar condition.

The output for each design consists of the heading entered by the user to identify the design, the date and time problem was run to distinguish between multiple runs of the same design if the heading was not altered. The heading is also printed at the top of all subsequent pages for the same design.

The short output also contains a summary of the control parameters, such as building code, column type and the loading conditions considered. When the loading case is a design loading condition, the output will include the loading and column section properties. When the loading case is a check loading, the output will consist of the loading and a message indicating whether or not the section was adequate. The exception to this rule is that when there is a single loading case and it is a check loading, then both the input loading and section properties are printed before the message of adequacy.

3. ENGINEERING LOGIC

3.1 Introduction

This chapter discusses the engineering decisions used in the column design program. In summary, the various steps are as follows:

(a) enter or select initial concrete section that is minimum consistent with loading,

(b) enter or select longitudinal reinforcement that is minimum consistent with reinforcing restraints and geometry,

(c) consider slenderness effects such as magnification factor, and minimum moment to determine design loading,

(d) compute axial load and moment capacity of trial section, and compare these values with applied axial load and required moment,

(e) modify trial section if required,

(f) evaluate longitudinal splices and select lateral reinforcement.

These steps are described in detail in the following sections.

3.2 Nomenclature and Sign Convention

The positive directions of the axial load and bending moments acting at the centroid of the column section are shown in Fig. 3.2.1. Reinforcement may be placed along only the x-face or the y-face, or both as long as double symmetry with respect to axes through the centroid is maintained. The

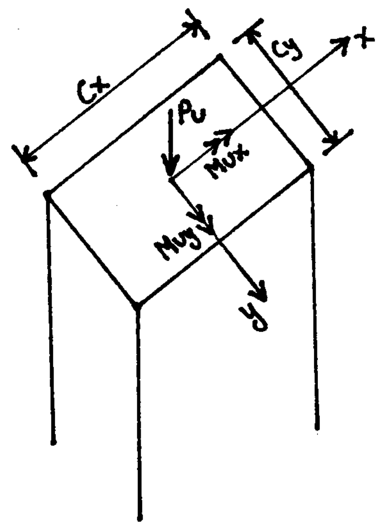


Fig. 3.2.1 Loading acting at the centroid of column section

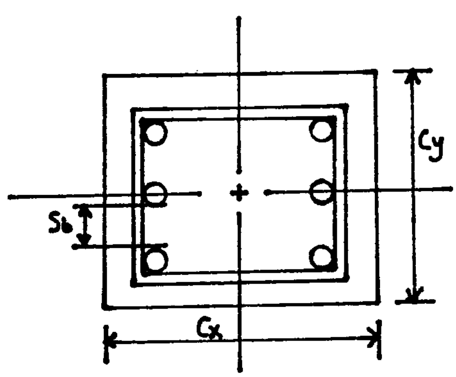


Fig. 3.2.2 Bars distributed in x-faces only

bars placed in x and y -faces are perpendicular to the respective axes as shown in Fig. 3.2.2 and 3.2.3. When bars are placed in all faces, the corner bars are considered to be located along the x-faces as shown in Fig. 3.2.4.

3.3 Uniaxial and Biaxial Bending

In uniaxial bending, user must indicate whether the compression member is braced or unbraced about the x-axis and enter the effective length factor and the unsupported length about the x-axis. The moment about the x-axis, Mux, acting with Pu, is primarily resisted by the reinforcement in the y-faces.

In biaxial bending, in addition to the above data input, the user must indicate whether the compression member is braced or unbraced about the y-axis and enter the effective length factor and the unsupported length about the y-axis. Normally, the unsupported length about the x-axis is the same as that about the y-axis. The moment about the y-axis, Muy, acting with Pu, is primarily resisted by the reinforcement in the x-faces.

When the entered moment value is less than the minimum moment specified in the building codes, the specified minimum moment is used.

The program determines whether column deflects in single or double curvature by examining the signs assigned to the moments at the top and bottom of the column. A negative sign to one of the entered moments indicates the

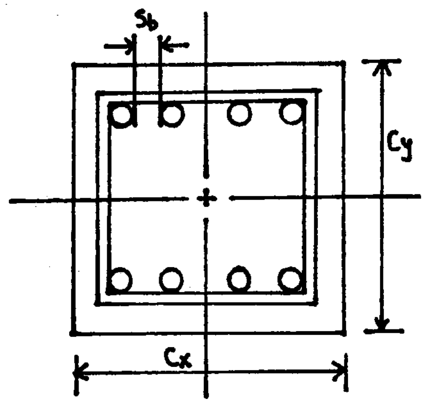


Fig. 3.2.3 Bars distributed in y-faces only

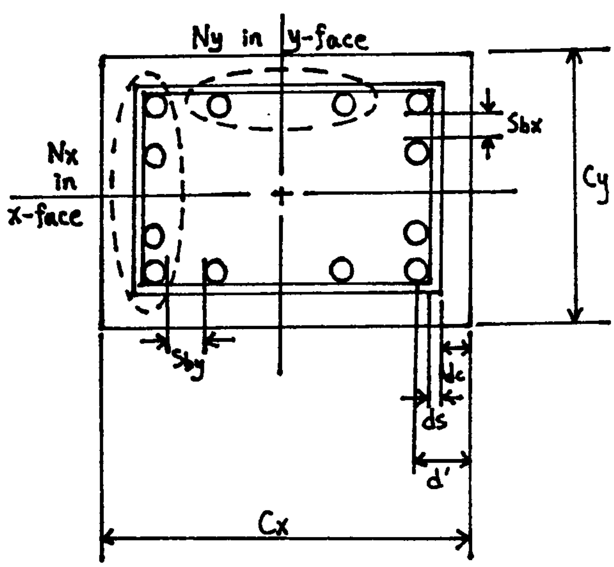


Fig. 3.2.4 Bars distributed in both faces

column deflects in double curvature. When the signs are the same, the column deflects in single curvature. The design moment is the maximum absolute value of the moment entered at either top or bottom of the column.

3.4 Concrete Dimensions

3.4.1 General Consideration

Dimensions of the column cross-section may be either user entered or determined by the program. If determined by the program each column dimension is selected to be an integer multiple of a predetermined length increment which has a default value of 2 in. for Imperial units and 50 mm for SI units or it may be user entered. To maintain a minimum initial trial so that all subsequent trials will be larger, all computed dimensions are rounded down to the next smaller value. For example, a computed value between 16.1 in. and 17.9 in. using the 2 in. default value would be selected as 16 in. Similarly, a value between 201 mm and 249 mm would be selected as 200 mm. If, however, the column cross section dimensions are entered either as specific values or, for a rectangular column as a single dimension and a column aspect ratio, the program will not alter these values, even if they are not an integer multiple of the length increment.

A minimum concrete column dimension is built into the program which is considered to be the minimum dimension that can be physically built with the type of materials envisioned by the code. This minimum is 8 in. or 200 mm depending on units used and is the minimum dimension that will be considered in the design mode. If a column dimension less than this minimum is entered, an error message to this effect is printed and the program will go to the next problem.

3.4.2 Selecting Initial Concrete Section

When the column dimensions are not entered, an initial value of the column area, Ag^{o} is computed from Eqn. 3.4.1.

$$Ag^{o} = Pu/\phi\ \alpha c[0.85f'c + \rho_{s}\ (fy - 0.85f'c)] \qquad (3.4.1)$$

where αc is the reduction factor to account for the reduced capacity caused by the bending moments. The values of αc used are 0.8 for tied and 0.85 for spiral columns which are values corresponding approximately for minimum moment for uniaxial bending. The value of ρ_{s} is the maximum reinforcement ratio to be used in this design which may be either user entered or defaults to a value of 0.03. This will result the smallest possible concrete cross-section consistent with the restraints for the initial column dimensions.

If a circular and square column is specified, the initial column dimensions are obtained directly using the values of Ag^{o}. If a rectangular column is specified by a single column dimension or by the column aspect ratio, the unknown values can also be calculated from Ag^{o} only. For a rectangular column when uni-axial bending is specified, and no dimension or aspect ratio is entered, the program computes parameter Cy from Eqn. 3.4.2 which assumes minimum eccentricity based on the input loading.

$$Cy = Mux/(0.1Pu) \qquad (3.4.2)$$

Cx can then be calculated from Ag^{o} only. For a rectangular column when bending about both axes is specified, a column aspect ratio R is determined from Eqn. 3.4.3 which in conjunction with Ag^{o} is used to obtain the initial values of Cx and Cy.

$$R = Cx/Cy = Muy/Mux \qquad (3.4.3)$$

If no moments are entered, R defaults to 1.0. This same aspect ratio is maintained for all future increments of dimension, if required.

3.4.3 Incrementing Concrete Section

Using the initial concrete section, the program attempts to select longitudinal reinforcement consistent with the reinforcing constraints. If no such reinforcement can be found, the column dimensions, if not entered, are incremented in accordance with the following rules and the process of selecting reinforcement repeated. Consistent with the input philosophy, when a concrete dimension is entered it will not be altered by the program. When a dimension is altered the increment will be the single step of the current value of the length increment.

When no column dimensions are entered but the shape is specified as square, both column dimensions are incremented equally by the length increment. If no column dimensions or aspect ratio are entered and the shape entered is rectangular, the smaller cross-section dimension is increased in increments until a square section is obtained. If a still larger section is required, the increments alterate between the directions first Cy then Cx attempting to maintain as close to a square section as possible. When a column aspect ratio is entered, a similar procedure exists except that the dimensions incremented are such as to most closely approximate the entered aspect ratio.

After a possible section of concrete with steel is selected, the slenderness ratio must be checked to determine whether the column is short or long. For uniaxial bending, when the column is long and the critical axial load is less than the nominal axial load, a larger column area is required. This is obtained by computing a new value of Ag as described in Section 3.6 and new column dimensions consistent with the rules for selecting initial trial section as outlined in Section 3.4.2 are obtained.

The trial section is then checked for adequacy in strength. When the section is not adequate, the longitudinal reinforcement is incremented first as described in Section 3.8. Should a suitable reinforcement pattern not be found the concrete dimensions will be increased in accordance with the rules given in the second paragraph of this section.

3.5 Selection of Longitudinal Reinforcement

All reinforcing patterns whether entered or determined by the program must be symmetrical about the centroidal co-ordinate axes. For tied columns, however, bars may be placed along all sides or only on opposite sides if cross-section is square or rectangular. For spiral columns, the bars must be in a circular pattern regardless of whether column cross-section is circular or square.

When using the program in check mode, the reinforcement pattern, the bar size, and the total number of bars must be entered. Normally, the bar size is entered by assigning equal values to the minimum and maximum bar size, but if they are not equal, the program will use only the minimum bar size.

When using the program in design mode, only the reinforcement pattern must be entered. Minimum and maximum bar size may be either user specified, or defaults to #5 and #11 in *ACI* and #15 and #35 in *CSA*. The maximum reinforcement ratio, which controls the maximum area of steel that can be used before the concrete section is incremented, is either entered or defaults to a value of 0.03.

The program computes the initial steel area by multiplying the trial concrete dimensions by the minimum percentage of steel permitted by the controlling building code. This will ensure that the most economical section is tried. When the specified number of bars is not entered, the program then computes the number of bars having a size equal to the minimum bar size and rounds this number upwards to the next even integer. This number is then compared to the maximum number of bars for that section which may be either user entered or defaults to the values given in Table 3.1. When the required total number of bars is greater than the maximum number, the bar size will be increased providing it has not reached the maximum bar size limit.

Dimension		Maximum number of bars
in.	mm	
≤16	≤400	8
18 - 22	450 - 550	12
24 - 30	600 - 750	16
32 - 36	800 - 900	20
etc.		

Table 3.1 Default values for maximum number of bars in section

The values given in Table 3.1 are considered to be representative of those used in practice and are calculated by the program when the column dimension exceeds 400 mm or 16 in. by the following expressions :

for *CSA*

NO. OF BARS =(C'/200 + 1)4

and for *ACI*

NO. OF BARS =(C'/8 + 1)4

where C' is integer equivalent of the column dimension along which the reinforcement is located, but for reinforcement pattern in both faces, the short dimension governs.

For the case when the user wishes to input the number of bars, and bars are in both faces, the number of bars in the x-face and y-face must be specified separately. The number of bars in the x-face includes the corner bars as shown in Fig. 3.2.4. When the specified number of bars is not entered, the program consistent with the pattern input computes the number of bars in the x and y -faces by considering the following conditions. If the bending moment about the x-axis is greater than the moment about the y-axis, the number of bars in the y-faces will be either greater than or equal to the number of bars in the x-faces not including the corner bars. This is accomplished by the following expressions (Fig. 3.5.1):

for bars in x-faces,

Nx = (Nmax+4.)/4.

for bars in y-faces,

Ny = (Nmax-4.)/4. + 0.51

where Nmax is a real number equivalent of the required total

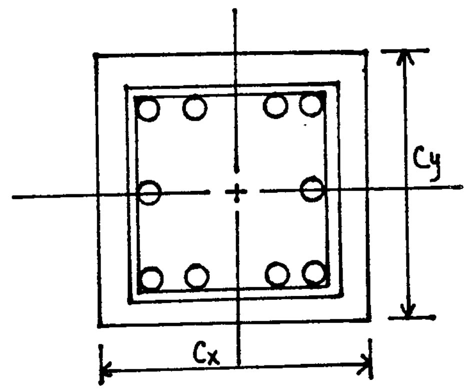


Fig. 3.5.1 Bars in y-faces greater than bars in x-faces

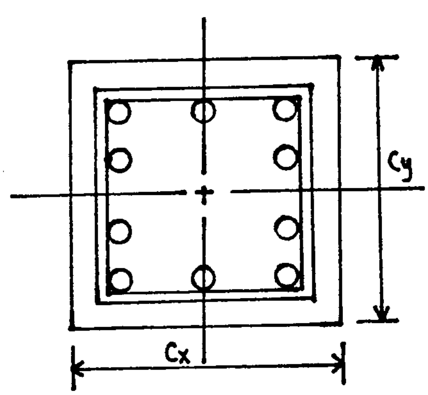


Fig. 3.5.2 Bars in x-faces greater than bars in y-faces

number of bars rounded upwards to the next even whole number, and Nx and Ny are integer equivalents of the values obtained.

Similarly, if moment about the x-axis is less than the moment about the y-axis, bars in the y-faces will be either less than or equal to bars in the x-faces not including the corner bars. The corresponding expressions are (Fig. 3.5.2): for bars in x-faces,

$$Nx = (Nmax+4.)/4. + 0.51$$

for bars in y-faces,

$$Ny = (Nmax-4.)/4.$$

The number of bars is also compared to the minimum number of bars permitted for that cross-section to satisfy building code requirements. When in check mode and the number of bars entered is less than the minimum number, an error message is printed and the program proceeds to the next problem. When in design mode and the number of bars either computed or entered is less than the minimum number, the number of bars is increased to this minimum value.

The required minimum clear bar spacing is determined by taking the larger value of either 1.5 times the bar diameter or a predetermined minimum clear bar spacing which has a default value of 1.5 in. for Imperial units and 40 mm for SI units. In calculating the actual bar spacing, the default values of the minimum diameter of lateral reinforcement and the minimum value of concrete cover specified by the building codes are used unless other values have been entered or the codes control the size of lateral reinforcement diameter consistent with the reinforcing constraints. If the bar spacing is less than the specified minimum spacing, a new design using the next larger bar size will be attempted providing the bar size has not reached the maximum bar size limit or that the number of bars to be used has not been entered, otherwise the computed concrete section will be increased to satisfy the reinforcement constraints. However should the concrete dimensions be user entered an error message indicating that the design cannot be completed with the entered constraints is printed and the program proceeds to next problem.

After a trial column cross-section including longitudinal reinforcement has been found with sufficient capacity to resist the loading, the value of the reinforcement ratio is computed. When this value exceeds the maximum reinforcement ratio applicable to the design, the concrete dimensions, if not user entered, are increased and a new cross-section is obtained. This process is repeated until a cross-section meeting all constraints is determined. Should this not be possible a message indicating the constraints that cannot be satisfied is printed and the program proceeds to the next problem.

3.6 Evaluation of Slenderness Effects

3.6.1 Computation of Moment Magnifier

Before checking the capacities of the trial cross-section, it must be determined whether the column is long, that is the effects of slenderness must be considered. This decision is made in accordance with the building code standards.

For members braced against side-sway, the slenderness effects may be neglected when the slenderness ratio, $k\ell u/r$, is less than $34-12(M_1/M_2)$, in which k is the input value of effective length factor, ℓu is the unsupported length of column, r is the radius of gyration of the cross-section of the column, M_1/M_2 is the ratio of moments at the ends of the column. For members not braced against side-sway, the slenderness effects may be neglected when the slenderness ratio is less than 22. For long columns the slenderness effect is considered using moment magnifier method to increase the applied moments with a magnification factor.

For long braced columns the moment magnifier, δ, is computed from the input loading and geometry.

$$\delta = Cm/[1-(Pu/\phi Pc)] \geq 1.0 \qquad (3.6.1)$$

where

$$Pc = \pi^2 EI/(k\ell u)^2 \qquad (3.6.2)$$

$$Cm = 0.6+0.4(M_1/M_2) \geq 0.4 \qquad (3.6.3)$$

If the equivalent moment factor, Cm , is less than 0.4, it is set to 0.4.

The critical axial load, Pc, is a function of the trial section and involves the flexural stiffness, EI. This is initially computed as

$$EI=(0.2Ec\ Ig + Es\ Ise)/(1+ \beta_d) \qquad (3.6.4)$$

in which Ec is modulus of elasticity of concrete, Ig is moment of inertia of gross concrete section about the centroidal axis, Es is modulus of elasticity of steel, β_d is ratio of maximum factored dead load moment to maximum factored total load moment, Ise is moment of inertia of reinforcement about the centroidal axis of the member cross-section as described in Section 3.6.2. For uniaxial bending about the x-axis, the critical axial load is compared with the applied axial load, Pu. When the critical axial load is greater than the nominal axial load, compute the moment magnifier. If the moment magnifier is less than 1.0, it is set to 1.0. However when the critical axial load is less than the nominal axial load, a larger column area is required. This is obtained by using a conservative value of EI from Eqn. 3.6.5 and computing a new value of Ig from Eqn. 3.6.6.

$$EI=0.4Ec\ Ig/[1+ \beta_d] \qquad (3.6.5)$$

$$Ig=\frac{Pu[1+ \beta_d][k\ell u/\pi]^2}{\phi 0.4Ec} \qquad (3.6.6)$$

in which the value of Pu/ϕ is used instead of Pc. When a new value of column area, Ag as a function of Ig, is obtained, new column dimensions consistent with the rules for selecting initial trial section as outlined in Section 3.4.2

are obtained. The process of selecting reinforcement and checking slenderness effects is repeated until the critical axial load is greater than the nominal axial load.

For long unbraced columns the sum for the axial load, ΣPu , and critical load, ΣPc , for all columns in that floor are required and these values must be entered manually. However when these values are not entered, a message indicating that insufficient information to compute moment magnifier is printed and the program proceeds to the next problem. The moment magnifier for long unbraced column is computed as follows :

$$\delta = Cm/[1-(\Sigma Pu/\phi\, \Sigma Pc)]$$

in which Cm=1.0

If the term $\phi \Sigma Pc$ is less than ΣPu, an error message to this effect is printed and the program proceeds to the next problem.

3.6.2 Evaluation of Ise

The evaluation of the moment of inertia of reinforcement Ise for long braced columns with bars distributed in four different patterns is described in the following parts.

(a) Considering moment of inertia of reinforcement is taken about the centroidal axis of the member cross section, and using direct application of statics, the general formula for moment of inertia of reinforcement is:

$$Ise = \Sigma_i [Asi Xsi^2]$$

in which Asi is the total area of bars at i line, Xsi is the distance from the centroid of bars at i line to the centroidal axis of the section.

For bars in the x-face only, moment of inertia of reinforcement about y-axis (Fig. 3.6.1):

$$Isey = 0.25 Ast \gamma x^2 \, Cx^2$$

where

$$\gamma x = [Cx-2dc-2ds-db]/Cx$$

Also, moment of inertia of reinforcement about x-axis (Fig. 3.6.2):

$$Isex = [4Ast\Sigma \, y(i)^2]/Nmax$$

where

$$y(i) = 0.5\gamma y Cy-(i-1)(db+Sb)$$

$$\gamma y = [Cy-2dc-2ds-db]/Cy$$

$$i = 1, L, \qquad L = (Nmax-1.)/4. + 0.51$$

in which y(i) is the distance from the centroid of bars at i line to the x-axis, i is the line where bars are located starting from the extreme bars, the term L here is rounded upwards to the nearer next integer.

(b) Similarly, for bars in the y-face only, moment of inertia of reinforcement about y-axis:

$$Isey = [4Ast\Sigma \, x(i)^2]/Nmax$$

where

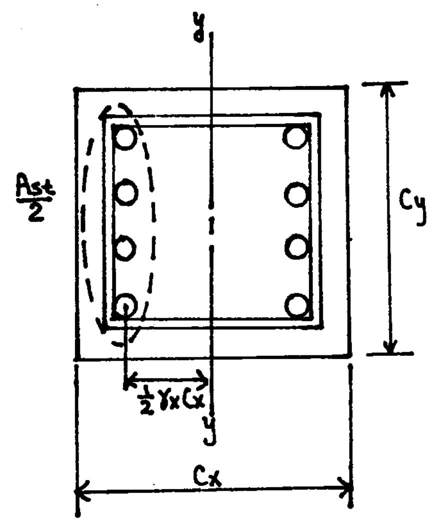


Fig. 3.6.1 Bars in x-faces only w.r.t. y-axis

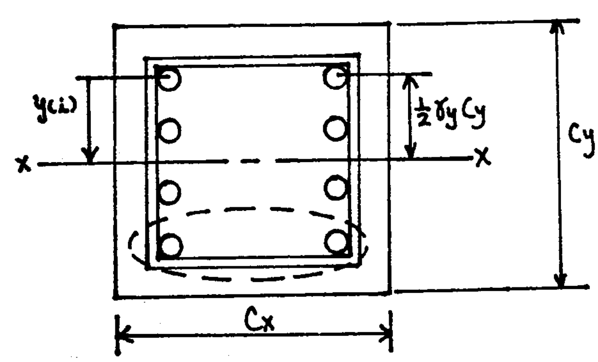


Fig. 3.6.2 Bars in x-faces only w.r.t. x-axis

$$x(i)=0.5\gamma x Cx-(i-1)(db+Sb)$$

$$\gamma x=[Cx-2dc-2ds-db]/Cx$$

$$i=1,L, \quad L=(Nmax-1.)/4. +0.51$$

in which X(i) is the distance from the centroid of bars at i line to the y-axis, the terms i and L here are similar to those defined in part (a).

Also, moment of inertia of reinforcement about x-axis:

$$Isex=0.25Ast\gamma y^2 Cy^2$$

where

$$\gamma y=[Cy-2dc-2ds-db]/Cy$$

(c) For bars in both faces, moment of inertia of reinforcement about y-axis (Fig. 3.6.3):

when no bars in y-face,

$$Isey=0.25Asx\gamma x^2 Cx^2$$

otherwise,

$$Isey=0.25Asx\gamma x^2 Cx^2 +(2Asy/Ny)\Sigma x(i)^2$$

where

$$x(i)=0.5\gamma x Cx-db-Sby-(i-1)(Sby+db)$$

$$i=1,L, \quad L=(2.Ny-1.)/4. +0.51$$

in which i is the line where bars are located starting from the extreme bars in y-faces, the term L here is defaults to 1 when bars are less than two in each y-face.

Also, moment of inertia of reinforcement about x-axis (Fig. 3.6.4):

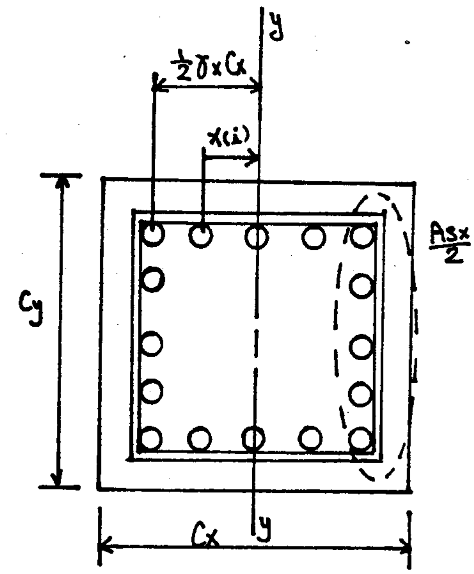


Fig. 3.6.3 Bars in both faces w.r.t. y-axis

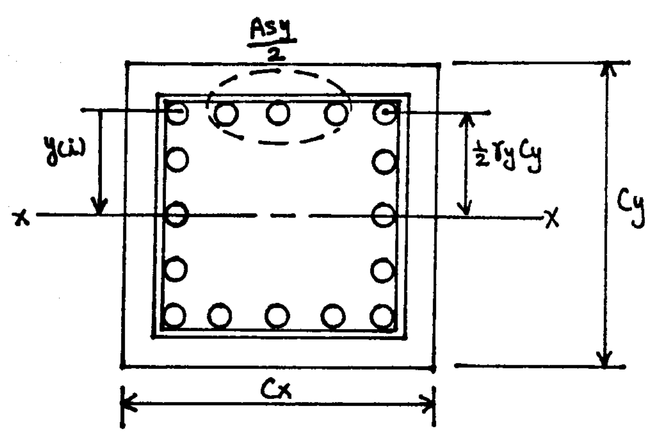


Fig. 3.6.4 Bars in both faces w.r.t. x-axis

$$Isex=0.25Asy\gamma y^2\ Cy^2 +(2Asx/Nx)\Sigma y(i)^2$$

where

$$y(i)=0.5\gamma yCy-(i-1)(Sbx+db)$$

$$i=1,L, \qquad L=[2.Nx-1.]/4. +0.51$$

in which the terms i and L here are similar to those defined in part (a).

(d) For bars in round face, moment of inertia of reinforcement about y-axis (Fig. 3.6.5):

$$Isey=[Ast(0.5Ds^2 +4\Sigma\ x(i)^2]/Nmax$$

where

$$x(i)=(0.5Ds)\cos(i\theta)$$

$$i=1,L, \qquad L=(Nmax-3.)/4. +0.51$$

in which i is the line, where bars are located not in diameter centroidal axis, starting from the extreme bars, the term L here is similar to that defined in part (a).

Also, moment of inertia of reinforcement about x-axis:

$$Isex=[Ast(0.5MDs^2 +4\Sigma\ y(i)^2]/Nmax$$

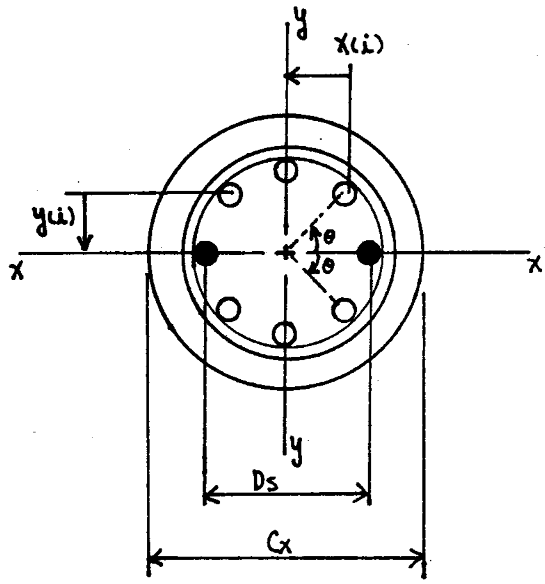
where

$$y(i)=(0.5Ds)\sin(i\theta)$$

$$M=(Nmax-L)/Nmax$$

$$i=1,L, \qquad L=(Nmax-4.)/4. +0.51$$

in which the term M is defaults to 1 when bars are located in the centroidal y-axis of the member cross section, otherwise, M is defaults to 0.



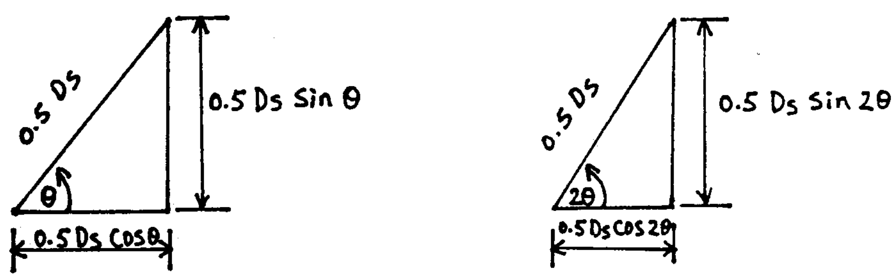


Fig. 3.6.5 Bars in circular core w.r.t. both axes

Note : The center bars in centroidal x-axis are used as relative reference line for measuring θ

3.6.3 Minimum Moment

Determine whether design of compression members is governed by either the minimum eccentricities specified in the building codes or the actual computed eccentricity. When the actual computed eccentricity is less than the specified minimum, the design moment is based on the specified minimum eccentricity rather than the entered moment.

3.7 Evaluation of Strength Capacity

When the strength capacity is computed, the trial section and loading have been modified for slenderness and minimum moment effects. For uniaxial bending, assume strains in steel and concrete are linearly distributed and the maximum usable compression strain in the extreme fibre of the concrete is limited to 0.003. When steel strains are greater than that corresponding to fy, the stress in the reinforcement is taken equal to fy (Fig. 3.7.1). Stresses in the concrete are assumed to follow the equivalent rectangular stress block for concrete in flexure.

To determine the section capacity it is necessary to find the location of the neutral axis. This is accomplished by considering the location of neutral axis as the unknown value and expressing the internal forces as a function of this value.

Theoretically it is possible to sum the moments of these internal forces about the point of application for the applied axial load. This could permit determining the

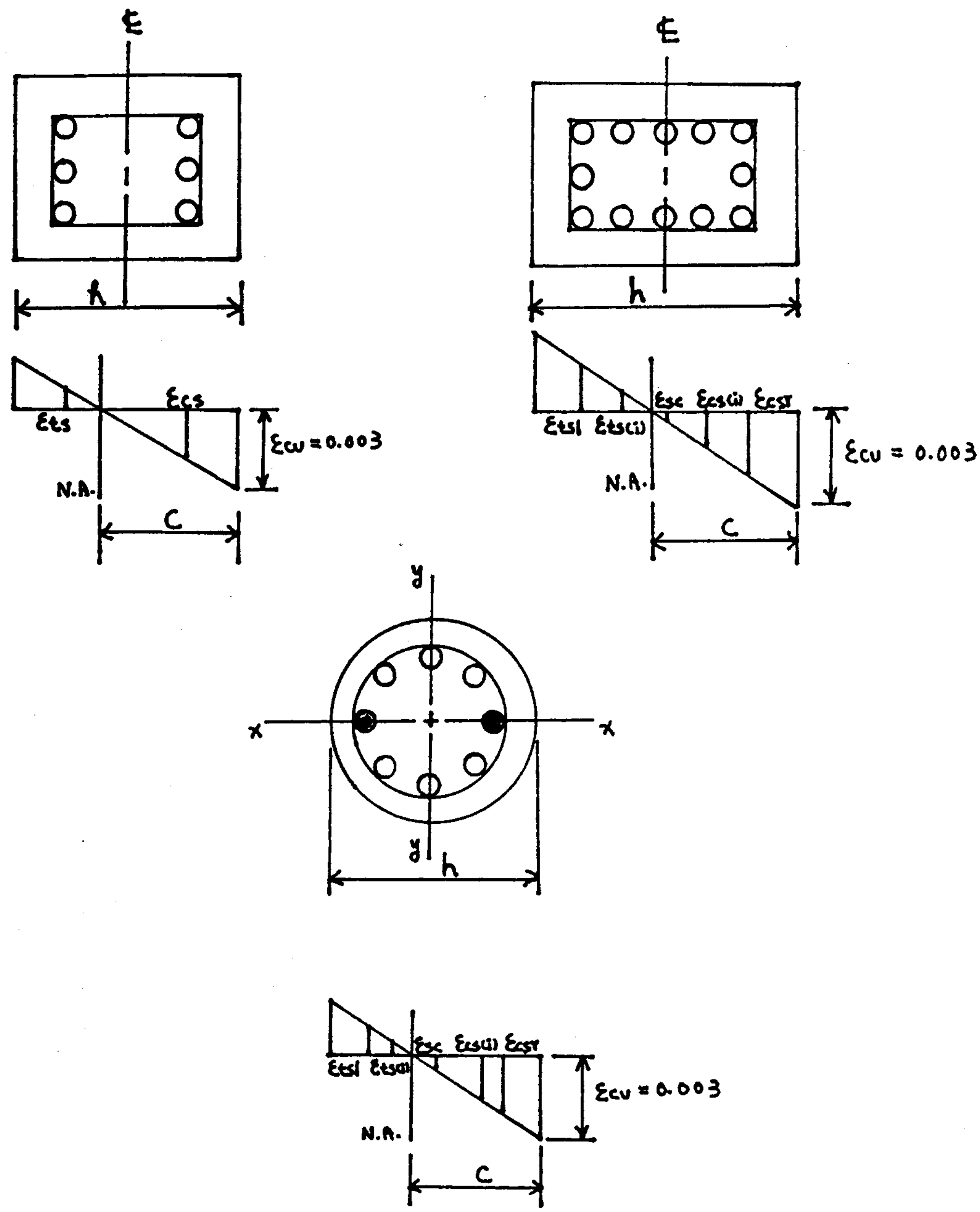


Fig. 3.7.1 Linear strain distribution for bars in two opposite faces only and in both faces and in circular core

Note : ε_{tsl} = strain in extreme tension bars

$\varepsilon_{ts(i)}$ = strain in tension steel at i column line

ε_{sc} = strain in center bars

$\varepsilon_{cs(i)}$ = strain in compression steel at i column line

ε_{csr} = strain in extreme compression bars location of the neutral axis for the minimum required moment capacity which also corresponds to the maximum axial load capacity. This axial load capacity will then be computed and compared to the applied axial load. If axial load capacity is greater section is satisfactory. If not trial section must be increased.

Although the program follows the above philosophy, the procedure was altered to overcome certain practical difficulties. For example unless a specific value of the location of neutral axis is used, it is not known whether the steel has yielded. In addition if depth of equivalent rectangular stress block is longer than the distance from extreme compression fibre to centroid of tension reinforcement, a correction for the effect of concrete displaced by compression steel will be considered. For these reasons, the location of the neutral axis was determined using the interval halving technique as described in the following paragraph of this section (Fig. 3.7.2).

The procedure of the interval halving technique is as follows:

(i) Initially two limits for the location of the neutral axis are determined; the upper limit as h/β_1 and the lower limit as the location of neutral axis at balanced condition

(ii) Using a location of neutral axis as midway between the two limits established in (i) compute the moment capacity.

(iii) If capacity is greater than applied moment, set lower limit as location of neutral axis obtained in (ii)

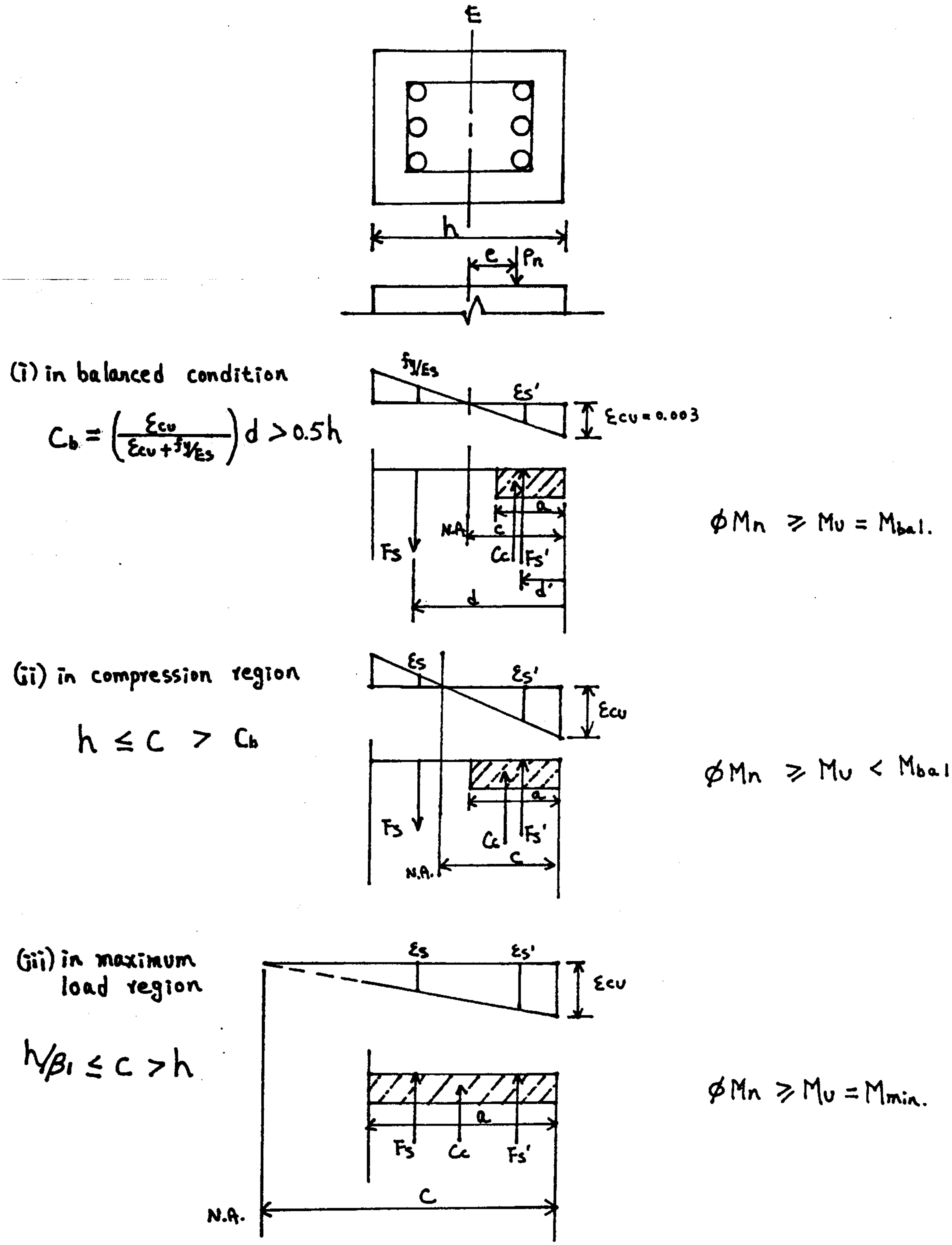


Fig. 3.7.2 Strength capacity at different location of neutral axis

Note : Cc = force in concrete 'fibre'
Fs' = force in compression steel
Fs = force in tension steel

and compute new location as midway between limits.

(iv) If capacity is less than applied moment, set upper limit as location of neutral axis obtained in (ii) and compute new location as midway between limits.

(v) Procedure repeated until difference between upper and lower limits is equal to or less than 5 mm or 0.2 in..

Using the above procedure the axial capacity will always be greater than that corresponding to balanced conditions, which is generally the most economical solution. However for uniaxial bending and in check and design mode using the above procedure the axial load corresponding to the moment capacity used for selecting the position of neutral axis will be the maximum axial load and not the minimum axial load. Therefore the program should not be used in check and design mode for members approaching beams, that is for members with larger bending moments about one axis but relatively small axial loads.

The determination of the strength capacity in concrete Cc for circular columns as well as for square columns with a circular core are more complicated than rectangular columns because of the difficulty in treating the circular segments and the steel bars placed around the circular core, however the same philosophy is followed.

For biaxial bending, a similar analysis is performed for Muy, the moment about y-axis acting in conjunction with Pu. The trial neutral axis and the strength capacity about y-axis are determined.

3.7.1 Adequacy Check in Strength

Before the resisting strength capacities of the trial section are compared with applied strength capacities, it must be determined whether the assumed capacity reduction factor of 0.7 for tied and 0.75 for spiral needs to be modified. This is accomplished by comparing the applied axial load Pu with P' which is the smaller of 0.1(f'c)Ag or Pb at the balanced condition. When Pu is less than P', a new capacity reduction factor is obtained from the program using the expressions (Fig. 3.7.3) :

$$\text{New } \phi = \phi + (0.9-\phi)(P'-Pu)/P'$$

The axial load capacity is checked against the applied axial load. When the axial load capacity is less than 97% of the applied axial load, in design mode, the section is modified, but when it happens in check mode a message is printed and the program proceeds to the next problem.

The investigation of adequacy check in strength in biaxial bending is not so simple as in uniaxial bending. In biaxial bending, the applied axial load Pu is tested against 0.1(f'c)Ag. When the value of Pu is greater than 0.1(f'c)Ag, the Bresler Reciprocal Load Method is used to verify the selected section, otherwise the PCA Load Contour Method is used.

In the Bresler Reciprocal Load Method, the general equation used to calcuate the provided axial load capacity, ϕPi, for a selected column section under biaxial bending is:

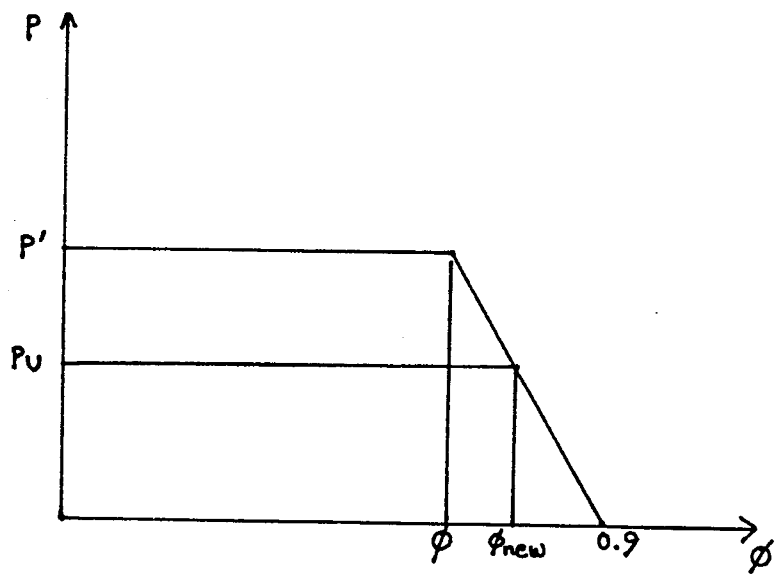


Fig. 3.7.3 Strength-Capacity reduction factor

$$1/(\phi Pi) = 1/(\phi Pnx) + 1/(\phi Pny) - 1/(\phi Po)$$

where

Pi=approximation of nominal axial load strength at eccentricities ex and ey

Pnx=nominal axial load strength for eccentricity ey along the y-axis only (x-axis is axis of bending)

Pny=nominal axial load strength for eccentricity ex along the x-axis only (y-axis is axis of bending)

Po=nominal axial load strength for zero eccentricity

in which Po = 0.85f'c(Ag-Ast) + fy(Ast)

In making an adequacy check, the provided axial load capacity, ØPi, is compared with the applied axial load Pu, in a manner similar to that in uniaxial bending.

The equation is simple in form and the variables are easy to determine and program. The equation also gives reasonable results for loads above balanced condition load (14). Bresler (25) indicates that the experimental test results have shown the equation to be reasonably accurate when flexure does not govern design.

When the value of Pu is less than 0.1(f'c)Ag, the PCA Load Contour Method is used to check the selected section. This approach, developed by Parme, Nieves and Gouwens, is an extension of the Bresler Load Contour Method described in the last paragraph of this section by approximating the load contour equations as straight lines (14).

$$(Mox)equiv = Mux + Muy(Mox/Moy)[(1-Bb)/Bb]$$

for $Muy/Mux \le Moy/Mox$

or $(Moy)equiv = Muy + Mux(Moy/Mox)[(1-Bb)/Bb]$

for $Muy/Mux \ge Moy/Mox$

in which $Bb = B25 + 0.2(Pu/\phi Cc - 0.25)/(0.85 + Cs/Cc)$

for $Pu/\phi \ge 0.25Cc$

or $Bb = B25 + (0.25 - Pu/\phi Cc)^2 (0.85 - Cs/2Cc)$

for $Pu/\phi < 0.25Cc$

where $Cc = f'cCxCy$

$Cs = Asfy$

B25 is the minimum value of Bb which occurs at $Pu/\phi = 0.25Cc$, Mux and Muy are the applied moment strengths of a column about its x-axis and y-axis, respectively, Mox is the resisting moment strength of a column about its x-axis only without bending about the y-axis, and Moy is the resisting moment strength of a column about its y-axis only without bending about the x-axis.

In the region where $Cs/Cc \ge 0.5$:

$B25 = 0.485 + 0.03Cc/Cs$

In the region where $Cs/Cc < 0.5$:

$B25 = 0.545 + 0.35(0.5 - Cs/Cc)$

Having determined which portion of the bilinear interaction curve is to be used, the selected section is verified. Design is adequate when

$[Mux/(Mox)equi] + [Muy/(Moy)equi][(1-Bb)/Bb] \le 1.0$

for $Muy/Mux \le Moy/Mox$

or $[Muy/(Moy)equi] + [Mux/(Mox)equi][(1-Bb)/Bb] \le 1.0$

for $Muy/Mux \ge Moy/Mox$

When the above equation gives a value greater than 1.03, in design mode the trial section needs to be modified, but when it happens in check mode a message is printed and the program proceeds to the next problem.

Parme, et al (25) indicate that this PCA approach yields values sufficiently accurate for design, even for sections that are rectangular or have unequal reinforcement in the two adjacent faces. Gouwens (14) mentions that this procedure is a simple but accurate, conservative and complete method of design.

For the Bresler Load Contour Method, the general equation is

$$[Mux/(Mox)equi]^{\alpha} + [Muy/(Moy)equi]^{\beta} = 1.0$$

in which the values of the exponents α and β are a function of the amount, distribution and location of reinforcement, the dimension of the column, and the strength and elastic properties of the steel and concrete. Wang and Salmon (26) mentions that a value of α that is applicable to the particular column has to be investigated. Bresler (25) indicates that it is reasonably accurate to assume that $\alpha = \beta$. With α set at unity, the equation will always yield conservative results but the use of the equation becomes very conservative for high axial loads or low percentage of reinforcement and should only be used when the applied axial load Pu is less than 0.1(f'c)Ag.

3.8 Modification of Trial Section

When the trial section is not adequate to provide the required strength, the longitudinal reinforcement is incremented first. When the specified number of bars is not entered or the required total number of bars is less than the maximum number of bars for that section, the required total number of bars is incremented by a integer of 2, and then the process of selecting longitudinal reinforcement repeated. If the specified number of bars is entered or the required total number of bars has reached the maximum number of bars, the bar size will be increased providing it has not reached the maximum bar size limit. Should a suitable reinforcement pattern not be found, the concrete dimensions will be increased in accordance with the rules for incrementing concrete section as outlined in Section 3.3.3.

3.9 Evaluation of Longitudinal Splices

Once a trial cross section is obtained that is consistent with the entered restraints and is adequate to satisfactorily carry the loading, this trial section is selected as the final design. The program then proceeds to examine possible splices types and to select the lateral reinforcement.

For tied columns, a check is made to determine whether the reinforcement pattern can be accommodated with normal lap splices and/or tangential lap splices (3). See Fig. 3.9.1 and Fig. 3.9.2. This is accomplished by the following

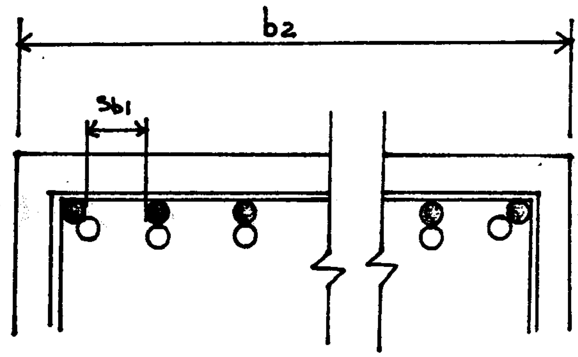


Fig. 3.9.1 Normal lap splices

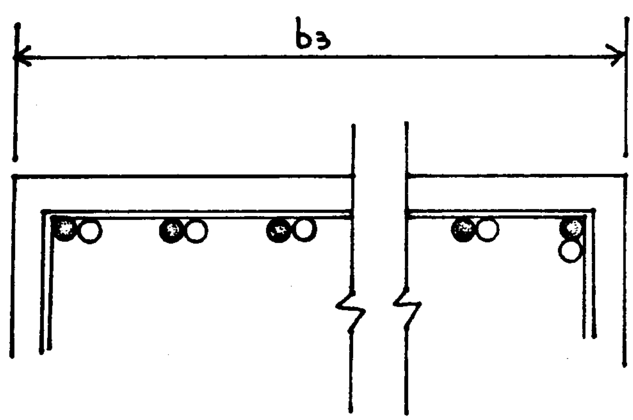


Fig. 3.9.2 Tangential lap splices

expressions :

For bar diameter, db, is less than or equal to #25 bar in *CSA* or #8 bar in *ACI* code,

$$b2=b1+[(2sb_1+2db)\cos\theta-0.586db-2sb_1]$$

where $b1=2(cover+tie\ diam.)+n*db+(n-1)sb_1$

$$\theta=\arcsin[(1-\sqrt{0.5})db]/(sb_1+db)$$

for normal lap splices

$$b3=2(cover+tie\ diam.)+(2n-1)db+(n-1)sb_1$$

for tangential lap splices

For bar diameter, db, is greater than #25 bar in *CSA* or #8 bar in *ACI*,

$$b2=b1+1.38db$$

where $b1=2(cover+tie\ diam.)+n*db+(n-1)1.5db$

for normal lap splices

$$b3=2(cover+tie\ diam.)+(2n-1)db+(n-1)1.5db$$

for tangential lap splices

in which n is number of bars per face, but for the case where bars are in both faces, n is the larger number of bars either in x or y -face, b1,b2 and b3 are the minimum column dimension where bars are located, and sb_1 is the minimum clear bar spacing. If b2 and/or b3 are less than the design column dimension where the reinforcement is located, an appropriate message is printed indicating which type of splices can be accommodated.

For spiral columns, a check is made to determine whether the reinforcement can be accommodated with normal lap splices and/or tangential lap splices (3). This is accomplished by determining the number of bars nn and nt which are the maximum number of bars that can be accommodated using normal and tangential splices, respectively using the following expressions :

When bar diameter, db, is less than or equal to #25 bar in *CSA* or #8 in *ACI* code,

$$nn = \frac{180}{\arcsin[(sb_1 + db)/(h-3db-2(cover+spiral\ diam.))]}$$

$$nt = 180/R$$

where $R = \arcsin[(sb_1 + db)/(h-db-2(cover+spiral\ diam.))] + \arcsin[db/(h-db-2(cover+spiral\ diam.))]$

When bar diameter, db, is greater than #25 in *CSA* or #8 in *ACI* code,

$$nn = \frac{180}{\arcsin[(2.5db)/(h-3db-2(cover+spiral\ diam.))]}$$

$$nt = 180/R1$$

where $R1 = \arcsin[(2.5db)/(h-db-2(cover+spiral\ diam.))] + \arcsin[db/(h-db-2(cover+spiral\ diam.))]$

A message indicating the types of splices that can be accommodated is printed.

Consistent with the building code provisions, the program does not permit the use of lap splices for bar diameters greater than #35 in *CSA* or #11 in *ACI*.

3.10 Selection of Lateral Reinforcement

For tied columns, the required tie spacing per set is the least of:

(a) 16 times the diameter of the smallest longitudinal bars;

(b) 48 times tie diameter;

(c) the least dimension of the compression members.

Lateral ties are required for all corner bars. If the minimum clear bar spacing between adjacent interior bars is greater than a predetermined maximum clear tied bar spacing (default values of 150 mm in *CSA* or 6 in. in *ACI*), lateral ties are required for interior bars.

For spiral columns, the ratio of spiral reinforcement, ρ_s defined as :

$$\rho_s = \frac{\text{Volume of spiral in one loop}}{\text{Volume of core for a length S}}$$

$$= \frac{\pi ds^2\ (Dc-ds)}{Dc^2 S}$$

Also the minimum value of spiral reinforcement to qualify as a spiral column has been determined from tests and is specified in the building codes as

$$\rho_s = 0.45(Ag/Ac-1)(f'c/fsy)$$

in which Dc is the diameter of the core, ds is the lateral spiral diameter, and S is the maximum spacing between spirals. These two equations for ρ_s are used to solve for S.

The minimum number of vertical spacers is also determined consistent with the code requirements.

4. ILLUSTRATIVE EXAMPLES

This chapter is intended to illustrate the use of the program, **COLUMN1**. Detailed mode processing and control commands and data entries are discussed in Appendix A : USER'S MANUAL. Several examples demonstrating most of the features of the program are presented. Input data files and computer standard outputs of sample problems are also presented.

4.1 Example 1 - Short Tied Column (Braced)

Design a square tied column for uniaxial bending about the x-axis with bars distributed along both faces in accordance with the requirement of CSA A23.3-M77. Column is braced with unsupported length of 3000 mm. The material properties for this project are f'c = 30 MPa, fy = 400 MPa and fsy = 400 MPa. All the loadings are unfactored and three load cases are considered.

Load Condition		Axial Force (KN)	Moment at Top (KN-M)	Moment at Bottom (KN-M)
LC1	:Dead Load	1880	120	-120
	Live Load	2720	190	50
LC2	:Dead Load	1880	120	-120
	Live Load	2160	270	70
LC3	:Dead Load	2000	150	150
	Live Load	3000	200	200

This example is executed in batch mode. The computer input is presented in Fig. 4.1.1. The computer output is presented in Fig. 4.1.2 and the selected column section is shown in Fig. 4.1.5.

In the input file when the optional entries are not entered, default values are assigned. For this example, default values for the maximum reinforcement ratio of 0.03 and minimum bar size of #15 are used. The program determines whether column deflects in single or double curvature by examining the signs assigned to the moments. A negative sign to one of the entered moments indicates the column bends in double curvature.

The output consists of an echo check of the essential design parameters including the design load conditions and the required section. Additional design messages are printed. For the section selected based on the first load case, the axial load capacity is only 0.986 times the applied axial load. Although this value is 1.4% low it is within the tolerance of 3% underdesign used by the program. For the additional load cases, the section is not altered but is checked for adequacy. For the second load case, the section is satisfactory. For the last load case, the section is not satisfactory to carry the imposed loadings. This is due in part to the moment carrying bending in single curvature. At this stage, the execution of the program is terminated because the variable **TPROB** is given a value of 6.

In keeping with the philosophy of the program in selecting a satisfactory solution using the minimum concrete section and bar size consistent with the restraints, the design terminated when a satisfactory design using #25 bars was obtained. This will generally be the most economical section. However in some cases the designer may want to see the effect of using a still larger bar which is accomplished be rerunning the problem with a specified minimum bar size of #30.

To illustrate the revision options using the interactive mode, Example 1 is rerun using the interactive mode and only the first load case. The selection of the interactive mode must, of course, be specified at the time of running the program. Input lines are then prompted and entered separately as requested and would essentially be the same as the first 7 lines of Fig. 4.1.1. The solution would be the same as obtained for the batch run and is shown in the first part of Fig. 4.1.4 which is the output using the hard copy option after all revisions have been made and the design terminated. Fig. 4.1.3 illustrates that portion of what is displayed on the screen at the time the revision of the minimum bar size is changed to #30. Note that the revision option permits altering only one line at a time but the entire line containing the revision must be entered. Once the revised line has been entered, the program executes the design using the revised values resulting in the display in the bottom portion of Fig. 4.1.4.

It should be noted that although the format required for an entry line is displayed the entry shown uses the option available under the MTS operating system which uses a comma to designate the end of a field specification. Note also that a blank space is required with no comma after the entry for reinforcement pattern since this is in A2 format.

It is seen that the new design shown in Fig. 4.1.6 has the same concrete dimensions as the previous one but uses 8 #30 bars which is 6.1% less steel for the design load condition. In this example the use of the larger bars would appear more economical but as with the first design the revised section is still not adequate for the last loading case.

```
SHORT TIED COLUMN1 (BRACED)
C T  3 0
   30.    400.    400.       0.            0.
S     0.0    0.0 0.00  0.0   0.0   0.0
B  0.0        0    0    0    0   0 0 0
U     1  1.000      3000.    0  0.000          0.
U     1880.    2720.      120.      190.   -120.      50.
D       0.       0.        0.        0.
U     1880.    2160.      120.      270.   -120.      70.
L       0.       0.        0.        0.
U     2000.    3000.      150.      200.    150.     200.
L       0.       0.        0.        0.
6
```

Fig. 4.1.1 Batch input for example 1

```
 SHORT TIED COLUMN1 (BRACED)                                 PAGE  1
************************************************************

                                00:52:23    JAN 27, 1984

TYPE = TIED              DESIGN CODE = CSA A23.3-M77

       ABOUT X-AXIS : BRACED, EFF. LENGTH =  3000. MM

MATERIAL PROPERTIES :

     CONCRETE    : COMP. STRENGTH   =  30. MPA
                   DENSITY          =  2400. KG/CUBIC METER
                   MODULUS OF ELAS. =  27691. MPA
     LONG. STEEL : YIELD STRESS     =  400. MPA
                   MODULUS OF ELAS. =  200000.MPA
     TIES        : YIELD STRESS     =  400. MPA

DESIGN LOAD CONDITION :

     PU =   7256. KN
     MUX =    491. KN-M,  MUY =      0. KN-M

CONCRETE DIMENSION :

     CX =  650. MM,      CY =  650. MM

LONGITUDINAL REINFORCEMENT :     REINFORCEMENT RATIO =0.014

      8- #25 BARS IN X-FACES
      4- #25 BARS IN Y-FACES

TIES :

     #10 @  403. MM SPACING

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN DOUBLE CURVATURE ABOUT X-AXIS

     FOR SECTION SELECTED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  0.986 TIMES
     THE APPLIED AXIAL LOAD

     TANGENTIAL LAP SPLICES CAN BE USED

     LATERAL TIES REQUIRED FOR INTERIOR BARS
     IN X-FACE

     LATERAL TIES REQUIRED FOR INTERIOR BARS
     IN Y-FACE

ADDITIONAL LOAD CONDITION :

     PU =   6304. KN
```

Fig. 4.1.2 Batch output for example 1 (continued)

```
 SHORT TIED COLUMN1 (BRACED)                                      PAGE  2
************************************************************

     MUX =    627. KN-M,  MUY =      0. KN-M

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN DOUBLE CURVATURE ABOUT X-AXIS

     FOR SECTION ENTERED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  1.009 TIMES
     THE APPLIED AXIAL LOAD

ADDITIONAL LOAD CONDITION :

     PU =   7900. KN
     MUX =    550. KN-M,  MUY =      0. KN-M

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

     FOR SECTION ENTERED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  0.906 TIMES
     THE APPLIED AXIAL LOAD

     ** SECTION IS NOT ADEQUATE FOR THIS LOADING  **
```

Fig. 4.1.2 (continued)

```
NEXT OPTION=? REVISE ABOVE PROBLEM,ENTER 1
              GO TO NEXT PROBLEM,ENTER 2
              TERMINATION OF PROGRAM,ENTER 3

1

REVISE OPTION=? ALTER MATERIAL ONLY,ENTER 1
                ALTER CONCRETE GEOM. ONLY,ENTER 2
                ALTER STEEL GEOM. ONLY,ENTER 3
                ALTER STABILITY GEOM. ONLY,ENTER 4
                ALTER LOADS ONLY,ENTER 5

3

RFACE=? PG=? BMAXX=? NBS=? BSMIN=#? BSMAX=#? DS=#? NX=? NY=?
              ( A2,F7.4,4I4,I3,2I2)
REINF. PATTERN : X FOR X-FACES, Y FOR Y-FACE
                 B FOR BOTH FACES, C FOR CIRCULAR
REINF. RATIO : 0 DEFAULT TO 0.03
MAX. PERMISSIBLE NO. BARS : 0 DEFAULT VALUE BY PROGRAM
TOTAL NO. BARS : (NX, NY REQUIRED IF BARS IN BOTH FACES)
MIN. BAR SIZE : 0 DEFAULT TO 15 OR 5
MAX. BAR SIZE : 0 DEFAULT TO 35 OR 11
MIN. LATERAL REINF. : 0 DEFAULT TO 10 OR 3
NO. BARS IN EACH X-FACE : (NBS IS ENTERED)
NO. BARS IN EACH Y-FACE : (NBS IS ENTERED)

B 0.,0,0,30,
```

Fig. 4.1.3 Interactive input for example 1

```
SHORT TIED COLUMN1 (BRACED)                                          PAGE  1
************************************************************
                                   00:57:03     JAN 27, 1984

TYPE = TIED                DESIGN CODE = CSA A23.3-M77

       ABOUT X-AXIS : BRACED, EFF. LENGTH =  3000. MM

MATERIAL PROPERTIES :

     CONCRETE     : COMP. STRENGTH     =  30. MPA
                    DENSITY            =  2400. KG/CUBIC METER
                    MODULUS OF ELAS.   =  27691. MPA
     LONG. STEEL  : YIELD STRESS       =  400. MPA
                    MODULUS OF ELAS.   =  200000.MPA
     TIES         : YIELD STRESS       =  400. MPA

DESIGN LOAD CONDITION :

     PU =    7256. KN
     MUX =      491. KN-M,   MUY =        0. KN-M

CONCRETE DIMENSION :

     CX =  650. MM,       CY =  650. MM

LONGITUDINAL REINFORCEMENT :      REINFORCEMENT RATIO =0.014

      8- #25 BARS IN X-FACES
      4- #25 BARS IN Y-FACES

TIES :

     #10 @  403. MM SPACING

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN DOUBLE CURVATURE ABOUT X-AXIS

     FOR SECTION SELECTED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  0.986 TIMES
     THE APPLIED AXIAL LOAD

     TANGENTIAL LAP SPLICES CAN BE USED

     LATERAL TIES REQUIRED FOR INTERIOR BARS
     IN X-FACE

     LATERAL TIES REQUIRED FOR INTERIOR BARS
     IN Y-FACE

REVISE OPTION = ALTER STEEL OF GEOMETRY PROPERTIES ONLY

CONCRETE DIMENSION :
```

Fig. 4.1.4 Interactive output for example 1 (continued)

```
SHORT TIED COLUMN1 (BRACED)                                     PAGE  2
************************************************************

    CX =  650. MM,      CY =  650. MM

LONGITUDINAL REINFORCEMENT :     REINFORCEMENT RATIO =0.013

     6- #30 BARS IN X-FACES
     2- #30 BARS IN Y-FACES

TIES :

    #10 @  478. MM SPACING

ADDITIONAL COMMENTS :

    COLUMN DEFLECTS IN DOUBLE CURVATURE ABOUT X-AXIS

    FOR SECTION SELECTED THE AXIAL LOAD CAPACITY
    UNDER APPLIED MOMENT IS  0.975 TIMES
    THE APPLIED AXIAL LOAD

    TANGENTIAL LAP SPLICES CAN BE USED
```

Fig. 4.1.4 (continued)

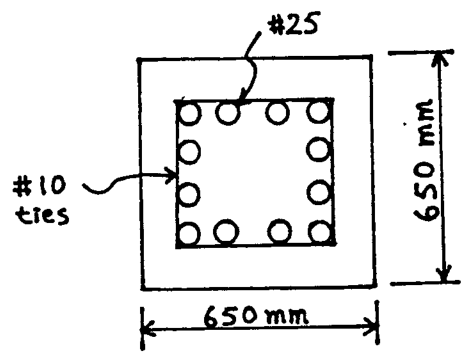


Fig. 4.1.5 Column section for example 1

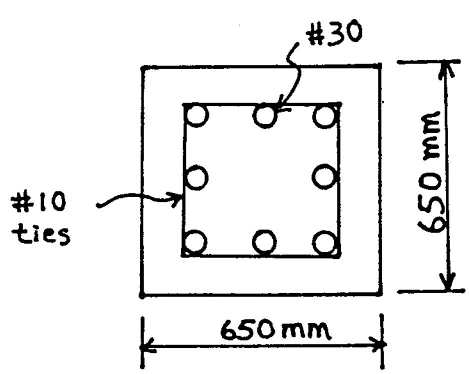


Fig. 4.1.6 Column section for example 1 specifying #30 bars

4.2 Example 2 - Spiral Column (CSA)

Design a circular spiral column for uniaxial bending about the x-axis in accordance with the requirement of CSA A23.3-M77. Column is braced with the unsupported length of 3000 mm. The material properties for this project are f'c = 30 MPa, fy = 350 MPa and fsy = 300 MPa. All the loadings are factored. This column is subjected to an axial dead and live load of 3520 KN and 5280 KN respectively, with a dead load and a live load moment at both the top and bottom of column of 112 KN-M and 168 KN-M respectively.

This example is executed in batch mode. The computer input is presented in Fig. 4.2.1. The computer output is presented in Fig. 4.2.2 and the selected column section is shown in Fig. 4.2.3.

The computer input and output are similar to that in example 1. In this example, the actual computed eccentricity is less than the specified minimum, therefore the minimum moment specified in the building codes is used. The axial load capacity is only 0.974 times the applied axial load which while 2.6% low is within the tolerance of 3% underdesign used by the program.

```
SPIRAL COLUMN (CSA)
C S  1 0
   30.    350.    300.       0.             0.
C    0.0    0.0 0.00  0.0  0.0  0.0
C  0.0        0   0    0    0  0
U    1  1.000     3000.   0  0.000          0.
F    3520.   5280.     112.     168.     112.     168.
D       0.       0.       0.       0.
6
```

Fig. 4.2.1 Input for example 2

```
 SPIRAL COLUMN (CSA)                                                    PAGE  1
************************************************************
                                        22:19:04     FEB  1, 1984

TYPE = SPIRAL                   DESIGN CODE = CSA A23.3-M77

       ABOUT X-AXIS : BRACED, EFF. LENGTH =  3000. MM

MATERIAL PROPERTIES :

     CONCRETE    : COMP. STRENGTH    =  30. MPA
                   DENSITY           =  2400. KG/CUBIC METER
                   MODULUS OF ELAS.  =  27691. MPA
     LONG. STEEL : YIELD STRESS      =  350. MPA
                   MODULUS OF ELAS.  =  200000.MPA
     SPIRAL      : YIELD STRESS      =  300. MPA

DESIGN LOAD CONDITION :

     PU =   8800. KN
     MUX =    280. KN-M,   MUY =       0. KN-M

CONCRETE DIMENSION :

     DIAMETER =   700. MM

LONGITUDINAL REINFORCEMENT :     REINFORCEMENT RATIO =0.026

     14- #30 BARS IN CIRCULAR

SPIRALS :

     #10 WITH   50. MM PITCH AND 3 SPACERS

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

     MINIMUM ECCENTRICITY ABOUT X-AXIS GOVERNS
     DESIGN MOMENT ABOUT X-AXIS =      308.
     IN PROJECT UNITS

     FOR SECTION SELECTED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  0.974 TIMES
     THE APPLIED AXIAL LOAD
```

Fig. 4.2.2 Output for example 2

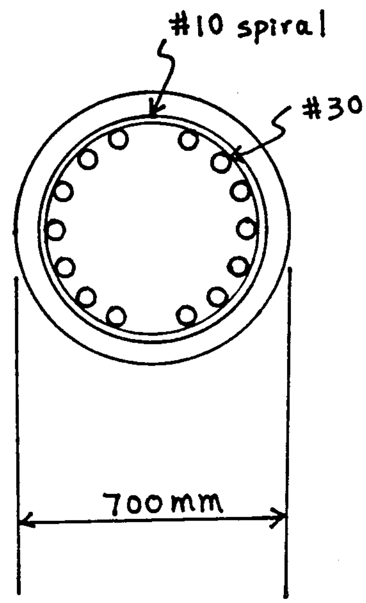


Fig. 4.2.3 Column section for example 2

4.3 Example 3 - Slender Braced Column (ACI)

Design a rectangular tied column for uniaxial bending about the x-axis with bars distributed along both faces in accordance with the requirements of ACI 318-77. Column is braced with the unsupported length of 330 in. The material properties for this project are f'c = 4000 psi, fy = 60000 psi and fsy = 60000 psi. Select the concrete dimension ratio of 0.9, the minimum and the maximum bar size of #9 and #18 respectively. All the loadings are factored. This column is subjected to an axial dead and live load of 224 kips and 336 kips respectively, a dead load moment at both the top and bottom of column of 131 kip-ft, and a live load moment at the top and bottom of column of 115 kip-ft and 196 kip-ft respectively.

This example is executed in batch mode. The computer input is presented in Fig. 4.3.1. The computer output is presented in Fig. 4.3.2 and the selected column section is shown in Fig. 4.3.3.

The computer input and output are similar to that in example 1. In this example, the column was found to be "slender" and the moment magnification factor was computed as 1.41. The section selected is adequate for this loading and is 5.3% overdesigned.

The data for this example was taken from "COLUMN EXAMPLE 2" in 'Design Handbook In Accordance With the Strength Design Method of ACI 318-77 : Volumn 2 - Columns' (3). In "COLUMN EXAMPLE 2", the moment magnification factor was found to be 1.44 and 12 #10 bars were used due to the restriction placed on the interaction diagrams used in the example which require reinforcement along all faces to have equal areas in each face. This restriction does not exist in **COLUMN1** so that different areas of steel in the different faces can be considered resulting in only 10 #10 bars being selected. Except for the reinforcement, **COLUMN1** gives corresponding results to those in "DESIGN EXAMPLE 2".

```
SLENDER BRACED COLUMN (ACI)
A T  1 0
 4000. 60000. 60000.      0.            0.
R     0.0    0.0 0.9   0.0  0.0  0.0
B  0.0        0    0    9  18
U     1  1.000        330.   0  0.000          0.
F      224.     336.     131.     115.     131.     196.
D        0.       0.       0.       0.
6
```

Fig. 4.3.1 Input for example 3

```
 SLENDER BRACED COLUMN (ACI)                                       PAGE  1
*************************************************************

                                        00:49:44     JAN 27, 1984

TYPE = TIED                      DESIGN CODE = ACI 318-77

       ABOUT X-AXIS : BRACED, EFF. LENGTH =    330. IN.

MATERIAL PROPERTIES :

     CONCRETE      : COMP. STRENGTH    =  4000. PSI
                     DENSITY           =  145. PCF
                     MODULUS OF ELAS. =    3644149. PSI
     LONG. STEEL : YIELD STRESS       =  60000. PSI
                     MODULUS OF ELAS. =  29000000.PSI
     TIES          : YIELD STRESS     =  60000. PSI

DESIGN LOAD CONDITION :

     PU   =      560. KIPS
     MUX =      327. KIP-FT,  MUY =       0. KIP-FT

CONCRETE DIMENSION :

     CX =  20.0 IN.,       CY =  22.0 IN.

LONGITUDINAL REINFORCEMENT :      REINFORCEMENT RATIO =0.029

      6- #10 BARS IN X-FACES
      4- #10 BARS IN Y-FACES

TIES :

     # 3 @ 18.0 IN. SPACING

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

     MAGNIFICATION FACTOR FOR MOMENT ABOUT X-AXIS = 1.41

     FOR SECTION SELECTED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  1.053 TIMES
     THE APPLIED AXIAL LOAD

     TANGENTIAL LAP SPLICES CAN BE USED

     LATERAL TIES REQUIRED FOR INTERIOR BARS
     IN Y-FACE
```

Fig. 4.3.2 Output for example 3

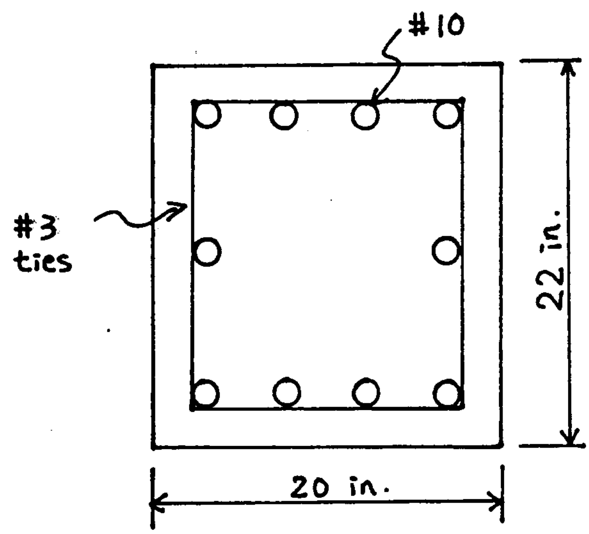


Fig. 4.3.3 Column section for example 3

4.4 Example 4 - Square Tied Column in Biaxial Bending

Design a 20 x 20 in. square tied column for biaxial bending about the x-axis and the y-axis, respectively with bars distributed along both faces in accordance with the requirement of ACI 318-77. Column is braced with the unsupported length of 100 in.. The material properties for this project are f'c = 6000 psi, fy = 60000 psi and fsy = 60000 psi. Select the maximum value for reinforcement ratio of 0.04 ,and specify the bar size of #10. All the loadings are factored and three load cases are considered.

Load Condition		Axial Force (KIPS)	Moment at Top x-axis (KIP-FT)	Moment at Top y-axis (KIP-FT)	Moment at Bottom x-axis (KIP-FT)	Moment at Bottom y-axis (KIP-FT)
LC1	:Dead Load	100	38	175	38	175
	Live Load	100	38	175	38	175
LC2	:Dead Load	56	49	21	49	21
	Live Load	84	70	32	70	32
LC3	:Dead Load	1000	400	0	400	0
	Live Load	0	0	0	0	0

This example is executed in batch mode. The computer input is presented in Fig. 4.4.1. The computer output is presented in Fig. 4.4.2 and the selected column section is shown in Fig. 4.4.3.

The dead and live load for both axial and moments loadings must be entered individually whether the loadings are factored or unfactored since in considering the slenderness effects, the maximum design dead load moment is taken into account.

The computer input and output are similar to that in example 1. For the first and second load cases, the PCA Load

Contour Method is used to check the selected section for adequacy in strength because the value of Pu is less than 0.1(f'c)Ag, and the section is found to be satisfactory to carry the imposed loadings. For the last load case, the minimum moment specified in the building codes governs, and the section is not satisfactory to carry the imposed loadings based on the axial load capacity is 0.953 times the applied axial load.

```
SQUARE TIED COLUMN IN BIAXIAL BENDING
A T  3 0
 6000. 60000. 60000.      0.           0.
S   20.0   0.0 0.0   0.0  0.0  0.0
B  0.04     0    0  10  10  0 0 0
B     1  1.000       100.   1  1.000      100.
F      100.     100.     38.      38.      38.      38.
D      175.     175.    175.     175.
F       56.      84.     49.      70.      49.      70.
L       21.      32.     21.      32.
F     1000.       0.    400.       0.     400.       0.
L        0.       0.      0.       0.
6
```

Fig. 4.4.1 Input for example 4

```
SQUARE TIED COLUMN IN BIAXIAL BENDING                              PAGE  1
*************************************************************

                                        22:21:15     FEB  1, 1984

TYPE = TIED                  DESIGN CODE = ACI 318-77

       ABOUT X-AXIS : BRACED, EFF. LENGTH =   100. IN.

       ABOUT Y-AXIS : BRACED, EFF. LENGTH =   100. IN.

MATERIAL PROPERTIES :

     CONCRETE     : COMP. STRENGTH    =  6000. PSI
                    DENSITY           =  145. PCF
                    MODULUS OF ELAS.  =   4463153. PSI
     LONG. STEEL  : YIELD STRESS      =  60000. PSI
                    MODULUS OF ELAS.  =  29000000.PSI
     TIES         : YIELD STRESS      =  60000. PSI

DESIGN LOAD CONDITION :

     PU  =     200. KIPS
     MUX =      76. KIP-FT,  MUY =    350. KIP-FT

CONCRETE DIMENSION :

     CX =  20.0 IN.,      CY =  20.0 IN.

LONGITUDINAL REINFORCEMENT :     REINFORCEMENT RATIO =0.032

      8- #10 BARS IN X-FACES
      2- #10 BARS IN Y-FACES

TIES :

     # 3 @ 18.0 IN. SPACING

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT Y-AXIS

     EQUIVALENT REQUIRED MOMENT ABOUT Y-AXIS
     ONLY =    446. IN PROJECT UNITS

     SELECTED SECTION IS ADEQUATE BASED ON
     MOMENT CAPACITIES

     TANGENTIAL LAP SPLICES CAN BE USED

     LATERAL TIES REQUIRED FOR INTERIOR BARS
     IN X-FACE
```

Fig. 4.4.2 Output for example 4 (continued)

```
****************************************************************

ADDITIONAL LOAD CONDITION :

     PU  =     140. KIPS
     MUX =     119. KIP-FT,  MUY =      53. KIP-FT

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT Y-AXIS

     EQUIVALENT REQUIRED MOMENT ABOUT X-AXIS
     ONLY =     183. IN PROJECT UNITS

     ENTERED SECTION IS ADEQUATE BASED ON
     MOMENT CAPACITIES

ADDITIONAL LOAD CONDITION :

     PU  =    1000. KIPS
     MUX =     400. KIP-FT,  MUY =       0. KIP-FT

ADDITIONAL COMMENTS :

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

     COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT Y-AXIS

     MINIMUM ECCENTRICITY ABOUT Y-AXIS GOVERNS
     DESIGN MOMENT ABOUT Y-AXIS =      100.
     IN PROJECT UNITS

     FOR SECTION ENTERED THE AXIAL LOAD CAPACITY
     UNDER APPLIED MOMENT IS  0.953 TIMES
     THE APPLIED AXIAL LOAD

     ** SECTION IS NOT ADEQUATE FOR THIS LOADING **
```

Fig. 4.4.2 (continued)

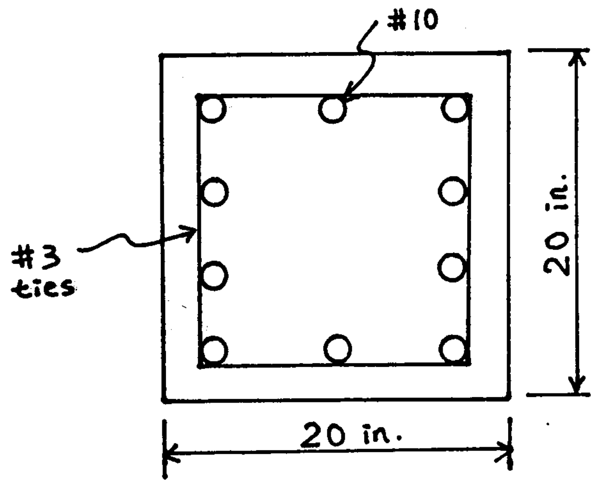


Fig. 4.4.3 Column section for example 4

4.5 Example 5 - Rect. Tied Column Check

Check a 15 x 24 in. rectangular tied column for uniaxial bending about the x-axis with 6 #8 bars distributed along the y-faces only in accordance with the requirement of ACI 318-77. Column is braced with the unsupported length of 100 in.. The material properties for this project are f'c = 3000 psi, fy = 50000 psi and fsy = 50000 psi. All the loadings are factored.

Load Condition		Axial Force (KIPS)	Moment at Top (KIP-FT)	Moment at Bottom (KIP-FT)
LC1	:Dead Load	400	266	266
	Live Load	0	0	0
LC2	:Dead Load	460	300	300
	Live Load	0	0	0

This example is executed in batch mode. The computer input and output are presented in Fig. 4.5.1 and Fig. 4.5.2 respectively.

The computer input and output are similar to that in example 1. In this example, the section is not altered but is checked for adequacy. For the first load case, the entered section is satisfactory. For the second load case, the entered section is not satisfactory to carry the imposed loadings.

```
RECT. TIED COLUMN CHECK
A T  2
 3000. 50000. 50000.
R   15.0  24.0
Y  0.0       0   6    8    8
U     1  1.000       100.
F      400.       0.     266.       0.     266.       0.
C
F      460.       0.     300.       0.     300.       0.
L
6
```

Fig. 4.5.1 Input for example 5

```
RECT. TIED COLUMN CHECK                                          PAGE  1
**********************************************************

                               21:49:21     FEB  2, 1984

TYPE = TIED                  DESIGN CODE = ACI 318-77

     ABOUT X-AXIS : BRACED, EFF. LENGTH =   100. IN.

MATERIAL PROPERTIES :

    CONCRETE     : COMP. STRENGTH    =  3000. PSI
                   DENSITY           =  145. PCF
                   MODULUS OF ELAS.  =   3155926. PSI
    LONG. STEEL  : YIELD STRESS      =  50000. PSI
                   MODULUS OF ELAS.  =  29000000.PSI
    TIES         : YIELD STRESS      =  50000. PSI

CHECK LOAD CONDITION :

    PU  =    400. KIPS
    MUX =    266. KIP-FT,  MUY =      0. KIP-FT

CONCRETE DIMENSION :

    CX =  15.0 IN.,      CY =  24.0 IN.

LONGITUDINAL REINFORCEMENT :     REINFORCEMENT RATIO =0.013

     6- # 8 BARS IN Y-FACES ONLY

ADDITIONAL COMMENTS :

    COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

    FOR SECTION ENTERED THE AXIAL LOAD CAPACITY
    UNDER APPLIED MOMENT IS  0.985 TIMES
    THE APPLIED AXIAL LOAD

ADDITIONAL LOAD CONDITION :

    PU  =    460. KIPS
    MUX =    300. KIP-FT,  MUY =      0. KIP-FT

ADDITIONAL COMMENTS :

    COLUMN DEFLECTS IN SINGLE CURVATURE ABOUT X-AXIS

    FOR SECTION ENTERED THE AXIAL LOAD CAPACITY
    UNDER APPLIED MOMENT IS  0.865 TIMES
    THE APPLIED AXIAL LOAD

    ** SECTION IS NOT ADEQUATE FOR THIS LOADING **
```

Fig. 4.5.2 Output for example 5

5. SUMMARY AND CONCLUSIONS

The reinforced concrete column design computer program, COLUMN1, run either in batch mode or in interactive mode has been developed to provide an opportunity for designers with little experience such as students to obtain more quickly a feel for reinforced concrete column design. The program is capable of performing design or analysis of most columns that will be encountered in practice in accordance with either the CSA3-A23.3-M77 or the ACI 318-77 building codes and so may be used in design offices or incorporated in a CAD system. A single column design or multiple column designs can be solved during a single run of the program. In interactive mode, a hard copy output may be obtained at the end of each run.

In design mode, the program proceeds to find the minimum feasible concrete section that may be used for the specified loads, moments, and material properties using the least possible reinforcement consistent with input constraints. This trial section is checked for slenderness effects, bar spacing and conformance to building code requirements. The user has the option of specifying all or any portion of the final cross-section. When the complete cross-section is entered (check mode) a message indicating the capacity of the column section is printed including a warning if not adequate.

A number of design examples are illustrated together with the computer outputs from the program COLUMN1. Results compare favorably with published concrete column designs obtained using traditional methods.

REFERENCES

1. ACI Code, "Building Code Requirements for Reinforced Concrete (ACI-318-77)", American Concrete Institute, Detroit, 1977.
2. ACI Commentary, "Commentary on Building Code Requirements for Reinforced Concrete (ACI-318-77)", American Concrete Institute, Detroit, 1977.
3. ACI Committee 340, "Design Handbook In Accordance With the Strength Design Method of ACI 318-77 : Volume 2 - Columns", ACI Special Publication, SP 17A(78), American Concrete Institute, Detroit, 1978, p. 214.
4. Bock, A.H., "Slenderness Effects : Notes on ACI 318-77 Building Code Requirements for Reinforced Concrete with Design Applications", Third Edition, Portland Cement Association, Illinois, 1980, part 12.
5. Christenser, K., "Writing Easy-To-Use Computer Program", Engineering Digest, Vol. 29, Number 5, May, 1983, p. 40.
6. Clarke, D., "Computer Aided Structural Design", A Wiley-Interscience Publication, Chichester, 1978.
7. "COLS - Strength Design of Reinforced Concrete Column Section", Canadian Portland Cement Association, Ottawa.
8. CSA Code, "Code for the Design of Concrete Structures for Buildings (CAN3-A23.3-M77)", Canadian Standards Association, Ontario, 1977.

9. CSA Code, "Concrete Materials and Methods of Concrete Construction, Method of Test for Concrete (CSA3-A23.1-M77, CSA3-A23.2-M77)", Canadian Standards Association, Ontario, 1977.

10. Dobbs, J.E., "Design for Flexure and Axial Load : Notes on ACI 318-77 Building Code Requirements for Reinforced Concrete with Design Applications", Third Edition, Portland Cement Association, Illinois, 1980, part 10.

11. Engineering Bulletin, "Strength Design of Reinforced Concrete Columns", EB009.02D, Portland Cement Association, Illinois, 1978.

12. Furlong, R.W., "Design for Flexure and Axial Load : Metric Design Handbook for Reinforced Concrete Elements In Accordance With the Strength Design Methods of CSA Standard CAN3-A23.3-M77", Canadian Portland Cement Association, Ottawa.

13. Ghosh, S.K., "Design Methods and Strength Requirements : Notes on ACI 318-77 Building Code Requirements for Reinforced Concrete with Design Applications", Third Edition, Portland Cement Association, Illinois, 1980, part 4.

14. Gouwens, A.J., "Biaxial Bending Simplified", Reinforced Concrete Columns, Publication SP-50, American Concrete Institute, Detroit, 1975, p. 233.

15. Halvorsen, G.T., "Reinforced Concrete Column Design Program", ACI Seminar, Kansas City, Fall, 1983.

16. MacGregor, J.G. and Breen, J.E. and Pfrang, E.O., "Design of Slender Concrete Columns", ACI Journal, Vol. 67, No. 1, January, 1970, p. 6.

17. MacGregor, J.G., Olehafen, U.H. and Hage, S.E., "A Re-examination of the EI Value for Slender Columns", Reinforced Concrete Columns, Publication SP-50, American Concrete Institute, Detroit, 1975, p. 1.

18. Mockry, E.F. and Darwin, D., "Slender Column Interaction Diagrams", Concrete International, June, 1982, p. 44.

19. Murrill, P.W. and Smith, C.L., "Fortran IV Programming for Engineers and Scientists", Intext Educational Publishers, New York, 1973.

20. Randall, R.W., "Design for Flexure : Notes on ACI 318-77 Building Code Requirements for Reinforced Concrete with Design Applications", Third Edition, Portland Cement Association, Illinois, 1980, part 9.

21. Rice, P.F., "Design for Reinforcement : Metric Design Handbook for Reinforced Concrete Elements In Accordance With the Strength Design Methods of CSA Standard CAN3-A23.3-M77", Canadian Portland Cement Association, Ottawa.

22. Rice, P.F. and Hoffman, E.S., "Structural Design Guide to the ACI Building Code", Second Edition, Van Nostrand Reinhold Company, New York, 1979, p. 276.

23. Rogers, P. and Baltay, M.L., "Reinforced Concrete Design for Buildings", Van Nostrand Reinhold Company, New York, 1973, p. 75.

24. Saatcioglu, M., "General Design Information : Metric Design Handbook for Reinforced Concrete Elements In Accordance With the Strength Design Methods of CSA Standard CAN3-A23.3-M77", Canadian Portland Cement Association, Ottawa.

25. Schultz, D.M., "Biaxial Loading : Notes on ACI 318-77 Building Code Requirements for Reinforced Concrete with Design Applications", Third Edition, Portland Cement Association, Illinois, 1980, part 11.

26. Wang, C.K. and Salmon, C.G., "Reinforced Concrete Design", Third Edition, Harper & Row Publishers, New York, 1979, p. 388.

APPENDIX A

USER'S MANUAL

A.1 Description

(a) *Operational Modes*

(i) Batch Mode

(ii) Interactive Mode

The first entry for each run of program *COLUMN1*, is either 0 for batch mode processing or 1 for interactive mode processing. The mode selected applies to all designs to be considered in that run. For interactive mode processing ,the return key or similar key must be pressed after a line of required data information is entered on the terminal keyboard.

(b) *Code Specifications*

(i) Canadian Standards Association Specification CAN3-A23.3-M77

(ii) American Concrete Institute Specification ACI 318-77

(c) *Units*

(i) S.I. units when using *CSA* code

(ii) Imperial or U.S. Customary units when using *ACI* code.

A.2 Control Commands

For use on terminal (MTS).

For batch mode processing,

$RUN COLUMN1 5=*DATAFILE* 4=*SOURCE* 7=*SINK*

6=*UNSORT* 8=*MESS* 9=*SORT*

For interactive mode processing,

$RUN COLUMN1 4=*SOURCE* 7=*SINK*

6=*UNSORT* 8=*MESS* 9=*SORT*

Details on I/O units:

Unit 5=DATAFILE refers to input data records stored in the file named DATAFILE.

Unit 4=*SOURCE* refers to the input from the terminal.

Unit 6=UNSORT refers to the unsorted output for the line printer.

Unit 9=SORT refers to the sorted output for the line printer.

Unit 8=MESS refers to the temporary storage of comments.

Unit 7=*SINK* refers to the output for the terminal.

If Device I/O units are not specified, an error message will be printed from MTS terminal informing the user that the specific device units are undefined.

A.3 Data Input

A description of data input is listed below. The second line of each description is a symbolic representation of the input line indicating the order of the entries.

Input is formatted and each of the entries in the input line is entered in according to its format field specification. When the program is executed using the

Michigan Terminal System (MTS) at the University of Alberta, free format input is permitted only for **F** format and **I** format, and individual entries are separated by a comma. All entries in an input line are required but all zero entries occurring at the end of a line may be omitted.

Input is the same for both batch and interactive mode but in interactive mode prompts are displayed for inputs.

(a) *Operational Modes :*

Mode FORMAT (I1)

Mode = 0 batch mode

= 1 interactive mode

(b) *Heading Record :*

HEADING FORMAT (15A4)

HEADING = Any user defined descriptive identifier not exceeding 60 characters

(c) *Input Control Design Record:*

(1) **CODE TYPE NLC OUT** FORMAT (2A2, 2I2)

CODE = C specifies CAN3-A23.3-M77

= A specifies ACI 318-77

TYPE = T indicates tied column

= S indicates spiral column

NLC = total number of load cases
If NLC not entered, defaults to 1.
In interactive mode, an entry for NLC is not required because the user has the option of later entering as many load cases as desired.

OUT = 0 standard output for design practice
= 1 long output for debugging purposes

(d) *Data Material Properties:*

(1) **FPC FY FSY WC ES** FORMAT (4F7.0, F12.0)

FPC = concrete strength (MPa or psi).
If value entered is outside range 20-40 MPa or 2000-6000 psi, design is executed but warning message is printed.

FY = yield stress for longitudinal reinforcement (MPa or psi). If fy is outside range of 300-400 MPa or 40000-60000 psi, design is executed but warning message is printed.

FSY = yield stress for spiral steel (MPa or psi). If fsy is outside range of 300-400 MPa or 40000-60000 psi, deisgn is executed but warning message is printed.

WC = mass density concrete (kg/m^3 or pcf).
If not entered default to 2400 kg/m^3 or 145 pcf.

ES = Modulus of elasticity for reinforcement (MPa or psi). If not entered default to 200000 MPa or 29000000 psi.

(e) *Data Geometry Properties:*

(1) Concrete geometry:

SHAPE Cx Cy R DC CXINCR CYINCR

FORMAT (A2, 2F6.1, F5.2, 3F5.1)

SHAPE = S indicates square cross-section

= R indicates rectangular cross-section

= C indicates circular cross-section

Cx = dimension of cross-section in x direction (mm or in)

Cy = dimension of cross-section in y direction (mm or in). If shape = S or R, value of Cy not used.

R = ratio of Cx/Cy. This value is used with either Cx or Cy, but when both Cx and Cy are entered, R is not used. When shape is not rectangular, R defaults to 1.0.

DC = minimum clear concrete cover (mm or in). If not entered, default value is 40 mm or 1.5 in..

CXINCR = length increments of dimension Cx value (mm or in). If not entered, default value is 50 mm or 2 in..

CYINCR = length increments of dimension Cy value (mm or in). If not entered, default value is 50 mm or 2 in..

(2) Reinforcement geometry:

RFACE PG BMAXX NBS BSMIN BSMAX DS NX NY

FORMAT (A2, F7.4, 4I4, I3, 2I2)

RFACE = X reinforcement configuration in x-face (lateral faces) only (Fig. 3.2.2).

= Y reinforcement configuration in y-face (end faces) only (Fig. 3.2.3).

= B reinforcement configuration in both x and y -faces (bars in x-faces include all corner bars) (Fig. 3.2.4).

= C means reinforcement configuration around the core diameter (Fig. 3.6.5).

PG = maximum percentage of reinforcement

If PG not entered default to 0.03.

If PG < 0.01 value set to 0.01.

If PG > 0.08 value set to 0.08.

BMAXX = entered maximum permissible number of bars in cross-section. If not entered default to the value given in Table 3.1 in Section 3.5 .

NBS = specified total number of bars

If NBS is entered and bars are in

both faces, NX and NY must be entered.

BSMIN = minimum bar size, 15 to 55 for
CSA or 5 to 18 for ACI.
If not entered default to 15 or 5.

BSMAX = maximum bar size, 15 to 55 for
CSA or 5 to 18 for ACI.
If not entered default to 35 or 11.

DS = minimum lateral reinforcement diameter,
i.e. bar size number is entered.
If not entered default value are
10 for CSA or 3 for ACI

NX = number of bars in each x-face
including corner bars

NY = number of bars in each y-face

(3) Stability information:

BENDING XBRACING KX LUX YBRACING KY LUY SPU SPC

FORMAT (A2, I4, F7.3, F10.0, I4, F7.3, F10.0, 2F8.0)

BENDING = U uni-axial bending
= B bi-axial bending

XBRACING = 0 column unbraced about x-axis
= 1 column braced about x-axis

KX = effective length factor about x-axis

LUX = unsupported length about x-axis (mm or in)

YBRACING = 0 column unbraced about y-axis

= 1 column braced about y-axis

KY = effective length factor about y-axis

LUY = unsupported length about y-axis (mm or in)

SPU = summation of axial design load for all of the columns in the storey (KN or kips)

SPC = summation of critical load for all of the columns in the storey (KN or kips)

(f)*Data Loads Conditions:*

(1) **FACT PD PL MDXT MLXT MDXB MLXB**

FORMAT (A2, 6F8.0)

FACT = U loading entered is unfactored

= F loading entered is factored

PD = axial dead load (must be positive since only compression loads permitted) (KN or kips)

PL = axial live load (must be positive) (KN or kips)

MDXT = dead load moment about x-axis on top of column (KN-m or kip-ft)

MLXT = live load moment about x-axis on top of column (KN-m or kip-ft)

MDXB = dead load moment about x-axis on bottom of column (KN-m or kip-ft)

MLXB = live load moment about x-axis on bottom of column (KN-m or kip-ft)

Note : The applied moment about x-axis is primarily resisted by the reinforcement in the y-faces.

Note : A negative sign to one of the entered moments at the top or bottom of column indicates the column deflects in double curvature. When the signs are the same, column deflects in single curvature. Design moment is the maximum absolute value of moments entered at either top or bottom of the column. Same philosophy applies for moments about the y-axis.

(2) **DESIGN MDYT MLYT MDYB MLYB**

FORMAT (A2, 4F8.0)

DESIGN = D designates a design, that is program will select all missing values of cross-section to satisfy loading

= L designates a check for adequacy in strength of selected section for multiple loading cases

= C designates a check of entered section

MDYT = dead load moment about y-axis on top of column (KN-m or kip-ft)

MLYT = live load moment about y-axis on top of column (KN-m or kip-ft)

MDYB = dead load moment about y-axis on bottom of column (KN-m or kip-ft)

MLYB = live load moment about y-axis on bottom of column (KN-m or kip-ft)

Note : The applied moment about y-axis is primarily resisted by the reinforcement in the x-faces.

(g) *Optional Data:*

(1) **OPTION**

In batch mode the option number indicates the line type at which new data will be entered. All following data lines must be repeated. A new heading may also be used. For example if a new design was to be run using same code and material properties but different concrete geometry requirements option 2 would be selected. A new heading and all data beginning with concrete geometry requirements through to new option number are repeated, but data usually entered before the concrete geometry requirement line is omitted and will default to values used for previous problem. Note, in interactive mode only one line is re-entered.

(i) in batch mode: FORMAT (I2)

OPTION = 1 means a different column design, all data records are to be re-entered.

= 2 means heading and new data entries beginning with concrete geometry.

= 3 means heading and new data entries beginning with reinforcement geometry.

= 4 means heading and new data entries beginning with stability geometry.

= 5 means heading and new data entries beginning with load conditions.

= 6 means last column design has been solved, program is terminated.

(ii) in interactive mode: FORMAT (I1)

Revised above design, enter 1

Go to next design, enter 2

Termination of program, enter 3

If 2 is entered, the option is the same as that in batch, and the next data records required are prompted by the program.

If 1 is entered, the next data record is specified as follows:

REVISE FORMAT (I1)

Alter material properties only, enter 1

Alter concrete geometry only, enter 2

Alter reinforcement geometry only, enter 3

Alter stability geometry only, enter 4

Alter loading conditions only, enter 5

APPENDIX B

LOGIC FLOW OF COLUMN1

A general flow chart showing the overall connectivity of the subroutines is given in Fig. B.1. Reference is made to specific subroutines in the program, **COLUMN1**, given in Appendix C with general comments as to the nature of the design steps being performed in that subroutine.

The sequence of events coded into the main executive program *COLUMN1* is as follows (Fig. B.1).

(1) Enter the mode processing parameters into the terminal.

(2) Set TPROB = 1 as initial value to indicate a new design with no previous data entered.

(3) Read the heading.

(4) If TPROB = 1 go to (5).

If TPROB ≠ 1 and

concrete geometry properties are to be varied, go to (9).

reinforcement geometry properties are to be varied, go to (9).

stability geometry properties are to be varied, go to (9).

loading geometry properties are to be varied, go to (11).

(5) Read the design control parameters (Sub: *INPUT1*). (If in batch mode, go to (7).)

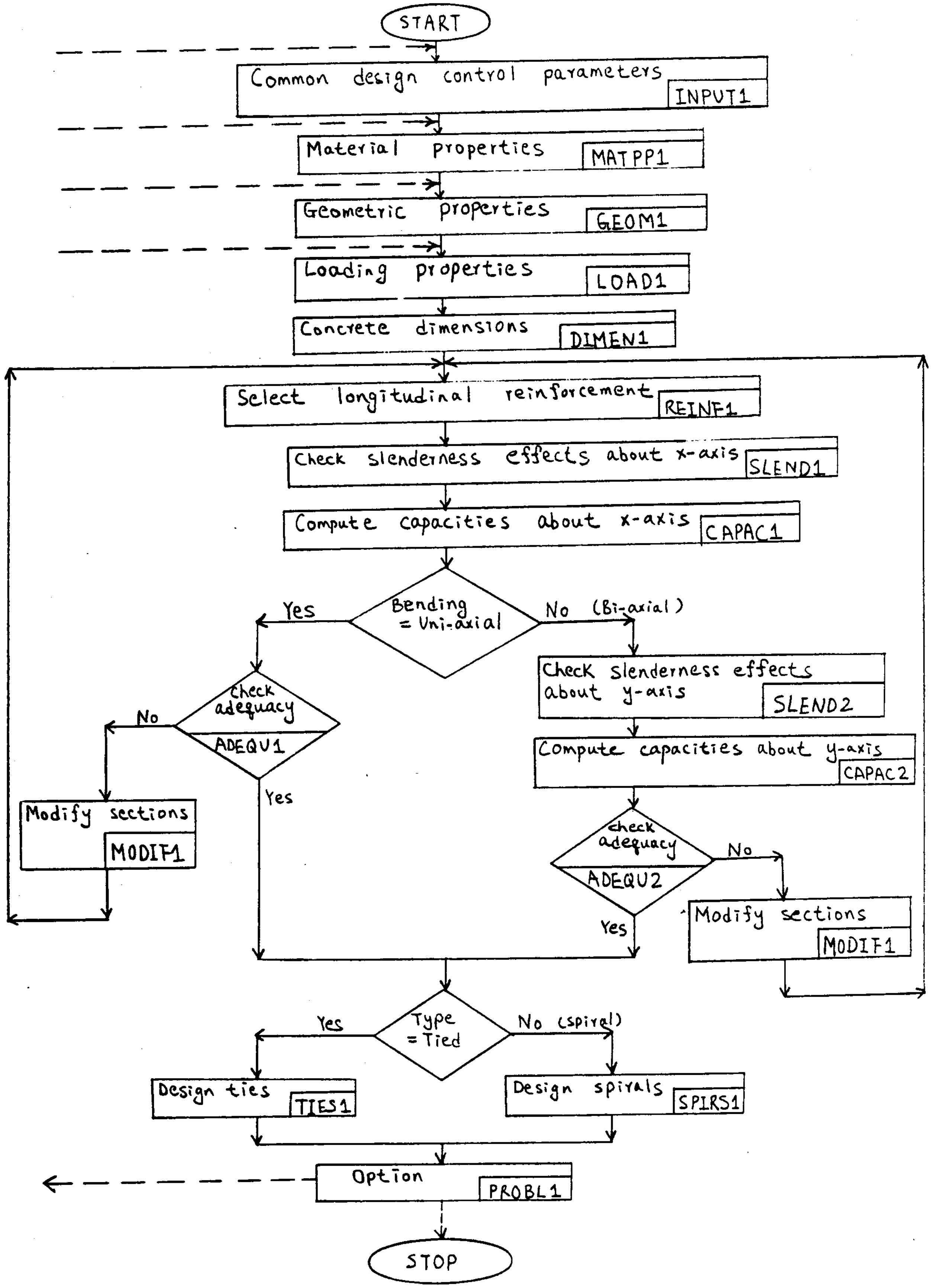


Fig. B.1 A general flow chart of COLUMN1

(6) Used only if in interactive mode.
If altering material properties only, go to (7).
If altering concrete geometry only, go to (9).
If altering reinforcement geometry only, go to (9).
If altering stability geometry only, go to (9).
If altering loading conditions only, go to (12).

(7) Read the material properties (Sub: *MATPP1*).
(If in batch mode, go to (9).)

(8) Used only if in interactive mode, if the revised design alters material properties only, go to (13).

(9) Read the geometry properties (Sub: *GEOM1*).
(If in batch mode, go to (11).)

(10) Used only if in interactive mode, if the revised design alters concrete geometry only, or alters steel geometry only, or alters stability geometry only, go to (13).

(11) Set N = NLC to indicate a new design with no previous counting of iterations in which NLC is total number of load cases which is the same number for every new set of problem (in interactive mode, NLC is equal to 1);
N is a counter do loop during a program run

of each design.

(12) Read the loading properties (Sub: *LOAD1*).

(13) If the condition of analysis is to check for adequacy in strength of selected column section for multiple loading cases, dimension default flag TDIMFG will be defaults to 1 for unaltered dimensions, and then go to (18).

(14) Select the initial concrete dimension (Sub: *DIMEN1*).

(15) If the entered dimension is less than minimum concrete column dimension, go to (33).

(16) Select longitudinal reinforcement (Sub: *REINF1*).

(17) If the reinforcement consistent with the entered restraints could not be obtained, a transfer of control to (33).

(18) Check slenderness effects about x-axis (Sub: *SLEND1*).

(19) If data input is incomplete for computing magnification factor, a transfer of control to (34). When new column dimensions are required due to stability problem, return to (16).

(20) Compute capacities about x-axis (Sub: *CAPAC1*).

(21) If it is bi-axial bending, go to (23).

(22) Check capacities for adequacy, then go to (27) (Sub: *ADEQU1*).

(23) Check slenderness effects about y-axis (Sub: *SLEND2*).

(24) If data input is incomplete for computing

magnification factor, a transfer of control to (35).

(25) Compute capacities about y-axis (Sub: *CAPAC2*).

(26) Check capacities for adequacy (Sub: *ADEQU2*).

(27) If the resisting strength is not adequate for a check mode, go to (33).
If design is adequate, go to (30).

(28) Otherwise modify trial section if required (Sub: *MODIF1*).

(29) If design cannot be completed consistent with the entered restraints, go to (33).
If a new section is tried, reiterate step (16).

(30) If it is a tied column, go to (32).

(31) Select lateral reinforcement for spiral column, and then go to (33) (Sub: *SPIRS1*).

(32) Select lateral reinforcement for tied (Sub: *TIES1*).

(33) If a long output is desired, go to (35).

(34) Otherwise a desired short output is printed (Sub: *OUTPUT*).

(35) If a counter of N (the number of load cases) is less than or equal to 1, go to (37).

(36) Otherwise sort the output with the heading printed at the top of all subsequent pages for the same design, and all messages information printed afterwards, then reiterate step (12). (Sub: *SORT*).

(37) Read the next design control parameter (Sub: PROBL1).
If in batch mode :
Enter 1 means a new set of all data records.

Enter 2 means heading varied with following concrete geometry

Enter 3 means heading varied with following reinforcement geometry

Enter 4 means heading varied with following stability geometry

Enter 5 means heading varied with following loading conditions

Enter 6 means the last column design is solved.

If in interactive mode :

Enter 1 means revising above design in which a next data record is required as follows

: enter 1 means altering material properties only;

: enter 2 means altering concrete of geometry only;

: enter 3 means altering reinforcement of geometry only;

: enter 4 means altering stability of geometry only;

: enter 5 means altering loading conditions only.

Enter 2 means going to next design in which the next data record required is the same as those in batch mode.

Enter 3 means termination of program.

(38) Then sort the output (Sub: *SORT*).

(If not revising above design in interactive mode, go to (40))

(39) Used only if revising above design in interactive mode, return to (6).

(40) If it is not the last column design, return to (3).

(41) Otherwise, end of design.

B.1 **Flags Default**

(a) *Dimension Default Flags*

DIMFLAG (TDIMFG) = 1 means Cx and Cy, or Cx and Cx/Cy, or Cy and Cx/Cy is entered (i.e. cannot alter either Cx or Cy)

= 2 means only Cx is entered (i.e. can alter Cy only)

= 3 means only Cy is entered (i.e. can alter Cx only)

= 4 means only Cx/Cy is entered (i.e. can alter Cx and Cy to most closely approximate the entered aspect ratio)

= 5 means Cx,Cy and Cx/Cy are unknown (i.e. can alter either Cx or Cy)

(b) *Reinforcement Default Flag*

MODAS (TMODAS) = 0 means column dimensions are incremented and process of selecting reinforcement repeated

= 1 means bar size is incremented and process of selecting reinforcement repeated

= 2 means number of bars is incremented and process of selecting reinforcement repeated

APPENDIX C

COMPUTER CODING

This appendix presents a source listing of the Main routine and the subroutines of the processing program COLUMN1.

```
C
C    ****************************************************************
C    *                        COLUMN1                               *
C    * A PROGRAM FOR DESIGN AND CHECK OF REINFORCED CONCRETE COLUMNS*
C    * IN ACCORDING TO CSA AND ACI SPECIFICATION BUILDING CODES     *
C    * RUNNING EITHER IN BATCH MODE OR IN INTERACTIVE MODE          *
C    *              IS CODED BY LEUNG CHI-KUN                       *
C    * AS PART OF M.SC. THESIS, DEPARTMENT OF CIVIL ENGINEERING,    *
C    * UNIVERSITY OF ALBERTA,EDMONTON, ALBERTA, CANADA, FALL, 1983. *
C    ****************************************************************
C
C THIS IS THE MAIN ROUTINE OF THE PROGRAM IN WHICH A SEQUENCE OF
C EVENTS IS CODED AND ALL CORRESPONDING SUBROUTINES ARE CALLED IF
C REQUIRED
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
      DIMENSION HED(15) .
      DIMENSION V(5)
C
C           IDENTIFIERS:DEFINE THE VARIABLES IN COMMON BLOCKS
C     PG=MAX. PERCENTAGE OF STEEL
C     ESMAX=MAX. STRESS = FY/ES
C     TYPE=TYPE OF COLUMN : 1 FOR TIED, 2 FOR SPIRAL
C     TREINF=A TRANSFER OF CONTROL IN SUBROUTINE REINF1, 1 GO TO
C            SUBROUTINE SLEND1, 2 PROCEED TO NEXT PROBLEM
C     TSLEN1=A TRANSFER OF CONTROL IN SUBROUTINE SLEND1, 1 RETURN
C            TO MAIN ROUTINE AND CONTINUE EXECUTION, 2 PROCEED TO
C            NEXT PROBLEM, 3 BACK TO SUBROUTINE REINF1
C     TBEND=1 UNIAXIAL BENDING, 2 BIAXIAL BENDING
C     TSLEN2=A TRANSFER OF CONTROL IN SUBROUTINE SLEND2, 1 RETURN
C            TO MAIN ROUTINE AND CONTINUE EXECUTION, 2 PROCEED TO
C            NEXT PROBLEM
C     TADEQ=A TRANSFER OF CONTROL IN SUBROUTINE ADEQU1 AND ADEQU2,
C           1 PROCEED TO NEXT PRONBLEM, 2 GO TO SUBROUTINE MODIF1,
C           3 SECTION IS ADEQUATE, THEN RETURN TO MAIN ROUTINE AND
C           CONTINUE EXECUTION
C     TPROB=A TRANSFER OF CONTROL FOR OPTIONAL DATA ENTRY, 1 ENTER
C           A NEW SET OF ALL DATA, 2 ENTER HEADING WITH FOLLOWING
C           CONCRETE OF GEOM. PROPERTIES, 3 ENTER HEADING WITH
C           FOLLOWING RREINFORCEMENT OF GEOM. PROPERTIES, 4 ENTER
C           HEADING WITH FOLLOWING STABILITY OF GEOM. PROPERTIES,
C           5 ENTER HEADING WITH FOLLOWING LOADING PROPERTIES,
C           6 TERMINATE THE RUN
C     TCODE=1 CSA CODE, 2 ACI CODE
C     WC=MASS DENSITY CONCRETE (KG/CU.M OR PCF)
C     ES=ELASTICITY MODULUS FOR NON-PRESTRESSED REINFORCEMENT
C        (MPA OR PSI)
C     DCC=MINIMUM CLEAR CONCRETE COVER (MM OR IN)
C     DSS=MINIMUM LATERAL REINFORCEMENT DIAMETER (#)
C     CXINCR=INCREMENT OF DIMENSION CX (MM OR IN)
C     CYINCR=INCREMENT OF DIMENSION CY (MM OR IN)
C     TI=DEFAULT MINIMUM BAR SIZE (#)
C     TK=DEFAULT MAXIMUM BAR SIZE (#)
C     TBS(9)=ARRAY DENOTATION OF BAR SIZE (#)
C     DB(9)=ARRAY DENOTATION OF BAR DIAMETER (MM OR IN)
C     FD=DEAD LOAD FACTOR
C     FL=LIVE LOAD FACTOR
C     ECU=MAX. USABLE STRAIN AT EXTREME CONCRETE COMPRESSION FIBER
C         =0.003
C     PMAX=MAX. ALLOWABLE REINFORCEMENT RATIO=0.08
C     PMIN=MIN. ALLOWABLE REINFORCEMENT RATIO=0.01
C     SB1=DEFAULT MIN. CLEAR BAR SPACING (MM OR IN)
C     EMIN=DEFAULT MIN. ECCENTRICITY (MM OR IN)
C     PHIV=DEFAULT CAPACITY REDUCTION FACTOR
C     RFAC=DEFAULT STRENGTH REDUCTION FACTOR
C     DSMIN=DEFAULT MIN. SPIRAL DIAMETER (MM OR IN)
C     SSMIN=DEFAULT MIN. CENTER-TO-CENTER BARS SPACING OF SPIRAL
C           (MM OR IN)
C     SSMAX=DEFAULT MAX. CENTER-TO-CENTER BARS SPACING OF SPIRAL
C           (MM OR IN)
C     TBMAX=MAX. PERMISSIBLE NO. OF BARS
C     SMAX=DEFAULT MAX. CLEAR TIED BAR SPACING (MM OR IN)
C     J=DENOTATION OF BAR SIZE # OR BAR DIAMETER
C     PRIME=3.1415927
C     MINDIM=DEFAULT MIN. DIMENSION IN PRACTICE (MM OR IN)
C     TESTDI=A TRANSFER OF CONTROL IN SUBROUTINE DIMEN1, 1 RETURN TO
C            MAIN ROUTINE AND CONTINUE EXECUTION, 2 PROCEED TO NEXT
C            PROBLEM
C     LATDS(20)=ARRAY DENOTATION OF LATERAL REINFORCEMENT DIAMETER
C               (MM OR IN)
C     MINLAT=DEFAULT MIN. LATERAL REINFORCEMENT DIAMETER (MM OR IN)
C     TBMAXX=ENTERED MAX. PERMISSIBLE NO. OF BARS
C     FPC=SPECIFIED CONCRETE STRENGTH (MPA OR PSI)
C     FY=SPECIFIED YIELD STRESS FOR LONGITUDINAL REINFORCEMENT
C        (MPA OR PSI)
C     FSY=SPECIFIED YIELD STRESS FOR SPIRAL STEEL (MPA OR PSI)
C     EC=ELASTICITY MODULUS FOR CONCRETE (MPA OR PSI)
C     B1=COEFFICIENT RELATING DEPTH OF EQUIVALENT RECT. STRESS
C        BLOCK TO DEPTH FROM COMPRESSION FACE TO NEUTRAL AXIS
C     TXBRAC=BRACING ABOUT X-AXIS : 0 FOR UNBRACED, 1 FOR BRACED
C     KX=EFFECTIVE LENGTH FACTOR ABOUT X-AXIS
C     LUX=UNSUPPORTED LENGTH ABOUT X-AXIS (MM OR IN)
C     TYBRAC=BRACING ABOUT Y-AXIS
C     KY=EFF. LENGTH ABOUT Y-AXIS
C     LUY=UNSUPPORTED LENGTH ABOUT Y-AXIS (MM OR IN)
C     SPU=SUM OF AXIAL DESIGN LOAD FOR COLUMNS IN STOREY (N OR LB)
C     SPC=SUM OF CRITICAL DESIGN LOAD FOR COLUMNS IN STOREY(N OR LB)
C     PD=AXIAL DEAD LOAD (N OR LB)
C     PL=AXIAL LIVE LOAD (N OR LB)
C     PU=REQUIRED AXIAL LOAD (N OR LB)
C     MX1=SMALLER FACTORED END MOMENT ABOUT X-AIXS (N-MM OR LB-IN)
C     MX2=LARGER FACTORED END MOMENT ABOUT X-AXIS (N-MM OR LB-IN)
C     BDX=RATIO OF MAX. FACTORED DEAD LOAD MOMENT TO MAX. FACTORED
C           TOTAL LOAD MOMENT ABOUT X-AXIS
C
```

```
C      MY1=SMALLER FACTORED END MOMENT ABOUT Y-AXIS (N-MM OR LB-IN)
C      MY2=LARGER FACTORED END MOMENT ABOUT Y-AXIS (N-MM OR LB-IN)
C      BDY=RATIO OF MAX. FACTORED DEAD LOAD MOMENT TO MAX. FACTORED
C           TOTAL LOAD MOMENT ABOUT Y-AXIS
C      TOUT=O FOR STANDARD SHORT OUTPUT, 1 FOR LONG OUTPUT IN DETAILS
C      TMODE=O FOR BATCH MODE, 1 FOR INTERACTIVE MODE
C      TSHAPE=SHAPE OF CROSS-SECTION, 1 FOR SQUARE,2 FOR RECTANGULAR,
C              3 FOR CIRCULAR
C      CX=DIMENSION OF COLUMN SECTION IN X DIRECTION (MM OR IN)
C      CY=DIMENSION OF COLUMN SECTION IN Y DIRECTION (MM OR IN)
C      R=RATIO OF CX/CY
C      TDESIG=CONDITION OF DESIGN OR ANALYSIS OF PROBLEM, 1 FOR
C              DESIGN PROBLEM, 2 FOR ADEQUACY CHECK OF SECTIONS FOR
C              MULTIPLE LOAD CASES, 3 FOR ADEQUACY CHECK OF GIVEN
C              SECTIONS
C      TRFACE=REINFORCEMENT PATTERNS, 1 FOR BARS IN X-FACES ONLY,
C              2 FOR BARS IN Y-FACES ONLY, 3 FOR BARS IN BOTH
C              FACES, 4 FOR BARS IN CIRCULAR CORE
C      NBS=SPECIFIED TOTAL NO. OF BARS
C      NMAX=COMPUTED TOTAL NO. OF BARS
C      TDIMFG=DIMENSION DEFAULT FLAGS, 1 FOR CX AND CY ARE KNOWN,
C              2 FOR CX ONLY IS KNOWN, 3 FOR CY ONLY IS KNOWN,
C              4 FOR R IS KNOWN, 5 FOR ALL CX, CY AND R ARE UNKNOWN
C      AG=CROSS AREA OF CROSS-SECTION (SQ. MM OR SQ. IN)
C      AS=MIN. AREA OF LONGITUDINAL REINFORCEMENT (SQ. MM OR SQ. IN)
C      CG=CHARACTERISTIC COEFFICIENT BASED ON SHAPE OF CROSS-SECTION
C      SB=COMPUTED MIN. CLEAR BAR SPACING (MM OR IN)
C      DP=DIST. FROM EXTREME COMPRESSION FIBER TO CENTROID OF
C         TENSION REINFORCEMENT (MM OR IN)
C      AST=TOTAL AREA OF LONGITUDINAL REINFORCEMENT(SQ. MM OR SQ. IN)
C      NX=NO. OF BARS IN LATERAL FACES INCLUDING CORNER BARS
C      NY=NO. OF BARS IN END FACES
C      SBX=ACTUAL CLEAR BAR SPACING MEASURED IN LATERAL FACES
C           (MM OR IN)
C      SBY=ACTUAL CLEAR BAR SPACING MEASURED IN END FACES (MM OR IN)
C      DS=DIST. FROM CENTROID OF EXTREME COMPRESSION REINFORCEMENT
C         TO CENTROID OF EXTREME TENSION REINFORCEMENT IN OPPOSITE
C         HALF OF MEMBER (MM OR IN)
C      Y(12)=ARRAY DENOTATION OF BARS DISTANCE FROM X-AXIS (MM OR IN)
C      ISEX=MOMENT OF INERTIA OF REINFORCEMENT ABOUT X-AXIS
C            (MM**4 OR IN**4)
C      X(12)=ARRAY DENOTATION OF BARS DISTANCE FROM Y-AXIS (MM OR IN)
C      ISEY=MOMENT OF INERTIA OF REINFORCEMENT ABOUT Y-AXIS
C            (MM**4 OR IN**4)
C      TMODAS=REINFORCEMENT DEFAULT FLAG, O MEANS NUMBER OF BARS
C              IS INCREMENTED AND THE PROCESS OF SELECTING
C              REINFORCEMENT REPEATED, 1 MEANS COLUMN DIMENSIONS
C              ARE INCREMENTED AND THE PROCESS OF SELECTING
C              REINFORCEMENT REPEATED, 2 MEANS BAR SIZE IS
C              INCREASED TOGETHER WITH USING MAXIMUM NUMBER OF
C              BARS
C      PGDES=DESIGN REINFORCEMENT RATIO
C      EY=ECCENTRICITY ABOUT X-AXIS (MM OR IN)
C      MCX=MAGNIFIED FACTORED MOMENT ABOUT X-AXIS (N-MM OR LB-IN)
C      MAPPX=APPLIED MOMENT ABOUT X-AXIS (N-MM OR LB-IN)
C      PUDESX=DESIGN AXIAL LOAD ABOUT X-AXIS (N OR LB)
C      MUDESX=DESIGN MOMENT ABOUT X-AXIS (N-MM OR LB-IN)
C      PBX=DESIGN AXIAL LOAD AT BALANCED CONDITION ABOUT X-AXIS
C           (N OR LB)
C      EX=ECCENTRICITY ABOUT Y-AXIS (MM OR IN)
C      MCY=MAGNIFIED FACTORED MOMENT ABOUT Y-AXIS (N-MM OR LB-IN)
C      MAPPY=APPLIED MOMENT ABOUT Y-AXIS (N-MM OR LB-IN)
C      PUDESY=DESIGN AXIAL LOAD ABOUT Y-AXIS (N OR LB)
C      MUDESY=DESIGN MOMENT ABOUT Y-AXIS (N-MM OR LB-IN)
C      PBY=DESIGN AXIAL LOAD AT BALANCED CONDITION ABOUT Y-AXIS
C           (N OR LB)
C      MOYREQ=REQUIRED EQUIVALENT UNIAXIAL MOMENT ABOUT X-AXIS
C              (N-MM OR LB-IN)
C      MOXREQ=REQUIRED EQUIVALENT UNIAXIAL MOMENT ABOUT Y-AXIS
C              (N-MM OR LB-IN)
C      YC=DISTANCE FROM EXTREME COMPRESSION FIBER TO NEUTRAL AXIS
C         PARALLEL TO X-AXIS (MM OR IN)
C      YB=DIST. FORM EXTREME COMPRESSION FIBER TO NEUTRAL AXIS
C         PARALLEL TO X-AXIS AT BALANCED CONDITION (MM OR IN)
C      ECS(12)=ARRAY DENOTATION OF STRAIN IN COMPRESSION STEEL
C                ABOUT X-AXIS
C      ETS(12)=ARRAY DENOTATION OF STRAIN IN TENSION STEEL
C                ABOUT X-AXIS
C      CS(12)=ARRAY DENOTATION OF FORCE IN COMPRESSION STEEL
C              ABOUT X-AXIS (N OR LB)
C      FT(12)=ARRAY DENOTATION OF FORCE IN TENSION STEEL
C              ABOUT X-AXIS (N OR LB)
C      XC=DIST. FROM EXTREME COMPRESSION FIBER TO NEUTRAL AXIS
C         PARALLEL TO Y-AXIS (MM OR IN)
C      XB=DIST. FROM EXTREME COMPRESSION FIBER TO NEUTRAL AXIS
C         PARALLEL TO Y-AXIS AT BALANCED CONDITION (MM OR IN)
C      ECSY(12)=ARRAY DENOTATION OF STRAIN IN COMPRESSION STEEL
C                ABOUT Y-AXIS
C      ETSY(12)=ARRAY DENOTATION OF STRAIN IN TENSION STEEL
C                ABOUT Y-AXIS
C      CSY(12)=ARRAY DENOTATION OF FORCE IN COMPRESSION STEEL
C                ABOUT Y-AXIS (N OR LB)
C      FTY(12)=ARRAY DENOTATION OF FORCE IN TENSION STEEL
C                ABOUT Y-AXIS (N OR LB)
C      TESTMO=A TRANSFER OF CONTROL IN SUBROUTINE MODIF1, 1 BACK TO
C              SUBROUTINE REINF1, 2 PROCEED TO NEXT PROBLEM
C      NLC=TOTAL NO. OF LOAD CASES
C      JPOB=ONLY USED FOR INTERACTIVE MODE, 1 FOR REVISED PROBLEM,
C           2 FOR NEXT PROBLEM, 3 FOR TERMINATION OF PROGRAM
C      TREVIS=ONLY USED IN INTERACTIVE MODE FOR JPOB=1, 1 ONLY FOR
C              ALTERING MATERIAL PROPERTIES, 2 ONLY FOR ALTERING
C              CONCRETE OF GEOM., 3 ONLY FOR ALTERING REINFORCEMENT
C              OF GEOM., 4 ONLY FOR ALTERING STABILITY OF GEOM., 5
C              ONLY FOR ALTERING LOADING PROPERTIES
C      TDSSS=LATERAL REINFORCEMENT USED (#)
C      SPACE=ACTUAL SPACING REQUIRED IN TIES (MM OR IN)
C      PITCH=ACTUAL PITCH REQUIRED IN SPIRALS (MM OR IN)
C      NSPACR=NO. OF SPACERS REQUIRED IN SPIRALS
C
C      A(9)=ARRAY DENOTATION OF AREA OF BAR SIZE (SQ. MM OR SQ. IN)
C      HED=HEADING DESCRIBES THE AIM OF THIS RUN OF THIS PROGRAM
C
C INITIALIZE TPROB FOR THE FIRST RUN
C
       TPROB=1
       WRITE(7,805)
  805  FORMAT('& MODE=? TYPE O BATCH, OR TYPE 1 INTERACTIVE,'/,
      * /'           THEN PRESS RETURN KEY'/)
       READ(4,806) TMODE
  806  FORMAT(I1)
```

```
 111  IF(TMODE.EQ.1) GO TO 900
      READ(5,101) HED
      GO TO 901
 900  WRITE(7,800)
 800  FORMAT('& HEADING= DESCRIPTIVE IDENTIFIER NOT EXCEEDING'/,
     * /'            60 CHARACTERS'/)
      READ(4,101) HED
 901  WRITE(6,200)HED
 200  FORMAT('1',1X,15A4,'PAGE  1'/1X,60('*')/)
      CALL TIME(6,-1,V)
      WRITE(6,1000)V
 1000 FORMAT(33X,2A4,5X,3A4/)
      GO TO (51,52,52,52,999,55),TPROB
C
C READ PROBLEM CONTROL PARAMETERS
C
 51   CALL INPUT1
      IF(TCODE.EQ.1.OR.TCODE.EQ.2) GO TO 701
      WRITE(6,700)
      N=1
      GO TO 72
 701  IF(TYPE.EQ.1.OR.TYPE.EQ.2) GO TO 2
      WRITE(6,702)
      N=1
      GO TO 72
C
C ONLY USED FOR INTERACTIVE MODE
C
 2    IF(TREVIS.EQ.1) GO TO 61
      IF(TREVIS.EQ.2) GO TO 52
      IF(TREVIS.EQ.3) GO TO 52
      IF(TREVIS.EQ.4) GO TO 52
      IF(TREVIS.EQ.5) GO TO 55
C
C READ MATERIAL PROPERTIES OF SECTION
C
 61   CALL MATPP1
      IF(TMODE.EQ.1) GO TO 10
      IF(FPC.LE.0.) GO TO 704
      IF(FY.LE.0.) GO TO 704
      IF(FSY.LE.0.) GO TO 704
      GO TO 10
 704  N=1
      GO TO 72
C
C ONLY USED FOR INTERACTIVE MODE
C
 10   IF(TREVIS.EQ.1) GO TO 62
C
C READ GEOMETRY PROPERTIES OF CONCRETE, REINFORCEMENT AND STABILITY
C
 52   CALL GEOM1
      IF(TSHAPE.EQ.0) GO TO 710
      IF(TRFACE.EQ.0) GO TO 710
      IF(TBEND.EQ.0) GO TO 710
      GO TO 711
 710  N=1
      GO TO 72
C
C ONLY USED FOR INTERACTIVE MODE
C
 711  IF(TREVIS.EQ.2)GO TO 62
      IF(TREVIS.EQ.3) GO TO 62
      IF(TREVIS.EQ.4) GO TO 62
C
C N IS A COUNTER DO LOOP, NLC IS KEEP AS AN INITIAL VALUE FOR
C THE STARTING OF EACH RUN
C
 999  N=NLC
C
C READ LOADING PROPERTIES
C
 55   CALL LOAD1
      IF(TDESIG.EQ.0) GO TO 715
      GO TO 62
 715  N=1
      GO TO 72
 62   IF(TDESIG.EQ.2) GO TO 20
C
C SELECT INITIAL CONCRETE DIMENSION
C
      CALL DIMEN1
      GO TO(6,14),TESTDI
C
C SELECT REINFORCEMENT
C
 6    CALL REINF1
      GO TO (17,14),TREINF
C
C IF TDESIG=2, SET TDIMFG=1 FOR KNOWN DIMENSIONS
C
 20   TDIMFG=1
C
C CHECK SLENDERNESS EFFECTS ABOUT X-AXIS
C
 17   CALL SLEND1
      GO TO (27,14,8),TSLEN1
C
C COMPUTE CAPACITIES ABOUT X-AXIS
C
 27   CALL CAPAC1
      IF(TBEND.EQ.2) GO TO 8
C
C CHECK CAPACITIES FOR ADEQUACY IN UNIAXIAL BENDING
C
      CALL ADEQU1
      GO TO 8
C
C CHECK SLENDERNESS EFFECT ABOUT Y-AXIS
C
 8    CALL SLEND2
      GO TO (22,14),TSLEN2
C
C COMPUTE CAPACITIES ABOUT Y-AXIS
C
 22   CALL CAPAC2
C
C CHECK CAPACITIES FOR ADEQUACY IN BIAXIAL BENDING
C
      CALL ADEQU2
```

```
  9    GO TO (14,16,11),TADEQ
C
C MODIFY SECTIONS IF REQUIRED
C
 16    CALL MODIF1
       GO TO (6,14),TESTMO
 11    IF (TYPE.EQ.1) GO TO 12
C
C SELECT LATERAL REINFORCEMENT FOR SPIRAL COLUMN
C
       CALL SPIRS1
       GO TO 14
C
C SELECT LATERAL REINFORCEMENT FOR TIED
C
 12    CALL TIES1
C
 14    IF(TOUT.EQ.1) GO TO 71
C
C A STANDARD SHORT OUTPUT IS REQUIRED
C
       CALL OUTPUT
C
C IF NOT MULTIPLE PROBLEMS OR N=1, GO TO 72
C
 71    IF(N.LE.1) GO TO 72
C
C SORT THE OUTPUT IN FORMATTED TYPE
C
       CALL SORT(HED,N,JPOB)
       GO TO 55
C
C READ THE NEXT PROBLEM CONTROL PARAMETERS
C
 72    CALL PROBL1
       CALL SORT(HED,N,JPOB)
       IF(JPOB.EQ.1) GO TO 2
       IF(TPROB.EQ.1) GO TO 703
       IF(TCODE.EQ.1) GO TO 1200
       IF(TCODE.EQ.2) GO TO 1200
       GO TO 56
 1200  IF(TYPE.EQ.1) GO TO 1201
       IF(TYPE.EQ.2) GO TO 1201
       GO TO 56
 1201  IF(FPC.LE.0.) GO TO 56
       IF(FY.LE.0.) GO TO 56
       IF(FSY.LE.0.) GO TO 56
 703   IF(TPROB.EQ.1.OR.TPROB.EQ.2) GO TO 712
       IF(TSHAPE.EQ.0) GO TO 56
 712   IF(TPROB.LE.3) GO TO 713
       IF(TRFACE.EQ.0) GO TO 56
 713   IF(TPROB.LE.4) GO TO 714
       IF(TBEND.EQ.0) GO TO 56
 714   IF(TPROB.EQ.5) GO TO 717
       GO TO 718
 717   IF(TDESIG.EQ.0) GO TO 56
 718   GO TO (111,111,111,111,111,56),TPROB
C
 56    STOP
C
 101   FORMAT(15A4)
 700   FORMAT(' ',5X,'INVALID BUILDING CODE ENTERED'/)
 702   FORMAT(' ',5X,'INVALID COLUMN TYPE ENTERED'/)
       END
C
C ***************************************************************
C *                                                             *
       SUBROUTINE INPUT1
C *                                                             *
C *THIS SUBROUTINE READS INPUT CONTROL PARAMETERS AND           *
C *SETS CODE DEFAULT                                            *
C *                                                             *
C ***************************************************************
C
       IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
       COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
       COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,T1,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
       COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
       COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
       COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
       COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
       COMMON TESTMO
       COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
C
       DIMENSION A(9)
       DATA NCODE1,NCODE2/'C','A'/,NTYPE1,NTYPE2/'T','S'/
C
       IF(TMODE.EQ.1) GO TO 900
       READ (5,101) JCODE,JTYPE,NLC,TOUT
       IF(NLC.LE.0) NLC=1
       GO TO 901
 900   WRITE(7,802)
 802   FORMAT('& CODE=?   TYPE=?   OUTPUT LAYOUT=? (2A2,I2)'/,
     * /' CODE: TYPE C FOR CAN3-A23.3-M77',
     * /'             A FOR ACI 318-77',
     * /' TYPE: TYPE T FOR TIED',
     * /'             S FOR SPIRAL',
     * /' OUTPUT: TYPE 0 FOR SHORT',
     * /'               1 FOR LONG'/)
       READ(4,803) JCODE,JTYPE,TOUT
C
C ONLY USED FOR INTERACTIVE MODE, ONE LOAD CASE IS REQUIRED
C
       NLC=1
 803   FORMAT(2A2,I2)
C
C CONVERT ALPHABET TO INTEGER
C
 901   IF(JCODE.EQ.NCODE1) GO TO 600
       IF(JCODE.EQ.NCODE2) GO TO 700
       IF(TMODE.EQ.1) GO TO 701
       TCODE=0
       RETURN
```

```
701  WRITE(7,702)
702  FORMAT('& INVALID BUILDING CODE ENTERED'/,
    * /' ENTER CODE: C FOR CSA',
    * /'             A FOR ACI'/)
     READ(4,703) JCODE
703  FORMAT(A2)
     GO TO 801
700  TCODE=2
     GO TO 601
600  TCODE=1
601  IF(JTYPE.EQ.NTYPE1) GO TO 602
     IF(JTYPE.EQ.NTYPE2) GO TO 704
     IF(TMODE.EQ.1) GO TO 705
     TYPE=0
     RETURN
705  WRITE(7,706)
706  FORMAT('& INVALID COLUMN TYPE ENTERED'/,
    * /' ENTER TYPE: T FOR TIED',
    * /'             S FOR SPIRAL'/)
     READ(4,703) JTYPE
     GO TO 801
704  TYPE=2
     GO TO 603
602  TYPE=1
603  IF(TOUT.NE.1) GO TO 95
     GO TO (97,98), TYPE
97 - WRITE(6,312)
     GO TO 99
98   WRITE(6,313)
99   GO TO(90,91),TCODE
90   WRITE(6,310)
     GO TO 95
91   WRITE(6,311)
C
C SET CODE DEFAULTS
C
 95  IF(TCODE.EQ.1) GO TO 10
     WC=145.
     ES=29000000.
     DCC=1.5
     CXINCR=2.
     CYINCR=2.
C
C BAR DESIGNATION SHOWN IN ARRAY FORM IN INCREMENT OF 1
C
     TBS(1)=5
     TBS(2)=6
     TBS(3)=7
     TBS(4)=8
     TBS(5)=9
     TBS(6)=10
     TBS(7)=11
     TBS(8)=14
     TBS(9)=18
     DB(1)=0.625
     DB(2)=0.750
     DB(3)=0.875
     DB(4)=1.000
     DB(5)=1.128
     DB(6)=1.270
     DB(7)=1.410
     DB(8)=1.693
     DB(9)=2.257
     MINLAT=0.375
     LATDS(3)=0.375
     LATDS(4)=0.500
     LATDS(5)=0.625
     LATDS(6)=0.750
     LATDS(7)=0.875
     LATDS(8)=1.000
     LATDS(9)=1.128
     LATDS(10)=1.270
     LATDS(11)=1.410
     LATDS(14)=1.693
     LATDS(18)=2.257
     SB1=1.5
     EMIN=1.0
     MINDIM=8.
     IF(TYPE.EQ.1) GO TO 11
     PHIV=0.75
     RFAC=0.85
     DSMIN=0.5
     SSMIN=1.0
     SSMAX=3.0
     GO TO 12
 11  PHIV=0.70
     RFAC=0.80
     SMAX=6.0
     GO TO 12
 10  WC=2400.
     ES=200000.
     DCC=40.0
     CXINCR=50.
     CYINCR=50.
     TBS(1)=15
     TBS(2)=20
     TBS(3)=25
     TBS(4)=30
     TBS(5)=35
     TBS(6)=45
     TBS(7)=55
     DB(1)=16.0
     DB(2)=19.5
     DB(3)=25.2
     DB(4)=29.9
     DB(5)=35.7
     DB(6)=43.7
     DB(7)=56.4
     MINLAT=11.3
     LATDS(6)=6.0
     LATDS(10)=11.3
     LATDS(15)=16.0
     LATDS(20)=19.5
     LATDS(25)=25.2
     LATDS(30)=29.9
     LATDS(35)=35.7
     LATDS(45)=43.7
     LATDS(55)=56.4
     SB1=40.
     EMIN=25.
```

```
      MINDIM=200.
      IF(TYPE.EQ.1) GO TO 13
      PHIV=0.75
      RFAC=0.85
      DSMIN=6.0
      SSMIN=25.0
      SSMAX=75.0
      GO TO 12
 13   PHIV=0.70
      RFAC=0.80
      SMAX=150.0
 12   ECU=0.003
      PMAX=0.08
      PMIN=0.01
      PRIME=3.1415927
C
      RETURN
C
C     ******FORMAT STATEMENTS******
C
 101  FORMAT(2A2,2I2)
 310  FORMAT(' ','DESIGN CODE = CSA A23.3-M77'/)
 311  FORMAT(' ','DESIGN CODE = ACI 318-77'/)
 312  FORMAT(' ','TYPE = TIED'/)
 313  FORMAT(' ','TYPE = SPIRAL'/)
C
      END
C
C  ****************************************************************
C  *                                                              *
      SUBROUTINE MATPP1
C  *                                                              *
C  *THIS SUBROUTINE READS MATERIAL PROPERTIES, ECHO CHECKS
C  *THE INPUT DATA AND EVALUATES CODE DEPENDENT VALUES            *
C  *                                                              *
C  ****************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXK
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      IF(TMODE.EQ.1) GO TO 900
      READ (5,100) FPC,FY,FSY,WCC,ESS
      GO TO 901
 900  WRITE(7,800)
 800  FORMAT('8 FPC= ?  FY= ?  FSY= ?  WCC= ?  ESS= ?'/,
     * /'          ( 4F7.0,F12.0)',
     * /' CONCRETE STR. : (MPA OR PSI)',
     * /' LONG. REINF. YIELD STRESS : (MPA OR PSI)',
     * /' SPIRAL STEEL YIELD STRESS : (MPA OR PSI)',
     * /' CONCRETE DENSITY : 0 DEFAULT TO 2400KG/CU.M OR 145 PCF',
     * /' MODULUS OF ELAST. REINF. : 0 DEFAULT TO 200000 MPA OR',
     * /'                                   29000000 PSI'/)
      READ (4,100) FPC,FY,FSY,WCC,ESS
 901  IF(TOUT.NE.1) GO TO 1
      WRITE(6,101)
      WRITE(6,200) FPC,WCC,FY,ESS,FSY
C
C EVALUATE CODE DEPENDENT VALUES AND THE ENTERED MATERIAL
C STRENGTH VALUES
C
 1    F=0.65
      B=0.85
C
C IF WC AND ESS ARE NOT ENTERED, THEIR VALUES WILL BE ASSIGNED
C BY DEFAULT
C
      IF(WCC.GT.0.1) WC=WCC
      IF(ESS.GT.0.1) ES=ESS
 705  IF(FPC.LE.0.) GO TO 310
      IF(FY.LE.0.)  GO TO 311
      IF(FSY.LE.0.) GO TO 312
      IF(TCODE.EQ.1) GO TO 10
      IF(FPC.GE.2000..AND.FPC.LE.8000.) GO TO 15
      IF(TMODE.EQ.1) WRITE(7,210)
      IF(TOUT.NE.1) GO TO 15
      WRITE(6,210)
 15   IF(FY.GE.40000..AND.FY.LE.60000.) GO TO 11
      IF(TMODE.EQ.1) WRITE(7,211)
      IF(TOUT.NE.1) GO TO 11
      WRITE(6,211)
 11   IF(FSY.GE.40000..AND.FSY.LE.60000.) GO TO 12
      IF(TMODE.EQ.1) WRITE(7,212)
      IF(TOUT.NE.1) GO TO 12
      WRITE(6,212)
 12   EC=WC**1.5*33.*SQRT(FPC)
      C=0.85-0.00005*(FPC-4000.)
      D=AMAX1(C,F)
      B1=AMIN1(D,B)
      GO TO 19
 10   IF(FPC.GE.20..AND.FPC.LE.40.) GO TO 16
      IF(TMODE.EQ.1) WRITE(7,210)
      IF(TOUT.NE.1) GO TO 16
      WRITE(6,210)
 16   IF(FY.GE.300..AND.FY.LE.400.) GO TO 13
      IF(TMODE.EQ.1) WRITE(7,211)
      IF(TOUT.NE.1) GO TO 13
      WRITE(6,211)
 13   IF(FSY.GE.300..AND.FSY.LE.400.) GO TO 14
      IF(TMODE.EQ.1) WRITE(7,212)
      IF(TOUT.NE.1) GO TO 14
      WRITE(6,212)
 14   EC=WC**1.5*0.043*SQRT(FPC)
      C=0.85-0.008*(FPC-30.)
      D=AMAX1(C,F)
      B1=AMIN1(D,B)
```

```
 19   ESMAX=FY/ES
      IF(TOUT.NE.1) RETURN
      WRITE(6,400)
      WRITE(6,300)FPC,WC,EC,FY,ES,FSY
      RETURN
 310  IF(TMODE.EQ.1) GO TO 700
      WRITE(6,220)
      RETURN
 311  IF(TMODE.EQ.1) GO TO 701
      WRITE(6,221)
      RETURN
 312  IF(TMODE.EQ.1) GO TO 702
      WRITE(6,222)
      RETURN
 700  WRITE(7,703)
 703  FORMAT(' & INVALID CONCRETE STRENGTH ENTERED'/,
     * /' ENTER CONCRETE STR. : (MPA OR PSI) (F7.0)'/)
      READ(4,704) FPC
 704  FORMAT(F7.0)
      GO TO 705
 701  WRITE(7,706)
 706  FORMAT('& INVALID LONGITUDINAL YIELD STRESS ENTERED'/,
     * /' ENTER LONG. YIELD STRESS : (MPA OR PSI) (F7.0)'/)
      READ(4,704) FY
      GO TO 705
 702  WRITE(7,707)
 707  FORMAT('& INVALID SPIRAL STEEL YIELD STRESS'/,
     * /' ENTER SPIRAL STEEL YIELD STRESS : (MPA OR PSI) (F7.0)'/)
      READ(4,704) FSY
      GO TO 705
C
C     ******FORMAT STATEMENTS******
C
 100  FORMAT(4F7.0,F12.0)
 200  FORMAT(' ',5X,'CONCRETE    : COMP. STRENGTH   =',F10.0,
     *1X,'PSI/MPA',
     *  /6X,'                DENSITY             =',F10.0,1X,'PCF',
     *'/KG/CUBIC METER',
     *    /6X,'LONG. STEEL : YIELD STRESS       =',F10.0,1X,'PSI/MPA',
     *    /6X,'              MODULUS OF ELAS. =',F15.0,1X,'PSI/MPA',
     *    /6X,'SPIRAL       : YIELD STRESS       =',F10.0,1X,'PSI/MPA'/)
 300  FORMAT(' ',5X,'CONCRETE    : COMP. STRENGTH   =',F10.0,1X,
     *'PSI/MPA',
     *    /6X,'              DENSITY             =',F10.0,1X,'PSI/',
     *'KG/CUBIC METER',
     *    /6X,'              MODULUS OF ELAS. =',F15.0,1X,'PSI/MPA',
     *    /6X,'LONG. STEEL : YIELD STRESS       =',F10.0,1X,'PSI/MPA',
     *    /6X,'              MODULUS OF ELAS. =',F15.0,1X,'PSI/MPA',
     *    /6X,'SPIRAL       : YIELD STRESS       =',F10.0,1X,'PSI/MPA'/)
 101  FORMAT(' ','THE FOLLOWING MATERIAL DATA WAS ENTERED : '/)
 400  FORMAT(' ','THE FOLLOWING MATERIAL DATA',
     * ' WAS ASSIGNED BY DEFAULT : '/)
 210  FORMAT(' ',5X,'THE ENTERED CONCRETE STRENGTH IS NOT',
     * ' WITHIN THE'/6X,'PREDETERMINED RANGE'/)
 211  FORMAT(' ',5X,'THE ENTERED LONGITUDINAL YIELD STRESS',
     * ' IS NOT WITHIN THE'/6X,'PREDETERMINED RANGE'/)
 212  FORMAT(' ',5X,'THE ENTERED SPIRAL YIELD STRESS IS NOT',
     * ' WITHIN THE'/6X,'PREDETERMINED RANGE'/)
 220  FORMAT(' ','COMMENTS :'//6X,'THE ENTERED CONCRETE STRENGTH',
     * ' IS INVALID'/)
 221  FORMAT(' ','COMMENTS :'//6X,
     *'THE ENTERED LONGITUDINAL YIELD STRESS',
     * ' IS INVALID'/)
 222  FORMAT(' ','COMMENTS :'//6X,
     *'THE ENTERED SPIRAL YIELD STRESS IS',
     * ' INVALID'/)
      END
C
C  ****************************************************************
C  *                                                              *
      SUBROUTINE GEOM1
C  *                                                              *
C  *THIS SUBROUTINE READS SECTION PROPERTIES, REINFORCEMENT       *
C  *AND STABILITY INFORMATION, EVALUATES SOME OF THEIR DATA       *
C  *PARAMETERS                                                    *
C  *                                                              *
C  ****************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
      DATA NSHAP1,NSHAP2,NSHAP3/'S','R','C'/
      DATA NFACE1,NFACE2,NFACE3,NFACE4/'X','Y','B','C'/
      DATA NBEND1,NBEND2/'U','B'/
C
C USED ONLY FOR REVISING GEOM. PROPERTIES IN INTERACTIVE MODE
C
      IF(TREVIS.EQ.4) GO TO 120
      IF(TREVIS.EQ.3) GO TO 111
      IF(TREVIS.EQ.2) GO TO 750
C
C USED FOR MULTIPLE PROBLEMS
C
      IF(TPROB.EQ.3) GO TO 111
      IF(TPROB.EQ.4) GO TO 120
 750  IF(TMODE.EQ.1) GO TO 900
      READ(5,100) JSHAPE,CX,CY,R,DCCC,CXINC,CYINC
      GO TO 901
 900  WRITE(7,800)
 800  FORMAT('& SHAPE=? CX=? CY=? R=? DCC=? CXINCR=? CYINCR=?'/,
     * /'          ( A2,2F6.1,F5.2,3F5.1)',
     * /' SHAPE : S FOR SQUARE, R FOR RECT., C FOR CIRCULAR',
     * /' DIM. IN X DIR. : (MM OR IN.)',
     * /' DIM. IN Y DIR. : (MM OR IN.)',
     * /' RATIO : (CX/CY)',
     * /' MIN. CLEAR CONCR. COVER : O DEFAULT TO 40 MM OR 1.5 IN.',
```

```
     * /' LENGTH INCRE. OF CX : O DEFAULT TO 50 MM OR 2 IN.',
   . * /' LENGTH INCRE. OF CY : O DEFAULT TO 50 MM OR 2 IN.'/)
      READ(4,100) JSHAPE,CX,CY,R,DCCC,CXINC,CYINC
C
C CONVERT ALPAHBET TO INTEGER
C
 901  IF(JSHAPE.EQ.NSHAP1) GO TO 600
      IF(JSHAPE.EQ.NSHAP2) GO TO 601
      IF(JSHAPE.EQ.NSHAP3) GO TO 770
      IF(TMODE.EQ.1) GO TO 771
      WRITE(6,772)
 772  FORMAT( ' ',5X,'INVALID SHAPE OF CROSS-SECT. ENTERED'/)
      TSHAPE=0
      RETURN
 771  WRITE(7,773)
 773  FORMAT('& INVALID SHAPE OF CROSS-SECT. ENTERED'/,
     * /' ENTER SHAPE : S FOR SQUARE',
     * /'                R FOR RECT.',
     * /'                C FOR CIRCULAR'/)
      READ(4,774) JSHAPE
 774  FORMAT(A2)
      GO TO 901
 770  TSHAPE=3
      GO TO 602
 600  TSHAPE=1
      GO TO 602
 601  TSHAPE=2
 602  IF(TOUT.NE.1) GO TO 111
      WRITE(6,299)
      GO TO (90,91,92),TSHAPE
 90   WRITE(6,310)
      GO TO 93
 91   WRITE(6,311)
      GO TO 93
 92   WRITE(6,312)
 93   IF(TSHAPE.EQ.3) GO TO 2880
      WRITE(6,2882) CX,CY,R
 2882 FORMAT(' ',2X,'CX = ',F6.1,1X,'IN/MM'/3X,'CY =',F6.1,1X,
     *'IN/MM'/3X,'CX/CY =',F5.2/)
      GO TO 2883
 2880 WRITE(6,2881)CX
 2881 FORMAT(' ',2X,'DIAMETER =',F6.1,1X,'IN/MM'/)
 2883 WRITE(6,201) DCCC,CXINC,CYINC
C
C IF ONLY REVISING CONCRETE PROPERTIES, GO TO 2
C
 111  IF(TREVIS.EQ.2) GO TO 2
C
      IF(TMODE.EQ.1) GO TO 903
      READ(5,101) JRFACE,PG,TBMAXX,NBS,TBSMI,TBSMA,TDSSS,NX,NY
      GO TO 904
 903  WRITE(7,804)
 804  FORMAT('& RFACE=? PG=? BMAXX=? NBS=? BSMIN=#? ',
     * ' BSMAX=#? DS=#? NX=? NY=?'/,
     * /'          ( A2,F7.4,4I4,I3,2I2)',
     * /' REINF. PATTERN : X FOR X-FACES, Y FOR Y-FACES',
     * /'                  B FOR BOTH FACES, C FOR CIRCULAR',
     * /' REINF. RATIO : O DEFAULT TO 0.03',
     * /' MAX. PERMISSIBLE NO. BARS : O DEFAULT VALUE BY PROGRAM',
     * /' TOTAL NO. BARS : (NX, NY REQUIRED IF BARS IN BOTH FACES')
      WRITE(7,799)
 799  FORMAT('& MIN. BAR SIZE : O DEFAULT TO 15 OR 5'/,
     * /' MAX. BAR SIZE : O DEFAULT TO 35 OR 11',
     * /' MIN. LATERAL REINF. : O DEFAULT OT 10 OR 3',
     * /' NO. BARS IN EACH X-FACE : (NBS IS ENTERED)',
     * /' NO. BARS IN EACH Y-FACE : (NBS IS ENTERED)'/)
      READ(4,101) JRFACE,PG,TBMAXX,NBS,TBSMI,TBSMA,TDSSS,NX,NY
C
C IF DATA IS NOT ENTERED, ITS VALUE WOULD BE ASSIGNED BY DEFAULT
C
 904  IF(TBMAXX.NE.0) TBMAX=TBMAXX
C
C CONVERT ALPHABET TOO INTEGER
C
 779  IF(JRFACE.EQ.NFACE1) GO TO 603
      IF(JRFACE.EQ.NFACE2) GO TO 604
      IF(JRFACE.EQ.NFACE3) GO TO 605
      IF(JRFACE.EQ.NFACE4) GO TO 775
      IF(TMODE.EQ.1) GO TO 776
      WRITE(6,777)
 777  FORMAT(' ',5X,'INVALID REINFORCEMENT PATTERN ENTERED'/)
      TRFACE=0
      RETURN
 776  WRITE(7,778)
 778  FORMAT(~'& INVALID REINF. PATTERN ENTERED'/,
     * /' ENTER REINF. PATTERN : X FOR X-FACES, Y FOR Y-FACES',
     * /'                        B FOR BOTH FACES, C FOR CIRCULAR'/)
      READ(4,774) JRFACE
      GO TO 779
 775  TRFACE=4
      GO TO 606
 603  TRFACE=1
      GO TO 606
 604  TRFACE=2
      GO TO 606
 605  TRFACE=3
C
C ASSIGN MIN. BAR SIZE # TO CORRESPONDING ARRAY DENOTATION
C
 606  IF(TCODE.EQ.1) GO TO 607
      IF(TBSMI.GT.10) GO TO 608
      IF(TBSMI.LT.5) GO TO 608
      TI=TBSMI-4
 608  IF(TBSMI.LT.5) TI=1
      IF(TBSMI.EQ.11) TI=7
      IF(TBSMI.GE.12.AND.TBSMI.LE.14) TI=8
      IF(TBSMI.GE.15) TI=9
      IF(TBSMA.GT.10) GO TO 609
      IF(TBSMA.LT.5) GO TO 609
      TK=TBSMA-4
 609  IF(TBSMA.LE.1) TK=7
      IF(TBSMA.GE.2.AND.TBSMA.LE.5) TK=1
      IF(TBSMA.EQ.11) TK=7
      IF(TBSMA.GE.12.AND.TBSMA.LE.14) TK=8
      IF(TBSMA.GE.15) TK=9
      GO TO 610
 607  IF(TBSMI.LE.15) TI=1
      IF(TBSMI.GE.16.AND.TBSMI.LE.20) TI=2
      IF(TBSMI.GE.21.AND.TBSMI.LE.25) TI=3
      IF(TBSMI.GE.26.AND.TBSMI.LE.30) TI=4
```

```
      IF(TBSMI.GE.31.AND.TBSMI.LE.35) TI=5
      IF(TBSMI.GE.36.AND.TBSMI.LE.45) TI=6
      IF(TBSMI.GE.46) TI=7
      IF(TBSMA.LE.1) TK=5
      IF(TBSMA.GE.2.AND.TBSMA.LE.15) TK=1
      IF(TBSMA.GE.16.AND.TBSMA.LE.20) TK=2
      IF(TBSMA.GE.21.AND.TBSMA.LE.25) TK=3
      IF(TBSMA.GE.26.AND.TBSMA.LE.30) TK=4
      IF(TBSMA.GE.31.AND.TBSMA.LE.35) TK=5
      IF(TBSMA.GE.36.AND.TBSMA.LE.45) TK=6
      IF(TBSMA.GE.46) TK=7
 610  IF(TOUT.NE.1) GO TO 120
      GO TO(97,98,99,89),TRFACE
 97   WRITE(6,315)
      GO TO 88
 98   WRITE(6,316)
      GO TO 88
 99   WRITE(6,317)
      GO TO 88
 89   WRITE(6,318)
 88   WRITE(6,202) TBS(TI),TBS(TK),NBS,PG,TBMAXX,TDSSS,NX,NY
C
C IF ONLY REVISING REINFORCEMENT PROPERTIES, GO TO 2
C
 120  IF(TREVIS.EQ.3) GO TO 2
C
      IF(TMODE.EQ.1) GO TO 1900
      READ(5,1100) JBEND,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,SPC
      GO TO 2999
 1900 WRITE(7,1800)
 1800 FORMAT('&BEND=?  XBRAC=?  KX=?  LUX=?  YBRAC=?  KY=? LUY=?',
     *' SPU=?  SPC=? '/,
     * /'         ( A2,I4,F7.3,F10.0,I4,F7.3,F10.0,2F8.0)',
     * /' BENDING : U FOR UNIAXIAL',
     * /'           B FOR BIAXIAL',
     * /' BRACING ABOUT X-AXIS : 0 FOR UNBRACED',
     * /'                        1 FOR BRACED',  .
     * /' EFF. LENGTH FACTOR : (ABOUT X-AXIS)')
      WRITE(7,798)
 798  FORMAT('& UNSUP. LENGTH ABOUT X-AXIS : (MM OR IN.)'/,
     * /' BRACING ABOUT Y-AXIS : 0 FOR UNBRACD',
     * /'                        1 FOR BRACED',
     * /' EFF. LENGTH FACTOR : (ABOUT Y-AXIS)',
     * /' UNSUP. LENGTH ABOUT Y-AXIS : (MM OR IN.)',
     * /' SUM AXIAL DESIGN LOAD : (KN OR KIPS)',
     * /' SUM CRITICAL LOAD : (KN OR KIPS)'/)
      READ(4,1100) JBEND,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,SPC
 2999 SPU=SPU*1000.
      SPC=SPC*1000.
C
C CONVERT ALPHABET TO INTEGER
C
 1901 IF(JBEND.EQ.NBEND1) GO TO 611
      IF(JBEND.EQ.NBEND2) GO TO 780
      IF(TMODE.EQ.1) GO TO 781
      WRITE(6,782)
 782  FORMAT(' ',5X,'INVALID BENDING ENTERED'/)
      TBEND=0
      RETURN
 781  WRITE(7,783)
 783  FORMAT('& INVALID BENDING ENTERED'/,
     * /' ENTER BENDING : U FOR UNIAXIAL',
     * /'                 B FOR BIAXIAL'/)
      READ(4,774) JBEND
      GO TO 1901
 780  TBEND=2
      GO TO 612
 611  TBEND=1
 612  IF(TOUT.NE.1) GO TO 2
      GO TO(190,191),TBEND
 190  WRITE(6,1310)
      GO TO 192
 191  WRITE(6,1311)
 192  WRITE(6,1200) TXBRAC,KX,LUX,TYBRAC,KY,LUY
C
C EVALLUATE SOME OF OPTIONAL DATA PARAMETERS
C
 2    IF(R.LE.0..AND.TSHAPE.NE.2) R=1.0
      IF(PG.LE.0.0001) PG=0.03
      IF(PG.LE.PMIN) PG=PMIN
      IF(PG.GE.PMAX) PG=PMAX
      IF(DCCC.GT.0.1) DCC=DCCC
      IF(CXINC.GT.0.1) CXINCR=CXINC
      IF(CYINC.GT.0.1) CYINCR=CYINC
      IF(TDSSS.EQ.0) GO TO 20
      DSS=LATOS(TDSSS)
      GO TO 21
 20   DSS=MINLAT
      IF(TCODE.EQ.1) TDSSS=10
      IF(TCODE.EQ.2) TDSSS=3
 21   IF(TOUT.NE.1) GO TO 3000
      WRITE(6,400)
 3000 IF(TOUT.NE.1) GO TO 3
      WRITE(6,300)
      WRITE(6,301) PG,DCC,DSS,CXINCR,CYINCR
 3    IF(TOUT.NE.1) GO TO 3001
      WRITE(6,500)
 3001 IF(TBSMI.LT.5) GO TO 4
      IF(TOUT.NE.1) GO TO 5
      WRITE(6,501) TBS(TI)
      GO TO 5
 4    IF(TOUT.NE.1) GO TO 5
      WRITE(6,502) TBS(TI)
 5    IF(TBSMA.LT.5) GO TO 6
      IF(TOUT.NE.1) GO TO 7
      WRITE(6,503) TBS(TK)
      GO TO 7
 6    IF(TOUT.NE.1) GO TO 7
      WRITE(6,504) TBS(TK)
 7    GO TO(8,9,10,11),TRFACE
 8    IF(TOUT.NE.1) GO TO 3002
      WRITE(6,505)
      GO TO 12
 9    IF(TOUT.NE.1) GO TO 3002
      WRITE(6,506)
      GO TO 12
 10   IF(TOUT.NE.1) GO TO 3002
      WRITE(6,507)
      GO TO 12
 11   IF(TOUT.NE.1) GO TO 3002
```

```
      WRITE(6,508)
 12   WRITE(6,509)
 3002 GO TO(13,14,15),TSHAPE
 13   IF(TOUT.NE.1) RETURN
      WRITE(6,510)
      GO TO 16
 14   IF(TOUT.NE.1) RETURN
      WRITE(6,511)
      GO TO 16
 15   IF(TOUT.NE.1) RETURN
      WRITE(6,512)
 16   IF(TSHAPE.EQ.3) GO TO 2884
      WRITE(6,513) CX,CY,R
      RETURN
 2884 WRITE(6,2885) CX
 2885 FORMAT(13X,'DIAMETER = ',F6.1,1X,'IN/MM'/)
      RETURN
C
C      ******FORMAT STATEMENTS******
C
 100  FORMAT(A2,2F8.1,F5.2,3F5.1)
 201  FORMAT(' ',2X,
     *'MINIMUM CLEAR CONCRETE COVER =',
     * F5.1,1X,'IN/MM'/3X,'CXINCR = ',F5.1,1X,'IN/MM'/3X,
     * 'CYINCR= ',F5.1,1X,'IN/MM')
 101  FORMAT(A2,F7.4,4I4,I3,2I2)
 202  FORMAT(' ',2X,'MINIMUM BAR SIZE =  #',
     * I4/3X,'MAXIMUM BAR SIZE =  #',I4/3X,
     * 'SPECIFIED TOTAL NUMBER OF BARS USED = ',I4/3X,
     * 'MAXIMUM PERCENTAGE OF STEEL = ',F7.4/3X,
     * 'ENTERED MAXIMUM PERMISSIBLE NO. OF BARS IN A FACE = ',
     * I4/3X,'MINIMUM LATERAL REINFORCEMENT DIAMETER',
     * ' ENTERED = # ',I3/3X,'NO. OF BARS IN X-FACE',
     * ' ONLY =',I2/3X,'NO. OF BARS IN Y-FACE ONLY = ',I2)
C
 300  FORMAT(2X,'PG',7X,'DCC',3X,'DSS',3X,'CXINCR',3X,
     * 'CYINCR')
 301  FORMAT(F7.4,3X,F5.1,1X,F6.3,3X,F5.1,4X,F5.1/)
 299  FORMAT(' ','THE FOLLOWING GEOMETRY',
     * ' DATA WAS ENTERED : '/)
 310  FORMAT(' ',2X,'SHAPE = SQUARE')
 311  FORMAT(' ',2X,'SHAPE = RECTANGULAR')
 312  FORMAT(' ',2X,'SHAPE = CIRCULAR')
 315  FORMAT(' ',2X,'REINFORCEMENT = X-FACE ONLY')
 316  FORMAT(' ',2X,'REINFORCEMENT = Y-FACE ONLY')
 317  FORMAT(' ',2X,'REINFORCEMENT = BOTH FACES')
 318  FORMAT(' ',2X,'REINFORCEMENT = AROUND THE CORE DIAMETER')
 400  FORMAT(' ','THE FOLLOWING GEOMETRIC DATA',
     * ' WAS ASSIGNED BY DEFAULT : '/)
 500  FORMAT(2X,'LONGITUDINAL REINFORCEMENT DETAILS:'/)
 501  FORMAT(13X,'MINIMUM SIZE=  #',I4)
 502  FORMAT(13X,'DEFAULT MINIMUM SIZE=  #',I4)
 503  FORMAT(13X,'MAXIMUM SIZE=  #',I4)
 504  FORMAT(13X,'DEFAULT MAXIMUM SIZE=  #',I4)
 505  FORMAT(13X,'REINFORCEMENT IN X-FACE ONLY'/)
 506  FORMAT(13X,'REINFORCEMENT IN Y-FACE ONLY'/)
 507  FORMAT(13X,'REINFORCEMENT IN BOTH FACES'/)
 508  FORMAT(13X,'REINFORCEMENT AROUND THE CORE DIAMETER'/)
 509  FORMAT(2X,'SPECIFIED DIMENSIONS:'/)
 510  FORMAT(13X,'SHAPE= SQUARE')
 511  FORMAT(13X,'SHAPE= RECTANGULAR')
 512  FORMAT(13X,'SHAPE= CIRCULAR')
 513  FORMAT(13X,'CX=',F6.1,1X,'IN/MM',3X,'CY=',F6.1,1X,'IN/MM..',
     *3X,'CX/CY= ',F5.2/)
 1100  FORMAT(A2,I4,F7.3,F10.0,I4,F7.3,F10.0,2F8.0)
 1200  FORMAT(' ',2X,'BRACING WHERE 0 VALUE FOR UNBRAC., 1 FOR',
     *' BRAC.'/3X,'XBRACING = ',I4/3X,'ROTATIONAL RESTRAINT',
     * ' ABOUT X-AXIS BENDING = ',F7.3/3X,'UNSUPPORTED LENGTH',
     * ' ABOUT X-AXIS = ',F10.0,1X,'IN/MM'/3X,'YBRACING = ',I4/3X,
     * 'ROTATIONAL RESTRAINT ABOUT Y-AXIS BENDING = ',
     * F7.3/3X,'UNSUPPORTED LENGTH ABOUT Y-AXIS = ',F10.0,1X,
     *'IN/MM'/)
 1310  FORMAT(' ',2X,'BENDING = UNI-AXIAL')
 1311  FORMAT(' ',2X,'BENDING = BI-AXIAL')
      END
C
C  ****************************************************************
C  *                                                              *
      SUBROUTINE LOAD1
C  *                                                              *
C  *THIS SUBROUTINE READS LOADING PROPERTIES, COMPUTES FACTORED   *
C  *LOADING WITH RESPECT TO BENDING CONDITIONS                    *
C  *                                                              *
C  ****************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(8)
      DATA NFACT/'F'/
      DATA NDES1,NDES2,NDES3/'D','L','C'/
C
      IF(TMODE.EQ.1) GO TO 902
      READ(5,101) JFACT,PD,PL,MDXT,MLXT,MDXB,MLXB
      GO TO 1990
 902  WRITE(7,804)
 804  FORMAT('& FACT=? PD=? PL=? MDXT=? MLXT=? MDXB=? MLXB=?'/,
     * /'            ( A2,6F8.0)',
     * /' FACT : U FOR UNFACTORED LOADING ENTERED',
     * /'        F FOR FACTORED LOADING ENTERED',
     * /' AXIAL DEAD AND LIVE LOAD : (KN OR KIPS)',
     * /' MOM. ABOUT X-AXIS - DEAD LOAD ON TOP : (KN-M OR KIP-FT)',
     * /'                     LIVE LOAD ON TOP : (KN-M OR KIP-FT)',
     * /'                     DEAD LOAD ON BOT.: (KN-M OR KIP-FT)',
     * /'                     LIVE LOAD ON BOT.: (KN-M OR KIP-FT)'/)
      READ(4,805) JFACT,PD,PL,MDXT,MLXT,MDXB,MLXB
 805  FORMAT(A2,6F8.0)
```

```
 1990 IF(TCODE.EQ.1) GO TO 1991
      PD=PD*1000.
      PL=PL*1000.
      MDXT=MDXT*12000.
      MLXT=MLXT*12000.
      MDXB=MDXB*12000.
      MLXB=MLXB*12000.
      GO TO 903
 1991 PD=PD*1000.
      PL=PL*1000.
      MDXT=MDXT*1000000.
      MLXT=MLXT*1000000.
      MDXB=MDXB*1000000.
      MLXB=MLXB*1000000.
C
C CONVERT ALPHABET TO INTEGER
C
 903  IF(JFACT.EQ.NFACT) GO TO 500
      TFACT=0
      GO TO 501
 500  TFACT=1
 501  IF(TOUT.NE.1) GO TO 2
      WRITE(6,150)
      IF(TFACT.EQ.1) GO TO 93
      WRITE(6,313)
      GO TO 94
 93   WRITE(6,312)
 94   PPD=PD/1000.
      PPL=PL/1000.
      IF(TCODE.EQ.1) GO TO 1992
      MMDXT=MDXT/12000.
      MMLXT=MLXT/12000.
      MMDXB=MDXB/12000.
      MMLXB=MLXB/12000.
      GO TO 1993
 1992 MMDXT=MDXT/1000000.
      MMLXT=MLXT/1000000.
      MMDXB=MDXB/1000000.
      MMLXB=MLXB/1000000.
 1993 WRITE(6,201) PPD,PPL,MMDXT,MMLXT,MMDXB,MMLXB
 2    IF(TMODE.EQ.1) GO TO 904
      READ(5,102) JDESIG,MDYT,MLYT,MDYB,MLYB
      GO TO 1994
 904  WRITE(7,806)
 806  FORMAT('& DESIGN=? MDYT=? MLYT=? MDYB=? MLYB=? (A2,4F8.0)'/,
     * /' DESIGN : D FOR PROGRAM DESIGNS A SECT.',
     * /' : L FOR ADEQ. CHECK OF SELECTED SECT.(MULT. LOAD CASES)',
     * /' : C FOR A CHECK OF ENTERED SECT.',
     * /' MOM. ABOUT Y-AXIS - DEAD LOAD ON TOP : (KN-M OR KIP-FT)',
     * /'                     LIVE LOAD ON TOP : (KN-M OR KIP-FT)',
     * /'                     DEAD LOAD ON BOT.: (KN-M OR KIP-FT)',
     * /'                     LIVE LOAD ON BOT.: (KN-M OR KIP-FT)'/)
      READ(4,807) JDESIG,MDYT,MLYT,MDYB,MLYB
 807  FORMAT(A2,4F8.0)
 1994 IF(TCODE.EQ.1) GO TO 1995
      MDYT=MDYT*12000.
      MLYT=MLYT*12000.
      MDYB=MDYB*12000.
      MLYB=MLYB*12000.
      GO TO 905
 1995 MDYT=MDYT*1000000.
      MLYT=MLYT*1000000.
      MDYB=MDYB*1000000.
      MLYB=MLYB*1000000.
C
C CONVERT ALPHABET TO INTEGER
C
 905  IF(JDESIG.EQ.NDES1) GO TO 502
      IF(JDESIG.EQ.NDES2) GO TO 503
      IF(JDESIG.EQ.NDES3) GO TO 750
      IF(TMODE.EQ.1) GO TO 751
      WRITE(6,752)
 752  FORMAT(' ',5X,'INVALID DESIGN CONDITION ENTERED'/)
      TDESIG=0
      RETURN
 751  WRITE(7,753)
 753  FORMAT('& INVALID DESIGN CONDITION ENTERED'/,
     * /' ENTER DESIGN : D FOR PROGRAM DESIGNS A SECT.',
     * /' : L FOR ADEQ. CHECK OF SELECTED SECT.(MULT. LOAD CASES)',
     * /' : C FOR A CHECK OF ENTERED SECT.'/)
      READ(4,754) JDESIG
 754  FORMAT(A2)
      GO TO 905
 750  TDESIG=3
      GO TO 504
 502  TDESIG=1
      GO TO 504
 503  TDESIG=2
 504  IF(TOUT.NE.1) GO TO 3
      MSPU=SPU/1000.
      MSPC=SPC/1000.
      IF(TCODE.EQ.1) GO TO 1996
      MMDYT=MDYT/12000.
      MMLYT=MLYT/12000.
      MMDYB=MDYB/12000.
      MMLYB=MLYB/12000.
      GO TO 1997
 1996 MMDYT=MDYT/1000000.
      MMLYT=MLYT/1000000.
      MMDYB=MDYB/1000000.
      MMLYB=MLYB/1000000.
 1997 WRITE(6,202) MMDYT,MMLYT,MMDYB,MMLYB,MSPU,MSPC
      GO TO(130,131,132),TDESIG
 130  WRITE(6,135)
      GO TO 3
 131  WRITE(6,136)
      GO TO 3
 132  WRITE(6,137)
C
C COMPUTE FACTORED LOADINGS
C
 3    IF(TFACT.EQ.1) GO TO 10
      FD=1.4
      FL=1.7
      PU=FD*PD+FL*PL
      MXT=FD*MDXT+FL*MLXT
      MXB=FD*MDXB+FL*MLXB
      MYT=FD*MDYT+FL*MLYT
      MYB=FD*MDYB+FL*MLYB
      GO TO 11
```

```
 10   PU=PD+PL
      MXT=MDXT+MLXT
      MXB=MDXB+MLXB
      MYT=MDYT+MLYT
      MYB=MDYB+MLYB
      FD=1.0
      FL=1.0
C
C DETERMINE WHICH IS SMALLER AND LARGER END MOMENTS
C
 11   IF(ABS(MXT).LE.O..AND.ABS(MXB).LE.O.) GO TO 602
      IF(ABS(MXT).GE.ABS(MXB)) GO TO 12
      MX1=MXT
      MX2=MXB
      GO TO 600
 12   MX1=MXB
      MX2=MXT
 600  IF(MDXB.GT.MDXT) GO TO 601
      BBDX=FD*MDXT/MX2
      BDX=ABS(BBDX)
      GO TO 13
 601  BBDX=FD*MDXB/MX2
      BDX=ABS(BBDX)
      GO TO 13
 602  BDX=O.
      MX1=0.0
      MX2=0.0
      GO TO 15
C
C COLUMN DEFLECTS IN SINGLE CURVATURE OR DOUBLE CURVATURE
C
 13   IF(MX1.LT.O..AND.MX2.GT.O.) GO TO 14
      IF(MX2.LT.O..AND.MX1.GT.O.) GO TO 14
      IF(TOUT.NE.1) GO TO 15
      IF(TMODE.EQ.1) WRITE(7,203)
      WRITE(6,203)
      GO TO 15
 14   IF(TOUT.NE.1) GO TO 15
      IF(TMODE.EQ.1) WRITE(7,204)
      WRITE(6,204)
C
C ONLY USED FOR BIAXIAL BENDING
C
 15   IF(TBEND.EQ.1) GO TO 999
      IF(ABS(MYT).LE.O..AND.ABS(MYB).LE.O.) GO TO 605
      IF(ABS(MYT).GE.ABS(MYB)) GO TO 18
      MY1=MYT
      MY2=MYB
      GO TO 603
 18   MY1=MYB
      MY2=MYT
 603  IF(MDYB.GT.MDYT) GO TO 604
      BBDY=FD*MDYT/MY2
      BDY=ABS(BBDY)
      GO TO 19
 604  BBDY=FD*MDYB/MY2
      BDY=ABS(BBDY)
      GO TO 19
 605  BDY=O.
      MY1=0.0
      MY2=0.0
      GO TO 17
 19   IF(MY1.LT.O..AND.MY2.GT.O.) GO TO 20
      IF(MY2.LT.O..AND.MY1.GT.O.) GO TO 20
      IF(TOUT.NE.1) RETURN
      IF(TMODE.EQ.1) WRITE(7,205)
      WRITE(6,205)
      GO TO 17
 20   IF(TOUT.NE.1) RETURN
      IF(TMODE.EQ.1) WRITE(7,206)
      WRITE(6,206)
      GO TO 17
 999  MY2=0.0
 17   IF(TOUT.NE.1) RETURN
      PPU=PU/1000.
      IF(TCODE.EQ.1) GO TO 1998
      MMX2=ABS(MX2)/12000.
      MMY2=ABS(MY2)/12000.
      GO TO 1999
 1998 MMX2=ABS(MX2)/1000000.
      MMY2=ABS(MY2)/1000000.
 1999 IF(TMODE.EQ.1) WRITE(7,303) PPU,MMX2,MMY2
      WRITE(6,303) PPU,MMX2,MMY2
      RETURN
C
C     ******FORMAT STATEMENTS******
C
 101  FORMAT(A2,6F8.0)
 201  FORMAT(' ',2X,'AXIAL DEAD LOAD = ',F8.0,1X,'KIPS/KN'/3X,
     * 'AXIAL LIVE LOAD =',
     * F8.0,1X,'KIPS/KN'/3X,'MDXT= ',F8.0,1X,'KIP-FT/KN-M',
     * /3X,'MLXT= ',F8.0,1X,'KIP-FT/KN-M'/3X,'MDXB= ',
     * F8.0,1X,'KIP-FT/KN-M'/3X,'MLXB= ',F8.0,1X,'KIP-FT/KN-M')
 102  FORMAT(A2,4F8.0)
 202  FORMAT(' ',2X,'MDYT= ',F8.0,1X,'KIP-FT/KN-M'/3X,'MLYT= '
     * ,F8.0,1X,'KIP-FT/KN-M'/3X,'MDYB= ',F8.0,1X,'KIP-FT/KN-M',
     */3X,'MLYB= ',F8.0,1X,'KIP-FT/KN-M'/3X,'SUMMATION',
     * ' OF PU FOR ALL COLUMN IN A STOREY = ',F8.0,1X,'KIPS/KN',
     */3X,'SUMMATION OF CRITICAL LOAD FOR ALL COLUMN IN A STOREY=',
     * F8.0,1X,'KIPS/KN'/)
C
 203  FORMAT(' ',6X,'COLUMN DEFLECTS IN SINGLE',
     * ' CURVATURE ABOUT X-AXIS'/)
 204  FORMAT(' ',6X,'COLUMN DEFLECTS IN DOUBLE',
     * ' CURVATURE ABOUT X-AXIS'/)
 205  FORMAT(' ',6X,'COLUMN DEFLECTS IN SINGLE',
     * ' CURVATURE ABOUT Y-AXIS'/)
 206  FORMAT(' ',6X,'COLUMN DEFLECTS IN DOUBLE',
     * ' CURVATURE ABOUT Y-AXIS'/)
C
 303  FORMAT(' ','LOAD CONDITION :'//2X,'PU =',F8.0,1X,
     * 'KIPS/KN',
     */2X,'MUX=',F8.0,1X,'KIP-FT/KN-M,',2X,'MUY=',F8.0,
     *1X,'KIP-FT/KN-M'/)
 150  FORMAT(' ','THE FOLLOWING LOADING',
     * ' DATA WAS ENTERED : '/)
 312  FORMAT(' ',2X,'LOADING = FACTORED')
 313  FORMAT(' ',2X,'LOADING = UNFACTORED')
 135  FORMAT(' ',2X,'CONDITION= DESIGN'/)
 136  FORMAT(' ',2X,'CONDITION= ADEQUACY CHECK FOR MULTIPLE LOAD',
```

```
     * ' CASE'/)
 137  FORMAT(' ',2X,'CONDITION= CHECK OF',
     * ' ENTERED SECTION'/)
      END
C
C  ****************************************************************
C  *                                                              *
      SUBROUTINE DIMEN1
C  *                                                              *
C  *THIS SUBROUTINE SELECTS INITIAL CONCRETE DIMENSIONS, ECHO     *
C  *CHECKS OF INPUT DATA AND ASSIGNS DIMENSION DEFAULT FLAGS      *
C  *TO CORRESPONDING DIMENSION ENTRY                              *
C  *                                                              *
C  ****************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      CG=1./12.
C
C DETERMINE GROSS SECTION AREA BY USING MAX. AXIAL COMPRESSION
C NOMINAL STRENGTH BASED ON MIN. ECCENTRICITY. ONLY USED FOR
C SELECTING TRIAL SECTION
C
      AG=AGG(PU,PHIV,RFAC,FPC,PG,FY)
C
C EVALUATE AND ECHO CHECK DATA ENTRY
C
      IF(CX.LE.0.1) GO TO 10
      IF(CX.LT.MINDIM) GO TO 30
      IF(CY.GT.0.1) GO TO 80
      IF(R.LE.0.1) GO TO 12
      IF(TSHAPE.EQ.2) GO TO 82
      IF(ABS(R-1.0).LE.0.) GO TO 82
      IF(TOUT.NE.1) GO TO 3000
      IF(TMODE.EQ.1) GO TO 2000
      GO TO 153
 2000 WRITE(7,102)
 153  WRITE(6,102)
 3000 R=1.0
 82   CY=CX/R
      GO TO 11
 80   IF(CY.LT.MINDIM) GO TO 31
      XOY=CX/CY
      IF(TSHAPE.EQ.2) GO TO 81
      IF(ABS(XOY-R).LT.0.01) GO TO 11
      IF(TOUT.NE.1) GO TO 3001
      IF(TMODE.EQ.1) GO TO 2001
      GO TO 154
 2001 WRITE(7,100)
 154  WRITE(6,100)
 3001 CY=CX
      GO TO 11
 81   IF(R.LE.0.1) GO TO 11
      IF(ABS(XOY-R).LT.0.01) GO TO 11
      IF(TOUT.NE.1) GO TO 11
      IF(TMODE.EQ.1) GO TO 2002
      GO TO 155
 2002 WRITE(7,101)
 155  WRITE(6,101)
      GO TO 11
C
C ENTRY OF DIMENSIONS IS CLASSIFIED INTO TDIMFG=2
C
 12   TDIMFG=2
      CY=AG/CX
C
C COMPUTED DIMENSION IS ROUND DOWN TO NEXT SMALLER INTEGER VALUE
C
      CY=CYINCR*IFIX(CY/CYINCR)
      IF(CY.LE.MINDIM) GO TO 34
      GO TO 35
C
C MIN. DIMENSION IS USED IF COMPUTED ONE IS LESS THAN THAT
C
 34   CY=MINDIM
      IF(TOUT.NE.1) GO TO 35
      IF(TMODE.EQ.1) GO TO 2004
      GO TO 156
 2004 WRITE(7,303)
 156  WRITE(6,303)
 35   AG=CX*CY
      GO TO 86
 10   IF(CY.LE.0.1) GO TO 14
      IF(CY.LT.MINDIM) GO TO 31
      IF(R.LE.0.1) GO TO 15
      IF(TSHAPE.EQ.2) GO TO 84
      IF(R.LE.1.0) GO TO 84
      IF(TOUT.NE.1) GO TO 3005
      IF(TMODE.EQ.1) GO TO 2005
      GO TO 157
 2005 WRITE(7,102)
 157  WRITE(6,102)
 3005 R=1.0
 84   CX=CY*R
      GO TO 11
C
C ENTRY OF DIMENSION IS CLASSIFIED INTO TDIMFG=3
C
 15   TDIMFG=3
      CX=AG/CY
C
C COMPUTED DIMENSION IS ROUND DOWN TO NEXT INTEGER VALUE
C
      CX=CXINCR*IFIX(CX/CXINCR)
```

```
      IF(CX.LE.MINDIM) GO TO 36
      GO TO 37
C
C MIN. DIMENSION IS USED IF COMPUTED ONE IS LESS THAN THAT
C
 36   CX=MINDIM
      IF(TOUT.NE.1) GO TO 37
      IF(TMODE.EQ.1) GO TO 2007
      GO TO 158
 2007 WRITE(7,304)
 158  WRITE(6,304)
 37   AG=CX*CY
      GO TO 86
 14   IF(R.LE.0.01) GO TO 16
      IF(TSHAPE.EQ.2) GO TO 85
      IF(R.LE.1.0) GO TO 85
      IF(TOUT.NE.1) GO TO 3006
      IF(TMODE.EQ.1) GO TO 2008
      GO TO 159
 2008 WRITE(7,102)
 159  WRITE(6,102)
 3006 R=1.0
C
C ENTRY OF DIMENSION IS CLASSIFIED INTO TDIMFG=4
C
 85   TDIMFG=4
      IF(TSHAPE.EQ.3) GO TO 22
      F=AG/R
      CY=SQRT(F)
      CY=CYINCR*IFIX(CY/CYINCR)
      IF(CY.LE.MINDIM) GO TO 40
      GO TO 41
 40   CY=MINDIM
      IF(TOUT.NE.1) GO TO 41
      IF(TMODE.EQ.1) GO TO 2010
      GO TO 160
 2010 WRITE(7,303)
 160  WRITE(6,303)
 41   CX=CY*R
      CX=CXINCR*IFIX(CX/CXINCR)
      IF(CX.LE.MINDIM) GO TO 42
      GO TO 43
 42   CX=MINDIM
      IF(TOUT.NE.1) GO TO 43
      IF(TMODE.EQ.1) GO TO 2011
      GO TO 161
 2011 WRITE(7,304)
 161  WRITE(6,304)
 43   AG=CX*CY
      GO TO 86
 22   CG=1./16.
      F=(4.*AG)/PRIME
      CX=SQRT(F)
      CX=CXINCR*IFIX(CX/CXINCR)
      IF(CX.LE.MINDIM) GO TO 38
      GO TO 39
 38   CX=MINDIM
      IF(TOUT.NE.1) GO TO 39
      IF(TMODE.EQ.1) GO TO 2013
      GO TO 162
 2013 WRITE(7,304)
 162  WRITE(6,304)
 39   CY=CX
      AG=(PRIME*CX**2)/4.
      GO TO 86
C
C ENTRY OF DIMENSION IS CLASSIFIED INTO TDIMFG=5
C
 16   TDIMFG=5
C
C FOR UNIAXIAL BENDING, THE CONCEPT OF MIN. ECCENTRICITY (0.1H)
C IS APPLIED . FOR BIAXIAL BENDING, RATIO OF CX/CY IS DETERMINED
C BY CONSIDERING MY2/MX2
C
      IF(TBEND.EQ.1) GO TO 87
C
C IF NO MOMENTS ARE ENTERED, R=1.0
C
      IF(ABS(MX2).LE.0..OR.ABS(MY2).LE.0.) GO TO 240
      R=ABS(MY2/MX2)
      GO TO 241
 240  R=1.0
 241  F=R*AG
      CX=SQRT(F)
      CX=CXINCR*IFIX(CX/CXINCR)
      IF(CX.LE.MINDIM) GO TO 44
      GO TO 45
 44   CX=MINDIM
      IF(TOUT.NE.1) GO TO 45
      IF(TMODE.EQ.1) GO TO 2015
      GO TO 163
 2015 WRITE(7,304)
 163  WRITE(6,304)
 45   CY=CX/R
      CY=CYINCR*IFIX(CY/CYINCR)
      IF(CY.LE.MINDIM) GO TO 46
      GO TO 47
 46   CY=MINDIM
      IF(TOUT.NE.1) GO TO 47
      IF(TMODE.EQ.1) GO TO 2016
      GO TO 164
 2016 WRITE(7,303)
 164  WRITE(6,303)
 47   AG=CX*CY
      GO TO 86
 87   CY=ABS(MX2)/(0.1*PU)
      CY=CYINCR*IFIX(CY/CYINCR)
      IF(CY.LE.MINDIM) GO TO 90
      GO TO 91
 90   CY=MINDIM
      IF(TOUT.NE.1) GO TO 91
      IF(TMODE.EQ.1) GO TO 2017
      GO TO 165
 2017 WRITE(7,303)
 165  WRITE(6,303)
 91   CX=AG/CY
      CX=CXINCR*IFIX(CX/CXINCR)
      IF(CX.LE.MINDIM) GO TO 92
      GO TO 47
 92   CX=MINDIM
```

```
      IF(TOUT.NE.1) GO TO 47
      IF(TMODE.EQ.1) GO TO 2018
      GO TO 166
 2018 WRITE(7,304)
 166  WRITE(6,304)
      GO TO 47
C
C ENTRY OF DIMENSION IS CLASSIFIED INTO TDIMFG=1
C
 11   TDIMFG=1
      GO TO 32
 30   IF(TMODE.EQ.1) GO TO 2019
      GO TO 167
 2019 WRITE(7,3105)
      WRITE(7,301)
 167  IF(TOUT.NE.1) GO TO 3100
      WRITE(6,3105)
      WRITE(6,301)
      GO TO 33
 3100 WRITE(8,301)
      GO TO 33
 31   IF(TMODE.EQ.1) GO TO 2020
      GO TO 168
 2020 WRITE(7,3105)
      WRITE(7,302)
 168  IF(TOUT.NE.1) GO TO 3101
      WRITE(6,3105)
      WRITE(6,302)
      GO TO 33
 3101 WRITE(8,302)
C
C PROCEED TO NEXT PROBLEM DUE TO ENTERED DIMENSIONS LESS THAN MINDIM
C
 33   TESTDI=2
      RETURN
 32   IF(TSHAPE.EQ.3) GO TO 20
      AG=CX*CY
      GO TO 83
 20   CG=1./16.
      AG=(PRIME*CX**2)/4.
 83   IF(TDESIG.EQ.1) GO TO 86
      IF(TOUT.NE.1) GO TO 3010
      IF(TMODE.EQ.1) WRITE(7,400)
      WRITE(6,400)
C
C RETURN TO MAIN ROUTINE TO CONTINUE EXECUTION
C
 3010 TESTDI=1
      RETURN
 86   AS=PMIN*AG
      IF(TOUT.NE.1) GO TO 3012
      IF(TSHAPE.EQ.3) GO TO 2880
      IF(TMODE.EQ.1) GO TO 2024
      GO TO 172
 2024 WRITE(7,201) CX,CY
 172  WRITE(6,201) CX,CY
 201  FORMAT(' ','INITIAL TRIAL CONCRETE DIMENSION :'//2X,'CX = ',
     * F6.1,1X,'IN/MM.,',8X,'CY =',F6.1,1X,'IN/MM'/)
      GO TO 3012
 2880 IF(TMODE.EQ.1) WRITE(7,2881) CX
 2881 FORMAT(' ','INITIAL TRIAL CONCRETE DIMENSION :'//2X,
     * 'DIAMETER = ',F6.1,1X,'IN/MM'/)
      WRITE(6,2881) CX
C
 3012 TESTDI=1
      RETURN
C
 301  FORMAT(' ',5X,'THE GIVEN DIMENSION CX IS LESS THAN THE',
     * ' MINIMUM DIMENSION : '/)
 302  FORMAT(' ',5X,' THE GIVEN DIMENSION CY IS LESS THAN THE',
     * ' MINIMUM DIMENSION : '/)
 303  FORMAT(' ',5X,'MINIMUM DIMENSION CY IS USED'/)
 304  FORMAT(' ',5X,'MINIMUM DIMENSION CX IS USED'/)
 100  FORMAT(' ',5X,'GEOMETRIC INPUT DATA IS WRONG',
     * ' DUE TO ASPECT RATIO'/6X,'FOR NON RECT. SHAPE'/)
 101  FORMAT(' ',5X,'GEOMETRIC INPUT DATA IS WRONG DUE TO ASPECT',
     * ' RATIO'/)
 102  FORMAT(' ',5X,'GEOMETRIC INPUT DATA IS WRONG DUE TO',
     * ' ASPECT RATIO,'/6X,'SET IT EQUAL TO 1.0'/)
 400  FORMAT(' ',5X,'ADEQUACY CHECK OF DESIGN COLUMN SECTION'/)
 3105 FORMAT(' ','COMMENTS :'/)
      END
C
C
      FUNCTION AGG(PU,PHIV,RFAC,FPC,PG,FY)
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
C
      AGG=PU/(PHIV*RFAC*(0.85*FPC+PG*(FY-0.85*FPC)))
C
      RETURN
      END
C
C   ****************************************************************
C   *                                                              *
      SUBROUTINE REINF1
C   *                                                              *
C   *THIS SUBROUTINE SELECTS REINFORCEMENT, CHECKS BARS SPACING    *
C   *AND REINFORCEMENT RATIO, COMPUTES MOMENT OF INERTIA OF        *
C   *REINFORCEMENT                                                 *
C   *                                                              *
C   ****************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
```

```
C
      REWIND 8
      IF(TOUT.NE.1) GO TO 3000
      IF(TESTMO.NE.1) GO TO 3100
      WRITE(8,3101) CX,CY
 3101 FORMAT(' ',5X,'NEW TRIAL CONCRETE DIMENSION :',
     * 2X,'CX = ',F6.1,2X,'CY = ',F6.1,2X,'IN PROJECT UNITS'/)
 3100 IF(TMODE.EQ.1) WRITE(7,400)
      WRITE(8,400)
 3000 IF(TMODAS.EQ.1) GO TO 1
      IF(TMODAS.EQ.2) GO TO 75
      IF(TDESIG.EQ.1) GO TO 15
C
C SPECIFIED BAR SIZE AND REQUIRED TOTAL NO. OF BARS ARE ENTERED
C
      J=TI
      TK=TI
      A(J)=(PRIME*DB(J)**2)/4.
      NMAX=NBS
      AS=NMAX*A(J)
      N=IFIX(NMAX/2.+0.51)
      GO TO 12
 1    NB=NBS
      GO TO 16
 75   NB=NBS
      A(J)=(PRIME*DB(J)**2)/4.
      N=IFIX(NMAX/2.+0.51)
      GO TO 12
C
C FOR DESIGN PROBLEM, SELECT POSSIBLE BAR SIZE AND NO. OF BARS
C
 15   J=TI
      NB=NBS
 16   A(J)=(PRIME*DB(J)**2)/4.
      IF(NB.NE.0) GO TO 62
      NMAX=IFIX(AS/A(J)+0.9)
      N=IFIX(NMAX/2.+0.51)
      NMAX=2*N
      IF(TBMAXX.NE.0) GO TO 74
C MAXIMUM PERMISSIBLE NUMBER OF BARS CONTROLLED
C BY THE LEAST DIMENSION IN WHICH BARS IS LOCATED AT
C
      GO TO(70,501,502,501),TRFACE
 502  IF(CX.GT.CY) GO TO 70
 501  DIM=CX
      GO TO 71
 70   DIM=CY
 71   IF(TCODE.EQ.1) GO TO 72
      IF(DIM.LE.16.) GO TO 73
      TBMAX=IFIX(DIM/8.0+1.0)*4
      GO TO 74
 73   TBMAX=8
      GO TO 74
 72   IF(DIM.LE.400.) GO TO 73
      TBMAX=IFIX(DIM/200.+1.)*4
 74   IF(NMAX.LE.TBMAX) GO TO 12
      IF(J.GE.TK) GO TO 360
      J=J+1
      GO TO 16
 62   NMAX=NB
      N=IFIX(NMAX/2.+0.51)
 12   SB=AMAX1(1.5*DB(J),SB1)
C
C CHECK MIN. LATERAL RIENFORCEMENT DIAMETER, DSS
C
      CALL REIN11(TCODE,TYPE,DSMIN,TBS,J,DSS,TDSSS)
      DP=DCC+DSS+DB(J)*0.5
      GO TO(17,18,19,20),TRFACE
C
C CHECK NO. OF BARS, BARS SPACING AND REINFORCEMENT RATIO
C
 17   H=CY
      GO TO 21
 18   H=CX
 21   G=H-2.*DP
      B=(N-1.)*(DB(J)+SB)
      IF(NMAX.GE.4) GO TO 22
 67   IF(TOUT.NE.1) GO TO 30
      WRITE(8,200)
      WRITE(8,201) NMAX,TBS(J),SB,DSS
      IF(TMODE.EQ.1) GO TO 670
      GO TO 30
 670  WRITE(7,200)
      WRITE(7,201)NMAX,TBS(J),SB,DSS
 30   IF(TDESIG.EQ.1) GO TO 350
      TREINF=2
      IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,202)
 131  WRITE(8,202)
      RETURN
C
C FOR DESIGN PROBLEM, ASSIGN MIN. NO. OF BARS TO NMAX IF LESS
C THAN THE MIN. ONE
C
 350  GO TO(351,351,352,353),TRFACE
 351  NMAX=4
      N=IFIX(NMAX/2.+0.51)
      GO TO 22
 352  NX=2
      GO TO 60
 353  NMAX=6
      N=IFIX(NMAX/2.+0.51)
      GO TO 354
C
C CHECK BAR SPACING
C
 22   IF(B.LE.G) GO TO 24
 68   IF(TOUT.NE.1) GO TO 33
      WRITE(8,200)
      WRITE(8,201) NMAX,TBS(J),SB,DSS
      IF(TMODE.EQ.1) GO TO 671
      GO TO 33
 671  WRITE(7,200)
      WRITE(7,201)NMAX,TBS(J),SB,DSS
 33   IF(NBS.NE.0) GO TO 360
C
C INCREASE BAR SIZE
C
      IF(J.GE.TK) GO TO 360
```

```
      J=J+1
      GO TO 16
 24   AST=NMAX*A(J)
      PGDES=AST/AG
C
C CHECK REINFORCEMENT RATIO
C
      IF(PGDES.LE.PG) GO TO 26
 360  GO TO(361,362,363,364,365),TDIMFG
 361  IF(TMODE.EQ.1) GO TO 367
      GO TO 368
 367  WRITE(7,369)
 368  WRITE(6,369)
C
C PROCEED TO NEXT PROBLEM RESULTING FROM LIMITING CONSTRAINTS
C
      TREINF=2
      RETURN
 362  CY=CY+CYINCR
      GO TO 366
 363  CX=CX+CXINCR
      GO TO 366
 364  IF(CX.GE.CY) GO TO 3200
      CX=CX+CXINCR
      CY=CX/R
      CY=CYINCR*IFIX(CY/CYINCR)
      GO TO 3201
 3200 CY=CY+CYINCR
      CX=CY*R
      CX=CXINCR*IFIX(CX/CXINCR)
 3201 IF(TSHAPE.EQ.3) GO TO 370
      GO TO 366
 370  AG=(PRIME*CX**2)/4.
      GO TO 371
 365  IF(CX.GE.CY) GO TO 362
      GO TO 363
 366  AG=CX*CY
 371  AS=PMIN*AG
      IF(TOUT.NE.1) GO TO 15
      WRITE(6,3101) CX,CY
      GO TO 15
 26   SB=(G-(N-1.)*DB(J))/(N-1.)
      IF(TOUT.NE.1) GO TO 29
      IF(TMODE.EQ.1) GO TO 2004
      GO TO 134
 2004 WRITE(7,204) NMAX,TBS(J),SB,DSS
 134  WRITE(6,204) NMAX,TBS(J),SB,DSS
      GO TO 29
C
C FOR BARS IN BOTH FACES
C
 19   IF(NBS.NE.0) GO TO 373
C
C COMPUTE NX AND NY
C
      IF(ABS(MX2).GE.ABS(MY2)) GO TO 372
      NX=IFIX((NMAX+4.)/4.+0.51)
      NY=IFIX((NMAX-4.)/4.)
      GO TO 373
 372  NX=IFIX((NMAX+4.)/4.)
      NY=IFIX((NMAX-4.)/4.+0.51)
 373  IF(2*NX-4) 63,60,60
 63   IF(TOUT.NE.1) GO TO 30
      IF(TMODE.EQ.1) WRITE(7,300)
      WRITE(6,300)
      WRITE(6,301) NX,NY,TBS(J),SB,DSS
      IF(TMODE.EQ.1) WRITE(7,301)NX,NY,TBS(J),SB,DSS
      GO TO 30
 60   GX=CY-2.*DP
      BX=(NX-1.)*(DB(J)+SB)
      IF(BX.LE.GX) GO TO 32
      GO TO 374
 32   GY=CX-2.*DP
      BY=(NY+1.)*(DB(J)+SB)
      IF(BY.LE.GY) GO TO 66
 374  IF(TOUT.NE.1) GO TO 33
      WRITE(6,300)
      IF(TMODE.EQ.1) WRITE(7,300)
      WRITE(6,301) NX,NY,TBS(J),SB,DSS
      IF(TMODE.EQ.1) WRITE(7,301)NX,NY,TBS(J),SB,DSS
      GO TO 33
 66   ASX=2.*NX*A(J)
      ASY=2.*NY*A(J)
      AST=ASX+ASY
      PGDES=AST/AG
      IF(PGDES.GT.PG) GO TO 360
      SBX=(GX-(NX-1.)*DB(J))/(NX-1.)
      SBY=(GY-(NY+1.)*DB(J))/(NY+1.)
      NNX=2*NX
      NNY=2*NY
      IF(TOUT.NE.1) GO TO 29
      IF(TMODE.EQ.1) GO TO 2005
      GO TO 135
 2005 WRITE(7,206) NNX,NNY,ASX,ASY,SBX,SBY,TBS(J),DSS
 135  WRITE(6,206) NNX,NNY,ASX,ASY,SBX,SBY,TBS(J),DSS
      GO TO 29
C
C FOR BARS IN CIRCULAR CORE
C
 20   DS=CX-2.*DP
      G=PRIME*DS
      B=NMAX*(DB(J)+SB)
      IF(NMAX.LT.6) GO TO 67
 354  IF(B.GT.G) GO TO 68
      AST=NMAX*A(J)
      PGDES=AST/AG
      IF(PGDES.GT.PG) GO TO 360
      SBBB=G/NMAX
      SB=SBBB-DB(J)
C
C COMPUTE ANGLE BETWEEN BARS
C
      HETA=SBBB/(0.5*DS)
      IF(TOUT.NE.1) GO TO 709
      IF(TMODE.EQ.1) GO TO 2007
      GO TO 137
 2007 WRITE(7,204) NMAX,TBS(J),SB,DSS
      WRITE(7,208) HETA,DS
 137  WRITE(6,204) NMAX,TBS(J),SB,DSS
      WRITE(6,208) HETA,DS
```

```
C
C DETERMINE ISEX IN THIS SUBROUTINE FOR CONVENIENT DATA NEEDED
C
 709  YSUM=O.
      TL=IFIX((NMAX-4.)/4.+0.51)
      DO 35 NI=1,TL
      STA=NI*HETA
      Y(NI)=0.5*DS*SIN(STA)
 35   YSUM=YSUM+Y(NI)**2
C
C DETERMINE WHETHER HAVING CENTER BARS OR NOT
C
      NO=NMAX/2
      NOO=NO/2
      IF(NO/2.-NOO.LE.0.0001) GO TO 88
      NM=0
      GO TO 87
 88   NM=1
 87   ISEX=(AST*(0.5*NM*DS**2+4.*YSUM))/NMAX
      GO TO 36
 29   GAMMAY=(CY-2.*DP)/CY
      GO TO(37,38,39,36),TRFACE
 37   YSUM=O.
      TL=IFIX((NMAX-1.)/4.+0.51)
      DO 40 NI=1,TL
      Y(NI)=0.5*GAMMAY*CY-(NI-1.)*(DB(J)+SB)
 40   YSUM=YSUM+Y(NI+**2
      ISEX=(4.*AST*YSUM)/NMAX
      GO TO 36
 38   ISEX=0.25*AST*GAMMAY**2*CY**2
      GO TO 36
 39   YSUM=O.
      TL=IFIX((2.*NX-1.)/4.+0.51)
      DO 41 NI=1,TL
      Y(NI)=0.5*GAMMAY*CY-(NI-1.)*(SBX+DB(J))
 41   YSUM=YSUM+Y(NI)**2
      ISEX=0.25*ASY*GAMMAY**2*CY**2+(2.*ASX*YSUM)/NX
 36   IF(TBEND.EQ.1) GO TO 47
C
C ALSO DETERMINE ISEY, ONLY USED FOR BIAXIAL BENDING
C
      IF(TRFACE.EQ.4) GO TO 43
      GAMMAX=(CX-2.*DP)/CX
      GO TO(44,45,46,43),TRFACE
 44   ISEY=0.25*AST*GAMMAX**2*CX**2
      GO TO 47
 45   XSUM=O.
      TL=IFIX((NMAX-1.)/4.+0.51)
      DO 48 NI=1,TL
      X(NI)=0.5*GAMMAX*CX-(NI-1.)*(DB(J)+SB)
 48   XSUM=XSUM+X(NI)**2
      ISEY=(4.*AST*XSUM)/NMAX
      GO TO 47
 46   XSUM=O.
      TL=IFIX((2.*NY-1.)/4.+0.51)
      IF(TL.LE.0) TL=1
      DO 49 NI=1,TL
      X(NI)=0.5*GAMMAX*CX-DB(J)-SBY-(NI-1.)*(SBY+DB(J))
 49   XSUM=XSUM+X(NI)**2
      IF(NY.EQ.0) GO TO 999
      ISEY=0.25*ASX*GAMMAX**2*CX**2+(2.*ASY*XSUM)/NY
      GO TO 47
 999  ISEY=0.25*ASX*GAMMAX**2*CX**2
      GO TO 47
 43   XSUM=O.
      TL=IFIX((NMAX-3.)/4.+0.51)
      DO 50 NI=1,TL
      CTA=NI*HETA
      X(NI)=0.5*DS*COS(CTA)
 50   XSUM=XSUM+X(NI)**2
      ISEY=(AST*(0.5*DS**2+4.*XSUM))/NMAX
C
 47   IF(TOUT.NE.1) GO TO 52
      IF(TMODE.EQ.1) GO TO 2009
      GO TO 138
 2009 WRITE(7,211) PGDES
 138  WRITE(8,211) PGDES
C
C RETURN TO MAIN ROUTINE AND CONTINUE EXECUTION
C
 52   TREINF=1
      RETURN
C
C     *****FORMAT STATEMENTS*****
C
 400  FORMAT(' ','SELECTED REINFORCEMENT DETAILS :'/)
 200  FORMAT(' ',1X,' NMAX',2X,'BS(J)',6X,
     * 'SB',6X,'DSS')
 201  FORMAT(I5,2X,I5,2F10.3/)
 202  FORMAT(' ',5X,'THE CODE',
     * ' REQUIREMENT FOR MINIMUM NUMBER'/6X,'OF LONGITUDINAL',
     * ' BARS IS NOT MEET.'/)
 204  FORMAT(' ',2X,'NUMBER',
     *' OF LONGITUDINAL BARS= ',I4/3X,'BAR SIZE= #',I4/,
     * 3X,'CLEAR BAR SPACING= ',F6.2,1X,'IN/MM'/3X,
     *'LATERAL TIES DIAM.= ',F6.3,1X,'IN/MM')
 206  FORMAT(' ',2X,'X-FACE=YES',37X,'Y-FACE=YES',/'NUMBER OF',
     *' LONGITUDINAL BARS= ',I4,8X,'NUMBER OF LONGITUDINAL BARS= ',
     * I4/'STEEL AREA= ',F12.1,1X,'SQ.IN/SQ.MM',17X,'STEEL AREA= ',
     *F12.1,1X,'SQ.IN/SQ.MM'/,'CLEAR BAR SPACING= ',F6.2,1X,'IN/MM',
     * 22X,'CLEAR BAR SPACING= ',F6.2,1X,'IN/MM'/10X,'****',
     *'ADDITIONAL INFORMATION ****'/10X,'BAR SIZE= #',I4/10X,
     * 'LATERAL TIES DIAMETER= ',F6.3,1X,'IN/MM'/)
 208  FORMAT(' ',2X,'ANGLE BETWEEN BARS= ',F7.3,3X,'RADIAN'/3X,
     * 'DIAMETER OF LONGITUDINAL STEEL CIRCLE= ',F7.2,1X,'IN/MM'/)
 211  FORMAT(' ',2X,'PG DESIGN= ',F6.4/)
C
 300  FORMAT(' ',2X,'NX',5X,'NY',4X,'BS(J)',6X,'SB',8X,
     * 'DSS')
 301  FORMAT(I5,2X,I5,2X,I5,2F10.3/)
 369  FORMAT(' ',5X,'DESIGN CANNOT BE COMPLETED COMPLETED'/6X,
     * 'CONSISTENT WITH THE ENTERED RESTRAINTS'/)
      END
C
C  ***********************************************************
C  *                                                         *
      SUBROUTINE REIN11(TCODE,TYPE,DSMIN,TBS,J,DSS,TDSSS)
C  *                                                         *
C  *THIS SUBROUTINE CHECKS MINIMUM LATERAL REINFORCEMENT DIAMETER *
C  *                                                         *
```

```
C   *************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      DIMENSION TBS(9)
C
      IF(TYPE.EQ.2) GO TO 10
      IF(TCODE.EQ.1) GO TO 11
      IF(TBS(J).LE.10) GO TO 13
      IF(DSS.GT.0.5) GO TO 13
      DSS=0.5
      TDSSS=4
      GO TO 13
 11   IF(TBS(J).LE.30) GO TO 14
      IF(DSS.GT.11.3) GO TO 13
      DSS=11.3
      TDSSS=10
      GO TO 13
 14   IF(DSS.GT.0.30*TBS(J)) GO TO 13
      DSS=11.3
      TDSSS=10
      GO TO 13
 10   IF(DSS.GT.DSMIN) GO TO 13
      DSS=DSMIN
      IF(TCODE.EQ.1) TDSSS=6
      IF(TCODE.EQ.2) TDSSS=4
 13   RETURN
      END
C
C   *************************************************************
C   *                                                           *
      SUBROUTINE SLEND1
C   *                                                           *
C   *THIS SUBROUTINE CHECKS SLENDERNESS EFFECTS ABOUT X-AXIS
C   *                                                           *
C   *************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      IF(TOUT.NE.1) GO TO 3000
      IF(TMODE.EQ.1) WRITE(7,500)
      WRITE(8,500)
 500  FORMAT(' ','CHECK SLENDERNESS EFFECTS BENDING',
     * ' ABOUT X-AXIS :'/)
C
C COMPUTE MOMENT OF INERTIA OF CROSS-SECTION AND RADIUS OF GYRATION
C AND SLENDERNESS EFFECTS ABOUT X-AXIS
C
 3000 IGX=CG*AG*CY**2
      RXT=IGX/AG
      RX=SQRT(RXT)
      SX=KX*LUX/RX
      IF(TXBRAC.EQ.1) GO TO 10
      IF(SX.LE.22.) GO TO 11
      IF(SPU.LE.0..AND.SPC.LE.0.) GO TO 12
      IF(PHIV*SPC.LE.SPU) GO TO 350
      CMX=1.0
      DELTAX=CMX/(1.-(SPU/(PHIV*SPC)))
      IF(DELTAX.LE.1.0) DELTAX=1.0
      GO TO 13
 11   DELTAX=1.0
      GO TO 50
C
C PROCEED TO NEXT PROBLEM RESULTING FROM INCOMPLETE INPUT PARAMETERS
C
 12   TSLEN1=2
      IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,201)
 131  WRITE(8,201)
      RETURN
 350  TSLEN1=2
      IF(TMODE.EQ.1) WRITE(7,351)
      WRITE(8,351)
      RETURN
 10   IF(ABS(MX2).LE.0.) GO TO 610
      MX12=ABS(MX1/MX2)
      GO TO 611
 610  MX12=1.0
 611  IF(SX.LT.34.-12.*MX12) GO TO 14
      IF(TOUT.NE.1) GO TO 3001
      IF(TMODE.EQ.1) GO TO 2002
      GO TO 132
 2002 WRITE(7,202)
 132  WRITE(8,202)
C
C COMPUTE EI VALUE AND CRITICAL LOAD
C   .
 3001 EIX=(0.2*EC*IGX+ES*ISEX)/(1.+BDX)
      PCX=(PRIME**2*EIX)/(KX*LUX)**2
      IF((PHIV*PCX).GT.PU) GO TO 40
      IF(TOUT.NE.1) GO TO 3002
      IF(TMODE.EQ.1) GO TO 2003
      GO TO 133
 2003 WRITE(7,400)
 133  WRITE(8,400)
C
C REDESIGN CONCRETE SECTION DUE TO STABILITY PROBLEM BY USING
C SEVERAL DESIGN CODE EQUATIONS AND THE FORMULA IG=CG*AG*H
C
 3002 IF(TDIMFG.EQ.1) GO TO 41
C
C SET PCX=PU/PHIV AND USE THE CONSERVATIVE CODE
C EQUATION OF EI FOR SLENDER COLUMN
C
```

```
      IGX=(PU/PHIV)*(KX*LUX/PRIME)**2*(1.+BDX)/(0.4*EC)
      GO TO (41,42,43,44,45),TDIMFG
 41   IF(TMODE.EQ.1) GO TO 2004
      GO TO 134
 2004 WRITE(7,401)
 134  WRITE(8,401)
C
C PROCEED TO NEXT PPOBLEM RESULTING FROM CONSTANT DIMENSIONS
C
      TSLEN1=2
      RETURN
C
C COMPARING NEW AREA WITH THE OLD ONE
C
 42   AGNEW=(IGX*CX**2/CG)**0.333333
      IF(AGNEW.GT.AG) GO TO 5000
C
C IF OLD AREA > NEW COMPUTED ONE, INC. DIMENSION W.R.T.
C ENTERED DIMENSIONS
C
      CY=CY+CYINCR
      GO TO 61
 5000 AG=AGNEW
      CY=AG/CX
      CY=CYINCR*IFIX(CY/CYINCR+0.999)
      IF(CY.LE.MINDIM) GO TO 60
      GO TO 61
 60   CY=MINDIM
      IF(TOUT.NE.1) GO TO 61
      IF(TMODE.EQ.1) GO TO 2005
      GO TO 135
 2005 WRITE(7,300)
 135  WRITE(8,300)
 61   AG=CX*CY
      GO TO 80
 43   AGNEW=IGX/(CG*CY**2)
      IF(AGNEW.GT.AG) GO TO 5001
      CX=CX+CXINCR
      GO TO 63
 5001 AG=AGNEW
      CX=AG/CY
      CX=CXINCR*IFIX(CX/CXINCR+0.999)
      IF(CX.LE.MINDIM) GO TO 62
      GO TO 63
 62   CX=MINDIM
      IF(TOUT.NE.1) GO TO 63
      IF(TMODE.EQ.1) GO TO 2006
      GO TO 136
 2006 WRITE(7,301)
 136  WRITE(8,301)
 63   AG=CX*CY
      GO TO 80
 44   IF(TSHAPE.EQ.3) GO TO 47
      A1=R*IGX/CG
      AGNEW=SQRT(A1)
      IF(AGNEW.GT.AG) GO TO 5002
      IF(CX.GT.CY) GO TO 5003
      CX=CX+CXINCR
      GO TO 65
 5003 CY=CY+CYINCR
      CX=CY*R
      CX=CXINCR*IFIX(CX/CXINCR+0.999)
      GO TO 67
 5002 AG=AGNEW
      A2=AG*R
      CX=SQRT(A2)
      CX=CXINCR*IFIX(CX/CXINCR+0.999)
      IF(CX.LE.MINDIM) GO TO 64
      GO TO 65
 64   CX=MINDIM
      IF(TOUT.NE.1) GO TO 65
      IF(TMODE.EQ.1) GO TO 2007
      GO TO 137
 2007 WRITE(7,301)
 137  WRITE(8,301)
 65   CY=CX/R
      CY=CYINCR*IFIX(CY/CYINCR+0.999)
      IF(CY.LE.MINDIM) GO TO 66
      GO TO 67
 66   CY=MINDIM
      IF(TOUT.NE.1) GO TO 67
      IF(TMODE.EQ.1) GO TO 2008
      GO TO 138
 2008 WRITE(7,300)
 138  WRITE(8,300)
 67   AG=CX*CY
      GO TO 80
 47   AG1=(PRIME*IGX)/(4.*CG)
      AGNEW=SQRT(AG1)
      IF(AGNEW.GT.AG) GO TO 5004
      CX=CX+CXINCR
      GO TO 69
 5004 AG=AGNEW
      A1=4.*AG/PRIME
      CX=SQRT(A1)
      CX=CXINCR*IFIX(CX/CXINCR+0.999)
      IF(CX.LE.MINDIM) GO TO 68
      GO TO 69
 68   CX=MINDIM
      IF(TOUT.NE.1) GO TO 69
      IF(TMODE.EQ.1) GO TO 2009
      GO TO 139
 2009 WRITE(7,301)
 139  WRITE(8,301)
 69   CY=CX
      AG=(PRIME*CX**2)/4.
      GO TO 80
 45   IF(TBEND.EQ.1) GO TO 81
      GO TO 82
 81   AGNEW=IGX/(CG*CY**2)
      IF(AGNEW.GT.AG) GO TO 5005
      CX=CX+CXINCR
      GO TO 73
 5005 AG=AGNEW
      CX=AG/CY
      CX=CXINCR*IFIX(CX/CXINCR+0.999)
      IF(CX.LE.MINDIM) GO TO 87
      GO TO 73
 87   CX=MINDIM
      IF(TOUT.NE.1) GO TO 73
```

```
      IF(TMODE.EQ.1) GO TO 2010
      GO TO 140
2010 WRITE(7,301)
140  WRITE(8,301)
     GO TO 73
82   A1=R*IGX/CG
     AGNEW=SQRT(A1)
     IF(AGNEW.GT.AG) GO TO 5006
     IF(CX.GT.CY) GO TO 5007
     CX=CX+CXINCR
     GO TO 71
5007 CY=CY+CYINCR
     CX=CY*R
     CX=CXINCR*IFIX(CX/CXINCR+0.999)
     GO TO 73
5006 AG=AGNEW
     A2=R*AG
     CX=SQRT(A2)
     CX=CXINCR*IFIX(CX/CXINCR+0.999)
     IF(CX.LE.MINDIM) GO TO 70
     GO TO 71
70   CX=MINDIM
     IF(TOUT.NE.1) GO TO 71
     IF(TMODE.EQ.1) GO TO 2011
     GO TO 141
2011 WRITE(7,301)
141  WRITE(8,301)
71   CY=CX/R
     CY=CYINCR*IFIX(CY/CYINCR+0.999)
     IF(CY.LE.MINDIM) GO TO 72
     GO TO 73
72   CY=MINDIM
     IF(TOUT.NE.1) GO TO 73
     IF(TMODE.EQ.1) GO TO 2012
     GO TO 142
2012 WRITE(7,300)
142  WRITE(8,300)
73   AG=CX*CY
80   TMODAS=0
     AS=PMIN*AG
     IF(TOUT.NE.1) GO TO 3005
     IF(TSHAPE.EQ.3) GO TO 2880
     IF(TMODE.EQ.1) GO TO 2014
     GO TO 144
2014 WRITE(7,403) CX,CY
144  WRITE(8,403) CX,CY
     GO TO 3005
2880 IF(TMODE.EQ.1) WRITE(7,2881) CX
2881 FORMAT(' ','NEW TRIAL CONCRETE DIM. :'/8X,'DIAMETER =',
    * F6.1,1X,'IN/MM'/)
     WRITE(8,2881) CX
C
C BACK TO SUBROUTINE REINF1 AND REITERATE THE STEPS
C
3005 TSLEN1=3
     RETURN
C
C DETERMINE THE GOVERNING ECCENTRICITY, COMPUTE END MOMENT MUX AND
C MAGNIFICATION FACTOR DELTAX FOR LONG COLUMNS
C
40   IF(ABS(MX2).LE.0.1) GO TO 15
     B=MX1/MX2
     GO TO 20
15   B=1.0
20   IF(TCODE.EQ.1) GO TO 16
     EMINY=0.6+0.03*CY
     GO TO 17
16   IF(TYPE.EQ.1) GO TO 18
     F=0.05*CY
     GO TO 19
18   F=0.1*CY
19   EMINY=AMAX1(F,EMIN)
17   EY=ABS(MX2)/PU
     IF(EY.LE.EMINY) GO TO 510
     GO TO 511
510  EY=EMINY
     IF(TCODE.EQ.1) MUXX=(PU*EY)/1000000.
     IF(TCODE.EQ.2) MUXX=(PU*EY)/12000.
     IF(TMODE.EQ.1) GO TO 550
     GO TO 551
550  WRITE(7,512) MUXX
551  WRITE(8,512) MUXX
511  MUX=PU*EY
     CMX=0.6+0.4*B
     IF(CMX.LT.0.4) CMX=0.4
     DELTAX=CMX/(1.-PU/(PHIV*PCX))
C IF DELTAX IS LESS THAN 1.0, IT WOULD BE DEFAULTS TO 1.0
     IF(DELTAX.LE.1.0) DELTAX=1.0
     GO TO 513
C
C DETERMINE THE GOVERNING ECCENTRICITY AND COMPUTE END MOMENT MUX
C FOR SHORT COLUMNS
14   IF(TOUT.NE.1) GO TO 3007
     IF(TMODE.EQ.1) GO TO 2015
     GO TO 145
2015 WRITE(7,203)
145  WRITE(8,203)
3007 DELTAX=1.0
     GO TO 50
13   IF(TCODE.EQ.1) GO TO 50
     EMINY=0.6+0.03*CY
     GO TO 51
50   IF(TYPE.EQ.1) GO TO 52
     F=0.05*CY
     GO TO 53
52   F=0.1*CY
53   EMINY=AMAX1(F,EMIN)
51   EY=ABS(MX2)/PU
     IF(EY.LE.EMINY) GO TO 514
     GO TO 516
514  EY=EMINY
     MUX=PU*EY
     IF(TCODE.EQ.1) MUXX=MUX/1000000.
     IF(TCODE.EQ.2) MUXX=MUX/12000.
     IF(TMODE.EQ.1) GO TO 552
     GO TO 553
552  WRITE(7,512) MUXX
553  WRITE(8,512) MUXX
516  MUX=PU*EY
513  IF(DELTAX.LE.1.001) GO TO 520
```

```
      WRITE(8,204)DELTAX
      IF(TMODE.EQ.1) WRITE(7,205)DELTAX
C
C COMPUTE MAGNIFIED MOMENT THAT APPLIED MOMENT
C
 620  MCX=DELTAX*MUX
      EY=MCX/PU
      MMINX=PU*EMINY
      MAPPX=MCX
C
C RETURN TO MAIN ROUTINE AND CONTINUE EXECUTION
C
      TSLEN1=1
      RETURN
C
C     *****FORMAT STATEMENTS*****
C
 201  FORMAT(' ',5X,'THIS IS A LONG UNBRACED FRAME. MORE INPUT',
     */6X,'PARAMETERS WILL BE REQUIRED TO COMPUTE'/6X,
     * 'MOMENT MAGNIFIER ABOUT X-AXIS ONLY'/)
 202  FORMAT(' ',5X,'SLENDERNESS EFFECT MUST BE CONSIDERED'/6X,
     * 'BY MAGNIFYING MOMENT ABOUT X-AXIS'/)
 203  FORMAT(' ',5X,'SLENDERNESS EFFECT MAY BE NEGLECTED'/)
 400  FORMAT(' ',5X,'THE CROSS SECTION OF COLUMN IS TOO SMALL',
     */6X,'TO PROVIDE STABILITY'/)
 401  FORMAT(' ',5X,'COLUMN DIMENSIONS ARE CONSTANT AND THE'/6X,
     * 'SECTION IS NOT ADEQUATE TO PROVIDE STABILITY'/)
 403  FORMAT(' ','NEW TRIAL CONCRETE DIM. : '/6X,'CX =',F6.1,1X,
     *'IN/MM',8X,'CY =',F6.1,1X,'IN/MM'/)
 300  FORMAT(' ',5X,'MINIMUM DIMENSION CY IS USED'/)
 301  FORMAT(' ',5X,'MINIMUM DIMENSION CX IS USED'/)
 512  FORMAT(' ',5X,'MINIMUM ECCENTRICITY ABOUT X-AXIS GOVERNS',
     * /6X,'DESIGN MOMENT ABOUT X-AXIS = ',F8.0/6X,'IN PROJECT',
     * ' UNIT'/)
 204  FORMAT(' ',5X,'MAGNIFICATION FACTOR FOR MOMENT ABOUT',
     * ' X-AXIS =',F5.2/)
 205  FORMAT(' ','COMMENTS :'//' ',5X,'MAGNIFICATION FACTOR',
     * ' FOR MOMENT ABOUT X-AXIS =',F5.2/)
 351  FORMAT(' ',5X,'SUM OF CRITICAL LOAD IS LESS THAN'/6X,
     * 'SUM OF NOMINAL AXIAL LOAD'/)
C
      END
C
C  ***********************************************************
C  *                                                         *
      SUBROUTINE CAPAC1
C  *                                                         *
C  *THIS SUBROUTINE INVESTIGATES STRENGTH CAPACITY IN        *
C  *COMPRESSION CONTROLS REGIONS ( PNX > PBX )                *
C  *                                                         *
C  ***********************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      IF(TOUT.NE.1) GO TO 4000
      IF(TMODE.EQ.1) WRITE(7,300)
      WRITE(8,300)
 300  FORMAT(' ','CAPACITY CHECK FOR BENDING ABOUT X-AXIS : '/)
 4000 IF(TRFACE.EQ.2) GO TO 10
C
C COMPUTE STRENGTH CAPACITIES WITH BARS IN LATERAL FACES OR
C IN BOTH FACES OR IN CIRCULAR CORE
C
      CALL CAPA12
      GO TO 11
C
C COMPUTE STRENGTH CAPACITIES WITH BARS IN END FACES ONLY
C
 10   CALL CAPA11
 11   IF(TOUT.NE.1) RETURN
      PUDX=PUDESX/1000.
      MPBX=PBX/1000.
      IF(TCODE.EQ.1) GO TO 1990
      MUDX=MUDESX/12000.
      GO TO 1991
 1990 MUDX=MUDESX/1000000.
 1991 WRITE(8,200) YC,YB,PUDX,MUDX,MPBX
      IF(TMODE.EQ.1) WRITE(7,200)YC,YB,PUDX,MUDX,MPBX
      RETURN
C
 200 `FORMAT(' ',2X,'LOCATION OF NEUTRAL AXIS // X-AXIS= ',
     * F8.1,1X,'IN/MM'/3X,'NEUTRAL AXIS AT BALANCED CONDITION= ',
     *F8.1,1X,'IN/MM'/3X,'DESIGN AXIAL',
     * ' LOAD WITH ONLY X-AXIS BENDING= ',F8.0,1X,'KIPS/KN'/3X,
     * 'DESIGN MOMENT ABOUT X-AXIS ONLY= ',F8.0,1X,'KIP-FT/KN-M'/3X,
     * 'LOAD AT BALANCED STRAIN CONDITION= ',F8.0,1X,'KIPS/KN'/)
C
      END
C
C  ***********************************************************
C  *                                                         *
      SUBROUTINE CAPA11
C  *                                                         *
C  *THIS SUBROUTINE COMPUTES LOCATION OF NEUTRAL AXIS AND    *
C  *DESIGN STRENGTH AT COMPRESSION CONTROL REGION AND AT     *
C  *BALANCED CONDITIONS FOR BARS DISTRIBUTED IN END FACES ONLY *
C  *                                                         *
C  ***********************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
```

```
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      D=CY-DP
C
C COMPUTE COMPRSSSION FORCE IN STEEL
C
      CSS=0.5*AST*(FY-0.85*FPC)
C
C USING HALFING INTERVAL TECHNIQUE TO FIND LOCATION OF NEUTRAL
C AXIS
C
      YL=(ECU*D)/(ECU+ESMAX)
      YU=CY/B1
 10   YC=(YL+YU)/2.
C
C COMPUTE FORCE IN TENSION REINFORCEMENT AND COMPRESSION FORCE
C IN CONCRETE
C
      FTT=0.5*AST*ECU*ES*((D-YC)/YC)
      CC=0.85*FPC*B1*YC*CX
      YY=YU-YL
      IF(TCODE.EQ.1) DIFFER=5
      IF(TCODE.EQ.2) DIFFER=0.2
      IF(ABS(YY).LT.DIFFER) GO TO 11
      FUN=F11(YC,FTT,CSS,CC,D,CY,EY,DP,B1)
      IF(FUN) 12,11,14
 12   YU=YC
      GO TO 10
 14   YL=YC
      GO TO 10
 11   IF(B1*YC.GT.D) GO TO 20
      GO TO 21
C
C THE EFFECT OF DISPLACED CONCRETE IS INCLUDED
C
 20   FTT=FTT+0.5*AST*0.85*FPC
 21   PNX=CC+CSS-FTT
      MNX=FTT*(D-0.5*CY)+CSS*(0.5*CY-DP)+CC*(0.5*CY-0.5*B1*YC)
      PUDESX=PHIV*PNX
      MUDESX=PHIV*MNX
C
      YB=(ECU*D)/(FY/ES+ECU)
      PBX=PHIV*0.85*FPC*(B1*YB*CX-0.5*AST)
      RETURN
      END
C
C
      REAL FUNCTION F11(YC,FTT,CSS,CC,D,CY,EY,DP,B1)
C
C TAKING MOMENTS OF THE INTERNAL FORCES ABOUT APPLIED AXIAL LOAD
C AND WRITING AN EQUATION IN TERMS OF FUNCTION OF UNKNOWN NEUTRAL
C AXIS
C
      F11=FTT*(D-(0.5*CY-EY))+CSS*(0.5*CY-EY-DP)+CC*(0.5*CY-EY
     *    -0.5*B1*YC)
      RETURN
      END
C
C  **************************************************************
C  *                                                            *
      SUBROUTINE CAPA12
C  *                                                            *
C  *THIS SUBROUTINE COMPUTES LOCATION OF NEUTRAL AXIS AND       *
C  *DESIGN STRENGTH AT THE COMPRESSION CONTROL REGION AND       *
C  *AT BALANCED CONDITION FOR BARS DISTRIBUTED IN LATERAL       *
C  *FACES OR IN BOTH FACES OR IN CIRCULAR CORE                  *
C  *                                                            *
C  **************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      D=CY-DP
C
C DETERMINE BALANCED AXIAL LOAD BY DIRECT APPLICATION OF STATICS
C
      YB=(ECU*D)/(ECU+ESMAX)
      AB=B1*YB
C
C DETERMINE CONTROL PARAMETERS IN COUNTER DO LOOP
C
      GO TO (10,11,12,13),TRFACE
 10   TL=IFIX((NMAX-1.)/4.+0.51)
      GO TO 14
 12   TL=IFIX((2.*NX-1.)/4.+0.51)
      GO TO 14
 13   TL=IFIX((NMAX-4.)/4.+0.51)
C
C DETERMINE STRAINS AT THE VARIOUS BARS AND EVALUATE THEM
C WHETHER THEY YIELD OR NOT
C
 14   DO 15 NI=1,TL
      ECS(NI)=((YB-0.5*CY+Y(NI))*ECU)/YB
      IF(ABS(ECS(NI)).GT.ESMAX) GO TO 80
```

```
      GO TO 81
 80   ECS(NI)=(ESMAX*ECS(NI))/ABS(ECS(NI))
 81   ETS(NI)=((Y(NI)+0.5*CY-YB)*ESMAX)/(D-YB)
      IF(ABS(ETS(NI)).GT.ESMAX) GO TO 82
      GO TO 15
 82   ETS(NI)=(ESMAX*ETS(NI))/ABS(ETS(NI))
 15   CONTINUE
C
C CHECK WHETHER SECTION HAVING EXTREME COMPRESSION BARS AND EXTREME
C TENSION BARS AND CENTER BARS OR NOT HAVING THEM
C
      GO TO (16,11,17,18),TRFACE
 16   NO=NMAX/2
      NOO=NO/2
C
C FOR BARS IN LATERAL FACES, CONSIDERING NO EXTREME BARS IN
C COMPRESSION AND IN TENSION
C
      ECSR=0.
      ETSL=0.
      GO TO 19
 17   NO=NX
      NOO=NO/2
      GO TO 20
 18   NO=NMAX/2
      NOO=NO/2
      IF(NO/2.-NOO.LE.0.0001) GO TO 20
      GO TO 21
 20   ECSR=((YB-DP)*ECU)/YB
      IF(ABS(ECSR).GT.ESMAX) GO TO 83
      GO TO 84
 83   ECSR=(ESMAX*ECSR)/ABS(ECSR)
 84   ETSL=ESMAX
      GO TO 19
 21   ECSR=0.
      ETSL=0.
 19   IF(TRFACE.EQ.4) GO TO 22
      IF(NO/2.-NOO.LE.0.0001) GO TO 23
 22   ESC=((YB-0.5*CY)*ECU)/YB
      IF(ABS(ESC).GT.ESMAX) GO TO 85
      GO TO 24
 85   ESC=(ESMAX*ESC)/ABS(ESC)
      GO TO 24
 23   ESC=0.
C
C CALCULATE FORCES IN SECTION AT BALANCED CONDITION
C
 24   GO TO (25,11,26,27),TRFACE
C
C BASED ON ECSR=0 AND ETSL=0
C
 25   CSR=0.
      FTL=0.
      GO TO 28
C
C COMPUTE FORCES IN EXTREME COMPRESSION AND IN EXTREME TENSION
C
 26   CSR=(NY*PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(NY*PRIME*DB(J)**2*ES*ETSL)/4.
C
C COMPUTE FORCES IN CONCRETE AND IN CENTER BARS
C
 28   CC=0.85*FPC*AB*CX
      IF(ESC.LE.0.) GO TO 29
      CSC=(2.*PRIME*DB(J)**2*(ES*ESC-0.85*FPC))/4.
      GO TO 30
 29   CSC=0.
      GO TO 30
C
C DETERMINE AREA OF SEGMENT FOR A CIRCULAR REINFORCEMENT PATTERN
C
 27   RR=0.5*CY
      IF(AB.LE.RR) GO TO 31
      PP=AB-RR
      GO TO 32
 31   PP=RR-AB
 32   O=RR**2-PP**2
      OO=SQRT(O)
      PR=PP/RR
      RFAD=ARCOS(PR)
      IF(AB.GT.RR) GO TO 86
      GO TO 87
 86   RFAD=PRIME-RFAD
 87   AREA=(CY**2*(RFAD-SIN(RFAD)*COS(RFAD)))/4.
      CC=0.85*FPC*AREA
      IF(ECSR.LE.0.) GO TO 33
      CSR=(PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(PRIME*DB(J)**2*ES*ETSL)/4.
      GO TO 34
 33   CSR=0.
      FTL=0.
 34   CSC=(2.*PRIME*DB(J)**2*(ES*ESC-0.85*FPC))/4.
C
C COMPUTE FORCES WITH BARS IN VARIOUS COMPRESSION AND TENSION SIDE
C
 30   CSSUM=0.
      FTSUM=0.
      DO 35 NI=1,TL
      CS(NI)=(2.*PRIME*DB(J)**2*(ES*ECS(NI)-0.85*FPC))/4.
      CSSUM=CSSUM+CS(NI)
      FT(NI)=(2.*PRIME*DB(J)**2*ES*ETS(NI))/4.
      FTSUM=FTSUM+FT(NI)
 35   CONTINUE
C
C CHECK TO SEE ALL FT(NI) AND FTL ARE ACTUALLY A TENSION, THAT IS
C WHETHER EFFECT OF THE DISPLACED CONCRETE FTP IS INCLUDED OR NOT
C
      NII=1
 36   FTP=((TL-NII+1.)*2.*PRIME*DB(J)**2*0.85*FPC)/4.
      IF(NII.GT.TL) GO TO 37
      IF(FT(NII).LT.0.) GO TO 37
      NII=NII+1
      GO TO 36
 37   IF(FTL.GE.0.) GO TO 38
      IF(TRFACE.EQ.4) GO TO 39
      FTL=FTL+(NY*PRIME*DB(J)**2*0.85*FPC)/4.
      GO TO 38
 39   FTL=FTL+(PRIME*DB(J)**2*0.85*FPC)/4.
 38   PBX=PHIV*(CC+CSR+CSC+CSSUM-FTL-FTSUM-FTP)
C
```

```
C COMPUTE DESIGN STRENGTH AT COMPRESSSION CONTROL REGION.
C APPLICATION MORE OR LESS IS SAME AS ABOVE.
C USING HALFING INTERVAL TECHNIQUE TO FIND LOCATION OF NEUTRAL
C AXIS
C
      YL=YB
      YU=CY/B1
 40   YC=(YL+YU)/2.
      YY=YU-YL
      IF(TCODE.EQ.1) DIFFER=5
      IF(TCODE.EQ.2) DIFFER=0.2
      IF(ABS(YY).LT.DIFFER) GO TO 41
      AC=B1*YC
C
C COMPUTE STRAINS IN THE STEEL
C
      DO 42 NI=1,TL
      ECS(NI)=((YC-0.5*CY+Y(NI))*ECU)/YC
      IF(ABS(ECS(NI)).GT.ESMAX) GO TO 88
      GO TO 89
 88   ECS(NI)=(ESMAX*ECS(NI))/ABS(ECS(NI))
 89   ETS(NI)=((Y(NI)+0.5*CY-YC)*ECU)/YC
      IF(ABS(ETS(NI)).GT.ESMAX) GO TO 90
      GO TO 42
 90   ETS(NI)=(ESMAX*ETS(NI))/ABS(ETS(NI))
 42   CONTINUE
C
C CHECK ECSR AND ETSL AND ESC EQUAL TO ZERO OR NOT
C
      GO TO (43,11,44,45),TRFACE
 45   IF(NO/2.-NOO.LE.0.0001) GO TO 44
      GO TO 46
 44   ECSR=((YC-DP)*ECU)/YC
      IF(ABS(ECSR).GT.ESMAX) GO TO 91
      GO TO 92
 91   ECSR=(ESMAX*ECSR)/ABS(ECSR)
 92   ETSL=((D-YC)*ECU)/YC
      IF(ABS(ETSL).GT.ESMAX) GO TO 93
      GO TO 43
 93   ETSL=(ESMAX*ETSL)/ABS(ETSL)
      GO TO 43
 46   ECSR=0.
      ETSL=0.
 43   IF(TRFACE.EQ.4) GO TO 47
      IF(NO/2.-NOO.LE.0.0001) GO TO 48
 47   ESC=((YC-0.5*CY)*ECU)/YC
      IF(ABS(ESC).GT.ESMAX) GO TO 94
      GO TO 49
 94   ESC=(ESMAX*ESC)/ABS(ESC)
      GO TO 49
 48   ESC=0.
C
C CALCULATE FORCES IN SECTION
C
 49   GO TO (50,11,51,52),TRFACE
 50   CSR=0.
      FTL=0.
      GO TO 53
 51   CSR=(NY*PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(NY*PRIME*DB(J)**2*ES*ETSL)/4.
 53   CC=0.85*FPC*AC*CX
C
C COMPUTE POINT OF APPLICATION FOR CC
C
      YCC=CY/2.-0.5*AC
      IF(ESC.LE.0.) GO TO 54
      CSC=(2.*PRIME*DB(J)**2*(ES*ESC-0.85*FPC))/4.
      GO TO 55
 54   CSC=0.
      GO TO 55
 52   IF(AC.LE.RR) GO TO 56
      PP=AC-RR
      GO TO 57
 56   PP=RR-AC
 57   Q=RR**2-PP**2
      QO=SQRT(Q)
      PR=PP/RR
      RFAD=ARCOS(PR)
      IF(AC.GT.RR) GO TO 95
      GO TO 96
 95   RFAD=PRIME-RFAD
 96   AREA=(CY**2*(RFAD-SIN(RFAD)*COS(RFAD)))/4.
      SEG=(CY**3*(SIN(RFAD))**3)/12.
C
C COMPUTE POINT OF APPLICATION FOR CC IN CIRCULAR BARS PATTERN
C
      YCC=SEG/AREA
      CC=0.85*FPC*AREA
      IF(ECSR.LE.0.) GO TO 58
      CSR=(PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(PRIME*DB(J)**2*ES*ETSL)/4.
      GO TO 59
 58   CSR=0.
      FTL=0.
 59   CSC=(2.*PRIME*DB(J)**2*(ES*ESC-0.85*FPC))/4.
 55   DO 60 NI=1,TL
      CS(NI)=(2.*PRIME*DB(J)**2*(ES*ECS(NI)-0.85*FPC))/4.
      FT(NI)=(2.*PRIME*DB(J)**2*ES*ETS(NI))/4.
 60   CONTINUE
C
C CHECK TO SEE ALL FT(NI) AND FTL ARE ACTUALLY A TENSION
C
      NI=1
 61   IF(FT(NI).GT.0.) GO TO 62
      FT(NI)=FT(NI)+(2.*PRIME*DB(J)**2*0.85*FPC)/4.
      NI=NI+1
      IF(NI.LE.TL) GO TO 61
 62   IF(FTL.GE.0.) GO TO 63
      IF(TRFACE.EQ.4) GO TO 64
      FTL=FTL+(NY*PRIME*DB(J)**2*0.85*FPC)/4.
      GO TO 63
 64   FTL=FTL+(PRIME*DB(J)**2*0.85*FPC)/4.
 63   FUN=F12(FT,CS,Y,FTL,CY,DP,EY,TL,CSR,CSC,CC,YCC)
      IF(FUN) 65,41,66
 65   YU=YC
      GO TO 40
 66   YL=YC
      GO TO 40
 41   FTSUM=0.
      CSSUM=0.
```

```
      DO 87 NI=1,TL
      FTSUM=FTSUM+FT(NI)
      CSSUM=CSSUM+CS(NI)
 87   CONTINUE
      PNX=CSR+CSSUM+CSC+CC-FTL-FTSUM
      FTSUM1=0.
      CSUM1=0.
      DO 100 NI=1,TL
      FTSUM1=FTSUM1+FT(NI)*Y(NI)
      CSUM1=CSUM1+CS(NI)*Y(NI)
 100  CONTINUE
      MNX=FTL*(0.5*CY-DP)+CSR*(0.5*CY-DP)+CC*YCC+FTSUM1+CSUM1
      PUDESX=PHIV*PNX
      MUDESX=PHIV*MNX
 11   RETURN
      END
C
C
      FUNCTION F12(FT,CS,Y,FTL,CY,DP,EY,TL,CSR,CSC,CC,YCC)
C
C TAKING MOMENTS OF THE INTERNAL FORCES ABOUT APPLIED AXIAL
C LOAD AND WRITING AN EQUATION IN TERMS OF FUNCTION OF UNKNOWN
C NEUTRAL AXIS
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      DIMENSION FT(12),CS(12),Y(12)
      FTSUM=0.
      CSUM=0.
      IF(TL.LE.0) GO TO 100
      DO 10 NI=1,TL
      FTSUM=FTSUM+FT(NI)*(Y(NI)+EY)
      CSUM=CSUM+CS(NI)*(Y(NI)-EY)
 10   CONTINUE
 100  F12=FTSUM+CSUM+FTL*(0.5*CY-DP+EY)+CSR*(0.5*CY-DP-EY)-CSC*EY
     *      -CC*(EY-YCC)
      RETURN
      END
C
C  ************************************************************
C  *                                                          *
      SUBROUTINE ADEQU1
C  *                                                          *
C  *THIS SUBROUTINE ADEQUACY CHECKS DESIGN STRENGTH CAPACITY  *
C  *WITH REQUIRED STRENGTH                                    *
C  *                                                          *
C  ************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
C DETERMINE WHETHER CAPACITY REDUCTION FACTOR NEEDS TO BE
C MODIFIED
C
      PDA=0.1*FPC*AG
      IF(PU.GE.PDA) GO TO 50
      PBXX=PBX/PHIV
      PDASH=AMIN1(PDA,PBXX)
C
C NEW PHIV VALUE
      PHIVN=PHIV+((0.9-PHIV)*(PDASH-PU))/PDASH
      IF(PHIVN.LT.PHIV) PHIVN=PHIV
      PBX=PBXX*PHIVN
      PUDESX=PUDESX*PHIVN/PHIV
      MUDESX=MUDESX*PHIVN/PHIV
 50   IF(TOUT.NE.1) GO TO 10
      IF(TMODE.EQ.1) WRITE(7,400)
      WRITE(6,400)
 400  FORMAT(' ','ADEQUACY CHECK FOR UNI-AXIAL BENDING : '/)
C
C ADEQUACY CHECK DESIGN AXIAL LOAD
C
 10   RATIOP=PUDESX/PU
      MOREA=RATIOP-1.0
      IF(PUDESX.GE.PU) GO TO 20
C
C SET TOLERANCE LIMIT OF 0.03 FOR COMPARING PROVIDED AXIAL
C LOAD AND MOMENT WITH APPLIED AXIAL LOAD AND MOMENT
C
      IF(MOREA+0.03.GE.0.) GO TO 20
      IF(TDESIG.EQ.1) GO TO 19
 133  IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,206)RATIOP
 131  WRITE(6,206)RATIOP
      IF(MOREA+0.03.GE.0.) GO TO 2880
      IF(TMODE.EQ.1) WRITE(7,2881)
      WRITE(6,2881)
 2881 FORMAT(' ',5X,'** SECTION IS NOT ADEQUATE FOR THIS',
     * ' LOADING **'/)
C
C PROCEED TO NEXT PROBLEM RESULTING FROM UNSATISFACTORY SECTION
C
 2880 TADEQ=1
      RETURN
 19   IF(TMODE.EQ.1) GO TO 2002
      GO TO 132
 2002 IF(TOUT.NE.1) GO TO 132
      WRITE(7,207) RATIOP
 132  WRITE(6,207) RATIOP
C
C GO TO SUBROUTINE MODIF1 TO INCREASE REINFORCEMENT
C
      TADEQ=2
      RETURN
 20   IF(TDESIG.NE.1) GO TO 133
```

```
      IF(TMODE.EQ.1) GO TO 2008
      GO TO 138
 2008 WRITE(7,205) RATIOP
 138  WRITE(6,205) RATIOP
C
C RETURN TO MAIN ROUTINE AND CONTINUE EXECUTION
C
      TADEQ=3
      RETURN
C
C     ***FORMAT STATEMENTS***
C
 205  FORMAT(' ',5X,'FOR SECTION SELECTED THE AXIAL LOAD CAPACITY',
     * /6X,'UNDER APPLIED MOMENT IS',
     * F7.3,' TIMES'/6X,'THE APPLIED AXIAL LOAD'/)
 206  FORMAT(' ',5X,'FOR SECTION ENTERED THE AXIAL LOAD CAPACITY',
     * /6X,'UNDER APPLIED MOMENT IS',
     * F7.3,' TIMES'/6X,'THE APPLIED AXIAL LOAD'/)
 207  FORMAT(' ',5X,'FOR SECTION SELECTED THE AXIAL LOAD CAPACITY',
     * /6X,'UNDER APPLIED MOMENT IS',
     * F7.3,' TIMES'/6X,'THE APPLIED AXIAL LOAD, AND IS NOT',
     * ' ACCEPTABLE,'/6X,'INCREASING REINFORCEMENT'/)
      END
C
C  *************************************************************
C  *                                                           *
      SUBROUTINE SLEND2
C  *                                                           *
C  *THIS SUBROUTINE CHECKS SLENDERNESS EFFECTS ABOUT Y-AXIS.
C  *APPLICATION MORE OR LESS IS SIMILAR TO SLEND1              *
C  *                                                           *
C  *************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREO,MOXREO,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      IF(TOUT.NE.1) GO TO 3000
      IF(TMODE.EQ.1) WRITE(7,300)
      WRITE(6,300)
 300  FORMAT(' ',2X,'CHECK SLENDERNESS EFFECTS BENDING',
     * ' ABOUT Y-AXIS'/1X,70('*')/)
C
C COMPUTE MOMENT OF INERTIA OF CROSS-SECTION AND RADIUS OF
C GYRATION AND SLENDERNESS EFFECTS ABOUT Y-AXIS
C
 3000 IGY=CG*AG*CX**2
      RYT=IGY/AG
      RY=SQRT(RYT)
      SY=KY*LUY/RY
      IF(TYBRAC.EQ.1) GO TO 10
      IF(SY.LE.22.) GO TO 11
      IF(SPU.LE.0.1.AND.SPC.LE.0.1) GO TO 12
      CMY=1.0
C
C COMPUTE MAGNIFICATION FACTOR DELTAY
C
      DELTAY=CMY/(1.-SPU/(PHIV*SPC))
      IF(DELTAY.LE.1.0) DELTAY=1.0
      GO TO 13
 11   DELTAY=1.0
      GO TO 50
C
C PROCEED TO NEXT PROBLEM RESULTING FROM INCOMPLETE INPUT PARAMETERS
C
 12   TSLEN2=2
      IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,201)
 131  WRITE(6,201)
      RETURN
 10   IF(ABS(MY2).LE.0.) GO TO 510
      MY12=ABS(MY1/MY2)
      GO TO 511
 510  MY12=1.0
 511  IF(SY.LT.34.-12.*MY12) GO TO 14
      IF(TOUT.NE.1) GO TO 3001
      IF(TMODE.EQ.1) GO TO 2002
      GO TO 132
 2002 WRITE(7,202)
 132  WRITE(6,202)
C
C COMPUTE EI VALUE AND CRITICAL LOAD
C
 3001 EIY=(0.2*EC*IGY+ES*ISEY)/(1.+BDY)
      PCY=(PRIME**2*EIY)/(KY*LUY)**2
C
C DETERMINE GOVERING ECCENTRICITY AND REQUIRED END MOMENT MUY
C FOR LONG COLUMNS
C
      IF(ABS(MY2).LE.0.1) GO TO 15
      B=MY1/MY2
      GO TO 20
 15   B=1.0
 20   IF(TCODE.EQ.1) GO TO 16
      EMINX=0.6+0.03*CX
      GO TO 17
 16   IF(TYPE.EQ.1) GO TO 18
      F=0.05*CX
      GO TO 19
 18   F=0.1*CX
 19   EMINX=AMAX1(F,EMIN)
 17   EX=ABS(MY2)/PU
      IF(EX.LE.EMINX) GO TO 500
      GO TO 501
 500  EX=EMINX
```

```
            IF(TCODE.EQ.1) MUYY=(PU*EX)/1000000.
            IF(TCODE.EQ.2) MUYY=(PU*EX)/12000.
            IF(TMODE.EQ.1) GO TO 550
            GO TO 551
   550      WRITE(7,502) MUYY
   551      WRITE(8,502) MUYY
   501      MUY=PU*EX
            CMY=0.6+0.4*B
            IF(CMY.LT.0.4) CMY=0.4
            DELTAY=CMY/(1.-PU/(PHIV*PCY))
            IF(DELTAY.LE.1.0) DELTAY=1.0
            GO TO 504
   14       IF(TOUT.NE.1) GO TO 3007
            IF(TMODE.EQ.1) GO TO 2003
            GO TO 133
   2003     WRITE(7,203)
   133      WRITE(8,203)
   3007     DELTAY=1.0
            GO TO 50
C DETERMINE GOVERNING ECCENTRICITY AND REQUIRED END MOMENT
C FOR SHORT COLUMNS
C
   13       IF(TCODE.EQ.1) GO TO 50
            EMINX=0.6+0.03*CX
            GO TO 51
   50       IF(TYPE.EQ.1) GO TO 52
            F=0.05*CX
            GO TO 53
   52       F=0.1*CX
   53       EMINX=AMAX1(F,EMIN)
   51       EX=ABS(MY2)/PU
            IF(EX.LE.EMINX) GO TO 505
            GO TO 506
   505      EX=EMINX
            MUY=PU*EX
            IF(TCODE.EQ.1) MUYY=MUY/1000000.
            IF(TCODE.EQ.2) MUYY=MUY/12000.
            IF(TMODE.EQ.1) GO TO 552
            GO TO 553
   552      WRITE(7,502) MUYY
   553      WRITE(8,502) MUYY
   506      MUY=PU*EX
   504      IF(DELTAY.LE.1.001) GO TO 820
            WRITE(8,204)DELTAY
            IF(TMODE.EQ.1) WRITE(7,204)DELTAY
C COMPUTE MAGNIFIED MOMENT THAT IS APPLIED MOMENT
   820      MCY=DELTAY*MUY
            EX=MCY/PU
            MMINY=PU*EMINX
            MAPPY=MCY
C
C RETURN TO MAIN ROUTINE AND CONTINUE EXECUTON
C
            TSLEN2=1
            RETURN
C
C           *****FORMAT STATEMENTS*****
C
   201      FORMAT(' ',5X,'THIS IS A LONG UNBRACED FRAME. MORE INPUT',
           */6X,'PARAMETERS WILL BE REQUIRED TO COMPUTE'/6X,
           * 'MOMENT MAGNIFIER ABOUT Y-AXIS ONLY'/)
   202      FORMAT(' ',5X,'SLENDERNESS EFFECT MUST BE CONSIDERED'/6X,
           * 'BY MAGNIFYING MOMENT ABOUT Y-AXIS'/)
   203      FORMAT(' ',5X,'SLENDERNESS EFFECT MAY BE NEGLECTED'/)
   502      FORMAT(' ',5X,'MINIMUM ECCENTRICITY ABOUT Y-AXIS GOVERNS',
           * /6X,'DESIGN MOMENT ABOUT Y-AXIS = ',F8.0/6X,'IN PROJECT',
           * ' UNIT'/)
   204      FORMAT(' ',5X,'MAGNIFICATION FACTOR FOR MOMENT ABOUT',
           * ' Y-AXIS =',F5.2/)
C
            END
C
C    ***********************************************************
C    *                                                         *
            SUBROUTINE CAPAC2
C    *                                                         *
C    *THIS SUBROUTINE INVESTIGATES STRENGTH CAPACITY ABOUT Y-AXIS *
C    *IN COMPRESSION CONTROLS REGIONS ( PNY > PBY ) AND AT BALANCED *
C    *CONDITION WITH BARS DISTRIBUTED IN LATERAL FACES OR IN   *
C    *END FACES OR IN BOTH FACES OR IN CIRCULAR CORE           *
C    *APPLICATION MORE OR LESS IS SIMILAR TO CAPAC1            *
C    *                                                         *
C    ***********************************************************
C
            IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
            COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
            COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
           * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
           * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
            COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
           * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
            COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
            COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
           * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
           * MAPPX,PUDESX,MUDESX,PBX
            COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
           * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
           * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
            COMMON TESTMO
            COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
            DIMENSION A(9)
C
            IF(TOUT.NE.1) GO TO 4000
            IF(TMODE.EQ.1) WRITE(7,300)
            WRITE(8,300)
   300      FORMAT(' ',2X,'CAPACITY CHECK FOR BENDING ABOUT Y-AXIS'/1X,
           * 70('*')/)
   4000     IF(TRFACE.EQ.1) GO TO 10
C
C COMPUTE DESIGN STRENGTH WITH BARS IN END FACES OR IN BOTH FACES
C OR IN CIRCULAR CORE
C
            CALL CAPA22
            GO TO 11
C
C COMPUTE DESIGN STRENGTH WITH BARS IN LATERAL FACES
C
   10       CALL CAPA21
   11       IF(TOUT.NE.1) RETURN
            PUDY=PUDESY/1000.
```

```
      MPBY=PBY/1000.
      IF(TCODE.EQ.1) GO TO 1990
      MUDY=MUDESY/12000.
      GO TO 1991
 1990 MUDY=MUDESY/1000000.
 1991 WRITE(6,200) XC,XB,PUDY,MUDY,MPBY
      IF(TMODE.EQ.1) WRITE(7,200)XC,XB,PUDY,MUDY,MPBY
      RETURN
C
 200  FORMAT(' ',2X,'LOCATION OF NEUTRAL AXIS // Y-AXIS= ',
     * F8.3,1X,'IN/MM'/3X,'NEUTRAL AXIS AT BALANCED CONDITION= ',
     *F8.3,1X,'IN/MM'/3X,'DESIGN AXIAL',
     * ' LOAD WITH ONLY Y-AXIS BENDING= ',F8.0,1X,'KIPS/KN'/3X,
     * 'DESIGN MOMENT ABOUT Y-AXIS ONLY= ',F8.0,1X,'KIP-FT/KN-M'/3X,
     * 'LOAD AT BALANCED STRAIN CONDITION= ',F8.0,1X,'KIPS/KN'/)
C
      END
C
C   ***********************************************************
C   *                                                         *
      SUBROUTINE CAPA21                                       *
C   *                                                         *
C   *THIS SUBROUTINE DETERMINES DESIGN STRENGTH AT BALANCED   *
C   *CONDITION AND IN COMPRESSION REGION WITH BARS IN         *
C   *LATERAL FACES.                                           *
C   *APPLICATION MORE OR LESS IS SIMILAR TO CAPA12            *
C   *                                                         *
C   ***********************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
      D=CX-DP
C
C COMPUTE COMPRESSION FORCE IN STEEL
C
      CSS=0.5*AST*(FY-0.85*FPC)
C
C USING HALFING INTERVAL TECHNIQUE TO FIND LOCATION OF NEUTRAL AXIS
C
      XL=(ECU*D)/(ECU+ESMAX)
      XU=CX/B1
 10   XC=(XL+XU)/2.
C
C COMPUTE FORCES IN TENSION REINFORCEMENT AND COMPRESSION CONCRETE
C
      FTT=0.5*AST*ECU*ES*((D-XC)/XC)
      CC=0.85*FPC*B1*XC*CY
      XX=XU-XL
      IF(TCODE.EQ.1) DIFFER=5
      IF(TCODE.EQ.2) DIFFER=0.2
      IF(ABS(XX).LT.DIFFER) GO TO 11
      FUN=F21(XC,FTT,CSS,CC,D,CX,EX,DP,B1)
      IF(FUN) 12,11,14
 12   XU=XC
      GO TO 10
 14   XL=XC
      GO TO 10
 11   IF(B1*XC.GT.D) GO TO 20
      GO TO 21
C
C EFFECT OF DISPLACED CONCRETE IS INCLUDED
C
 20   FTT=FTT+0.5*AST*0.85*FPC
 21   PNY=CC+CSS-FTT
      MNY=FTT*(D-0.5*CX)+CSS*(0.5*CX-DP)+CC*(0.5*CX-0.5*B1*XC)
      PUDESY=PHIV*PNY
      MUDESY=PHIV*MNY
C
C COMPUTE BALANCED AXIAL LOAD
C
      XB=(ECU*D)/(FY/ES+ECU)
      PBY=PHIV*0.85*FPC*(B1*XB*CY-0.5*AST)
      RETURN
      END
C
C
      REAL FUNCTION F21(XC,FTT,CSS,CC,D,CX,EX,DP,B1)
C
C TAKING MOMENT OF INTERNAL FORCES ABOUT APPLIED AXIAL LOAD
C AND WRITING AN EQUATION IN TERM OF FUNCTION OF UNKNOWN
C NEUTRAL AXIS
C
      F21=FTT*(D-(0.5*CX-EX))+CSS*(0.5*CX-EX-DP)+CC*(0.5*CX-EX
     *    -0.5*B1*XC)
      RETURN
      END
C
C   ***********************************************************
C   *                                                         *
      SUBROUTINE CAPA22                                       *
C   *                                                         *
C   *THIS SUBROUTINE DETERMINES DESIGN STRENGTH AT BALANCED   *
C   *CONDITION AND IN COMPRESSION CONTROL REGION WITH BARS    *
C   *DISTRIBUTED IN END FACES OR IN BOTH FACES OR IN CIRCULAR CORE.*
C   *APPLICATON MORE OR LESS IS SIMILAR TO CAPA12             *
C   *                                                         *
C   ***********************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
```

```
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(8)
C
      D=CX-DP
C
C COMPUTE BALANCED AXIAL LOAD BY USING DIRECT APPLICATION OF STATICS
C
      XB=(ECU*D)/(ECU+ESMAX)
      AB=B1*XB
C
C COMPUTE CONTROL PARAMETERS IN COUNTER DO LOOP
C
      GO TO (10,11,12,13),TRFACE
 11   TL=IFIX((NMAX-1.)/4.+0.51)
      GO TO 14
 12   TL=IFIX((2.*NY-1.)/4.+0.51)
      IF(TL.LE.0) GO TO 1
      GO TO 14
 13   TL=IFIX((NMAX-3.)/4.+0.51)
C
C COMPUTE STRAINS AT VARIOUS BARS AND CHECK WHETHER THEY YIELD
C OR NOT
C
 14   DO 15 NI=1,TL
      ECSY(NI)=((XB-0.5*CX+X(NI))*ECU)/XB
      IF(ABS(ECSY(NI)).GT.ESMAX) GO TO 80
      GO TO 81
 80   ECSY(NI)=(ESMAX*ECSY(NI))/ABS(ECSY(NI))
 81   ETSY(NI)=((X(NI)+0.5*CX-XB)*ESMAX)/(D-XB)
      IF(ABS(ETSY(NI)).GT.ESMAX) GO TO 82
      GO TO 15
 82   ETSY(NI)=(ESMAX*ETSY(NI))/ABS(ETSY(NI))
 15   CONTINUE
C
C CHECK WHETHER HAVING EXTREME COMPRESSION BARS AND EXTREME TENSION
C BARS AND CENTER BARS OR NOT HAVING THEM
C
 1    GO TO (10,16,17,18),TRFACE
 16   NO=NMAX/2
      NOO=NO/2
C
C CONSIDERING NO EXTREME COMPRESSION AND TENSION BARS IN END FACES
C
      ECSR=0.
      ETSL=0.
      GO TO 19
 17   NO=NY
      NOO=NO/2
      GO TO 20
 18   NO=NMAX/2+1
      NOO=NO/2
 20   ECSR=((XB-DP)*ECU)/XB
      IF(ABS(ECSR).GT.ESMAX) GO TO 83
      GO TO 84
 83   ECSR=(ESMAX*ECSR)/ABS(ECSR)
 84   ETSL=ESMAX
      IF(NO.EQ.0) GO TO 23
 19   IF(NO/2.-NOO.LE.0.0001) GO TO 23
      ESC=((XB-0.5*CX)*ECU)/XB
      IF(ABS(ESC).GT.ESMAX) GO TO 85
      GO TO 24
 85   ESC=(ESMAX*ESC)/ABS(ESC)
      GO TO 24
 23   ESC=0.
C
C CALCULATE FORCES IN SECTION
C
 24   GO TO (10,25,26,27),TRFACE
C
C BASED ON ECSR=0 AND ETSL=0
C
 25   CSR=0.
      FTL=0.
      CC=0.85*FPC*AB*CY
      GO TO 30
 26   CSR=(NX*PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(NX*PRIME*DB(J)**2*ES*ETSL)/4.
      CC=0.85*FPC*AB*CY
      IF(TL.LE.0) GO TO 28
      GO TO 30
 28   CSSUM=0.
      FTSUM=0.
      GO TO 2
C
C DETERMINE AREA OF SEGMENT FOR A CIRCULAR REINFORCEMENT PATTERN
C
 27   RR=0.5*CX
      IF(AB.LE.RR) GO TO 31
      PP=AB-RR
      GO TO 32
 31   PP=RR-AB
 32   Q=RR**2-PP**2
      QQ=SQRT(Q)
      PR=PP/RR
      RFAD=ARCOS(PR)
      IF(AB.GT.RR) GO TO 86
      GO TO 87
 86   RFAD=PRIME-RFAD
 87   AREA=(CX**2*(RFAD-SIN(RFAD)*COS(RFAD)))/4.
      CC=0.85*FPC*AREA
      CSR=(PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(PRIME*DB(J)**2*ES*ETSL)/4.
C
C COMPUTE FORCES IN VARIOUS BARS
C
 30   CSSUM=0.
      FTSUM=0.
      DO 35 NI=1,TL
      CSY(NI)=(2.*PRIME*DB(J)**2*(ES*ECSY(NI)-0.85*FPC))/4.
      CSSUM=CSSUM+CSY(NI)
      FTY(NI)=(2.*PRIME*DB(J)**2*ES*ETSY(NI))/4.
```

```
      FTSUM=FTSUM+FTY(NI)
   35 CONTINUE
    2 IF(ESC.LE.0.) GO TO 29
      CSC=(2.*PRIME*DB(J)**2*(ES*ESC-0.85*FPC))/4.
      GO TO 21
   29 CSC=0.
C
C CHECK WHETHER ALL FORCES IN TENSION REINFORCEMENT ARE ACTUALLY
C A TENSION OR NOT, THAT IS THE EFFECT OF THE DISPLACED CONCRETE
C FTP IS INCLUDED OR NOT
C
   21 IF(TL.LE.0) GO TO 4
      NII=1
   36 FTP=((TL-NII+1.)*2.*PRIME*DB(J)**2*0.85*FPC)/4.
      IF(NII.GT.TL) GO TO 37
      IF(FTY(NII).LT.0.) GO TO 37
      NII=NII+1
      GO TO 36
    4 FTP=0.
   37 IF(FTL.GE.0.) GO TO 38
      IF(TRFACE.EQ.4) GO TO 39
      FTL=FTL+(NX*PRIME*DB(J)**2*0.85*FPC)/4.
      GO TO 38
   39 FTL=FTL+(PRIME*DB(J)**2*0.85*FPC)/4.
   38 PBY=PHIV*(CC+CSR+CSC+CSSUM-FTL-FTSUM-FTP)
C
C COMPUTE DESIGN STRENGTH IN COMPRESSION CONTROL REGION
C
C USING HALFING INTERVAL TECHNIQUE TO FIND LOCATION OF NEUTRAL AXIS
C
      XL=XB
      XU=CX/B1
   40 XC=(XL+XU)/2.
      XX=XU-XL
      IF(TCODE.EQ.1) DIFFER=5
      IF(TCODE.EQ.2) DIFFER=0.2
      IF(ABS(XX).LT.DIFFER) GO TO 41
      AC=B1*XC
C
C COMPUTE STRAINS AT VARIOUS BARS AND CHECK THEY YIELD OR NOT
C
      IF(TL.LE.0) GO TO 3
      DO 42 NI=1,TL
      ECSY(NI)=((XC-0.5*CX+X(NI))*ECU)/XC
      IF(ABS(ECSY(NI)).GT.ESMAX) GO TO 88
      GO TO 89
   88 ECSY(NI)=(ESMAX*ECSY(NI))/ABS(ECSY(NI))
   89 ETSY(NI)=((X(NI)+0.5*CX-XC)*ECU)/XC
      IF(ABS(ETSY(NI)).GT.ESMAX) GO TO 90
      GO TO 42
   90 ETSY(NI)=(ESMAX*ETSY(NI))/ABS(ETSY(NI))
   42 CONTINUE
C
C CHECK ECSR AND ETSL AND ESC EQUAL TO ZERO OR NOT
C
    3 GO TO (10,43,44,44),TRFACE
   44 ECSR=((XC-DP)*ECU)/XC
      IF(ABS(ECSR).GT.ESMAX) GO TO 91
      GO TO 92
   91 ECSR=(ESMAX*ECSR)/ABS(ECSR)
   92 ETSL=((D-XC)*ECU)/XC
      IF(ABS(ETSL).GT.ESMAX) GO TO 93
      GO TO 94
   93 ETSL=(ESMAX*ETSL)/ABS(ETSL)
   94 IF(NO.EQ.0) GO TO 48
   43 IF(NO/2.-NOD.LE.0.0001) GO TO 48
      ESC=((XC-0.5*CX)*ECU)/XC
      IF(ABS(ESC).GT.ESMAX) GO TO 95
      GO TO 49
   95 ESC=(ESMAX*ESC)/ABS(ESC)
      GO TO 49
   48 ESC=0.
C
C CALCULATE FORCES IN SECTION
C
   49 GO TO (10,50,51,52),TRFACE
C
C BASED ON ECSR=0 AND ETSL=0
C
   50 CSR=0.
      FTL=0.
      CC=0.85*FPC*AC*CY
C
C COMPUTE POINT OF APPLICATION FOR CC
C
      XCC=CX/2.-0.5*AC
      GO TO 55
   51 CSR=(NX*PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(NX*PRIME*DB(J)**2*ES*ETSL)/4.
      CC=0.85*FPC*AC*CY
      XCC=CX/2.-0.5*AC
      IF(TL.LE.0) GO TO 53
      GO TO 55
C
C DETERMINE AREA OF SEGMENT IN CIRCULAR REINFORCEMENT PATTERN
C
   52 IF(AC.LE.RR) GO TO 56
      PP=AC-RR
      GO TO 57
   56 PP=RR-AC
   57 Q=RR**2-PP**2
      QQ=SQRT(Q)
      PR=PP/RR
      RFAD=ARCOS(PR)
      IF(AC.GT.RR) GO TO 98
      GO TO 97
   98 RFAD=PRIME-RFAD
   97 AREA=(CX**2*(RFAD-SIN(RFAD)*COS(RFAD)))/4.
      SEG=(CX**3*(SIN(RFAD))**3)/12.
C
C COMPUTE POINT OF APPLICATION FOR CC
C
      XCC=SEG/AREA
      CC=0.85*FPC*AREA
      CSR=(PRIME*DB(J)**2*(ES*ECSR-0.85*FPC))/4.
      FTL=(PRIME*DB(J)**2*ES*ETSL)/4.
   55 DO 60 NI=1,TL
      CSY(NI)=(2.*PRIME*DB(J)**2*(ES*ECSY(NI)-0.85*FPC))/4.
      FTY(NI)=(2.*PRIME*DB(J)**2*ES*ETSY(NI))/4.
   60 CONTINUE
```

```
 53     IF(ESC.LE.0.) GO TO 54
        CSC=(2.*PRIME*DB(J)**2*(ES*ESC-0.85*FPC))/4.
        GO TO 46
 54     CSC=0.
C
C CHECK TO SEE ALL FTY(NI) AND FTL ARE ACTUALLY A TENSION OR NOT
C
 46     IF(TL.LE.0) GO TO 62
        NI=1
 61     IF(FTY(NI).GT.0.) GO TO 62
        FTY(NI)=FTY(NI)+(2.*PRIME*DB(J)**2*0.85*FPC)/4.
        NI=NI+1
        IF(NI.LE.TL) GO TO 61
 62     IF(FTL.GE.0.) GO TO 63
        IF(TRFACE.EQ.4) GO TO 64
        FTL=FTL+(NX*PRIME*DB(J)**2*0.85*FPC)/4.
        GO TO 63
 64     FTL=FTL+(PRIME*DB(J)**2*0.85*FPC)/4.
 63     IF(TL.LE.0) GO TO 6
        GO TO 7
 6      FTY(1)=0.
        CSY(1)=0.
 7      FUN=F22(FTY,CSY,X,FTL,CX,DP,EX,TL,CSR,CSC,CC,XCC)
        IF(FUN) 65,41,66
 65     XU=XC
        GO TO 40
 66     XL=XC
        GO TO 40
 41     FTSUM=0.
        CSSUM=0.
        IF(TL.LE.0) GO TO 8
        DO 67 NI=1,TL
        FTSUM=FTSUM+FTY(NI)
        CSSUM=CSSUM+CSY(NI)
 67     CONTINUE
 8      PNY=CSR+CSSUM+CSC+CC-FTL-FTSUM
        FTSUM1=0.
        CSUM1=0.
        IF(TL.LE.0) GO TO 9
        DO 100 NI=1,TL
        FTSUM1=FTY(NI)*X(NI)+FTSUM1
        CSUM1=CSUM1+CSY(NI)*X(NI)
 100    CONTINUE
 9      MNY=FTL*(0.5*CX-DP)+CSR*(0.5*CX-DP)+CC*XCC+FTSUM1+CSUM1
        PUDESY=PHIV*PNY
        MUDESY=PHIV*MNY
 10     RETURN
        END
C
C
        FUNCTION F22(FTY,CSY,X,FTL,CX,DP,EX,TL,CSR,CSC,CC,XCC)
C
C TAKING MOMENT OF INTERNAL FORCES ABOUT APPLIED AXIAL LOAD
C AND WRITING AN EQUATION IN TERMS OF FUNCTION OF UNKNOWN
C NEUTRAL AXIS
C
        IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
        DIMENSION FTY(12),CSY(12),X(12)
        FTSUM=0.
        CSUM=0.
        IF(TL.LE.0) GO TO 100
        DO 10 NI=1,TL
        FTSUM=FTSUM+FTY(NI)*(X(NI)+EX)
        CSUM=CSUM+CSY(NI)*(X(NI)-EX)
 10     CONTINUE
 100    F22=FTSUM+CSUM+FTL*(0.5*CX-DP+EX)+CSR*(0.5*CX-DP-EX)-CSC*EX
     *      -CC*(EX-XCC)
        RETURN
        END
C
C   ***************************************************************
C   *                                                             *
        SUBROUTINE ADEQU2
C   *                                                             *
C   *THIS SUBROUTINE CHECKS DESIGN STRENGTH CAPACITY WITH THE     *
C   *REQUIRED STRENGTH IN BIAXIAL BENDING                         *
C   *                                                             *
C   ***************************************************************
C
        IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
        COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
        COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     *  FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     *  TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
        COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     *  SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
        COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
        COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     *  Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     *  MAPPX,PUDESX,MUDESX,PBX
        COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     *  YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     *  XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
        COMMON TESTMO
        COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
        DIMENSION A(9)
        EQUIVALENCE (MOX,MUDESX),(MOY,MUDESY),(MX,MAPPX),(MY,MAPPY)
C
        IF(TOUT.NE.1) GO TO 4000
        IF(TMODE.EQ.1) WRITE(7,400)
        WRITE(8,400)
 400    FORMAT(' ',2X,'ADEQUACY CHECK FOR BIAXIAL BENDING'/1X,
     *  70('*')/)
 4000   PDASH=0.1*FPC*AG
C
C DETERMINE WHETHER CHECKING DESIGN AXIAL LOAD OR DESIGN MOMENT
C
        IF(PU.GE.PDASH) GO TO 10
        GO TO 11
C
C USING BRESLER RECIPROCAL LOAD METHOD
C
 10     PO=PHIV*(0.85*FPC*(AG-AST)+FY*AST)
        PIO=1./PUDESX+1./PUDESY-1./PO
C
C COMPUTE PROVIDED AXIAL LOAD
C
        PUPRO=1./PIO
        RATPX=PUPRO/PU
        MOREC=RATPX-1.0
```

```
      IF(PUPRO.GE.PU) GO TO 65
C
C SET A TOLERANCE LIMIT OF 0.03 FOR COMPARING PROVIDED
C AXIAL LOAD WITH APPLIED AXIAL LOAD
C
      IF(MOREC+0.03.GE.0.) GO TO 65
      IF(TDESIG.EQ.1) GO TO 34
 75   IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,215)RATPX
 131  WRITE(8,215)RATPX
      IF(MOREC+0.03.GE.0.) GO TO 2880
      IF(TMODE.EQ.1) WRITE(7,2881)
      WRITE(8,2881)
 2881 FORMAT(' ',5X,'** SECTION IS NOT ADEQUATE FOR THIS',
     * ' LOADING **'/)
C
C PROCEED TO NEXT PROBLEM RESULTING FROM UNSATISFACTORY SECTION
C
 2880 TADEQ=1
      RETURN
 34   IF(TMODE.EQ.1) GO TO 2002
      GO TO 132
 2002 IF(TOUT.NE.1) GO TO 132
      WRITE(7,216)RATPX
 132  WRITE(8,216)RATPX
C
C GO TO SUBROUTINE MODIF1 TO INCREASE REINFORCEMENT
C
      TADEQ=2
      RETURN
 65   IF(TDESIG.NE.1) GO TO 75
      IF(TMODE.EQ.1) GO TO 2016
      GO TO 146
 2016 WRITE(7,217) RATPX
 146  WRITE(8,217) RATPX
      GO TO 70
C
C COMPUTE NEW CAPACITY REDUCTION FACTOR, PHIVN
 11   PBXX=PBX/PHIV
      PBYY=PBY/PHIV
      PBB=AMIN1(PBXX,PBYY)
      PDAA=AMIN1(PBB,PDASH)
      PHIVN=PHIV+((0.9-PHIV)*(PDAA-PU))/PDAA
      IF(PHIVN.LT.PHIV) PHIVN=PHIV
      MOX=MOX*PHIVN/PHIV
      MOY=MOY*PHIVN/PHIV
      MX=MX*PHIVN/PHIV
      MY=MY*PHIVN/PHIV
      PN=PU/PHIVN
      BM=MOX/MOY
C
C USING LOAD CONTOUR METHOD:PARME AND GOUWENS APPROACH
C
      CCC=FPC*CX*CY
      CSC=FY*AST
C
C CALCULATE B25, THE MINIMUM VALUE OF BB WHICH OCCURS AT
C P=0.25*CCC
C
      IF(CSC/CCC.GE.0.5) GO TO 16
      B25=0.545+0.35*(0.5-CSC/CCC)**2
      GO TO 17
 16   B25=0.485+0.03*(CCC/CSC)
 17   IF(PN/CCC.GE.0.25) GO TO 18
      BB=B25+(0.25-PN/CCC)**2*(0.85-0.5*(CSC/CCC))
      GO TO 19
 18   BB=B25+(0.2*(PN/CCC-0.25))/(0.85+CSC/CCC)
C
C DETERMINE WHICH PORTION OF THE BILINEAR INTERACTION
C CURVE IS TO BE USED
C
C DETERMINE EQUIVALENT REQUIRED UNIAXIAL MOMENT
C
 19   MOXREQ=MX+(MY*BM)*((1.-BB)/BB)
      MOYREQ=MY+(MX/BM)*((1.-BB)/BB)
C
C USING EQN. BI : A STRAIGHT-LINE APPROXIMATION TO THE LOAD CONTOUR
C
      IF(MY/MX.GE.1./BM) GO TO 20
      BI=MX/MOXREQ+(MY/MOYREQ)*((1.-BB)/BB)
      IF(TCODE.EQ.1) GO TO 1990
      MOXRE=MOXREQ/12000.
      GO TO 1991
 1990 MOXRE=MOXREQ/1000000.
 1991 IF(TMODE.EQ.1) GO TO 2017
      GO TO 147
 2017 IF(TOUT.NE.1) GO TO 147
      WRITE(7,200) MOXRE
 147  WRITE(8,200) MOXRE
      GO TO 21
 20   BI=(MX/MOXREQ)*((1.-BB)/BB)+MY/MOYREQ
      IF(TCODE.EQ.1) GO TO 1992
      MOYRE=MOYREQ/12000.
      GO TO 1993
 1992 MOYRE=MOYREQ/1000000.
 1993 IF(TMODE.EQ.1) GO TO 2018
      GO TO 148
 2018 IF(TOUT.NE.1) GO TO 148
      WRITE(7,201) MOYRE
 148  WRITE(8,201) MOYRE
C
C SET TOLERANCE LIMIT OF 0.03
C
 21   IF(BI.LE.1.03) GO TO 22
      IF(TDESIG.EQ.1) GO TO 23
      IF(TMODE.EQ.1) GO TO 2020
      GO TO 150
 2020 WRITE(7,203)
 150  WRITE(8,203)
C
C PROCEED TO NEXT PROBLEM RESULTING FROM BI VALUE > 1.0
C
      TADEQ=1
      RETURN
 23   IF(TMODE.EQ.1) GO TO 2021
      GO TO 151
 2021 IF(TOUT.NE.1) GO TO 151
      WRITE(7,204)
 151  WRITE(8,204)
```

```
C
C GO TO SUBROUTINE MODIF1 TO INCREASE REINFORCEMENT
C
      TADEQ=2
      RETURN
 22   IF(TDESIG.NE.1) GO TO 133
      IF(TMODE.EQ.1) GO TO 2022
      GO TO 152
 2022 WRITE(7,205)
 152  WRITE(8,205)
      GO TO 70
 133  IF(TMODE.EQ.1) WRITE(7,420)
      WRITE(8,420)
      TADEQ=1
      RETURN
C
C RETURN TO MAIN ROUTINE AND CONTINUE EXECUTION
C
 70   TADEQ=3
      RETURN
C
C     ***FORMAT STATEMENTS***
C
 200  FORMAT(' ',5X,'EQUIVALENT REQUIRED MOMENT ABOUT X-AXIS',
     */6X,'ONLY =',F8.0,' IN PROJECT UNITS'/)
 201  FORMAT(' ',5X,'EQUIVALENT REQUIRED MOMENT ABOUT Y-AXIS',
     * /6X,'ONLY =',F8.0,' IN PROJECT UNITS'/)
 203  FORMAT(' ',5X,'ENTERED SECTION IS NOT ADEQUATE BASED ON'/6X,
     * 'MOMENT CAPACITIES'/)
 204  FORMAT(' ',5X,'SELECTED SECTION IS NOT ADEQUATE BASED ON'/6X,
     * 'MOMENT CAPACITIES, INCREASING REINFORCEMENT '/)
 205  FORMAT(' ',5X,'SELECTED SECTION IS ADEQUATE BASED ON'/6X,
     * 'MOMENT CAPACITIES'/)
 215  FORMAT(' ',5X,'FOR SECTION ENTERED THE AXIAL LOAD CAPACITY',
     * /6X,'UNDER APPLIED MOMENT IS',
     * F7.3,' TIMES'/6X,'THE APPLIED AXIAL LOAD'/)
 216  FORMAT(' ',5X,'FOR SECTION SELECTED THE AXIAL LOAD CAPACITY',
     * /6X,'UNDER APPLIED MOMENT IS',
     * F7.3,' TIMES'/6X,'THE APPLIED AXIAL LOAD, AND IS NOT',
     * ' ACCEPTABLE,'/6X,'INCREASING REINFORCEMENT '/)
 217  FORMAT(' ',5X,'FOR SECTION SELECTED THE AXIAL LOAD CAPACITY',
     * /6X,'UNDER APPLIED MOMENT IS',
     * F7.3,' TIMES'/6X,'THE APPLIED AXIAL LOAD'/)
 420  FORMAT(' ',5X,'ENTERED SECTION IS ADEQUATE BASED ON'/6X,
     * 'MOMENT CAPACITIES'/)
      END
C
C  ***************************************************************
C  *                                                             *
      SUBROUTINE MODIF1
C  *                                                             *
C  *THIS SUBROUTNE MODIFIES SECTIONS                             *
C  *                                                             *
C  ***************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,T1,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
      EQUIVALENCE (MOX,MUDESX),(MOY,MUDESY),(MX,MAPPX),(MY,MAPPY)
C
      IF(TOUT.NE.1) GO TO 13
      IF(TMODE.EQ.1) WRITE(7,400)
      WRITE(8,400)
 400  FORMAT(' ','MODIFICATION INFORMATION :'/)
C
C CHECK REQUIRED TOTAL NO. OF BARS IS ENTERED OR NOT
C
 13   IF(NBS.NE.0) GO TO 7
C
C CHECK MAX. PERMISSIBLE NO.. OF BARS IS USED OR NOT
C
      IF(NMAX.LT.TBMAX) GO TO 8
C
C INCREASE BAR SIZE
C
 7    IF(J.GE.TK) GO TO 10
      J=J+1
      TMODAS=1
      GO TO 20
C
C INCREASE TOTAL NO. OF BARS USED
C
 8    NMAX=NMAX+2
      TMODAS=2
      GO TO 20
C
C CHECK INCREASING COLUMN DIMENSIONS IS ALLOWED OR NOT
C
 10   GO TO(14,15,16,17,18),TDIMFG
 14   IF(TOUT.NE.1) GO TO 130
      IF(TMODE.EQ.1) GO TO 2000
      GO TO 130
 2000 WRITE(7,322)
 130  WRITE(8,322)
C
C PROCEED TO NEXT PROBLEM RESULTING FROM LIMITING CONSTRAINTS
C
      TESTMO=2
      RETURN
C
C BACK TO SUBROUTINE REINF1 TO REITERATE THE STEPS
C
 20   TESTMO=1
      RETURN
 15   CY=CY+CYINCR
      GO TO 22
```

```
 16   CX=CX+CXINCR
      GO TO 22
 17   IF(CX.GE.CY) GO TO 3200
      CX=CX+CXINCR
      CY=CX/R
      CY=CYINCR*IFIX(CY/CYINCR)
      GO TO 3201
 3200 CY=CY+CYINCR
      CX=CY*R
      CX=CXINCR*IFIX(CX/CXINCR)
 3201 IF(TSHAPE.EQ.3) GO TO 65
      GO TO 22
 65   AG=(PRIME*CX**2)/4.
      GO TO 66
 18   IF(CX.GE.CY) GO TO 15
      CX=CX+CXINCR
 22   AG=CX*CY
C
C SET FLAG STATING THAT A NEW TRIAL STEEL SECTION IS SELECTED
C
 66   TMODAS=0
      AS=PMIN*AG
      IF(TOUT.NE.1) GO TO 4002
      IF(TSHAPE.EQ.3) GO TO 2880
      IF(TMODE.EQ.1) GO TO 2002
      GO TO 132
 2002 WRITE(7,203) CX,CY
 132  WRITE(6,203) CX,CY
      GO TO 4002
 2880 IF(TMODE.EQ.1) WRITE(7,2881)CX
      WRITE(6,2881)CX
 2881 FORMAT(' ',5X,'MODIFIED CONCRETE DIMENSION :'/2X,'DIAMETER =',
     * F6.1,2X,'IN PROJECT UNITS'/)
C
C BACK TO SUBROUTINE REINF1 TO REITERATE THE STEPS
C
 4002 TESTMO=1
      RETURN
C
C     ***FORMAT STATEMENTS***
C
 203  FORMAT(' ',5X,'MODIFIED CONCRETE DIMENSION :',
     *2X,'CX= ',F6.1,2X,'CY= ',F6.1,2X,'IN PROJECT UNITS'/)
 322  FORMAT(' ',5X,'DESIGN CANNOT BE COMPLETED'/6X,
     *'CONSISTENT WITH ENTERED RESTRAINTS'/)
C
      END
C
C  ***********************************************************
C  *                                                         *
      SUBROUTINE SPIRS1
C  *                                                         *
C  *THIS SUBROUTINE CHECKS REINFORCEMENT WITH LAP SPLICES AND *
C  *SELECTS LATERAL REINFORCEMENT FOR SPIRAL COLUMN           *
C  *                                                         *
C  ***********************************************************
C
       IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,S)
       COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
       COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
       COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
       COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
       COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
       COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
       COMMON TESTMO
       COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
       DIMENSION A(9)
C
C CALCULATE CENTER-TO-CENTER SPACING OF SPIRAL BARS
C
       IF(TOUT.NE.1) GO TO 3001
       IF(TMODE.EQ.1) WRITE(7,300)
       WRITE(6,300)
       GO TO 3001
 300   FORMAT(' ','SELECT SPIRAL :'/)
 3000 SMIN=DSS+SSMIN
       SMAX=DSS+SSMAX
       S3=SMAX-SMIN
C
C DETERMINE MIN. RATIO OF SPIRAL REINFORCEMENT, PSMIN
C
       DC=CX-2.*DCC
       ACORE=(PRIME*DC**2)/4.
       PSMIN=0.45*(AG/ACORE-1.)*(FPC/FSY)
C
C COMPUTE MAX. SPACING, S1
C
       S1=(PRIME*DSS**2*(DC-DSS))/(DC**2*PSMIN)
       IF(TCODE.EQ.1) GO TO 10
       PITCH=AMIN1(S1,S3)
       GO TO 11
C
C DETERMINE DISTANCE BETWEEN TURNS OF SPIRALS
C
 10    S2=DC/6.
       PITCH=AMIN1(S1,S2,S3)
       IF(TOUT.NE.1) GO TO 11
       IF(TMODE.EQ.1) GO TO 2002
       GO TO 132
 2002 WRITE(7,202)TDSSS,PITCH
 132  WRITE(6,202)TDSSS,PITCH
 202  FORMAT(' ',5X,'#',I2,1X,', PITCH =',F6.1,2X,
     * 'IN PROJECT UNITS'/)
C
C DETERMINE NUMBER OF SPACERS NEEDED FOR SPIRALS
C
 11    IF(TCODE.EQ.1) GO TO 12
       CCX1=20.
       CCX2=30.
       CCX3=24.
       CCX4=0.625
       GO TO 2000
 12    CCX1=500.
```

```
      CCX2=750.
      CCX3=600.
      CCX4=19.5
 2000 IF(DSS.LT.CCX4) GO TO 13
      IF(CX.LE.CCX3) GO TO 14
 17   NSPACR=4
      GO TO 15
 14   NSPACR=3
      GO TO 15
 13   IF(CX.LT.CCX1) GO TO 16
      IF(CX.LT.CCX2) GO TO 14
      GO TO 17
 16   NSPACR=2
 15   IF(TOUT.NE.1) RETURN
      IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,201) NSPACR
 131  WRITE(8,201) NSPACR
 201  FORMAT(' ',5X,'NUMBER OF SPACERS REQUIRED= ',I3)
      RETURN
C
C DETERMINE WHETHER REINFORCEMENT CAN BE ACCOMMODATED WITH
C LAP SPLICES OR NOT
C
 3001 IF(TCODE.EQ.1) GO TO 135
      BAR=DB(4)
      BARL=DB(7)
      GO TO 136
 135  BAR=DB(3)
      BARL=DB(5)
 136  IF(DB(J).GE.BAR) GO TO 137
      A2=(SB1+DB(J))/(CX-3.*DB(J)-2.*(DCC+DSS))
      NN=IFIX(180./ARSIN(A2)+0.51)
      A3=(SB1+DB(J))/(CX-DB(J)-2.*(DCC+DSS))
      A4=DB(J)/(CX-DB(J)-2.*(DCC+DSS))
      NT=IFIX(180./(ARSIN(A3)+ARSIN(A4))+0.51)
      GO TO 138
 137  A2=(2.5*DB(J))/(CX-3.*DB(J)-2.*(DCC+DSS))
      NN=IFIX(180./ARSIN(A2)+0.51)
      A3=(2.5*DB(J))/(CX-DB(J)-2.*(DCC+DSS))
      A4=DB(J)/(CX-DB(J)-2.*(DCC+DSS))
      NT=IFIX(180./(ARSIN(A3)+ARSIN(A4))+0.51)
C
C CHECK LAP SPLICES CAN BE USED OR NOT
C
 138  IF(DB(J).GT.BARL) GO TO 151
      GO TO 145
C
C CHECK NORMAL LAP SPLICES
C
 141  IF(NMAX.GT.NN) GO TO 143
      GO TO 3000
 151  IF(TMODE.EQ.1) WRITE(7,152) BARL
      WRITE(8,152) BARL
      GO TO 3000
 143  IF(TMODE.EQ.1) WRITE(7,146)
      WRITE(8,146)
      GO TO 3000
C
C CHECK TANGENTIAL LAP SPLICES
C
 145  IF(NMAX.GT.NT) GO TO 147
      GO TO 141
 147  IF(TMODE.EQ.1) WRITE(7,150)
      WRITE(8,150)
      GO TO 3000
C
C
 146  FORMAT(' ',5X,'NORMAL LAP SPLICES CAN BE USED'/)
 150  FORMAT(' ',5X,'TANGENTIAL LAP SPLICES CAN BE USED'/)
 152  FORMAT(' ',5X,'LAP SPLICES SHOULD NOT BE USED',
     * ' FOR DIAMETER'/6X,'BAR LARGER THAN',F9.4,1X,
     * 'IN PROJECT UNITS'/)
      END
C
C  ****************************************************************
C  *                                                              *
      SUBROUTINE TIES1
C  *                                                              *
C  *THIS SUBROUTINE SELECTS LATERAL REINFORCEMENT FOR TIED COLUMN *
C  *                                                              *
C  ****************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
      DIMENSION A(9)
C
C SELECT COLUMN TIES SPACING REQUIREMENT PER SET; S
C
      IF(TOUT.NE.1) GO TO 3002
      IF(TMODE.EQ.1) WRITE(7,304)
      WRITE(8,304)
      GO TO 3002
 3003 SD=16.*DB(J)
      SS=48.*DSS
      SPACE=AMIN1(SD,SS,CX,CY)
      IF(TOUT.NE.1) GO TO 3000
      IF(TMODE.EQ.1) GO TO 3001
      GO TO 3100
 3001 WRITE(7,206)TDSSS,SPACE
 3100 WRITE(8,206) TDSSS,SPACE
C
C DETERMINE THE LATERAL TIES REQUIRED FOR INTERIOR BARS
C AND ALL CORNER BARS
C
 3000 GO TO(10,10,11,14),TRFACE
```

```
 10   IF(SB.LE.SMAX) GO TO 13
      IF(TOUT.NE.1) GO TO 20
      IF(TMODE.EQ.1) GO TO 2000
      GO TO 130
 2000 WRITE(7,200)
 130  WRITE(8,200)
      GO TO 20
 13   IF(TOUT.NE.1) GO TO 20
      IF(TMODE.EQ.1) GO TO 2001
      GO TO 131
 2001 WRITE(7,201)
 131  WRITE(8,201)
 20   IF(NMAX/2.GE.4) GO TO 132
      GO TO 14
 132  IF(TMODE.EQ.1) GO TO 2002
      GO TO 133
 2002 WRITE(7,300)
 133  WRITE(8,300)
      GO TO 14
C
C FOR BARS IN BOTH FACES
C
 11   IF(SBX.LE.SMAX) GO TO 15
      IF(TOUT.NE.1) GO TO 16
      IF(TMODE.EQ.1) GO TO 2003
      GO TO 134
 2003 WRITE(7,202)
 134  WRITE(8,202)
      GO TO 16
 15   IF(TOUT.NE.1) GO TO 16
      IF(TMODE.EQ.1) GO TO 2004
      GO TO 135
 2004 WRITE(7,203)
 135  WRITE(8,203)
 16   IF(SBY.LE.SMAX) GO TO 17
      IF(TOUT.NE.1) GO TO 21
      IF(TMODE.EQ.1) GO TO 2005
      GO TO 136
 2005 WRITE(7,204)
 136  WRITE(8,204)
      GO TO 21
 17   IF(TOUT.NE.1) GO TO 21
      IF(TMODE.EQ.1) GO TO 2006
      GO TO 137
 2006 WRITE(7,205)
 137  WRITE(8,205)
 21   IF(NX.GE.4) GO TO 138
      GO TO 143
 138  IF(TMODE.EQ.1) GO TO 2007
      GO TO 139
 2007 WRITE(7,301)
 139  WRITE(8,301)
 143  IF(NY.GE.2) GO TO 140
      GO TO 14
 140  IF(TMODE.EQ.1) GO TO 2008
      GO TO 141
 2008 WRITE(7,302)
 141  WRITE(8,302)
 14   RETURN
C
C DETERMINE WHETHER REINFORCEMENT CAN BE ACCOMMODATED WITH
C LAP SPLICES
C
 3002 GO TO(145,146,147,14),TRFACE
C
C COMPUTE NO. OF BARS PER FACE BETWEEN SPLICES
C
 145  NF=NMAX/2
      B4=CY
      GO TO 148
 146  NF=NMAX/2
      B4=CX
      GO TO 148
 147  IF(NX.GT.NY) GO TO 170
      NF=NY
      B4=CX
      GO TO 148
 170  NF=NX
      B4=CY
 148  IF(TCODE.EQ.1) GO TO 149
      BAR=DB(4)
      BARL=DB(7)
      GO TO 150
 149  BAR=DB(3)
      BARL=DB(5)
 150  IF(DB(J).GE.BAR) GO TO 151
      BB1=2.*(DCC+DSS)+NF*DB(J)+(NF-1.)*SB1
      A1=(1.-0.5**0.5)*DB(J)/(SB1+DB(J))
      HETA=ARSIN(A1)
      B2=BB1+(2.*SB1+2.*DB(J))*COS(HETA)-0.866*DB(J)-2.*SB1
      B3=2.*(DCC+DSS)+(2.*NF-1.)*DB(J)+(NF-1.)*SB1
      GO TO 152
 151  BB1=2.*(DCC+DSS)+NF*DB(J)+(NF-1.)*1.5*DB(J)
      B2=BB1+1.38*DB(J)
      B3=2.*(DCC+DSS)+(2.*NF-1.)*DB(J)+(NF-1.)*1.5*DB(J)
C
C CHECK LAP SPLICES CAN BE USED OR NOT
C
 152  IF(DB(J).GT.BARL) GO TO 165
      GO TO 159
C
C CHECK FOR NORMAL LAP SPLICES
C
 155  IF(B4.GT.B2) GO TO 157
      GO TO 3003
 165  IF(TMODE.EQ.1) WRITE(7,166) BARL
      WRITE(8,166) BARL
      GO TO 3003
 157  IF(TMODE.EQ.1) WRITE(7,160)
      WRITE(8,160)
      GO TO 3003
C
C CHECK FOR TANGENTIAL LAP SPLICES
C
 159  IF(B4.GT.B3) GO TO 161
      GO TO 155
 161  IF(TMODE.EQ.1) WRITE(7,163)
      WRITE(8,163)
      GO TO 3003
C
```

```
C      ***FORMAT STATEMENTS***
C
 200  FORMAT(' ',5X,'TIES ARE REQUIRED ON INTERMEDIATE BARS',
     * ' BECAUSE BAR SPACING'/6X,'EXCEEDS MAXIMUM CLEAR BAR',
     * ' SPACING.'/)
 201  FORMAT(' ',5X,'OMIT TIES ON INTERMEDIATE BARS BECAUSE BAR'
     * /6X,'SPACING DOES NOT EXCEED MAXIMUM CLEAR BAR SPACING'/)
 202  FORMAT(' ',5X,'TIES ARE REQUIRED ON INTERMEDIATE BARS IN',
     * ' XFACE BECAUSE BAR'/6X,'SPACING EXCEEDS MAXIMUM CLEAR',
     * ' BAR SPACING.'/)
 203  FORMAT(' ',5X,'OMIT TIES ON INTERMEDIATE BARS IN XFACE',
     *' BECAUSE BAR'/6X,'SPACING DOES NOT EXCEED MAXIMUM CLEAR BAR',
     * ' SPACING'/)
 204  FORMAT(' ',5X,'TIES ARE REQUIRED ON INTERMEDIATE BARS IN',
     * ' YFACE BECAUSE BAR'/6X,'SPACING EXCEEDS MAXIMUM CLEAR',
     * ' BAR SPACING.'/)
 205  FORMAT(' ',5X,'OMIT TIES ON INTERMEDIATE BARS IN YFACE',
     *' BECAUSE BAR'/6X,'SPACING DOES NOT EXCEED MAXIMUM CLEAR BAR',
     * 'SPACING'/)
 206  FORMAT(' ',5X,'#',I2,1X,'@ ',F6.1,2X,'IN PROJECT UNITS',
     * ' SPACING'/)
 300  FORMAT(' ',5X,'LATERAL TIES REQUIRED FOR INTERIOR BARS',
     * /6X,'IN ALTERNATE FACE'/)
 301  FORMAT(' ',5X,'LATERAL TIES REQUIRED FOR INTERIOR BARS',
     * /6X,'IN X-FACE'/)
 302  FORMAT(' ',5X,'LATERAL TIES REQUIRED FOR INTERIOR BARS',
     * /6X,'IN Y-FACE'/)
 304  FORMAT(' ','SELECT TIES :'/)
 160  FORMAT(' ',5X,'NORMAL LAP SPLICES CAN BE USED'/)
 163  FORMAT(' ',5X,'TANGENTIAL LAP SPLICES CAN BE USED'/)
 166  FORMAT(' ',5X,'LAP SPLICES SHOULD NOT BE USED',
     * ' FOR DIAMETER'/6X,'BAR LARGER THAN',F9.4,1X,
     * 'IN PROJECT UNITS'/)
C
      END
C
C  ************************************************************
C  *                                                          *
      SUBROUTINE OUTPUT
C  *                                                          *
C  *THIS SUBROUTINE IS FORMATTED A SHORT OUTPUT LAYOUT        *
C  *                                                          *
C  ************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PO,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
C
C ONLY USED FOR REVISING PROBLEM IN INTERACTIVE MODE
C
      IF(TREVIS.EQ.1) GO TO 206
      IF(TREVIS.EQ.2) GO TO 213
      IF(TREVIS.EQ.3) GO TO 213
      IF(TREVIS.EQ.4) GO TO 208
      IF(TREVIS.EQ.5) GO TO 211
C
C ONLY USED FOR ADEQUACY CHECK IN MULTIPLE LOAD CASES
C
      IF(TDESIG.EQ.2) GO TO 203
      IF(TYPE.EQ.1.AND.TCODE.EQ.1) GO TO 227
      GO TO 231
 227  WRITE(6,100)
      IF(TMODE.EQ.1) WRITE(7,100)
 231  IF(TYPE.EQ.1.AND.TCODE.EQ.2) GO TO 228
      GO TO 232
 228  WRITE(6,101)
      IF(TMODE.EQ.1) WRITE(7,101)
 232  IF(TYPE.EQ.2.AND.TCODE.EQ.1) GO TO 229
      GO TO 233
 229  WRITE(6,102)
      IF(TMODE.EQ.1) WRITE(7,102)
 233  IF(TYPE.EQ.2.AND.TCODE.EQ.2) GO TO 230
      GO TO 208
 230  WRITE(6,103)
      IF(TMODE.EQ.1) WRITE(7,103)
 208  LX=KX*LUX
      IF(TXBRAC.EQ.1) GO TO 204
      IF(TCODE.EQ.1) GO TO 150
      WRITE(6,109)LX
      IF(TMODE.EQ.1) WRITE(7,109)LX
      GO TO 205
 150  WRITE(6,151)LX
      IF(TMODE.EQ.1) WRITE(7,151)LX
      GO TO 205
 204  IF(TCODE.EQ.1) GO TO 152
      WRITE(6,110)LX
      IF(TMODE.EQ.1) WRITE(7,110)LX
      GO TO 205
 152  WRITE(6,153)LX
      IF(TMODE.EQ.1) WRITE(7,153)LX
 205  IF(TBEND.EQ.1) GO TO 215
      LY=KY*LUY
      IF(TYBRAC.EQ.1) GO TO 207
      IF(TCODE.EQ.1) GO TO 154
      WRITE(6,111)LY
      IF(TMODE.EQ.1) WRITE(7,111)LY
      GO TO 215
 154  WRITE(6,155)LY
      IF(TMODE.EQ.1) WRITE(7,155)LY
      GO TO 215
 207  IF(TCODE.EQ.1) GO TO 156
      WRITE(6,112)LY
      IF(TMODE.EQ.1) WRITE(7,112)LY
      GO TO 215
 156  WRITE(6,157)LY
      IF(TMODE.EQ.1) WRITE(7,157)LY
C
```

```
C ONLY USED FOR REVISING STABILITY GEOM. PROPERTIES IN
C INTERACTIVE MODE
C
 215  IF(TREVIS.EQ.4) GO TO 213
 205  WRITE(6,104)
      IF(TMODE.EQ.1) WRITE(7,104)
      IF(TCODE.EQ.1) GO TO 200
      WRITE(6,105)FPC,WC,EC,FY,ES
      IF(TMODE.EQ.1) WRITE(7,105)FPC,WC,EC,FY,ES
      GO TO 201
 200  WRITE(6,106)FPC,WC,EC,FY,ES
      IF(TMODE.EQ.1) WRITE(7,106)FPC,WC,EC,FY,ES
 201  IF(TYPE.EQ.1) GO TO 202
      IF(TCODE.EQ.1) GO TO 209
      WRITE(6,107) FSY
      IF(TMODE.EQ.1) WRITE(7,107) FSY
      GO TO 203
 209  WRITE(6,113)FSY
      IF(TMODE.EQ.1) WRITE(7,113)FSY
      GO TO 203
 202  IF(TCODE.EQ.1) GO TO 210
      WRITE(6,108)FSY
      IF(TMODE.EQ.1) WRITE(7,108)FSY
      GO TO 203
 210  WRITE(6,114)FSY
      IF(TMODE.EQ.1) WRITE(7,114)FSY
C
C ONLY USED FOR REVISING MATERIAL PROPERTIES IN INTERACTIVE MODE
C
 203  IF(TREVIS.EQ.1) GO TO 213
 211  IF(TDESIG.EQ.1) WRITE(6,2880)
      IF(TDESIG.EQ.2) WRITE(6,2881)
      IF(TDESIG.EQ.3) WRITE(6,2882)
      IF(TCODE.EQ.1) GO TO 212
      MPU=PU/1000.
      MMX2=ABS(MX2)/12000.
      MMY2=ABS(MY2)/12000.
      WRITE(6,115)MPU,MMX2,MMY2
      IF(TMODE.EQ.1) WRITE(7,115)MPU,MMX2,MMY2
      GO TO 214
 212  MPU=PU/1000.
      MMX2=ABS(MX2)/1000000.
      MMY2=ABS(MY2)/1000000.
      WRITE(6,116)MPU,MMX2,MMY2
      IF(TMODE.EQ.1) WRITE(7,116)MPU,MMX2,MMY2
 214  IF(TDESIG.EQ.2) GO TO 226
 213  WRITE(6,117)
      IF(TMODE.EQ.1) WRITE(7,117)
      IF(TCODE.EQ.1) GO TO 216
      IF(TSHAPE.EQ.3) GO TO 2883
      WRITE(6,118)CX,CY
      IF(TMODE.EQ.1) WRITE(7,118)CX,CY
      GO TO 217
 2883 WRITE(6,2884) CX
      IF(TMODE.EQ.1) WRITE(7,2884) CX
      GO TO 217
 216  IF(TSHAPE.EQ.3) GO TO 2885
      WRITE(6,119)CX,CY
      IF(TMODE.EQ.1) WRITE(7,119)CX,CY
      GO TO 217
 2885 WRITE(6,2886) CX
      IF(TMODE.EQ.1) WRITE(7,2886) CX
C
C ONLY USED FOR EXISTING DIAGNOSED ERROR IN SUBROUTINE DIMEN)
C
 217  IF(TESTDI.EQ.2) GO TO 226
      IF(TRFACE.EQ.3) NMAX=2*(NX+NY)
      PGDES=(NMAX*PRIME*DB(J)**2)/(AG*4.)
      WRITE(6,120)PGDES
      IF(TMODE.EQ.1) WRITE(7,120)PGDES
      GO TO(218,219,220,221),TRFACE
 218  WRITE(6,121)NMAX,TBS(J)
      IF(TMODE.EQ.1) WRITE(7,121)NMAX,TBS(J)
      GO TO 222
 219  WRITE(6,122)NMAX,TBS(J)
      IF(TMODE.EQ.1) WRITE(7,122)NMAX,TBS(J)
      GO TO 222
 220  NNX=2*NX
      NNY=2*NY
      WRITE(6,123)NNX,TBS(J),NNY,TBS(J)
      IF(TMODE.EQ.1) WRITE(7,123)NNX,TBS(J),NNY,TBS(J)
      GO TO 222
 221  WRITE(6,124)NMAX,TBS(J)
      IF(TMODE.EQ.1) WRITE(7,124)NMAX,TBS(J)
 222  IF(SPACE.LE.O..AND.PITCH.LE.O.) GO TO 226
      IF(TYPE.EQ.1) GO TO 223
      IF(TCODE.EQ.1) GO TO 224
      WRITE(6,126)TDSSS,PITCH,NSPACR
      IF(TMODE.EQ.1) WRITE(7,126)TDSSS,PITCH,NSPACR
      GO TO 226
 224  WRITE(6,127)TDSSS,PITCH,NSPACR
      IF(TMODE.EQ.1) WRITE(7,127)TDSSS,PITCH,NSPACR
      GO TO 226
 223  IF(TCODE.EQ.1) GO TO 225
      WRITE(6,128)TDSSS,SPACE
      IF(TMODE.EQ.1) WRITE(7,128)TDSSS,SPACE
      GO TO 226
 225  WRITE(6,129)TDSSS,SPACE
      IF(TMODE.EQ.1) WRITE(7,129)TDSSS,SPACE
 226  WRITE(6,130)
      IF(TMODE.EQ.1) WRITE(7,130)
C
C EVALUATE THE ENTERED VALUES OF MATERIAL STRENGTH
C
      IF(TCODE.EQ.1) GO TO 545
      IF(FPC.GE.2000..AND.FPC.LE.6000.) GO TO 540
      WRITE(6,530)
 540  IF(FY.GE.40000..AND.FY.LE.60000.) GO TO 541
      WRITE(6,531)
 541  IF(FSY.GE.40000..AND.FSY.LE.60000.) GO TO 542
      WRITE(6,532)
      GO TO 542
 545  IF(FPC.GE.20..AND.FPC.LE.40.) GO TO 543
      WRITE(6,530)
 543  IF(FY.GE.300..AND.FY.LE.400.) GO TO 544
      WRITE(6,531)
 544  IF(FSY.GE.300..AND.FSY.LE.400.) GO TO 542
      WRITE(6,532)
C
C DETERMINE COLUMN DEFLECT IN SINGLE OR DOUBLE CURVATURE
```

```
C
 542  IF(ABS(MX1).LE.0.) GO TO 328
      IF(MX1.LT.0..AND.MX2.GT.0.) GO TO 327
      IF(MX2.LT.0..AND.MX1.GT.0.) GO TO 327
      WRITE(6,303)
      IF(TMODE.EQ.1) WRITE(7,303)
      GO TO 328
 327  WRITE(6,304)
      IF(TMODE.EQ.1) WRITE(7,304)
 328  IF(TBEND.EQ.1) GO TO 500
      IF(ABS(MY1).LE.0.) GO TO 1000
      IF(MY1.LT.0..AND.MY2.GT.0.) GO TO 329
      IF(MY2.LT.0..AND.MY1.GT.0.) GO TO 329
 1000 WRITE(6,305)
      IF(TMODE.EQ.1) WRITE(7,305)
      GO TO 500
 329  WRITE(6,306)
      IF(TMODE.EQ.1) WRITE(7,306)
C
C EMPTY STORAGE PARAMETERS
C
 500  PITCH=0.
      SPACE=0.
      RETURN
C
C     FORMAT
C
  100 FORMAT(' ','TYPE = TIED',15X,'DESIGN CODE = CSA A23.3-M77'/)
  101 FORMAT(' ','TYPE = TIED',15X,'DESIGN CODE = ACI 318-77'/)
  102 FORMAT(' ','TYPE = SPIRAL',15X,'DESIGN CODE = CSA A23.3-M77'/)
  103 FORMAT(' ','TYPE = SPIRAL',15X,'DESIGN CODE = ACI 318-77'/)
  104 FORMAT(' ','MATERIAL PROPERTIES :'/)
  105 FORMAT(' ',5X,'CONCRETE     : COMP. STRENGTH   =',F7.0,
     *  1X,'PSI'/6X,'                DENSITY          =',F6.0,
     *  1X,'PCF'/6X,'                MODULUS OF ELAS. =',F11.0,
     *  1X,'PSI'/6X,'LONG. STEEL  : YIELD STRESS     =',F8.0,
     *  1X,'PSI'/6X,'                MODULUS OF ELAS. =',F11.0,'PSI')
  106 FORMAT(' ',5X,'CONCRETE     : COMP. STRENGTH   =',F5.0,
     *  1X,'MPA'/6X,'                DENSITY          =',F7.0,
     *1X,'KG/CUBIC METER',
     *          /6X,'                MODULUS OF ELAS. =',F8.0,
     *  1X,'MPA'/6X,'LONG. STEEL  : YIELD STRESS     =',F6.0,
     *  1X,'MPA'/6X,'                MODULUS OF ELAS. =',F9.0,'MPA')
  107 FORMAT(' ',5X,'SPIRAL       : YIELD STRESS     =',F8.0,
     *1X,'PSI'/)
  108 FORMAT(' ',5X,'TIES         : YIELD STRESS     =',F8.0,
     *1X,'PSI'/)
  113 FORMAT(' ',5X,'SPIRAL       : YIELD STRESS     =',F6.0,
     *1X,'MPA'/)
  114 FORMAT(' ',5X,'TIES         : YIELD STRESS     =',F6.0,
     *1X,'MPA'/)
  109 FORMAT(8X,'ABOUT X-AXIS : UNBRACED, EFF. LENGTH =',F7.0,
     *1X,'IN.'/)
  110 FORMAT(8X,'ABOUT X-AXIS : BRACED, EFF. LENGTH =',F7.0,
     *1X,'IN.'/)
  111 FORMAT(8X,'ABOUT X-AXIS : UNBRACED, EFF. LENGTH =',F7.0,
     *1X,'IN.'/)
  112 FORMAT(8X,'ABOUT Y-AXIS : BRACED, EFF. LENGTH =',F7.0,
     *1X,'IN.'/)
  151 FORMAT(8X,'ABOUT X-AXIS : UNBRACED, EFF. LENGTH =',F7.0,
     *1X,'MM'/)
  153 FORMAT(8X,'ABOUT X-AXIS : BRACED, EFF. LENGTH =',F7.0,
     *1X,'MM'/)
  155 FORMAT(8X,'ABOUT Y-AXIS : UNBRACED, EFF. LENGTH =',F7.0,
     *1X,'MM'/)
  157 FORMAT(8X,'ABOUT Y-AXIS : BRACED, EFF. LENGTH =',F7.0,
     *1X,'MM'/)
  115 FORMAT(' ',5X,'PU  =',F8.0,1X,
     *'KIPS'/6X,'MUX =',F8.0,1X,'KIP-FT,',2X,'MUY =',F8.0,1X,
     *'KIP-FT'/)
  116 FORMAT(' ',5X,'PU =',F8.0,1X,'KN',
     */6X,'MUX =',F8.0,1X,'KN-M,',2X,'MUY =',F8.0,1X,'KN-M'/)
  117 FORMAT(' ','CONCRETE DIMENSION :'/)
  118 FORMAT(' ',5X,'CX =',F6.1,1X,'IN.,',6X,'CY =',F6.1,1X,'IN.'/)
  119 FORMAT(' ',5X,'CX =',F6.0,1X,'MM,',6X,'CY =',F6.0,1X,'MM'/)
  120 FORMAT(' ','LONGITUDINAL REINFORCEMENT :',5X,
     *'REINFORCEMENT RATIO =',F5.3/)
  121 FORMAT(' ',5X,I2,'- #',I2,1X,'BARS IN X-FACES ONLY'/)
  122 FORMAT(' ',5X,I2,'- #',I2,1X,'BARS IN Y-FACES ONLY'/)
  123 FORMAT(' ',5X,I2,'- #',I2,1X,'BARS IN X-FACES'/6X,
     *I2,'- #',I2,1X,'BARS IN Y-FACES'/)
  124 FORMAT(' ',5X,I2,'- #',I2,1X,'BARS IN CIRCULAR'/)
  126 FORMAT(' ','SPIRALS :'//' ',5X,'#',I2,1X,'WITH',
     * F5.1,1X,'IN. PITCH AND',I2,1X,'SPACERS'/)
  127 FORMAT(' ','SPIRALS :'//' ',5X,'#',I2,1X,'WITH',
     * F6.0,1X,'MM PITCH AND',I2,1X,'SPACERS'/)
  128 FORMAT(' ','TIES :'//' ',5X,'#',I2,1X,'@',F5.1,1X,'IN.',
     *' SPACING'/)
  129 FORMAT(' ','TIES :'//' ',5X,'#',I2,1X,'@',F6.0,1X,'MM',
     *' SPACING'/)
  130 FORMAT(' ','ADDITIONAL COMMENTS :'/)
  303 FORMAT(' ',5X,'COLUMN DEFLECTS IN SINGLE',
     * ' CURVATURE ABOUT X-AXIS'/)
  304 FORMAT(' ',5X,'COLUMN DEFLECTS IN DOUBLE',
     * ' CURVATURE ABOUT X-AXIS'/)
  305 FORMAT(' ',5X,'COLUMN DEFLECTS IN SINGLE',
     * ' CURVATURE ABOUT Y-AXIS'/)
  306 FORMAT(' ',5X,'COLUMN DEFLECTS IN DOUBLE',
     * ' CURVATURE ABOUT Y-AXIS'/)
  530 FORMAT(' ',5X,'THE ENTERED CONCRETE STRENGTH IS NOT',
     * ' WITHIN THE'/6X,'PREDETERMINED RANGE'/)
  531 FORMAT(' ',5X,'THE ENTERED LONGITUDINAL YIELD STRESS',
     * ' IS NOT WITHIN THE'/6X,'PREDETERMINED RANGE'/)
  532 FORMAT(' ',6X,'THE ENTERED SPIRAL YIELD STRESS IS NOT',
     * ' WITHIN THE'/6X,'PREDETERMINED RANGE'/)
 2880 FORMAT(' ','DESIGN LOAD CONDITION :'/)
 2881 FORMAT(' ','ADDITIONAL LOAD CONDITION :'/)
 2882 FORMAT(' ','CHECK LOAD CONDITION :'/)
 2884 FORMAT(' ',5X,'DIAMETER = ',F6.1,1X,'IN.'/)
 2886 FORMAT(' ',5X,'DIAMETER = ',F6.0,1X,'MM'/)
C
      END
C
C   ************************************************************
C   *                                                          *
      SUBROUTINE PROBL1
C   *                                                          *
C   *THIS SUBROUTINE READS OPTIONAL CONTROL PARAMETER          *
C   *                                                          *
C   ************************************************************
```

```
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N,$)
      COMMON PG,ESMAX,TYPE,TREINF,TSLEN1,TBEND,TSLEN2,TADEQ,TPROB
      COMMON TCODE,WC,ES,DCC,DSS,CXINCR,CYINCR,TI,TK,TBS(9),DB(9),
     * FD,FL,ECU,PMAX,PMIN,SB1,EMIN,PHIV,RFAC,DSMIN,SSMIN,SSMAX,
     * TBMAX,SMAX,J,PRIME,MINDIM,TESTDI,LATDS(20),MINLAT,TBMAXX
      COMMON FPC,FY,FSY,EC,B1,TXBRAC,KX,LUX,TYBRAC,KY,LUY,SPU,
     * SPC,PD,PL,PU,MX1,MX2,BDX,MY1,MY2,BDY,TOUT,TMODE
      COMMON TSHAPE,CX,CY,R,TDESIG,TRFACE,NBS,NMAX
      COMMON TDIMFG,AG,AS,CG,SB,DP,AST,NX,NY,SBX,SBY,DS,
     * Y(12),ISEX,X(12),ISEY,TMODAS,PGDES,EY,MCX,
     * MAPPX,PUDESX,MUDESX,PBX
      COMMON EX,MCY,MAPPY,PUDESY,MUDESY,PBY,MOYREQ,MOXREQ,
     * YC,YB,ECS(12),ETS(12),CS(12),FT(12),
     * XC,XB,ECSY(12),ETSY(12),CSY(12),FTY(12)
      COMMON TESTMO
      COMMON NLC,JPOB,TREVIS,TDSSS,SPACE,PITCH,NSPACR
C
      DIMENSION A(9)
C
C EMPTY STORAGE PARAMETERS
C
      TMODAS=0
      TREVIS=0
      TPROB=0
C
      IF(TMODE.EQ.1) GO TO 1
      GO TO 9
 1    WRITE(7,101)
 101  FORMAT('& NEXT OPTION= ? REVISE ABOVE PROBLEM,ENTER 1'/,
     */'  .                    GO TO NEXT PROBLEM,ENTER 2',
     */'                       TERMINATION OF PROGRAM,ENTER 3'/)
      READ(4,1000) JPOB
      GO TO (500,501,502),JPOB
 502  TPROB=6
      GO TO 6
 501  WRITE(7,102)
 102  FORMAT('& NEXT OPTION= ? ALL DATA REENTERED,ENTER 1'/,
     */'      FOLLOWING CONCRETE GEOM. VARIED WITH HEADING,ENTER 2',
     */'      FOLLOWING STEEL GEOM. VARIED WITH HEADING,ENTER 3',
     */'      FOLLOWING STABILITY GEOM. VARIED WITH HEAD.,ENTER 4',
     */'      FOLLOWING LOADING PROP. VARIED WITH HEADING,.ENTER 5',
     */'                LAST COLUMN DESIGN PROBLEM,ENTER 6'/)
      READ(4,1000)TPROB
      GO TO 6
 500  WRITE(7,103)
 103  FORMAT('& REVISE OPTION= ? ALTER MATERIAL ONLY,ENTER 1'/,
     */'                         ALTER CONCRETE GEOM. ONLY,ENTER 2',
     */'                         ALTER STEEL GEOM. ONLY,ENTER 3',
     */'                         ALTER STABILITY GEOM. ONLY,ENTER 4',
     */'                         ALTER LOADS ONLY,ENTER 5'/)
      READ(4,1000)TREVIS
      GO TO (503,504,505,506,507),TREVIS
 503  WRITE(8,401)
      GO TO 6
 504  WRITE(8,402)
      GO TO 6
 505  WRITE(8,403)
      GO TO 6
 506  WRITE(8,404)
      GO TO 6
 507  WRITE(8,405)
      GO TO 6
 9    READ(5,100) TPROB
C
 100  FORMAT(I2)
 1000 FORMAT(I1)
 401  FORMAT(' ','REVISE OPTION = ALTER MATERIAL',
     * ' PROPERTIES ONLY'/)
 402  FORMAT(' ','REVISE OPTION = ALTER CONCRETE',
     * ' OF GEOMETRY PROPERTIES ONLY'/)
 403  FORMAT(' ','REVISE OPTION = ALTER STEEL OF',
     * ' GEOMETRY PROPERTIES ONLY'/)
 404  FORMAT(' ','REVISE OPTION = ALTER STABILITY',
     * ' OF GEOMETRY PROPERTIES ONLY'/)
 405  FORMAT(' ','REVISE OPTION = ALTER LOADS PROPERTIES',
     * ' ONLY'/)
C
 6    RETURN
      END
C
C  ***************************************************************
C  *                                                             *
      SUBROUTINE SORT(HEAD,N,JPOB)
C  *                                                             *
C  *THIS SUBROUTINE IS TO SORT THE COMPUTER RESULTS SO THAT      *
C  *A HEADING WILL BE PRINTED ON THE TOP OF EVERY NEW PAGE.      *
C  *ASSUMPTION : THE PRINTER TRUNCATES ANY EXTRA CHARACTERS      *
C  *IF THE LENGTH OF A PRINTED LINE HAS EXCEEDED DEFAULT LIMIT   *
C  *                                                             *
C  ***************************************************************
C
      IMPLICIT REAL(A-H,I,K-M,P,R,S,W-Y),INTEGER(T,J,N)
      DIMENSION LINE(50),HEAD(15)
C
      ENDFILE 8
      REWIND 8
      TNJ=8
 307  IF(TNJ.EQ.8) GO TO 407
      ENDFILE 8
      REWIND 8
C
      ENDFILE 6
      REWIND 6
C
C NUMBER IS THE TOTAL NUMBER OF LINES PRINTED ON EACH PAGE
C
      NUMBER=58
      LOCA=3
      NPAGE=2
  507 READ(5,987,END=707)LINE
      LOCA=LOCA+1
      IF (LOCA .GT. NUMBER) GO TO 807
      WRITE(9, 987)LINE
      GO TO 507
  807 WRITE(9, 986)HEAD,NPAGE
      WRITE(9, 987)LINE
      LOCA=5
      NPAGE=NPAGE+1
      GO TO 507
```

```
  986 FORMAT('1',1X,15A4,'PAGE',I3/1X,60('*')/)
  987 FORMAT(50A4)
  707 REWIND 6
      REWIND 8
      RETURN
C
 407  READ(8,987,END=808)LINE
      WRITE(6,987)LINE
      GO TO 407
 808  TNJ=10
      IF(JPOB.EQ.1) GO TO 100
      IF(N.GT.1) GO TO 110
      GO TO 307
 110  N=N-1
 100  ENDFILE 8
      REWIND 6
      RETURN
      END
End of file
```

<u>*RECENT STRUCTURAL ENGINEERING REPORTS*</u>

Department of Civil Engineering

University of Alberta

84. *Finite Element Programs for Frame Analysis* by M. El Zanaty and D.W. Murray, April 1980.

85. *Test of a Prestressed Concrete Secondary Containment Structure* by J.G. MacGregor, S.H. Simmonds and S.H. Rizkalla, April 1980.

86. *An Inelastic Analysis of the Gentilly-2 Secondary Containment Structure* by D.W. Murray, C. Wong, S.H. Simmonds and J.G. MacGregor, April 1980.

87. *Nonlinear Analysis of Axisymmetric Reinforced Concrete Structures* by A.A. Elwi and D.W. Murray, May 1980.

88. *Behavior of Prestressed Concrete Containment Structures - A Summary of Findings* by J.G. MacGregor, D.W. Murray, S.H. Simmonds, April 1980.

89. *Deflection of Composite Beams at Service Load* by L. Samantaraya and J. Longworth, June 1980.

90. *Analysis and Design of Stub-Girders* by T.J.E. Zimmerman and R. Bjorhovde, August 1980.

91. *An Investigation of Reinforced Concrete Block Masonry Columns* by G.R. Sturgeon, J. Longworth and J. Warwaruk, September 1980.

92. *An Investigation of Concrete Masonry Wall and Concrete Slab Interaction* by R.M. Pacholok, J. Warwaruk and J. Longworth, October 1980.

93. *FEPARCS5 - A Finite Element Program for the Analysis of Axisymmetric Reinforced Concrete Structures - Users Manual* by A. Elwi and D.W. Murray, November 1980.

94. *Plastic Design of Reinforced Concrete Slabs* by D.M. Rogowsky and S.H. Simmonds, November 1980.

95. *Local Buckling of W Shapes Used as Columns, Beams, and Beam-Columns* by J.L. Dawe and G.L. Kulak, March 1981.

96. *Dynamic Response of Bridge Piers to Ice Forces* by E.W. Gordon and C.J. Montgomery, May 1981.

97. *Full-Scale Test of a Composite Truss* by R. Bjorhovde, June 1981.

98. *Design Methods for Steel Box-Girder Support Diaphragms* by R.J. Ramsay and R. Bjorhovde, July 1981.

99. *Behavior of Restrained Masonry Beams* by R. Lee, J. Longworth and J. Warwaruk, October 1981.

100. *Stiffened Plate Analysis by the Hybrid Stress Finite Element Method* by M.M. Hrabok and T.M. Hrudey, October 1981.

101. *Hybslab - A Finite Element Program for Stiffened Plate Analysis* by M.M. Hrabok and T.M. Hrudey, November 1981.

102. *Fatigue Strength of Trusses Made From Rectangular Hollow Sections* by R.B. Ogle and G.L. Kulak, November 1981.

103. *Local Buckling of Thin-Walled Tubular Steel Members* by M.J. Stephens, G.L. Kulak and C.J. Montgomery, February 1982.

104. *Test Methods for Evaluating Mechanical Properties of Waferboard: A Preliminary Study* by M. MacIntosh and J. Longworth, May 1982.

105. *Fatique Strength of Two Steel Details* by K.A. Baker and G.L. Kulak, October 1982.

106. *Designing Floor Systems for Dynamic Response* by C.M. Matthews, C.J. Montgomery and D.W. Murray, October 1982.

107. *Analysis of Steel Plate Shear Walls* by L. Jane Thorburn, G.L. Kulak, and C.J. Montgomery, May 1983.

108. *Analysis of Shells of Revolution* by N. Hernandez and S.H. Simmonds, August 1983.

109. *Tests of Reinforced Concrete Deep Beams* by D.M. Rogowsky, J.G. MacGregor and S.Y. Ong, September 1983.

110. *Shear Strength of Deep Reinforced Concrete Continuous Beams* by D.M. Rogowsky and J.G. MacGregor, September 1983.

111. *Drilled-In Inserts in Masonry Construction* by M.A. Hatzinikolas, R. Lee, J. Longworth and J. Warwaruk, October 1983.

112. *Ultimate Strength of Timber Beam Columns* by T.M. Olatunji and J. Longworth, November 1983.

113. *Lateral Coal Pressures in a Mass Flow Silo* by A.B.B. Smith and S.H. Simmonds, November 1983.

114. *Experimental Study of Steel Plate Shear Walls* by P.A. Timler and G.L. Kulak, November 1983.

115. *End Connection Effects on the Strength of Concrete Filled HSS Columns* by S.J. Kennedy and J.G. MacGregor, April 1984.

116. *Reinforced Concrete Column Design Program* by C-K. Leung and S.H. Simmonds, April 1984.