TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/TX-83/18+201-1F		

4. Title and Subtitle	5. Report Date / 6. Performing Organization Code
STRUCTURAL EVALUATION OF BRIDGES FOR OVERLOAD CONDITIONS	5. Report Date: August 1983 6. Performing Organization Code:
7. Author(s) Robert E. Cornwell, Richard W. Stolleis, and C. P. Johnson	8. Performing Organization Report No. Research Report 201-1F
9. Performing Organization Name and Address Center for Transportation Research The University of Texas at Austin Austin, Texas 78712-1075	10. Work Unit No. 11. Contract or Grant No. Research Study 3-5-77-201
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation; Transportation Planning Division P. O. Box 5051 Austin, Texas 78763	13. Type of Report and Period Covered Final 14. Sponsoring Agency Code

15. Supplementary Notes

Study conducted in cooperation with the U. S. Department of Transportation, Federal Highway Administration. Research Study Title: "Structural Evaluation of Existing Bridges for Load Rating and Carrying Capacity"

16. Abstract

This research focused on developing a pre-processor (data generator) and a post-processor for an existing computer program, SLAB49, for the analysis of integral grid-beam systems. The pre- and post-processors were specialized for four widely used bridge classes: (1) steel beam and slab section, (2) prestressed girder and slab section, (3) slab section with or without integral curbs, and (4) simple span pan-formed sections. Regularly skewed bridges may also be considered.

The pre-processor takes simplified inputs with regard to bridge type, geometry, and loading and provides automatically the more detailed information required by the parent program SLAB49. The post-processor searches through the results from SLAB49 and provides a more convenient arrangement for output thereby expediting the interpretation of the results of the analysis. The pre-processor accesses data bases containing bridge class information and overload configurations. A computer program was developed which allows for the data bases to be updated for future implementation as needed.

17. Key Words	18. Distribution Statement
bridge, grid-beam, integral, systems, computer program, structural evaluation, overload	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	290	

Form DOT F 1700.7 (8-69)

STRUCTURAL EVALUATION OF BRIDGES FOR OVERLOAD CONDITIONS

by

Robert E. Cornwell
Richard W. Stolleis
C. P. Johnson

Research Report Number 201-1F

Structural Evaluation of Existing Bridges for Load Rating and Carrying Capacity

Research Project 3-5-77-201

conducted for

Texas
State Department of Highways and Public Transportation

in cooperation with the
U. S. Department of Transportation
Federal Highway Administration

by the

CENTER FOR TRANSPORTATION RESEARCH

BUREAU OF ENGINEERING RESEARCH

THE UNIVERSITY OF TEXAS AT AUSTIN

August 1983

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine, manufacture, design or composition of matter, or any new and useful improvement thereof, or any variety of plant which is or may be patentable under the patent laws of the United States of America or any foreign country.

PREFACE

This report summarizes the major developments resulting from the research project "Structural Evaluation of Existing Bridges for Load Rating and Carrying Capacity." In this report we present the pre- and post-processors for SLAB49 and the data base generator. Input guides, program documentations, listings, and an example problem are also contained in this report.

Several individuals have made contributions to this research. The authors thank Mr. Hugh Thompson, Mr. Roland Eichman, and Ms. Azam Waugh for their contributions. The interest shown by John Panak, Ralph Banks, and Fred Herber and their inputs during the course of this work are appreciated. The help given by the staff of the Center for Transportation Research in producing this report is also appreciated. In particular we wish to thank Art Frakes for editing the report and Kay Lee for typing the draft and final copy of this report.

Robert Cornwell
Richard Stolleis
C. P. Johnson

March 1981

ABSTRACT

This research focused on developing a pre-processor (data generator) and a post-processor for an existing computer program, SLAB49, for the analysis of integral grid-beam systems. The pre- and post-processors were specialized for four widely used bridge classes: (1) steel beam and slab section, (2) prestressed girder and slab section, (3) slab section with or without integral curbs, and (4) simple span pan-formed sections. Regularly skewed bridges may also be considered.

The pre-processor takes simplified inputs with regard to bridge type, geometry, and loading and provides automatically the more detailed information required by the parent program SLAB49. The post-processor searches through the results from SLAB49 and provides a more convenient arrangement for output thereby expediting the interpretation of the results of the analysis. The pre-processor accesses data bases containing bridge class information and overload configurations. A computer program was developed which allows for the data bases to be updated for future implementation as needed.

SUMMARY

This research centered on the design and testing of pre- and post-processors for the purpose of expediting the structural evaluation of bridges (including those skewed) subjected to overload configurations. The pre-processor allows the engineer to describe the structure and loading in a simple and natural way for four widely used bridge classes. The pre-processor was designed to take abbreviated inputs for each bridge class and then prepare the more detailed information for the parent program SLAB49. The post-processor was specifically designed to give selected outputs from the parent program for the purpose of simplifying the interpretation of the results of the analysis. Options were included for obtaining CAL-COMP plots for input data verification and for graphical display of selected output. The pre-processor accesses data bases for bridge class (main member) information and truck configurations. A data base generator was developed to allow the user to save those main members and truck configurations which are expected to be needed on a frequent basis.

The total package of computer programs (i.e., the pre- and post-processors, the data base generator together with the parent program SLAB49) is herein termed the Texas Bridge Analysis and Rating program (TBAR). TBAR allows for rapid access to analytical tools but with more convenient and simple input and output arrangements than were available at the onset of this work. The most probable application of TBAR is for those structures for which line member analysis methods are limited in treating complex overload conditions. This is especially true for skewed structures. In these cases TBAR can be used to realistically and rapidly evaluate four widely used bridge classes for particular overload conditions.

As a result of this research, an analysis tool requiring minimum human resources is available and provides for rapid and accurate structural evaluation data for existing bridges. It could serve to possibly reduce unnecessary restrictions now being placed on legal loadings as well as providing credible indicators for routing overweight permit loads for particular overloaded configurations. It could also be used to develop more rational distribution factors for line member analysis methods.

IMPLEMENTATION STATEMENT

As a result of this research three computer programs were developed to provide simplified input/output arrangements for the parent computer program SLAB49. They consist of the pre-processor (SLBDG4), the post-processor (SEARCH), and the data base generator (BASGEN). These programs were developed on the CDC 6600 and the Cyber 750/175 at The University of Texas at Austin. The final versions of SLBDG4, SEARCH, and BASGEN have also been adapted, in cooperation with the contact representatives from the Bridge, Maintenance Operations, and Automation Divisions, to the IBM computer facilities of the Texas State Department of Highways and Public Transportation for ongoing use. In regard to implementation, this report contains:

(a) Input Guide for SLAB49 (Appendix A)

(b) Documentation of SLBDG4 (Appendix B)

(c) FORTRAN Listing of SLBDG4 (Appendix C)

(d) Input Guide for BASGEN (Appendix D)

(e) Documentation for BASGEN (Appendix E)

(f) FORTRAN Listing of BASGEN (Appendix F)

(g) Documentation of SEARCH (Appendix G)

(h) FORTRAN Listing of SEARCH (Appendix H)

(i) An example problem (Chapter 3 and Appendices I, J, K, and L).

The three programs listed above, together with the parent program SLAB49, are termed the Texas Bridge Analysis and Rating Program (TBAR). It is envisioned that the Texas Bridge Analysis and Rating Program will be used initially under the supervision of an engineer with background in structural evaluation at a central location. As experience is gained in the use of TBAR and its limitations, it may be implemented on a broader basis at diverse locations under the guidance of the supervising engineer at the central location.

TABLE OF CONTENTS

APPENDICES

CHAPTER 1. INTRODUCTION

Nature of the Problem

Highway departments in Texas and other states regularly face two demanding problems:

(1) evaluation of existing bridges for purposes of posting allowable loads, and

(2) issuing special overweight load permits.

Line member analysis methods that consider a longitudinal strip of the bridge and incorporate distribution values recommended by AASHTO standard specifications are limited in treating complex overload conditions. There is a need therefore to be able to realistically and rapidly evaluate an existing bridge for a particular overload condition in which the bridge cross section and skew can be accurately modelled on a day-to-day basis. This requires rapid access to available analytical tools, but with more convenient and simple input and output arrangements than are presently available.

The Approach

Most of the present analytical methods were designed to be very general, i.e., capable of treating special and complex structures, and were therefore austere. By deferring these structures, the remainder can be classified, and convenient input and output arrangements can be developed for each class. This can be accomplished by developing data generators (i.e., pre-processors) that take simplified inputs with regard to the bridge type, its geometry and loading, and then provide automatically the more detailed information that is required by existing programs. Post-processors can then search the output from the existing program and devise more convenient arrangements for output to expedite the interpretation of the results of the analysis.

Bridge Classes

The bridge classes selected for this work are:

(1) The steel beam and slab section (Fig 1a),

(2) Prestressed girder and slab section (Fig 1b),

(3) The simple span or continuous slab with or without integral curbs (Fig 2a), and

(4) Simple span pan-formed sections (Fig 2b).

These four bridge classes form a large percentage of the bridges in Texas. Pre- and post-processors specifically designed for these classes would enable analyses to be performed for a large number of bridges and would thus expedite the structural evaluation of existing bridges for particular overload conditions.

Analysis Procedure

A discrete element procedure developed in Project 56 under the direction of Professor Hudson Matlock and Mr. John J. Panak was selected from existing programs for this work (Ref 1). It is referred to as SLAB49, the parent program. In this procedure the actual cross section is replaced with an equivalent bending system consisting of a slab with embedded beams. It allows for analytical modelling in two dimensions which enables one to account for the influence of the position of the load configuration in both the transverse and longitudinal directions. It can also account (as a result of this work) for the effect of skew. Experience has shown that cross sections with several repeating main members can be treated effectively by this discrete element procedure. Since this program was highly developed at the onset of this work, no modification of the parent program was required. Thus the task at hand centered on the pre- and post-processors.

The Pre-Processor (SLBDG4)

In designing the pre-processor (SLBDG4), complex cases were deferred, thus allowing the structure and loading to be described in the simplest form. This was made possible by enforcing regularity in main member and diaphragm spacing, uniform skew, etc., and by tailoring the pre-processor for the specific classes mentioned earlier. The basic function of the pre-processor was to take the abbreviated inputs and then prepare the more elaborate tables

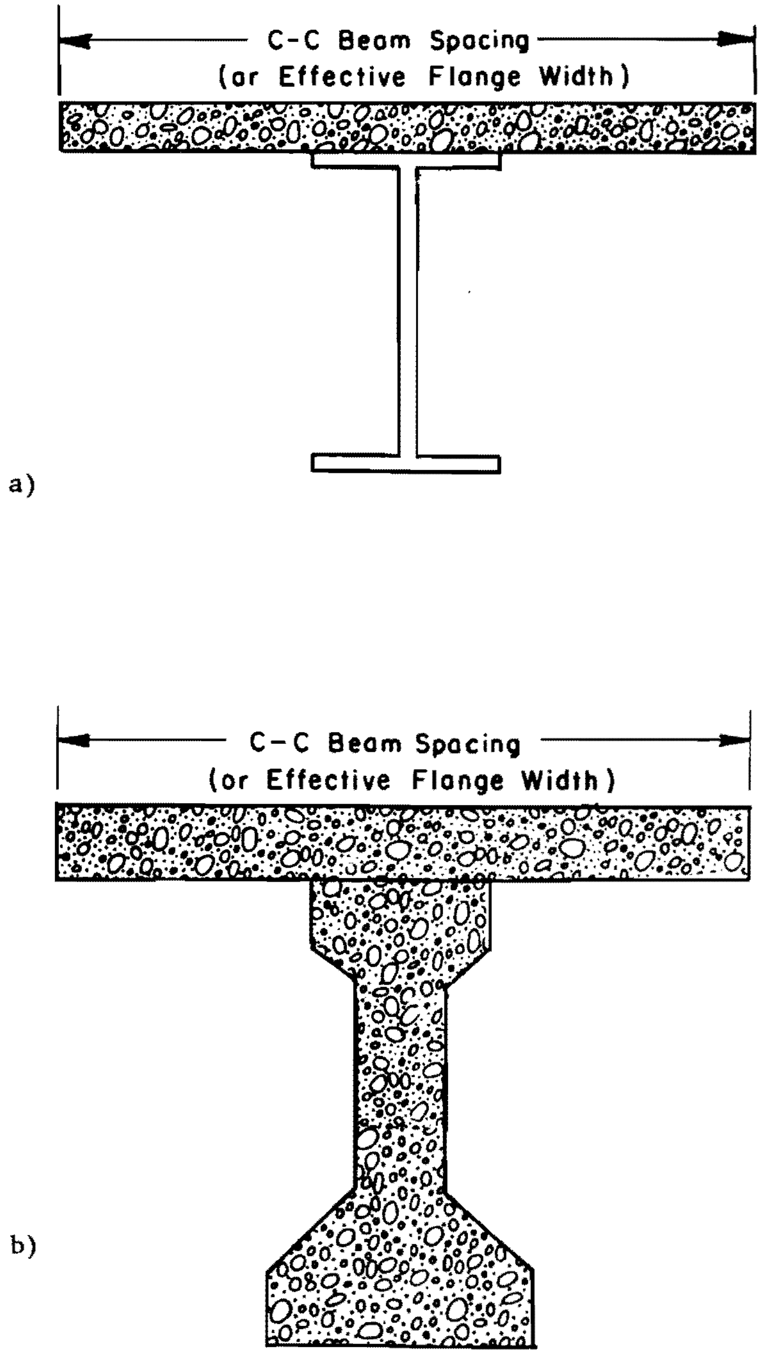


Fig 1. Typical main members of a) steel beam and slab section (class 1) and b) prestressed girder and slab section (class 2).

Fig 2a. Typical slab with or without integral curbs (class 3).

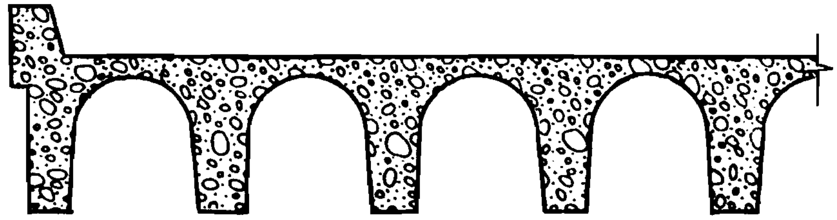


Fig 2b. Typical pan-formed section (class 4).

required by the parent program. The pre-processor is further described in Chapter 2 and the abbreviated inputs for treating each bridge class are described in the Input Guide for SLAB49 with supplementary notes (i.e., Appendix A). A flow chart and detailed documentation of SLBDG4 is contained in Appendix B to allow another programmer to modify this pre-processor. Also, to provide aid in future changes and development, the program has extensive comment cards and all FORTRAN statements use American standards compatible on CDC, IBM, and other similar large computer systems. The FORTRAN listing of SLBDG4 is given in Appendix C.

The Data Bases

Main member types and load types are stored on data bases which are in turn accessed by the pre-processor which prepares the more detailed information required by the parent program (SLAB49). Main member and load types may be easily added to the data base. The data bases, which are considered an important component of this work, are thus expandable, thereby enhancing future implementation. Data bases are prepared by use of the computer program BASGEN as described below.

Preparation of the Data Bases (BASGEN)

The computer program BASGEN was written as a part of this research effort for preparing the data bases which are referred to as:

(1) The Truck Data Base,

(2) Steel Girder Data Base,

(3) Reinforced Girder Data Base, and

(4) Prestressed Concrete Girder Data Base.

The Truck Data Base is used to save a particular truck configuration for later use by the pre-processor (SLBDG4) which allows for a simple way of specifying load placement for a given structure. The particular load configuration consists of a "name," number of wheels, and offsets of each wheel from a user-determined reference point. SLBDG4 is then able to access the particular load configuration from the "name" previously given to it on the data base. SLBDG4 then apportions all wheel loads to the various grid (mesh) points by merely specifying the location of the reference point on the bridge, as described in Chapter 2. CAL-COMP plot routines are included for plotting the plan view of the grid showing truck locations as specified by the simplified user inputs. Routines for checking of load positioning to avoid overlapping of trucks, etc., have been included. The remaining three data bases contain the necessary main member information needed to establish the stiffness properties required by the parent program SLAB49. Again, SLBDG4 is able to access this information from the "name" previously given to it on the data base.

A detailed input guide for preparing the various data bases is contained in Appendix D. Also in Appendix D we show typical information for each data base and define the various quantities in each data base. A detailed documentation of BASGEN is contained in Appendix E. Appendix F contains a FORTRAN listing of the computer program BASGEN.

The Post-Processor (SEARCH)

A post-processor in the form of a computer program called SEARCH was included to facilitate the interpretation of the rather lengthy output from the parent program SLAB49 (Ref 1). It allows for selected outputs to be printed such as the stress along a main member (top and bottom) and the stress at the top of the slab. Also, CAL-COMP plots of these stresses may be obtained if desired. The post-processor is described in Chapter 2 and a flow chart for SEARCH is included in Appendix G. Also, a detailed documentation of SEARCH is contained in Appendix G. The FORTRAN listing of SEARCH is contained in Appendix H.

CHAPTER 2. THE COMPUTER PROGRAMS

General

In the foregoing is described in general terms the nature of this research and the developments that were a result of this work. This section explains in more detail the total package of computer programs comprising this work which is herein termed the Texas Bridge Analysis and Rating Program (TBAR) and some of the important considerations which were necessary in developing TBAR.

The flow of input and output data between the pre-processor, the discrete element program, and the post-processor developed in this project is shown in Fig 3. In step 1, simplified input data is prepared by the engineer which describes the geometric and material properties of the highway bridge and the truck loading on it. Typical inputs are the number of bridge spans, span length, roadway width, concrete strength, and number of girders. Also this input contains the type of truck(s) and their location on the bridge as well as the type of main member comprising the slab-girder system. On the basis of this information, truck and girder data bases are searched and specific information regarding the trucks (number of wheels, wheel location within the configuration, and load per wheel) and main members (slab thickness, girder dimensions, etc.) is retrieved and input to the pre-processor. Specific truck and main member information not contained in the data bases may also be input for special cases as needed.

The pre-processor takes the above inputs and generates two sets of output data as shown in Fig 3. Step 2 is a verification of the input data and contains an echo print of the input data and messages regarding location of trucks and main member information as retrieved from their respective data bases. Step 3 is the primary product of the pre-processor, i.e., the generated input data for the parent program SLAB49, and step 4 is the complex output from SLAB49.

In designing the post-processor it was decided that the primary product of the structural analysis of a highway bridge would be information regarding the stresses in the slab and the top and bottom stresses of the main members.

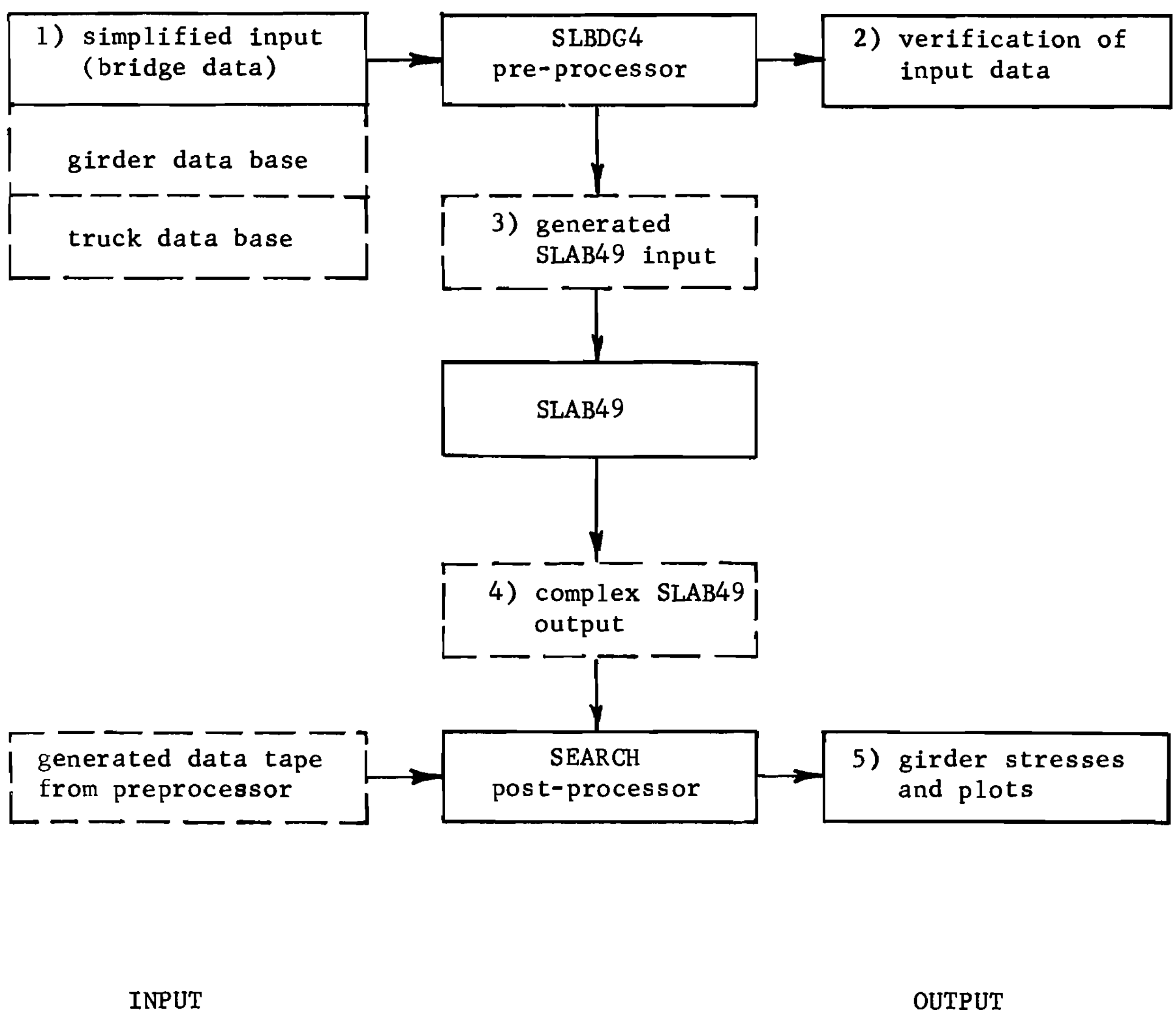


Fig 3. System data flow.

The output information tape (data file) produced by the pre-processor specifies main member output data and transfers physical data for post-processing. This physical data together with the complex output from SLAB49 (step 4) becomes the input for the post-processor. Step 5 in Fig 3 is the post-processor output which is a combination of tabulated stresses, CAL-COMP plots of stresses along the main members of the bridge, and a list of maximum stresses for the specified loading.

Throughout the above process of transferring and manipulating data, the highway engineer is actually involved only with step 1, which is the simplified input for the pre-processor, and steps 2 and 5, which are pre- and post-processor output. The required input for step 1 is described in the input guide of Appendix A. This input guide was designed to allow the frequent user to rapidly fill out the card forms without having to look through pages of variable descriptions. The supplementary notes at the end of the input guide allow the new user to gain familiarity with the input requirements.

Girder Related Calculations

The program calculates stiffnesses for the slab, girders, and composite action of the slab and girder. The slab calculations are made using the formulas and methods described in Center for Highway Research Report 56-25 (Ref 1). The girder stiffness is based on the modulus of elasticity and the moment of inertia of the girder section. The program can model reduced stiffness zones by the use of zones of composite and noncomposite action that can be input in Table 2F of Appendix A. Within the limits of the zones, the slab and girder are considered to act compositely as one section to resist bending. The user can model different structure behaviors by changing the limits of the zones to meet the needs of the analysis. It should be noted that the stresses computed in the post-processor use the same moment of inertia as used to compute the structure stiffnesses; therefore the slab stresses in zones of composite action are computed differently from those in noncomposite zones.

Truck Load Calculations

After the pre-processor has obtained the necessary data, shown in Fig 4, from input or data base, the truck wheel loads are applied to the grid system. The program checks for wheel loads that are located beyond the transverse limits of the bridge and for any trucks that may overlap. Wheel loads that occur beyond the longitudinal limits of the bridge, due to truck placement by user, are assumed to be zero and are not applied to the grid as loads.

Using the data in Fig 4, the global location of the reference, the local wheel coordinates, and the direction of travel, the program calculates the position of each wheel in relation to the grid system. The order of the wheel input is unimportant for the computations, but if wheels are input in some easy-to-follow pattern all users will be able to understand the data without unnecessary confusion. The positions of the trucks shown in Fig 5 provide the basis for application of grid loads.

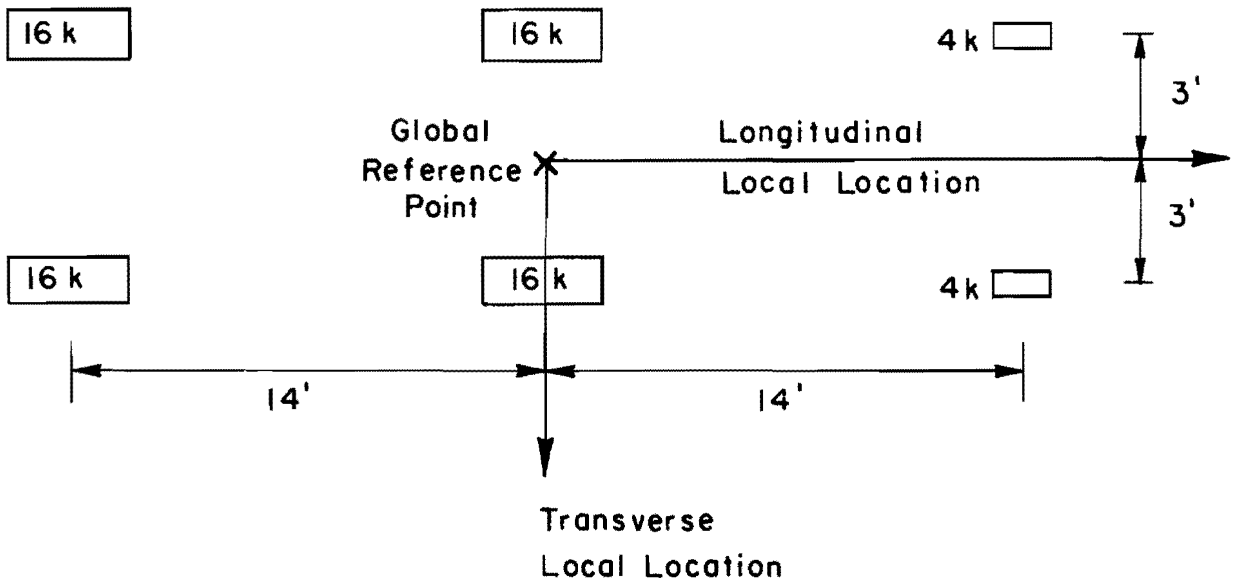
The local coordinates are added to or subtracted from the global reference point to obtain the global location of the wheel. The global wheel location is used to find the four nodes that surround the load, and by the scheme shown in Fig 6, the wheel load is distributed to those surrounding node points. The top portion of Fig 6 shows the computations required to distribute a typical wheel load, where P is the wheel load and the subscripted P's are the grid loads created by the wheel load. The bottom portion of Fig 6 shows a typical apportionment of wheel number 6 of truck number 1.

The Discrete Element Analysis Program

The existing discrete element analysis program, SLAB49, is used as the analytical base of this project. The main purpose of the project is to provide easy-to-use inputs and outputs without changing the structure of existing programs, such as SLAB49. The user of the system programs should be aware of the basic limitations of the discrete element model, and with the reference project report 56-25 (Ref 1), the user can understand the limits of the model and the storage allocations. Storage can be easily modified for larger computer systems in the SLAB49 program.

Description of the Post-Processor

The primary input for the post-processor is the table of resultant moments created by SLAB49. The discrete element moments and a tape generated by the pre-processor provide the post-processor with enough data to convert moments to stresses and provide the user specified outputs that are desired.




```
HS2ØV14       6
    4.        3.       14.
    4.       -3.       14.
   16.        3.        Ø.
   16.       -3.        Ø.
   16.        3.      -14.
   16.       -3.      -14.
```

Fig 4. Local truck coordinates.

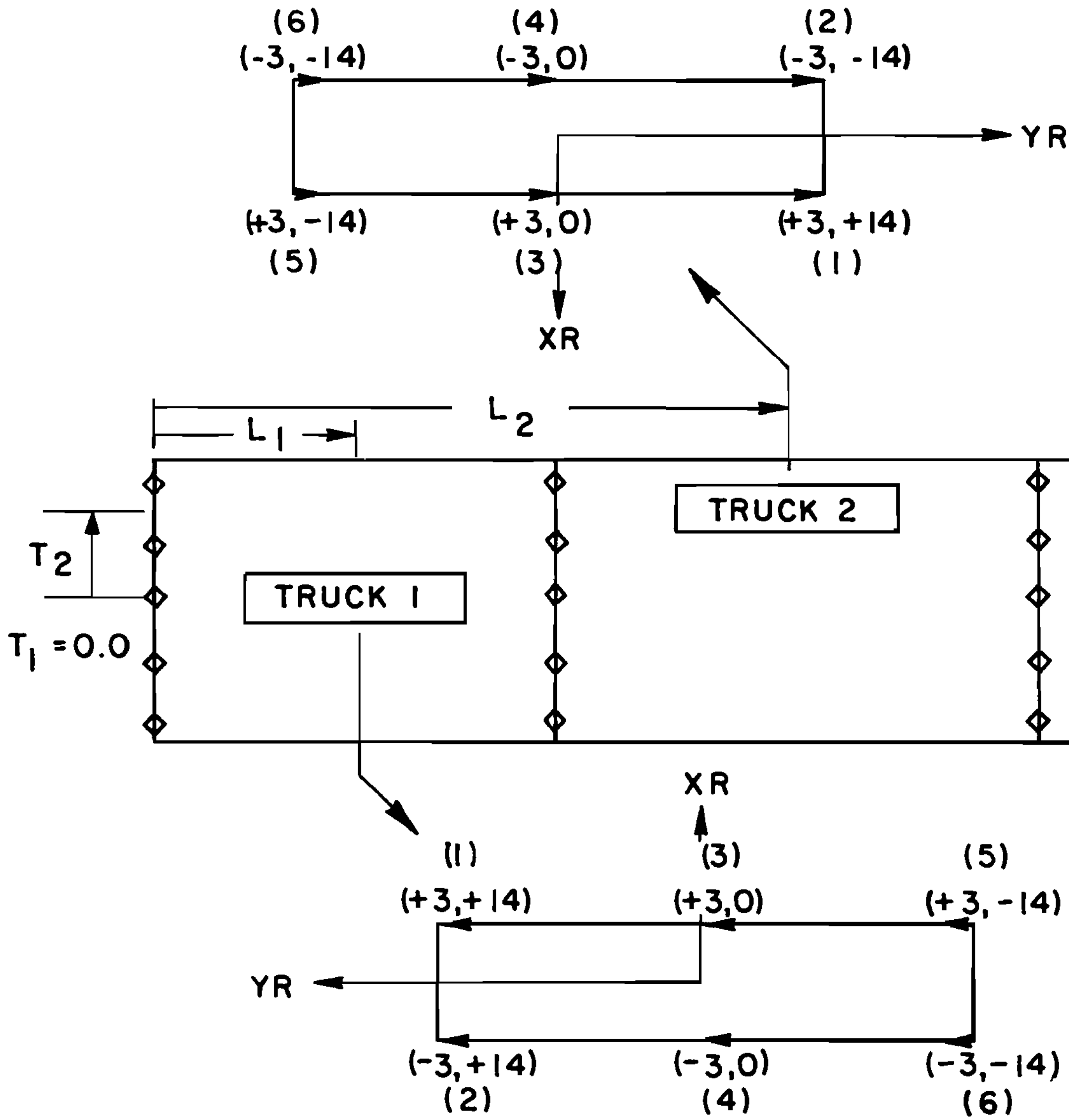
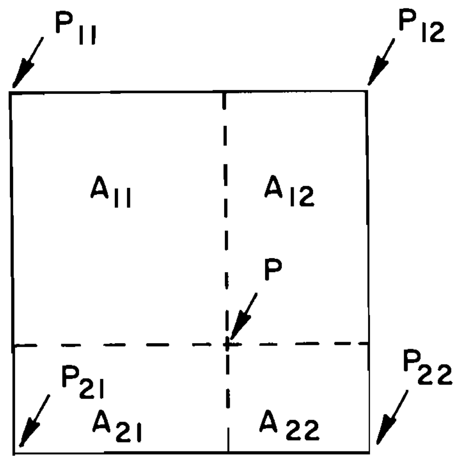


Fig 5. Truck and grid reference positions.



$$P_{22} = P \times (A_{11} / A_t)$$

$$P_{12} = P \times (A_{21} / A_t)$$

$$P_{21} = P \times (A_{12} / A_t)$$

$$P_{11} = P \times (A_{22} / A_t)$$

$$A_t = A_{11} + A_{12} + A_{21} + A_{22}$$

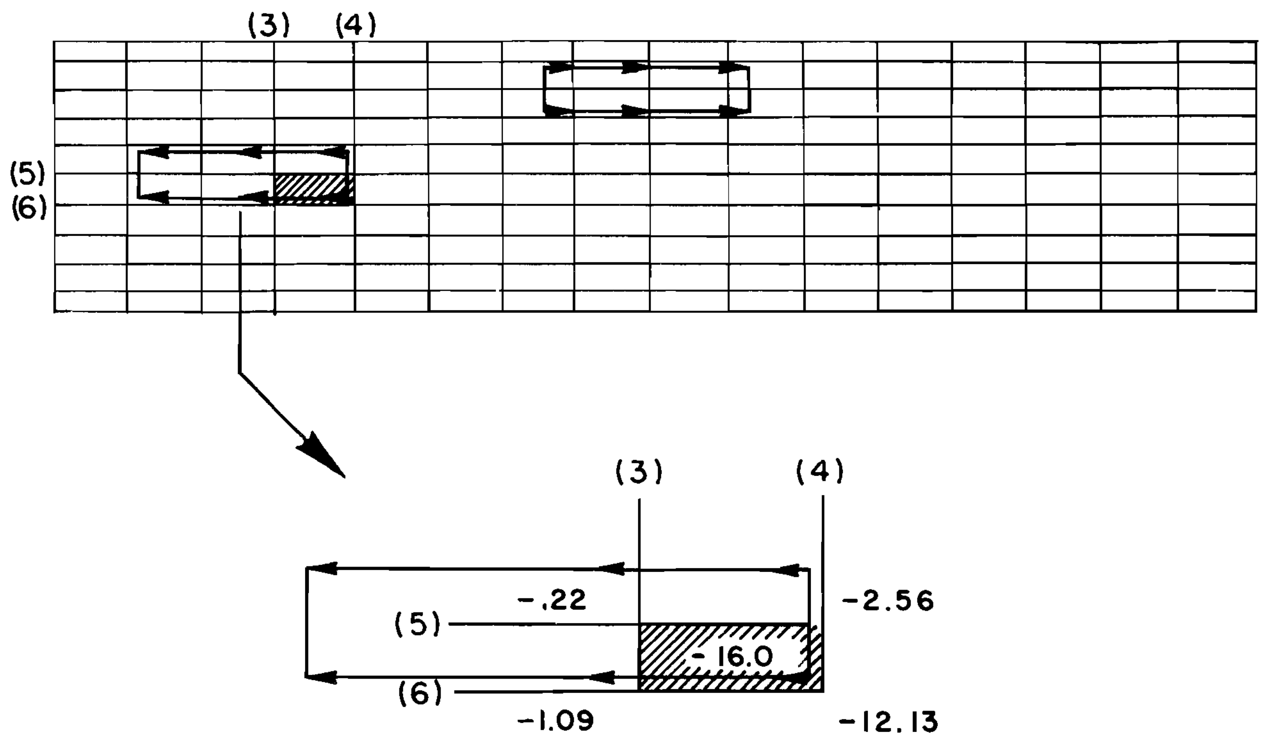


Fig 6. Wheel load allocation to nodes.

By the use of IF statements to evaluate unique alphanumeric characters in the SLAB49 output, the post-processor locates data necessary for conversion of the moments to stresses and all required data that is not transferred directly from the pre-processor. The data from SLAB49 is stored and then the stresses are computed using the methods described in Ref 1. The stresses computed are based on the moments output from the discrete element program and so the computed stresses reflect the characteristic variations of the discrete element model.

After the post-processor has converted all moments to stresses, the resulting stresses are output according to the user specified options. In the pre-processor, the user specifies data to be output along a particular girder and whether a plot is desired. The outputs are printed and plotted as specified on the tape and, as additional output, the post-processor locates maximum stresses along girders and in the slab. One other output of the post-processor is the extrapolated stresses at interior supports.

The post-processor calculates and outputs the stresses at interior supports for the slab and the girders based on the linear extrapolation of stresses on either side of the support. Since interior supports frequently fall between the nodes of the grid system, it is helpful to the user to note these values. These extrapolated values are not exact results and are not considered when providing the location of the maximum stresses as discussed above, but the user must be aware of these stresses that occur at interior supports.

The Data Bases

To execute the pre-processor there is extra data that describes girders and trucks that must be obtained to model stiffnesses and loads. There are two ways open to the user to obtain this extra data:

(1) input by the user as additional data, as described in the Input Guide (Appendix A), or

(2) use of data bases that contain alphanumeric names to identify the data base entries, and the appropriate information.

The second method of obtaining data is best suited for information that is used frequently by the highway engineer, such as standard bridge girders or standard trucks used regularly for loadings. The data bases that are available in this program are:

(1) girder data bases for steel, prestressed, and reinforced concrete (pan-formed) girders, and

(2) a truck data base containing patterns of loads that represent standard truck loading.

CHAPTER 3. AN EXAMPLE PROBLEM

Description of the Problem

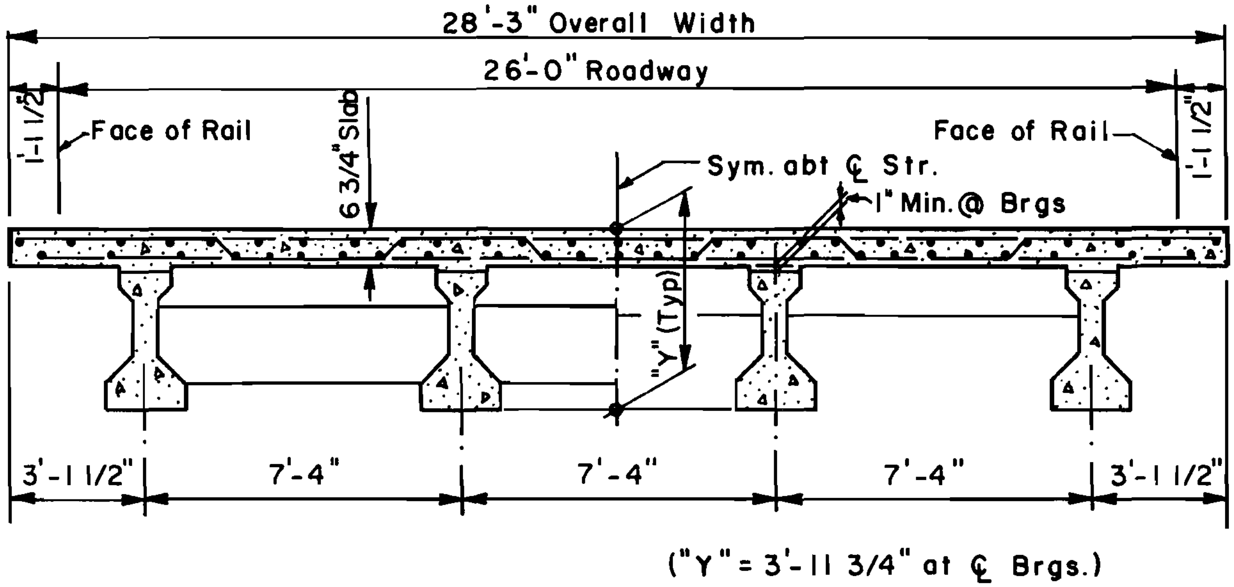
The basic philosophy used in developing these programs and the intended uses for the programs have been discussed previously. A typical problem that may be faced by the bridge engineer is the analysis of an off-system structure for an updated rating. The structure selected for this example problem is a typical H15 design prestressed girder, single-span bridge.

The information required to analyze the bridge includes details about: the structure (width, span, slab thickness, etc.); the materials (concrete weight, concrete strength, etc.); and the girders (how many, type, name, prestressed strand information, etc.). We also need to have information about the diaphragms, the zone of composite action, the loads (for this problem we use standard AASHTO loading patterns), and the desired output information.

The example problem is illustrated in Fig 7. The nominal 55-foot span bridge has "Type C" prestressed girders that are assumed to act compositely with the 6.75-inches-thick bridge deck. Although the roadway is 26 feet wide, the concrete bridge deck is 28 feet 3 inches wide, and the entire bridge width is used for analysis. The concrete diaphragms are spaced at 10 feet 0 inches starting at 6 feet 10-1/2 inches from the beginning of the bridge. The span length is 53 feet 9 inches from center to center of the bearing pads, and it is assumed that the zone of composite action is for the entire length of the bridge. This information about the bridge structure follows the form of Table 2 of the Input Guide in Appendix A.

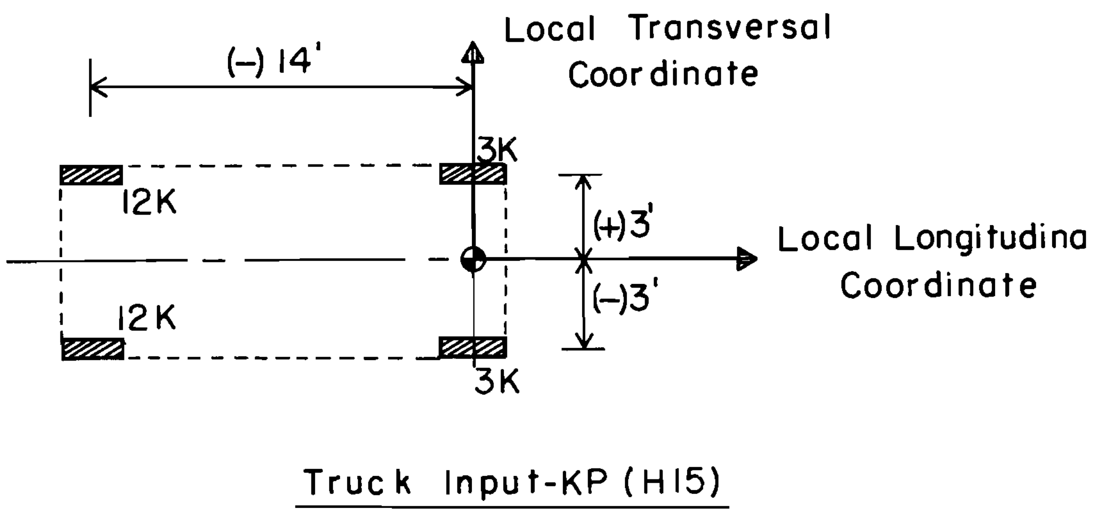
The example structure is subjected to two separate load cases: (1) two standard AASHTO H15 trucks and (2) a single truck loading of a truck located from the data base, HS20V14. Truck load patterns are illustrated in Fig 8. The truck information is input using the format described in Table 3 of the Input Guide (see Appendix A).

The first load case consists of two H15 trucks in adjacent lanes with the longitudinal load position referenced to the centroid of truck as shown in Fig 9. The truck is named "INPUT-KP." For any one structure that is analyzed, one



Typical Transverse Section

Fig 7.



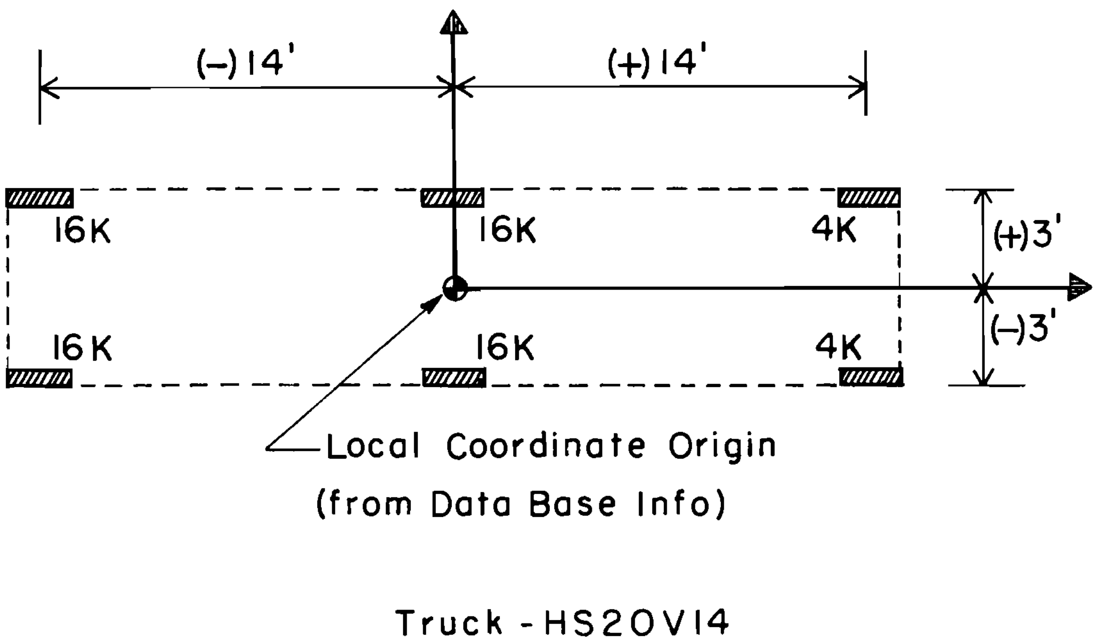
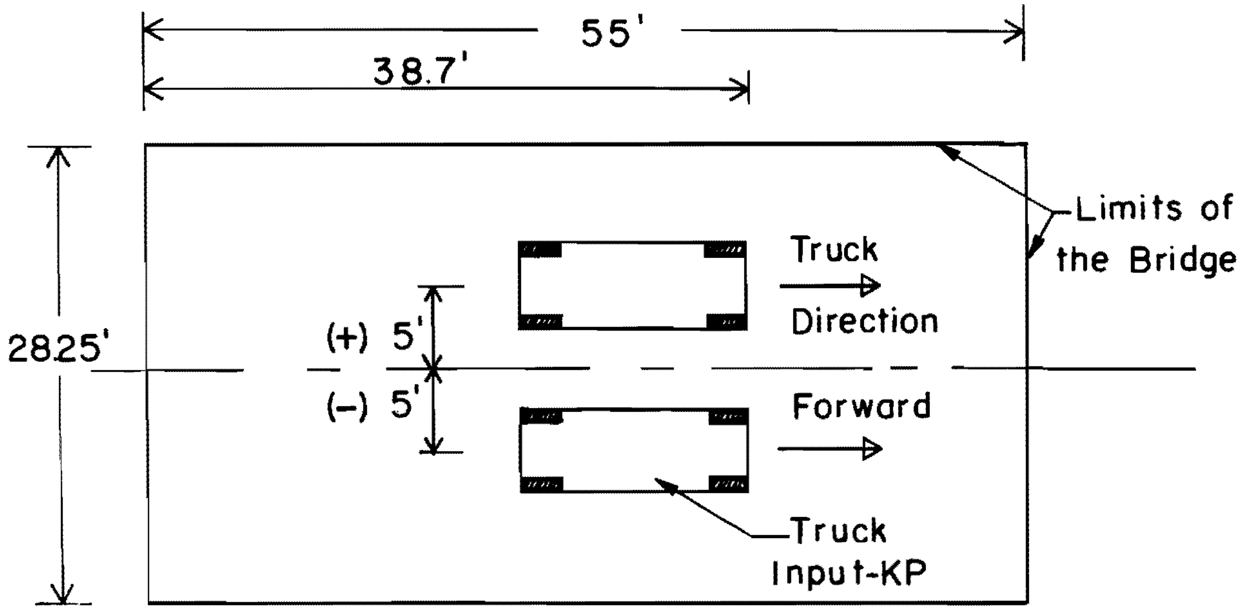
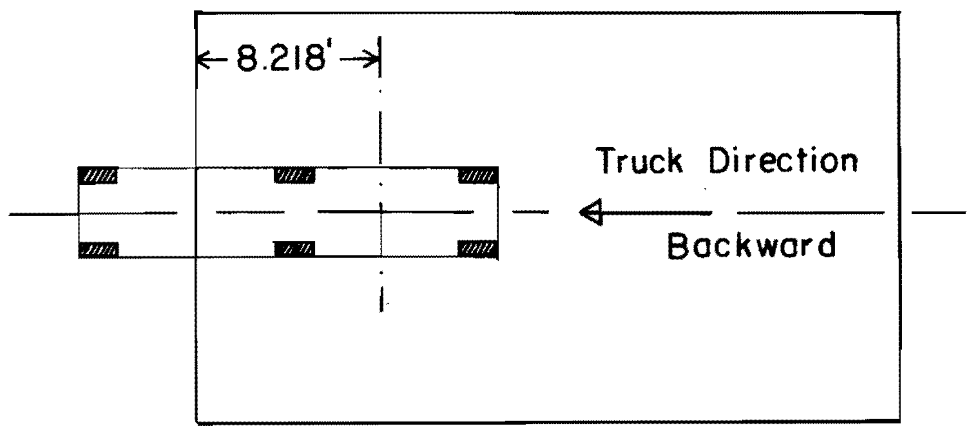


Fig 8. Typical pattern loads.



(a) Load case 1.



(b) Load case 2.

Fig 9. Load cases 1 and 2.

truck load pattern can be named "INPUT-KP." This load pattern will be stored during execution of the program and may be recalled at any time during execution to be loaded on to the structure (Note: The truck is not permanently stored in the data base). This truck can be called up for any subsequent offspring problems.

The second load case, shown in Fig 9, utilizes the truck data base. The load scheme consists of one H20 truck, called "HS20V14" in the data base. The truck is located on the centerline of the bridge with the centroid of the loads placed longitudinally as shown in Fig 9. Two wheels are off the bridge to demonstrate that this feature is possible.

The post-processor outputs from each parent and each offspring problem can be controlled using Table 4 of the Input Guide (see Appendix A). The user limits the data printed out by the post-processor, as generally the user is interested in only a portion of the structure; therefore, the options in Table 4 allow the user to save paper and eliminate the need to read through large quantities of output data.

Input and Output for the Example Problem

The formatted card images for the example problem are contained in Table 1 and a computer listing of this input data for SLBDG4 is contained in Table 2. The output generated by the pre-processor (SLBDG4) is contained in Appendix I while the complex input/output from SLAB49 is included in Appendix J. The output from the post-processor is shown in Appendix K. Finally, we include in Appendix L plots of each load case which show the grid and the truck location on the grid. For each load case we show CAL-COMP plots for girders 2 and 3. There are four plots for each girder: (1) centerline deflection, (2) stresses at top of slab, (3) stresses at top of girder, and (4) stresses at bottom of girder.

The pre- and post-processors must be executed along with SLAB49 in the proper sequence and with the required tape files. These are problems that must be worked out for each operating system.

TABLE 1. FORMATTED CARD IMAGES FOR SLBDG4

IDENTIFICATION EXAMPLE PROBLEM   CODED BY RWS   DATE 10/2/80   PAGE 1 OF 1

```
EXAMPLE PROBLEM ; RWS,CPJ ; 10/2/80
PRESTRESSED GIRDER BRIDGE , TYPE C  GIRDER ; TWO  LOAD CASES
   Ø1     TWO STANDARD AASHTO H15 TRUCKS
    2    Ø    1    1    Ø
     28.25     3.125   12   2Ø      6.34       Ø.Ø
     150.0    3000.0      0.15    3.0E+6
    4     PRES      TYPE C      12           28.91      2Ø.Ø     14.76       Ø.Ø
    5           6.91     46.84    2.ØE1Ø
    1
     53.75
    1
       Ø.Ø     53.75
    2    Ø    1
INPUT-KP        1.3Ø       5.Ø      38.7 FORW
    4
       3.Ø      -3.Ø       0.Ø
       3.Ø       3.Ø       Ø.Ø
      12.Ø      -3.Ø     -14.Ø
      12.Ø       3.Ø     -14.Ø
INPUT-KP        1.3Ø      -5.Ø      38.7 FORW
    2    2    Ø
    2
    3      15.Ø      35.Ø
   Ø2     ONE STANDARD AASHTO H20 FROM THE DATA BASE
    2    Ø    Ø    1    Ø
    1    1    1
HS20V14         1.30       Ø.Ø     8.218 BACK
    2    2    Ø
    2
    3
CEASE
```

TABLE 2. COMPUTER LISTING OF DATA FOR SLBDG4

```
EXAMPLE PROBLEM ; RWS,CPJ ; 10/02/89
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ;  TWO LOAD CASES
   01      TWO STANDARD AASHTO H15 TRUCKS
    2    0    1    1    0
     28.25     3.125   12   .20      6.34       0.0
     150.0    3000.0      0.15     3.0E+6
    4      PRES       TYPE C         12          28.91      20.0     10.76     0.0
    5            6.91     46.84     2.0E10
    1
     53.75
    1
       0.0     53.75
    2    0    1
INPUT=KP        1.30      5.0      38.7 FORW
    4
       3.0      -3.0      0.0
       3.0       3.0      0.0
      12.0      -3.0    -14.0
      12.0       3.0    -14.0
INPUT=KP        1.30     -5.0      38.7 FORW
    2    2    0
    2
    3      15.0     35.0
   02      ONE STANDARD AASHTO H20 FROM THE DATA BASE
    2    0    0    1    0
    1    1    1
HS20V14         1.30      0.0      8.218 BACK
    2    2    0
    2
    3
 CEASE
```

CHAPTER 4. SUMMARY

This work was intended to serve the purpose of extending the usefulness of an existing discrete-element computer program for the analysis of orthogonal and uniformly skewed bridge floor systems. The discrete-element program SLAB49 used in this work has been described and documented in Ref 1 and thus has not been dealt with to a substantial extent in this report. The major emphasis of this work was directed towards a primary need of user-oriented data-generation routines for selected classes of bridge structures having regular geometry with prismatic main members which are uniformly spaced for which SLAB49 could be used as the basic solver.

By deferring complex cases, it was possible to identify four classes of bridges for which convenient input and output arrangements could be developed. This was accomplished by developing data generators (i.e., the pre-processor) for each class for the purpose of taking simplified inputs with regard to main member type, bridge geometry, and loading, and then provide automatically the more detailed information required by the discrete-element computer program. Post-processors for each class were then developed for searching the output from SLAB49 for the purpose of devising more convenient output arrangements than were presently available.

The four widely used bridge classes for which pre- and post-processors were designed are: (1) the steel beam and slab section, (2) prestressed girder and slab section, (3) simple span or continuous slab with or without integral curbs, and (4) simple span pan-formed sections. Bridge class information and truck configuration may be stored on data bases for those cases requiring frequent access. The parent program SLAB49 together with the three computer programs developed in this work, i.e., (1) the pre-processor (SLBDG4), (2) the data base generator (BASGEN), and (3) the post-processor (SEARCH), have been termed the Texas Bridge Analysis and Rating Program (TBAR). TBAR has been adapted to the Texas SDHPT computer facilities for structural evaluation of existing bridges for overload conditions on an ongoing basis.

REFERENCES

1. Panak, John J., and Hudson Matlock, "A Discrete-Element Method of Analysis for Orthogonal Slab and Grid Bridge Floor Systems," Research Report No. 56-25, Center for Highway Research, The University of Texas at Austin, May 1972.

2. Panak, John J., "Skewed Multi-Beam Bridges with Precast Box Girders," Research Report No. 206-1F, Texas State Department of Highways and Public Transportation, September 1977.

APPENDIX A

INPUT GUIDE FOR SLAB49

GUIDE FOR INPUT DATA FOR SLBDG4

with supplementary notes

by

Robert E. Cornwell and Richard W. Stolleis

February 1979

GUIDE FOR INPUT DATA - CARD FORMS

IDENTIFICATION OF RUN (two cards per run)

Enter descriptive alphanumeric information - date , user's name , and structure identification .

1 … 80

Include descriptions of girders and trucks that are input using the special input options .

1 … 80

IDENTIFICATION OF PROBLEM (one card per problem)

PROBLEM NUM		Alphanumeric description of the problem with load descriptions .
1–5		11–80

TABLE 1. CONTROL INFORMATION (one card for each problem , parent and offspring)

NUM OF PROB'S	ECHO PRINT	3-D PLOT	DEAD WT	TABLE 4 KEEP
	ENTER "1" TO EXECUTE			
*	**			
1–5	10	15	20	25

*Enter the total number of problems in this run .
**Card image print out provided at end of output listing .

NOTE: A run consists of one parent problem and any offspring problems for the same structure. The parent problem is the first problem with all the structural details (input first) and the offspring problems are the same structure with repositioned loads (input after the complete parent problem) .

TABLE 2A.-2F. BRIDGE GEOMETRY (omit Table 2A.-2F. for all offspring problems)

TABLE 2A. BRIDGE DESCRIPTION (one card per run)

TOTAL BRIDGE WIDTH (FT)	SLAB OVERHANG (FT)	TRANS DIV	LONG DIV	SLAB THICKNESS (IN)	*	SKEW ANGLE (DEGREES)	INTEGRAL CURB HEIGHT (FT)	INTEGRAL CURB WIDTH (FT)
1 – 10	20	25	30	40		43 – 50	60	70

* Input "LF" or "RF" .

TABLE 2B. MATERIAL PROPERTIES (one card per run)

CONCRETE WEIGHT (PCF)	CONCRETE STRENGTH (PSI)	POISSON'S RATIO	MODULUS OF CONCRETE (PSI) *	MODULUS OF STEEL (PSI) *
1 – 10	20	30	40	50

* Enter values for modulus of elasticity using a right justified E-field . If the values are not input , the program will use standard values that are set internally .

TABLE 2C. GIRDER INFORMATION

1) GIRDER DESCRIPTION (one card per run , complete only applicable information)

NUM OF GIRDERS	GIRDER TYPE *	GIRDER NAME **	NUM OF STRANDS	INITIAL FORCE PER STRAND (KIPS)	% LOSS (PERCENT)	CENTERLINE ECCENTRICITY (IN)	HAUNCH (IN)
1 – 5	11 – 14	21 – 28	31 – 35	41 – 50	60	70	80

* Legal types are "PRES" , "STEE" , "REIN" , and "SLAB" (for slab bridge enter "SLAB" , omit all other data) .

** Enter girder name as appears in girder data base , e.g. W36x160 , or enter "INPUT" left justified and complete one card of the next section .

TABLE 2C., continued .

2) INPUT GIRDER INFORMATION (include one of the following two cards if girder name is "INPUT")

a) INPUT STEEL OR PRESTRESSED GIRDER INFORMATION (include this card for input ("STEE" or "PRES" girder types)

CROSS SECT AREA (IN-2)	MOMENT OF INERTIA (IN-4)	GIRDER DEPTH (IN)	TOP FLANGE (IN)	CENTROID TO BOTTOM (IN)
1–10	20	30	40	50

b) INPUT REINFORCED CONCRETE GIRDER INFORMATION (include for input "REIN" girder type)

INTERIOR SECTION PROPERTIES			EXTERIOR SECTION PROPERTIES				
MOMENT OF INERTIA (IN-4)	CROSS SECT AREA (IN-2)	CENTROID TO BOTTOM (IN)	MOMENT OF INERTIA (IN-4)	CROSS SECT AREA (IN-2)	CENTROID TO BOTTOM (IN)	EQUIVALENT SLAB THICKNESS (IN)	TOTAL SECTION DEPTH (IN)
1–10	20	30	40	50	60	70	80

TABLE 2D. DIAPHRAGM INFORMATION (one card per run)

NUM OF DIAPHRAGMS	STARTING LOCATION (FT)	ENDING LOCATION (FT)	DIAPHRAGM STIFFNESS (LB-IN)
1–5	11–20	30	40 **

** Enter diaphragm stiffness in a right justified E-field .

TABLE 2E. SPAN INFORMATION (two or more cards for more than eight spans , omit for offspring problems)

NUM OF
SPANS

(one card per run followed by enough cards to enter all span lengths @ eight spans per card)

1 5

SPAN LENGTH
(FT)

(enter eight span lengths per card , use additional cards if required)

1 10 20 30 40 50 60 70 80

TABLE 2F. ZONES OF COMPOSITE ACTION (one or more cards , omit for offspring problems)

NUM OF
ZONES

(one card per run followed by enough cards for all zones @ four per card)

1 5

DISTANCE TO START AND END OF ZONES OF COMPOSITE ACTION (omit this card if there are zero zones)

START LOCATION (FT) | END LOCATION (FT)

(enter four pair of locations per card , use additional cards if required)

1 10 20 30 40 50 60 70 80

TABLE 3. TRUCK INFORMATION

A. GENERAL TRUCK INFORMATION

1) TRUCK CONTROL INFORMATION (one card per problem)

NUM OF TRUCKS	PRINT LOADS	TRUCK PLOT
	ENTER "1" TO EXECUTE	
1–5	10	15

2) INDIVIDUAL TRUCK DESCRIPTION AND PLACEMENT (one card for each truck to be loaded in this problem)

TRUCK NAME	LOAD IMPACT FACTOR	TRANSVERSE LOCATION (FT)	LONGITUDINAL LOCATION (FT)	DIRECTION
*				
1–8	11–20	30	40	42–45

DIRECTION: FORW-Forward
BACK-Backward

* Enter a truck name from the data base or "INPUT ___" . If the same truck is to be repositioned in a subsequent offspring problem , enter "INPUT-KP" and in any offspring problems enter the same name (INPUT-KP) without the section for input truck information. This option may be used for one input truck per run , the wheel locations are stored within the program and do not change the existing data base .

B. INPUT TRUCK INFORMATION (include this section for each truck name "INPUT ___" , except for truck names "INPUT-KP" that have been saved in a previous problem)

NUM OF WHEELS
1–5

(one card for each truck name "INPUT___" followed by one card for each wheel on the truck)

WHEEL LOAD (KIPS)	LOCAL TRANS LOCATION (FT)	LOCAL LONG LOCATION (FT)
1–10	20	30

(one card for each wheel in the load pattern)

TABLE 4. OUTPUT AREAS

NUM OF OUTPUTS | OUTPUT TYPE | MOMENT PRINT OUT

(one card for each problem followed by one card for each girder output)

1–5	6–10	11–15
	*	**

* Enter "1" for data print out , enter "2" for data and plot of stresses .
** Enter "1" for print out of moments from SLAB49 .

If "NUM OF OUTPUTS" is zero then all the resultant stresses are printed out .

GIRDER NUMBER | DATA START (FT) | DATA STOP (FT)

(one card for each girder output , omit this card if "NUM OF OUTPUTS" is zero)

1–5	11–20	21–30

* If data start and stop is not specified , then all data along that girder is printed out .

JOB TERMINATION (one card per run)

Additional alphanumeric data

1–5	11–80
*	

* Enter "CEASE" for last card of the run , or delete this card for offspring problem to follow .

FOR OFFSPRING PROBLEMS COMPLETE :

1) PROBLEM IDENTIFICATION CARD
2) TABLES 1, 3, and 4; omit TABLE 2 in all offspring problems

ADDITIONAL OUTPUT FROM SLAB49 (include this section if output from SLAB49 is to be printed)

NOTE : This information is to be included after the postprocessor data (TABLE 4) and only when a "1" is placed in column 30 of the TABLE 1. CONTROL INFORMATION card .
This section will normally be used for special studies and should not be included if the postprocessor is used for results interpretation .

PROFILE OUTPUT AREAS (10 maximum , these outputs are produced in the SLAB49 output)

NUM OF AREAS

(one card per problem)

NUM OF AREAS
1–5

PROFILE OUTPUT DESCRIPTION (one card for each output area , omit this card for "NUM OF AREAS" equal to zero)

TRANSVERSE START (FT)	LONGITUDE START (FT)	TRANSVERSE STOP (FT)	LONGITUDE STOP (FT)	DEFL	LONG MOM'T	TRANS MOM'T	PRINCIPLE STRESS
1–10	20	30	40	45	50	55	60
				*	*	*	**

* Enter "1" for slab , or "2" for beam .
Enter "1" or "2" DO NOT mix output types for the run .

** Enter "1" or blank .

GENERAL OUTPUT AREAS (10 maximum , these areas limit the results output from SLAB49)

NUM OF AREAS

(one card per problem , each area is defined by a start and end location below)

NUM OF AREAS
1–5

(enter four pairs of start and end per card , use additional cards if required)

START LOCATION (FT)	END LOCATION (FT)						
1–10	20	30	40	50	60	70	80

DESCRIPTION OF INPUT DATA

TABLE 1. CONTROL INFORMATION

This table consists of one card with various parameters to control the programs. The first parameter is the total number of problems in this run, for one parent problem and two offspring problems the total number of problems is three. The echo print option allows the user to obtain a print out of the card images generated by SLAB 49 by placing a one in column 10. A pseudo three dimensional plot of the deflected shape of the structure can be obtained from SLAB49 by placing a one in column 15. By placing a one in column 20, the dead weight of the structure is applied to the structure grid model. For prestressed girder structures the dead weight does not include the weight of the girder, because that is included in the prestressing stress calculations. The option, Table 4 Keep, allows the user to keep the previous values of Table 4 by a one entered in column 25 and deleting all of the cards of Table 4. For parent problems enter a one in column 25 and delete Table 4 and the postprocessor will print stress at all grid locations.

TABLE 2. BRIDGE GEOMETRY

The geometry of the structure determines the stiffness at each grid point, the support locations, and other important values that model the bridge structure. This table consists of eight to ten cards vital to the model of the structure.

TABLE 2A. BRIDGE DESCRIPTION

This table includes information for the basic layout of the structure and grid model. The total bridge width is the overall distance across the bridge measured perpendicular to the bridge centerline. The overhang is the distance from the centerline of the exterior girder to the edge of the structure measured perpendicular to the edge of the structure. The user selects the number of divisions across the bridge (trans div) and the number of divisions along the bridge (long div) and the program uses these numbers to compute the grid dimensions. Due to the geometric constraints of the structure the number of divisions across the bridge is recomputed internally to make the centerline of each girder and a grid line coincide. The maximum grid size is 24 x 48 for the current version of processors and these number of divisions may be oriented either in the longitudinal or transverse direction. If the grid exceeds 24 divisions in one direction the maximum number of divisions in the other direction is 24. The skew angle allows the user to model a structure that are uniformly skewed with the same angle of skew at all support lines. For a skewed bridge the grid is not skewed, but the supports and limits of stiffnesses are changed to model the structure correctly. To denote a direction of skew use "LF" or "RF" in column 41 and 42 of this card to denote left forward or right forward skew, as shown below:

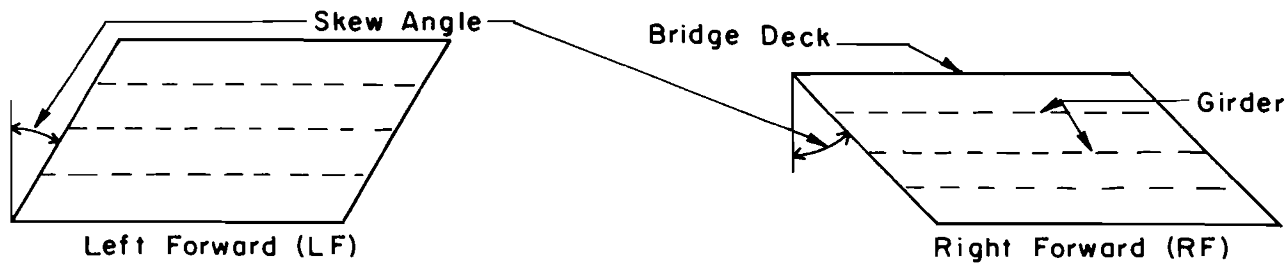


TABLE 2B. MATERIAL PROPERTY

This table describes material properties of the structure. The concrete weight and the concrete strength are values based on the design of the structure. Poisson's ratio is used for the slab computations, it is always positive, usually 0.15 to 0.20. The modulus of the elasticity for concrete or steel is set internally if the user leaves either or these entries blank. The modulus of steel is set to 29E+6 psi and the modulus of concrete is computed using the formula E = 33.0 * weight**1.5 * strength**0.5.

TABLE 2C. GIRDER INFORMATION

This table consists of one or two cards that describe the girder properties that are necessary for analysis of the structure. The first card of this table is used in all parent problems. This card contains the number of girders in the structure and alphanumeric designations for girder type and name. The girder type will always be one of four types: "PRES", "STEE", "REIN" or "SLAB". The girder name must be a legal name that appears in the girder data base or, "INPUT___" left justified in the 8 columns provided. For girder name equal to "INPUT___" the user is required to input all girder information on one card that is described in Table 2C.(2). For the case "SLAB", girder information is not used by the processors, because "SLAB" denotes a slab bridge with no girders. The girder type "PRES" requires additional girder information entered in columns 31 to 80 of the girder description card. This information is used to compute the stresses that occur in the girder due to girder deadload and prestressing forces. The required inputs include: number of strands, initial force per strand (see table below), % loss of the strand force due to shrinkage, creep, etc. (20% to 30%), centerline eccentricity measured from centroid of girder to centroid of strands (down is positive), and the height of the haunch.

PRESTRESSING STRAND DATA

STRAND NAME	INITIAL FORCE PER STRAND (KIPS)
1/2 - 270	28.91
7/16 - 270	21.70
3/8 - 270	16.10
1/2 - 250	25.20
7/16 - 250	28.90
3/8 - 250	14.00

For girder name equal to "INPUT" the data base is not searched for the necessary girder information, therefore additional girder input is required. The additional input data required is placed on one card inserted after the girder description card. The additional data varies for the different girder types. The data required for the R.C. girders is based on the total section properties of the girder and slab. The equivalent slab thickness is the slab thickness used to compute the slab stiffness. For variable thickness slabs (pan formed), this value should be determined by some reasonable modeling technique. The exterior section properties include the edge girders and the slab that attributes composite stiffness to that girder and those sections are assumed symmetrical about the girder centerline.

TABLE 2D. DIAPHRAGM INFORMATION

The Diaphragm Information defines the location of all interior diaphragms. The number of diaphrams should not include the two end diaphragms and the starting and ending locations are the distances to the first and last interior diaphragms respectively, as measured from the front edge of the bridge (see sketch below). The diaphragms of the skewed case are assumed to be staggered due to the bridge skew and all diaphragm placements are rounded to the nearest grid location. The diaphragm stiffness is entered in an E-field with units of pounds and inches, based on the modules of elasticity and moment of inertia about the longitudinal axis (EI).

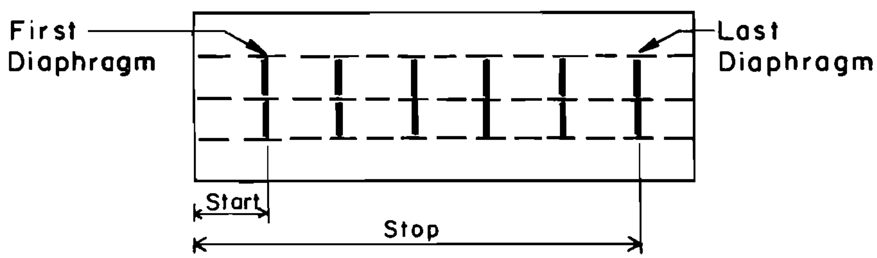


TABLE 2E. SPAN INFORMATION

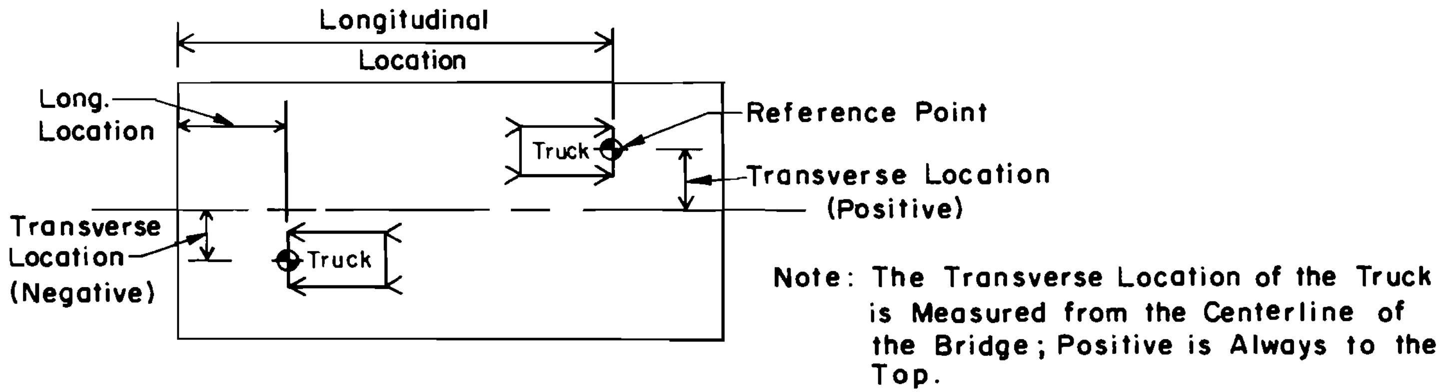
The span information consists of one card with the number of spans entered and additional cards with the span lengths entered in order from the beginning to the end of the continous unit to be analyzed.

TABLE 2F. ZONES OF COMPOSITE ACTION

Zones of composite action are regions of the structure that the slab and girder are considered to act together. These regions are usually areas in which the slab is in compression. In areas where the slab is in tension, only girder stiffness attributes to the beam bending stiffness. Enter the number of zones on one card and for number of zones greater than zero include additional cards with the zone limits. If the number of composite zones is zero there is no contribution of the slab to the beam stiffnesses. Enter the distance along the bridge to the start and end of each zone of composite action as measured along the length of the structure.

TABLE 3. TRUCK INFORMATION

This table includes all of the data that will be used to place concentrated loads on the structure. The number of trucks to be included in this problem is input on the first card of this table along with two options to control the output of the preprocessor. Enter a one in column 10 to obtain the calculated grid loads that are produced from each wheel and enter a one in column 15 to obtain a plot of the grid with the trucks and wheels drawn as positioned for the problem. On the second card is entered the name of the truck, the impact factor (1.1 to 1.4) and the location and direction of the truck. The truck name is the alphanumeric name as appears in the data base, or "INPUT---" or "INPUT-KP". The transverse and forward locations determine the location of the reference point for the truck and the direction "FORW" or "BACK" gives the orientation of the truck to the structure (see sketch below).



For truck name equal to "INPUT---" additional information in part b. follows directly after the truck description card. The additional information includes one card with the number of wheels to be entered and one card for each wheel. The wheel cards give the location of the wheels relative to the reference point, see sketch above, and the wheel load in kips. A special case of input truck is the truck name "INPUT-KP", for this truck name the program saves the local wheel loads and locations and in any problems after the first entry of "INPUT-KP" the user must complete only the truck description card. This allows the user to input a truck and use it in several load patterns without reentering the extra input truck information.

TABLE 4. OUTPUT AREAS FROM THE POSTPROCESSOR

This table allows the user to obtain data and plots along the centerline of any girder without the need to sort through all the results for every grid location. If a one is entered in column 25 of Table 1, this table is deleted and all results are printed as resultant stresses. The number of outputs is the total number of girder outputs that are desired and the type refers to data or data and plot outputs that can be obtained from the postprocessor. Enter "1" for data only and enter "2" to obtain data output and plots along the girder centerline. A print out of the resultant moments from SLAB49 can be obtained by placing a one in column 15, this allows the user to check conversion of moments to stresses or to convert moments in the diaphragms. Following this first card input one card for each girder output. The number of the girder to be output is the number assigned by counting the girders from the origin across the bridge. If limited data is desired the limits are entered in columns 6 through 25, but if the limits are left blank the program will print out all data for that girder.

ADDITIONAL OUTPUT FROM SLAB49

This section is provided to allow user's that are familiar with SLAB49 to obtain outputs by the use of Tables 8 and 9 of that program. The distances that are input in this table are rounded to the nearest grid location and the outputs are produced by SLAB49. To add this optional portion to the input card sequence enter a one in column 30 of Table 1, and insert all of the cards from this addition input after Table 4.

APPENDIX B

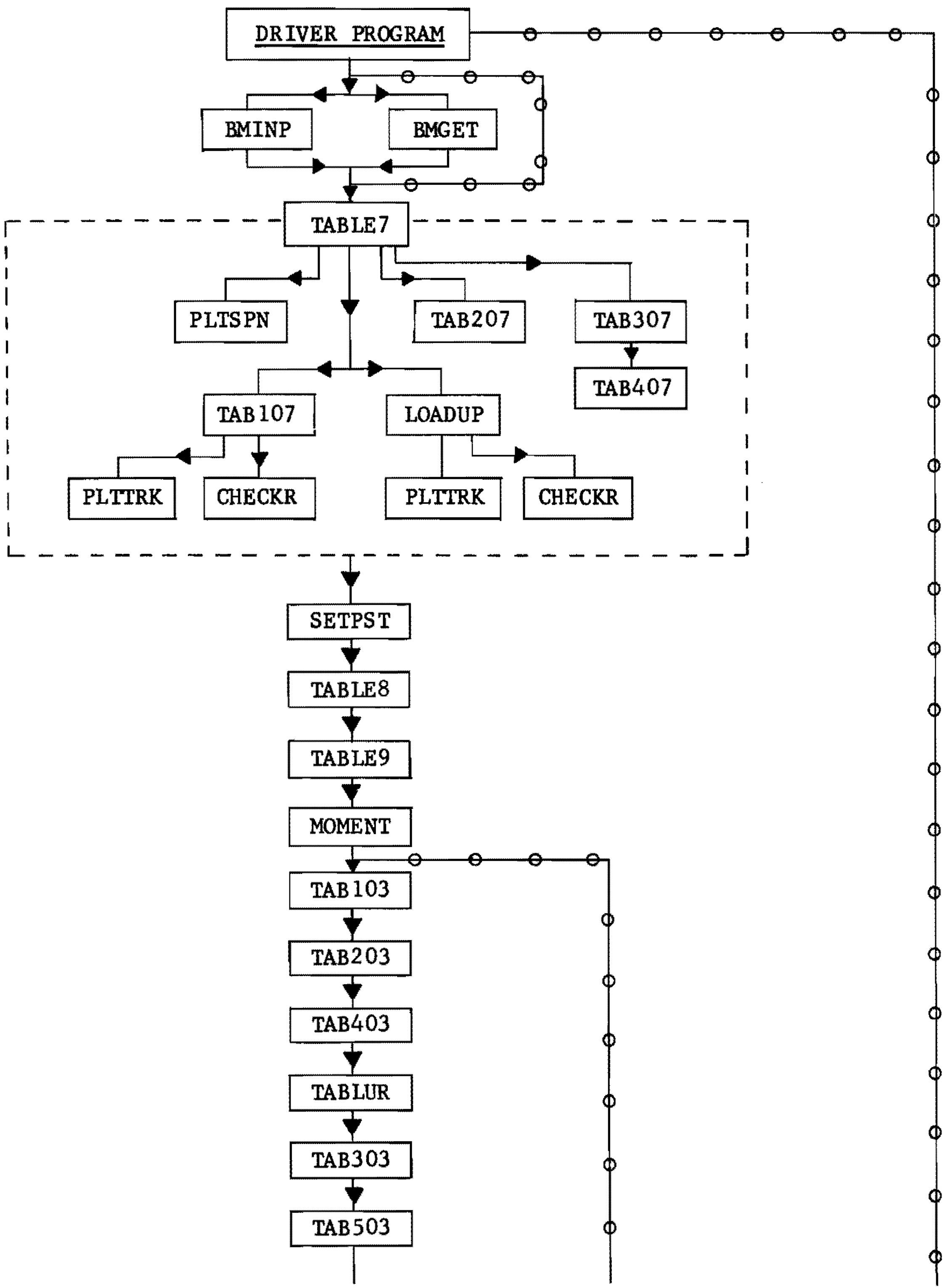
DOCUMENTATION OF SLBDG4

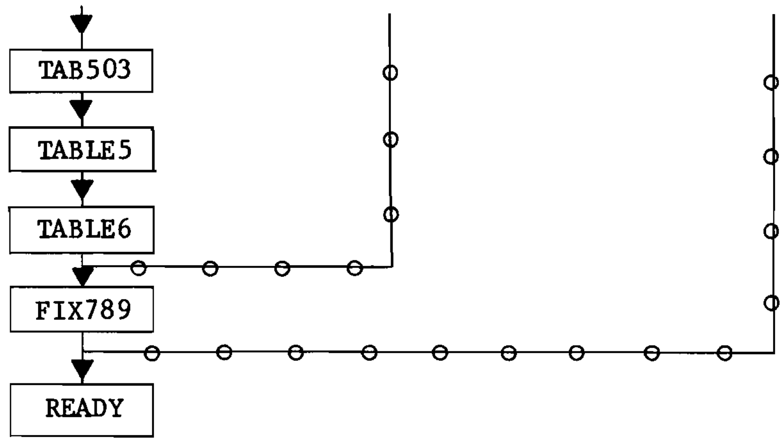
APPENDIX B. DOCUMENTATION OF SLBDG4

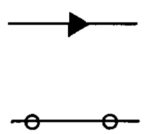
GENERAL

The only function of the SLAB49 pre-processor (SLBDG4) is to enable the user of the TBAR package to easily and efficiently use the SLAB49 analysis program. See Fig 10 for the general flow of the pre-processor. In order to make the function of the TBAR as transparent as possible to the user, several unique operations are carried out. The first of these involves the way in which the grid coordinate system is referenced. SLAB49 requires that input be in the form of an X-Y coordinate system. The number of grid points in the X direction is always greater than or equal to ($\geq$) the number of grid points in the Y direction. When the data is input the user will specify all data with longitudinal (X) and transverse (Y) coordinates. If the number of grid points in the transverse direction is ($\geq$) the number of grid points in the longitudinal direction, the pre-processor will automatically transpose the results internally (i.e., Y = longitudinal direction, X = transverse direction). When the results of SLAB49 are subsequently processed by the post-processor, the coordinate system will be transposed once again so that the user never has to worry about the X-Y reference system unless he views unprocessed SLAB49 results.

The second unique operation carried out by the pre-processor is the way by which a skewed bridge geometry is handled. A rectangular grid is generated that will envelop the entire skew bridge. This will result in two triangles of grid points that technically do not exist. These two sets of points are given zero stiffness and not allowed to have any load applied to them. The grid points that fall directly on either side of the actual end of the skewed bridge are modified to represent a pinned end condition at the actual support point (usually falling between two grid points). See Ref 2 for a description of this method. When results are printed the two triangular areas at either end of the span will show null results (i.e., stress = moment = displacement = 0.0).


DRIVER PROGRAM
BMINP
BMGET
TABLE7
PLTSPN
TAB207
TAB307
TAB407
TAB107
LOADUP
PLTTRK
CHECKR
PLTTRK
CHECKR
SETPST
TABLE8
TABLE9
MOMENT
TAB103
TAB203
TAB403
TABLUR
TAB303
TAB503


TAB503
TABLE5
TABLE6
FIX789
READY


normal program flow
offspring problem flow

The third and final major operation of the pre-processor is the automatic generation of offspring problems. During an offspring problem it is assumed that only the loading and output controls might change. For this reason only TABLE7, TABLE8, TABLE9, and SETPST are accessed during offspring generation. All other routines and tables are assumed constant and are therefore bypassed.

MAIN PROGRAM

The purpose of the main program is to control and coordinate the operation of the pre-processor subroutines. The first action undertaken by this routine is the geometry of the bridge and the physical layout of the entire grid system. This includes span lengths, girder locations, composite section locations, grid spacing, etc. The next major operation performed is to calculate the truck loading and output areas. Even though this is required last in the SLAB49 input, this operation must be carried out first so that the number of cards contained in TABLE7, TABLE8, and TABLE9 will be known. This information is stored temporarily (this is the first information required in the SLAB49 input). All bridge and span information is generated next. Due to the nature of the input, the number of cards contained in Tables 3 through 6 can be calculated at the beginning of the execution. The information concerning Tables 7, 8, and 9 is now retrieved and printed. At this point a check is done to see if an offspring problem is requested; if so, Tables 7, 8, and 9 will be generated for the new load case.

SUBIBM

SUBJECT: GIRDER SUBROUTINES

RELATED SUBROUTINES:
1) BMGET
2) BMINP
3) TAB203

1) SUBROUTINE BMGET (INAM, JNAM, BEAMI, TK, GIRDSP, AREA, OVERH1, POIS, BEAMD, SC, IGRIP, SLABI, ARINT, AREND, NX)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

INAM JNAM	- two (2) four-character alphanumeric variables which define the girder to be used in the analysis
BEAMI	- girder moment of inertia
TK	- slab thickness
GIRDSP	- transverse spacing of girders
AREA	- x-sectional area of girder
OVERH1	- actual (non-rounded) slab overhang in transverse direction
POIS	- Poisson's Ratio for concrete
BEAMD	- girder depth
SC	- ratio of (steel modulus/concrete modulus)
IGRIP	- four-character alphanumeric variable to define the type of girder used in the analysis
SLABI	- moment of inertia of slab x-section of unit width
ARINT	- x-sectional area of interior reinforced concrete girder
AREND	- x-sectional area of exterior reinforced concrete girder
NX	- control variable to determine if integral curb is present

PURPOSE: Subroutine BMGET controls and coordinates all SLBDG4 access to the three girder data bases created with the use of BASGEN. After the correct data base is determined, control is switched to the specific section of the routine for that data base. All information obtained from the data bases is in units of inches. A conversion is immediately undertaken to obtain units of feet, which is consistent with the remainder of the program. After the information concerning the girder to be used in the analysis is obtained, the subroutine computes gross cross-section properties, i.e., composite and noncomposite section properties.

2) SUBROUTINE BMINP (BEAMI, TK, GIRDSP, AREA, OVERH1, POIS, BEAMD, SC, IGRIP, SLABI, ARINT, AREND, NX)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

- BEAMI - girder moment of inertia
- TK - slab thickness
- GIRDSP - transverse spacing of girders
- AREA - x-sectional area of girders
- OVERH1 - actual (non-rounded) slab overhang in transverse direction
- POIS - Poisson's Ratio for concrete
- BEAMD - girder depth
- SC - ratio of (steel modulus/concrete modulus)
- IGRIP - four-character alphanumeric variable to determine the type of girder used in the analysis
- SLABI - moment of inertia of slab x-section of unit width
- ARINT - x-sectional area of interior reinforced concrete girder
- AREND - x-sectional area of exterior reinforced concrete girder
- NX - control variable to determine if integral curb is present

PURPOSE: BMINP performs the same function as subroutine BMGET. This subroutine allows the user to input a girder in the input stream that is not presently in any of the girder data bases. Such capabilities may be useful in one-of-a-kind or experimental case studies.

3) SUBROUTINE TAB203 (OVERH1, JDIV, GIRDSP, ES, EL, AREA, NGIRDR, BEAMI, HY, CONCW, TK, IGRTP, ARINT, AREND, HX, BRIDL)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

OVERH1 - actual (non-rounded) slab overhang in transverse direction

JDIV - number of grid divisions in transverse direction

GIRDSP - transverse spacing of girders

ES - steel modulus

EC - concrete modulus

AREA - x-sectional area of girder

NGIRDR - number of girders in transverse direction

BEAMI - girder moment of inertia

HY - spacing of grid points in transverse direction

CONCW - density of concrete, pcf

TK - slab thickness

IGRTP - four-character alphanumeric variable to define the type of girder used in the analysis

ARINT - x-sectional area of interior reinforced concrete girder

AREND - x-sectional area of exterior reinforced concrete girder

HX - spacing of grid points in longitudinal direction

BRIDL - total length of bridge (along a single side)

PURPOSE: TAB203 is used to convert information obtained in subroutines BMGET and BMINP into a form compatible with SLAB49. This is done by calculating girder stiffness and grid deadweight values and applying them to the appropriate grid points. Along any longitudinal girder location two passes are made. This allows the end grid point to obtain a stiffness value equal to one-half that of an interior grid point. This is necessary to model the reduced stiffness found at the ends of the girders in a finite difference model. Output from this routine is used in constructing TABLE3 of the SLAB49 input.

SUBIBM

SUBJECT: SLAB SUBROUTINES

RELATED SUBROUTINES:
1) TABCUR
2) TAB103
3) TAB303
4) TAB403
5) TAB503
6) TABLE5
7) TABLE6
8) PLTSPN

1) SUBROUTINE TABCUR (IDIV, JDIV, CURBA, CURBI, NX, EC, CONCW, DECURB, IGRTP, HX, HY, BRIDL, NT3)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

IDIV	-	no. of grid divisions in longitudinal direction
JDIV	-	no. of grid divisions in transverse direction
CURBA	-	x-sectional area of integral curb
CURBI	-	moment of inertia of integral curb
NX	-	control variable to determine if integral curb is present
EC	-	concrete modulus
DECURB	-	dead weight to be applied to grid points due to integral curb
IGRTP	-	four-character alphanumeric variable to define the type of girder used in the analysis
HX	-	spacing of grid points in longitudinal direction
HY	-	spacing of grid points in transverse direction
BRIDL	-	total length of bridge (along a single side)
NT3	-	FORTRAN Unit 3

PURPOSE: TABCUR calculates the bending stiffness and dead weight to be added longitudinally about the extreme transverse sides of the slab. Two passes are made when adding these additional values to the grid system in order to correctly define the reduced stiffness occurring at the corner grid points. Output from this routine is used in constructing TABLE3 of the SLAB49 input.

2) SUBROUTINE TAB103 (IDIV, JDIV, EC, POIS, TK, IOPT, IGRTP, SLABI, HX, BRIDL, HY, DX)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

IDIV - no. of grid divisions in longitudinal direction

JDIV - no. of grid divisions in transverse direction

EC - concrete modulus

POIS - Poisson's Ratio for concrete

TK - slab thickness

IOPT - control variable to allow user to specify zero slab stiffness

IGRTP - four-character alphanumeric variable to define the type of girder used in the analysis

SLABI - moment of inertia of slab x-section of unit width

HX - spacing of grid points in longitudinal direction

BRIDL - total length of bridge (along a single side)

HY - spacing of grid points in transverse direction

DX - bending stiffness of slab in longitudinal direction

PURPOSE: TAB103 calculates the bending stiffness of the slab in the longitudinal and transverse directions. The slab stiffness is then added to the grid point stiffnesses one longitudinal row at a time (two passes per row). This is necessary to account for any skew the bridge may obtain. The output generated by this routine is used in constructing TABLE3 of the SLAB49 input.

3) SUBROUTINE TAB303 (NSPAN, JDIV, HY, ID, SPNGK, IGRTP, HX)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

NSPAN - no. of spans that comprise the bridge under analysis

JDIV - no. of grid divisions in transverse direction

HY - spacing of grid points in transverse direction

ID - no. of supported points (in transverse direction) along any one span

SPNGK - spring constant used to represent a supported point

IGRTP - four-character alphanumeric variable used to define the type of girder used in the analysis

HX - spacing of grid points in longitudinal direction

PURPOSE: TAB303 is designed to locate the support points for all spans contained in the bridge. Once located, this routine will calculate a spring stiffness and axial load necessary to model a support point that falls between two grid points. The spring model is defined in Ref 2. The output of this routine is used to construct TABLE3 of the SLAB49 input. The axial load associated with each support point is written to scratch storage for later use by subroutine TABLE 6.

4) SUBROUTINE TAB403 (BEAMI, NCOMP, ES, EC, HY, NGIRDR, IGRTP, HX, JDIV)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

BEAMI - girder moment of inertia

NCOMP - no. of composite sections along bridge

ES - steel modulus

EC - concrete modulus

HY - spacing of grid points in transverse direction

NGIRDR - no. of girders in transverse direction

IGRTP - four-character alphanumeric variable used to define the type of girder used in the analysis

HX - spacing of grid points in longitudinal direction

JDIV - no. of grid divisions in transverse direction

PURPOSE: This subroutine calculates the additional stiffness that will be gained in the longitudinal direction by composite action of the slab and girder. This additional stiffness is only added to the sections of the bridge where the user specifies that composite action will occur. The output of this routine is used in constructing TABLE3 of the SLAB49 input.

5) SUBROUTINE TAB503 (NDIA, DIAST, DIASP, STDIA, NGIRDR, GIRDSP, JDIV, NGRDIV, HX, HY, BRIDL, IDIV, NX, NASHOK)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

NDIA	- no. of transverse rows of diaphragms located longitudinally along the bridge
DIAST	- starting location of diaphragms, from "backward" end of bridge
DIASP	- spacing of diaphragms along bridge
STDIA	- stiffness of diaphragms
NGIRDR	- no. of girders in transverse direction
GIRDSP	- transverse spacing of girders
JDIV	- no. of grid divisions in transverse direction
NGRDIV	- no. of transverse grid divisions between girders
HX	- spacing of grid points in longitudinal direction
HY	- spacing of grid points in transverse direction
BRIDL	- total bridge length (along a single side)
IDIV	- no. of grid divisions in longitudinal direction
NX	- no. of grid divisions in transverse direction of overhang
NASHOK	- control variable to determine if overhang diaphragms are present

PURPOSE: TAB503 locates and generates the diaphragms contained in the bridge model. If the bridge is square the diaphragms are assumed to be evenly spaced longitudinally and perpendicular to the span centerline. If the bridge is skewed the diaphragms are evenly spaced longitudinally but in the transverse direction are parallel to the skewed end of the slab. One row of diaphragms is located at each end while all others are evenly spaced in between. The output from TAB503 is used in constructing TABLE3 of the SLAB49 input.

6) SUBROUTINE TABLE5 (EC, TK, POIS, IDIV, JDIV, IOPT, IGRTP, SLABI, BRIDL, HX, HY)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

EC - modulus of concrete

TK - slab thickness

POIS - Poisson's Ratio for steel

IDIV - no. of grid divisions in longitudinal direction

JDIV - no. of grid divisions in transverse direction

IOPT - control variable to allow user to specify zero slab stiffness

IGRTP - four-character alphanumeric variable to define the type of girder used in the analysis

SLABI - moment of inertia of slab x-section of unit width

BRIDL - total bridge length (along a single side)

HX - spacing of grid points in longitudinal direction

HY - spacing of grid points in transverse direction

PURPOSE: TABLE5 calculates the twisting stiffness of the bridge slab. The calculated stiffness values are applied to the grid system in exactly the same way as the bending stiffness is applied, i.e., one longitudinal row of grid points at a time, one-half the stiffness each pass. This method of application allows the reduced stiffness of the end of the slab to be modeled correctly. The output from TABLE5 is used to construct TABLE5 in the SLAB49 input.

7) SUBROUTINE TABLE6 (NUMB6)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

NUMB6 - no. of support points at which axial loads must be applied

PURPOSE: The axial loads necessary to model a supported point between two grid points is output by TABLE6. This axial loading was calculated in subroutine TAB303. The support point model used in this program is derived in Ref 2. The output from this routine is used to construct TABLE3 of the SLAB49 input.

8) SUBROUTINE PLTSPN (IDIV, JDIV, HX, HY, IGRTP)

CALLING ROUTINE - TABLE7

Required Routines - CAL-COMP Plot Routines

DEFINITION OF INPUT VARIABLES -

- IDIV - no. of grid divisions in longitudinal direction
- JDIV - no. of grid divisions in transverse direction
- HX - spacing of grid points in longitudinal direction
- HY - spacing of grid points in transverse direction
- IGRTP - four-character alphanumeric variable used to define which type of girder is used in the analysis

PURPOSE: All bridge plotting, spans and support points, is plotted by PLTSPN. A scaling factor is built into this routine such that the bridge will be 13.4 inches long if it is longer than it is wide, and 8.0 inches wide if it is wider than it is long. Once the scaling factor is set everything else will be scaled accordingly. The support points will be designated by placing a diamond at its correct location, even if this is between grid points. Spans are assumed to run from support point to support point; this insures that they line up with a row of longitudinal grid points; therefore they are not designated in any other way. Finally, the coordinate system is noted at the left end and top of the plot.

SUBIBM

SUBJECT: APPLIED LOADING - TRUCK AND DEAD WEIGHT

RELATED SUBROUTINES:

1) DEADLO
2) MOMENT
3) TABLE7
4) TAB107
5) TAB207
6) TAB307
7) TAB407
8) CHECKR
9) LOADUP
10) PLTTRK

1) SUBROUTINE DEADLO (NGIRDR, IDEED, IGIRDL, IDIV, JDIV, DECURB, CONCW, HX, HY, TK, BRIDL, IGRTP)

CALLING ROUTINE - FIX 789

Required Routines - None

DEFINITION OF INPUT VARIABLES -

NGIRDR - no. of girders in transverse direction

IDEED - control variable to determine if bridge deadload is to be calculated

IGIRDL - array containing girder/slab gridpoint deadweights as calculated in TAB203

IDIV - no. of grid divisions in longitudinal direction

JDIV - no. of grid divisions in transverse direction

DECURB - grid deadload to integral curb

CONCW - unit weight of concrete (PCF)

HX - spacing of gridpoints in longitudinal direction

HY - spacing of gridpoints in transverse direction

TK - slab thickness

BRIDL - total bridge length (along a single side)

IGRTP - four-character alphanumeric variable used to define the type of girder used in the analysis

PURPOSE: DEADLO is designed to apply the girder/slab deadweight calculated in TAB203 to the grid system. If girders are present then the deadweight of the girder and contributing slab area are applied along the row of gridpoints corresponding to the transverse location of each girder. If the bridge under study is a flat slab design (no girders) then each longitudinal row of grid points is given a contributory slab deadweight load. If an integral curb is present then its deadweight is added to either the exterior longitudinal row of gridpoints or exterior girders, whichever case is appropriate. The output from DEADLO is used in constructing TABLE3 of the SLAB49 input.

2) SUBROUTINE MOMENT (AREA, BEAMD, CENT, BEAMI, CONCW, NOSTR, PLOS, PULL, EMID, TOP, BOT, SPL)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

AREA - x-sectional area of prestressed concrete girder

BEAMD - girder depth

CENT - distance from centroid of x-section to bottom of x-section

BEAMI - moment of inertia of girder

CONCW - unit weight of concrete (pcf)

NOSTR - no. of prestressing stands present in a girder

PLOS - percent loss of prestressing to creep, etc.

PULL - load in each prestressing cable

EMID - distance from centroid of x-section to centroid of prestressing cables, at centerline of span

TOP - resultant stress at top of girder

BOT - resultant stress at bottom of girder

SPL - span length

PURPOSE: MOMENT calculates the stress, due to girder deadweight and pre-stressing, at the centerline of a prestressed girder. This stress consists of three stresses acting in unison. The first is the stress due to the deadload of the girder. The second is due to the axial compressive force applied by the pre-stressing cable. The third is the stress due to the moment from the eccentricity of the pre-stressing cable. The total stress is computed and printed in the output; it is left up to the analyst to add this stress to any obtained during the study.

3) SUBROUTINE TABLE7 (IDIV, JDIV, HX, HY, IGRTP, BRIDL, IFOR)

CALLING ROUTINE - Main Program

Required Routines - PLTSPN
TAB107
TAB207
TAB307

DEFINITION OF INPUT VARIABLES -

IDIV - no. of grid divisions in longitudinal direction

JDIV - no. of grid divisions in transverse direction

HX - spacing of grid points in longitudinal direction

HY - spacing of grid points in transverse direction

IGRTP - four-character alphanumeric variable used to define the type of girder used in the analysis

BRIDL - total bridge length (along a single side)

IFOR - control variable to define when an input truck is saved for later analysis cases

PURPOSE: TABLE7 is used to control all operations concerning the truck loading of the bridge. Its first step is to read in the truck information supplied by the user. At this point a decision is made as to whether a span plot is requested; if it is PLTSPN is called. A correction for skew is carried out and TAB107 is called. TAB107 reads the selected truck data base and retrieves all stored information. The final action of TABLE7 is to control the distribution of the truck wheel loads to the proper grid locations. This is accomplished by accessing TAB207 and TAB307.

4) SUBROUTINE TAB107 (ITYPE, JTYPE, XPOS, YPOS, DIR, NWHEL, IPLOT, IDIV, HX, NOB, IG, JDIV, HY, BRIDL, IFOR)

CALLING ROUTINE - TABLE7

Required Routines - LOADUP
CHECKR
PLTTRK

DEFINITION OF INPUT VARIABLES -

ITYPE
JTYPE - Two (2) four-character alphanumeric variables which define the truck(s) to be used in the analysis

XPOS - longitudinal location of origin of the local truck coordinates

YPOS - transverse location of origin of the local truck coordinates

DIR - control variable to define the direction of truck

NWHEL - no. of wheels on a given truck

IPLOT - control variable to determine if a truck and span plot is to be made

IDIV - no. of grid divisions in the longitudinal direction

HX - spacing of grid points in longitudinal direction

NOB - control variable to determine if the wheel of any truck is off the side of the bridge

IG - designation, in order of input, of a truck with wheels off the side of the bridge

JDIV - no. of grid divisions in the transverse direction

HY - spacing of grid points in transverse direction

BRIDL - total length of bridge (along a single side)

IFOR - control variable to define when an input truck is saved for later analysis cases

PURPOSE: Subroutines TAB107 and LOADUP perform exactly the same type of work. TAB107 is used if the truck to be input will be retrieved from a truck data base, while LOADUP will be activated if a special truck is to be input with the rest of the data. Once the global location of the origin of the local truck coordinate system is determined, TAB107 locates each wheel in the global coordinate system by adding or subtracting its local coordinate from the global coordinate. TAB107 next checks to see if any of the truck's wheels are off the end, longitudinally; if they are the wheel is given a load of zero (0.0). Next, subroutine CHECKR is called; this routine determines if any truck overlap is present. If two trucks do overlap, then a warning message is issued. Finally, a summary of wheel loads is printed.

5) SUBROUTINE TAB207 (LINE, GRID, D, N, H)

CALLING ROUTINE - TABLE7

Required Routines - None

DEFINITION OF INPUT VARIABLES -

LINE - L/T* location of the closest to, but greater than, row of grid points to a given wheel (integer)

GRID - L/T distance of the closest to, but greater than, row of grid points to a given wheel (real)

D - L/T location of the wheel under analysis

N - no. of grid divisions in the L/T direction

H - spacing of grid points in the L/T direction

*For a single call of TAB207 all variables refer to either longitudinal or transverse values (L or T).

PURPOSE: TAB207 calculates the location of the grid point that is closest to, but greater in both longitudinal and transverse distance from the global origin than, the wheel under study. This is done by a complicated use of "if" statements.

6) SUBROUTINE TAB307 (ALOAD, XDIST, YDIST, HX, HY, J, IDEGRE, DLF)

CALLING ROUTINE - TABLE7

Required Routines - TAB407

DEFINITION OF INPUT VARIABLES -

ALOAD	-	wheel load to be divided among surrounding grids
XDIST	-	longitudinal location of wheel under analysis
YDIST	-	transverse location of wheel under analysis
HX	-	spacing of grid points in longitudinal direction
HY	-	spacing of grid points in transverse direction
J	-	no. of wheel under analysis
IDEGRE	-	control variable to determine amount of grid loading output to be printed
DLF	-	dynamic load factor to be applied to static loads

PURPOSE: With the use of the grid location generated in TAB207 and the global wheel location supplied by TAB107/LOADUP, the subroutine TAB307 divides the wheel load among the four surrounding grids. This division is accomplished by cutting the grid space into four areas by drawing perpendiculars through the wheel location. Each grid is then given a portion of the total load corresponding to the ratio of the area in its adjacent rectangle to the area of the grid space. TAB307 then calls TAB407 to sum all grid loads that may occur at a single grid point.

7) SUBROUTINE TAB407 (I, J, P, IDEGRE)

CALLING ROUTINE - TAB307

Required Routines - None

DEFINITION OF INPUT VARIABLES -

I - longitudinal location of grid point being summed

J - transverse location of grid point being summed

P - load being summed

IDEGRE - control variable to determine amount of grid loading output to be printed

PURPOSE: TAB407 keeps a running tab of the grids that have loads applied to them due to trucks located on the bridge. When a duplicate location is loaded TAB407 sums the two loads together. The user may also direct TAB407 to print out all grids that receive a portion of any wheel load. The summation will still occur internally but each grid loading will be printed as it is received.

8) SUBROUTINE CHECKR (IG, X, Y, X1, X2, Y1, Y2, ITYPE, JTYPE, NWHEL)

CALLING ROUTINE - TAB107

Required Routines - None

DEFINITION OF INPUT VARIABLES -

IG - designation, in order of input, of a truck that overlaps with another truck on the bridge

X - array containing longitudinal location of all truck wheels

Y - array containing transverse location of all truck wheels

X1 - minimum longitudinal location of wheels on truck

Y1 - minimum transverse location of wheels on trucks

X2 - maximum longitudinal location of wheels on truck

Y2 - maximum transverse location of wheels on truck

ITYPE
JTYPE - two (2) four-character alphanumeric variables which define the truck(s) to be used in the analysis

NWHEL - no. of wheels on the truck under analysis

PURPOSE: CHECKR is designed to record the location of each truck as it is processed by TAB107/LOADUP. When the second or later truck is being processed it will be brought into CHECKR and with a series of "If" statements checked to insure that it does not overlap with any other truck. The second will be checked against the first, the third against the second, etc., until all trucks are processed. If a truck is found that overlaps with one or more trucks then all overlapping truck identifications will be printed.

9) SUBROUTINE LOADUP (ITYPE, JTYPE, XPOS, YPOS, DIR, NWHEL, IPLOT, IDIV, JDIV, HX, HY, NOB, IG, BRIDL, IFOR)

CALLING ROUTINE - TABLE7

Required Routines - CHECKR
PLTTRK

DEFINITION OF INPUT VARIABLES -

ITYPE
JTYPE - two (2) four-character alphanumeric variables which define the truck(s) to be used in the analysis

XPOS - longitudinal location of origin of the local truck coordinates

YPOS - transverse location of origin of the local truck coordinates

DIR - control variable to define the direction of truck

NWHEL - no. of wheels on a given truck

IPLOT - control variable to determine if a truck and span plot is to be made

IDIV - no. of grid divisions in the longitudinal direction

HX - spacing of grid points in longitudinal direction

NOB - control variable to determine if the wheel of any truck is off the side of the bridge

IG - designation, in order of input, of a truck with wheels off the side of the bridge

JDIV - no. of grid divisions in the transverse direction

HY - spacing of grid points in transverse direction

BRIDL - total length of bridge (along a single side)

IFOR - control variable to define when an input truck is saved for later analysis cases

PURPOSE: Subroutine LOADUP and subroutine TAB107 perform exactly the same type of work. TAB107 is used if the truck to be input will be retrieved from a truck data base, while LOADUP will be activated if a special truck is to be input with the rest of the data. Once the global location of the origin of the local truck coordinate system is determined, LOADUP locates each wheel in the global coordinate system by adding or subtracting its local coordinate from the global coordinate. LOADUP next checks to see if any of the truck's wheels are off the end, longitudinal, and if they are the wheel is given a load of zero (0.0). Next, subroutine CHECKR is called; this routine determines if any truck overlap is present. If two trucks do overlap, then a warning message is issued. Finally, a summary of wheel loads is printed.

10) SUBROUTINE PLTTRK (NWHEL, X1, X2, Y1, Y2)

CALLING ROUTINE - TAB107
LOADUP

Required Routines - CAL-COMP Plot Routines

DEFINITION OF INPUT VARIABLES -

NWHEL - no. of wheels on truck being plotted

X1 - minimum longitudinal location of wheels on truck being plotted

Y1 - minimum transverse location of wheels on truck being plotted

X2 - maximum longitudinal location of wheels on truck being plotted

Y2 - maximum transverse location of wheels on truck being plotted

PURPOSE: PLTTRK is used to make a CAL-COMP plot of each truck located on the bridge. Each truck is scaled to the span size calculated in PLTSPN. The direction of each truck is designated by which way the arrows mark the location of each wheel point. If a truck is determined to have one or more wheels off the side of the bridge no plot of its location will be made. If a truck is determined to have one or more wheels located off the end of the bridge then it will be plotted so that its location is understood.

1) SUBROUTINE TABLE8 (HX, HY, ISPEL)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

HX - spacing of grid points in longitudinal direction

HY - spacing of grid points in transverse direction

ISPEL - control variable to determine if TABLE8 and TABLE9 input for SLAB49 is to be generated

PURPOSE: The areas for special output such as deflections, x-moments, y-moments, and stresses are defined with the use of TABLE8. TABLE8 is specifically designed to allow the user to specify restricted areas for special output. This may be a line of grid points along a girder, or several girders, or it may consist of a restricted area of slab output. If a line of grid points is specified, girder information, then an area of grid points, slab information, cannot be specified. This restriction is due to SLAB49 constraints, not pre-processor constraints. TABLE8 generates all the information needed by TABLE8 of the SLAB49 input.

2) SUBROUTINE TABLE9 (ISPEL)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

ISPEL - control variable to determine if TABLE8 and TABLE9 input for SLAB49 is to be generated

PURPOSE: TABLE9, unlike TABLE8, is used to specify regions of general output. General output is defined as all the information contained within a specified area. In many cases this may be more information than the user is looking for. In order to use all the capabilities of the post-processor, though, it is necessary that TABLE9 be used to specify all areas where the post-processor will be required to print or plot any information. Due to this special requirement the default for TABLE9 output is the entire bridge. The user should be aware of the consequences of overriding this default.

3) SUBROUTINE FIX789 (N, NGIRDR, IDEED, ISPEL, NUMBB, NUMB9, IA, JA, XA, IB, JB, IC, JC, ID, JD, IGIRDL, IGRTP, IDIU, JDIV, CONCW, HX, HY, TK, BRIDL, DECURB)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

- N - no. of grids loaded by trucks
- NGIRDR - no. of girders in transverse direction
- IDEED - control variable to determine if bridge deadload is to be calculated
- ISPEL - control variable to determine if TABLE8 and TABLE9 input for SLAB49 is to be generated
- NUMB8 - no. of special output areas defined by TABLE8
- NUMB9 - no. of general output areas defined by TABLE9
- IA - vector containing longitudinal locations of loaded grids
- JA - vector containing transverse locations of loaded grids
- XA - vector containing grid loads of trucks
- IB - vector containing minimum longitudinal locations of special output areas
- JB - vector containing minimum transverse locations of special output areas
- IC - vector containing maximum longitudinal locations of special output areas
- JC - vector containing maximum transverse locations of special output areas
- ID - vector containing minimum longitudinal locations of general output areas
- JD - vector containing maximum longitudinal locations of general output areas
- IGIRDL - array containing girder/slab deadweights as calculated by TAB203
- IGRTP - four-character alphanumeric variable used to define the type of girder used in the analysis
- IDIV - no. of grid divisions in the longitudinal direction
- JDIV - no. of grid divisions in the transverse direction
- CONCW - unit weight of concrete (pcf)

HX - spacing of grid points in the longitudinal direction

HY - spacing of grid points in the transverse direction

TK - slab thickness

BRIDL - total bridge length (along a single side)

DECURB - deadweight due to integral curb to be applied to each edge grid point

PURPOSE: The sole purpose of FIX789 is to allow the information generated by TABLE7, TABLE8, and TABLE9 to be printed in a logical sequence. The first step taken is to print the truck-grid loadings as recorded by TAB407. Next, special output areas, generated in TABLE8, are printed. Finally TABLE9, general output areas, is printed. This method is necessary since these three tables must come last in the SLAB49 input, the number of cards must be known at the beginning. This method allows the processing and counting to be carried out first and the printing to be carried out last.

1) SUBROUTINE SETPST (BRIDL)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

BRIDL - total length of bridge (along a single side)

PURPOSE: SETPST allows the user to input post-processor controls at the time pre-processor input is specified. This method allows the preparation of only one data deck, thus simplifying and minimizing user/program interaction. The type of post-processor input transferred by SETPST includes number and type of plots and control of general SLAB49 output. SETPST also prints a grid orientation note to be used when interpreting SLAB49 output.

2) SUBROUTINE READY

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF OUTPUT ARRAYS -

COMIY - array to transfer centroids, moments of inertia for composite sections, plus the centroid of the beam x-section

IGIRL - array to transfer the girder/slab deadweights as computed in TAB203

BRIG1 - array to transfer span lengths and locations of composite sections

BRIG2 - array to transfer (HX, HY, TK, TURN, BEAMI, BEAMD, EC, ES, TOPS, BOTS)

IBRG2 - array to transfer (IGRIP, INAM, LRD, NGRDER, NCOMP, IDIV, JDIV, JNAM, NSPON, NPLOT, NDAT, NPUT)

CURHW - array to transfer (CUHT, CUWD, HAUNCH)

NAMS - array to transfer names of girders to be plotted and/or printed in post-processor

YDAT - array to transfer longitudinal locations of starting and ending positions of girders to be plotted and/or printed in post-processor

NFORM - informational data to be passed to post-processor

IYNUMB9 - array containing the locations and number of general output areas to be printed in post-processor

BRIG3 - array containing information defining the skew of the bridge

PURPOSE: The sole purpose of READY is to pass the necessary problem information to the post-processor in the most efficient manner possible. READY accomplishes this with the use of one written statement of the condensed information. Upon arrival in the post-processor INPT sorts the information into the recognized form.

APPENDIX C

FORTRAN LISTING OF PRE-PROCESSOR - SLBDG4

```
C-----PROGRAM SLBDG4 AS RECD FROM UT ON 24 MAY 79 AND WRITTEN HERE ON   080C80
C
C       MINOR MODIFICATIONS WERE MADE BY R. STCLLIS DURING CCT 80.      080C80
C       THIS PROGRAM IS A BRIDGE DATA GENERATOR FCR THE SLAB49 PROGRAM. 080C80
C       PROGRAM NAME ORIGINALLY CALLED SUB - HEREAFTER CALLED SLBDG4.   080C80
C
C     PROGRAM SUB (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=CUTPUT,TAPE7,         080C80
C    *            TAPE8,TAPE9,TAPE10,TAPE11,TAPE13,PLTFILE)             080C80
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /A      / ILINE,JLINE,XLINE,YLINE,IPTN(300),JPTN(300),
     C                 XLOAD(300),N,IDUMC
      COMMON /B      / SCLX,SCLY,YT,IDIR,NGIRDR,NGRDIV,NX,CVERH,OVERH1,
     C                 NSPAN,IDUM1
      COMMON /C      / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                 JDIST1(12,50),JDIST2(12,50)
      COMMON /D      / IX1(20),IY1(2C),IX2(20),IY2(20),IDEFL(20),
     C                 IXMOM(20),IYMCM(2C),ISTRS(20),NUMB8,IDUM8
      COMMON /E      / IY19(20),IY29(20),NUMB9,IDUM9
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /INF    / INFO(58),INFO3,INF3
      COMMON /RD     / HX,HY,TK,TURN,BEAMI,BEAMD,EC,ES,TCP ,BCT ,
     C                 IGRTP,INAM,LRD,NGRDER,NCOMP,IDIV,JCIV,JNAM,
     C                 NSPON,NPLOT,NDAT,NPUT,NAMS(50),YDAT(50,2)
      COMMON /BM     / EWID(2),YBAR(2),CCMPI(2),CENT,CUHT,CUWD,HAUNCH
      DIMENSION        IDIT(20)
      DATA             IPRS,NOBM,IPAN,ISTE /4HPRES,4HSLAB,4HREIN,4HSTEE/
     C                 ,LFOR,IENPP,ISP,INPU /2HLF,4H    ,4HCEAS,4HINPU/
     C                 ,IPRBL/4H****/
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          IEMPTY,ITHETA,ITURN,IANGLE
C---
C---  ASSIGN TAPE UNITS
      NINP      = 5
      NOUT      = 6
      NT7       = 7
      NT8       = 8
      NT9       = 9
      NT10      = 10
      NT11      = 11
      NT3       = 13
      NPLT      = 71
C---  INTIALIZE VARIABLES FOR COMPUTATION CONTROL
      IFOR      = 0
      LOT2      = 0
      ICHO      = 0
C---  SET INTIAL POST PROCESSOR OPTICNS
      NPLOT     = 0
      NDAT      = 1
      NPUT      = 0
C---  SKIP PAGE
      WRITE (NOUT,11)
C---  READ LABELING INFORMATION CONCERNING THE PRCBLEM.
      READ  (NINP,301) (INFO(I),I = 1,40)
      WRITE (NOUT,623) (INFO(I),I = 1,40)
    1 READ  (NINP,302) INF3,INFO3,(INFO(I),I = 41,58)
      IF (ISP.EQ.INF3) GOTO 998
      IF (LOT2.NE.0)  WRITE (NOUT,11)
      WRITE (NOUT,624) INF3,INFO3,(INFO(I),I = 41,58)
C---  CHECK FOR BLANK PROBLEM NUMBER
      IEMPTY    = IENPP.EQ.INF3.AND.IENPP.EQ.INFC3
C---  PRINT NOTE IF PROBLEM NUMBER IS BLANK
      IF (IEMPTY) WRITE (NOUT,619)
      IF (IEMPTY) INF3 = IPRBL
```

```
C---   READ OPTIONS
       READ  (NINP,501) LRD,NCHO,I3D,IDEED,IPOST,ISPEL
C---   CHECK FOR ECHO PRINT OPTION
       IF (NCHO.EQ.1) ICHO = 1
       WRITE (NOUT,601)
       WRITE (NOUT,602) LRD,NCHO,I3D,IDEED,IPOST
C---   SET VALUES FOR SLAB 49 OPTIONS
       IPRSTR     = 1
       IPROF      = 0
       ISTCHK     = 0
       IOPT       = 0
       IDEAD      = IABS (IDEED-1)
                         MULTOP = 1
       IF (LOT2.NE.0) MULTOP = -1
C---   READ PHYSICAL DATA CONCERNING PROBLEM
       IF (MULTOP.EQ.-1) GOTO  90
       READ  (NINP,502) RWIDTH,OVERH1,IIDV,JDIV,TK,NTWT,THETA,CUHT,CUWD
       READ  (NINP,503) CONCW,CONCS,PCIS,EC,ES
       READ  (NINP,504) NGIRDR,IGRTP,INAM,JNAM,NCSTR,PULL,PLCS,
      C                  EMID,HAUNCH
       WRITE (NOUT,603) RWIDTH,OVERH1,IIDV,JDIV,TK,NTWT,THETA
       IF (CUHT.GT.0.01) WRITE (NOUT,604) CUHT,CUWD
       TK         = TK/12.0
C---   CALC VALUES FOR INTEGRAL CURB
       CURBA      = (CUHT*CUWD)
       CURBI      = CURBA*((CUHT**2/12.0)+((.5*(TK+CUHT))**2))
C---   CALCULATE HX,OVERHANG,AND GIRDER SPACING
       IF (IGRTP.EQ.NOBM) NGIRDR = 0
       GIRDR      = NGIRDR
       GIRDSP     = (RWIDTH - 2. * OVERH1) / (GIRDR-1.)
C---   PRINT CALCULATED GIRDER SPACING
       IF (IGRTP.EQ.NOBM) GOTO  5
       WRITE (NOUT,621) GIRDSP
       GOTO 10
C---   RECOMPUTE GIRDER SPACING FOR SLAB BRIDGE
     5 GIRDSP     = RWIDTH
    10 CONTINUE
C---   COMPUTE THE NUMBER OF DIVISIONS BETWEEN GIRDERS
       DVII       = IIDV
       AGRD       = 1.0 - 2.0 * OVERH1 / RWIDTH
       BGRD       = NGIRDR - 1
       ND         = DVII * AGRD / BGRD + 0.5
                          NGRDIV = ND
       IF (IGRTP.EQ.NOBM) NGRDIV = IIDV
C---   CALC LENGTH OF DIVISIONS ACROSS BRIDGE
       GRDIV      = NGRDIV
       HX         = (GIRDSP / GRDIV)
                          X = (OVERH1/HX+.5)
       IF (IGRTP.EQ.NOBM) X = 0.0
       NX         = X
       XNX        = NX
C---   CALC NEW OVERHANG (REDINED FCR COMPUTATIONS)
       OVERH      = HX * XNX
       IF (ABS(OVERH1-OVERH).LT.0.0001) GCTO 15
       WRITE (NOUT,613) OVERH
    15 CONTINUE
C---   CALCULATE DIVISIONS ACROSS BRIDGE
                          IDIV = 2 * NX + (NGIRDR-1) * NGRDIV
       IF (IGRTP.EQ.NOBM) IDIV = NGRDIV
       IF (IIDV.NE.IDIV)  WRITE (NOUT,626) IDIV
C---   SET VALUES FOR MODULUS OF ELASTICTY
       IF (EC.LT.10.0) EC = 33.0 * SQRT(CCNCS) * CCNCW**1.5
       IF (ES.LT.10.0) ES = 29E+6
```

```
   20 WRITE (NOUT,605) CONCW,CONCS,PCIS,EC,ES
      SC          = ES/EC
C---  CONVERT CONCRETE PARAMETERS
      EC          = EC*(144./1000.)
      ES          = ES*(144./1000.)
      CONCW       = CONCW/1000.
C---  SET SPRING CONSTANT
      SPNGK       = 1.000E+8
C---  CHECK FOR LEGAL GIRDER NAME
      IF (IGRTP.EQ.ISTE.OR.IGRTP.EQ.IPAN.OR.IGRTP.EQ.IPRS) GOTO 25
      IF (INAM.EQ.INPU) GOTO 25
      IF (IGRTP.EQ.NOBM) GOTO 30
      WRITE (NOUT,620) IGRTP
      STOP
   25 CONTINUE
      WRITE (NOUT,606) NGIRDR,IGRTP,INAM,JNAM
      IF (IGRTP.NE.IPRS) GOTO 35
      WRITE (NOUT,607) NOSTR,PULL,PLOS,EMID,HAUNCH
      PLOS        = PLOS / 100.0
      HAUNCH      = HAUNCH / 12.0
      EMID        = EMID / 12.0
      GOTO 35
   30 WRITE (NOUT,618)
   35 CONTINUE
C---  CALCULATE SKEW ANGLE AND TANGENT
      THEA        = THETA/180.*4.*ATAN(1.)
      ANGLE       = 0.0
      IANGLE      = (ABS(COS(THEA))).LT.0.0001
      IF (IANGLE) GOTO 40
      ANGLE       = SIN(THEA)/COS(THEA)
                        THETA = 110.
      IF (NTWT.EQ.LFOR) THETA = 80.
   40 ARM         = FLOAT(IDIV)*HX
C---  SET LOGICAL VARIABLE FOR BRIDGE SKEW
      ITHETA      = NTWT.EQ.LFOR
C---  TEST TO DETERMINE DIRECTION OF IDIV AND JDIV COORDINATES.
      ADV         = IDIV
      BDV         = JDIV
                        TURN = 1.
      IF (ADV/BDV.GT.1.0) TURN = -1.0
C---  SET LOGICAL VARIABLE FOR BRIDGE ORIENTATION
      ITURN       = TURN.LT.0.0
C---  CALCULATE TRANSVERSE LOCATIONS OF GIRDERS
      IGIRDL(1) = NX
      DO 45   I = 2,NGIRDR
   45 IGIRDL(I) = IGIRDL(I-1)+NGRDIV
      IF (IGRTP.EQ.NOBM) GOTO 55
C---  CHECK FOR INPUT GIRDER
      IF (INAM.NE.INPU) GOTO  50
      CALL BMINP (BEAMI,TK,GIRDSP,AREA,OVERH1,PCIS,
     C             BEAMD,SC,IGRTP,SLABI,ARINT,AREND,NX)
      GOTO 55
C---  RETREIVE GIRDER INFORMATION FROM TAPE
   50 CALL BMGET (INAM,JNAM,BEAMI,TK,GIRDSP,AREA,OVERH1,PCIS,
     C             BEAMD,SC,IGRTP,SLABI,ARINT,AREND,NX)
   55 CONTINUE
C---  READ DIAPHRAGM INFORMATION
      READ  (NINP,506) NDIA,DIAST,DIASP,STDIA,NASHCK
      WRITE (NOUT,11)
      WRITE (NOUT,608) NDIA
      IF (NDIA.EQ.0) GOTO  65
      WRITE (NOUT,609) DIAST,DIASP,STDIA
   65 CONTINUE
```

```
C---  READ SPAN DATA
      BRIDL      = 0.0
      READ  (NINP,507) NSPAN
      READ  (NINP,509) (SPANL(I),I=1,NSPAN)
      WRITE (NOUT,610) NSPAN
C---  COMPUTE BRIDGE LENGTH
      DO   70 I  = 1,NSPAN
      BRIDL      = BRIDL+SPANL(I)
   70 WRITE (NOUT,611) I,SPANL(I)
      WRITE (NOUT,612) BRIDL
C---  CALCULATE LENGTH OF DIVISIONS ALONG BRIDGE
      HY         = (BRIDL+ARM*ANGLE)/FLOAT(JDIV)
C---  READ COMPOSITE SECTION INFORMATION
      READ  (NINP,507)  NCOMP
      WRITE (NOUT,614)
      WRITE (NOUT,617) NCOMP
      YDIST1(1) = 0.0
      YDIST2(1) = 0.0
      IF (NCOMP.EQ.0) GOTO  80
      READ  (NINP,509) (YDIST1(I),YDIST2(I),I=1,NCOMP)
      DO 75   I = 1,NCOMP
   75 WRITE (NOUT,616) I,YDIST1(I),I,YDIST2(I)
   80 CONTINUE
C---  RESET COMPOSITE ZONES FOR R.C. BRIDGES
      IF (IGRTP.NE.IPAN)      GOTO 85
      ACRCHK     = ABS (YDIST2(1) - BRIDL)
      IF (ACRCHK.LT.0.001)    GOTO 85
      NCOMP      = 1
      YDIST1(1) = 0.0
      YDIST2(1) = BRIDL
      WRITE (NOUT,622)
   85 CONTINUE
C---  CALL ROUTINE TO CALCULATE TRUCK LOADS AND MAKE PLOT
   90 CALL TABLE7 (IDIV,JDIV,HX,HY,IGRTP,BRIDL,IFOR)
C---  CALL ROUTINES TO READ INFO AND COMPUTE VALUES FOR OUTPUTS
      YDAT(1,1) = 0.0
      YDAT(1,2) = BRIDL
      IF (IPOST.EQ.1) GOTO 95
      CALL SETPST (BRIDL)
      CALL TABLE8 (HX,HY,ISPEL)
      CALL TABLE9 (ISPEL)
   95 CONTINUE
      IF (IGRTP.NE.IPRS) GOTO 100
C---  CALL ROUTINE TO COMPUTE PRESTRESSING VALUES
      SPL        = SPANL (1)
      CALL MOMENT (AREA,BEAMD,CENT,BEAMI,CONCW,NCSTR,FLOS,
     C                PULL,EMID,TOP,BOT,SPL)
  100 IONE       = 1
      IZERO      = 0
C---  INITIALIZE GENERATED SLAB 49 INPUT WITH TABLE 1 INFO
      IF (MULTOP.EQ.-1) GOTO 105
      WRITE (NT3,301) (INFO(I),I = 1,20)
      WRITE (NT3,301) (INFO(I),I = 21,40)
  105 WRITE (NT3,302) INF3,INFO3,(INFO(I),I = 41,58)
                              MM    = 1
      IF (CURBA.LT.0.001) MM    = 0
                              KK    = 0
      IF (NDIA.NE.0)          KK    = 1
                              LL    = 1
      IF (IGRTP.EQ.IPAN)  LL    = 0
                              LM    = 1
      IF (IGRTP.EQ.NOBM)  LM    = 0
C---  CALCULATE THE NUMBER OF CARDS FOR TABLE 3
```

```
                          NPT3 = NGIRDR
      IF (IGRTP.EQ.NOBM)  NPT3 = IDIV + 1
      NUMB3       = ((IDIV+1)*2)+LM*(NPT3*2+NCOMP*2*NPT3*LL)+(NSPAN+1)
     C              *NPT3*2+(2*NDIA+4)*(NGIRDR-1+(NASHCK*2))*KK+4*MM
C---  CALC NUMBER OF CARDS FOR TABLE 5
      NUMB5       = IDIV
C---  CALC NUMBER OF CARDS FOR TABLE 6
      NUMB6       = NPT3 * (NSPAN+1)
C---  CALC NUMBER OF CARDS IN TABLE 7
      NUMB7       = N + 2 * IDEED * NPT3
      IF (MULTOP.EQ.-1) GOTO 110
C---  WRITE TABLE 1 CONTROL DATA ON TAPE
      WRITE (NT3,303) IZERO,IZERO,IZERO,IZERO,IZERO,IZERO,IZERO,IZERO,
     C                MULTOP
      WRITE (NT3,304) IONE,NUMB3,IZERO,NUMB5,NUMB6,NUMB7,NUMB8,NUMB9,
     C                ISTCHK,IPRSTR,IPROF,I3D
C---  CALCULATE NUMBER OF CARDS FOR PROBLEM
      LOT1        = NUMB3 + NUMB5 + NUMB6 + NUMB7 + NUMB8 + NUMB9 + 1
      GOTO 115
C---  WRITE TABLE 1 FOR OFFSPRING
  110 WRITE (NT3,303) IONE,IONE,IZERO,IONE,IONE,IZERO,IZERO,IZERO,MULTOP
      WRITE (NT3,304) IZERO,IZERO,IZERO,IZERO,IZERO,NUMB7,NUMB8,NUMB9,
     C                ISTCHK,IPRSTR,IPROF,I3D
C---  COUNT NUMBER OF CARDS FOR OFFSPRING PROBLEM
      LOT1        = NUMB7 + NUMB8 + NUMB9
  115 CONTINUE
C---  WRITE TABLE 2 ON TAPE
      IF (MULTOP.EQ.-1) GOTO 140
      IF (ITURN) GOTO 120
      WRITE (NT3,305) IDIV,JDIV,HX,HY,PCIS,TK
      GOTO 125
  120 WRITE (NT3,305) JDIV,IDIV,HY,HX,PCIS,TK
  125 CONTINUE
C---  CALL SUBROUTINE TO SET UP -SLAB STIFFNESS- TABLE 3
      CALL TAB103 (IDIV,JDIV,EC,POIS,TK,IOPT,IGRTP,SLABI,
     C             HX,BRIDL,HY,DX)
C---  CALL ROUTINE FOR -GIRDER STIFNESS- TABLE 3
      CALL TAB203 (OVERH1,JDIV,GIRDSP,ES,EC,AREA,NGIRDR,BEAMI,
     C             HY,CONCW,TK,IGRTP,ARINT,AREND,HX,BRIDL)
      IF (IGRTP.EQ.NOBM.OR.IGRTP.EQ.IPAN) GOTO 130
C---  CALL ROUTINE FOR -ADDITIONAL COMP. STIFFNESS- TABLE 3
      CALL TAB403 (BEAMI,NCOMP,ES,EC,HY,NGIRDR,IGRTP,HX,JDIV)
  130 CONTINUE
C---  ADD CURB STIFFNESS AND DEAD WT.
      CALL TABCUR (IDIV,JDIV,CURBA,CURBI,NX,EC,CONCW,DECURB,
     C             IGRTP,HX,HY,BRIDL,NT3)
C---  CALL ROUTINE FOR -SUPPORTS- TABLE 3
      CALL TAB303 (NSPAN,JDIV,HY,NPT3,SPNGK,IGRTP,HX)
C---  CALL ROUTINE FOR -DIAPHRAGM STIFFNESS- TABLE 3
      CALL TAB503 (NDIA,DIAST,DIASP,STDIA,NGIRDR,GIRDSP,JDIV,
     C             NGRDIV,HX,HY,BRIDL,IDIV,NX,NASHCK)
C---  CALL ROUTINE TO SET UP TABLE 5 ON TAPE
      CALL TABLE5 (EC,TK,POIS,IDIV,JDIV,IOPT,IGRTP,SLABI,BRIDL,
     C             HX,HY)
C---  CALL ROUTINE TO SET UP TABLE 6 ON TAPE
      CALL TABLE6 (NUMB6)
C---  CALL ROUTINE TO WRITE TABLE 7,8,AND 9 TO TAPE
  140 IF (ITURN) GOTO 145
      CALL FIX789 (N,NGIRDR,IDEED,ISPEL,NUMB8,NUMB9,IPTN,JPTN,XLOAD,
     C             IX1,IY1,IX2,IY2,IY19,IY29,IGIRDL,IGRTP,
     C             IDIV,JDIV,CONCW,HX,HY,TK,BRIDL,DECURB)
      GOTO 150
  145 CALL FIX789 (N,NGIRDR,IDEED,ISPEL,NUMB8,NUMB9,JPTN,IPTN,XLOAD,
```

```
    C              IX1,IY1,IX2,IY2,IY19,IY29,IGIRDL,IGRTP,
    C              IDIV,JDIV,CONCW,HX,HY,TK,BRIDL,DECURB)
  150 CONTINUE
C---  COMPUTE THE TOTAL NUMBER OF CARDS FCR JOB
      LOT2       = LOT2 + LOT1
C--- RENAME VARIABLES FOR POSTPROCESSING TAPE
                          NGRDER = NGIRDR
      IF (IGRTP.EQ.NOBM) NGRDER = IDIV + 1
      NSPON      = NSPAN
C---  CALL ROUTINE TO MAKE THE TAPE FCR FOSTPROCESSING
      CALL READY
      GOTO 1
  998 WRITE (NT3,306) INF3,INFO3,(INFO(I),I=41,54)
C---  ECHO PRINT THE INPUT FOR SLAB49
      IF (ICHO.NE.1) GOTO 160
      WRITE (NOUT,11)
      WRITE (NOUT,625)
      REWIND NT3
      LIO        = 6 + (3*(LRD-1)) + LOT2
      DO 155 J   = 1,LIO
      READ  (NT3,301)  (IDIT(I),I = 1,20)
  155 WRITE (NOUT,623) (IDIT(I),I = 1,20)
  160 CONTINUE
C---
   11 FORMAT (5H1      ,80X,10HI----TRIM ,//)
  301 FORMAT (20A4)
  302 FORMAT (2A4,2X,17A4,1A2)
  303 FORMAT (8I5,5X,I5)
  304 FORMAT (8I5,5X,4I5)
  305 FORMAT (2I5,10X,4E10.3)
  306 FORMAT (10X,2A4,5X,14A4)
  501 FORMAT (7I5)
  502 FORMAT (2F10.3,2I5,F1C.3,A2,F8.3,2F10.3)
  503 FORMAT (3F10.3,2E10.3)
  504 FORMAT (1I5,5X,1A4,6X,2A4,2X,1I5,5X,4F10.0)
  506 FORMAT (1I5,5X,2F10.3,E10.3,1I5)
  507 FORMAT (I5)
  509 FORMAT (8F10.3)
  601 FORMAT (//5X,23HTABLE 1. CONTRCL DATA      )
  602 FORMAT (
     C    /8X,45HNUMBER OF LOAD CASES                             ,I10,
     C    /8X,45HPRINT OF GENERATED SLAB49 INPUT (1=YES)          ,I10,
     C    /8X,45H3-D PLOT OPTION                                  ,I10,
     C    /8X,45HBRIDGE DEAD WEIGHT OPTION                        ,I10,
     C    /8X,45HTABLE 4 RETAINED FROM PREVIOUS PRCBLEM           ,I10,//)
  603 FORMAT (5X,25HTABLE 2. BRIDGE GEOMETRY    ,/
     C    /7X,14HA. DESCRIPTION
     C    /8X,45HTOTAL BRIDGE WIDTH                     (FT)      ,F10.2,
     C    /8X,45HSLAB OVERHANG                          (FT)     ,F10.2,
     C    /8X,45HDIVISIONS ACROSS BRIDGE - TRANSVERSE             ,1I10,
     C    /8X,45HDIVISIONS ALONG BRIDGE - LONGITUDINAL            ,1I10,
     C    /8X,45HSLAB THICKNESS                         (IN)      ,F10.2,
     C    /8X,45HBRIDGE ANGLE OF SKEW              (DEGREES)      ,1X,A2,
     C    ,F7.2 )
  604 FORMAT ( 8X,20HINTEGRAL CURB HEIGHT  ,15X,10H (FT)         ,F10.2,
     C    /8X,20HINTEGRAL CURB WIDTH    ,15X,10H (FT)            ,F10.2)
  605 FORMAT (/7X,23HB. MATERIAL PROPERTIES
     C    /8X,45HCONCRETE WEIGHT                       (PCF)      ,F10.2,
     C    /8X,45HCONCRETE STRENGTH                     (PSI)      ,F10.2,
     C    /8X,45HPOISSONS RATIO                                   ,F10.4,
C    C    /8X,45HMODULUS OF ELASTICITY - CONCRETE      (PSI)      ,E10.3,      CDC
     C    /8X,45HMODULUS OF ELASTICITY - CONCRETE      (PSI)      ,1PE10.3,    IBM
C    C    /8X,45HMODULUS OF ELASTICITY - STEEL         (PSI)      ,E10.3)      CDC
```

```
     C    /8X,45HMODULUS OF ELASTICITY - STEEL       (PSI)       ,1PE10.3)        IBM
  606 FORMAT (/7X,23HC. GIRDER INFORMATICN      ,
     C    /8X,45HNUMBER OF GIRDERS                                ,I10,
     C    /8X,45HGIRDER TYPE                                      ,1X,A4
     C    /8X,45HGIRDER NAME                                      ,1X,2A4)
  607 FORMAT (
     C     8X,45HNUMBER OF STRANDS                                ,I10,
     C    /8X,45HINITIAL FORCE IN STRAND            (KIPS)        ,F10.2,
     C    /8X,45HPERCENT LOSS OF FORCE IN STRAND                  ,F10.2,
     C    /8X,45HECCENTRICITY (MIDSPAN)               (IN)        ,F10.2,
     C    /8X,45HHAUNCH                               (IN)        ,F10.2)
  608 FORMAT (/,7X,13HD. DIAPHRAGMS,/,8X,20HNUMBER CF DIAPHRAGMS25X,I10)
  609 FORMAT (
     C     8X,45HDISTANCE - START TO FIRST DIAPHRAGM (FT)         ,F10.2,
     C    /8X,45HDISTANCE - START TO LAST DIAPHRAGM  (FT)         ,F10.2,
C    C    /8X,45HSTIFFNESS OF DIAPHRAGMS (EI)     (LB-IN)         ,E10.3)        CDC
     C    /8X,45HSTIFFNESS OF DIAPHRAGMS (EI)     (LB-IN)         ,1PE10.3)      IBM
  610 FORMAT (/7X,19HE. SPAN INFORMATION,/8X,21HNUMBER OF SPANS INPUT
     C          24X,I10   )
  611 FORMAT (8X,14HLENGTH OF SPAN  I3,19X,4H(FT)5X,F10.2 )
  612 FORMAT (8X,22HTOTAL LENGTH OF BRIDGE  14X,4H(FT)5X,F10.2)
  613 FORMAT (8X,
     C    45HOVERHANG REDEFINED FOR CCMPUTATIONS (FT)       ,F10.2)
  614 FORMAT (/7X,28HF. ZONES OF COMPOSITE ACTICN    )
  616 FORMAT (8X,20HBEGIN COMPOSITE ZONE I3,13X,4H(FT)5X,F10.2,/,
     C         8X,18HEND COMPOSITE ZONE,I3,15X,4H(FT)5X,F10.2)
  617 FORMAT (8X,33HNUMBER OF COMPOSITE ZONES INPUT   12X,I10)
  618 FORMAT (/7X,23HC. GIRDER INFORMATICN      ,/,
     C  /8X,26HTHIS BRIDGE HAS NO GIRDERS         )
  619 FORMAT (//,8X,45HPRCBLEM NUMBER IS BLANK AND HAS BEEN REPLACED  )
  620 FORMAT (//,8X,13HGIRDER TYPE *,A4,13H* IS ILLEGAL    )
  621 FORMAT (8X,23HCOMPUTED GIRDER SPACING,13X,4H(FT),5X,F10.2)
  622 FORMAT (/8X,44HZONES OF COMPOSITE ACTICN ARE REDEFINED TO
     C    /8X,32HBRIDGE LENGTH FOR THIS PRCBLEM.             )
  623 FORMAT (5X,20A4)
  624 FORMAT (//,5X,7HPROBLEM,/,5X,2A4,2X,17A4,1A2)
  625 FORMAT (5X,21HINPUT DATA FOR SLAB49  ,
     C     //,7X,42HUNITS ARE KIPS AND FEET FOR SLAB49 INPUT    ,//)
  626 FORMAT (8X,35HDIVISIONS ACROSS BRIDGE - REDEFINED,10X,1I10)
C---
      STOP
      END
      SUBROUTINE BMGET (INAM,JNAM,BEAMI,IK,GIRDSP,AREA,OVERH1,POIS,
     C                  BEAMD,SC,IGRTP,SLABI,ARINT,AREND,NX)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BM     / EWID(2),YBAR(2),CCMPI(2),CENT,CUHT,CUWD,HAUNCH
      DATA             IPRS,IPAN,ISTE /4HPRES,4HREIN,4HSTEE/
C---
C---  THIS SUBROUTINE RETRIEVES BEAM INFCRMATICN
C---  IT ALSO COMPUTES THE COMPOSITE SECTION YBAR AND I
C---
C---  DETERMINE DATA BASE TO BE USED
                           NTAPE = NT8
      IF (IGRTP.EQ.IPRS) NTAPE = NT7
      IF ( IGRTP.EQ.IPAN) NTAPE = NT10
      REWIND NTAPE
C---  READ NUMBER OF GIRDERS IN DATA BASE
      READ  (NTAPE) NBEAM
      IFIND    = -1
      IF (IGRTP.EQ.IPAN) GOTO 1
      DO 2   I = 1,NBEAM
C---  READ DATA BASE INFORMATICN FOR PRESTRESSED OR STEEL
      READ  (NTAPE) IBMTP,JBMTP,AREA1,BEAM,DBEAM,TFL,CENTR
```

```
C---  CHECK FOR CORRECT GIRDER
      IF (INAM.NE.IBMTP) GOTO 2
      IF (JNAM.NE.JBMTP) GOTO 2
      WRITE (NOUT,606) AREA1,BEAM,DBEAM,TFL,CENTR
C---  SET VALUES FOR GIRDER INFORMATION
      CENT     = CENTR / 12.0
      BEAMD    = DBEAM/12.
      FLT      = TFL/12.
      AREA     = AREA1/144.
      BEAMI    = BEAM/(12.**4)
      IFIND    = IFIND+1
    2 CONTINUE
      GOTO 4
C---  READ R.C.GIRDER DATA BASE INFORMATION
    1 DO 3   I = 1,NBEAM
      READ  (NTAPE) IBMTP,JBMTP,COMI1,ARIN,YBAR1,
     C              COMI2,AREN,YBAR2,SLABT,BEAMP
C---  CHECK FOR CORRECT R.C.GIRDER
      IF (IBMTP.NE.INAM) GOTO 3
      IF (JBMTP.NE.JNAM) GOTO 3
      WRITE (NOUT,607) COMI1,ARIN,YBAR1,CCMI2,AREN,YBAR2,SLABT,BEAMP
      IFIND    = IFIND+1
C---  SET VALUES FOR GIRDER INFORMATION
      CONV     = 12.**4
      BEAMD    = BEAMP/12.
      ARINT    = ARIN/144.
      AREND    = AREN/144.
      COMPI(1) = (COMI1 - (GIRDSP * SLABT**3) / (1.0-POIS**2)) / CONV
      COMPI(2) = (COMI2 - ((GIRDSP/2.0 + OVERH1) * SLABT**3) /
     C                     (1.0-POIS**2)) / CONV
      TK       = SLABT / 12.0
      SLABI    = SLABT**3/CONV
      YBAR(1)  = YBAR1/12.0
      YBAR(2)  = YBAR2/12.0
      YB2      = YBAR (2)
      AREN2    = AREND + CUHT * CUWD
C---  CONVERT EXT. SECTION FOR INTEGRAL CURB
                 CJT = 0
      IF (NX.EQ.0) CJT = 1
      YBAR(2)  = ((YBAR(2)*AREND)+(BEAMD+CUHT/2.0)*CUHT*CUWD*CJT)/AREN2
      COMPI(2) = COMPI(2) + ((AREND*(YB2-YBAR(2)))**2) +
     C           (CUHT*CUWD*(CUHT/2.0+BEAMD-YBAR(2))**2)+
     C           (CUHT*(CUWD**3)/12.0))*CJT
      AREND    = AREN2
      BEAMI    = COMPI(2)
      CENT     = YB2
      FLT      = 0.0
      BEAMD    = BEAMD - TK
    3 CONTINUE
C---  PRINT OUT NOTES FOR LOCATION IN DATA BASE
    4 CONTINUE
      IF (IFIND) 5,6,7
    5 WRITE (NOUT,601) INAM,JNAM
      STOP
    6 WRITE (NOUT,602) INAM,JNAM
      IF (IGRTP.NE.IPAN) GOTO 8
C---  PRINT OUT BEAM INFORMATION FOR R.C.GIRDER BRIDGE
      WRITE (NOUT,605) COMPI(1),COMPI(2),TK,BEAMD
      GOTO 999
    7 IFIND    = IFIND+1
      WRITE (NOUT,603) INAM,JNAM,IFIND
      GOTO 999
    8 CONTINUE
```

```
C---  COMPUTE FLANGE WIDTH AND PARAMETERS
      EWIDE3   = GIRDSP
                          EWIDE = FLT+(12.*TK)
      IF (EWIDE3.LT.EWIDE) EWIDE = EWIDE3
      EWID(1)  = EWIDE
                                EWID(2) = EWID(1)-(EWID(1)/2.-OVERH1)
      IF (EWID(2).GT.EWID(1)) EWID(2) = EWID(1)
                         SCIN = 1.0
      IF (IGRTP.EQ.ISTE) SCIN = 1.0 / SC
C---  COMPUTE YBAR AND COMPOSITE MOMENT OF INERTIA - STEEL AND PRESTRESS
      BM10     = SCIN*TK*(BEAMD+HAUNCH+TK/2.0)
      BM11     = AREA * CENT
      BM12     = SCIN*HAUNCH*FLT*(BEAMD+HAUNCH/2.0)
      BM13     = SCIN*CUHT*CUWD*(BEAMD+HAUNCH+TK+CUHT/2.0)
      AB10     = SCIN * TK
      AB12     = SCIN * HAUNCH * FLT
      AB13     = SCIN * CUHT * CUWD
C---  COMPUTE COMPOSITE SECTION PROPERTIES
      DO 10  J = 1,2
                   CJ1 = J - 1
      IF (NX.NE.0) CJ1 = 0.0
      BM10     = BM10 * EWID(J)
      AB10     = AB10 * EWID(J)
      YBAR (J) = (BM10+BM11+BM12+BM13*CJ1)
     C           /(AB10+AREA+AB12+AB13*CJ1)
      COMPI(J) = (AB10*(BEAMD+HAUNCH+TK/2.0-YBAR(J))**2)
     C           +(AREA*(CENT-YBAR(J))**2)+(AB12*(BEAMD+HAUNCH/2.0-
     C           YBAR(J))**2)+(AB13*CJ1*(BEAMD+HAUNCH+TK+CUHT/2.0-
     C           YBAR(J))**2)+BEAMI+CJ1*CUWD*CUHT**3/12.0
      BM10     = BM10 / EWID(J)
      AB10     = AB10 / EWID(J)
   10 CONTINUE
C---  PRINT OUT BEAM INFORMATION FOR PRESTRESSED OR STEEL GIRDER
      WRITE (NOUT,604) BEAMD,CENT,BEAMI,YBAR(1),COMPI(1)
C---
  601 FORMAT (8X,2A4,30H NOT FOUND IN DATA BASE          )
  602 FORMAT (8X,2A4,30H LOCATED FROM DATA BASE          )
  603 FORMAT (8X,2A4 ,22H FOUND IN DATA BASE   I3,7H TIMES )
  604 FORMAT (8X,14HGIRDER - DEPTH 22X,4H(FT) 5X,F10.4,
     C  /8X,45HGIRDER - CENTROID TO BOTTOM           (FT)      ,F10.4,
     C  /8X,45HGIRDER - MOMENT OF INERTIA          (FT-4)      ,F10.4,
     C  /8X,45HCOMPOSITE SECT - CENTROID TO BOTTOM (FT)        ,F10.4,
     C  /8X,45HCOMPOSITE SECTION - I               (FT-4)      ,F10.4)
  605 FORMAT (
     C   8X,45HCOMPOSITE SECTION - I (INTERIOR)   (FT-4)       ,F10.4,
     C  /8X,45HCOMPOSITE SECTION - I (EXTERIOR)   (FT-4)       ,F10.4,
     C  /8X,45HEQUIVALENT SLAB THICKNESS             (FT)       ,F10.4,
     C  /8X,45HEQUIVALENT GIRDER DEPTH               (FT)       ,F10.4)
  606 FORMAT (8X,40HGIRDER - CROSS SECTIONAL AREA     (IN-2) ,5X,F10.2
     C ,/8X,45HGIRDER - MOMENT OF INERTIA         (IN-4)       ,F10.0,
     C  /8X,45HGIRDER - DEPTH                       (IN)       ,F10.2,
     C  /8X,45HGIRDER - TOP FLANGE WIDTH            (IN)       ,F10.2,
     C  /8X,45HGIRDER - CENTROID TO BOTTOM          (IN)       ,F10.2
     C  //11X,30H COMPUTED GIRDER INFORMATION    )
  607 FORMAT (
     C   8X,45HTOTAL SECTION - I      (INTERIOR)  (IN-4)       ,F10.0,
     C  /8X,45HCROSS SECTIONAL AREA  (INTERIOR)  (IN-2)        ,F10.2,
     C  /8X,45HCENTROID TO BOTTOM    (INTERIOR)    (IN)        ,F10.2,
     C  /8X,45HTOTAL SECTION - I     (EXTERIOR)  (IN-4)        ,F10.0,
     C  /8X,45HCROSS SECTIONAL AREA  (EXTERIOR)  (IN-2)        ,F10.2,
     C  /8X,45HCENTROID TO BOTTOM    (EXTERIOR)    (IN)        ,F10.2,
     C  /8X,45HEQUIVALENT SLAB THICKNESS           (IN)        ,F10.2,
     C  /8X,45HTOTAL SECTION DEPTH                 (IN)        ,F10.2
```

```
     C  //11X,30H COMPUTED GIRDER INFORMATION    )
C---
  999 RETURN
      END
      SUBROUTINE BMINP (BEAMI,TK,GIRDSP,AREA,OVERH1,PCIS,
     C                  BEAMD,SC,IGRTP,SLABI,ARINT,AREND,NX)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BM     / EWID(2),YBAR(2),COMPI(2),CENT,CUHT,CUWD,HAUNCH
      DATA              IPRS,IPAN,ISTE /4HPRES,4HREIN,4HSTEE/
C---
C---  THIS ROUTINE READS INPUT GIRDER INFORMATICN
C---  IT ALSO COMPUTES THE COMPOSITE SECTION YBAR AND I
C---
C-----IPRS NOT REFERENCED - SET TO DUMMY TO AVOID DIAGNOSTIC - JJP      090C80
            IDUMY = IPRS                                                090C80
C---  CHECK GIRDER TYPE
      IF (IGRTP.NE.IPAN) GOTO  1
C---  READ R.C.GIRDER INFORMATION
      READ  (NINP,501) COMI1,ARIN,YBAR1,COMI2,AREN,YBAR2,SLABT,BEAMP
C---  ECHO PRINT INPUT DATA
      WRITE (NOUT,601) COMI1,ARIN,YBAR1,COMI2,AREN,YBAR2,
     C                 SLABT,BEAMP
C---  SET VALUES FOR GIRDER INFORMATION
      CONV      = 12**4
      BEAMD     = BEAMP / 12.
      ARINT     = ARIN / 144.
      AREND     = AREN / 144.
      COMPI(1) = (COMI1 - (GIRDSP * SLABT**3) / (1.0-POIS**2)) / CONV
      COMPI(2) = (COMI2 - ((GIRDSP/2.0 + OVERH1) * SLABT**3) /
     C                      (1.0-POIS**2)) / CONV
      TK        = SLABT / 12.0
      SLABI     = SLABT**3/CONV
      YBAR (1) = YBAR1/12.0
      YBAR (2) = YBAR2/12.0
      YB2       = YBAR (2)
C---  CONVERT EXT. SECTION FOR INTEGRAL CURB
                  CJT = 0
      IF (NX.EQ.0) CJT = 1
      AREN2     = AREND + CUHT * CUWD * CJT
      YBAR (2) = ((YBAR(2)*AREND)+(BEAMD+CUHT/2.0)*CUHT*CUWD*CJT)/AREN2
      COMPI(2) = COMPI(2) + ((AREND*(YB2-YBAR(2))**2) +
     C            (CUHT*CUWD*(CUHT/2.0+BEAMD-YBAR(2))**2)+
     C            (CUHT*(CUWD**3)/12.0))*CJT
      AREND     = AREN2
      BEAMI     = COMPI(2)
      CENT      = YB2
      FLT       = 0.0
      BEAMD     = BEAMD - TK
      WRITE (NOUT,603) COMPI(1),COMPI(2),TK,BEAMD
      GOTO 999
    1 CONTINUE
C---  READ GIRDER INFORMATION FOR PRESTRESSED OR STEEL
      READ  (NINP,502) AREA1,BEAM,DBEAM,TFL,CENT
C---  ECHO PRINT INPUT DATA
      WRITE (NOUT,602) AREA1,BEAM,DBEAM,TFL,CENT
C---  SET VALUES FOR GIRDER INFORMATICN
      CENT      = CENT/12.0
      BEAMD     = DBEAM/12.
      FLT       = TFL/12.
      AREA      = AREA1/144.
      BEAMI     = BEAM/(12.**4)
C---  COMPUTE FLANGE WIDTH AND PARAMETERS
      EWIDE3    = GIRDSP
```

```
                          EWIDE = FLT+(12.*TK)
      IF (EWIDE3.LT.EWIDE) EWIDE = EWIDE3
      EWID(1)  = EWIDE
                             EWID(2) = EWID(1)-(EWID(1)/2.-OVERH1)
      IF (EWID(2).GT.EWID(1)) EWID(2) = EWID(1)
                        SCIN = 1.0
      IF (IGRTP.EQ.ISTE) SCIN = 1.0 / SC
C---  COMPUTE YBAR AND COMPOSITE MOMENT OF INERTIA - STEEL AND PRESTRESS
      BM10     = SCIN*TK*(BEAMD+HAUNCH+TK/2.0)
      BM11     = AREA * CENT
      BM12     = SCIN*HAUNCH*FLT*(BEAMD+HAUNCH/2.0)
      BM13     = SCIN*CUHT*CUWD*(BEAMD+HAUNCH+TK+CUHT/2.0)
      AB10     = SCIN * TK
      AB12     = SCIN * HAUNCH * FLT
      AB13     = SCIN * CUHT * CUWD
C---  COMPUTE COMPOSITE SECTION PROPERTIES
      DO 3   J = 1,2
                   CJ1 = J - 1
      IF (NX.NE.0) CJ1 = 0.0
      BM10     = BM10 * EWID(J)
      AB10     = AB10 * EWID(J)
      YBAR (J) = (BM10+BM11+BM12+BM13*CJ1)
     C           /(AB10+AREA+AB12+AB13*CJ1)
      COMPI(J) = (AB10*(BEAMD+HAUNCH+TK/2.0-YBAR(J))**2)
     C           +(AREA*(CENT-YBAR(J))**2)+(AB12*(BEAMD+HAUNCH/2.0-
     C           YBAR(J))**2)+(AB13*CJ1*(BEAMD+HAUNCH+TK+CUHT/2.0-
     C           YBAR(J))**2)+BEAMI+CJ1*CUWD*CUHT**3/12.0
      BM10     = BM10 / EWID(J)
      AB10     = AB10 / EWID(J)
    3 CONTINUE
      WRITE (NOUT,604) BEAMD,CENT,BEAMI,YBAR(1),COMPI(1)
C---
  501 FORMAT ( 8F10.0 )
  502 FORMAT ( 6F10.0 )
  601 FORMAT (
     C   8X,45HTOTAL SECTION - I      (INTERIOR)  (IN-4)      ,F10.0,
     C  /8X,45HCROSS SECTIONAL AREA  (INTERIOR)  (IN-2)      ,F10.2,
     C  /8X,45HCENTROID TO BOTTOM    (INTERIOR)    (IN)      ,F10.2,
     C  /8X,45HTOTAL SECTION - I     (EXTERIOR)  (IN-4)      ,F10.0,
     C  /8X,45HCROSS SECTIONAL AREA  (EXTERIOR)  (IN-2)      ,F10.2,
     C  /8X,45HCENTROID TO BOTTOM    (EXTERIOR)    (IN)      ,F10.2,
     C  /8X,45HEQUIVALENT SLAB THICKNESS               (IN)      ,F10.2,
     C  /8X,45HTOTAL SECTION DEPTH                     (IN)      ,F10.2
     C  //11X,30H COMPUTED GIRDER INFORMATION    )
  602 FORMAT (8X,40HGIRDER - CROSS SECTIONAL AREA     (IN-2) ,5X,F10.2,
     C  /8X,45HGIRDER - MOMENT OF INERTIA          (IN-4)      ,F10.0,
     C  /8X,45HGIRDER - DEPTH                        (IN)      ,F10.2,
     C  /8X,45HGIRDER - TOP FLANGE WIDTH             (IN)      ,F10.2,
     C  /8X,45HGIRDER - CENTROID TO BOTTOM           (IN)      ,F10.2
     C  //11X,30H COMPUTED GIRDER INFORMATION    )
  603 FORMAT (
     C   8X,45HCOMPOSITE SECTION - I (INTERIOR)  (FT-4)      ,F10.4,
     C  /8X,45HCOMPOSITE SECTION - I (EXTERIOR)  (FT-4)      ,F10.4,
     C  /8X,45HEQUIVALENT SLAB THICKNESS            (FT)      ,F10.4,
     C  /8X,45HEQUIVALENT GIRDER DEPTH              (FT)      ,F10.4)
  604 FORMAT (8X,14HGIRDER - DEPTH 22X,4H(FT) 5X,F10.4,
     C  /8X,45HGIRDER - CENTROID TO BOTTOM          (FT)      ,F10.4,
     C  /8X,45HGIRDER - MOMENT OF INERTIA         (FT-4)      ,F10.4,
     C  /8X,45HCOMPOSITE SECT - CENTROID TO BOTTOM (FT)      ,F10.4,
     C  /8X,45HCOMPOSITE SECTION - I              (FT-4)      ,F10.4)
C---
  999 RETURN
      END
```

```
      SUBROUTINE TAB103 ( IDIV,JDIV,EC,PCIS,TK,IOPT,IGRTP,SLABI,
     C                    HX,BRIDL,HY,DX)
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DATA             IPAN/4HREIN/
C---
C---  THIS SUBROUTINE CALCULATES THE SLAB BENDING
C---  STIFFNESS IN THE X AND Y DIRECTIONS.
C---
      DX       = 0.0
      DY       = 0.0
      IF (IOPT.EQ.1) GOTO 2
C---  COMPUTE SLAB STIFFNESS FOR R.C.GIRDER BRIDGE
      IF (IGRTP.NE.IPAN) GOTO 1
      DX       = SLABI*EC/((1.0-POIS**2)*2.0)
      GOTO 2
C---  COMPUTE SLAB STIFFNESS FOR FLAT SLAB
    1 DX       = ((EC*TK**3)/(12.*(1.-POIS**2)))/2.
    2 CONTINUE
      DY       = DX
C---  SET DO LOOP CONTROL
      ISB      = IDIV + 1
      DO 5  I =  1,ISB
C---  COMPUTE LIMITS FOR SLAB STIFFNESS
                   DIS = FLOAT(I-1)*HX*ANGLE
      IF (ITHETA) DIS = (ARM-FLOAT(I-1)*HX)*ANGLE
      LOC1     = DIS/HY
                          LOC2 = (DIS+BRIDL)/HY+1.
      IF (LOC2.GT.JDIV) LOC2 = JDIV
      DXX      = DX
      DYY      = DY
C---  SET STIFFNESS FOR EDGES OF BRIDGE
      IF (I.NE.1.AND.I.NE.ISB) GOTO 3
      DXX      = DX / 2.0
      DYY      = DY / 2.0
    3 CONTINUE
C---  WRITE TABLE 3 -SLAB STIFFNESS- ON TAPE
      IF (ITURN) GOTO 4
      K        = I-1
      WRITE (NT3,301) K,LOC1,K,LOC2,DXX,DYY
      LOC1     = LOC1+1
      LOC2     = LOC2-1
      WRITE (NT3,301) K,LOC1,K,LOC2,DXX,DYY
      GOTO 5
    4 K        = I-1
      WRITE (NT3,301) LOC1,K,LOC2,K,DYY,DXX
      LOC1     = LOC1+1
      LOC2     = LOC2-1
      WRITE (NT3,301) LOC1,K,LOC2,K,DYY,DXX
    5 CONTINUE
C---
C 301 FORMAT (4I5,2E10.3)                                               CDC
  301 FORMAT (4I5,1P2E10.3)                                             IBM
C---
      RETURN
      END
      SUBROUTINE TAB203 (OVERH1,JDIV,GIRDSP,ES,EC,AREA,NGIRDR,BEAMI,
     C                   HY,CONCW,TK,IGRTP,ARINT,AREND,HX,BRIDL)
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /C      / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                 JDIST1(12,50),JDIST2(12,50)
```

```
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BM     / EWID(2),YBAR(2),CCMPI(2),CENT,CUHT,CUWD,HAUNCH
      COMMON /G      / LOCS(200,4),SPAXIS(200)
      COMMON /DLOADS / DLOAD(50),IDLOC(50,2)
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DATA             IPRS/4HPRES/,ISTE/4HSTEE/,NOBM/4HSLAB/,
     C                 IPAN/4HREIN/
C---
C---  THIS SUBROUTINE CALCULATES BEAM STIFFNESSES
C---  AND ASSIGNS THE APPROPRIATE VALUES TO
C---  THE GRID SYSTEM. THIS ROUTINE ALSO CALCULATES THE
C---  DEAD LOADS TO BE APPLIED IN -DEADLC-
C---
                        EBM = EC
      IF (IGRTP.EQ.ISTE) EBM = ES
                        PCF = 0.490
      IF (IGRTP.EQ.IPRS) PCF = 0.0
      IF (IGRTP.EQ.NOBM) GOTO 999
C---  COMPUTE GIRDER STIFFNESS FOR STEEL AND PRESTRESSED GIRDERS
      FY             = BEAMI * EBM / 2.0
      DO 6     J1 = 1,NGIRDR
      IF (IGRTP.NE.IPAN) GOTO 2
C---  FIND THE PROPER VALUE FOR R.C.GIRDER STIFFNESS
      FY             = COMPI(2) * EC / 2.0
      AREA           = AREND
      IF ((J1.EQ.1).OR.(J1.EQ.NGIRDR)) GOTO 1
      AREA           = ARINT
      FY             = COMPI(1) * EC / 2.0
C---  SET DEAD WEIGHT FOR R.C.GIRDER
    1 DEAD1          = AREA * CONCW * HY * (-.5)
      GOTO 3
C---  SET DEAD WEIGHT FOR STEEL AND PRESTRESSED GIRDERS
    2                                 WIDTH = GIRDSP
      IF ((J1.EQ.1).OR.(J1.EQ.NGIRDR)) WIDTH = GIRDSP/2.+CVERH1
      DEAD1          = ((PCF*AREA*HY)+(CONCW*TK*WIDTH*HY))*(-.5)
    3 DLOAD(J1)      = DEAD1
C---  FIND BEGINNING AND ENDING LOCATIONS FOR STIFFNESS AND DEAD WT.
                 DIS = FLOAT(IGIRDL(J1))*HX*ANGLE
      IF (ITHETA) DIS = (ARM-FLOAT(IGIRDL(J1))*HX)*ANGLE
      LOC1           = DIS/HY
                       LOC2 = (DIS+BRIDL)/HY+1.
      IF (LOC2.GT.JDIV) LOC2 = JDIV
      IDLOC(J1,1) = LOC1
      IDLOC(J1,2) = LOC2
      DO 6     I1 = 1,2
C---  WRITE BEAM STIFFNESS ONTO TAPE
      IF (ITURN) GOTO 4
      WRITE (NT3,301) IGIRDL(J1),LOC1,IGIRDL(J1),LOC2,FY
      GOTO 5
    4 WRITE (NT3,302) LOC1,IGIRDL(J1),LOC2,IGIRDL(J1),FY
    5 LOC1           = LOC1 + 1
      LOC2           = LOC2 - 1
    6 CONTINUE
C---
C 301 FORMAT ( 4I5 , 30X , E10.3 )                                      CDC
  301 FORMAT ( 4I5 , 30X , 1PE10.3 )                                    IBM
C 302 FORMAT ( 4I5 , 20X , E10.3 )                                      CDC
  302 FORMAT ( 4I5 , 20X , 1PE10.3 )                                    IBM
C---
  999 RETURN
      END
      SUBROUTINE TABCUR (IDIV,JDIV,CURBA,CURBI,NX,EC,CCNCW,DECURB,
```

```
     C                          IGRTP,HX,HY,BRIDL,NT3)
      COMMON /SKEW    / THEA,THETA,ANGLE,ARM
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
      DATA              NOBM/4HSLAB/
C---
C---  THIS ROUTINE COMPUTES THE INTEGRAL CURB DEAD WT. AND STIFFNESS
C---
      DECURB  = 0.0
      IF (CURBA.LT.0.001) GOTO 999
      FYCURB  = 0.0
      IF (IGRTP.EQ.NOBM) GOTO 1
      IF (NX.EQ.0) GOTO 2
C---  COMPUTE CURB STIFFNESS AND DEAD WEIGHT
    1 FYCURB  = CURBI * EC * 0.5
    2 DECURB  = CURBA * CONCW * HY * (-0.5)
      II      = 0
      DO 3 I1 = 1,2
C---  FIND STARTING AND ENDING LOCATIONS
                   DIS = FLOAT(II) * HX * ANGLE
      IF (ITHETA) DIS = ARM * ANGLE - DIS
      LOC1    = DIS / HY
                         LOC2 = (DIS+BRIDL)/HY + 1.0
      IF (LOC2.GT.JDIV) LOC2 = LOC2 - 1
      DO 4 J1 = 1,2
      IF (ITURN) GOTO 6
C---  WRITE TABLE 3 -CURB STIFFNESS- ON TAPE
      WRITE (NT3,301) II,LOC1,II,LOC2,FYCURB
      GOTO 5
    6 WRITE (NT3,302) LOC1,II,LOC2,II,FYCURB
    5 LOC2    = LOC2 - 1
      LOC1    = LOC1 + 1
    4 CONTINUE
    3 II      = IDIV
C---
C 301 FORMAT ( 4I5 , 30X , E10.3 )                                      CDC
  301 FORMAT ( 4I5 , 30X , 1PE10.3 )                                    IBM
C 302 FORMAT ( 4I5 , 20X , E10.3 )                                      CDC
  302 FORMAT ( 4I5 , 20X , 1PE10.3 )                                    IBM
C---
  999 RETURN
      END
      SUBROUTINE TAB303 (NSPAN,JDIV,HY,ID,SPNGK,IGRTP,HX)
      COMMON /SKEW    / THEA,THETA,ANGLE,ARM
      COMMON /G       / LOCS(200,4),SPAXIS(200)
      COMMON /C       / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                  JDIST1(12,50),JDIST2(12,50)
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
      DATA      NOBM/4HSLAB/
C---
C---  THIS SUBROUTINE LOCATES SUPPORTED POINTS
C---  AND ASSIGNS THEM VALUES.
C---
      JJ          = 0
      IEND        = NSPAN + 1
                   SPAN = 0.0
      DO 1      I = 1,IEND
      IF (I.NE.1) SPAN = SPAN + SPANL(I-1)
      DO 1      J = 1,ID
      JJ          = JJ + 1
C---  IDENTIFY LOCATIONS ACROSS THE BRIDGE
```

```
                        ICROSS = IGIRDL (J)
      IF (IGRTP.EQ.NOBM) ICROSS = J - 1
C---  IDENTIFY SPRING LOCATIONS
                    DIS = FLOAT(ICROSS)*HX*ANGLE
      IF (ITHETA) DIS = (ARM-FLOAT(ICROSS)*HX)*ANGLE
      DIS          = DIS+SPAN
      LOC1         = DIS/HY
      A            = DIS-FLOAT(LOC1)*HY
      LOC2         = DIS/HY+1.
      B            = FLOAT(LOC2)*HY-DIS
      IF (LOC2.GT.JDIV) LOC2 = LOC2-1
C---  CALC SPRING AND AXIAL FORCES
      SPAXIS(JJ) = (-SPNGK)*A*B/HY
      SPR1         = B*SPNGK/HY
      SPR2         = A*SPNGK/HY
      IF (ITURN) GOTO 20
C---  WRITE TABLE 3 -SPRING SUPPORTS- ON TAPE
      WRITE (NT3,301) ICROSS,LOC1,ICROSS,LOC1,SPR1
      WRITE (NT3,301) ICROSS,LOC2,ICROSS,LOC2,SPR2
      LOCS(JJ,1) = ICROSS
      LOCS(JJ,2) = LOC2
      LOCS(JJ,3) = ICROSS
      LOCS(JJ,4) = LOC2
      GOTO 1
   20 WRITE (NT3,301) LOC1,ICROSS,LOC1,ICROSS,SPR1
      WRITE (NT3,301) LOC2,ICROSS,LOC2,ICROSS,SPR2
      LOCS(JJ,1) = LOC2
      LOCS(JJ,2) = ICROSS
      LOCS(JJ,3) = LOC2
      LOCS(JJ,4) = ICROSS
    1 CONTINUE
C---
C 301 FORMAT (4I5,50X,E10.3)                                              CDC
  301 FORMAT (4I5,50X,1PE10.3)                                            IBM
C---
      RETURN
      END
      SUBROUTINE TAB403 (BEAMI,NCOMP,ES,EC,HY,NGIRDR,IGRTP,
     C                  HX,JDIV)
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /C      / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                 JDIST1(12,50),JDIST2(12,50)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BM     / EWID(2),YBAR(2),COMPI(2),CENT,CUHT,CUWD,HAUNCH
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DATA             IPRS/4HPRES/
C---
C---  THIS SUBROUTINE CALCULATES THE ADDITIONAL COMPOSITE
C---  STIFFNESS
C---
      IF (NCOMP.EQ.0) GOTO 999
C---  FIND MODULUS OF ELASTICITY FOR GIRDERS
                        EBM = ES
      IF (IGRTP.EQ.IPRS) EBM = EC
      DO 1      J = 1,NCOMP
C---  SET LIMITS FOR ADDITIONAL COMPOSITE STIFFNESS
                    Y1 = YDIST1(J)/HY+.5
      IF (J.EQ.1)   Y1 = YDIST1(J)/HY
                    Y2 = YDIST2(J) / HY + 0.5
      IF (J.EQ.NCOMP) Y2 = YDIST2(J) / HY + 1.0
      J1DIST      = Y1
      J2DIST      = Y2
```

```
      DO 1        I = 1,NGIRDR
C---  CALC ADDITIONAL STIFFNESS (ONE-HALF)
                                        COMP = COMPI(1)
      IF ((I.EQ.1).OR.(I.EQ.NGIRDR)) COMP = COMPI(2)
      FYADD         = (COMP-BEAMI) * EBM / 2.0
C---  CORRECT STIFFNESS LIMITS FOR SKEWED BRIDGE
                    DIS = FLOAT(IGIRDL(I))*HX*ANGLE
      IF (ITHETA) DIS = (ARM-FLOAT(IGIRDL(I))*HX)*ANGLE
      JDIST1(J,I) = (FLOAT(J1DIST)*HY+DIS)/HY+.5
                          JDIST2(J,I) = (FLOAT(J2DIST)*HY+DIS)/HY+.5
                          JDT2        = JDIST2(J,I)
      IF (JDT2.GT.JDIV) JDIST2(J,I) = JDIV
      J1            = JDIST1(J,I) + 1
      J2            = JDIST2(J,I) - 1
C---  WRITE TABLE 3 -ADDITIONAL GIRDER STIFFNESS- ON TAPE
      IF (ITURN) GOTO 2
      WRITE (NT3,301) IGIRDL(I),JDIST1(J,I),IGIRDL(I),JDIST2(J,I),FYADD
      WRITE (NT3,301) IGIRDL(I),J1,IGIRDL(I),J2,FYADD
      GOTO 1
    2 WRITE (NT3,302) JDIST1(J,I),IGIRDL(I),JDIST2(J,I),IGIRCL(I),FYADD
      WRITE (NT3,302) J1,IGIRDL(I),J2,IGIRDL(I),FYADD
    1 CONTINUE
C---
C 301 FORMAT (4I5,30X,E10.3)                                                CDC
  301 FORMAT (4I5,30X,1PE10.3)                                              IBM
C 302 FORMAT (4I5,20X,E10.3)                                                CDC
  302 FORMAT (4I5,20X,1PE10.3)                                              IBM
C---
  999 RETURN
      END
      SUBROUTINE TAB503 (NDIA,DIAST,DIASP,STDIA,NGIRDR,GIRDSP,JDIV,
     C                    NGRDIV,HX,HY,BRIDL,IDIV,NX,NASHOK)
      COMMON /SKEW    / THEA,THETA,ANGLE,ARM
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
C---
C---  THIS ROUTINE PLACES THE DIAPHRAGM STIFFNESS CN THE
C---  GRID SYSTEM. ONE AT EACH END AND NDIA DIAPHRAGMS
C---  SPACED EQUALLY ALONG THE STRUCTURE.
C---
      IF (NDIA.EQ.0) GOTO 999
C---  CALC DIAFRAM STIFFNESS (ONE-HALF)
      STDIA    = STDIA / (2.0*1000.0*144.0)
C---  CALC DIAFRAM SPACING
      DIAF     = NDIA - 1
      DIASP    = (DIASP-DIAST) / DIAF
C---  CALC NUMBER OF DIAPHRAGMS ACROSS BRIDGE
                          NACRS = NGIRDR - 1
      IF (NASHOK.EQ.1) NACRS = NACRS + 2
                  STEPD = 0.0
C---  CALC GIRDER END DISPLACEMENTS FOR SKEWED BRIDGE
      IF (ABS(ANGLE).LT.0.001) GOTO 1
                  STEPD = GIRDSP * ANGLE
      IF (ITHETA) STEPD = -STEPD
    1 CONTINUE
                          LL = NX
      IF (NASHOK.EQ.1) LL = 0
      DO 2 LI = 1,NACRS
C---  CALC OFFSET FOR SKEWED CASE
                                        LL2 = LL + NGRDIV
      IF (NASHOK.EQ.1.AND.LI.EQ.1) LL2 = LL + NX
      IF (LL2.GT.IDIV)                  LL2 = IDIV
```

```
      LA      = LL + 1
      LB      = LL2 - 1
      IF (LB.LT.LA) LB=LA
                    DIS = FLOAT(LL) * HX * ANGLE
      IF (ITHETA) DIS = (ARM - FLOAT(LL2) * HX) * ANGLE
                         ISD = DIS/HY + 0.5
      IF (ITURN) GOTO 5
C---  WRITE TABLE 3 -DIAPHRAGM STIFFNESS- ON TAPE
      WRITE (NT3,301) LL,ISD,LL2,ISD,STDIA
      WRITE (NT3,301) LA,ISD,LB,ISD,STDIA
      DIS1    = DIS + DIAST
      DO 3  I = 1,NDIA
                         ISD = DIS1/HY + 0.5
      WRITE (NT3,301) LL,ISD,LL2,ISD,STDIA
      WRITE (NT3,301) LA,ISD,LB,ISD,STDIA
    3 DIS1    = DIS1 + DIASP
                         ISD = (DIS + BRIDL + STEPD) / HY +0.5
      IF (ISD.GT.JDIV) ISD = JDIV
      WRITE (NT3,301) LL,ISD,LL2,ISD,STDIA
      WRITE (NT3,301) LA,ISD,LB,ISD,STDIA
      GOTO 2
C---  WRITE TABLE 3 -DIAPHRAGM STIFFNESS- FOR TURN = -1
    5 WRITE (NT3,302) ISD,LL,ISD,LL2,STDIA
      WRITE (NT3,302) ISD,LA,ISD,LB,STDIA
      DIS1    = DIS + DIAST
      DO 4  I = 1,NDIA
                         ISD = DIS1/HY + 0.5
      WRITE (NT3,302) ISD,LL,ISD,LL2,STDIA
      WRITE (NT3,302) ISD,LA,ISD,LB,STDIA
    4 DIS1    = DIS1 + DIASP
                         ISD = (DIS + BRIDL + STEPD) / HY +0.5
      IF (ISD.GT.JDIV) ISD = JDIV
      WRITE (NT3,302) ISD,LL,ISD,LL2,STDIA
      WRITE (NT3,302) ISD,LA,ISD,LB,STDIA
    2 LL      = LL2
C---
C 301 FORMAT (4I5,20X,E10.3)                                            CDC
  301 FORMAT (4I5,20X,1PE10.3)                                          IBM
C 302 FORMAT (4I5,30X,E10.3)                                            CDC
  302 FORMAT (4I5,30X,1PE10.3)                                          IBM
C---
  999 RETURN
      END
      SUBROUTINE TABLE5 (EC,TK,POIS,IDIV,JDIV,IOPT,IGRTP,SLABI,
     C                   BRIDL,HX,HY)
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DATA             IPAN/4HREIN/
C---
C---  THIS SUBROUTINE CALCULATES THE TWISTING STIFFNESS
C---  OF THE SLAB PORTION OF THE BRIDGE.
C---
                       TK1 = 0
      IF (IOPT.EQ.0) TK1 = TK
C---  COMPUTE TWISTING STIFFNESS
                                       CTWIST = (EC*TK1**3)/(12.*
     C                                                 (1.0+POIS))
      IF (IGRTP.EQ.IPAN.AND.IOPT.EQ.0) CTWIST = SLABI*EC/(1.0+POIS)
      DO 2  I = 1,IDIV
C---  COMPUTE SKEW OFFSET
                    DIS = FLOAT(I)*HX*ANGLE
```

```
      IF (ITHETA) DIS = (ARM-FLOAT(I)*HX)*ANGLE
C---  FIND STARTING AND ENDING LOCATION
      LOC1    = DIS/HY+1.0
                          LOC2 = (DIS+BRIDL)/HY+1.0
      IF (LOC2.GT.JDIV) LOC2 = LOC2 - 1
      IF (ITURN) GOTO 1
C---  WRITE TABLE 5 -TWISTING STIFFNESS- ON TAPE
      WRITE (NT3,301) I,LOC1,I,LOC2,CTWIST
      GOTO 2
    1 WRITE (NT3,301) LOC1,I,LOC2,I,CTWIST
    2 CONTINUE
C---
C 301 FORMAT (4I5,E10.3)                                                 CDC
  301 FORMAT (4I5,1PE10.3)                                               IBM
C---
      RETURN
      END
      SUBROUTINE TABLE6 (NUMB6)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /G      / LOCS(200,4),SPAXIS(200)
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
C---
C---  THIS SUBROUTINE WRITE THE AXIAL FORCES FOR SUPPORTS ON TAPE
C---
      IF (ITURN) GOTO 4
      DO 1 I = 1,NUMB6
    1 WRITE (NT3,301) (LOCS(I,J),J=1,4),SPAXIS(I)
      GOTO 3
    4 DO 2 I = 1,NUMB6
    2 WRITE (NT3,302) (LOCS(I,J),J=1,4),SPAXIS(I)
    3 CONTINUE
C---
C 301 FORMAT (4I5,50X,E10.3)                                             CDC
  301 FORMAT (4I5,50X,1PE10.3)                                           IBM
C 302 FORMAT (4I5,40X,E10.3)                                             CDC
  302 FORMAT (4I5,40X,1PE10.3)                                           IBM
C---
      RETURN
      END
      SUBROUTINE FIX789 (N,NGIRDR,IDEED,ISPEL,NUMB8,NUMB9,
     C                   IA,JA,XA,IB,JB,IC,JC,ID,JD,IGIRDL,IGRTP,
     C                   IDIV,JDIV,CONCW,HX,HY,TK,BRICL,DECURB)
      DIMENSION          IA(200),JA(200),XA(200),IB(20),JB(20),IC(20),
     C                   JC(20),ID(20),JD(20),IGIRDL(50)
      COMMON /D      / IPONT(80),IDEFL(20),IXMOM(20),IYMOM(20),ISTRS(20)
     C                 ,IDUD(2)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
C---
C---  THIS ROUTINE WRITES TABLE 7,8,AND 9 ON TAPE
C---
      CALL DEADLO (NGIRDR,IDEED,IGIRDL,IDIV,JDIV,DECURB,
     C             CONCW,HX,HY,TK,BRIDL,IGRTP)
C---  WRITE TABLE 7 -TRUCK LOADS- TO TAPE
      IF (N.EQ.0) GOTO 2
      DO 1 I = 1,N
      AXLOAD = ABS ( XA(I) )
      IF (AXLOAD.LT.0.001) GOTO 1
C---  RESET SIGN OF LOAD FOR UT - HWYLIB VERISON OF SLAB49
C     XA(I)    = - XA(I)                                                 CDC
      XA(I)    =   XA(I)                                                 IBM
      WRITE (NT3,301) IA(I),JA(I),IA(I),JA(I),XA(I)
    1 CONTINUE
```

```
    2 CONTINUE
C---  WRITE TABLE 8 ON TAPE
      IF (ISPEL.EQ.0) GOTO 999
      IF (NUMB8.EQ.0) GOTO 4
      DO 3 I = 1,NUMB8
      WRITE (NT3,302) IB(I),JB(I),IC(I),JC(I),IDEFL(I)
     C                ,IXMOM(I),IYMOM(I),ISTRS(I)
    3 CONTINUE
    4 CONTINUE
C---  WRITE TABLE 9 ON TAPE
      IF (NUMB9.EQ.0) GOTO 6
      DO 5 I = 1,NUMB9
      WRITE (NT3,303) ID(I),JD(I)
    5 CONTINUE
    6 CONTINUE
C---
C 301 FORMAT ( 4I5 , 40X , E10.3 )                                CDC
  301 FORMAT ( 4I5 , 40X , 1PE10.3 )                              IBM
  302 FORMAT ( 8I5 )
  303 FORMAT ( 2 ( 5X,I5 ) )
C---
  999 RETURN
      END
      SUBROUTINE DEADLO (NGIRDR,IDEED,IGIRDL,IDIV,JDIV,DECURB,
     C                   CONCW,HX,HY,TK,BRIDL,IGRTP)
      COMMON /DLOADS / DLOAD(50),IDLOC(50,2)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DIMENSION        IGIRDL(50)
      DATA             NOBM /4HSLAB/
C---
C---  THIS ROUTINE APPLIES THE DEAD LOAD CALCULATED IN -TAB203-
C---  FOR SLAB STRUCTURES THE DEAD WT. IS COMPUTED BELOW.
C---  CURB DEAD WT. IS ADDED IN AT THIS POINT
C---
      IF (IDEED.EQ.0) GOTO 999
      IF (IGRTP.EQ.NOBM) GOTO 4
      DO 1 I = 1,NGIRDR
C---  ADD CURB DEAD WT. TO EXTERIOR GIRDER LOAD
                                DLOD = DLOAD(I)
      IF (I.EQ.1.OR.I.EQ.NGIRDR) DLOD = DLOAD(I) + DECURB
C---  RESET SIGN OF LOAD FOR UT - HWYLIB VERSION OF SLAB49
C                               DLOD = - DLOD                     CDC
                                DLOD =   DLOD                     IBM
C---  SET LIMITS FOR GIRDER/SLAB STRUCTURE DEAD WEIGHT
      LOC1   = IDLOC(I,1)
      LOC2   = IDLOC(I,2)
      LGDR   = IGIRDL(I)
C---  WRITE DEAD WEIGHT FOR GIRDER/SLAB STRUCTURE
      DO 1 J = 1,2
      IF (ITURN) GOTO 2
      WRITE (NT3,301) LGDR,LOC1,LGDR,LOC2,DLOD
      GOTO 3
    2 WRITE (NT3,301) LOC1,LGDR,LOC2,LGDR,DLOD
    3 LOC1   = LOC1 + 1
      LOC2   = LOC2 - 1
    1 CONTINUE
      GOTO 999
C---  DEAD WEIGHT CALC FOR SLAB STRUCTURES
    4 ISB    = IDIV + 1
C---  CALC DEAD WEIGHT
```

```
      DWT     = CONCW * HY * HX * TK * (-0.5)
      DO 7 I = 1,ISB
      SD      = I - 1
      K       = I - 1
C---  FIND LIMITS OF DEAD WEIGHT
                              DIS  = SD * HX * ANGLE
      IF (ITHETA)             DIS  = (ARM - SD * HX) * ANGLE
      LOC1   = DIS / HY
                              LOC2 = ((DIS + BRIDL) / HY) + 1.0
      IF (LOC2.GT.JDIV)       LOC2 = JDIV
                              DWT1 = DWT
      IF (I.EQ.1.OR.I.EQ.ISB) DWT1 = DWT / 2.0 + DECURB
C---  RESET SIGN OF LOAD FOR UT - HWYLIB VERSION OF SLAB49
C                             DWT1 = - DWT1                                CDC
                              DWT1 =   DWT1                                IBM
C---  WRITE SLAB STRUCTURE DEAD WIGHT TO TAPE
      DO 7 J = 1,2
      IF (ITURN) GOTO 5
      WRITE (NT3,301) K,LOC1,K,LOC2,DWT1
      GOTO 6
    5 WRITE (NT3,301) LOC1,K,LOC2,K,DWT1
    6 LOC1     = LOC1 + 1
      LOC2     = LOC2 - 1
    7 CONTINUE
C---
C 301 FORMAT ( 4I5 , 40X , E10.3 )                                         CDC
  301 FORMAT ( 4I5 , 40X , 1PE10.3 )                                       IBM
C---
  999 RETURN
      END
      SUBROUTINE TABLE7 (IDIV,JDIV,HX,HY,IGRTP,BRIDL,IFCR)
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /LOAD   / P(75),PP(75),XR(75),YR(75),X(75),Y(75)
      COMMON /A      / ILINE,JLINE,XLINE,YLINE,IPTN(300),JPTN(300),
     C                 XLOAD(300),N,IDUMC
      COMMON /B      / SCLX,SCLY,YT,IDIR,NGIRDR,NGRDIV,NX,CVERH,OVERH1,
     C                 NSPAN,IDUM1
      COMMON /C      / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                 JDIST1(12,50),JDIST2(12,50)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BUFFR  / IBUF(1000)
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DATA             ILB,IRF/4HBACK,4HFORW/
C---
C---  THIS SUBROUTINE AND THE ONES IT CALLS CALCULATES
C---  THE LOADING THAT THE TRUCKS WILL APPLY TO THE
C---  THE BRIDGE.IT IS ALSO CAPABLE OF PRODUCING A PLOT
C---  OF THE TRUCKS AND SPANS.
C---
      N       = 0
      REWIND NT9
C---  READ NUMBER OF TRUCKS IN DATA BASE
      READ  (NT9) NLOAD
      REWIND NT9
      IG      = 0
C---  READ TRUCK LOADINGS INFO
      READ  (NINP,501) NTRUCK,IDEGRE,IPLOT
      WRITE (NOUT,601) NTRUCK,IDEGRE,IPLOT,NLOAD
      IF (IPLOT.NE.1) GOTO 1
C---  CALL ROUTINE TO PLOT GRID FOR BRIDGE
      CALL PLTSPN (IDIV,JDIV,HX,HY,IGRTP)
    1 CONTINUE
```

```
      IF (NTRUCK.EQ.0) GOTO 999
      DO 3 I = 1,NTRUCK
C---  READ TRUCK INPUT INFO
      READ  (NINP,502) ITYPE, JTYPE, DLF, XPOS, YPOS, IDIR
C---  PRINT TRUCK INFO
      WRITE (NOUT,604) ITYPE,JTYPE,DLF,XPOS,YPOS
C---  CORRECT WHEEL PLACEMENT TO NEGATIVE EDGE OF BRIDGE
      XPOS = XPOS + FLOAT(IDIV)*HX/2.
C---  CORRECT LOCATION FOR SKEW
                   DIS = XPOS*ANGLE
      IF (ITHETA) DIS = (ARM-XPOS)*ANGLE
      YYPOS  = YPOS+DIS
      YPOS   = YYPOS
C---  PRINT HEADING FOR TRUCK LOADING OUTPUT
      WRITE (NOUT,603) I
C---  SET VARIABLE FOR TRUCK DIRECTION
                        DIR =  0.
      IF (IDIR.EQ.ILB) DIR = -1.0
      IF (IDIR.EQ.IRF) DIR = 1.0
C---  PRINT TRUCK DIRECTION
      IF (DIR) 6,9,7
    9 WRITE (NOUT,605)
      GOTO 8
    6 WRITE (NOUT,606)
      GOTO 8
    7 WRITE (NOUT,607)
    8 YPOS   = YYPOS
C---  CALL ROUTINE TO LOCATE TRUCK DATA
      CALL TAB107 (ITYPE,JTYPE,XPOS,YPOS,DIR,NWHEL,IPLCT,IDIV,HX,
     C             NOB,I,JDIV,HY,BRIDL,IFOR)
      IF (NOB.GT.0) GOTO 3
      DO 2 J = 1,NWHEL
C---  CALL ROUTINE TO CALC GRID LOADING
      CALL TAB207 (ILINE, XLINE, X(J), IDIV,HX)
      CALL TAB207 (JLINE, YLINE, Y(J), JDIV, HY)
      CALL TAB307 ( P(J), X(J), Y(J), HX, HY, J, IDEGRE,DLF)
    2 CONTINUE
    3 CONTINUE
      IF (IPLOT.NE.1) GOTO 999
C---  CALL TO END OF PLOT
C     CALL ENDPLT                                                       CDC
      CALL PLOTS (15.0,0,0.0,999)                                       IBM
C---
  501 FORMAT (3I5)
  502 FORMAT (2A4,2X,3F10.2,1X,A4)
  601 FORMAT (//5X,26HTABLE 3. TRUCK INFORMATION    ,/,
     C  /8X,45HNUMBER OF TRUCKS FOR LOAD CASE                   ,I10,
     C  /8X,45HOUTPUT GRID LOADING (1=YES)                      ,I10,
     C  /8X,45HCALCOMP PLOT OF LOAD CASE (1=YES)                ,I10,
     C  /8X,45HNUMBER OF TRUCKS IN DATA BASE                    ,I10)
  603 FORMAT ( /7X,    6HTRUCK I2,26H DESCRIPTION AND PLACEMENT    )
  604 FORMAT (8X,10HTRUCK NAME 36X,2A4,
     C  /8X,45HLOAD IMPACT FACTOR                              ,F10.2,
     C  /8X,45HTRUCK LOCATION - TRANSVERSE          (FT)       ,F10.2,
     C  /8X,45HTRUCK LOCATION - LONGITUDINAL        (FT)       ,F10.2)
  605 FORMAT (8X,15HTRUCK DIRECTION  28X,12HNO DIRECTION  )
  606 FORMAT (8X,15HTRUCK DIRECTION  31X,8HBACKWARD       )
  607 FORMAT (8X,15HTRUCK DIRECTION  31X,7HFORWARD        )
C---
  999 RETURN
      END
      SUBROUTINE TAB107 (ITYPE,JTYPE,XPOS,YPOS,DIR,NWHEL,IPLOT,IDIV,HX,
     C                   NOB,IG,JDIV,HY,BRIDL,IFOR)
```

```
      COMMON /SKEW    / THEA,THETA,ANGLE,ARM
      COMMON /LOAD    / P(75),PP(75),XR(75),YR(75),X(75),Y(75)
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      DIMENSION        DP(75), XP(75), YP(75)
      LOGICAL            ITHETA,ITURN
      DATA               INPU/4HINPU/
C---
C---  THIS SUBROUTINE RETRIEVES INFORMATICN CCNCERNING
C---  THE INDIVIDUAL TRUCKS THAT ARE LOCATED CN THE BRIDGE
C---  TO MAKE UP THE LOADING.IT ALSO CHECKS TO MAKE SURE
C---  THAT THE SPECIFIED LOCATIONS DO NOT RESULT IN ANY
C---  WHEELS BEING OFF THE SIDE OF THE BRIDGE.
C---
C---  CALL ROUTINE FOR INPUT TRUCK
      IF (ITYPE.EQ.INPU) CALL LOADUP (ITYPE,JTYPE,XPCS,YPCS,DIR,NWHEL,
     C                                IPLOT,IDIV,JDIV,HX,HY,NOB,IG,
     C                                BRIDL,IFCR)
      IF (ITYPE.EQ.INPU) GOTO 999
      REWIND NT9
      READ  (NT9) NLOAD
      IFIND  = -1
C---  READ TRUCK INFO FROM DATA BASE
      DO 2 I = 1, NLOAD
      READ  (NT9) IBASE, JBASE, NWHEL, NBCK
C---  READ WHEEL INFO FROM DATA BASE
      DO 7 L = 1,NWHEL
    7 READ  (NT9) PP(L),XR(L),YR(L)
C---  CHECK TRUCK NAME FROM DATA BASE
      IF (IBASE.NE. ITYPE) GOTO 2
      IF (JBASE.NE. JTYPE) GOTO 2
      NOB    = 0
      DIV    = IDIV
      DJV    = JDIV
C---  COMPUTE WHEEL LOCATION FROM LOCAL CCORD. AND DIRECTICN
      DO 1 J = 1, NWHEL
      DP(J)  = PP(J)
      XP(J)  = XR(J)
      YP(J)  = YR(J)
      P(J)   = PP(J)
      X(J)   = XPOS + XR(J)*DIR
      XCHK   = DIV * HX
      XCHK1  = X(J)
      IF (XCHK1.GT.XCHK.OR.XCHK1.LT.0.0) NOB = NOB + 1
      Y(J)   = YPOS + YR(J)*DIR
C---  CORRECT LOCATION FOR SKEW
                  DIS = X(J)*ANGLE
      IF (ITHETA) DIS = (ARM-X(J))*ANGLE
C---  LOCATE ENDS OF BRIDGE
      YL1    = INT(DIS/HY)
      YLOC1  = YL1*HY
                      YL2 = INT((DIS+BRIDL)/HY+1.0)
      IF (DJV.LT.YL2) YL2 = DJV
      YLOC2  = YL2*HY
C---  RESET LOADS FOR OFF THE LENGTH OF BRIDGE
      IF (Y(J).GT.YLOC2) P(J) = 0.0
      IF (Y(J).LT.YLOC1) P(J) = 0.0
    1 CONTINUE
      IF (NOB.GT.0) WRITE (NOUT,601) IG,NOB
C---  CALL ROUTINE FOR OVERLAPPING TRUCK CHECK
      CALL CHECKR (IG,X,Y,X1,X2,Y1,Y2,ITYPE,JTYPE,NWHEL)
      IF (NOB.GT.0) GOTO 6
      IF (IPLOT.NE.1) GOTO 6
```

```
C---   CALL ROUTINE TO PLOT TRUCK
       CALL PLTTRK (NWHEL,X1,X2,Y1,Y2)
     6 CONTINUE
       IFIND  = IFIND + 1
     2 CONTINUE
C---   PRINT NOTE FOR TRUCKS FROM DATA BASE
       IF (IFIND) 3, 4, 5
     3 WRITE (NOUT,602)  ITYPE, JTYPE
       STOP
     4 WRITE (NOUT,603)  ITYPE, JTYPE
C---   PRINT WHEEL INPUT DATA
       WRITE (NOUT,605) NWHEL
       WRITE (NOUT,606) (DP(J),XP(J),YP(J),J=1,NWHEL)
       GOTO 999
     5 IFIND  = IFIND + 1
       WRITE (NOUT,604)  ITYPE, JTYPE, IFIND
C---
   601 FORMAT (8X,6HTRUCK I2,2X,4HHAS I2,18HWHEELS CFF BRIDGE
      C        /8X,29HAND LOAD CALCULATIONS ABORTED   /)
   602 FORMAT ( 8X,2A4, 24H  NOT FOUND IN DATA BASE )
   603 FORMAT ( 8X,2A4, 24H  LOCATED FROM DATA BASE )
   604 FORMAT ( 8X,2A4, 21H  OCCURS IN DATA BASE,I5, 6H TIMES )
   605 FORMAT (/8X,32HWHEEL DATA FROM TRUCK DATA BASE  ,
      C   /8X,35HNUMBER OF WHEELS                    ,1I5,
      C   /8X,30H              LOCAL       LCCAL        ,
      C   /8X,30H    LOAD      TRANS       LCNG         ,
      C   /8X,30H   (KIPS)      (FT)       (FT)         ,/ )
   606 FORMAT (7X,3F10.2)
C---
   999 RETURN
       END
       SUBROUTINE TAB207 (LINE, GRID, D, N, H)
C---
C---   THIS ROUTINE COMPUTES THE LCCATION CF GRID PCINTS
C---   NEXT TO AN APPLIED WHEEL LOAD
C---
       E      = D
       LINE   = 0
       DO 1 I = 1,N
       T      = I
       T      = ABS(T*H-D)
       IF (T.GT.E) GOTO 1
       E      = T
       LINE   = I
     1 CONTINUE
                      GRID = LINE
                      GRID = GRID * H
       IF (D.GT.GRID) LINE = LINE + 1
       IF (D.GT.GRID) GRID = GRID + H
C---
       RETURN
       END
       SUBROUTINE TAB307 (ALOAD,XDIST,YDIST,HX,HY,J,IDEGRE,DLF)
       COMMON /LOAD   / P(75),PP(75),XR(75),YR(75),X(75),Y(75)
       COMMON /A      / ILINE,JLINE,XLINE,YLINE,IPTN(300),JPTN(300),
      C                 XLOAD(300),N,IDUMC
       COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
C---
C---   THIS SUBROUTINE DIVIDES EACH INDIVIDUAL WHEEL LCAD
C---   AMONG THE 4 SURROUNDING GRID POINTS.THE IMPACT
C---   FACTOR IS ALSO BROUGHT IN AT THIS POINT.
C---
C---   COMPUTE RATIO DISTANCES FOR LOADS
```

```
      X2      = XLINE - XDIST
      Y2      = YLINE - YDIST
      X1      = HX - X2
      Y1      = HY - Y2
      ALOAD   = ALOAD * DLF
      AP      = ALOAD / (HX * HY)
C---  COMPUTE GRID LOADS FROM A WHEEL
      P11     = AP * Y2 * X2
      P21     = AP * Y2 * X1
      P12     = AP * Y1 * X2
      P22     = AP * Y1 * X1
C---  COMPUTE GRID LOCATIONS
      I1      = ILINE - 1
      J1      = JLINE - 1
      I2      = ILINE
      J2      = JLINE
C---  PRINT HEADER FOR LOADING OUTPUT (IDEGRE=1)
      IF (IDEGRE.NE.1) GOTO 1
      WRITE (NOUT,601) J
      WRITE (NOUT,602)
    1 CONTINUE
C---  CALL ROUTINE TO SUM LOADS AT GRID LOCATIONS
      CALL TAB407 (I1, J1, P11,IDEGRE)
      CALL TAB407 (I1, J2, P12,IDEGRE)
      CALL TAB407 (I2, J1, P21,IDEGRE)
      CALL TAB407 (I2, J2, P22,IDEGRE)
C---
  601 FORMAT (/,8X,5HWHEEL,I3  )
  602 FORMAT (  8X,15HX-GRID LOCATION,5X,15HY-GRID LOCATION,11X,
     C          9HGRID LOAD )
C---
      RETURN
      END
      SUBROUTINE TAB407 (I,J,P,IDEGRE)
      COMMON /A       / ILINE,JLINE,XLINE,YLINE,IPTN(300),JPTN(300),
     C                  XLOAD(300),N,IDUMO
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
C---
C---  THIS SUBROUTINE SUMS UP THE LOADS THAT OCCUR
C---  SIMULTANEOUSLY AT THE GRID LOCATIONS DUE TO THE
C---  SPECIFIED TRUCK LOADING. BY PLACEMENT OF A -1-
C---  IN COLUMN 10 OF THE TRUCK LOAD CARD INDIVIDUAL
C---  WHEEL LOADS,DIVIDED AMONG THE 4 NEIGHBORING GRID
C---  POINTS, CAN BE OBTAINED IN THE PRINTOUT.
C---
      IF(IDEGRE.NE.1) GOTO 4
C---  PRINT GRID LOADINGS
      IF (ITURN) GOTO 5
      WRITE (NOUT,601) I,J,P
      GOTO 4
    5 WRITE (NOUT,601) J,I,P
C---  CHECK VALUE OF LOAD
    4 IF ((P.LT.0.001).AND.(P.GT.-0.001)) GOTO 999
C---  SET GRID LOCATION FOR LOAD
      N             = N + 1
      IPTN(N)       = I
      JPTN(N)       = J
C---  SET LOAD VALUE
      XLOAD(N)      = -P
C---  SET GRID VALUES FOR CHECK
      ITEST         = I
```

```
      JTEST         = J
      IF(N.EQ.1) GOTO 3
C---  CHECK GRID LOCATION FOR PREVIOUS LCAD AND SUM LCADS
      NM1           = N - 1
      DO 2       JJ = 1,NM1
      INDEX         = N - JJ
      ICHEK         = IPTN(INDEX)
      JCHEK         = JPTN(INDEX)
      IF (ITEST.NE.ICHEK.OR.JTEST.NE.JCHEK) GOTO 2
      XLOAD(INDEX) = XLOAD(INDEX) - P
      N             = N - 1
    2 CONTINUE
    3 CONTINUE
C---
  601 FORMAT (13X,I5,15X,I5,15X,F10.2)
C---
  999 RETURN
      END
      SUBROUTINE PLTSPN (IDIV,JDIV,HX,HY,IGRTP)
      COMMON /SKEW   / THEA,THETA,ANGLE,ARM
      COMMON /LOAD   / P(75),PP(75),XR(75),YR(75),X(75),Y(75)
      COMMON /B      / SCLX,SCLY,YT,IDIR,NGIRDR,NGRDIV,NX,CVERH,OVERH1,
     C                 NSPAN,IDUM1
      COMMON /C      / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                 JDIST1(12,50),JDIST2(12,50)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BUFFR  / IBUF(1000)
      COMMON /INF    / INFO(58),INFO3,INF3
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DIMENSION        INO(4)
      DATA             NOBM,INO(1),INO(2) /4HSLAB,4HPRCB,4HLEM /
C---
C---  THIS SUBROUTINE PLOTS THE SPANS AND SUPPCRT PCINTS
C---  THAT COMPRISE THE BRIDGE.ACCESS TO THIS SUBROUTINE
C---  IS MADE THROUGH THE USE OF THE -IPLCT- CPTICN ON
C---  THE TRUCK LOAD CARD.
C---
C---  CALL BEGIN OF PLOT FOR TRUCK PLOT
C-----INITIATE PLOT ROUTINES FOR CDC OR IBM
      CALL PLOTS(IBUF,1000,NPLT)                                        210C80
C     CALL BGNPLT (4LPLOT,40.0,20,20)                                   NULL
C---  CALC SCALING FACTORS FOR GRID PLOT
      DIV      = IDIV
      DJV      = JDIV
      CL       = 5.7
      RATIO    = (HX*DIV)/(HY*DJV)
      IF (RATIO.GT.0.60) GOTO 1
      SCLHY    = 13.4/DJV
      SCLHX    = (HX/HY) * SCLHY
      GOTO 2
    1 SCLHX    = 8.0/DIV
      SCLHY    = (HY/HX)*SCLHX
    2 CONTINUE
C---  PLOT LONGITUDINAL GRID LINES
      YDIM     = SCLHX*DIV
      YB       = CL-YDIM/2.
      YO       = YB
      YT       = CL+YDIM/2.
      IIDIV    = IDIV+1
      DISTY    = SCLHY*DJV+1.
      DO 3   I = 1,IIDIV
C     CALL PLT (1.,YB,3)                                                NULL
```

```
      CALL PLOT (1.,YB,3)                                               210C80
C     CALL PLT (DISTY,YB,2)                                             NULL
      CALL PLOT (DISTY,YB,2)                                            210C80
    3 YB        = YB + SCLHX
      XL        = 1.
      JJDIV     = JDIV+1
C---  PLOT TRANSVERSE GRID LINES
      DO 4    I = 1,JJDIV
C     CALL PLT (XL,YO,3)                                                NULL
      CALL PLOT (XL,YO,3)                                               210C80
C     CALL PLT (XL,YT,2)                                                NULL
      CALL PLOT (XL,YT,2)                                               210C80
    4 XL        = XL + SCLHY
      SCLX      = (SCLHX/HX)
      SCLY      = (SCLHY/HY)
      IEND      = NSPAN+1
      ID        = IDIV+1
      SPAN      = 0.
      IF (IGRTP.EQ.NOBM) NGIRDR = ID
C---  PLOT SUPPORT POINTS
      DO 6    J = 1,IEND
      DO 5    I = 1,NGIRDR
                     DIS = FLOAT(IGIRDL(I))*HX*ANGLE
      IF (ITHETA) DIS = (ARM-FLOAT(IGIRDL(I))*HX)*ANGLE
                     DIS = DIS+SPAN
      PLTY      = DIS/HY*SCLHY+1.
      PLTX      = YT-FLOAT(IGIRDL(I))*SCLHX
C---  SET ICBD SYMBOL FOR CDC OR IBM
C     IP        = 3                                                        CDC
      IP        = 5                                                        IBM
    5 CALL SYMBOL (PLTY,PLTX,.21,IP,0.,-1)
      IF (J.EQ.IEND) GOTO 6
      SPAN      = SPAN+SPANL(J)
    6 CONTINUE
      IF (IGRTP.EQ.NOBM) NGIRDR = 0
C---  LABEL PROBLEM NUMBER ON TRUCK PLOT
      INO(3)    = INF3
      INO(4)    = INFO3
      YPRB      = CL - 0.80
      XPRB      = 0.40
      DO 8    I = 1,4
      CALL SYMBOL (XPRB,YPRB,0.14,INC(I),+90.0,4)
    8 YPRB      = YPRB + 0.56
C---  LABEL AXIS FOR SLAB 49 RESULTS INTERPRETATICN
      YLO       = YT+.15
      XLO       = DISTY/2.-.80
      YSO       = CL-.8
      XSO       = .83
      IF (ITURN) GOTO 7
      CALL SYMBOL (XLO,YLO,.14,12HY-COORDINATE,0.,12)
      CALL SYMBOL (XSO,YSO,.14,12HX-COORDINATE,+90.,12)
      GOTO 999
    7 CALL SYMBOL (XLO,YLO,.14,12HX-COORDINATE,0.,12)
      CALL SYMBOL (XSO,YSO,.14,12HY-COORDINATE,+90.,12)
C---
  999 RETURN
      END
      SUBROUTINE PLTTRK (NWHEL,X1,X2,Y1,Y2)
      COMMON /LOAD   / P(75),PP(75),XR(75),YR(75),X(75),Y(75)
      COMMON /B      / SCLX,SCLY,YT,IDIR,NGIRDR,NGRDIV,NX,OVERH,OVERH1,
     C                 NSPAN,IDUM1
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /BUFFR  / IBUF(1000)
```

```
      DATA              IRF/4HFORW/
C---
C---  THIS SUBROUTINE PLOTS THE TRUCKS THAT ARE LOCATED ON
C---  THE BRIDGE FOR THE SPECIFIED LOADING.ACCESS TO
C---  PLTTRK IS MADE THROUGH THE USE OF THE -IPLOT-
C---  OPTION ON THE TRUCK LOAD CARD.
C---
C---  SET CHARACTER CODE FOR CDC OR IBM
C     IP        = 4                                                   CDC
      IP        = 2                                                   IBM
C---  PLOT WHEEL LOCATION BY ARROW SYMBOL
      DO 1   J = 1,NWHEL
      XPT       =  1.0 + Y(J) * SCLY
      YPT       =  YT - (X(J)) * SCLX
C---  CHOOSE DIRECTION OF ARROW SHOWING TRUCK DIRECTION
                        AIP =    90
      IF (IDIR.EQ.IRF) AIP = - 90
    1 CALL SYMBOL (XPT,YPT,.14,IP,AIP,-1)
C---  DRAW IMAGINARY BOUNDS OF TRUCK
      XP1       = 1.0 + Y1 * SCLY
      XP2       = 1.0 + Y2 * SCLY
      YP1       = YT - (X1) * SCLX
      YP2       = YT - (X2) * SCLX
C     CALL PLT (XP1,YP1,3)                                        NULL
      CALL PLOT (XP1,YP1,3)                                       210C80
C     CALL PLT (XP2,YP1,2)                                        NULL
      CALL PLOT (XP2,YP1,2)                                       210C80
C     CALL PLT (XP2,YP2,2)                                        NULL
      CALL PLOT (XP2,YP2,2)                                       210C80
C     CALL PLT (XP1,YP2,2)                                        NULL
      CALL PLOT (XP1,YP2,2)                                       210C80
C     CALL PLT (XP1,YP1,2)                                        NULL
      CALL PLOT (XP1,YP1,2)                                       210C80
C---
      RETURN
      END
      SUBROUTINE CHECKR (IG,X,Y,X1,X2,Y1,Y2,ITYPE,JTYPE,NWHEL)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      DIMENSION         X(75),Y(75),AX(20,2),AY(20,2),MT(20,2)
C---
C---  THIS SUBROUTINE CHECKS THE LOCATIONS OF ALL THE TRUCKS
C---  THAT ARE LOCATED ON THE BRIDGE FOR THE SPECIFIED
C---  LOADING TO INSURE THAT THEY DO NOT OVERLAP.
C---
C---  SAVE THE NAME OF THE TRUCK
      MT(IG,1) = ITYPE
      MT(IG,2) = JTYPE
C---  INITIALIZE THE LOCATION OF TRUCK LIMITS
      X1        = X(1)
      X2        = X1
      Y1        = Y(1)
      Y2        = Y1
C---  FIND THE EXTREME EDGES OF THE TRUCK
      DO 1    I = 2,NWHEL
      IF (X(I).GT.X1) X1 = X(I)
      IF (X(I).LT.X2) X2 = X(I)
      IF (Y(I).GT.Y1) Y1 = Y(I)
      IF (Y(I).LT.Y2) Y2 = Y(I)
    1 CONTINUE
C---  SAVE THE TRUCK LIMITS
      AX(IG,1) = X1
      AX(IG,2) = X2
      AY(IG,1) = Y1
```

```
      AY( IG,2) = Y2
    2 CONTINUE
      IF (IG.EQ.1) GOTO 999
      JST       = IG - 1
      DO 6    J = 1,JST
      DO 3    K = 1,2
C---  CHECK FOR OVERLAPPING TRUCKS
      IF ((X1.GE.AX(J,K)).AND.(X2.LE.AX(J,K)))  GOTO 5
      GOTO 3
    5 DO 4  KT = 1,2
      IF ((Y1.LT.AY(J,KT)).OR.(Y2.GT.AY(J,KT))) GCTC 4
      WRITE (NOUT,601) IG,(MT(IG,K2),K2=1,2),J,(MT(J,K2),K2=1,2)
      GOTO 6
    4 CONTINUE
    3 CONTINUE
    6 CONTINUE
C---
  601 FORMAT (8X,5HTRUCK ,I2,2X,2A4,2X,3HAND,3X,5HTRUCK ,I2,2X,2A4,3X,
     C         7HOVERLAP      )
C---
  999 RETURN
      END
      SUBROUTINE LOADUP (ITYPE,JTYPE,XPOS,YPOS,DIR,NWHEL,IPLOT,
     C                    IDIV,JDIV,HX,HY,NOB,IG,BRIDL,IFOR)
      COMMON /SKEW    / THEA,THETA,ANGLE,ARM
      COMMON /LOAD    / P(75),PP(75),XR(75),YR(75),X(75),Y(75)
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
      DIMENSION         PSV(50),XSV(50),YSV(50)
      DATA              JSV /4HT-KP/
C---
C---  THIS SUBROUTINE TAKES AS INPUT TRUCKS THAT
C---  ARE NOT IN THE PERMANENT DATA BASE
C---
C---  CHECK FOR SAVED INPUT TRUCK
      IF (JTYPE.EQ.JSV) GOTO 2
C---  READ TRUCK INFORMATION
      READ  (NINP,501) NWHEL
      DO 1 L = 1,NWHEL
    1 READ  (NINP,502) PP(L),XR(L),YR(L)
      GOTO 4
C---  CHECK FOR TRUCK ALREADY STORED OR TO READ IN TRUCK
    2 IF (IFOR.LT.1) READ (NINP,501) NSAVE
      NWHEL   = NSAVE
      DO 3 I = 1,NWHEL
      IF (IFOR.LT.1) READ (NINP,502) PSV(I),XSV(I),YSV(I)
C---  SET TRUCK LOADS EQUAL TO SAVED LOADS THAT WERE READ IN
      PP(I)   = PSV(I)
      XR(I)   = XSV(I)
      YR(I)   = YSV(I)
    3 CONTINUE
C---  SET TRUCK SAVE PARAMETER TO KNOW TRUCK IS SAVED
      IFOR    = 1
    4 CONTINUE
      NOB     = 0
      DIV     = IDIV
      DJV     = JDIV
      DO 5 J = 1,NWHEL
      P(J)    = PP(J)
C---  CALC TRUCK LOCATION
      X(J)    = XPOS+XR(J)*DIR
      XCHK    = DIV * HX
```

```
      XCHK1  = X(J)
      IF (XCHK1.GT.XCHK.OR.XCHK1.LT.0.0) NOB = NCB + 1
      Y(J)   = YPOS+YR(J)*DIR
               DIS = X(J)*ANGLE
      IF (ITHETA) DIS = (ARM-XR(J))*ANGLE
C---  LOCATE THE END OF THE BRIDGE
      YL1    = INT(DIS/HY)
      YLOC1  = YL1*HY
      YL2    = INT((DIS+BRIDL)/HY+1.0)
      YLOC2  = YL2*HY
C---  RESET LOADS IF OFF BRIDGE LENGTH
      IF (Y(J).GT.YLOC2) P(J) = 0.0
      IF (Y(J).LT.YLOC1) P(J) = 0.0
    5 CONTINUE
      IF (NOB.GT.0) WRITE (NOUT,601) IG,NOB
C---  CALL ROUTINE FOR OVERLAPPING TRUCK CHECK
      CALL CHECKR (IG,X,Y,X1,X2,Y1,Y2,ITYPE,JTYPE,NWHEL)
C---  PRINT TRUCK WHEEL INPUT DATA
      WRITE (NOUT,602) NWHEL
      WRITE (NOUT,603) (PP(J),XR(J),YR(J),J=1,NWHEL)
      IF (NOB.GT.0)     GOTO 999
      IF (IPLOT.NE.1)  GOTO 999
C---  CALL ROUTINE TO PLOT TRUCK CN GRID
      CALL PLTTRK (NWHEL,X1,X2,Y1,Y2)
C---
  501 FORMAT (2I5)
  502 FORMAT (3F10.2)
  601 FORMAT (8X,6HTRUCK I2,2X,4HHAS I2,18HWHEELS OFF BRIDGE
     C      /8X,29HAND LOAD CALCULATIONS ABORTED   /)
  602 FORMAT (/8X,32HWHEEL DATA FRCM TRUCK INPUT           ,
     C    /8X,35HNUMBER OF WHEELS                     ,I15,
     C    /8X,30H              LOCAL      LCCAL       ,
     C    /8X,30H    LOAD      TRANS      LCNG        ,
     C    /8X,30H   (KIPS)      (FT)       (FT)       ,/ )
  603 FORMAT (7X,3F10.2)
C---
  999 RETURN
      END
      SUBROUTINE SETPST (BRIDL)
      COMMON /RD      / HX,HY,TK,TURN,BEAMI,BEAMD,EC,ES,TOP,BOT,
     C                  IGRTP,INAM,LRD,NGPDER,NCOMP,IDIV,JDIV,JNAM
     C                  ,NSPON,NPLOT,NDAT,NPUT,NAMS(50),YDAT(50,2)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL  / ITHETA,ITURN
      LOGICAL          ITHETA,ITURN
      DATA             XTX,YTY/1HX,1HY/
C---
C---  THIS ROUTINE READS THE POSTPROCESSCR DATA AND PRINTS
C---  THE AXIS ORIENTATION NOTE. IT ALSO PRINTS OUT THE
C---  COMPUTED GRID SIZE
C---
C---  READ POSTPROCESSOR OUTPUT INFORMATICN
      READ  (NINP,501) NPLOT,NDAT,NPUT
      WRITE (NOUT,603) NPLOT,NDAT,NPLT
      YDAT(1,1) = 0.0
      YDAT(1,2) = BRIDL
C---  CHECK NUMBER OF OUTPUTS TO BE READ
      IF (NPLOT.LT.1) GOTO 2
C---  READ POST PROCESSOR DATA TO BE OUTPUT
      DO 1    I = 1,NPLOT
      READ  (NINP,502) NAMS(I),(YDAT(I,J),J=1,2)
C---  ECHO PRINT DATA
      WRITE (NOUT,604) NAMS(I)
```

```
    1 IF (YDAT(I,2).GT.0.001) WRITE (NOUT,605) (YDAT(I,J),J=1,2)
    2 CONTINUE
C---  PRINT OUT THE GRID NOTE
      WRITE (NOUT,606) IDIV,JDIV,HX,HY
C---  FIND VARIABLES FOR ORIENTATICN NOTE
      XTURN      = XTX
      YTURN      = YTY
      IF (ITURN) GOTO 3
      XTURN      = YTY
      YTURN      = XTX
C---  PRINT AXIS ORIENTATION NOTE
    3 WRITE (NOUT,607) XTURN,YTURN
C---
  501 FORMAT (3I5)
  502 FORMAT (1I5,5X,2F10.2)
  603 FORMAT (5H1      80X,10HI----TRIM   ,///,
     C   5X,21HTABLE 4. OUTPUT AREAS  //,
     C    7X,26HA. POSTPROCESSOR OUTPUT
     C  /8X,45HNUMBER OF GIRDER OUTPUTS                         ,I10,
     C  /8X,45HDATA OUTPUT (1=DATA ONLY,2=DATA+PLOT)            ,I10,
     C  /8X,45HPRINT OUT OF SLAB49 MCMENTS (1=YES)              ,I10 )
  604 FORMAT (8X,30HGIRDER NUMBER TO BE CUTPUT     ,15X,1I10)
  605 FORMAT (8X,17HBEGIN DATA OUTPUT 19X,4H(FT)5X,F10.2,
     C  /8X,15HEND DATA OUTPUT 21X,4H(FT)5X,F10.2)
  606 FORMAT (4(/),5X,25HCOMPUTED GRID INFORMATICN
     C  /,8X,45HNUMBER OF INCREMENTS ACRCSS BRIDGE              ,I10,
     C  /,8X,45HNUMBER OF INCREMENTS ALONG BRIDGE               ,I10,
     C  /,8X,45HINCREMENT LENGTH ACROSS BRIDGE          (FT) ,F10.3,
     C  /,8X,45HINCREMENT LENGTH ALONG BRIDGE           (FT) ,F10.3)
  607 FORMAT(///,12X,A1,/,7X,13H* * * * * * *,/,7X,1H*,/,
     C  7X,46H*   THE COORDINATE AXES FOR THE INTERPRETATICN,/,
     C  5X,A1,1X,41H*   OF RESULTS FROM SLAB 49 ARE AS SHOWN.,/,
     C  7X,1H*,/,7X,1H*,/)
C---
      RETURN
      END
      SUBROUTINE TABLE8 (HX,HY,ISPEL)
      COMMON /SKEW    / THEA,THETA,ANGLE,ARM
      COMMON /D       / IX1(20),IY1(20),IX2(20),IY2(20),IDEFL(20),
C    C                  IDEFL(20),IXMOM(20),IYMOM(20),ISTRS(20),NUMB8,IDUM8  NULL
     C                  IXMOM(20),IYMOM(2C),ISTRS(20),NUMB8,IDUM8         100C80
      COMMON /C       / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12),
     C                  JDIST1(12,50),JDIST2(12,50)
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
C---
C---  THIS SUBROUTINE DEFINES THE LINES CR AREAS FOR
C---  CALCULATION OF DEFLECTIONS,X-MOMENTS,Y-MOMENTS,
C---  AND STRESSES.
C---
      NUMB8    = 0
      NPTS     = 0
C---  CHECK TABL 4 OPTION
      IF (ISPEL.EQ.0) GOTO 999
C---  READ IN SPECIAL OUTPUT INFORMATICN
      READ  (NINP,501) NPTS,NAREA
      WRITE (NOUT,601) NPTS,NAREA
      IF (NPTS.LT.1) GOTO 999
C---  CHECK DATA TYPE
      DO 1   I = 1,NPTS
      NUMB8    = NUMB8+1
C---  READ AND WRITE SPECIAL OUTPUT INFORMATION FCR SLAB OUTPUT
```

```
      READ  (NINP,502) X1,Y1,X2,Y2,IDEFLX,IXMM,IYMM,ISTRSX
      WRITE (NOUT,602) I,X1,Y1,X2,Y2
      WRITE (NOUT,603) IDEFLX,IXMM,IYMM,ISTRSX
C---  CONVERT DISTANCES TO STATIONS AND STORE
      IX1(I)   = X1/HX+.5
      IX2(I)   = X2/HX+.5
      IY1(I)   = Y1/HY+.5
      IY2(I)   = Y2/HY+.5
C---  STORE OPTIONS
      IDEFL(I) = IDEFLX
      IXMOM(I) = IXMM
      IYMOM(I) = IYMM
      ISTRS(I) = ISTRSX
      IF (.NOT.ITURN) GOTO 1
      IXTMP    = IX2(I)
      IYTMP    = IY2(I)
      IX2(I)   = IX1(I)
      IX1(I)   = IXTMP
      IY2(I)   = IY1(I)
      IY1(I)   = IYTMP
      IXMOM(I) = IYMM
      IYMOM(I) = IXMM
    1 CONTINUE
C---
  501 FORMAT (2I5)
  502 FORMAT (4F10.3,4I5)
  601 FORMAT (/7X,18HB. SPECIAL OUTPUT
     C  /8X,45HNUMBER OF SPECIAL OUTPUT AREAS                       ,I10,
     C  /8X,45HTYPE OF OUTPUTS (1=SLAB,2=BEAM)                      ,I10)
  602 FORMAT (/8X,19HSPECIAL OUTPUT AREA  I3,
     C  /8X,45HBEGIN LONGITUDINAL DISTANCE           (FT)           ,F10.2,
     C  /8X,45HEND LONGITUDINAL DISTANCE             (FT)           ,F10.2,
     C  /8X,45HBEGIN TRANSVERSE DISTANCE             (FT)           ,F10.2,
     C  /8X,45HEND TRANSVERSE DISTANCE               (FT)           ,F10.2)
  603 FORMAT (
     C   8X,45HDEFLECTION OUTPUT (1=YES)                            ,I10,
     C  /8X,45HLONGITUDINAL MOMENTS (1=SLAB,2=BEAM)                 ,I10,
     C  /8X,45HTRANSVERSE MOMENTS   (1=SLAB,2=BEAM)                 ,I10,
     C  /8X,45HPRINCIPLE MOMENTS                                    ,I10)
C---
  999 RETURN
      END
      SUBROUTINE TABLE9 (ISPEL)
      COMMON /RD      / HX,HY,TK,TURN,BEAMI,BEAMD,EC,ES,TCP,BOT,
     C                  IGRTP,INAM,LRD,NGRDER,NCOMP,IDIV,JDIV,JNAM
     C                  ,NSPON,NPLOT,NDAT,NPUT,NAMS(50),YDAT(50,2)
      COMMON /E       / IY19(20),IY29(20),NUMB9,IDUM9
      COMMON /UNITS   / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
      COMMON /LOGCL   / ITHETA,ITURN
      LOGICAL           ITHETA,ITURN
      DIMENSION         Y1(20),Y2(20)
C---
C---  THIS SUBROUTINE LOCATES THE LONGITUDINAL
C---  BOUNDS OF THE PRINTED OUTPUT.
C---
      NPTS     = 0
      NUMB9    = 0
C---  SET LIMITS FOR ALL DATA TO BE PRINTED
      IY19 (1) = 0
                 IY29( 1) = JDIV
      IF (ITURN) IY29( 1) = IDIV
      ICHK     = IY29( 1)
C---  CHECK TABLE 4 OPTION
```

```
      IF (ISPEL.EQ.0) GOTO 2
C---  READ NUMBER OF GENERAL OUTPUTS
      READ  (NINP,501) NPTS
      WRITE (NOUT,601) NPTS
      IF (NPTS.LT.1) GOTO 2
C---  READ GENERAL OUTPUT LIMITS
      READ  (NINP,502) ((Y1(I),Y2(I)),I=1,NPTS)
      WRITE (NOUT,602) (I,Y1(I),I,Y2(I),I=1,NPTS)
      DO 1   I = 1,NPTS
C---  CONVERT DISTANCES TO STATIONS AND STORE
      IY19(I)  = Y1(I)/HY+.5
      IY29(I)  = Y2(I)/HY+.5
C---  CHECK FOR LIMITS BEYOND BRIDGE LENGTH
      IF (IY19(I).GT.ICHK) IY19(I) = ICHK
      IF (IY29(I).GT.ICHK) IY29(I) = ICHK
      NUMB9     = NUMB9+1
    1 CONTINUE
    2 CONTINUE
C---
  501 FORMAT (I5)
  502 FORMAT (8F10.3)
  601 FORMAT (/7X,17HC. GENERAL OUTPUT
     C         /8X,25HNUMBER OF AREAS OUTPUT    20X,I10)
  602 FORMAT (/8X,19HBEGIN DISTANCE AREA  I3,14X,4H(FT)5X,F10.2,
     C         /8X,17HEND DISTANCE AREA  I3,16X,4H(FT)5X,F10.2)
C---
      RETURN
      END
      SUBROUTINE MOMENT (AREA,BEAMD,CENT,BEAMI,CONCW,NOSTR,PLOS,
     C                   PULL,EMID,TOP,BOT,SPL)
      COMMON /UNITS  / NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
C--
C--   THIS ROUTINE CALCULATES THE STRESSES AT MIDSPAN DUE TO DEAD WEIGHT
C---  AND PRESTRESSING FORCES.
C---
      SMT    = (BEAMD-CENT) / BEAMI
      SMB    =  CENT / BEAMI
C---  STRESS DUE TO DEAD LOAD OF GIRDER
      DLM    = (CONCW * AREA * SPL**2) / 8.0
      FDT    = - DLM * SMT
      FDB    =   DLM * SMB
C---  STRESS FROM AXIAL CABLE FORCE
      TPULL =    NOSTR * PULL * ( 1.0 - PLOS )
      POA    = - TPULL / AREA
C---  STRESS FROM PRESTRESSING CABLE MOMENT
      FPT    =   TPULL * EMID * SMT
      FPB    = - TPULL * EMID * SMB
C---  RESULT STRESS FROM PRESTRESSING AND DEAD WT. ( CONVERT TO PSI )
      TOP    = ( FDT + POA + FPT ) * 1000.0 / 144.0
      BOT    = ( FDB + POA + FPB ) * 1000.0 / 144.0
      WRITE (NOUT,601) TOP,BOT
C---
  601 FORMAT (//5X,40HMIDSPAN STRESSES INDUCED BY PRESTRESSING
     C        ,/,8X,23H AND GIRDER DEAD WEIGHT
     C  //8X,45HSTRESS AT GIRDER TOP                 (PSI)      ,F10.2,
     C   /8X,45HSTRESS AT GIRDER BOTTOM              (PSI)      ,F10.2)
C---
      RETURN
      END
      SUBROUTINE READY
      COMMON /BM     / EMID(2),CCMIY(5),CURHW(3)
      COMMON /C      / IGIRL(50),BRIG1(36),I1(400),I2(400),I3(400)
      COMMON /RD     / BRIG2(10),IBRG2(12),NAMS(50),YDAT(50,2)
```

```
      COMMON /INF     / NFORM(60)
      COMMON /E       /    IYNUM9(42)
      COMMON /SKEW    / THEA,BRIG3(3)
      COMMON /UNITS  /NINP,NOUT,NT3,NT7,NT8,NT9,NT10,NT11,NPLT
C---
C---  THIS SUBROUTINE MAKES A TAPE THAT IS USED BY
C---  THE SLAB49 POSTPROCESSOR
C---
      WRITE (NT11) COMIY,IGIRL,BRIG1,BRIG2,IBRG2,CURHW,
     C               NAMS,YDAT,NFORM,IYNUM9,BRIG3
C---
      RETURN
      END
```

APPENDIX D

INPUT GUIDE FOR BASGEN

INPUT GUIDE FOR DATA BASE GENERATOR

SLBDG4 - SLAB49 - SEARCH SEQUENCE

by

Robert E. Cornwell and Richard Stolleis

NUMBER OF PROBLEMS BEING RUN
5

STEEL GIRDER DATA BASE

TYPE
4

NO. OF BEAMS BEING INPUT	ALPHANUMERIC INFORMATION
5	80

GIRDER NAME		X-SECTIONAL AREA	MOMENT OF INERTIA	DEPTH OF X-SECTION	TOP FLANGE WIDTH	DISTANCE CENTROID-TO-BOTTOM
8	11	20	30	40	50	60

REINFORCED CONCRETE GIRDER DATA BASE

TYPE
4

NO. OF BEAMS BEING INPUT	ALPHANUMERIC INFORMATION
5	80

GIRDER NAME
8

INTERIOR SECTION			EXTERIOR SECTION				
MOMENT OF INERTIA	X-SECTIONAL AREA	DISTANCE CENTROID-TO-BOTTOM	MOMENT OF INERTIA	X-SECTIONAL AREA	DISTANCE CENTROID-TO-BOTTOM	EQUIVALENT SLAB THICKNESS	DEPTH OF X-SECTION
10	20	30	40	50	60	70	80

PRESTRESSED GIRDER DATA BASE

TYPE
4

NO. OF BEAMS BEING INPUT	ALPHANUMERIC INFORMATION
5	80

GIRDER NAME		X-SECTONAL AREA	MOMENT OF INERTIA	DEPTH OF X-SECTION	TOP FLANGE WIDTH	DISTANCE CENTROID-TO-BOTTOM
8	11	20	30	40	50	60

TRUCK DATA BASE

TYPE	OUTPUT FORM	
4	21	25

NO. OF TRUCKS BEING INPUT	ALPHANUMERIC INFORMATION
5	80

TRUCK NAME	NO. OF WHEELS	
8	11	15

WHEEL LOAD	LOCAL COORDINATES: TRANSVERSE LOCATION	LOCAL COORDINATES: LONGITUDINAL LOCATION
10	20	30

Steps for Implementing Data Base Generator

1) Input the number of data bases to be created and/or printed

2) Fill in selected data base information

3) To print a selected data base put a zero "Ø" in columns 1-5 of no. of beams or no. of trucks card

4) For truck output option: Long - long form of output (all wheels and loads printed)
   Short - short form of output (only truck names and no. of wheels printed)

APPENDIX E

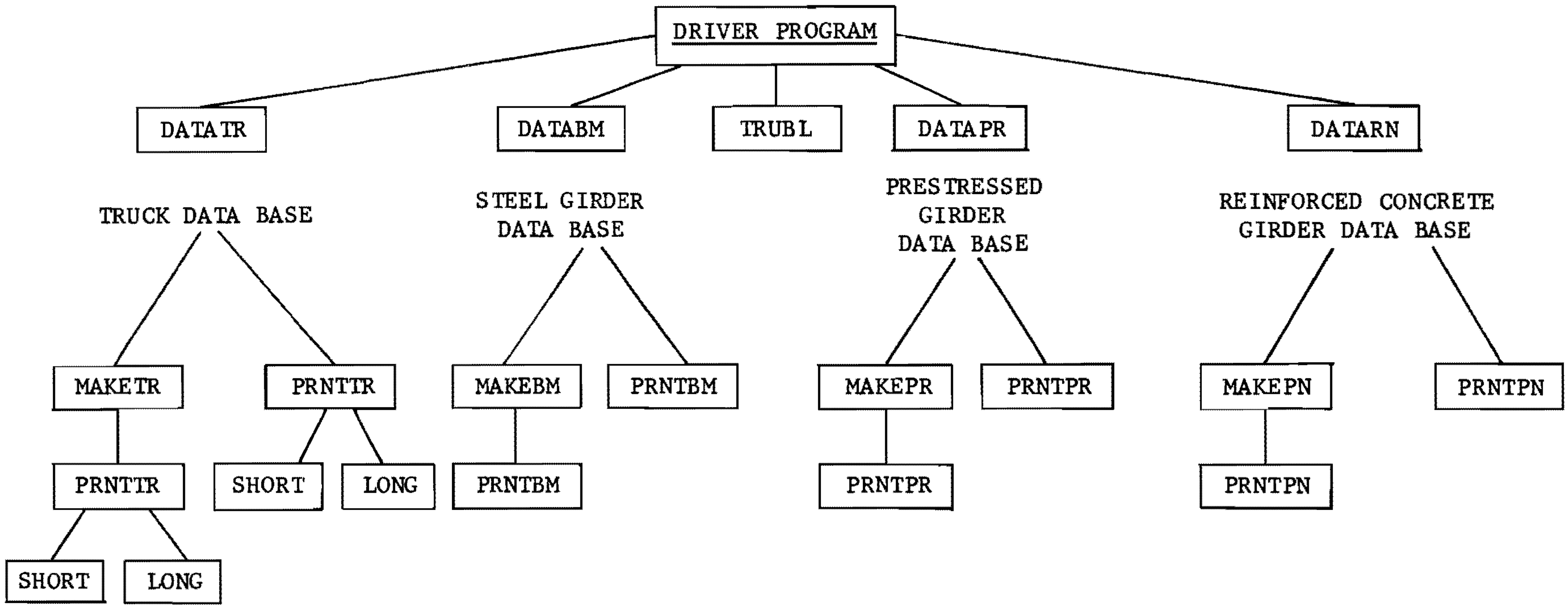
DOCUMENTATION FOR BASGEN

APPENDIX E. DOCUMENTATION FOR BASGEN

DATA BASE GENERATOR

The purpose of the data base generator is to allow the users of the TBAR (Texas Bridge Analysis program) to easily and effectively retrieve information that is necessary for the bridge classes considered herein. The data base generator/accessor allows the user to retrieve information in four general categories. The first of these data bases contains important truck information. The remaining three are used to access girder information; they are steel girder data base, reinforced concrete girder data base, and prestressed concrete girder data base. The data base generator is designed to be executed prior to carrying out an analysis using the TBAR programs. As many as four data bases may be generated and/or printed during a single execution of the program. The program is structured so that the addition of more data bases requires a minimum of effort.

The flow chart in Fig 11 shows all possible paths that may be followed. The only box where more than one path may be taken during a single execution of the DATA BASE GENERATOR/ACCESSOR is the | DRIVER PROGRAM | box. As many as four data bases may be generated in a single run. The data bases are described on the following pages.


DRIVER PROGRAM
DATATR
DATABM
TRUBL
DATAPR
DATARN
TRUCK DATA BASE
STEEL GIRDER DATA BASE
PRESTRESSED GIRDER DATA BASE
REINFORCED CONCRETE GIRDER DATA BASE
MAKETR
PRNTTR
MAKEBM
PRNTBM
MAKEPR
PRNTPR
MAKEPN
PRNTPN
PRNTTR
SHORT
LONG
PRNTBM
PRNTPR
PRNTPN
SHORT
LONG

DATA BASE GENERATOR

SUBJECT: TRUCK DATA BASE

RELATED SUBROUTINES:
1) DATATR
2) MAKETR
3) PRNTTR
4) SHORT
5) LONG

1) SUBROUTINE DATATR (NFORM1)

CALLING ROUTINE - Main Program

Required Subroutines - MAKETR
PRNTTR

DEFINITION OF INPUT VARIABLES -

NFORM1 - Controls the type of output generated for a given execution. See Subroutines SHORT and LONG for detailed description.

PURPOSE: The purpose of this subroutine is to control the action of the subroutines MAKETR and PRNTTR. If a new data base is to be generated the subroutine MAKETR will be accessed. If the data base is to be printed the subroutine PRNTTR will be engaged. In either case a decision is made to use the long or short forms of output (see subroutines LONG and SHORT).

2) SUBROUTINE MAKETR (SORL)

CALLING ROUTINE - DATATR

Required Subroutines - PRNTTR

DEFINITION OF INPUT VARIABLES -

SORL - output control variable

PURPOSE: This subroutine is designed to generate a new truck data base and then call PRNTTR to print out the new data base as it is created. The new data base is contained on FORTRAN Unit 9. The data base will contain the following information: the number of trucks in the data base, followed by two (2) four-character alphanumeric variables that specify the truck name, and the number of wheels on the truck. This information is then followed by one line per wheel specifying the wheel load and the local (truck coordinate system) X and Y position of the wheel.

3) SUBROUTINE PRNTTR (SORL)

CALLING ROUTINE - DATATR
MAKETR

Required Routines - SHORT
LONG

DEFINITION OF INPUT VARIABLES -

SORL - output control variable

PURPOSE: Subroutine PRNTTR prints an existing truck data base by (1) reading the number of entries in the data base and (2) iteratively reading and printing, with appropriate headings, the information associated with each truck. Function (1) will be carried out within the subroutine PRNTTR. Function (2) will be completed in subroutines SHORT or LONG depending on the output control variable SORL.

4) SUBROUTINE SHORT

CALLING ROUTINE - PRNTTR

Required Subroutines - None

PURPOSE: SHORT will print the information contained in an existing truck data base in a condensed form. The printed information will consist of the following: the total number of trucks contained in the data base at present, followed by the name of each truck and its total number of wheels.

5) SUBROUTINE LONG

CALLING ROUTINE - PRNTTR

Required Subroutines - None

PURPOSE: LONG will print the information contained in an existing truck data base in an extended form. The information concerning the data base will be printed in the following manner. First, the total number of trucks contained in the data base will be printed. This will be followed by the name and number of wheels on the first truck, plus a detailed description of each wheel, i.e., wheel load, and local X and Y positions. This information will be repeated until all trucks have been printed. An example is given in Fig 12.

NEW TRUCK DATA BASE CREATED

1ST TEST RUN OF LONG FORM

THE NUMBER OF TRUCKS LISTED IN DATA BASE IS 3

NAME	NO. OF WHEELS		WHEEL LOAD (KIPS)	LOCAL-X LOCATION (FT)	LOCAL-Y LOCATION (FT)
HS20V14	6	1	4.	3.	14.
		2	4.	-3.	14.
		3	16.	3.	0.
		4	16.	-3.	0.
		5	16.	3.	-14.
		6	16.	-3.	-14.
HS20V12	6		4.	3.	14.
			4.	-3.	14.
			16.	3.	0.
			16.	-3.	0.
			16.	3.	-12.
			16.	-3.	-12.
HS20V16	6		4.	3.	14.
			4.	-3.	14.
			16.	3.	0.
			16.	-3.	0.
			16.	3.	-16.
			16.	-3.	-16.

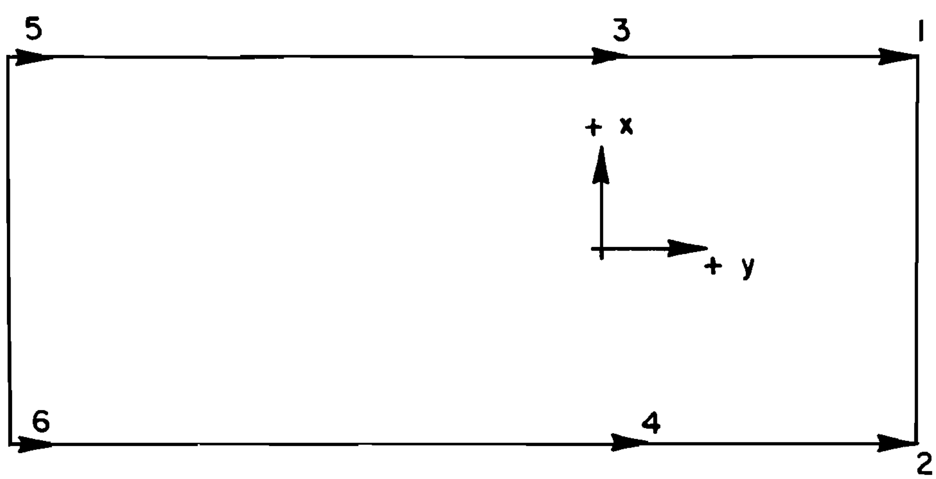


Fig 12.

DATA BASE GENERATOR

SUBJECT: STEEL GIRDER DATA BASE

RELATED SUBROUTINES:
1) DATABM
2) MAKEBM
3) PRNTBM

1) SUBROUTINE DATABM

CALLING ROUTINE - Main Program

Required Subroutines - MAKEBM
PRNTBM

PURPOSE: The purpose of the subroutine DATABM is to control the action of the subroutines MAKEBM and PRNTBM. When a new data base is to be generated, the subroutine MAKEBM will be called. If an existing data base is to be printed, PRNTBM will be accessed.

2) SUBROUTINE MAKEBM

CALLING ROUTINE - DATABM

Required Subroutines - PRNTBM

PURPOSE: This subroutine is designed to generate and print a new steel girder data base. The new data base will be contained on FORTRAN Unit 8. The new data base will contain the following information: two (2) four-character alphanumeric variables that define the girder designation. This information is followed on the same card by girder x-sectional area, girder moment of inertia, girder depth, top flange width, bottom flange width, and web thickness. All units are assumed to be in inches. MAKEBM will then access PRNTBM to produce a printer copy of the new data base.

3) SUBROUTINE PRNTBM

CALLING ROUTINE - DATABM
MAKEBM

Required Routines - None

PURPOSE: The purpose of PRNTBM is to print an existing steel girder data base. This is accomplished by (1) reading the number of girders in the data base and (2) iteratively reading and printing each girder and associated information. The information printed for each girder contains the girder's x-sectional area, moment of inertia, depth, top flange width, bottom flange width, and web thickness. All values have units of inches. An example is given in Fig 13.

NEW STEEL GIRDER DATA BASE CREATED

1ST TEST RUN OF WHOLE-BMDATA

THE NUMBER OF GIRDERS LISTED IN DATA BASE IS 5

BEAM NAME	① AREA OF BEAM (IN)	② MOMENT OF INERTIA (IN)	③ DEPTH OF MEMBER (IN)	④ TOP FLANGE WIDTH (IN)	⑤ DISTANCE CENTROID TO BOTTOM (IN)
W33X118	34.8	5900.	32.9	11.5	.7
W30X124	36.5	5360.	30.2	10.5	.9
W36X135	39.8	7820.	35.5	11.9	.8
W24X61	18.0	1540.	23.7	7.0	.6
W36X260	76.5	17300.	36.2	16.6	1.4

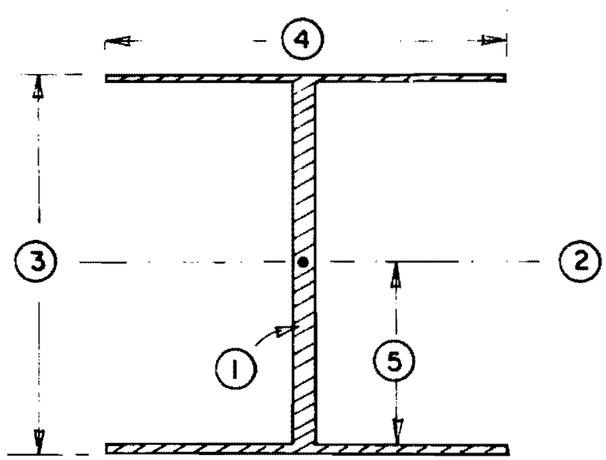


Fig 13.

DATA BASE GENERATOR

SUBJECT: REINFORCED CONCRETE GIRDER DATA BASE

RELATED SUBROUTINES:

1) DATAPR
2) MAKEPR
3) PRNTPR

1) SUBROUTINE DATAPR

CALLING ROUTINE - Main Program

Required Routines - MAKEPR
PRNTPR

PURPOSE: The purpose of DATAPR is to control the action of the subroutines MAKEPR and PRNTPR. If a new reinforced concrete girder data base is to be created DATAPR will engage MAKEPR to complete the work. If an existing reinforced concrete girder data base is only to be printed then DATAPR will call PRNTPR.

2) SUBROUTINE MAKEPR

CALLING ROUTINE - DATAPR

Required Routines - PRNTPR

PURPOSE: MAKEPR is designed to create a new prestressed concrete girder data base. The new data base will be contained on FORTRAN Unit 7. The following information will be contained in the new data base: two (2) four-character alphanumeric variables that are a unique identifier to each girder. This is then followed, on the same card, by x-sectional area, moment of inertia, depth, top flange width, and the distance from the bottom girder to the centroid. All units are to be specified in inches. PRNTPR will then be called to produce a paper copy of the new data base.

3) SUBROUTINE PRNTPR

CALLING ROUTINE - DATAPR
MAKEPR

Required Routines - None

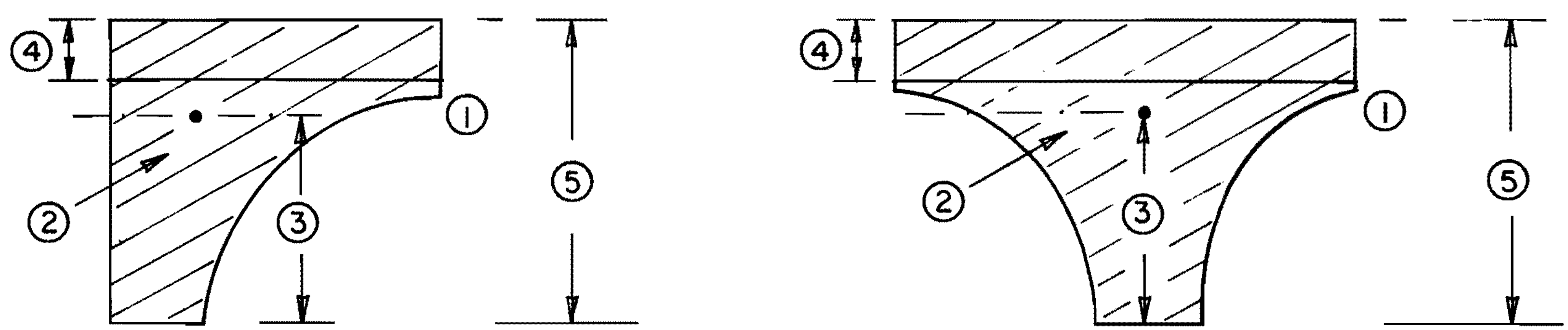
PURPOSE: PRNTPR will print all information contained in an existing prestressed concrete girder data base. This is accomplished by (1) reading the number of girders residing in the data base and (2) successively reading and printing each girder and its associated information. The information printed for each prestressed concrete girder contains the girder's x-sectional area, moment of inertia, depth, top flange width, and the distance from the bottom to the centroid of the x-section. All dimensions are given in units of inches. An example is given in Fig 14.

I-----TRIM

NEW REINFORCED GIRDER DATA BASE CREATED

THE NUMBER OF GIRDERS LISTED IN DATA BASE IS 4

REINF CONC GIRDER NAME	-----INTERIOR SECTION----- MOMENT OF INERTIA (IN) (1)	X-SECT. AREA (IN) (2)	CENTROID TO BOTTOM (IN) (3)	*****EXTERIOR SECTION***** MOMENT OF INERTIA (IN) (1)	X-SECT. AREA (IN) (2)	CENTROID TO BOTTOM (IN) (3)	EQUIVALENT SLAB THICKNESS (IN) (4)	TOTAL SECTION DEPTH (IN) (5)
PN-33	46176.	494.00	21.02	38042.	391.40	19.35	6.12	33.00
PN-34	51405.	530.00	21.87	42290.	413.80	20.12	7.52	34.00
PN-24	18098.	392.00	15.60	14651.	295.00	14.39	5.99	24.00
PN-25	20715.	428.00	16.34	16757.	317.10	15.09	7.32	25.00



Exterior Section

Interior Section

Fig 14.

DATA BASE GENERATOR

SUBJECT: REINFORCED CONCRETE GIRDER DATA BASE

RELATED SUBROUTINES:
1) DATAPN
2) MAKEPN
3) PRNTPN

1) SUBROUTINE DATAPN

CALLING ROUTINE - Main Program

Required Routines - MAKEPN
PRNTPN

PURPOSE: DATAPN controls the operation of subroutines MAKEPN and PRNTPN. When a new reinforced concrete girder data base is to be formed DATAPN will cause MAKEPN to be accessed. If an existing data base is to be printed PRNTPN will be called.

2) SUBROUTINE MAKEPN

CALLING ROUTINE - DATAPN

Required Routines - PRNTPN

PURPOSE: MAKEPN is designed to generate a new reinforced concrete girder data base. The new data base will be contained on FORTRAN Unit 10. The data base will contain the following information: the number of trucks in the data base, followed by two (2) four-character alphanumeric variables specifying the girder designation. The following information will be given on the line directly following each girder designation: moment of inertia of interior girder, x-sectional area of interior girder, and the distance from the centroid to the bottom of girder for an interior section. On the same card the moment of inertia of an exterior girder, the x-sectional area, and distance from the centroid to the bottom of an exterior girder, along with equivalent slab thickness and total section depth, will be given. All units should be in inches. MAKEPN will then call PRNTPN to produce a hard copy of the new data base.

3) SUBROUTINE PRNTPN

CALLING ROUTINE - DATAPN
MAKEPN

Required Routines - None

PURPOSE: The purpose of PRNTPN is to print an existing reinforced concrete girder data base. The printing is carried out by (1) reading the number of girders in the data base and (2) iteratively reading and printing the following, two (2) lines per girder. A girder designation consisting of two (2) four-character alphanumeric variables and the following characteristics: moment of inertia, x-sectional area, and girder centroid to bottom for an interior x-section, followed by moment of inertia, x-sectional area, and girder centroid to bottom for an exterior section. Finally the equivalent slab thickness and total section depth are printed on the same card. All dimensions are given in units of inches. An example is given in Fig 15.

NEW PRESTRESSED GIRDER DATA BASE CREATED

1ST TEST RUN OF WHOLE-PRDATA

THE NUMBER OF BEAMS LISTED IN DATA BASE IS 4

BEAM NAME	(1) AREA OF BEAM (IN)	(2) MOMENT OF INERTIA (IN)	(3) DEPTH OF MEMBER (IN)	(4) TOP FLANGE WIDTH (IN)	(5) DISTANCE CENTROID TO BOTTOM (IN)
TYPE B	288.0	32900.	32.0	8.1	15.0
TYPE C	361.0	43300.	34.0	14.0	15.0
TYPE D	476.0	154900.	50.0	19.0	20.0
TYPE E	500.0	160000.	55.0	20.0	20.0

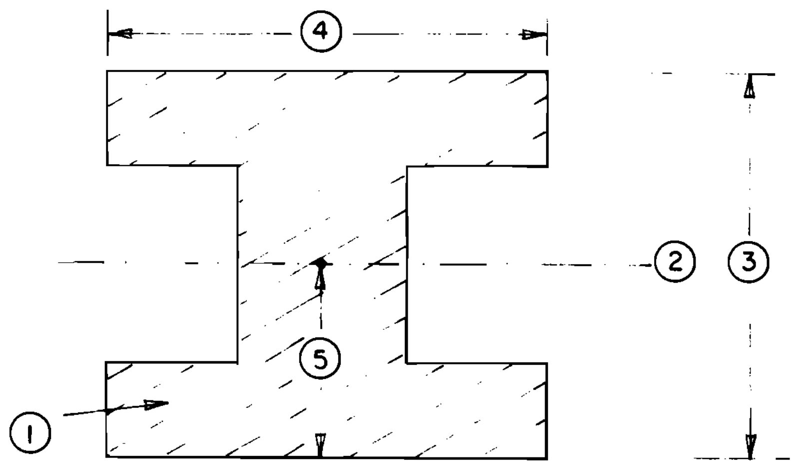


Fig 15.

1) SUBROUTINE TRUBL (ID, ID1, ID2)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

ID
ID1
ID2 - three (3) four-character alphanumeric identifies that define the incorrect data base trying to be accessed.

PURPOSE: TRUBL will print a note to the user of the data base generator indicating that an incorrect data base identifier has been specified. The program will then be terminated by TRUBL.

APPENDIX F

FORTRAN LISTING OF

DATA BASE GENERATOR - BASGEN

```
C       PROGRAM BASGEN ( INPUT, OUTPUT, TAPE7, TAPE8, TAPE9,
C                        TAPE10, TAPE5 = INPUT, TAPE6 = OUTPUT )
C
C       THIS DRIVER PROGRAM READS THE NUMBER OF INDIVIDUAL DATA BASES
C       TO BE PRINTED AND/OR CREATED, THEN ITERATIVELY READS AN ALPHA-
C       NUMERIC IDENTIFIER AND CALLS THE APPROPRIATE DATA BASE GENER-
C       ATOR/PRINTER.
C
C       VARIABLE NAMES AND USAGE FOR MAIN PROGRAM AND SUBROUTINES
C
C
C       MAIN PROGRAM
C
C       NPROB - NUMBER OF DATA BASES TO BE CREATED AND/CR PRINTED.
C       ID,ID1- FOUR CHARACTER ALPHANUMERIC IDENTIFIERS THAT INDICATE
C       ID2,    WHICH DATA BASE PRINTER/GENERATCR IS TC BE ACCESSED.
C  NOTE    ID IS THE ONLY VARIABLE USED TC IDENTIFY DATA BASE PRINTER/
C           GENERATOR.  ID1 AND ID2 ARE USED TC PRINT ERRONEOUS NAME
C           IN CASE OF ERROR.
C       NPRE  - IDENTIFIER FOR PRESTRESSED DATA BASE PRINTER/GENERATCR.
C       NST   - IDENTIFIER FOR STEEL GIRDER DATA BASE PRINTER/GENERATOR.
C       NREIN  - IDENTIFIER FOR PAN GIRDER DATA BASE PRINTER/GENERATOR.
C       NTRK  - IDENTIFIER FOR TRUCK DATA BASE PRINTER/GENERATOR.
C       NTYPE - TELLS DATA BASE GENERATOR FCR TRUCK DAT EASE WHAT TYPE
C               OF OUTPUT FORMAT IS DESIRED.
C
C
C       SUBROUTINES DATABM,MAKEBM,PRNTBM (STEEL GIRDER)
C
C       NBEAM    - NUMBER OF GIRDERS IN DATA BASE
C       MESS     - AN ARRAY USED TO HOLD A MESSAGE TC BE PRINTEC BEFORE
C                  DATA BASE INFORMATION
C       INAM,    - TOGETHER, THEY DESCRIBE AN 8 ALPHANUMERIC CHARACTER
C       JNAM       GIRDER NAME
C       AREA     - CROSS SECTIONAL AREA OF GIRDER
C       BEAMI    - MOMENT OF INERTIA
C       BEAMD    - BEAM DEPTH
C       CENT     - DISTANCE CENTROID TO BOTTOM CF GIRDER
C       TFL      - TOP FLANGE WIDTH
C
C       SUBROUTINES DATAPN,MAKEPN,PRNTPN (P.C. GIRDER)
C
C---    NBEAM = NUMBER OF GIRDERS IN DATA BASE
C---    IJ,JJ = NAME OF GIRDER
C---    C1    = MOMENT OF INERTIA OF INTERIOR SECTICN CF BRIDGE
C---    AI    = CROSS SECTIONAL AREA OF INTERIOR SECT.
C---    Y1    = CENTROID TO BOTTOM CF GIRDER - INTERIOR SECT.
C---    C2    = MOMENT OF INERTIA OF EXTERIOR SECT.
C---    AE    = CROSS SECTIONAL AREA OF EXTERIOR SECT.
C---    Y2    = CENTROID TO BOTTOM OF GIRDER - EXTERIOR SECT.
C---    SLI   = EQUIVALENT SLAB THICKNESS
C---    BMD   = TOTAL SECTION DEPTH
C---    MESS  = AN ARRAY USED TO HOLD A MESSAGE TO BE PRINTED BEFCRE DATA
C               BASE INFORMATION.
C
C
C       SUBROUTINES DATAPR,MAKEPR,PRNTPR (PRESTRESSED GIRDER)
C
C       NBEAM    - NUMBER OF BEAMS IN DATA BASE
C       MESS     - AN ARRAY USED TO HOLD A MESSAGE TO BE PRINTED BEFORE
C                  DATA BASE INFORMATICN
C       INAM,    - TOGETHER, THEY DESCRIBE AN 8 ALPHANUMERIC CHARACTER
C       JNAM       BEAM NAME
```

```
C       AREA     - CROSS SECTIONAL AREA OF BEAM
C       BEAMI    - MOMENT OF INERTIA
C       BEAMD    - BEAM DEPTH
C       TFL      - TOP FLANGE WIDTH
C       CENT     - DISTANCE CENTROID TO BOTTOM CF GIRDER
C
C
C       SUBROUTINES DATATR,MAKETR,PRNTTR,LCNG,SHORT (TRUCK)
C
C       INAM,JNAM - TOGETHER THEY CONSTITUTE AN 8 CHARACTER ALPHA-
C                   NUMERIC TRUCK NAME.
C       NTRUCK    - NUMBER OF TRUCKS IN DATA BASE.
C       P         - LOAD ASSOCIATED WITH A WHEEL
C       X         - LOCAL X COORDINATE ASSCCIATED WITH A WHEEL.
C       Y         - LOCAL Y COORDINATE ASSCCIATED WITH A WHEEL.
C       NWHEEL    - NUMBER OF WHEELS ON TRUCK
C       NDUM      - DUMMY VARIABLE.
C       MESS      - USED TO HOLD A MESSAGE TO BE PRINTEC BEFCRE
C                   DATA BASE INFORMATION.  USER INPUT.
C       NFORM1    - USED TO DETERMINE THE CUTPUT FORMAT THE USER
C                   DESIRES.
C       N1        - USED AS CONSTANT BY PRCGRAM TC CCMPARE WITH
C                   NFORM1.
C       NBLANK    - USED TO TEST FOR DEFAULT LCNG FCRM CF OUTPUT.
C       SORL      - A LOGICAL VARIABLE USED TO TELL SUBRCUTINE
C                   PRNTTR WHICH OUTPUT FORMAT IS DESIRED.
C       NT        - A TRUNCATION OF NTRUCK/2.  USED TC PRINT TWC
C                   TRUCKS ON THE SAME LINE.
C
C
C
C-----TOO MANY VALUES FOR DATA, REDUCE TC 4 TO AVCID DIAGNCSTIC - JJP    090C80
C       DIMENSION NAME(20)                                                NULL
        DIMENSION NAME( 4)                                                090C80
        DATA NAME /4HPRES,4HSTEE,4HREIN,4HTRUC/
        LOGICAL CHECK
C       CURRENT NUMBER OF DATA BASE PRINTER/GENERATORS
        NGEN = 5
C
C   NOTE     TO ADD A DATA BASE GENERATOR/PRINTER RCUTINE TC THIS
C            PROGRAM
C            1)  SET NGEN EQUAL TO THE NEW NUMBER CF GENERATCR/PRINTERS.
C            2)  SET NAME(NGEN) EQUAL TO THE CODE WCRD THAT WILL ACCESS
C                THAT GENERATOR.  E.G. NAME(1) = 4HPRES
C            3)  GO TO STATEMENT NUMBER 2 AND ADD AN IF STATEMENT THAT
C                WILL CALL THE NEW ROUTINE IF ACCESSED.
C
C       TAPE NUMBERS
        NINP = 5
        NOUT = 6
C
C
C       READ NUMBER OF PROBLEMS
        READ(NINP,1)NPROB
        WRITE(NOUT,3)NPROB
        DO  2 I = 1,NPROB
C       READ IDENTIFIER
        READ(NINP,4)ID,ID1,ID2,NTYPE
C       THE NEXT 3 LINES DETERMINE IF A VALID IDENTIFIER WAS READ
        CHECK = .FALSE.
        DO 5 J = 1,NGEN
      5 CHECK = CHECK.OR.(ID.EQ.NAME(J))
        IF(.NOT.CHECK) CALL TRUBL(ID,ID1,ID2)
```

```
      IF( ID.EQ.NAME( 1)) CALL DATAPR
      IF( ID.EQ.NAME( 2)) CALL DATABM
      IF( ID.EQ.NAME( 3)) CALL DATAPN
      IF( ID.EQ.NAME( 4)) CALL DATATR ( NTYPE )
    2 CONTINUE
C
    1 FORMAT( 1I5)
    3 FORMAT( 1H1,10X,46HTHE NUMBER OF DATA BASE PROBLEMS REQUESTED IS ,
     .,1I5//)
    4 FORMAT(3A4,8X,1A4)
C
      STOP
      END
C     PROGRAM DATABM ( INPUT, OUTPUT, TAPE8, TAPE5 = INPUT,
C                      TAPE6 = OUTPUT )
C
      SUBROUTINE DATABM
C
C     THIS SUBROUTINE PERFORMS ONE OF TWO FUNCTIONS
C
C 1) GIVEN A DATA CARD WITH THE FIRST 5 COLUMNS BLANK IT WILL SHOW AN
C    EXISTING STEEL GIRDER DATA BASE.  THE REST OF THE CARD IS INTER-
C    PRETED AS A 75 CHARACTER ALPHANUMERIC MESSAGE TO BE ECHO PRINTED.
C
C 2) OTHERWISE, CREATE A DATA BASE CONTAINING STEEL GIRDER INFORMATION
C    FOR EACH INDIVIDUAL GIRDER, ONE LINE OF INFORMATION IS WRITTEN
C    ON TAPE(NT8).  EACH LINE CONTAINS (IN THIS ORDER)
C
C     2 ALPHANUMERIC VARIABLES TO IDENTIFY STEEL GIRDER
C    CROSS SECTIONAL AREA
C    MOMENT OF INERTIA
C    BEAM DEPTH
C    TOP FLANGE WIDTH
C    BOTTOM FLANGE WIDTH
C    WEB THICKNESS
C
C    ALL UNITS ARE ASSUMED TO BE INCHES
C    ALL I/O ON DATA BASES IS PERFORMED USING STANDARD UNFORMATTED
C    FORTRAN WRITE STATEMENTS.
C
C
C
C
      COMMON NBEAM,MESS( 19),NINP,NOUT,NT8
C
C     TAPE NUMBERS
C
      NOUT = 6
      NINP = 5
      NT8 = 8
C     READ NBEAM, IF ZERO PRINT EXISTING DATA BASE,(SUB. PRNTBM)
C                 IF NONZERO CREATE A DATA BASE(SUB. MAKEBM).
C
      READ(NINP,1)NBEAM,MESS
      IF(NBEAM.NE.0) CALL MAKEBM
      IF(NBEAM.EQ.0) CALL PRNTBM
C
    1 FORMAT( 1I5,18A4,1A3)
C
      RETURN
      END
C
      SUBROUTINE MAKEBM
```

```
      COMMON NBEAM,MESS(19),NINP,NOUT,NT8
C
C     THIS SUBROUTINE WILL CREATE A DATA BASE, THEN CALL SUBRCUTINE
C     PRNTBM TO PRODUCE PRINTED OUTPUT.
C
      WRITE(NOUT,1)
C
C     BEGIN DATA BASE BY WRITING THE NUMBER OF ENTRIES IN BASE
      REWIND NT8
      WRITE(NT8)NBEAM
C     READ AND WRITE (NBEAM) LINES OF INFORMATICN ABCUT (NBEAM) GIRDERS.
      DO 2 I=1,NBEAM
      READ(NINP,3)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT
    2 WRITE(NT8)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT
C     PRINT RESULTING DATA BASE
      CALL PRNTBM

    1 FORMAT(1H1,85X,10HI-----TRIM,/
     .       ,7X,34HNEW STEEL GIRDER DATA BASE CREATED,//)
    3 FORMAT(2A4,2X,6F10.2)

      RETURN
      END


      SUBROUTINE PRNTBM
      COMMON NBEAM,MESS(19),NINP,NOUT,NT8
C     THIS SUBROUTINE PRINTS AN EXISTING DATA BASE BY
C             1)  READING THE NUMBER OF GIRDERS IN THE DATA BASE.
C             2)  ITERATIVELY LOOKING AT EACH GIRDER ENTRY AND
C                 PRINTING THE INFORMATICN ASSOCIATED WITH IT.

      REWIND NT8
      IF(NBEAM.EQ.0) WRITE(NOUT,1)
      READ(NT8)NBEAM
      WRITE(NOUT,2)MESS,NBEAM
C     PRINT OUT HEADINGS FOR DATA BASE INFCRMATICN
      WRITE(NOUT,3)
C     ITERATIVELY PRINT OUT GIRDER INFORMATION
      DO 4 I=1,NBEAM
      READ(NT8)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT
    4 WRITE(NOUT,5)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT

    1 FORMAT(1H1,85X,10HI-----TRIM,/
     .       ,7X,30HSTEEL GIRDER DATA BASE PRINTED,//)
    2 FORMAT(5X,18A4,1A3,//7X,39HTHE NUMBER OF GIRDERS LISTED IN DATA BA
     .       ,5HSE IS,1I3,//)
    3 FORMAT(6X,4HBEAM,7X,4HAREA,6X,6HMOMENT,8X,8HDEPTH OF,4X
     .       ,3HTOP,9X,8HDISTANCE,/,6X,4HNAME,7X,2HOF,8X,2HOF,
     .        12X,6HMEMBER,6X,6HFLANGE,6X,8HCENTRCID,/,
     .        17X,4HBEAM,6X,7HINERTIA,19X,5HWIDTH,7X,9HTC BCTTOM,/,
     .       17X,4H(IN),6X,4H(IN),10X,4H(IN),8X,4H(IN),8X,
     .       4H(IN)//)
    5 FORMAT(5X,2A4,2X,1F6.1,4X,1F8.0,4X,1F8.1,4X,1F8.1,4X,1F8.1,/)

      RETURN
      END
C     PROGRAM DATAPN ( INPUT, OUTPUT, TAPE10, TAPE5 = INPUT,
C    .                 TAPE6 = OUTPUT )

      SUBROUTINE DATAPN

C     THIS SUBROUTINE PERFORMS ONE OF TWC FUNCTICNS
```

```
C
C 1) GIVEN A DATA CARD WITH THE FIRST 5 CCLUMNS BLANK IT WILL SHOW AN
C    EXISTING PAN GIRDER DATA BASE.  THE REST OF THE CARD IS INTER-
C    PRETED AS A 75 CHARACTER ALPHANUMERIC MESSAGE TO BE ECHO PRINTED.
C
C 2) OTHERWISE, CREATE A DATA BASE CCNTAINING PAN GIRCER INFCRMATION
C    FOR EACH INDIVIDUAL GIRDER, ONE LINE OF INFCRMATICN IS WRITTEN
C    ON TAPE(NT10).  EACH LINE CONTAINS (IN THIS CRDER)
C
C---  NUMBER OF GIRDERS IN DATA BASE
C---  NAME OF GIRDER
C---  MOMENT OF INERTIA OF INTERIOR SECTION OF BRIDGE
C---  CROSS SECTIONAL AREA OF INTERIOR SECT.
C---  CENTROID TO BOTTOM OF GIRDER - INTERIOR SECT.
C---  MOMENT OF INERTIA OF EXTERIOR SECT.
C---  CROSS SECTIONAL AREA OF EXTERICR SECT.
C---  CENTROID TO BOTTOM OF GIRDER - EXTERIOR SECT.
C---  EQUIVALENT SLAB THICKNESS
C---  TOTAL SECTION DEPTH
C
C    ALL UNITS ARE ASSUMED TO BE INCHES
C    ALL I/O ON DATA BASES IS PERFORMED USING STANDARD UNFORMATTED
C    FORTRAN WRITE STATEMENTS.


      COMMON NBEAM,MESS(19),NINP,NOUT,NT10
C     TAPE NUMBERS

      NINP = 5
      NOUT = 6
      NT10 = 10

C     READ NBEAM, IF ZERO PRINT EXISTING DATA BASE,(SUB. PRNTPN)
C                 IF NONZERO CREATE A DATA BASE(SUB. MAKEPN).

      READ(NINP,1)NBEAM,MESS
      IF(NBEAM.NE.0) CALL MAKEPN
      IF(NBEAM.EQ.0) CALL PRNTPN

    1 FORMAT(1I5,18A4,1A3)

      RETURN
      END

      SUBROUTINE MAKEPN
      COMMON NBEAM,MESS(19),NINP,NOUT,NT10

C     THIS SUBROUTINE WILL CREATE A DATA BASE, THEN CALL SUBROUTINE
C     PRNTPN TO PRODUCE PRINTED OUTPUT.

      WRITE(NOUT,1)

C     BEGIN DATA BASE BY WRITING THE NUMBER OF ENTRIES IN BASE
      REWIND NT10
      WRITE(NT10)NBEAM
C     READ AND WRITE (NBEAM) LINES OF INFORMATION ABOUT (NBEAM) GIRDERS.
      DO 2 I=1,NBEAM
      READ(NINP,3)IJ,JJ,C1,AI,Y1,C2,AE,Y2,SLI,BMD
    2 WRITE(NT10)IJ,JJ,C1,AI,Y1,C2,AE,Y2,SLI,BMD
C     PRINT RESULTING DATA BASE
      CALL PRNTPN
```

```
    1 FORMAT(1H1,85X,10HI-----TRIM,/
     .       ,7X,39HNEW REINFORCED GIRDER DATA BASE CREATEC,//)
    3 FORMAT(2A4,/,8F10.3)

      RETURN
      END


      SUBROUTINE PRNTPN
      COMMON NBEAM,MESS(19),NINP,NOUT,NT10
C     THIS SUBROUTINE PRINTS AN EXISTING DATA BASE BY
C             1)  READING THE NUMBER OF GIRDERS IN THE DATA BASE.
C             2)  ITERATIVELY LOOKING AT EACH GIRDER ENTRY AND
C                 PRINTING THE INFORMATICN ASSOCIATED WITH IT.

      REWIND NT10
      IF(NBEAM.EQ.0) WRITE(NOUT,1)
      READ(NT10)NBEAM
      WRITE(NOUT,2)MESS,NBEAM
C     PRINT OUT HEADINGS FOR DATA BASE INFORMATICN
      WRITE(NOUT,3)
C     ITERATIVELY PRINT OUT GIRDER INFORMATICN
      DO 4 I=1,NBEAM
      READ(NT10)IJ,JJ,C1,AI,Y1,C2,AE,Y2,SLI,BMD
    4 WRITE(NOUT,5)IJ,JJ,C1,AI,Y1,C2,AE,Y2,SLI,BMD

    1 FORMAT(1H1,85X,10HI-----TRIM,/
     .       ,7X,35HREINFORCED GIRDER DATA BASE PRINTED,//)
    2 FORMAT(5X,18A4,1A3,//7X,39HTHE NUMBER OF GIRDERS LISTED IN DATA BA
     .       ,5HSE IS,1I3,//)
    3 FORMAT(11X,26H-----INTERIOR SECTION-----,1X,
     .            26H*****EXTERIOR SECTION*****,/,
     .         3X,5HREINF,3X,6HMOMENT,3X,7HX-SECT.,2X,8HCENTRCID,1X,
     .       6HMOMENT,3X,7HX-SECT.,2X,8HCENTRCID,2X,10HEQUIVALENT,3X,
     .       5HTOTAL,/,
     .         3X,4HCONC,4X,2HOF,7X,4HAREA,5X,2HTC,7X,2HCF,7X,4HAREA,
     .         5X,2HTO,8X,4HSLAB,9X,7HSECTION,/,
     .   3X,6HGIRDER,2X,7HINERTIA,2X,4H(IN),5X,6HBOTTCM,3X,7HINERTIA,
     .         2X,4H(IN),5X,6HBOTTOM,4X,9HTHICKNESS,4X,5HDEPTH,/,3X,
     .         4HNAME,
     .         4X,4H(IN),14X,4H(IN),5X,4H(IN),14X,4H(IN),6X,4H(IN),9X,
     .         4H(IN),//,)
    5 FORMAT ( 3X,2A4,1F7.0,2F8.2,1F9.0,2F9.2,2(2X,1F9.2))
      RETURN
      END
C     PROGRAM DATAPR ( INPUT, OUTPUT, TAPE7, TAPE5 = INPUT,
C    .                 TAPE6 = OUTPUT )

      SUBROUTINE DATAPR

C     THIS SUBROUTINE PERFORMS ONE OF TWO FUNCTICNS
C
C 1) GIVEN A DATA CARD WITH THE FIRST 5 CCLUMNS BLANK IT WILL SHOW AN
C    EXISTING PRESTRESSED BEAM CROSS SECTION DATA BASE.  THE REST OF
C    THE CARD IS INTERPRETED AS A 75 CHARACTER ALPHANUMERIC MESSAGE
C    TO BE ECHO PRINTED.
C
C 2) OTHERWISE, CREATE A DATA BASE CCNTAINING PRESTRESSED BEAM INFO.
C    FOR EACH INDIVIDUAL BEAM, ONE LINE CF INFORMATICN IS WRITTEN
C    ON TAPE(NT7).  EACH LINE CONTAINS (IN THIS ORDER)
C
C    2 ALPHANUMERIC VARIABLES TO IDENTIFY PRESTRESSED BEAM
C    CROSS SECTIONAL AREA
```

```
C     MOMENT OF INERTIA
C     BEAM DEPTH
C     TOP FLANGE WIDTH
C     DISTANCE FROM BOTTOM TO CENTROID.
C     DUMMY VARIABLE USED TO KEEP THE FORM OF THIS DATA BASE
C     THE SAME AS THE STEEL GIRDER DATA BASE.
C
C     ALL UNITS ARE ASSUMED TO BE INCHES
C     ALL I/O ON DATA BASES IS PERFORMED USING STANDARD UNFORMATTED
C     FORTRAN WRITE STATEMENTS.



      COMMON NBEAM,MESS(19),NINP,NOUT,NT7

C     TAPE NUMBERS

      NINP = 5
      NOUT = 6
      NT7 = 7

C     READ NBEAM, IF ZERO PRINT EXISTING DATA BASE,(SUB. PRNTPR)
C                 IF NONZERO CREATE A DATA BASE(SUB. MAKEPR).

      READ(NINP,1)NBEAM,MESS
      IF(NBEAM.NE.0) CALL MAKEPR
      IF(NBEAM.EQ.0) CALL PRNTPR

    1 FORMAT(1I5,18A4,1A3)

      RETURN
      END

      SUBROUTINE MAKEPR
      COMMON NBEAM,MESS(19),NINP,NOUT,NT7

C     THIS SUBROUTINE WILL CREATE A DATA BASE, THEN CALL SUBROUTINE
C     PRNTPR TO PRODUCE PRINTED OUTPUT.

      WRITE(NOUT,1)

C     BEGIN DATA BASE BY WRITING THE NUMBER OF ENTRIES IN BASE
      REWIND NT7
      WRITE(NT7)NBEAM
C     READ AND WRITE (NBEAM) LINES OF INFORMATION ABOUT (NBEAM) BEAMS.
      DO 2 I=1,NBEAM
      READ(NINP,3)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT
    2 WRITE(NT7)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT
C     PRINT RESULTING DATA BASE
      CALL PRNTPR

    1 FORMAT(1H1,85X,10HI-----TRIM,/
     .       ,7X,40HNEW PRESTRESSED GIRDER DATA BASE CREATED,//)
    3 FORMAT(2A4,2X,6F10.2)

      RETURN
      END


      SUBROUTINE PRNTPR
      COMMON NBEAM,MESS(19),NINP,NOUT,NT7
C     THIS SUBROUTINE PRINTS AN EXISTING DATA BASE BY
```

```
C               1)  READING THE NUMBER OF BEAMS IN THE DATA BASE.
C               2)  ITERATIVELY LOOKING AT EACH BEAM ENTRY AND
C                   PRINTING THE INFORMATICN ASSOCIATED WITH IT.

      REWIND NT7
      IF(NBEAM.EQ.0) WRITE(NOUT,1)
      READ(NT7)NBEAM
      WRITE(NOUT,2)MESS,NBEAM
C     PRINT OUT HEADINGS FOR DATA BASE INFORMATICN
      WRITE(NOUT,3)
C     ITERATIVELY PRINT OUT BEAM INFCRMATION
      DO 4 I=1,NBEAM
      READ(NT7)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT
    4 WRITE(NOUT,5)INAM,JNAM,AREA,BEAMI,BEAMD,TFL,CENT

    1 FORMAT(1H1,85X,10HI-----TRIM,/
     .       ,7X,36HPRESTRESSED GIRDER DATA BASE PRINTED,//)
    2 FORMAT(5X,18A4,1A3,//7X,
     .42HTHE NUMBER OF BEAMS LISTED IN DATA BASE IS,1I3,//)
    3 FORMAT(6X,4HBEAM,7X,4HAREA,6X,6HMOMENT,8X,8HDEPTH OF,4X
     .       ,3HTOP,9X,8HDISTANCE,/,6X,4HNAME,7X,2HCF,8X,2HCF,
     .        12X,6HMEMBER,6X,6HFLANGE,6X,8HCENTROID,/,
     .        17X,4HBEAM,6X,7HINERTIA,19X,5HWIDTH,7X,9HTO BCTTCM,/,
     .       17X,4H(IN),6X,4H(IN),10X,4H(IN),8X,4H(IN),8X,
     .       4H(IN)//)
    5 FORMAT(5X,2A4,2X,1F6.1,4X,1F8.0,4X,1F8.1,4X,1F8.1,4X,1F8.1,
     .        4X,/)

      RETURN
      END
      SUBROUTINE TRUBL ( ID,ID1,ID2 )
C
C     THIS SUBROUTINE IS CALLED BY THE DRIVER PRCGRAM WHEN AN INCORRECT
C     DATA BASE PRINTER/GENERATOR CODE (ID,ID1,ID2 IN DRIVER) IS INPUT
C     TO THE PROGRAM.  THE INCORRECT CODE IS PRINTED ANC THE PRCGRAM
C     IS TERMINATED.
C
C     TAPE NUMBER
      NOUT = 6
      WRITE(NOUT,1)ID,ID1,ID2
      STOP
    1 FORMAT(1H1,10X,16HTHE INPUT CODE -,3A4,
     .       35H-IS INCORRECT.  PROGRAM TERMINATED.)
      END
C-----PROGRAM DATATR ( INPUT, OUTPUT, TAPE9 )

      SUBROUTINE DATATR ( NFORM1 )

C     THIS SUBROUTINE PERFORMS ONE OF TWC FUNCTICNS

C 1)  GIVEN A DATA CARD WITH THE FIRST FIVE COLUMNS BLANK IT
C     WILL SHOW AN  EXISTING TRUCK DATA BASE.  THE  TWO AL-
C     TERNATIVES OF OUTPUT ARE THE SHORT AND LONG FCRMS.

C     A)  SHORT FORM - THE NUMBER  OF TRUCKS IN THE DATA BASE
C         IS PRINTED, THEN ALL TRUCK NAMES AND THE NUMBER OF
C         WHEELS ON EACH TRUCK IS PRINTED.

C     B)  LONG FORM - THE NUMBER OF TRUCKS IN THE DATA BASE IS
C         PRINTED, THEN ALL TRUCK NAMES.  WITH EACH TRUCK THE
C         FOLLOWING DATA IS GIVEN

C         NUMBER OF WHEELS
```

```
C          NUMBER OF REAR AXLE WHEELS
C          LOAD ASSOCIATED WITH EACH WHEEL AND THE CORRESPONDING
C          LOCAL X AND Y COORDINATES OF EACH  WHEEL


C      THE LONG FORM IS IMPLEMENTED BY THE WORD LONG PUNCHED
C      IN COLUMNS 21-24 OF THE SAME CARD THAT IS USED TO
C      ACTIVATE THIS DATA BASE PRINTER/GENERATOR.  THE SHORT
C      FORM IS SPECIFIED BY THE WORD SHORT PUNCHED IN COLS.
C      21-25.  THE DEFAULT FORM IS THE LONG FORM AND CAN BE
C      IMPLEMENTED BY PUTTING NOTHING IN THE SPACE RESERVED
C      FOR THE CODE WORDS (LONG,SHORT).

C 2)   OTHERWISE, CREATE A TRUCK DATA BASE THAT CONTAINS THE FOLLOWING
C      INFORMATION
C
C      NUMBER OF TRUCKS IN THE DATA BASE
C      FOR EACH TRUCK
C         (2) FOUR CHARACTER ALPHANUMERIC VARIABLES THAT SPECIFY
C         A TRUCK NAME
C         NUMBER OF WHEELS
C         NUMBER OF WHEELS ON THE BACK AXLE
C         FOR EACH WHEEL
C            LOAD ASSOCIATED WITH WHEEL (KIPS) AND THE CORRESPONDING
C            LOCAL X AND LOCAL Y COORDINATES OF THAT WHEEL(FEET)


C  NOTE    THE SHORT OR LONG FORM MUST BE SPECIFIED FOR BOTH
C          FUNCTIONS 1 AND 2 (ABOVE).


      COMMON NTRUCK,MESS(19),NINP,NOUT,NT9
      DATA N1,NBLANK/4HSHOR,4H    /
C-----NBLANK NOT REFERENCED - SET TO DUMMY TO AVOID DIAGNOSTIC - JJP    090C80
                NDUMY = NBLANK                                          090C80


C      TAPE NUMBERS

      NINP = 5
      NOUT = 6
      NT9  = 9

      WRITE(NOUT,3)
C      READ NTRUCK, IF ZERO PRINT EXISTING DATA BASE(SUB. PRNTTR)
C                   IF NON-ZERO CREATE A DATA BASE(SUB. MAKETR)
      READ(NINP,1)NTRUCK,MESS
      IF(NTRUCK.NE.0)
     .CALL MAKETR(N1.NE.NFORM1)
      IF(NTRUCK.EQ.0)
     .CALL PRNTTR(N1.NE.NFORM1)
      RETURN
    1 FORMAT(1I5,18A4,1A3)
    3 FORMAT(1H1,85X,10HI-----TRIM,//)

      END
      SUBROUTINE MAKETR (SORL)
C      THIS SUBROUTINE WILL CREATE A TRUCK DATA BASE THEN CALL
C      SUBROUTINE PRNTTR TO PRINT THE DATA BASE


      COMMON NTRUCK,MESS(19),NINP,NOUT,NT9
```

```
      LOGICAL SORL

      NDUM = 0
C     BEGIN DATA BASE BY WRITING THE NUMBER OF ENTRIES IN EASE.
      REWIND NT9
      WRITE(NT9)NTRUCK
C     READ AND WRITE (NTRUCK) MAJOR ENTRIES(1 OR MORE LINES/ENTRY).
      DO  1 I = 1,NTRUCK
C     READ AND WRITE TRUCK NAME, NUMBER CF WHEELS AND EACK WHEELS.
      READ(NINP,2)INAM,JNAM,NWHEEL
      WRITE(NT9)INAM,JNAM,NWHEEL,NDUM
      DO  1 J = 1,NWHEEL
C     READ AD WRITE (NWHEEL) LINES.  EACH LINE CONTAINS WHEEL LOAD
C     AND LOCAL X AND Y COORDINATES OF WHEEL.
      READ(NINP,3)P,X,Y
    1 WRITE(NT9)P,X,Y
      WRITE(6,4)
C     NOW PRINT DATA BASE BY CALLING SUBROUTINE PRNTTR
      CALL PRNTTR(SORL)
      RETURN
    2 FORMAT(2A4,2X,2I5)
    3 FORMAT(3F10.2)
    4 FORMAT(7X,27HNEW TRUCK DATA BASE CREATED,//)
      END
      SUBROUTINE PRNTTR (SORL)
C     THIS SUBROUTINE PRINTS AN EXISTING TRUCK DATA BASE BY
C     1)  READING THE NUMBER OF ENTRIES IN THE DATA BASE
C     2)  ITERATIVELY READING AD PRINTING (WITH APPROPRIATE HEADINGS)
C         THE INFORMATION ASSOCIATED WITH EACH TRUCK.
C  NOTE   ONLY FUNCTION 1(ABOVE) IS PERFCRMED IN THIS SUBROUTINE.
C         THE LOGICAL VARIABLE SORL DETERMINES WHICH CUTPUT FORMAT
C         IS DESIRED AND ACCORDINGLY, IS USED TO CALL SUBRCUTINE
C         SHORT OR SUBROUTINE LONG TC IMPLEMENT THE DESIRED OUTPUT
C         FORMAT.

      COMMON NTRUCK,MESS(19),NINP,NOUT,NT9
      LOGICAL SORL

      IF (NTRUCK.EQ.0) WRITE (6,1)
      REWIND NT9
      READ(NT9)NTRUCK
      IF(SORL)    CALL  LONG
      IF(.NOT.SORL)CALL SHORT
    1 FORMAT (7X,23HTRUCK DATA BASE PRINTED,//)

      RETURN
      END
      SUBROUTINE SHORT
C     THIS SUBROUTINE PRINTS
C     1)  THE NUMBER OF TRUCKS ENTRIES IN AN EXISTING TRUCK
C         DATA BASE.
C     2)  PRINTS ALL TRUCK NAMES AND THE NUMBER OF WHEELS EACH
C         TRUCK HAS.

      COMMON NTRUCK,MESS(19),NINP,NOUT,NT9

      REWIND NT9
      READ(NT9)NTRUCK
C     PRINT OUT HEADING FOR DATA BASE INFO.  ALSC USER INPUTTED
C     MESSAGE AND NO. OF TRUCKS.
      WRITE(NOUT,2)MESS,NTRUCK
      NT = NTRUCK / 2
      DO  3 I = 1,NT
```

```
C       NOW ITERATIVELY READ (TRUCK NAME AND NO. OF WHEELS CN TRUCK)
C       IN GROUPS OF TWO AND PRINT THEM ON ONE LINE.
        READ(NT9)INAM,JNAM,NWHEEL
        WRITE(NOUT,5)INAM,JNAM,NWHEEL
C       FORWARD TAPE(FILE) TO NEXT TRUCK NAME
        DO   4 J = 1,NWHEEL
      4 READ(NT9)
        READ(NT9)INAM,JNAM,NWHEEL
        WRITE(NOUT,6)INAM,JNAM,NWHEEL
        DO   3 J = 1,NWHEEL
      3 READ(NT9)
C       IF THERE IS AN ODD NUMBER OF TRUCKS, READ AND WRITE INFCRMATION
C       ON LAST TRUCK IN BASE
        IF(NTRUCK.EQ.(NTRUCK/2)*2)RETURN
        READ(NT9)INAM,JNAM,NWHEEL
        WRITE(NOUT,5)INAM,JNAM,NWHEEL
        RETURN
      2 FORMAT(5X,18A4,1A3,//,7X,
       .44HTHE NUMBER OF TRUCKS LISTED IN DATA BASE IS ,1I3,///,
       .5X,10HTRUCK NAME,10X,13HNO. OF WHEELS,10X,10HTRUCK NAME,
       .10X,13HNO. OF WHEELS,//)
      5 FORMAT(6X,2A4,16X,1I2)
      6 FORMAT(1H+,49X,2A4,16X,1I2,/)

        END
        SUBROUTINE LONG
C       THIS SUBROUTINE PRINTS
C       1)  THE NUMBER OF TRUCK ENTRIES IN AN EXISTING TRUCK DATA BASE
C       2)  ALL THE TRUCK NAMES AND ALL INFORMATICN ASSOCIATED WITH
C           EACH TRUCK, WHICH IS
C
C           NO. OF WHEELS
C           NO. OF WHEELS ON THE BACK AXLE
C           ALL WHEEL LOADS AND THEIR CORRESPONDING
C           LOCAL X AND Y COORDINATES

        COMMON NTRUCK,MESS(19),NINP,NOUT,NT9

        REWIND NT9
        READ(NT9)NTRUCK
C       WRITE OUT HEADINGS, USER INPUTTED MESSAGE, NC. CF TRUCKS
        WRITE(NOUT,2)MESS,NTRUCK
C       NOW ITERATIVELY READ AND WRITE TRUCK NAME, NC. CF WHEELS,
C       NO. OF BACK AXLE WHEELS
        DO   8 I = 1,NTRUCK
        READ(NT9)INAM,JNAM,NWHEEL
        WRITE(NOUT,4)INAM,JNAM,NWHEEL
C       ITERATIVELY READ AND WRITE WHEEL LCADS AND LCCATIONS
C       FOR THE I(TH) TRUCK
        DO   3 J = 1,NWHEEL
        READ(NT9)P,X,Y
        IF(J.EQ.1)WRITE(NOUT,5)P,X,Y
      3 IF(J.GT.1)WRITE(NOUT,6)P,X,Y
      8 WRITE(NOUT,7)

      2 FORMAT(5X,18A4,1A3,//,7X,
       .43HTHE NUMBER OF TRUCKS LISTED IN DATA BASE IS,1I3,//,
       .9X,4HNAME,8X,6HNO. OF,6X,5HWHEEL,7X,7HLOCAL-X,
       .6X,7HLOCAL-Y,/,
       .21X,6HWHEELS,6X,4HLOAD,8X,8HLOCATICN,5X,
       .8HLOCATION,/,
       .33X,6H(KIPS),6X,4H(FT),9X,4H(FT),//)
      4 FORMAT(7X,2A4,7X,1I2,11X,1I2)
```

```
5 FORMAT(1H+,33X,1F4.0,10X,1F4.0,9X,1F4.0)
6 FORMAT(34X,1F4.0,10X,1F4.0,9X,1F4.0)
7 FORMAT(/)

  RETURN
  END
```

APPENDIX G

DOCUMENTATION OF SEARCH

APPENDIX G. DOCUMENTATION OF SEARCH

GENERAL

The sole purpose for the post-processor to SLAB49 is to make the results more readily interpreted. The post-processor requires a few basic pieces of information from the pre-processor (girders to be printed/plotted, grid spacing, bridge geometry, etc.) and the raw output from SLAB49 (slab moments, girder moments, and grid displacements). The post-processor then generates printouts of the stress along a girder, top and bottom, and slab top stress along the same set of grid points. If it is desired, a plot of top and bottom girder stresses, top slab stresses, and grid displacements can be generated. The stress is corrected to account for the different section moduli that occur at a composite non-composite interface. The post-processor will, as has already been explained in the section covering the pre-processor, transpose a converted coordinate system so that the user of the TBAP will not have to determine if the coordinate system has been rotated. The final stress calculation carried out by the post-processor is the extrapolation of the stress at an interior support point. This is carried out by using a linear extrapolation procedure from the two closest grid point values on both sides of the supported point. A flow chart for the post-processor (SEARCH) is shown in Fig 16.

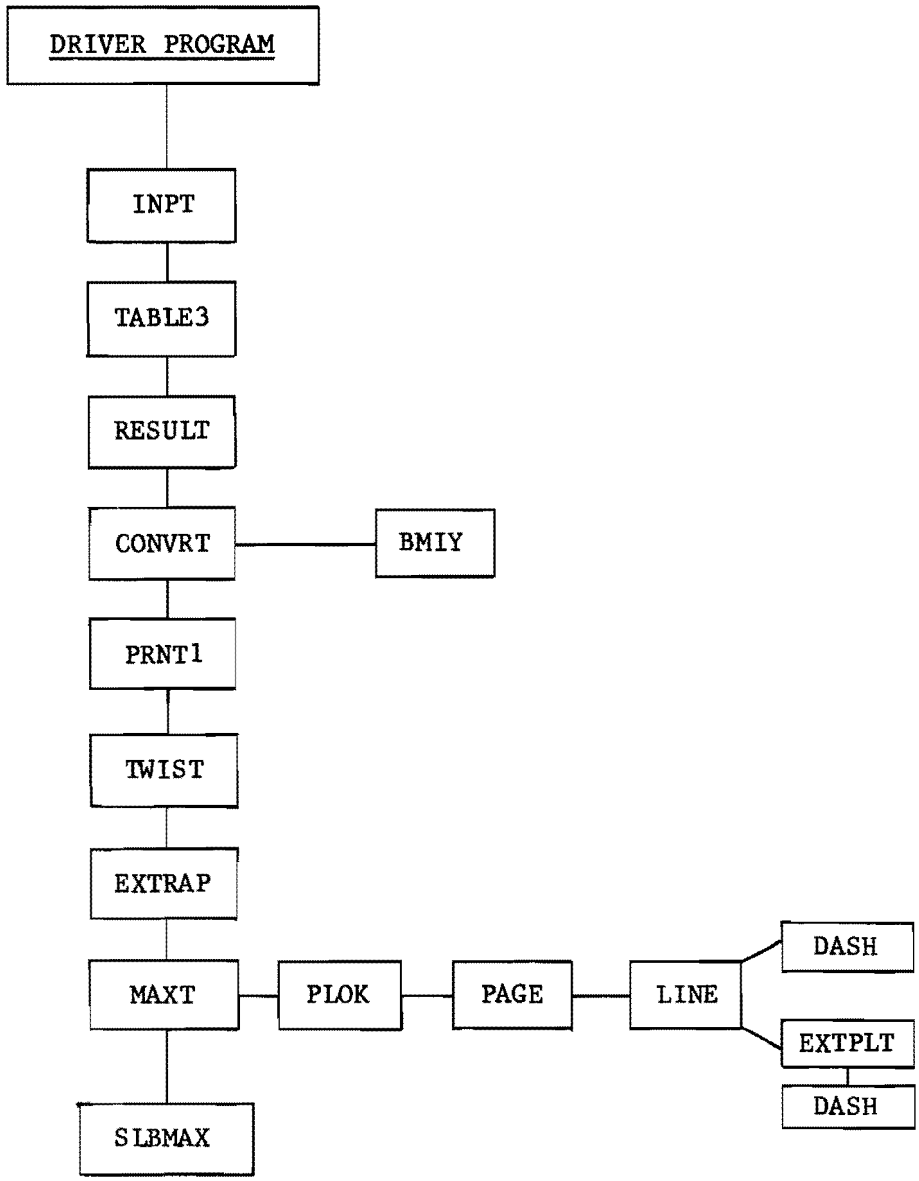


Fig 16. Flow chart for SEARCH.

SEARCH

SUBJECT: INFORMATION TRANSFER AND MANAGEMENT

RELATED SUBROUTINES: 1) INPT

2) TABLE3

3) RESULT

1) SUBROUTINE INPT (NT11)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES AND INPUT ARRAY -

NT11 - FORTRAN unit used to transfer information from pre-processor to post-processor

COMIY - array containing centroids, moments of inertia for composite sections, plus the centroid of the beam x-section

IGRDL - array containing the girder/slab deadweights as computed in TAB203

BRIG1 - array containing span lengths and locations of composite sections

BRIG2 - array containing (HX, HY, TK, TURN, BEAMI, BEAMD, EC, ES, TOPS, BOTS)

IBRG2 - array containing (IGRTP, INAM, LRD, NGRDR, NCOMP, IDIV, JDIV, JNAM, NSPON, NPLOT, NDAT, NDUT)

CURHW - array containing (CURHT, CUWD, HAUNCH)

NAMS - array containing names of girders to be plotted and/or printed in post-processor

YDAT - array containing longitudinal locations of starting and ending positions of girders to be plotted and/or printed in post-processor

NFORM - informational data to be used in post-processor

IYNUMB9 - array containing the locations and number of general output areas to be printed in post-processor

BRIG3 - array containing information defining the skew of the bridge

PURPOSE: The purpose of INPT is to retrieve and interpret information passed to the post-processor from the pre-processor. This is done by the use of a single unformatted read statement. After execution of the read statement is complete all the transferred information is stored in its correct position in the various common blocks. Those common blocks are then used throughout the program. The information is transferred to the post-processor using FORTRAN Unit 11.

2) SUBROUTINE TABLE3

CALLING ROUTINE - Main Program

Required Routines - None

PURPOSE: The only function performed by TABLE3 is to determine the limits of the composite sections and locate supported points. This is accomplished by reading through TABLE3 of the SLAB49 input. Blank cards are used to read lines of no interest.

3) SUBROUTINE RESULTS (IX, NPUT, IRED, SLBPRN, ISLBP, JSLBP)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

IX - no. of grid points in the longitudinal direction (i.e., IDI + 1)

NPUT - echo print option of SLAB49 output

IRED - control option to suppress girder reading if no girders are present in model

SLBPRN - principal slab stress

ISLBP - longitudinal grid location of principal slab stress

JSLBP - transverse grid location of principal slab stress

PURPOSE: RESULTS reads all output from SLAB49 that is vital to the post-processor. This includes slab moments and deflections, principal slab stress, and beam moments. As each of these results is read into the post-processor it is stored in an array. The x-slab moment is stored in SLM (, , 1), the y-slab moment in SLM (, , 2). The X and Y beam moments are stored similarly in BM (, , 1) and BM (, , 2). The deflection of each grid point is stored in DEF (,). These results are then echo printed as the user desires to see their values.

SEARCH

SUBJECT: RESULT MODIFICATION FOR EASY INTERPRETATION

RELATED SUBROUTINES:

1) CONVRT
2) BMIY
3) PRNT1
4) TWIST
5) MAXT
6) SLBMAX
7) EXTRAP

1) SUBROUTINE CONVRT (IX, IY)

CALLING ROUTINE - Main Program

Required Routines - BMIY

DEFINITION OF INPUT VARIABLES -

IX - no. of grid points in the longitudinal direction (i.e., IDIV + 1)

IY - no. of grid points in the transverse direction (i.e., JDIV + 1)

PURPOSE: CONVRT uses the method described in "finite difference models of orthogonal bridge slabs," Ref 1, to calculate the stress in the slab and girders. CONVRT calls subroutine BMIY to calculate the conversion factors for moment-to-stress calculations. The final product of this subroutine is the calculation of the stress in the top of the slab, the top of the girder, and the bottom of the girder parallel to the longitudinal direction of the span.

2) SUBROUTINE BMIY (ISTA, JSTA, BMI, YBAR, YT, CURYB, IYES)

CALLING ROUTINE - CONVRT

Required Routines - None

DEFINITION OF INPUT VARIABLES -

- ISTA - longitudinal location of grid point at which moment of inertia and centroid are to be determined
- JSTA - transverse location of grid point at which moment of inertia and centroid are to be determined
- BMI - girder moment of inertia
- YBAR - centroid of girder x-section
- YT - distance from top of slab to centroid of composite section
- CURYB - distance from top of integral curb to centroid of integral curb/slab x-section
- IYES - control variable to determine if a grid point is in a composite section or not

PURPOSE: BMIY will generate values used in the conversion of moments to stresses in subroutine CONVRT. The necessary values are x-section moment of inertia at the grid point under analysis and the distance from the centroid of the x-section to the point where the stress is desired. The routine takes composite action into account when calculating x-section moments of inertia, i.e., slab and girder in composite zone, girder only in a noncomposite zone.

3) SUBROUTINE PRNT1 (IX, IY, IGIRDL)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

IX - no. of grid points in the longitudinal direction (i.e., IDIV + 1)

IY - no. of grid points in the transverse direction (i.e., JDIV + 1)

IGIRDL - array containing transverse locations of girders to be printed

PURPOSE: All resultant stresses calculated in CONVRT are printed in PRNT1. The stress at a given point is printed on a single line in the following order: x-slab stress, y-slab stress, deflection, girder top stress, and girder bottom stress. Each line is headed by the longitudinal and transverse grid location being printed. The default is that all girder points in the longitudinal direction are printed for each girder. If no girders are present all grid points are printed.

4) SUBROUTINE TWIST (IDIV)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

IDIV - no. of grid divisions in the longitudinal direction

PURPOSE: TWIST allows for the results from SLAB49 to be transposed when the bridge has more divisions in the transverse direction than it does in the longitudinal direction. This allows the user, when only the results of the pre- and post-processor are viewed, to easily interpret and correlate data from each program. This transposing of results is accomplished with two major steps. First, SLM (, , 1) and SLM (, , 2) are interchanged; second, the resulting matrix, SBD (50, 50, 5) is transposed while the third delimeter (5) is held constant.

5) SUBROUTINE MAXT

CALLING ROUTINE - Main Program

Required Routines - PLOK

PURPOSE: MAXT does a comparison of all stresses occurring along a line of grid points that coincide with a single girder. Once the maximum value of slab stress, girder top stress, girder bottom stress, and deflection are determined, they are stored in the array JMAX. These girder and slab maximums are then used in scaling the stress results when a CAL-COMP plot is generated for the specified girder. The maximum physical dimension of any value is defined as 1.5 inches; this prevents plots from overlapping. The routine also determines the location at which the maximum occurs.

6) SUBROUTINE SLBMAX (IDIV, JDIV, TURN)

CALLING ROUTINE - Main Program

Required Routines - None

DEFINITION OF INPUT VARIABLES -

- IDIV - no. of grid divisions in the longitudinal direction
- JDIV - no. of grid divisions in the transverse direction
- TURN - control variable to define the orientation of the X-Y axis system

PURPOSE: SLBMAX serves much the same purpose as MAXT. Its sole purpose is to locate the maximum transverse slab stress occurring in the specified output areas. (The default is the entire bridge area.)

7) SUBROUTINE EXTRAP

CALLING ROUTINE - Main Program

Required Routines - None

PURPOSE: The purpose of EXTRAP is to give the analyst a general feeling for the value of stress at a support. This becomes necessary when a support point falls between two grid points (stress in the model is only calculated at the grid points). A forward and backward two-point linear extrapolation is used. If the two methods give similar results the analyst can be fairly comfortable with the predicted values; if drastically differing results occur, engineering judgment must be used.

SEARCH

SUBJECT: PLOTTING RESULTS

REQUIRED ROUTINES:
1) PLOK
2) PAGE
3) LINE
4) EXTPLT
5) DASH

1) SUBROUTINE PLOK (IG, ICONT, IX1)

CALLING ROUTINE - MAXT

Required Routines - CAL-COMP Plot Routines
PAGE

DEFINITION OF INPUT VARIABLES -

IG - identity (number) of girder being plotted

ICONT - control variable to designate the first girder being plotted for which basic scaling information must be calculated

IX1 - no. of grid points along the girder being plotted

PURPOSE: PLOK controls all the plotting that is carried out by the post-processor. Several operations other than control are performed by PLOK. They are (1) calculation of the ratio of composite to noncomposite x-section moments of inertia. This is necessary when calculating the instantaneous change in stress that occurs with an instantaneous change in stiffness. (2) The offset of the girder in the longitudinal direction due to the skew of the bridge is calculated. This is necessary if only the part of the rectangular grid that actually contains the bridge spans is to be plotted. (3) The limits of composite action are identified. PLOK then calls PAGE which sets up the plot.

2) SUBROUTINE PAGE (ICHK, DIS, SCY, IA, IB, IG, IGR, IX1, RAT)

CALLING ROUTINE - PLOK

Required Routines - CAL-COMP Plot Routines
LINE

DEFINITION OF INPUT VARIABLES -

ICHK - control variable to define the page being plotted, first page or second page

DIS - skew offset at transverse location of girder being plotted

SCY - scaling factor in the longitudinal direction

IA, IB - control variables to determine what result is being plotted

IG - identity (number) of girder being plotted

IGR - matrix identity of girder being plotted (IG + 1)

IX1 - no. of grid points along the girder being plotted

RAT - ratio of composite to noncomposite moments of inertia, used to calculate magnitude of stress jumps at composite-noncomposite interfaces

PURPOSE: PAGE does the bulk of the plotting that is done by the post-processor. In generating a plot the following steps are executed. The identifying information concerning the problem is printed: this inclues girder number, problem number, plot titles, and grid labels. At the end of PAGE's execution the only information not plotted are the actual stress and displacement values, and the extrapolated values of stress occurring at the support points. These are done in LINE and EXTPLT respectively.

3) SUBROUTINE LINE (XPAGE, DIS, SCY, IG, IGR, IPLT, IXX, RAT)

CALLING ROUTINE - PAGE

Required Routines - CAL-COMP Plot Routines
DASH
EXTPLT

DEFINITION OF INPUT VARIABLES -

XPAGE - actual distance (inches) moving perpendicular to the line being plotted to the side of the page

DIS - skew offset distance at transverse location of girder being plotted

SCY - scaling factor in the longitudinal direction

IG - identity (number) of girder being plotted

IGR - matrix identity of girder being plotted

IPLT - control variable to determine what result is being plotted

IXX - no. of grid points along the girder being plotted

RAT - ratio of composite to noncomposite moments of inertia, used to calculate magnitude of stress jumps at composite-noncomposite interfaces

PURPOSE: LINE plots the actual values of slab stress, girder top stress, girder bottom stress, and displacement that are found with subroutine RESULT. It uses the composite-noncomposite ratios calculated in PLOK to show an instantaneous change in stress at the composite interfaces. This has two purposes: (1) it allows the designer to see if his choice of composite areas was accurate, and (2) it allows a more accurate interpolation of stress between grid points in the area of a composite interface.

4) SUBROUTINE EXTPLT (IPLT, IG, IGR, HY, SCY, SCX, NSPAN)

CALLING ROUTINE - LINE

Required Routines - DASH

DEFINITION OF INPUT VARIABLES -

IPLT - control variable to determine which result is being plotted

IG - identity (number) of girder being plotted

IGR - matrix identity of girder being plotted

HY - spacing of grid points in the transverse direction

SCY - scaling factor in the longitudinal direction

SCX - scaling factor for the result being plotted

NSPAN - no. of spans comprising bridge

PURPOSE: EXTPLT plots the extrapolated values of the stress occurring at the supported points. A dashed line is used to connect the closest point on either side of the actual support and the value predicted at the support when a linear extrapolation is used. While this stress will not be exact, it will bring the uncertainty of the stress result to the attention of the analyst.

5) SUBROUTINE DASH (X1, Y1, X2, Y2)

CALLING ROUTINE - LINE
EXTPLT

Required Routines - CAL-COMP Plot Routines

DEFINITION OF INPUT VARIABLES -

X1 - scaled value of result at beginning of dashed line

Y1 - scaled value of longitudinal location at beginning of dashed line

X2 - scaled value of result at end of dashed line

Y2 - scaled value of longitudinal location at end of dashed line

PURPOSE: DASH is used to connect two points on the page by a dashed line. The actual number of dashes depends on the length of the line--it may vary anywhere from 5 to 11 dashes per line.

APPENDIX H

FORTRAN LISTING OF POST-PROCESSOR - SEARCH

```
C      PROGRAM SEARCH (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C     *                    TAPE11,PLTFILE)
       COMMON  /RESL/ SLM(50,50,2),BM(50,50,2),DEF(50,50)
       COMMON  /RD   / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
      C               ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCCMP,IDIV,JDIV
      C               ,JNAM,NSPAN,NPLCT,NDAT,NPUT,NAMS(50),YDAT(50,2)
      C               ,CUHT,CUWD,HAUNCH
       COMMON  /C    / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
      C               ,JDIST1(12,50),JDIST2(12,50),MCRT(50,2),JDD(12,2)
      C               ,LART(50,11)
       COMMON  /INF / INFO(58),INFO3,INF3
       COMMON  /UNIT/ NINP,NOUT
       COMMON  /E    / IY1(20),IY2(20),NUM89,IDUM9
       COMMON  /EXTR/ VVAX(50,3),AXX(50,3),XML(50,11,3),XMR(50,11,3)
       COMMON  /SKEW/ THETA,ANGLE,ARM,LD,NDUM
       COMMON  /PLTN/ LOC(50,4),VMAX(4),NVAL(4),IBUF(1000),NPLT,NDUB
       DIMENSION       XL(6)
       DATA            ISTE,IPRS,NAME,NAME2/4HSTEE,4HPRES,4HDEFL,4HE 3./
      C               ,NOBM/4HSLAB/
C-----ISTE NOT REFERENCED - SET TO DUMMY TO AVOID DIAGNOSTIC - JJP       100C80
               IDUMY = ISTE                                              100C80
C--
C--    SET TAPE UNIT NUMBERS
       NINP     = 5
       NOUT     = 6
       NT11     = 11
       NPLT     = 71
C--    INITIATE PROBLEM COUNTER
       LD       = 0
C--    READ NECESSARY INPUT FROM PREPROCESSOR
     5 CALL INPT (NT11)
C--    LOCATE TABLE 3 IN SLAB49 OUTPUT
    10 READ (NINP,620) NESL
       IF (NESL.EQ.NAME2) GOTO 15
       GOTO 10
    15 IF (LD.EQ.0) CALL TABLE3
C--    PRINT PAGE TRIM
       WRITE (NOUT,601)
C--    PRINT JOB AND PROBLEM INFO
       IF (LD.EQ.0) WRITE (NOUT,602) (INFO(I),I = 1,40)
       WRITE (NOUT,603) INF3,INFO3,(INFO(I),I = 41,58)
C--    PRINT OPTIONS
       WRITE (NOUT,606) LRD,NPLOT,NDAT,NPLT
C--    COUNT NUMBER OF PROBLEMS
       LD       = LD + 1
       TIK      = TK * 12.0
       EC1      = EC * 1000.0 / 144.0
       ES1      = ES * 1000.0 / 144.0
C--    COMPUTE TRANS. AND LONG. ARRAY CONTROLS
       IX       = IDIV + 1
       IY       = JDIV + 1
       IF (TURN.GT.0.0) GOTO 20
       IX       = JDIV + 1
       IY       = IDIV + 1
C--    PRINT CONSTANTS TABLE
    20 WRITE (NOUT,604) IDIV,JDIV,HX,HY,TIK
       IF (IGRTP.NE.NOBM) GOTO 25
       WRITE (NOUT,612)
       GOTO 30
    25 WRITE (NOUT,611) NGIRDR,IGRTP,INAM,JNAM,BEAMD,CENT,BEAMI,
      C                 CYBAR(1),COMPI(1),EC1,ES1
    30 CONTINUE
C--    PRINT WARNING FOR LIMITED OUTPUTS
```

```
      IPR      = IY - 1
      IF ((IY1(1).EQ.0).AND.(IY2(1).EQ.IPR)) GOTC 40
      WRITE (NOUT,607)
      DO 35 I = 1,NUMB9
      BEGN     = FLOAT(IY1(I)) * HY
      EN2D     = FLOAT(IY2(I)) * HY
   35 WRITE (NOUT,608) BEGN,EN2D
      WRITE (NOUT,609)
   40 CONTINUE
C--   LOCATE RESULTS
   45 READ (NINP,619) NESL
      IF (NESL.EQ.NAME) GOTO 50
      GOTO 45
C--   SET VARIABLE FOR SLAB ONLY OUTPUT
   50 IRED     = 0
      IF ((IGRTP.NE.NOBM).OR.(CUHT.GT.0.001)) IRED = 1
C--   CALL ROUTINE TO READ AND ECHO PRINT MOMENTS
      CALL RESULT (IX,NPUT,IRED,SLBPRN,ISLBP,JSLBP)
C--   CALL ROUTINE TO CONVERT MOMENTS TO STRESSES
      CALL CONVRT (IX,IY)
C--   CALL ROUTINE TO PRINT RESULTS
      CALL PRNT1 (IX,IY,IGIRDL)
C--   CALL ROUTINE TO REORDER ARRAYS
      IF (TURN.LT.0.0)  CALL TWIST (IX,IY)
C--   CALL ROUTINE TO EXTRAPOLATE VALUES AT INTERICR SUPPORTS
      IF (NSPAN.GT.1)   CALL EXTRAP
C--   PRINT NOTE AND PAGE SPACE
      WRITE (NOUT,601)
      WRITE (NOUT,614)
C--   CALL ROUTINE TO FIND GIRDER MAXIMUMS. THIS ROUTINE CALLS
C--   THE PLOTTING ROUTINES FOR DESIRED CUTPUTS
      IF (IGRTP.NE.NOBM) CALL MAXT
C--   CALL ROUTINE TO FIND TRANSVERSE SLAB MAXIMUM
      CALL SLBMAX (IDIV,JDIV,TURN)
C--   PRINT PRINCIPLE SLAB STRESSES
      SLBPRN   = - SLBPRN * 1000.0 / 144.C
      WRITE (NOUT,613) ISLBP,JSLBP,SLBPRN
C--   PRINT EXTRAPOLATED VALUES
      IF (NSPAN.EQ.1) GOTO 65
      WRITE (NOUT,601)
      WRITE (NOUT,615)
      NSPN     = NSPAN - 1
      DO 60 M = 1,NSPN
      WRITE (NOUT,616)  M
      DO 60 II=1,NGIRDR
C--   CONVERT STRESSES TO PSI FOR PRINT CUT
      DO 55 I = 1,3
      I2       = I + 3
      XL (I)   = XML (II,M,I) * 1000.0 / 144.0
      XL (I2)  = XMR (II,M,I) * 1000.0 / 144.0
C--   CHECK FOR SLAB TYPE AND SET GIRDER VALUES TC ZERC
      IF (IGRTP.NE.NOBM) GOTO 55
      IF (I.EQ.1)        GOTO 55
      XL (I)   = 0
      XL (I2)  = 0
   55 CONTINUE
      WRITE (NOUT,617) II,(XL(I),I=1,3)
   60 WRITE (NOUT,618)    (XL(I),I=4,6)
   65 CONTINUE
C--   PRINT PRESTRESSING VALUES
      IF (IGRTP.EQ.IPRS) WRITE (NOUT,610) TOPS,BOTS
      IF (LD.LT.LRD) GOTO 5
C--
```

```
  601 FORMAT (5H1       ,80X,10HI----TRIM  ,// )
  602 FORMAT (5X,20A4,/,5X,20A4)
  603 FORMAT (//,5X,7HPROBLEM  ,/,6X,2A4,2X,17A4,1A2)
  604 FORMAT (//,5X,9HCONSTANTS  /
     C  /,8X,45HNUMBER OF INCREMENTS ACRCSS BRIDGE              ,I10,
     C  /,8X,45HNUMBER OF INCREMENTS ALONG BRIDGE               ,I10,
     C  /,8X,45HINCREMENT LENGTH ACROSS BRIDGE          (FT)    ,F10.4,
     C  /,8X,45HINCREMENT LENGTH ALONG BRIDGE           (FT)    ,F10.4,
     C  /,8X,45HSLAB THICKNESS                          (IN)    ,F10.4)
  606 FORMAT (//,5X,12HCONTROL DATA  ,/,
     C  /,8X,45HNUMBER OF PROBLEMS INPUT (LOAD CASES)           ,5X,I5,
     C  /,8X,45HNUMBER OF GIRDER OUTPUTS                        ,5X,I5,
     C  /,8X,45HDATA OUTPUT (1 = DATA ONLY,2 = DATA+PLOT)       ,5X,I5,
     C  /,8X,45HPRINT SLAB49 RESULTS (1 = YES)                  ,5X,I5 )
  607 FORMAT ( //,5X,18HLIMITS OF RESULTS  ,/,
     C  /,8X,45HALL RESULTS AND MAXIMUM STRESSES ARE LIMITED
     C  /,8X,45HTO THE SPECIFIED GENERAL OUTPUT AREAS.
     C //,8X,45HSTARTING LOCATION       ENDING LCCATICN
     C  /,8X,45H      (FT)                     (FT)             ,/)
  608 FORMAT (9X,F10.3,13X,F10.3)
  609 FORMAT ( /,8X,12HWARNING
     C       ,32HMAXIMUM STRESSES ARE LIMITED TO
     C  /,8X,45H THE AREAS ABOVE AND MAY NOT INCLUDE THE
     C  /,8X,45H LARGEST STRESSES THAT OCCUR FCR THIS
     C  /,8X,25H LOAD CASE.                                    )
  610 FORMAT (//5X,40HMIDSPAN STRESSES INDUCED BY PRESTRESSING
     C        23H AND GIRDER DEAD WEIGHT
     C  //8X,47HTHESE STRESSES ARE NCT INCLUDED IN THE MAXIMUMS
     C   /8X,47HTHAT ARE COMPUTED ABOVE AND MUST BE ADDED INTO
     C   /8X,40HTHE SOLUTION
     C  //8X,45HSTRESS AT GIRDER TOP                   (PSI)      ,F10.2,
     C   /8X,45HSTRESS AT GIRDER BOTTOM                (PSI)      ,F10.2)
  611 FORMAT (
     C     8X,45HNUMBER OF GIRDERS IN BRIDGE                      ,I10,
     C  /,8X,45HGIRDER TYPE                                       ,1X,A4,
     C  /,8X,45HGIRDER NAME                                       ,1X,2A4,
     C  /,8X,45HGIRDER - TOTAL DEPTH                     (FT)    ,F10.4,
     C  /,8X,45HGIRDER - CENTROID TO BOTTOM              (FT)    ,F10.4,
     C  /,8X,45HGIRDER - MOMENT OF INERTIA             (FT-4)    ,F10.4,
     C  /,8X,45HCOMPOSITE SECT - CENTRCID TO BCTTCM      (FT)    ,F10.4,
     C  /,8X,45HCOMPOSITE SECT - MOMENT CF INERTIA     (FT-4)    ,F10.4,
C    C  /,8X,45HMODULUS OF ELASTICITY - CONCRETE        (PSI)    ,E10.3 ,       CDC
     C  /,8X,45HMODULUS OF ELASTICITY - CONCRETE        (PSI)    ,1PE10.3,      IBM
C    C  /,8X,45HMODULUS OF ELASTICITY - STEEL           (PSI)    ,E10.3 )       CDC
     C  /,8X,45HMODULUS OF ELASTICITY - STEEL           (PSI)    ,1PE10.3 )     IBM
  612 FORMAT (//8X,37HTHIS IS A SLAB BRIDGE WITH NC GIRDERS )
  613 FORMAT (///,5X,31HPRINCIPAL SLAB STRESS MAXIMUM   ,//,
     C         5X,4H   X,2X,4H   Y ,2X,8H  STRESS  ,/,19X,5H(PSI),//,
     C         3X, 2(2X,I4),2X,F11.2,/)
  614 FORMAT (/,5X,25HMAXIMUM STRESSES CCMPUTED   ,//)
  615 FORMAT (////,5X,44HEXTRAPOLATED STRESSES FOR INTERICR SUPPORTS )
  616 FORMAT ( /,7X,25HINTERIOR SUPPORT NUMBER - ,I4,
     C  //,7X,6HGIRDER,3X8HSLAB TOP,5X10HGIRDER TOP,3X13HGIRDER BOTTOM
     C   /,9X,2HNO,5X,7HFORWARD,7X,7HFORWARD,7X,7HFORWARD
     C   /,16X,8HBACKWARD,6X,8HBACKWARD,6X,8HBACKWARD,
     C   /,17X,5H(PSI)  ,10X,5H(PSI)  ,10X,5H(PSI)    )
  617 FORMAT ( / , 8X , I3 , 3F14.3 )
  618 FORMAT (     11X   ,    3F14.3 )
  619 FORMAT (15X,A4)
  620 FORMAT ( 9X,A4 )
C--
      STOP
      END
```

```
      SUBROUTINE INPT (NT11)
      COMMON  /RD  / COMIY(5),BRIG2(10),IBRG2(12),NAMS(50),YDAT(50,2)
     C               ,CURHW(3)
      COMMON  /C   / IGIRDL(50),BRIG1(36),ITR1(1300),JCD(12,2)
     C               ,LART(50,11)
      COMMON  /INF / NFORM(60)
      COMMON  /RESL/ SB(12500)
      COMMON  /E   / IYNUM9(42)
      COMMON  /SKEW/ BRIG3(3),LD,NDUM
C--
C--   THIS ROUTINE READS DATA FROM TAPE PRODUCED
C--   BY THE PREPROCESSOR
C--
      READ (NT11) COMIY,IGIRDL,BRIG1,BRIG2,IBRG2,CURHW,
     C            NAMS,YDAT,NFORM,IYNUM9,BRIG3
C--
C--   SET RESULTS ARRAYS EQUAL TO ZERO
C--
      DO 1  I = 1,12500
    1 SB (I)  = 0.0
C--
      RETURN
      END
      SUBROUTINE TABLE3
      COMMON  /RD  / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C               ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
     C               ,JNAM,NSPAN,NPLOT,NCAT,NPUT,NAMS(50),YDAT(50,2)
     C               ,CUHT,CUWD,HAUNCH
      COMMON  /C   / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
     C               ,JDIST1(12,50),JDIST2(12,50),MORT(50,2),JDD(12,2)
     C               ,LART(50,11)
      COMMON  /UNIT/ NINP,NOUT
      DATA           NOBM/4HSLAB/,IPAN/4HREIN/
C--
C--   THIS ROUTINE FINDS THE LIMITS OF COMPOSITE SECTIONS AND
C--   LOCATION OF SUPPORTS FROM TABLE 3 FROM SLAB49
C--
      CURBA   = CUHT * CUWD
                          MM = 1
      IF (CURBA.LT.0.001) MM = 0
                          N1 = NGIRDR
      IF (IGRTP.EQ.NOBM)  N1 = 0
      ICT     = 2 * (IDIV + 1 + N1) + 4
C--   READ JUNK LINES
      DO 5 I  = 1,ICT
    5 READ (NINP,501)
      IF (IGRTP.EQ.IPAN.OR.IGRTP.EQ.NOBM) GOTO 15
      IF (NCOMP.EQ.0) GOTO 15
C--   READ COMPOSITE SECTION LIMITS
      DO 10 I = 1,NCOMP
      DO 10 J = 1,NGIRDR
      READ (NINP,502) JDIST1(I,J),JDIST2(I,J)
      READ (NINP,501)
   10 CONTINUE
C--   READ JUNK LINES
   15 IF (MM.EQ.0) GOTO 25
      DO 20 I = 1,4
   20 READ (NINP,503) N01
   25 CONTINUE
      DO 30 I = 1,NGIRDR
C--   READ START OF BRIDGE STATIONS
      READ (NINP,503) MORT(I,1)
   30 READ (NINP,501)
```

```
      DO 35 J = 1,NSPAN
      DO 35 I = 1,NGIRDR
C--   READ INTERIOR SUPPORT STATIONS
      READ (NINP,504) LART(I,J)
C--   READ END OF BRIDGE STATIONS
   35 READ (NINP,504) MORT(I,2)
C--
  501 FORMAT (   )
  502 FORMAT (9X,I3,4X,I3)
  503 FORMAT (9X,I3)
  504 FORMAT (16X,I3)
C--
      RETURN
      END
      SUBROUTINE RESULT (IX,NPUT,IRED,SLBPRN,ISLBP,JSLBP)
      COMMON  /UNIT/ NINP,NOUT
      COMMON  /RESL/ SLM(50,50,2),BM(50,50,2),DEF(50,50)
      COMMON  /E    / IY1(20),IY2(20),NUMB9,IDUM9
C-----THE VARIABLE FOR PRINTING REACTIONS NEEDED DIMENSIONING           300C80
      DIMENSION   SREACT(50,50)                                         300C80
C--
C--   THIS ROUTINE READS RESULTS AND STORES THEM
C--   IN THE APPROPRIATE ARRAYS
C--   AT THIS TIME PRINCIPLE STRESS MAXIMUM IS FOUND
C--
      SLBP      = 0.0
      DO 20  L = 1,NUMB9
C--   SET LOCATION FROM TABLE 4 PREPROCESSOR OPTION
      IYS       = IY1 (L) + 1
      IYE       = IY2 (L) + 1
      DO 10  J = IYS,IYE
      READ (NINP,501) TRASH
      DO 10  I = 1,IX
C--   SET GRID LOCATION
      ISTA      =  I - 1
      JSTA      = IYE - J
C--   READ SLAB MOMENTS AND DEFLECTIONS
      READ (NINP,502) DEF(ISTA+1,JSTA+1),(SLM(ISTA+1,JSTA+1,K),K=1,2)
C    C                ,SLBIT,SREACT                                     NULL
     1                ,SLBIT,SREACT(ISTA+1,JSTA+1)                      300C80
C--   CHECK PRINCIPLE STRESS MAXIMUM FROM SLAB49 OUTPUT
      SLB2      = ABS (SLBIT)
      IF (SLB2.LT.SLBP) GOTO 5
      SLBPRN    = SLBIT
      SLBP      = ABS (SLBPRN)
      ISLBP     = ISTA
      JSLBP     = JSTA
    5 CONTINUE
      IF (IRED.NE.1) GOTO 10
C--   READ BEAM MOMENTS
      READ (NINP,503) (BM(ISTA+1,JSTA+1,K),K = 1,2)
   10 CONTINUE
C--   ECHO PRINT RESULTANT MOMENTS
      IF (NPUT.LT.1) GOTO 20
C--   PRINT TITLES
      WRITE (NOUT,604)
      WRITE (NOUT,601)
      DO 15  J = IYS,IYE
      WRITE (NOUT,602)
      DO 15  I = 1,IX
C--   SET ARRAY STATIONS
      ISTA      = I-1
      IST       = I
```

```
      JSTA      = IYE - J
      JST       = JSTA + 1
C--   ECHO PRINT MOMENTS
      WRITE (NOUT,603) ISTA,JSTA,DEF(IST,JST),(SLM(IST,JST,K),K = 1,2)
C    C        ,SREACT,(BM(IST,JST,K),K = 1,2)                            NULL
     C        ,SREACT(IST,JST),(BM(IST,JST,K),K = 1,2)                   300C80
   15 CONTINUE
   20 CONTINUE
C--
  501 FORMAT ( F5.0 )
  502 FORMAT ( 11X,3E11.3,11X,E11.3,6X,E11.3 )
  503 FORMAT ( 22X,2E11.3 )
C 601 FORMAT (//,5X,17HINPUT FROM SLAB49   ,/,                           NULL
C    C /,7X,33HRESULTANT MOMENTS AND DEFLECTIONS                         NULL
C    C /,7X,18HOUTPUT FROM SLAB49                                        NULL
C    C //7X,45HUNITS FOR THESE MOMENTS AND DEFLECTICNS ARE               NULL
C    C /,7X,20HFEET AND KIPS.                                            NULL
  601 FORMAT (    35H      OPTIONAL OUTPUT FROM SLAB49                   300C80
     1  /  5X,    50H  RESULTANT DEFLECTICNS, MOMENTS AND REACTIONS      300C80
     2  // 5X,    45H  UNITS FOR THESE RESULTS ARE FEET AND KIPS         300C80
     C //,33X,6HSLAB X,7X,6HSLAB Y,/33X,6HMOMENT,7X,6HMOMENT,/,
     C    33X,6HBEAM X,7X,6HBEAM Y,6X,7HSUPPORT,/,7X,4H  X  ,4H  Y  ,2X,
     C    11HDEFLECTIONS,5X,6HMOMENT,7X,6HMOMENT,6X,8HREACTICN )
  602 FORMAT ( / )
C 603 FORMAT (7X,2I4,4(2X,   1E11.3),/,28X,2(2X,   1E11.3) )                CDC
  603 FORMAT (7X,2I4,4(2X,1P1E11.3),/,28X,2(2X,1P1E11.3) )                  IBM
  604 FORMAT (5H1      ,80X,10HI----TRIM  ,// )
C--
      RETURN
      END
      SUBROUTINE CONVRT (IX,IY)
      COMMON  /RESL/ SLM(50,50,2),BM(50,50,2),DEF(50,50)
      COMMON  /RD  / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C               ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
     C               ,JNAM,NSPAN,NPLOT,NCAT,NPUT,NAMS(50),YDAT(50,2)
     C               ,CUHT,CUWD,HAUNCH
      COMMON  /C   / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
     C               ,JDIST1(12,50),JDIST2(12,50),MCRT(50,2),JDD(12,2)
     C               ,LART(50,11)
      LOGICAL        IEDGE
      DATA           NOBM/4HSLAB/
C--
C--   THIS ROUTINE CONVERTS MOMENTS OUTPUT FROM SLAB49
C--   TO STRESSES AND STORES THEM IN THE ARRAYS
C--
C--   CHOOSE CORRECT RESULTS ARRAY FCR MCMENTS ALCNG GIRDER
                     KCON = 2
      IF (TURN.LT.0.0) KCON = 1
      DO 25 J     = 1,IY
      L           = 1
      DO 25    I = 1,IX
C--   SET GRID LOCATION
      ISTA        = I - 1
      JSTA        = J - 1
                     IB = ISTA
      IF (TURN.LT.0.0) IB = JSTA
      IEDGE       = IB.EQ.0.OR.IB.EQ.IDIV
C--   SET BEAM MOMENT TO BE CONVERTED
                     BMM = BM(I,J,KCCN)
      IF (IGRTP.EQ.NOBM) BMM = 0.0
C--   CALL ROUTINE TO SET VALUES FOR STRESS CCNVERSICN
      CALL BMIY (ISTA,JSTA,ERTIA,YBAR,YT,YCUR,IYES)
      Y           = YBAR - BEAMD
```

```
      DO 5      K = 1,2
C--   COMPUTE SLAB STRESSES AT TOP
                SLM(I,J,K) = -SLM(I,J,K) * 6.0 / (TK**2)
      IF (IEDGE) SLM(I,J,K) =  SLM(I,J,K) * 2.0
C--   COMPUTE BEAM STRESSES TOP AND BOTTCM
      BM(I,J,K)   = BMM*Y/ERTIA
      Y           = YBAR
    5 CONTINUE
      SL          = SLM (I,J,KCON)
      IF (IYES.NE.1)       GOTO 15
C--   COMPUTE SLAB TOP STRESSES AT THE GIRDERS IN COMPCSITE ZCNE
      IF (IGRTP.EQ.NOBM)   GOTO 15
      IF (IB.NE.IGIRDL(L)) GOTO 15
      SL          = SL * 2.0 * YT / TK
      L           = L + 1
      GOTO 20
C--   COMPUTE SLAB STRESS AT INTEGRAL CURB
   15 IF (IEDGE)  SL = SL * 2.0 *YCUR / TK
   20 SLM (I,J,KCON) = SL
   25 CONTINUE
C--
      RETURN
      END
      SUBROUTINE BMIY (ISTA,JSTA,BMI,YBAR,YT,CURYB,IYES)
      COMMON  /RD  / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C               ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCCMP,IDIV,JDIV
     C               ,JNAM,NSPAN,NPLCT,NCAT,NPUT,NAMS(50),YDAT(50,2)
     C               ,CUHT,CUWD,HAUNCH
      COMMON  /C   / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
     C               ,JDIST1(12,50),JDIST2(12,50),MCRT(50,2),JDD(12,2)
     C               ,LART(50,11)
      DATA           IPAN/4HREIN/,NOBM/4HSLAB/
C--
C--   THIS ROUTINE SETS THE VALUE FOR THE MCMENT CF INERTIA AND Y-BAR
C--   USED TO CALCULATE BEAM AND SLAB STRESSES IN SUBRCUTINE CONVRT
C--
      IL        = 0
C--   CALC ARRAY LOCATION
      IT        = ISTA
      JT        = JSTA
      IF (TURN.GE.0.0) GOTO 5
      IT        = JSTA
      JT        = ISTA
C--   SET VALUE OF INTEGRAL CURB YBAR
    5 CURYB     = TK / 5.0
      IF (CUHT.LT.0.001) GOTO 10
      CURYB     = CUHT - ((CUHT**2 * CUWD / 2.0) + (TK * (HX / 2.0) *
     C            (CUHT + TK / 2.0))) / (CUHT * CUWD + TK * HX / 2.0)
   10 IF (IGRTP.EQ.NOBM) GOTO 35
C--   SET GIRDER I FOR COMPOSITE ZCNE ACTION
      BMI       = COMPI(2)
      YBAR      = CYBAR(2)
      IF ((IT.EQ.IGIRDL(1)).OR.(IT.EC.IGIRDL(NGIRDR))) GOTC 15
      BMI       = COMPI(1)
      YBAR      = CYBAR(1)
C--   CALC YT FOR SLAB STRESS AT GIRDER IN COMPOSITE ZCNE
   15 YT        = BEAMD - YBAR + TK + HAUNCH
      IYES      = 1
      IF (IGRTP.EQ.IPAN) GOTO 999
C--   CHECK FOR TRANSVERSE LOCATION AT A GIRDER
      IG        = 0
      IYES      = 0
      DO 20 IP  = 1,NGIRDR
```

```
   20 IF (IT.EQ.IGIRDL(IP)) IG = IP
      IF (IG.EQ.0) GOTO 30
C--   CHECK FOR ZONE OF COMPOSITE ACTION
   25 IL        = IL +1
      IF (IL.GT.NCOMP) GOTO 30
      IF (JT.GT.JDIST2(IL,IG)) GOTO 25
      IF (JT.GE.JDIST1(IL,IG)) IYES = 1
      IF (JT.GE.JDIST1(IL,IG)) GOTO 999
C--   RESET VALUES FOR ZONE OF NONCOMPOSITE ACTICN
   30 BMI       = BEAMI
      YBAR      = CENT
      GOTO 999
C--   SET VALUES FOR SLAB BRIDGE
   35 BMI       = 1.0
      YT        = CURYB + TK
      BEAMD     = CUHT
      YBAR      = 0.0
C--
  999 RETURN
      END
      SUBROUTINE PRNT1 (IX,IY,IGIRDL)
      COMMON  /RESL/ SLBM(50,50,4),DEF(50,50)
      COMMON  /RD  / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C               ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
     C               ,JNAM,NSPAN,NPLOT,NDAT,NPUT,NAMS(50),YDAT(50,2)
     C               ,CUHT,CUWD,HAUNCH
      COMMON  /UNIT/ NINP,NOUT
      DIMENSION      IGIRDL(50),DAT(5),ITL(6)
      DATA           IPRS,NOBM/4HPRES,4HSLAB/,
     C               ITL(1),ITL(2),ITL(3),ITL(4),ITL(5),ITL(6)
     C             / 4HLONG,4HITUD,4HINAL,4H TRA,4HNSVE,4HRSE /
C--
C--   THIS ROUTINE PRINTS RESULTS AS SPECIFIED IN TABLE 4 CF
C--   THE PREPRCESSOR. IF TABLE 4 OPTION IS USED FCR GENERAL
C--   OUTPUT LIMITS, THE RESULTS MAY CONTAIN MANY ZERCES.
C--
      LST       = 1
C--   PRINT TITLES FOR STRESS OUTPUT
      WRITE (NOUT,601)
      WRITE (NOUT,602)
      WRITE (NOUT,603)
C--   WRITE PRESTRESSED NOTE
      IF (IGRTP.EQ.IPRS) WRITE (NOUT,608)
    5 CONTINUE
      NA        = NAMS (LST)
C--   SET LOOP CONTROLS FOR GENERAL OUTPUT OF TOTAL BRIDGE
      IS        = 1
      JS        = 1
      IXI       = IX
      IYI       = IY
      IF ((NPLOT.EQ.0).OR.(NDAT.GT.2)) GCTO 10
C--   RESET LOOP CONTROLS FOR SPECIFIED GIRDERS
                           IS = IGIRDL (NA) + 1
      IF (IGRTP.EQ.NOBM) IS = NA + 1
      IXI       = IS
      IF (YDAT(LST,2).LE.0.001)  GOTC 10
      JS        = YDAT(LST,1)/HY+1.5
      IYI       = YDAT(LST,2)/HY+1.5
   10 CONTINUE
C--   LOOP TO PRINT RESULTANT STRESSES
      DO 40 I = IS,IXI
      IF (IXI.NE.IS) GOTO 15
      IF (IGRTP.EQ.NOBM) WRITE (NOUT,609) NA
```

```
      IF (IGRTP.NE.NOBM) WRITE (NOUT,604) NA
   15 CONTINUE
      IF (I.NE.IS) GOTO 25
      IF (TURN.LT.0.0) GOTO 20
      WRITE (NOUT,605) (ITL(K),K=4,6),(ITL(J),J=1,3)
      GOTO 25
   20 WRITE (NOUT,605) (ITL(K),K=1,3),(ITL(J),J=4,6)
   25 CONTINUE
      WRITE (NOUT,606)
      DO 40 J = JS,IYI
C--   SET ARRAY LOCATION
      II       = I
      JJ       = J
      IF  (TURN.GT.0.0) GOTO 30
      II       = J
      JJ       = I
C--   SET GRID LOCATION
   30 ISTA     = II - 1
      JSTA     = JJ - 1
C--   CONVERT UNITS FOR PRINT OUT
      DAT (1) = DEF (II,JJ) * 12.0
      DO 35 K = 1,4
      K1       = K + 1
   35 DAT(K1) = SLBM (II,JJ,K) * 1000.0 / 144.0
C--   PRINT RESULTS
   40 WRITE (NOUT,607) ISTA,JSTA,(DAT(K),K=1,5)
C--   CHECK FOR MORE SPECIFIED PLOT AND DATA
      IF (LST.GE.NPLOT) GOTO 999
      LST      = LST + 1
      GOTO 5
C--
  601 FORMAT (5H1       ,80X,10HI----TRIM  ,// )
  602 FORMAT ( //,5X,27HCOMPUTED RESULTANT STRESSES ,/)
  603 FORMAT (  /,8X,30HTHESE STRESSES ARE COMPUTED BY
     C               30H METHODS DESCRIBED IN          ,/
     C            ,8X,32HCHFR REPORT 56 - 25  FOR SLAB49 )
  604 FORMAT (  /,8X,30HSTRESSES ALONG GIRDER NUMBER - ,I3 )
  605 FORMAT (//,6X,4H  X ,4H  Y ,2X,11H DEFLECTION ,1X, 3A4
     C        ,1X, 3A4  ,2X,11HGIRDER TOP  ,2X,13HGIRDER BOTTOM
     C        ,/,28X,12HSLAB  STRESS,1X,12HSLAB  STRESS,
     C          2X,11H  STRESS   ,2X,11H  STRESS    ,
     C  /,19X,4H(IN),8X,5H(PSI),8X,5H(PSI),8X,5H(PSI),8X,5H(PSI)   )
  606 FORMAT ( / )
  607 FORMAT ( 5X , 2I4 , 5F13.3 )
  608 FORMAT (/8X,33HTHESE STRESSES DO NOT INCLUDE THE
     C              33H STRESSES INDUCED BY PRESTRESSING  /)
  609 FORMAT (//8X,26HSTRESSES ALONG GRID LINE -  ,I3)
C--
  999 RETURN
      END
      SUBROUTINE TWIST (IDIV)
      COMMON  /RESL/ SBD(50,50,5)
      DIMENSION      SBT(50,50)
C--
C--   THIS ROUTINE CHANGES THE ORDER OF THE RESULTS ARRAYS
C--   FOR CASES   TURN = -1.0. THIS ORDERS THE ARRAYS THE
C--   SAME FOR ALL BRIDGES TO ALLOW FOR EASY PLOTTING
C--
C--   SET LIMITS FOR REORDERING ARRAYS
      II          = IDIV + 1
      JJ          = II
C--   CHANGE ORDER OF SLAB ARRAYS
      DO 5      J = 1,II
```

```
      DO 5      I = 1,JJ
      AT          = SBD(I,J,1)
      SBD(I,J,1) = SBD(I,J,2)
      SBD(I,J,2) = AT
    5 CONTINUE
C--   CHANGE ORDER OF GIRDER ARRAYS
      DO 20     K = 1,5
      DO 10     J = 1,II
      DO 10     I = 1,JJ
   10 SBT(J,I)    = SBD(I,J,K)
      DO 15     J = 1,JJ
      DO 15     I = 1,II
   15 SBD(I,J,K) = SBT(I,J)
   20 CONTINUE
C--
      RETURN
      END
      SUBROUTINE MAXT
      COMMON  /UNIT/ NINP,NOUT
      COMMON  /PLTN/ LOC(50,4),VMAX(4),NVAL(4),IBUF(1000),NPLT,NDUB
      COMMON  /RESL/ ST(50,50),SB(50,50,4)
      COMMON  /RD  / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C               ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
     C               ,JNAM,NSPAN,NPLOT,NCAT,NPUT,NAMS(50),YDAT(50,2)
     C               ,CUHT,CUWD,HAUNCH
      COMMON  /C   / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
     C               ,JDIST1(12,50),JDIST2(12,50),MORT(50,2),JDD(12,2)
     C               ,LART(50,11)
      DATA           BOT,TOM,TOP,POT /3HBCT,3HTCM,3HTCP,3H    /
C--
C--   THIS SUBROUTINE FINDS MAXIMUMS ALONG THE GIRDERS
C--   IT ALSO PRINTS MAXIMUMS AND CALLS PLOT ROUTINES
C--
      IXX       = JDIV
      IPLT      = 1
                         IX1 = IXX + 1
      IF (TURN.LT.0.0) IX1 = IDIV + 1
      WRITE (NOUT,601)
      DO 45 IC = 1,NGIRDR
C--   SET INITIAL VALUES FOR COMPARISONS
      I         = IGIRDL(IC) + 1
      BG        = 0.0
      SM        = 0.0
      JLS       = 0
      JLB       = 0
      ILS       = 0
      ILB       = 0
      PS        = POT
      PB        = POT
      QS        = POT
      QB        = POT
      DSS       = 0.0
      DTB       = 0.0
      DO 20  K = 1,4
C--   SET INITIAL GLOBAL MAX
      VMAX(K)   = 0.0
      L         = 0
      DO 15  J = 1,IX1
      IF ((J.EQ.1).OR.(J.EQ.IX1)) GOTO 5
C--   SET VALUES FOR LOCAL MAXIMUM CHECK
      A         = ABS(SB(I,J-1,K))
      B         = ABS(SB(I,J,K))
      C         = ABS(SB(I,J+1,K))
```

```
C--    CHECK FOR LOCAL MAXIMUM
       IF ((A.GT.B).OR.(C.GT.B))   GOTO 15
C--    UPDATE POSITIVE STRESS MAXIMUM
       L          = L + 1
       LOC(L,K) = J
       IF (B.GT.VMAX(K)) VMAX (K) = B
     5 IF ((K.EQ.1).OR.(K.EQ.4)) GOTO 15
       IF (SB(I,J,K).LT.0.0) GOTO 7
       IF (BG.GT.SB(I,J,K))  GOTO 15
       BG         = SB(I,J,K)
       ILB        = I - 1
       JLB        = J - 1
       DTB        = FLOAT(JLB) * HY
       PB         = BOT
       QB         = TOM
       IF (K.EQ.3) GOTO 15
       PB         = TOP
       QB         = POT
       GOTO 15
C--    UPDATE NEGATIVE STRESS MAX
     7 IF (SM.LT.SB(I,J,K))  GOTO 15
       SM         = SB(I,J,K)
       ILS        = I - 1
       JLS        = J - 1
       DSS        = FLOAT(JLS) * HY
       PS         = BOT
       QS         = TOM
       IF (K.EQ.3) GOTO 15
       PS         = TOP
       QS         = POT
    15 CONTINUE
C--    SET STATION FOR OVERALL MAXIMUM
       NVAL(K)  = L
    20 CONTINUE
C--    CONVERT TO PSI FOR PRINT OUT
       SM1        = SM * 1000.0 / 144.0
       BG1        = BG * 1000.0 / 144.0
       IF (TURN.LT.0.0) GOTO 25
C--    PRINT RESULTANT STRESS MAX AND MIN
       WRITE (NOUT,602)  ILS,JLS,SM1,PS,QS,DSS,IC,
      C                  ILB,JLB,BG1,PB,QB,DTB,IC
       GOTO 30
    25 WRITE (NOUT,602)  JLS,ILS,SM1,PS,QS,DSS,IC,
      C                  JLB,ILB,BG1,PB,QB,DTB,IC
    30 CONTINUE
C--    CHECK IF PLOTS TO BE MADE
       IF (NDAT.EQ.1) GOTO 40
       IF (NPLOT.EQ.0) GOTO 40
       NAM        = 0
C--    CHECK FOR GIRDER TO BE PLOTTED
       DO 35  I = 1,NPLOT
    35 IF (IC.EQ.NAMS(I)) NAM = IC
       IF (NAM.EQ.0) GOTO 40
C--    CALL PLOTTING ROUTINES
       CALL PLOK ( NAM, IPLT, IX1 )
       IPLT       = IPLT + 1
    40 CONTINUE
       WRITE (NOUT,603)
    45 CONTINUE
C--
   601 FORMAT (29X,18HLOCATION  DISTANCE   ,/,20X,
      C    35HMAXIMUM      IN      ALONG      GIRDER  ,/,
      C    9X,6HX     Y,5X,6HSTRESS,4X,6HGIRDER,3X,6HGIRDER,
```

```
     C    4X,6HNUMBER,/,20X,5H(PSI) ,15X,4H(FT) ,/)
  602 FORMAT (5X,2I5,2X,F11.2,2X,2A3,F10.2,I7)
  603 FORMAT ( )
C--
      RETURN
      END
      SUBROUTINE SLBMAX (IDIV,JDIV,TURN)
      COMMON  /RESL/ SLM(50,50,2),BM(50,50,2),DEF(50,50)
      COMMON  /UNIT/ NINP,NOUT
C--
C--   THIS ROUTINE LOCATES ABSOLUTE MAXIMUM TRANSVERSE SLAB STRESS
C--
      KT       = 1
C--   SET INITIAL VALUES
      A        = ABS (SLM(1,1,KT))
      II       = IDIV + 1
      JI       = JDIV + 1
C--   LOCATE SLAB STRESS VALUES FOR COMPARISON
      DO 15 I = 1,II
      DO 15 J = 1,JI
      IF  (J.LT.JI)     GOTO 5
      IF  (I.EQ.II)     GOTO 15
      B        = ABS (SLM(I+1,1,KT))
      GOTO 4
    5 B        = ABS (SLM(I,J+1,KT))
C--   CHECK FOR OVERALL MAXIMUM
    4 IF (A.GE.B)       GOTO 15
      IM       = I
      JM       =  J + 1
      A        = B
      IF (J.LT.JI)      GOTO 15
      IM       = I + 1
      JM       = 1
   15 CONTINUE
C--   PRINT TRANSVERSE SLAB STRESS MAXIMUM
      IS       = IM - 1
      JS       = JM - 1
      SLBST    = SLM(IM,JM,KT) * 1000.0 / 144.0
      IF (TURN.GE.0.0) GOTO 20
      WRITE (NOUT,601) JS,IS,SLBST
      GOTO 25
   20 WRITE (NOUT,601) IS,JS,SLBST
   25 CONTINUE
C--
  601 FORMAT (///,5X,31HTRANSVERSE SLAB STRESS MAXIMUM    ,//,
     C         5X,4H    X,2X,4H    Y ,2X,8H   STRESS    ,/,19X,5H(PSI),//,
     C         3X, 2(2X,I4),2X,F11.2,/)
C--
      RETURN
      END
      SUBROUTINE EXTRAP
      COMMON  /EXTR/ VVAX(50,3),AXX(50,3),XML(50,11,3),XMR(50,11,3)
      COMMON  /SKEW/ THETA,ANGLE,ARM,LD,NDUM
      COMMON  /C    / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
     C                ,JD(24,50),MORT(50,2),JDD(12,2)
     C                ,LART(50,11)
      COMMON  /RD   / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C                ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
     C                ,JNAM,NSPAN,NPLOT,NDAT,NPUT,NAMS(50),YDAT(50,2)
     C                ,CUHT,CUWD,HAUNCH
      COMMON  /RESL/ ST(50,50),SB(50,50,4)
      DATA           NOBM/4HSLAB/
C--
```

```
C--    THIS SUBROUTINE DOES A FORWARD AND BACKWARD
C--    LINEAR EXTRAPOLATION FOR THE APPROXIMATE VALUES
C--    OF NEGATIVE MOMENT STRESSES AT INTERIOR SUPPORTS
C--
       NSPANN        = NSPAN-1
       DO 5       K  = 1,3
       DO 5      IC  = 1,NGIRDR
       VVAX(IC,K)    = 0.0
       IF (IGRTP.EQ.NOBM.AND.K.GT.1) GOTO 5
                            IGLC = IGIRDL(IC)
       IF (IGRTP.EQ.NOBM)   IGLC = IC
                            DIS = FLOAT(IGLC)*HX*ANGLE
       IF (THETA.LT.90.)    DIS = (ARM-FLOAT(IGLC)*HX)*ANGLE
       DO 4       M  = 1,NSPANN
       DIS           = DIS+SPANL(M)
C--    CALC LOCATION FOR INTERIOR SUPPORT TO BE EXTRAPOLATED
       AX            = ABS((DIS-HY*FLOAT(LART(IC,M)))/HY)
       BX            = 1.-AX
       AXX(IC,M)     = AX
C--    CALC TRANS. AND LONG. GRID LOCATION FOR EXTRAPOLATIONS
       I             = IGLC+1
       J             = LART(IC,M)+1
C--    FORWARD INTERPOLATION
       E             = SB(I,J-1,K)
       F             = SB(I,J,K)
       X1            = F-E
       XML(IC,M,K)   = F+X1*AX
C--    BACKWARD INTERPOLATION
       F             = SB(I,J+1,K)
       E             = SB(I,J+2,K)
       X1            = F-E
       XMR(IC,M,K)   = F+X1*BX
       BL            = ABS(XML(IC,M,K))
       BR            = ABS(XMR(IC,M,K))
C--    DETERMINE MAX VALUE OF EXTRAPOLATED VALUES FOR GIRDER
       IF (BL.GT.VVAX(IC,K))  VVAX(IC,K)=BL
       IF (BR.GT.VVAX(IC,K))  VVAX(IC,K)=BR
     4 CONTINUE
     5 CONTINUE
C--
       RETURN
       END
       SUBROUTINE PLOK ( IG, ICONT, IX1 )
       COMMON   /RD   / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
      C                 ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
      C                 ,JNAM,NSPAN,NPLOT,NCAT,NPUT,NAMS(50),YDAT(50,2)
      C                 ,CUHT,CUWD,HAUNCH
       COMMON   /C    / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
      C                 ,JDIST1(12,50),JDIST2(12,50),MORT(50,2),JDD(12,2)
      C                 ,LART(50,11)
       COMMON   /EXTR/  VVAX(50,3),AXX(50,3),XML(50,11,3),XMR(50,11,3)
       COMMON   /SKEW/  THETA,ANGLE,ARM,LD,NDUM
       COMMON   /PLTN/  LOC(50,4),VMAX(4),NVAL(4),IBUF(1000),NPLT,NDUB
       DIMENSION        RAT(2,2)
       DATA             NOBM / 4HSLAB /
C--
C--    THIS ROUTINE INITIATES THE PLOTS FOR THIS PROGRAM
C--
       IF ( ICONT.GT.1 ) GOTO 1
C--    FIND SCALE FACTOR FOR Y-SCALE
       DJV         = JDIV
       SCY         = 8.0 / (DJV*HY)
C--    CALC. JUMP RATIOS FOR COMPOSITE-NONCOMPOSITE POINTS
```

```
      RINT       = COMPI (1) * CENT / BEAMI
      REXT       = COMPI (2) * CENT / BEAMI
      RAT (1,1) = RINT / ABS ( BEAMD - CYBAR(1) )
      RAT (1,2) = RINT / CYBAR (1)
      RAT (2,1) = REXT / ABS ( BEAMD - CYBAR(2) )
      RAT (2,2) = REXT / CYBAR (2)
    1 CONTINUE
C--   CALC. GIRDER OFFSET FOR SKEW CASE
      GIRDL      = IGIRDL (IG)
                         DIS = GIRDL * HX * ANGLE
      IF (THETA.LT.90.0) DIS = ( ARM - GIRDL * HX ) * ANGLE
      IGR        = IGIRDL (IG) + 1
C--   FIND COMPOSITE LIMITS FOR GIRDER
      JDD (1,1) = 0
      JDD (1,2) = 0
      IF ( NCOMP.EQ.0 ) GOTO 3
      DO 2    M = 1 , NCOMP
      JDD (M,1) = JDIST1 ( M,IG )
      JDD (M,2) = JDIST2 ( M,IG )
    2 CONTINUE
    3 CONTINUE
C--   COMPARE RESULTANT AND EXTRAPOLATED MAXIMUMS
      IF ( NSPAN.EQ.1 ) GOTO 5
      DO 4    I = 1 , 3
      IF ( VVAX (IG,I).GT.VMAX (I) ) VMAX (I) = VVAX (IG,I)
    4 CONTINUE
    5 CONTINUE
C--   INITIATE PLOTS
C     CALL BGNPLT ( 4LPLOT, 20.0, 50, 50 )                            CDC
      CALL PLOTS  ( IBUF, 1000, NPLT )                                IBM
C--   CALL ROUTINE TO LAYOUT PAGES
      ICHK       = 1
      IA         = 4
      IB         = 1
      CALL PAGE ( ICHK, DIS, SCY, IA, IB, IG, IGR, IX1, RAT )
      IF ( IGRTP.EQ.NOBM ) GOTO 6
C--   SET UP SECOND PAGE FOR GIRDER PLOTS
      ICHK       = 3
      IA         = 2
      IB         = 3
      CALL PAGE ( ICHK, DIS, SCY, IA, IB, IG, IGR, IX1, RAT )
    6 CONTINUE
C--   CALL TO END PLOTS
C     CALL ENDPLT                                                     CDC
      CALL PLOT  ( 20.0, 0.0, 999 )                                   IBM
C--
      RETURN
      END
      SUBROUTINE PAGE ( ICHK, DIS, SCY, IA, IB, IG, IGR, IX1, RAT )
      COMMON  /INF / INFO(58),INFO3,INF3
      COMMON  /PLTN/ LOC(50,4),VMAX(4),NVAL(4),IBUF(1000),NPLT,NOUB
      DIMENSION      INO (6), RAT(1,1)
      DATA           INO(1), INO(2) / 4HPROB,4HLEM /
C--
C--   THIS ROUTINE LAYS OUT EACH PAGE TO BE PLOTTED
C--
C--   DRAW PAGE TRIM
C     CALL PLT  ( 0.0,  0.0, -3 )                                     CDC
      CALL PLOT ( 0.0,  0.0, -3 )                                     IBM
C     CALL PLT  ( 0.0, 11.0,  2 )                                     CDC
      CALL PLOT ( 0.0, 11.0,  2 )                                     IBM
C--   LABEL PROBLEM AND GIRDER NUMBER
      FIGRD      = IG
```

```
      INO (3)    = INF3
      INO (4)    = INFO3
      XGN        = 3.85
      YGN        = 9.5
      XPN        = 3.75
      YPN        = 10.0
C--   PROBLEM NUMBER
      DO 1    I = 1,4
      CALL SYMBOL ( XPN, YPN, 0.14, INO (I) , 0.0,   4 )
    1 XPN        = XPN + 0.56
C--   GIRDER NUMBER
      CALL SYMBOL ( XGN, YGN, 0.14, 6HGIRDER, 0.0,   6 )
      XGN        = XGN + 1.2
      CALL NUMBER ( XGN, YGN, 0.14, FIGRD   , 0.0, - 1 )
C--   CHECK FOR FIRST OR SECOND PAGE OF PLOTS
      IF ( ICHK.GT.1 ) GOTO 2
C--   LABEL FIRST PAGE PLOTS
C---  WRITE ABSOLUTE MAXIMUMS
      APNT = VMAX(IA) * 12.0
      CALL SYMBOL ( 1.4 , 1.5 , 0.07 ,16HABS MAX (IN)  = , 90.0 , 16 )
      CALL NUMBER ( 1.4 , 2.7 , 0.07 , APNT , 90.0 , 3 )
      CALL SYMBOL ( 2.17,0.8,0.07,21HCENTERLINE DEFLECTION , 0.0,21 )
      CALL SYMBOL ( 2.0 ,1.5,0.C7, 2HUP                    ,90.0, 2 )
      CALL SYMBOL ( 3.8 ,1.5,0.07, 4HDOWN                  ,90.0, 4 )
      CALL SYMBOL ( 5.37,0.8,0.07,19HSTRESS - SLAB TOP     , 0.0,19 )
      GOTO 3
    2 CONTINUE
C--   PLOT SECOND PAGE TITLE
C---  WRITE ABSOLUTE MAXIMUMS
      APNT = VMAX(IA) * 1000.0/144.0
      CALL SYMBOL ( 1.4 , 1.5 , 0.07 ,16HABS MAX (PSI) = , 90.0 , 16 )
      CALL NUMBER ( 1.4 , 2.7 , 0.07 , APNT , 90.0 , 3 )
      CALL SYMBOL ( 2.20,C.8,0.C7,20HSTRESS - GIRDER TOP   , 0.0,20 )
      CALL SYMBOL ( 2.0 ,1.5,0.07, 8HTEN  (+)              ,90.0, 8 )
      CALL SYMBOL ( 3.8 ,1.5,0.07, 8HCOMP (-)              ,90.0, 8 )
      CALL SYMBOL ( 5.33,0.8,0.07,22HSTRESS - GIRDER BOTTOM, 0.0,22 )
    3 CONTINUE
C--   LABEL PLOT ON RIGHT SIDE OF PAGE
      CALL SYMBOL ( 5.2 , 1.5, 0.07, 8HTEN  (+) , 90.0,  8 )
      CALL SYMBOL ( 7.0 , 1.5, 0.07, 8HCCMP (-) , 90.0,  8 )
C---  WRITE ABSOLUTE MAXIMUMS
      BPNT = VMAX(IB) * 1000.0/144.0
      CALL SYMBOL ( 4.6 , 1.5 , 0.07 , 16HABS MAX (PSI) = , 90.0 , 16 )
      CALL NUMBER ( 4.6 , 2.7 , 0.07 , BPNT , 90.0 , 3 )
C--   PLOT THE DIAGRAMS
      XPAGE      = 2.9
      CALL LINE ( XPAGE, DIS, SCY, IG, IGR, IA, IX1, RAT )
      XPAGE      = 6.1
      CALL LINE ( XPAGE, DIS, SCY, IG, IGR, IB, IX1, RAT )
C--   DRAW PAGE BOUNDARY
C     CALL PLT  ( 8.5, 11.0,  3 )                                      CDC
      CALL PLOT ( 8.5, 11.0,  3 )                                      IBM
C     CALL PLT  ( 8.5,  0.0, -2 )                                      CDC
      CALL PLOT ( 8.5,  0.0, -2 )                                      IBM
C--
      RETURN
      END
      SUBROUTINE LINE ( XPAGE, DIS, SCY, IG, IGR, IPLT, IXX, RAT )
      COMMON  /RD  / CYBAR(2),COMPI(2),CENT,HX,HY,TK,TURN,BEAMI,BEAMD,EC
     C              ,ES,TOPS,BOTS,IGRTP,INAM,LRD,NGIRDR,NCOMP,IDIV,JDIV
     C              ,JNAM,NSPAN,NPLOT,NDAT,NPUT,NAMS(50),YDAT(50,2)
     C              ,CUHT,CUWD,HAUNCH
      COMMON  /C   / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
```

```
      C                 ,JDIST1(12,50),JDIST2(12,50),MORT(50,2),JDD(12,2)
      C                 ,LART(50,11)
       COMMON  /RESL/ ST(50,50),SB(50,50,4)
       COMMON  /PLTN/ LOC(50,4),VMAX(4),NVAL(4),IBUF(1000),NPLT,NDUB
       DIMENSION       RAT ( 2,2 )
       LOGICAL         PLOTX,PLTEND
C--
C--    THIS ROUTINE DRAWS THE PLOT OF ONE GIRDER ARRAY WITH SUPPORTS
C--    AND LOCAL MAXIMUMS SHOWN. IT ALSO CALLS EXTPLT FOR
C--    EXTRAPOLATED VALUES TO BE PLOTTED.
C--
C--    RESET ORIGIN
C      CALL PLT  ( XPAGE, 1.5, -3 )                                     CDC
       CALL PLOT ( XPAGE, 1.5, -3 )                                     IBM
C--    SET ICBD FOR IBM OR CDC
C      IPS       = 3                                                    CDC
       IPS       = 5                                                    IBM
C--    PLOT SPANS
       SPRT      = DIS * SCY
       NSPN1     = NSPAN + 1
       DO 1    I = 1, NSPN1
C--    PLOT SUPPORTED POINT
       CALL SYMBOL ( 0.0, SPRT, 0.14, IPS, 90.0, -2 )
       IF ( I.EQ.NSPN1 ) GOTO 1
                         SPRT = SPRT + SPANL (I) * SCY
     1 IF ( SPRT.GT.8.0 ) SPRT = 8.0
C--    DRAW LINE TO END OF GRID FOR SKEWED CASE
C      CALL PLT  ( 0.0, 8.0, 2 )                                        CDC
       CALL PLOT ( 0.0, 8.0, 2 )                                        IBM
C--    SET ORIENTATION FOR IBM OR CDC.
C      AGIE      = 0.0                                                  CDC
       AGIE      = 90.0                                                 IBM
       LSTOP     = NVAL (IPLT)
C--    PLOT LOCAL MAXIMUMS ALONG THE LINE
       DO 2    L = 1, LSTOP
       OCL       = LOC ( L,IPLT ) - 1
       ALOCM     = OCL * HY * SCY
C--    DRAW THE LOCAL MAXIMUM AND LABEL THE STATION
       CALL SYMBOL ( 0.0 , ALOCM, 0.07, 13 , AGIE, -1 )
       CALL NUMBER ( 0.15, ALOCM, 0.07, OCL, 90.0, -1 )
     2 CONTINUE
C--    PLACE PEN AT BEGINNING OF GRID
C      CALL PLT  ( 0.0, 0.0, 3 )                                        CDC
       CALL PLOT ( 0.0, 0.0, 3 )                                        IBM
C--    SET SCALE FACTOR FROM MAXIMUM
       SCX       = 1.5 / VMAX (IPLT)
       COMPR     = 1.0
C--    SET LOGICAL VARIABLE FOR FIRST PAGE PLOTS
       PLOTX     = IPLT.EQ.1.OR.IPLT.EQ.4
       IF ( PLOTX ) GOTO 3
C--    SET RATIO FOR COMP. JUMP
                                    COMPR = RAT ( 1,IPLT-1 )
       IF ( IG.EQ.1.OR.IG.EQ.NGIRDR ) COMPR = RAT ( 2,IPLT-1 )
     3 IJD       = 1
       L         = 1
C--    PLOT LINE
       DO 7    J = 1, IXX
       XPLT      = - SB ( IGR,J,IPLT ) * SCX
       JSTA      = J - 1
       RJ        = JSTA
       YPLT      = RJ * HY * SCY
       IF ( PLOTX ) GOTO 6
C--    COMPOSITE JUMP CHECK
```

```
      IF ( JSTA.NE.JDD ( L,IJD )) GOTO 6
      PLTEND    = J.EQ.1.OR.J.EQ.IXX
      IF ( PLTEND ) GOTO 5
C--   SET VALUES FOR COMPOSITE JUMP
      X1        = XPLT
      X2        = XPLT * COMPR
      IF ( IJD.EQ.2 ) GOTO 4
      X1        = X2
      X2        = XPLT
    4 CONTINUE
      Y1        = YPLT
      Y2        = YPLT
C--   PLOT POINT BEFORE JUMP
C     CALL PLT  ( X1, Y1, 2 )                                           CDC
      CALL PLOT ( X1, Y1, 2 )                                           IBM
C--   PLOT DASH FOR THE JUMP
      CALL DASH ( X1, Y1, X2, Y2 )
    5 CONTINUE
C--   SET ARRAY POSITION FOR NEXT COMPOSITE CHECK
                       IJD = IJD + 1
      IF ( IJD.GT.2 ) IJD = 1
      IF ( IJD.EQ.1 ) L   = L + 1
      IF ( .NOT.PLTEND ) GOTO 7
    6 CONTINUE
C     CALL PLT  ( XPLT, YPLT, 2 )                                       CDC
      CALL PLOT ( XPLT, YPLT, 2 )                                       IBM
    7 CONTINUE
C--   RETURN PEN TO LINE FOR SKEW
C     CALL PLT  ( 0.0, 8.0, 2 )                                         CDC
      CALL PLOT ( 0.0, 8.0, 2 )                                         IBM
C--   CALL ROUTINE TO PLOT EXTRAPOLATED VALUES ALONG GRID LINE
      CALL EXTPLT ( IPLT, IG, IGR, HY, SCY, SCX, NSPAN )
C--   PUT ORIGIN BACK TO PAGE CORNER
C     CALL PLT  ( -XPAGE, -1.5, -3 )                                    CDC
      CALL PLOT ( -XPAGE, -1.5, -3 )                                    IBM
C--
      RETURN
      END
      SUBROUTINE EXTPLT ( IPLT, IG, IGR, HY, SCY, SCX, NSPAN )
      COMMON  /C    / IGIRDL(50),SPANL(12),YDIST1(12),YDIST2(12)
     C                ,JDIST1(12,50),JDIST2(12,50),MCRT(50,2),JDD(12,2)
     C                ,LART(50,11)
      COMMON  /EXTR/ VVAX(50,3),AXX(50,3),XML(50,11,3),XMR(50,11,3)
      COMMON  /RESL/ ST(50,50),SB(50,50,4)
C--
C--   THIS ROUTINE PLOTS THE VALUES EXTRAPOLATED IN SUBROUTINE
C--   EXTRAP. CALLED FROM LINE
C--
C--   CHECK FOR PLOTS WITH EXTRAPOLATED VALUES
      IF ( IPLT.EQ.4 ) GOTO 999
      IF ( NSPAN.EQ.1) GOTO 999
      NSTOP     = NSPAN - 1
      DO 1    M = 1, NSTOP
C--   CHECK FOR SUPPORT CLOSE TO GRID STATION
      XCK       = AXX ( IG,M )
      IF (XCK.GT.0.995.OR.XCK.LT.0.005) GOTO 1
C--   LOCATION OF GRID LOCATION BEFORE SUPPORT
      J         = LART ( IG,M ) + 1
      RJ        = J - 1
C--   SET VALUES FOR FORWARD INTERPOLATION PLOT
      X1        = - SB ( IGR, J, IPLT ) * SCX
      Y1        = RJ * HY * SCY
      X2        = - XML ( IG, M, IPLT ) * SCX
```

```
      Y2          = ( RJ + AXX ( IG,M )) * HY * SCY
      CALL DASH ( X1, Y1, X2, Y2 )
C--   SET VALUES FOR BACKWARD INTERPOLATION PLOT
      X1          = - XMR ( IG , M, IPLT ) * SCX
      Y1          = Y2
      J1          = J + 1
      X2          = - SB  ( IGR,J1, IPLT ) * SCX
      Y2          = ( RJ + 1.0 ) * HY * SCY
      CALL DASH ( X1, Y1, X2, Y2 )
    1 CONTINUE
C--
  999 RETURN
      END
      SUBROUTINE DASH ( X1, Y1, X2, Y2 )
C--
C--   THIS ROUTINE MAKES A DASHED LINE FROM (X1,Y1) TC (X2,Y2).
C--
C--   FIND TOTAL LENGTH OF DASHED LINE
      DX          = X2 - X1
      DY          = Y2 - Y1
      ALNGTH      = SQRT ( DX ** 2 + DY ** 2 )
C--   FIND NUMBER OF DASHES
      NDASH       = ALNGTH / 0.07
      IF ( NDASH.LT.5  ) NDASH = 5
      IF ( NDASH.GT.11 ) NDASH = 11
C--   MAKE NDASH AN ODD NUMBER
      NDASH       = (( NDASH / 2 ) * 2 ) + 1
      DASHN       = NDASH
C--   SET DASH LENGTH
      XSTEP       = DX / DASHN
      YSTEP       = DY / DASHN
      XLOC        = X1
      YLOC        = Y1
      IPEN        = 3
      NSTOP       = NDASH + 1
      DO 1     I = 1, NSTOP
C     CALL PLT  ( XLOC, YLOC, IPEN )                                        CDC
      CALL PLOT ( XLOC, YLOC, IPEN )                                        IBM
C--   UPDATE PEN LOCATION
      XLOC        = XSTEP + XLOC
      YLOC        = YSTEP + YLOC
C--   SET PEN POSITION
                        IPEN = IPEN + 1
      IF ( IPEN.GT.3 ) IPEN = 2
    1 CONTINUE
C--
      RETURN
      END
```

APPENDIX I

OUTPUT FROM THE PRE-PROCESSOR (SLBDG4)

```
EXAMPLE PROBLEM ; RWS.CPJ ; 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ;  TWO LOAD CASES


PROBLEM
   01      TWO STANDARD AASHTO H15 TRUCKS


TABLE 1. CONTROL DATA

   NUMBER OF LOAD CASES                                              2
   PRINT OF GENERATED SLAB49 INPUT (1=YES)                           0
   3-D PLOT OPTION                                                   1
   BRIDGE DEAD WEIGHT OPTION                                         1
   TABLE 4 RETAINED FROM PREVIOUS PROBLEM                            0


TABLE 2. BRIDGE GEOMETRY

  A. DESCRIPTION
   TOTAL BRIDGE WIDTH                          (FT)              28.25
   SLAB OVERHANG                               (FT)               3.13
   DIVISIONS ACROSS BRIDGE - TRANSVERSE                             12
   DIVISIONS ALONG BRIDGE - LONGITUDINAL                            20
   SLAB THICKNESS                              (IN)               6.34
   BRIDGE ANGLE OF SKEW                   (DEGREES)               0.00
   COMPUTED GIRDER SPACING                     (FT)               7.33
   OVERHANG REDEFINED FOR COMPUTATIONS         (FT)               2.44
   DIVISIONS ACROSS BRIDGE - REDEFINED                              11

  B. MATERIAL PROPERTIES
   CONCRETE WEIGHT                            (PCF)             150.00
   CONCRETE STRENGTH                          (PSI)            3000.00
   POISSONS RATIO                                                .1500
   MODULUS OF ELASTICITY - CONCRETE           (PSI)          3.000E+06
   MODULUS OF ELASTICITY - STEEL              (PSI)          2.900E+07

  C. GIRDER INFORMATION
   NUMBER OF GIRDERS                                                 4
   GIRDER TYPE                                                 PRES
   GIRDER NAME                                                 TYPE C
   NUMBER OF STRANDS                                                12
   INITIAL FORCE IN STRAND                   (KIPS)              28.91
   PERCENT LOSS OF FORCE IN STRAND                               20.00
   ECCENTRICITY (MIDSPAN)                      (IN)              14.76
   HAUNCH                                      (IN)               0.00
   GIRDER - CROSS SECTIONAL AREA             (IN-2)             361.00
   GIRDER - MOMENT OF INERTIA                (IN-4)             43300.
   GIRDER - DEPTH                              (IN)              34.00
   GIRDER - TOP FLANGE WIDTH                   (IN)              14.00
   GIRDER - CENTROID TO BOTTOM                 (IN)              15.00

       COMPUTED GIRDER INFORMATION
   TYPE C    LOCATED FROM DATA BASE
   GIRDER - DEPTH                              (FT)             2.8333
   GIRDER - CENTROID TO BOTTOM                 (FT)             1.2500
   GIRDER - MOMENT OF INERTIA                (FT-4)             2.0882
   COMPOSITE SECT - CENTROID TO BOTTOM (FT)                     2.3717
   COMPOSITE SECTION - I                     (FT-4)             7.2834
```

D. DIAPHRAGMS

NUMBER OF DIAPHRAGMS		5
DISTANCE - START TO FIRST DIAPHRAGM	(FT)	6.91
DISTANCE - START TO LAST DIAPHRAGM	(FT)	46.84
STIFFNESS OF DIAPHRAGMS (FT)	(LB-IN)	2.000E+10

E. SPAN INFORMATION

NUMBER OF SPANS INPUT		1
LENGTH OF SPAN 1	(FT)	53.75
TOTAL LENGTH OF BRIDGE	(FT)	53.75

F. ZONES OF COMPOSITE ACTION

NUMBER OF COMPOSITE ZONES INPUT		1
BEGIN COMPOSITE ZONE 1	(FT)	0.00
END COMPOSITE ZONE 1	(FT)	53.75

TABLE 3. TRUCK INFORMATION

NUMBER OF TRUCKS FOR LOAD CASE	2
OUTPUT GRID LOADING (1=YES)	0
CALCOMP PLOT OF LOAD CASE (1=YES)	1
NUMBER OF TRUCKS IN DATA BASE	3

TRUCK 1 DESCRIPTION AND PLACEMENT

TRUCK NAME		INPUT-KP
LOAD IMPACT FACTOR		1.30
TRUCK LOCATION - TRANSVERSE	(FT)	5.00
TRUCK LOCATION - LONGITUDINAL	(FT)	38.70
TRUCK DIRECTION		FORWARD

WHEEL DATA FROM TRUCK INPUT
NUMBER OF WHEELS 4

LOAD (KIPS)	LOCAL TRANS (FT)	LOCAL LONG (FT)
3.00	-3.00	0.00
3.00	3.00	0.00
12.00	-3.00	-14.00
12.00	3.00	-14.00

TRUCK 2 DESCRIPTION AND PLACEMENT

TRUCK NAME		INPUT-KP
LOAD IMPACT FACTOR		1.30
TRUCK LOCATION - TRANSVERSE	(FT)	-5.00
TRUCK LOCATION - LONGITUDINAL	(FT)	38.70
TRUCK DIRECTION		FORWARD

WHEEL DATA FROM TRUCK INPUT
NUMBER OF WHEELS 4

LOAD (KIPS)	LOCAL TRANS (FT)	LOCAL LONG (FT)
3.00	-3.00	0.00
3.00	3.00	0.00
12.00	-3.00	-14.00

12.00 3.00 -14.00

```
TABLE 4. OUTPUT AREAS

  A. POSTPROCESSOR OUTPUT
   NUMBER OF GIRDER OUTPUTS                                     2
   DATA OUTPUT (1=DATA ONLY,2=DATA+PLOT)                        2
   PRINT OUT OF SLAB49 MOMENTS (1=YES)                          0
   GIRDER NUMBER TO BE OUTPUT                                   2
   GIRDER NUMBER TO BE OUTPUT                                   3
   BEGIN DATA OUTPUT                        (FT)            15.00
   END DATA OUTPUT                          (FT)            35.00


COMPUTED GRID INFORMATION
   NUMBER OF INCREMENTS ACROSS BRIDGE                          11
   NUMBER OF INCREMENTS ALONG BRIDGE                           20
   INCREMENT LENGTH ACROSS BRIDGE               (FT)        2.444
   INCREMENT LENGTH ALONG BRIDGE                (FT)        2.688


        Y
  * * * * * * *
  *
  *   THE COORDINATE AXES FOR THE INTERPRETATION
X *   OF RESULTS FROM SLAB 49 ARE AS SHOWN.
  *
  *


MIDSPAN STRESSES INDUCED BY PRESTRESSING
    AND GIRDER DEAD WEIGHT

   STRESS AT GIRDER TOP                     (PSI)          313.64
   STRESS AT GIRDER BOTTOM                  (PSI)        -1623.35
```

```
PROBLEM
   02      ONE STANDARD AASHTO H20 FROM THE DATA BASE


TABLE 1. CONTROL DATA

   NUMBER OF LOAD CASES                                       2
   PRINT OF GENERATED SLAB49 INPUT (1=YES)                    0
   3-D PLOT OPTION                                            0
   BRIDGE DEAD WEIGHT OPTION                                  1
   TABLE 4 RETAINED FROM PREVIOUS PROBLEM                     0



TABLE 3. TRUCK INFORMATION

   NUMBER OF TRUCKS FOR LOAD CASE                             1
   OUTPUT GRID LOADING (1=YES)                                1
   CALCOMP PLOT OF LOAD CASE (1=YES)                          1
   NUMBER OF TRUCKS IN DATA BASE                              3

  TRUCK  1 DESCRIPTION AND PLACEMENT
   TRUCK NAME                                           HS20V14
   LOAD IMPACT FACTOR                                      1.30
   TRUCK LOCATION - TRANSVERSE           (FT)              0.00
   TRUCK LOCATION - LONGITUDINAL         (FT)              8.22
   TRUCK DIRECTION                                      BACKWARD
   HS20V14    LOCATED FROM DATA BASE

   WHEEL DATA FROM TRUCK DATA BASE
   NUMBER OF WHEELS                       6
               LOCAL       LOCAL
       LOAD    TRANS       LONG
      (KIPS)    (FT)       (FT)

       4.00       3.00      14.00
       4.00      -3.00      14.00
      16.00       3.00       0.00
      16.00      -3.00       0.00
      16.00       3.00     -14.00
      16.00      -3.00     -14.00

   WHEEL  1
   X-GRID LOCATION      Y-GRID LOCATION              GRID LOAD
            4                 -1                          0.00
            4                  0                          0.00
            5                 -1                          0.00
            5                  0                          0.00

   WHEEL  2
   X-GRID LOCATION      Y-GRID LOCATION              GRID LOAD
            6                 -1                          0.00
            6                  0                          0.00
            7                 -1                          0.00
            7                  0                          0.00
```

WHEEL 3

X-GRID LOCATION	Y-GRID LOCATION	GRID LOAD
4	3	14.25
4	4	.88
5	3	5.34
5	4	.33

WHEEL 4

X-GRID LOCATION	Y-GRID LOCATION	GRID LOAD
6	3	5.34
6	4	.33
7	3	14.25
7	4	.88

WHEEL 5

X-GRID LOCATION	Y-GRID LOCATION	GRID LOAD
4	8	11.09
4	9	4.04
5	8	4.16
5	9	1.52

WHEEL 6

X-GRID LOCATION	Y-GRID LOCATION	GRID LOAD
6	8	4.16
6	9	1.52
7	8	11.09
7	9	4.04

TABLE 4. OUTPUT AREAS

A. POSTPROCESSOR OUTPUT

NUMBER OF GIRDER OUTPUTS	2
DATA OUTPUT (1=DATA ONLY,2=DATA+PLOT)	2
PRINT OUT OF SLAB49 MOMENTS (1=YES)	0
GIRDER NUMBER TO BE OUTPUT	2
GIRDER NUMBER TO BE OUTPUT	3

COMPUTED GRID INFORMATION

NUMBER OF INCREMENTS ACROSS BRIDGE		11
NUMBER OF INCREMENTS ALONG BRIDGE		20
INCREMENT LENGTH ACROSS BRIDGE	(FT)	2.444
INCREMENT LENGTH ALONG BRIDGE	(FT)	2.688

```
        Y
  * * * * * * *
  *
  *   THE COORDINATE AXES FOR THE INTERPRETATION
X *   OF RESULTS FROM SLAB 49 ARE AS SHOWN.
  *
  *
```

MIDSPAN STRESSES INDUCED BY PRESTRESSING
AND GIRDER DEAD WEIGHT

STRESS AT GIRDER TOP	(PSI)	313.64
STRESS AT GIRDER BOTTOM	(PSI)	-1623.35

```
02 OCT 80 UNIVERSITY OF TEXAS 170/750B UT2D-215

CE1Z551-022

19.23.14 NEW JOB - SUBMIT CC=XT2 TM=95
19.23.22 IDENTIFY,57
19.23.22 SPL,55
19.23.22 +  20000CM     0.000CP       0MS       0MT
19.23.22 READPF 3813 EXAMPLE RICHB SCHB TAPE7 TAPE8 TAPE9 TAPE10
19.23.26  COPIED FILE EXAMPLE.
19.23.28  COPIED FILE RICHB.
19.23.28  COPIED FILE SCHB.
19.23.29  COPIED FILE TAPE7.
19.23.29  COPIED FILE TAPE8.
19.23.30  COPIED FILE TAPE9.
19.23.30  COPIED FILE TAPE10.
19.23.30 +  24000CM     0.338CP  1849MS       0MT
19.23.30 SETCORE
19.23.30 RICHB,EXAMPLE.
19.23.31 +  50000CM     0.344CP  1892MS       0MT
19.23.31 + 100000CM     0.368CP  1953MS       0MT
19.23.32 + 150000CM     0.510CP  2344MS       0MT
19.23.32   FL USED 101534.
19.23.32   LOAD TIME      1.928 TM SEC.
19.23.32 -  66100CM     0.548CP  2375MS       0MT
19.23.33    21 PLOTTER RECORDS
19.23.33   313 SQUARE INCHES PLOT
19.23.33    88 PLOTTER SECONDS
19.23.34    20 PLOTTER RECORDS
19.23.34   313 SQUARE INCHES PLOT
19.23.34    85 PLOTTER SECONDS
19.23.34  .. STOP
19.23.34 0  66100CM     1.044CP  2983MS       0MT
19.23.34  .. XTIME =    2.739 TM SECS.    FL USED =    66100B
19.23.34 -   1100CM     1.045CP  3015MS       0MT
19.23.34 DISPOSE OUTPUT ID=57
```

APPENDIX J

OUTPUT FROM THE PARENT PROGRAM (SLAB49)

PROGRAM SLAB 49 =DEVELOPMENT DECK= MATLOCK,PANAK, ENDRES REV DATE 13 JUL 71

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                             *
*   THIS PROGRAM IS BEING USED AT YOUR OWN RISK.              *
*   CHANGES MAY OCCUR AFTER THE ABOVE REVISION DATE.          *
*   PLEASE REPORT DIFFICULTIES TO THE ABOVE PEOPLE            *
*   AT THE CENTER FOR HIGHWAY RESEARCH, UT AT AUSTIN.         *
*                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

EXAMPLE PROBLEM ; RHS,CPJ ; 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ; TWO LOAD CASES

PROB
01 TWO STANDARD AASHTO H15 TRUCKS

TABLE 1. CONTROL DATA

	TABLE NUMBER							
	2	3	4	5	6	7	8	9
KEEP FROM PRECEDING PROBLEM (1=YES)	0	0	0	0	0	0	0	0
NUM CARDS INPUT THIS PROBLEM	1	98	0	11	8	40	0	0

MULTIPLE LOAD OPTION	1
STATICS CHECK OPTION	0
PRIN STRESS OPTION	1
PROFILE PLOT OPTION	0
3-D PLOT OPTION	1

TABLE 2. CONSTANTS

NUMBER OF INCREMENTS IN X DIRECTION	11
NUMBER OF INCREMENTS IN Y DIRECTION	20
INCREMENT LENGTH IN X DIRECTION	2.444E+00
INCREMENT LENGTH IN Y DIRECTION	2.688E+00
POISSONS RATIO	1.500E-01
SLAB THICKNESS	5.283E-01

TABLE 3. JOINT STIFFNESS AND LOAD DATA

FROM JOINT		THRU JOINT		DX	DY	FX	FY	Q	S
0	0	0	20	1.358E+03	1.358E+03	-0	-0	-0	-0
0	1	0	19	1.358E+03	1.358E+03	-0	-0	-0	-0
1	0	1	20	2.716E+03	2.716E+03	-0	-0	-0	-0
1	1	1	19	2.716E+03	2.716E+03	-0	-0	-0	-0
2	0	2	20	2.716E+03	2.716E+03	-0	-0	-0	-0
2	1	2	19	2.716E+03	2.716E+03	-0	-0	-0	-0
3	0	3	20	2.716E+03	2.716E+03	-0	-0	-0	-0
3	1	3	19	2.716E+03	2.716E+03	-0	-0	-0	-0
4	0	4	20	2.716E+03	2.716E+03	-0	-0	-0	-0
4	1	4	19	2.716E+03	2.716E+03	-0	-0	-0	-0

5	0	5	20	2.716E+03	2.716E+03	-0	-0	-0	-0
5	1	5	19	2.716E+03	2.716E+03	-0	-0	-0	-0
6	0	6	20	2.716E+03	2.716E+03	-0	-0	-0	-0
6	1	6	19	2.716E+03	2.716E+03	-0	-0	-0	-0
7	0	7	20	2.716E+03	2.716E+03	-0	-0	-0	-0
7	1	7	19	2.716E+03	2.716E+03	-0	-0	-0	-0
8	0	8	20	2.716E+03	2.716E+03	-0	-0	-0	-0
8	1	8	19	2.716E+03	2.716E+03	-0	-0	-0	-0
9	0	9	20	2.716E+03	2.716E+03	-0	-0	-0	-0
9	1	9	19	2.716E+03	2.716E+03	-0	-0	-0	-0
10	0	10	20	2.716E+03	2.716E+03	-0	-0	-0	-0
10	1	10	19	2.716E+03	2.716E+03	-0	-0	-0	-0
11	0	11	20	1.358E+03	1.358E+03	-0	-0	-0	-0
11	1	11	19	1.358E+03	1.358E+03	-0	-0	-0	-0
1	0	1	20	-0	-0	-0	4.510E+05	-0	-0
1	1	1	19	-0	-0	-0	4.510E+05	-0	-0
4	0	4	20	-0	-0	-0	4.510E+05	-0	-0
4	1	4	19	-0	-0	-0	4.510E+05	-0	-0
7	0	7	20	-0	-0	-0	4.510E+05	-0	-0
7	1	7	19	-0	-0	-0	4.510E+05	-0	-0
10	0	10	20	-0	-0	-0	4.510E+05	-0	-0
10	1	10	19	-0	-0	-0	4.510E+05	-0	-0
1	0	1	20	-0	-0	-0	1.088E+06	-0	-0
1	1	1	19	-0	-0	-0	1.088E+06	-0	-0
4	0	4	20	-0	-0	-0	1.122E+06	-0	-0
4	1	4	19	-0	-0	-0	1.122E+06	-0	-0
7	0	7	20	-0	-0	-0	1.122E+06	-0	-0
7	1	7	19	-0	-0	-0	1.122E+06	-0	-0
10	0	10	20	-0	-0	-0	1.088E+06	-0	-0
10	1	10	19	-0	-0	-0	1.088E+06	-0	-0
1	0	1	0	-0	-0	-0	-0	-0	1.000E+08
1	1	1	1	-0	-0	-0	-0	-0	0
4	0	4	0	-0	-0	-0	-0	-0	1.000E+08
4	1	4	1	-0	-0	-0	-0	-0	0
7	0	7	0	-0	-0	-0	-0	-0	1.000E+08
7	1	7	1	-0	-0	-0	-0	-0	0
10	0	10	0	-0	-0	-0	-0	-0	1.000E+08
10	1	10	1	-0	-0	-0	-0	-0	0
1	20	1	20	-0	-0	-0	-0	-0	1.000E+08
1	20	1	20	-0	-0	-0	-0	-0	0
4	20	4	20	-0	-0	-0	-0	-0	1.000E+08
4	20	4	20	-0	-0	-0	-0	-0	0
7	20	7	20	-0	-0	-0	-0	-0	1.000E+08
7	20	7	20	-0	-0	-0	-0	-0	0
10	20	10	20	-0	-0	-0	-0	-0	1.000E+08
10	20	10	20	-0	-0	-0	-0	-0	0
1	0	4	0	-0	-0	6.944E+04	-0	-0	-0
2	0	3	0	-0	-0	6.944E+04	-0	-0	-0
1	3	4	3	-0	-0	6.944E+04	-0	-0	-0
2	3	3	3	-0	-0	6.944E+04	-0	-0	-0
1	6	4	6	-0	-0	6.944E+04	-0	-0	-0
2	6	3	6	-0	-0	6.944E+04	-0	-0	-0
1	10	4	10	-0	-0	6.944E+04	-0	-0	-0
2	10	3	10	-0	-0	6.944E+04	-0	-0	-0
1	14	4	14	-0	-0	6.944E+04	-0	-0	-0
2	14	3	14	-0	-0	6.944E+04	-0	-0	-0
1	17	4	17	-0	-0	6.944E+04	-0	-0	-0
2	17	3	17	-0	-0	6.944E+04	-0	-0	-0
1	20	4	20	-0	-0	6.944E+04	-0	-0	-0
2	20	3	20	-0	-0	6.944E+04	-0	-0	-0
4	0	7	0	-0	-0	6.944E+04	-0	-0	-0
5	0	6	0	-0	-0	6.944E+04	-0	-0	-0

4	3	7	3	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
5	3	6	3	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
4	6	7	6	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
5	6	6	6	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
4	1Ø	7	1Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
5	1Ø	6	1Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
4	14	7	14	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
5	14	6	14	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
4	17	7	17	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
5	17	6	17	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
4	2Ø	7	2Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
5	2Ø	6	2Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	Ø	1Ø	Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	Ø	9	Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	3	1Ø	3	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	3	9	3	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	6	1Ø	6	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	6	9	6	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	1Ø	1Ø	1Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	1Ø	9	1Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	14	1Ø	14	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	14	9	14	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	17	1Ø	17	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	17	9	17	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
7	2Ø	1Ø	2Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø
8	2Ø	9	2Ø	-Ø	-Ø	6.944E+Ø4	-Ø	-Ø	-Ø

TABLE 4. JOINT STIFFNESS AND LOAD DATA CONTD

FROM JOINT	THRU JOINT	RX	RY	TX	TY

NONE

TABLE 5. MESH STIFFNESS DATA

FROM MESH		THRU MESH		C
1	1	1	2Ø	4.617E+Ø3
2	1	2	2Ø	4.617E+Ø3
3	1	3	2Ø	4.617E+Ø3
4	1	4	2Ø	4.617E+Ø3
5	1	5	2Ø	4.617E+Ø3
6	1	6	2Ø	4.617E+Ø3
7	1	7	2Ø	4.617E+Ø3
8	1	8	2Ø	4.617E+Ø3
9	1	9	2Ø	4.617E+Ø3
1Ø	1	1Ø	2Ø	4.617E+Ø3
11	1	11	2Ø	4.617E+Ø3

TABLE 6. BAR STIFFNESS DATA

FROM BAR		THRU BAR		PX	PY	PRX	PRY
1	1	1	1	-Ø	-Ø	-Ø	Ø
4	1	4	1	-Ø	-Ø	-Ø	Ø

7	1	7	1	-0	-0	-0	0
10	1	10	1	-0	-0	-0	0
1	20	1	20	-0	-0	-0	0
4	20	4	20	-0	-0	-0	0
7	20	7	20	-0	-0	-0	0
10	20	10	20	-0	-0	-0	0

TABLE 7. MULTIPLE LOAD DATA

FROM JOINT		THRU JOINT		QM
1	0	1	20	7.233E-01
1	1	1	19	7.233E-01
4	0	4	20	7.809E-01
4	1	4	19	7.809E-01
7	0	7	20	7.809E-01
7	1	7	19	7.809E-01
10	0	10	20	7.233E-01
10	1	10	19	7.233E-01
6	14	6	14	1.595E+00
6	15	6	15	1.064E+00
7	14	7	14	7.445E-01
7	15	7	15	4.964E-01
8	14	8	14	5.318E-01
8	15	8	15	3.545E-01
9	14	9	14	1.808E+00
9	15	9	15	1.205E+00
6	9	6	9	8.608E+00
6	10	6	10	2.428E+00
7	9	7	9	4.017E+00
7	10	7	10	9.466E-01
8	9	8	9	2.869E+00
8	10	8	10	6.761E-01
9	9	9	9	9.756E+00
9	10	9	10	2.299E+00
2	14	2	14	1.808E+00
2	15	2	15	1.205E+00
3	14	3	14	5.318E-01
3	15	3	15	3.545E-01
4	14	4	14	7.445E-01
4	15	4	15	4.964E-01
5	14	5	14	1.595E+00
5	15	5	15	1.064E+00
2	9	2	9	9.756E+00
2	10	2	10	2.299E+00
3	9	3	9	2.869E+00
3	10	3	10	6.761E-01
4	9	4	9	4.017E+00
4	10	4	10	9.466E-01
5	9	5	9	8.608E+00
5	10	5	10	2.428E+00

TABLE 8. PROFILE OUTPUT AREAS

FROM JOINT	THRU JOINT	DEFL (1=YES)	X MOMENT	Y MOMENT (1=SLAB,2=BEAM)	PRIN MOM OR STRESS (1=YES)

NONE

TABLE 9. PRINTED OUTPUT LIMITS

FROM Y STA	THRU Y STA

NONE

PROGRAM SLAB 49 -DEVELOPMENT DECK- MATLOCK,PANAK, ENDRES REV DATE 13 JUL 71

THIS PROGRAM IS BEING USED AT YOUR OWN RISK.
CHANGES MAY OCCUR AFTER THE ABOVE REVISION DATE.
PLEASE REPORT DIFFICULTIES TO THE ABOVE PEOPLE
AT THE CENTER FOR HIGHWAY RESEARCH, UT AT AUSTIN.

EXAMPLE PROBLEM ; RWS,CPJ ; 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ; TWO LOAD CASES

PROB (CONTD)
01 TWO STANDARD AASHTO H15 TRUCKS

RESULTS

SLAB X MOMENT AND X TWISTING MOMENT ACT IN THE X DIRECTION (ABOUT Y AXIS)
Y TWISTING MOMENT = -X TWISTING MOMENT, COUNTERCLOCKWISE BETA ANGLES ARE
POSITIVE FROM THE X AXIS TO THE DIRECTION OF THE LARGEST PRINCIPAL STRESS
SLAB MOMENTS ARE PER UNIT WIDTH

BEAM MOMENTS ARE TOTAL PER BEAM

X	Y	DEFL	SLAB X MOMENT BEAM X MOMENT	SLAB Y MOMENT BEAM Y MOMENT	SLAB X TWISTING MOMENT	LARGEST PRINCIPAL SLAB STRESS	BETA X TO LARGEST STRESS	SUPPORT REACTION
0	20	6.476E-05	-5.686E-15 0	-1.105E-14 0	3.408E-02	-7.520E-01	-45.0	0
1	20	-2.365E-07	2.073E-02 5.299E-01	3.096E-03 -7.566E-03	7.858E-02	1.956E+00	41.8	2.365E+01
2	20	-1.965E-05	1.046E-02 5.473E-01	-1.043E-14 0	8.203E-02	1.880E+00	43.2	0
3	20	-1.552E-05	4.948E-03 2.588E-01	-1.938E-14 0	7.201E-02	1.602E+00	44.0	0
4	20	-2.638E-07	-8.500E-03 -4.347E-01	-1.270E-03 3.103E-03	4.541E-02	-1.084E+00	-42.7	2.638E+01
5	20	-3.701E-06	1.528E-03 7.992E-02	2.722E-15 0	1.361E-02	3.095E-01	43.4	0
6	20	-3.701E-06	1.528E-03 7.992E-02	-3.832E-15 0	-1.361E-02	3.095E-01	-43.4	0
7	20	-2.638E-07	-8.500E-03 -4.347E-01	-1.270E-03 3.103E-03	-4.541E-02	-1.084E+00	42.7	2.638E+01
8	20	-1.552E-05	4.948E-03 2.588E-01	-8.844E-15 0	-7.201E-02	1.602E+00	-44.0	0
9	20	-1.965E-05	1.046E-02 5.473E-01	-6.592E-15 0	-8.203E-02	1.880E+00	-43.2	0
10	20	-2.365E-07	2.073E-02 5.299E-01	3.096E-03 -7.566E-03	-7.858E-02	1.956E+00	-41.8	2.365E+01
11	20	6.476E-05	-1.787E-15	9.930E-15	-3.408E-02	7.520E-01	-45.0	0

			0	0				
0	19	-5.355E-03	2.692E-14	5.988E-02	7.326E-02	2.345E+00	56.1	0
			0	0				
1	19	-5.629E-03	2.975E-02	1.116E-01	1.614E-01	5.099E+00	52.1	0
			0	6.213E+01				
2	19	-5.889E-03	4.889E-02	1.126E-01	1.643E-01	5.334E+00	54.5	0
			0	0				
3	19	-6.112E-03	7.004E-02	1.181E-01	1.415E-01	5.108E+00	49.8	0
			0	0				
4	19	-6.277E-03	9.322E-02	1.266E-01	9.004E-02	4.331E+00	54.3	0
			0	6.673E+01				
5	19	-6.358E-03	9.072E-02	1.235E-01	2.799E-02	2.999E+00	60.2	0
			0	0				
6	19	-6.358E-03	9.072E-02	1.235E-01	-2.799E-02	2.999E+00	-60.2	0
			0	0				
7	19	-6.277E-03	9.322E-02	1.266E-01	-9.004E-02	4.331E+00	-54.3	0
			0	6.673E+01				
8	19	-6.112E-03	7.004E-02	1.181E-01	-1.415E-01	5.108E+00	-49.8	0
			0	0				
9	19	-5.889E-03	4.889E-02	1.126E-01	-1.643E-01	5.334E+00	-54.5	0
			0	0				
10	19	-5.629E-03	2.975E-02	1.116E-01	-1.614E-01	5.099E+00	-52.1	0
			0	6.213E+01				
11	19	-5.355E-03	-5.307E-14	5.988E-02	-7.326E-02	2.345E+00	-56.1	0
			0	0				
0	18	-1.061E-02	-1.062E-13	1.102E-01	7.900E-02	3.272E+00	62.3	0
			0	0				
1	18	-1.111E-02	3.026E-02	2.122E-01	1.693E-01	6.737E+00	59.1	0
			0	1.204E+02				
2	18	-1.161E-02	8.024E-02	2.140E-01	1.641E-01	6.973E+00	56.1	0
			0	0				
3	18	-1.206E-02	1.323E-01	2.279E-01	1.358E-01	6.968E+00	54.7	0
			0	0				
4	18	-1.240E-02	1.927E-01	2.473E-01	8.879E-02	6.727E+00	53.5	0
			0	1.294E+02				
5	18	-1.256E-02	1.804E-01	2.381E-01	3.041E-02	5.399E+00	66.7	0
			0	0				
6	18	-1.256E-02	1.804E-01	2.381E-01	-3.041E-02	5.399E+00	-66.7	0
			0	0				
7	18	-1.240E-02	1.927E-01	2.473E-01	-8.879E-02	6.727E+00	-53.5	0
			0	1.294E+02				
8	18	-1.206E-02	1.323E-01	2.279E-01	-1.358E-01	6.968E+00	-54.7	0
			0	0				
9	18	-1.161E-02	8.024E-02	2.140E-01	-1.641E-01	6.973E+00	-56.1	0
			0	0				
10	18	-1.111E-02	3.026E-02	2.122E-01	-1.693E-01	6.737E+00	-59.1	0
			0	1.204E+02				
11	18	-1.061E-02	4.108E-14	1.102E-01	-7.900E-02	3.272E+00	-62.3	0
			0	0				
0	17	-1.557E-02	1.810E-13	1.510E-01	8.343E-02	4.041E+00	66.1	0
			0	0				
1	17	-1.631E-02	3.856E-02	3.071E-01	1.813E-01	8.565E+00	63.3	0
			-9.813E-02	1.747E+02				
2	17	-1.707E-02	1.217E-01	2.883E-01	1.669E-01	8.416E+00	58.3	0
			2.053E+00	0				
3	17	-1.773E-02	1.967E-01	3.188E-01	1.205E-01	8.445E+00	58.4	0
			3.894E+00	0				
4	17	-1.823E-02	2.695E-01	3.580E-01	9.002E-02	8.901E+00	58.1	0
			5.643E+00	1.882E+02				

5	17	-1.848E-02	2.744E-01	3.224E-01	4.087E-02	7.433E+00	60.2	0
			5.913E+00	0				
6	17	-1.848E-02	2.744E-01	3.224E-01	-4.087E-02	7.433E+00	-60.2	0
			5.913E+00	0				
7	17	-1.823E-02	2.695E-01	3.580E-01	-9.002E-02	8.901E+00	-58.1	0
			5.643E+00	1.882E+02				
8	17	-1.773E-02	1.967E-01	3.188E-01	-1.205E-01	8.445E+00	-58.4	0
			3.894E+00	0				
9	17	-1.707E-02	1.217E-01	2.883E-01	-1.669E-01	8.416E+00	-58.3	0
			2.053E+00	0				
10	17	-1.631E-02	3.856E-02	3.071E-01	-1.813E-01	8.565E+00	-63.3	0
			-9.813E-02	1.717E+02				
11	17	-1.557E-02	1.642E-13	1.510E-01	-8.343E-02	4.041E+00	-66.1	0
			0	0				
0	16	-2.011E-02	-4.409E-13	1.904E-01	8.330E-02	4.767E+00	69.4	0
			0	0				
1	16	-2.110E-02	7.190E-03	3.852E-01	1.913E-01	9.998E+00	67.3	0
			0	2.226E+02				
2	16	-2.215E-02	2.308E-01	4.055E-01	1.676E-01	1.090E+01	58.8	0
			0	0				
3	16	-2.300E-02	2.758E-01	4.345E-01	9.808E-02	1.036E+01	64.4	0
			0	0				
4	16	-2.362E-02	2.655E-01	4.533E-01	9.166E-02	1.055E+01	67.8	0
			0	2.450E+02				
5	16	-2.402E-02	4.201E-01	4.574E-01	5.414E-02	1.066E+01	54.5	0
			0	0				
6	16	-2.402E-02	4.201E-01	4.574E-01	-5.414E-02	1.066E+01	-54.5	0
			0	0				
7	16	-2.362E-02	2.655E-01	4.533E-01	-9.166E-02	1.055E+01	-67.8	0
			0	2.450E+02				
8	16	-2.300E-02	2.758E-01	4.345E-01	-9.808E-02	1.036E+01	-64.4	0
			0	0				
9	16	-2.215E-02	2.308E-01	4.055E-01	-1.676E-01	1.090E+01	-58.8	0
			0	0				
10	16	-2.110E-02	7.190E-03	3.852E-01	-1.913E-01	9.998E+00	-67.3	0
			0	2.226E+02				
11	16	-2.011E-02	2.884E-13	1.904E-01	-8.330E-02	4.767E+00	-69.4	0
			0	0				
0	15	-2.414E-02	1.013E-12	2.105E-01	7.599E-02	5.131E+00	72.3	0
			0	0				
1	15	-2.537E-02	-4.571E-02	4.524E-01	1.520E-01	1.064E+01	74.3	0
			0	2.662E+02				
2	15	-2.673E-02	4.042E-01	6.640E-01	1.397E-01	1.558E+01	66.5	0
			0	0				
3	15	-2.774E-02	3.632E-01	6.213E-01	1.131E-01	1.427E+01	69.4	0
			0	0				
4	15	-2.845E-02	2.117E-01	5.336E-01	7.204E-02	1.180E+01	77.9	0
			0	2.973E+02				
5	15	-2.902E-02	6.109E-01	7.393E-01	2.383E-02	1.598E+01	79.8	0
			0	0				
6	15	-2.902E-02	6.109E-01	7.393E-01	-2.383E-02	1.598E+01	-79.8	0
			0	0				
7	15	-2.845E-02	2.117E-01	5.336E-01	-7.204E-02	1.180E+01	-77.9	0
			0	2.973E+02				
8	15	-2.774E-02	3.632E-01	6.213E-01	-1.131E-01	1.427E+01	-69.4	0
			0	0				
9	15	-2.673E-02	4.042E-01	6.640E-01	-1.397E-01	1.558E+01	-66.5	0
			0	0				
10	15	-2.537E-02	-4.571E-02	4.524E-01	-1.520E-01	1.068E+01	-74.3	0
			0	2.662E+02				

11	15	-2.414E-02	1.394E-13	0	2.145E-01	0	-7.599E-02	5.131E+00	-72.3	0

0	14	-2.759E-02	-4.521E-13	0	2.300E-01	0	5.236E-02	5.189E+00	77.8	0
1	14	-2.901E-02	4.069E-02	-5.079E-01	5.302E-01	3.038E+02	9.469E-02	1.178E+01	79.4	0
2	14	-3.049E-02	3.062E-01	5.768E+00	5.712E-01	0	1.069E-01	1.309E+01	70.6	0
3	14	-3.171E-02	3.867E-01	7.713E+00	6.118E-01	0	1.252E-01	1.435E+01	66.0	0
4	14	-3.260E-02	4.167E-01	8.392E+00	6.386E-01	3.414E+02	5.233E-02	1.398E+01	77.4	0
5	14	-3.313E-02	5.682E-01	1.235E+01	6.407E-01	0	-6.087E-03	1.378E+01	-85.2	0
6	14	-3.313E-02	5.682E-01	1.235E+01	6.407E-01	0	6.087E-03	1.378E+01	85.2	0
7	14	-3.260E-02	4.167E-01	8.392E+00	6.386E-01	3.414E+02	-5.233E-02	1.398E+01	-77.4	0
8	14	-3.171E-02	3.867E-01	7.713E+00	6.118E-01	0	-1.252E-01	1.435E+01	-66.0	0
9	14	-3.049E-02	3.062E-01	5.768E+00	5.712E-01	0	-1.069E-01	1.309E+01	-70.6	0
10	14	-2.901E-02	4.069E-02	-5.079E-01	5.302E-01	3.038E+02	-9.469E-02	1.178E+01	-79.4	0
11	14	-2.759E-02	5.973E-14	0	2.300E-01	0	-5.236E-02	5.189E+00	-77.8	0

0	13	-3.040E-02	3.277E-13	0	2.902E-01	0	3.939E-02	6.351E+00	82.4	0
1	13	-3.194E-02	4.625E-02	0	5.759E-01	3.298E+02	8.252E-02	1.265E+01	81.3	0
2	13	-3.353E-02	2.602E-01	0	6.050E-01	0	9.250E-02	1.351E+01	75.9	0
3	13	-3.492E-02	4.115E-01	0	6.611E-01	0	9.684E-02	1.493E+01	71.1	0
4	13	-3.597E-02	5.658E-01	0	7.187E-01	3.755E+02	4.683E-02	1.573E+01	74.3	0
5	13	-3.649E-02	5.753E-01	0	7.071E-01	0	1.976E-03	1.520E+01	89.1	0
6	13	-3.649E-02	5.753E-01	0	7.071E-01	0	-1.976E-03	1.520E+01	-89.1	0
7	13	-3.597E-02	5.658E-01	0	7.187E-01	3.755E+02	-4.683E-02	1.573E+01	-74.3	0
8	13	-3.492E-02	4.115E-01	0	6.611E-01	0	-9.684E-02	1.493E+01	-71.1	0
9	13	-3.353E-02	2.602E-01	0	6.050E-01	0	-9.250E-02	1.351E+01	-75.9	0
10	13	-3.194E-02	4.625E-02	0	5.759E-01	3.298E+02	-8.252E-02	1.265E+01	-81.3	0
11	13	-3.040E-02	-2.531E-14	0	2.902E-01	0	-3.939E-02	6.351E+00	-82.4	0

0	12	-3.243E-02	-5.065E-13	0	3.183E-01	0	5.063E-02	7.012E+00	81.2	0
1	12	-3.410E-02	6.102E-02	0	6.153E-01	3.514E+02	9.493E-02	1.357E+01	80.5	0
2	12	-3.580E-02	2.581E-01	0	6.378E-01	0	7.728E-02	1.404E+01	78.9	0
3	12	-3.732E-02	4.367E-01	0	7.049E-01	0	5.903E-02	1.543E+01	78.0	0
4	12	-3.847E-02	6.554E-01		7.839E-01		3.482E-02	1.704E+01	75.8	0

			0	4.062E+02				
5	12	-3.901E-02	5.957E-01	7.575E-01	9.243E-03	1.630E+01	86.7	0
			0	0				
6	12	-3.901E-02	5.957E-01	7.575E-01	-9.243E-03	1.630E+01	-86.7	0
			0	0				
7	12	-3.847E-02	6.554E-01	7.839E-01	-3.482E-02	1.704E+01	-75.8	0
			0	4.062E+02				
8	12	-3.732E-02	4.367E-01	7.049E-01	-5.993E-02	1.543E+01	-78.0	0
			0	0				
9	12	-3.580E-02	2.581E-01	6.378E-01	-7.728E-02	1.404E+01	-78.9	0
			0	0				
10	12	-3.410E-02	6.102E-02	6.153E-01	-9.493E-02	1.357E+01	-80.5	0
			0	3.514E+02				
11	12	-3.243E-02	3.311E-13	3.183E-01	-5.063E-02	7.012E+00	-81.2	0
			0	0				
0	11	-3.359E-02	3.135E-13	3.367E-01	6.854E-02	7.526E+00	78.9	0
			0	0				
1	11	-3.543E-02	1.035E-01	6.513E-01	1.339E-01	1.467E+01	77.0	0
			0	3.686E+02				
2	11	-3.726E-02	3.018E-01	5.931E-01	7.339E-02	1.313E+01	76.6	0
			0	0				
3	11	-3.885E-02	4.599E-01	6.878E-01	-4.858E-03	1.479E+01	-88.8	0
			0	0				
4	11	-4.004E-02	6.655E-01	8.321E-01	3.008E-02	1.801E+01	79.8	0
			0	4.339E+02				
5	11	-4.062E-02	6.208E-01	6.992E-01	4.356E-02	1.545E+01	66.0	0
			0	0				
6	11	-4.062E-02	6.208E-01	6.992E-01	-4.356E-02	1.545E+01	-66.0	0
			0	0				
7	11	-4.004E-02	6.655E-01	8.321E-01	-3.008E-02	1.801E+01	-79.8	0
			0	4.339E+02				
8	11	-3.885E-02	4.599E-01	6.878E-01	4.858E-03	1.479E+01	88.8	0
			0	0				
9	11	-3.726E-02	3.018E-01	5.931E-01	-7.339E-02	1.313E+01	-76.6	0
			0	0				
10	11	-3.543E-02	1.035E-01	6.513E-01	-1.339E-01	1.467E+01	-77.0	0
			0	3.686E+02				
11	11	-3.359E-02	-3.730E-14	3.367E-01	-6.854E-02	7.526E+00	-78.9	0
			0	0				
0	10	-3.384E-02	6.748E-13	3.088E-01	7.139E-02	6.976E+00	77.6	0
			0	0				
1	10	-3.590E-02	8.248E-02	6.711E-01	2.665E-01	1.664E+01	68.9	0
			-2.378E-01	3.819E+02				
2	10	-3.707E-02	4.950E-01	2.627E-01	1.280E-01	1.186E+01	23.9	0
			1.192E+01	0				
3	10	-3.954E-02	5.124E-01	5.241E-01	-1.893E-01	1.521E+01	-45.9	0
			1.134E+01	0				
4	10	-4.061E-02	3.786E-01	8.322E-01	9.100E-02	1.827E+01	79.1	0
			6.636E+00	4.594E+02				
5	10	-4.140E-02	7.507E-01	3.215E-01	2.092E-01	1.797E+01	22.1	0
			1.837E+01	0				
6	10	-4.140E-02	7.507E-01	3.215E-01	-2.092E-01	1.797E+01	-22.1	0
			1.837E+01	0				
7	10	-4.061E-02	3.786E-01	8.322E-01	-9.100E-02	1.827E+01	-79.1	0
			6.636E+00	4.594E+02				
8	10	-3.954E-02	5.124E-01	5.241E-01	1.893E-01	1.521E+01	45.9	0
			1.134E+01	0				
9	10	-3.797E-02	4.950E-01	2.627E-01	-1.280E-01	1.186E+01	-23.9	0
			1.192E+01	0				
10	10	-3.590E-02	8.248E-02	6.711E-01	-2.665E-01	1.664E+01	-68.9	0

			-2.378E-01	3.819E+02				
11	10	-3.384E-02	-3.078E-13	3.088E-01	-7.139E-02	6.976E+00	-77.6	0
			0	0				
0	9	-3.325E-02	-2.525E-13	2.016E-01	-1.210E-03	4.334E+00	-89.7	0
			0	0				
1	9	-3.547E-02	-5.756E-01	5.685E-01	3.739E-02	-1.240E+01	-1.9	0
			0	3.796E+02				
2	9	-3.843E-02	1.892E+00	2.075E+00	2.316E-03	4.460E+01	89.3	0
			0	0				
3	9	-3.961E-02	8.220E-01	1.515E+00	-8.861E-02	3.282E+01	-82.8	0
			0	0				
4	9	-4.013E-02	-1.021E+00	6.296E-01	1.063E-02	-2.195E+01	-.4	0
			0	4.637E+02				
5	9	-4.190E-02	1.912E+00	2.253E+00	6.155E-02	4.867E+01	80.1	0
			0	0				
6	9	-4.190E-02	1.912E+00	2.253E+00	-6.155E-02	4.867E+01	-80.1	0
			0	0				
7	9	-4.013E-02	-1.021E+00	6.296E-01	-1.063E-02	-2.195E+01	.4	0
			0	4.637E+02				
8	9	-3.961E-02	8.220E-01	1.515E+00	8.861E-02	3.282E+01	82.8	0
			0	0				
9	9	-3.843E-02	1.892E+00	2.075E+00	-2.316E-03	4.460E+01	-89.3	0
			0	0				
10	9	-3.547E-02	-5.756E-01	5.685E-01	-3.739E-02	-1.240E+01	1.9	0
			0	3.796E+02				
11	9	-3.325E-02	-4.154E-13	2.016E-01	1.210E-03	4.334E+00	89.7	0
			0	0				
0	8	-3.210E-02	4.163E-13	2.613E-01	-8.877E-02	6.294E+00	-72.9	0
			0	0				
1	8	-3.414E-02	-1.441E-01	5.987E-01	-2.939E-01	1.507E+01	-70.8	0
			0	3.596E+02				
2	8	-3.645E-02	9.481E-01	6.626E-01	-1.784E-01	2.222E+01	-25.7	0
			0	0				
3	8	-3.780E-02	6.102E-01	7.303E-01	1.037E-01	1.698E+01	60.0	0
			0	0				
4	8	-3.858E-02	-2.114E-01	6.925E-01	-1.144E-01	1.519E+01	-82.9	0
			0	4.291E+02				
5	8	-3.972E-02	1.127E+00	7.565E-01	-1.888E-01	2.594E+01	-22.8	0
			0	0				
6	8	-3.972E-02	1.127E+00	7.565E-01	1.888E-01	2.594E+01	22.8	0
			0	0				
7	8	-3.858E-02	-2.114E-01	6.925E-01	1.144E-01	1.519E+01	82.9	0
			0	4.291E+02				
8	8	-3.780E-02	6.102E-01	7.303E-01	-1.037E-01	1.698E+01	-60.0	0
			0	0				
9	8	-3.645E-02	9.481E-01	6.626E-01	1.784E-01	2.222E+01	25.7	0
			0	0				
10	8	-3.414E-02	-1.441E-01	5.987E-01	2.939E-01	1.507E+01	70.8	0
			0	3.596E+02				
11	8	-3.210E-02	-3.635E-13	2.613E-01	8.877E-02	6.294E+00	72.9	0
			0	0				
0	7	-3.025E-02	6.126E-13	2.894E-01	-1.090E-01	7.006E+00	-71.5	0
			0	0				
1	7	-3.198E-02	4.110E-02	5.809E-01	-2.632E-01	1.479E+01	-67.9	0
			0	3.332E+02				
2	7	-3.376E-02	4.594E-01	5.216E-01	-1.723E-01	1.431E+01	-50.1	0
			0	0				
3	7	-3.511E-02	4.568E-01	5.994E-01	7.785E-03	1.289E+01	86.9	0
			0	0				

4	7	-3.605E-02	3.207E-01	6.981E-01	-9.127E-02	1.546E+01	-77.1	0
			0	3.851E+02				
5	7	-3.675E-02	7.077E-01	5.914E-01	-1.167E-01	1.677E+01	-31.8	0
			0	0				
6	7	-3.675E-02	7.077E-01	5.914E-01	1.167E-01	1.677E+01	31.8	0
			0	0				
7	7	-3.605E-02	3.207E-01	6.981E-01	9.127E-02	1.546E+01	77.1	0
			0	3.851E+02				
8	7	-3.511E-02	4.568E-01	5.994E-01	-7.785E-03	1.289E+01	-86.9	0
			0	0				
9	7	-3.376E-02	4.594E-01	5.216E-01	1.723E-01	1.431E+01	50.1	0
			0	0				
10	7	-3.198E-02	4.110E-02	5.809E-01	2.632E-01	1.479E+01	67.9	0
			0	3.332E+02				
11	7	-3.025E-02	1.859E-13	2.894E-01	1.090E-01	7.006E+00	71.5	0
			0	0				
0	6	-2.761E-02	3.355E-13	2.766E-01	-9.854E-02	6.624E+00	-72.3	0
			0	0				
1	6	-2.903E-02	7.921E-02	5.336E-01	-2.047E-01	1.316E+01	-69.0	0
			-1.078E-02	3.024E+02				
2	6	-3.045E-02	2.260E-01	4.686E-01	-1.539E-01	1.168E+01	-64.1	0
			4.072E+00	0				
3	6	-3.170E-02	3.643E-01	5.436E-01	-7.146E-02	1.222E+01	-70.7	0
			7.396E+00	0				
4	6	-3.263E-02	5.004E-01	6.439E-01	-7.627E-02	1.455E+01	-66.6	0
			1.056E+01	3.370E+02				
5	6	-3.311E-02	5.056E-01	5.477E-01	-5.411E-02	1.257E+01	-55.6	0
			1.108E+01	0				
6	6	-3.311E-02	5.056E-01	5.477E-01	5.411E-02	1.257E+01	55.6	0
			1.108E+01	0				
7	6	-3.263E-02	5.004E-01	6.439E-01	7.627E-02	1.455E+01	66.6	0
			1.056E+01	3.370E+02				
8	6	-3.170E-02	3.643E-01	5.436E-01	7.146E-02	1.222E+01	70.7	0
			7.396E+00	0				
9	6	-3.045E-02	2.260E-01	4.686E-01	1.539E-01	1.168E+01	64.1	0
			4.072E+00	0				
10	6	-2.903E-02	7.921E-02	5.336E-01	2.047E-01	1.316E+01	69.0	0
			-1.078E-02	3.024E+02				
11	6	-2.761E-02	4.685E-13	2.766E-01	9.854E-02	6.624E+00	72.3	0
			0	0				
0	5	-2.421E-02	1.106E-13	2.393E-01	-8.434E-02	5.718E+00	-72.4	0
			0	0				
1	5	-2.537E-02	4.652E-02	4.614E-01	-1.682E-01	1.120E+01	-70.5	0
			0	2.634E+02				
2	5	-2.656E-02	1.717E-01	4.703E-01	-1.479E-01	1.142E+01	-67.6	0
			0	0				
3	5	-2.763E-02	3.058E-01	5.080E-01	-1.139E-01	1.202E+01	-65.8	0
			0	0				
4	5	-2.844E-02	4.612E-01	5.570E-01	-7.503E-02	1.286E+01	-61.3	0
			0	2.899E+02				
5	5	-2.883E-02	4.259E-01	5.379E-01	-2.763E-02	1.170E+01	-76.9	0
			0	0				
6	5	-2.883E-02	4.259E-01	5.379E-01	2.763E-02	1.170E+01	76.9	0
			0	0				
7	5	-2.844E-02	4.612E-01	5.570E-01	7.503E-02	1.286E+01	61.3	0
			0	2.899E+02				
8	5	-2.763E-02	3.058E-01	5.080E-01	1.139E-01	1.202E+01	65.8	0
			0	0				
9	5	-2.656E-02	1.717E-01	4.703E-01	1.479E-01	1.142E+01	67.6	0
			0	0				

10	5	-2.537E-02	4.652E-02 0	4.614E-01 2.634E+02	1.682E-01	1.120E+01	70.5	0
11	5	-2.421E-02	6.373E-14 0	2.393E-01 0	8.434E-02	5.718E+00	72.4	0
0	4	-2.017E-02	-4.098E-13 0	1.962E-01 0	-7.512E-02	4.766E+00	-71.3	0
1	4	-2.110E-02	3.027E-02 0	3.845E-01 2.203E+02	-1.577E-01	9.556E+00	-69.2	0
2	4	-2.206E-02	1.403E-01 0	4.036E-01 0	-1.526E-01	1.018E+01	-65.4	0
3	4	-2.293E-02	2.507E-01 0	4.293E-01 0	-1.286E-01	1.068E+01	-62.4	0
4	4	-2.359E-02	3.722E-01 0	4.569E-01 2.376E+02	-8.071E-02	1.087E+01	-58.8	0
5	4	-2.391E-02	3.521E-01 0	4.564E-01 0	-2.493E-02	9.933E+00	-77.2	0
6	4	-2.391E-02	3.521E-01 0	4.564E-01 0	2.493E-02	9.933E+00	77.2	0
7	4	-2.359E-02	3.722E-01 0	4.569E-01 2.376E+02	8.071E-02	1.087E+01	58.8	0
8	4	-2.293E-02	2.507E-01 0	4.293E-01 0	1.286E-01	1.068E+01	62.4	0
9	4	-2.206E-02	1.403E-01 0	4.036E-01 0	1.526E-01	1.018E+01	65.4	0
10	4	-2.110E-02	3.027E-02 0	3.845E-01 2.203E+02	1.577E-01	9.556E+00	69.2	0
11	4	-2.017E-02	-1.221E-14 0	1.962E-01 0	7.512E-02	4.766E+00	71.3	0
0	3	-1.559E-02	-1.383E-13 0	1.430E-01 0	-7.527E-02	3.768E+00	-66.8	0
1	3	-1.630E-02	3.161E-02 -1.810E-01	3.030E-01 1.729E+02	-1.600E-01	8.106E+00	-65.2	0
2	3	-1.703E-02	1.149E-01 1.738E+00	3.226E-01 0	-1.577E-01	8.763E+00	-61.7	0
3	3	-1.769E-02	1.937E-01 3.744E+00	3.370E-01 0	-1.337E-01	8.965E+00	-59.1	0
4	3	-1.819E-02	2.739E-01 5.795E+00	3.491E-01 1.825E+02	-8.522E-02	8.699E+00	-56.9	0
5	3	-1.843E-02	2.725E-01 5.730E+00	3.559E-01 0	-2.721E-02	7.824E+00	-73.4	0
6	3	-1.843E-02	2.725E-01 5.730E+00	3.559E-01 0	2.721E-02	7.824E+00	73.4	0
7	3	-1.819E-02	2.739E-01 5.795E+00	3.491E-01 1.825E+02	8.522E-02	8.699E+00	56.9	0
8	3	-1.769E-02	1.937E-01 3.744E+00	3.370E-01 0	1.337E-01	8.965E+00	59.1	0
9	3	-1.703E-02	1.149E-01 1.738E+00	3.226E-01 0	1.577E-01	8.763E+00	61.7	0
10	3	-1.630E-02	3.161E-02 -1.810E-01	3.030E-01 1.729E+02	1.600E-01	8.106E+00	65.2	0
11	3	-1.559E-02	-7.938E-14 0	1.430E-01 0	7.527E-02	3.768E+00	66.8	0
0	2	-1.062E-02	-2.282E-13 0	1.070E-01 0	-7.635E-02	3.155E+00	-62.5	0
1	2	-1.110E-02	2.492E-02 0	2.094E-01 1.192E+02	-1.626E-01	6.538E+00	-59.8	0
2	2	-1.160E-02	8.016E-02 0	2.177E-01 0	-1.607E-01	6.959E+00	-56.6	0
3	2	-1.204E-02	1.312E-01	2.282E-01	-1.359E-01	6.964E+00	-54.8	0

			0	0				
4	2	-1.237E-02	1.838E-01	2.397E-01	-8.777E-02	6.532E+00	-53.8	0
			0	1.257E+02				
5	2	-1.253E-02	1.819E-01	2.394E-01	-2.887E-02	5.404E+00	-67.4	0
			0	0				
6	2	-1.253E-02	1.819E-01	2.394E-01	2.887E-02	5.404E+00	67.4	0
			0	0				
7	2	-1.237E-02	1.838E-01	2.397E-01	8.777E-02	6.532E+00	53.8	0
			0	1.257E+02				
8	2	-1.204E-02	1.312E-01	2.282E-01	1.359E-01	6.964E+00	54.8	0
			0	0				
9	2	-1.160E-02	8.016E-02	2.177E-01	1.607E-01	6.959E+00	56.6	0
			0	0				
10	2	-1.110E-02	2.492E-02	2.094E-01	1.626E-01	6.538E+00	59.8	0
			0	1.192E+02				
11	2	-1.062E-02	2.047E-13	1.070E-01	7.635E-02	3.155E+00	62.5	0
			0	0				
0	1	-5.357E-03	1.697E-13	5.859E-02	-7.105E-02	2.282E+00	-56.2	0
			0	0				
1	1	-5.624E-03	2.751E-02	1.103E-01	-1.575E-01	4.983E+00	-52.4	0
			0	6.156E+01				
2	1	-5.879E-03	4.882E-02	1.116E-01	-1.613E-01	5.256E+00	-50.5	0
			0	0				
3	1	-6.097E-03	6.915E-02	1.160E-01	-1.387E-01	5.013E+00	-49.8	0
			0	0				
4	1	-6.258E-03	8.806E-02	1.227E-01	-8.895E-02	4.214E+00	-50.5	0
			0	6.490E+01				
5	1	-6.340E-03	9.109E-02	1.206E-01	-2.824E-02	2.960E+00	-58.8	0
			0	0				
6	1	-6.340E-03	9.109E-02	1.206E-01	2.824E-02	2.960E+00	58.8	0
			0	0				
7	1	-6.258E-03	8.806E-02	1.227E-01	8.895E-02	4.214E+00	50.5	0
			0	6.490E+01				
8	1	-6.097E-03	6.915E-02	1.160E-01	1.387E-01	5.013E+00	49.8	0
			0	0				
9	1	-5.879E-03	4.882E-02	1.116E-01	1.613E-01	5.256E+00	50.5	0
			0	0				
10	1	-5.624E-03	2.751E-02	1.103E-01	1.575E-01	4.983E+00	52.4	0
			0	6.156E+01				
11	1	-5.357E-03	4.929E-14	5.859E-02	7.105E-02	2.282E+00	56.2	0
			0	0				
0	0	6.314E-05	-1.514E-15	-2.271E-16	-3.406E-02	-7.321E-01	45.0	0
			0	0				
1	0	-2.344E-07	2.016E-02	3.011E-03	-7.685E-02	1.911E+00	-41.8	2.344E+01
			5.155E-01	-7.364E-03				
2	0	-1.926E-05	1.026E-02	7.376E-15	-8.039E-02	1.842E+00	-43.2	0
			5.368E-01	0				
3	0	-1.521E-05	4.842E-03	8.001E-15	-7.033E-02	1.565E+00	-44.0	0
			2.533E-01	0				
4	0	-2.569E-07	-8.397E-03	-1.254E-03	-4.475E-02	-1.069E+00	42.7	2.569E+01
			-4.294E-01	3.066E-03				
5	0	-3.775E-06	1.564E-03	-8.040E-15	-1.375E-02	3.129E-01	-43.4	0
			8.180E-02	0				
6	0	-3.775E-06	1.564E-03	2.699E-15	1.375E-02	3.129E-01	43.4	0
			8.180E-02	0				
7	0	-2.569E-07	-8.397E-03	-1.254E-03	4.475E-02	-1.069E+00	-42.7	2.569E+01
			-4.294E-01	3.066E-03				
8	0	-1.521E-05	4.842E-03	5.398E-16	7.033E-02	1.565E+00	44.0	0
			2.533E-01	0				
9	0	-1.926E-05	1.026E-02	-3.199E-15	8.039E-02	1.842E+00	43.2	0

```
                              5.368E-01            0
10   0 -2.344E-07  2.016E-02  3.011E-03  7.685E-02  1.911E+00  41.8  2.344E+01
                              5.155E-01 -7.360E-03
11   0  6.314E-05 -3.944E-16 -5.916E-17  3.406E-02 -7.321E-01 -45.0          0
                                      0            0


STATICS CHECK.            SUMMATION OF REACTIONS =     1.983E+02

                          MAXIMUM STATICS CHECK ERROR AT STA   2  Q = -1.951E+01


                           TIME FOR THIS PROBLEM =      0 MINUTES   11.669 SECONDS


                           ELAPSED     TIME =     0 MINUTES   45.746 SECONDS
```

PROGRAM SLAB 49 -DEVELOPMENT DECK- MATLOCK,PANAK, ENDRES REV DATE 13 JUL 71

```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
.                                                       .
.  THIS PROGRAM IS BEING USED AT YOUR OWN RISK.         .
.  CHANGES MAY OCCUR AFTER THE ABOVE REVISION DATE.     .
.  PLEASE REPORT DIFFICULTIES TO THE ABOVE PEOPLE       .
.  AT THE CENTER FOR HIGHWAY RESEARCH, UT AT AUSTIN.    .
.                                                       .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

EXAMPLE PROBLEM ; RWS,CPJ ; 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ; TWO LOAD CASES

PROB
02 ONE STANDARD AASHTO H20 FROM THE DATA BASE

TABLE 1. CONTROL DATA

	TABLE NUMBER 2	3	4	5	6	7	8	9
KEEP FROM PRECEDING PROBLEM (1=YES)	1	1	0	1	1	0	0	0
NUM CARDS INPUT THIS PROBLEM	0	0	0	0	0	24	0	0

MULTIPLE LOAD OPTION	-1
STATICS CHECK OPTION	0
PRIN STRESS OPTION	1
PROFILE PLOT OPTION	0
3-D PLOT OPTION	0

TABLE 2. CONSTANTS
USING DATA FROM THE PREVIOUS PROBLEM

NUMBER OF INCREMENTS IN X DIRECTION	11
NUMBER OF INCREMENTS IN Y DIRECTION	20
INCREMENT LENGTH IN X DIRECTION	2.444E+00
INCREMENT LENGTH IN Y DIRECTION	2.688E+00
POISSONS RATIO	1.500E-01
SLAB THICKNESS	5.283E-01

TABLE 3. JOINT STIFFNESS AND LOAD DATA

USING DATA FROM THE PREVIOUS PROBLEM

FROM JOINT		THRU JOINT		DX	DY	FX	FY	Q	S
0	0	0	20	1.358E+03	1.358E+03	-0	-0	-0	-0
0	1	0	19	1.358E+03	1.358E+03	-0	-0	-0	-0
1	0	1	20	2.716E+03	2.716E+03	-0	-0	-0	-0
1	1	1	19	2.716E+03	2.716E+03	-0	-0	-0	-0
2	0	2	20	2.716E+03	2.716E+03	-0	-0	-0	-0
2	1	2	19	2.716E+03	2.716E+03	-0	-0	-0	-0
3	0	3	20	2.716E+03	2.716E+03	-0	-0	-0	-0
3	1	3	19	2.716E+03	2.716E+03	-0	-0	-0	-0

4	0	4	20	2.716E+03	2.716E+03	-0	-0	-0	-0
4	1	4	19	2.716E+03	2.716E+03	-0	-0	-0	-0
5	0	5	20	2.716E+03	2.716E+03	-0	-0	-0	-0
5	1	5	19	2.716E+03	2.716E+03	-0	-0	-0	-0
6	0	6	20	2.716E+03	2.716E+03	-0	-0	-0	-0
6	1	6	19	2.716E+03	2.716E+03	-0	-0	-0	-0
7	0	7	20	2.716E+03	2.716E+03	-0	-0	-0	-0
7	1	7	19	2.716E+03	2.716E+03	-0	-0	-0	-0
8	0	8	20	2.716E+03	2.716E+03	-0	-0	-0	-0
8	1	8	19	2.716E+03	2.716E+03	-0	-0	-0	-0
9	0	9	20	2.716E+03	2.716E+03	-0	-0	-0	-0
9	1	9	19	2.716E+03	2.716E+03	-0	-0	-0	-0
10	0	10	20	2.716E+03	2.716E+03	-0	-0	-0	-0
10	1	10	19	2.716E+03	2.716E+03	-0	-0	-0	-0
11	0	11	20	1.358E+03	1.358E+03	-0	-0	-0	-0
11	1	11	19	1.358E+03	1.358E+03	-0	-0	-0	-0
1	0	1	20	-0	-0	-0	4.510E+05	-0	-0
1	1	1	19	-0	-0	-0	4.510E+05	-0	-0
4	0	4	20	-0	-0	-0	4.510E+05	-0	-0
4	1	4	19	-0	-0	-0	4.510E+05	-0	-0
7	0	7	20	-0	-0	-0	4.510E+05	-0	-0
7	1	7	19	-0	-0	-0	4.510E+05	-0	-0
10	0	10	20	-0	-0	-0	4.510E+05	-0	-0
10	1	10	19	-0	-0	-0	4.510E+05	-0	-0
1	0	1	20	-0	-0	-0	1.088E+06	-0	-0
1	1	1	19	-0	-0	-0	1.088E+06	-0	-0
4	0	4	20	-0	-0	-0	1.122E+06	-0	-0
4	1	4	19	-0	-0	-0	1.122E+06	-0	-0
7	0	7	20	-0	-0	-0	1.122E+06	-0	-0
7	1	7	19	-0	-0	-0	1.122E+06	-0	-0
10	0	10	20	-0	-0	-0	1.088E+06	-0	-0
10	1	10	19	-0	-0	-0	1.088E+06	-0	-0
1	0	1	0	-0	-0	-0	-0	-0	1.000E+08
1	1	1	1	-0	-0	-0	-0	-0	0
4	0	4	0	-0	-0	-0	-0	-0	1.000E+08
4	1	4	1	-0	-0	-0	-0	-0	0
7	0	7	0	-0	-0	-0	-0	-0	1.000E+08
7	1	7	1	-0	-0	-0	-0	-0	0
10	0	10	0	-0	-0	-0	-0	-0	1.000E+08
10	1	10	1	-0	-0	-0	-0	-0	0
1	20	1	20	-0	-0	-0	-0	-0	1.000E+08
1	20	1	20	-0	-0	-0	-0	-0	0
4	20	4	20	-0	-0	-0	-0	-0	1.000E+08
4	20	4	20	-0	-0	-0	-0	-0	0
7	20	7	20	-0	-0	-0	-0	-0	1.000E+08
7	20	7	20	-0	-0	-0	-0	-0	0
10	20	10	20	-0	-0	-0	-0	-0	1.000E+08
10	20	10	20	-0	-0	-0	-0	-0	0
1	0	4	0	-0	-0	6.944E+04	-0	-0	-0
2	0	3	0	-0	-0	6.944E+04	-0	-0	-0
1	3	4	3	-0	-0	6.944E+04	-0	-0	-0
2	3	3	3	-0	-0	6.944E+04	-0	-0	-0
1	6	4	6	-0	-0	6.944E+04	-0	-0	-0
2	6	3	6	-0	-0	6.944E+04	-0	-0	-0
1	10	4	10	-0	-0	6.944E+04	-0	-0	-0
2	10	3	10	-0	-0	6.944E+04	-0	-0	-0
1	14	4	14	-0	-0	6.944E+04	-0	-0	-0
2	14	3	14	-0	-0	6.944E+04	-0	-0	-0
1	17	4	17	-0	-0	6.944E+04	-0	-0	-0
2	17	3	17	-0	-0	6.944E+04	-0	-0	-0
1	20	4	20	-0	-0	6.944E+04	-0	-0	-0
2	20	3	20	-0	-0	6.944E+04	-0	-0	-0

4	0	7	0	-0	-0	6.944E+04	-0	-0	-0
5	0	6	0	-0	-0	6.944E+04	-0	-0	-0
4	3	7	3	-0	-0	6.944E+04	-0	-0	-0
5	3	6	3	-0	-0	6.944E+04	-0	-0	-0
4	6	7	6	-0	-0	6.944E+04	-0	-0	-0
5	6	6	6	-0	-0	6.944E+04	-0	-0	-0
4	10	7	10	-0	-0	6.944E+04	-0	-0	-0
5	10	6	10	-0	-0	6.944E+04	-0	-0	-0
4	14	7	14	-0	-0	6.944E+04	-0	-0	-0
5	14	6	14	-0	-0	6.944E+04	-0	-0	-0
4	17	7	17	-0	-0	6.944E+04	-0	-0	-0
5	17	6	17	-0	-0	6.944E+04	-0	-0	-0
4	20	7	20	-0	-0	6.944E+04	-0	-0	-0
5	20	6	20	-0	-0	6.944E+04	-0	-0	-0
7	0	10	0	-0	-0	6.944E+04	-0	-0	-0
8	0	9	0	-0	-0	6.944E+04	-0	-0	-0
7	3	10	3	-0	-0	6.944E+04	-0	-0	-0
8	3	9	3	-0	-0	6.944E+04	-0	-0	-0
7	6	10	6	-0	-0	6.944E+04	-0	-0	-0
8	6	9	6	-0	-0	6.944E+04	-0	-0	-0
7	10	10	10	-0	-0	6.944E+04	-0	-0	-0
8	10	9	10	-0	-0	6.944E+04	-0	-0	-0
7	14	10	14	-0	-0	6.944E+04	-0	-0	-0
8	14	9	14	-0	-0	6.944E+04	-0	-0	-0
7	17	10	17	-0	-0	6.944E+04	-0	-0	-0
8	17	9	17	-0	-0	6.944E+04	-0	-0	-0
7	20	10	20	-0	-0	6.944E+04	-0	-0	-0
8	20	9	20	-0	-0	6.944E+04	-0	-0	-0

ADDITIONAL DATA FOR THIS PROBLEM

NONE

TABLE 4. JOINT STIFFNESS AND LOAD DATA CONTD

FROM JOINT	THRU JOINT	RX	RY	TX	TY

NONE

TABLE 5. MESH STIFFNESS DATA

USING DATA FROM THE PREVIOUS PROBLEM

	FROM MESH	THRU MESH		C
1	1	1	20	4.617E+03
2	1	2	20	4.617E+03
3	1	3	20	4.617E+03
4	1	4	20	4.617E+03
5	1	5	20	4.617E+03
6	1	6	20	4.617E+03
7	1	7	20	4.617E+03
8	1	8	20	4.617E+03
9	1	9	20	4.617E+03
10	1	10	20	4.617E+03
11	1	11	20	4.617E+03

ADDITIONAL DATA FOR THIS PROBLEM

NONE

TABLE 6. BAR STIFFNESS DATA

FROM BAR		THRU BAR		PX	PY	PBX	PBY
USING DATA FROM THE PREVIOUS PROBLEM							
1	1	1	1	-0	-0	-0	0
4	1	4	1	-0	-0	-0	0
7	1	7	1	-0	-0	-0	0
10	1	10	1	-0	-0	-0	0
1	20	1	20	-0	-0	-0	0
4	20	4	20	-0	-0	-0	0
7	20	7	20	-0	-0	-0	0
10	20	10	20	-0	-0	-0	0
ADDITIONAL DATA FOR THIS PROBLEM							

NONE

TABLE 7. MULTIPLE LOAD DATA

FROM JOINT		THRU JOINT		QM
1	0	1	20	7.233E-01
1	1	1	19	7.233E-01
4	0	4	20	7.809E-01
4	1	4	19	7.809E-01
7	0	7	20	7.809E-01
7	1	7	19	7.809E-01
10	0	10	20	7.233E-01
10	1	10	19	7.233E-01
4	3	4	3	1.425E+01
4	4	4	4	8.753E-01
5	3	5	3	5.345E+00
5	4	5	4	3.282E-01
6	3	6	3	5.345E+00
6	4	6	4	3.282E-01
7	3	7	3	1.425E+01
7	4	7	4	8.753E-01
4	8	4	8	1.109E+01
4	9	4	9	4.041E+00
5	8	5	8	4.157E+00
5	9	5	9	1.516E+00
6	8	6	8	4.157E+00
6	9	6	9	1.516E+00
7	8	7	8	1.109E+01
7	9	7	9	4.041E+00

TABLE 8. PROFILE OUTPUT AREAS

FROM JOINT	THRU JOINT	DEFL (1=YES)	X MOMENT	Y MOMENT (1=SLAB,2=BEAM)	PRIN MOM OR STRESS (1=YES)

NONE

TABLE 9. PRINTED OUTPUT LIMITS

FROM Y STA	THRU Y STA	
		NONE

PROGRAM SLAB 49 -DEVELOPMENT DECK- MATLOCK,PANAK, ENDRES REV DATE 13 JUL 71

THIS PROGRAM IS BEING USED AT YOUR OWN RISK.
CHANGES MAY OCCUR AFTER THE ABOVE REVISION DATE.
PLEASE REPORT DIFFICULTIES TO THE ABOVE PEOPLE
AT THE CENTER FOR HIGHWAY RESEARCH, UT AT AUSTIN.

EXAMPLE PROBLEM ; RWS,CPJ ; 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ; TWO LOAD CASES

PROB (CONTD)
02 ONE STANDARD AASHTO H20 FROM THE DATA BASE

RESULTS--USING STIFFNESS DATA FROM PREVIOUS PROBLEM 01

SLAB X MOMENT AND X TWISTING MOMENT ACT IN THE X DIRECTION (ABOUT Y AXIS)
Y TWISTING MOMENT = -X TWISTING MOMENT, COUNTERCLOCKWISE BETA ANGLES ARE
POSITIVE FROM THE X AXIS TO THE DIRECTION OF THE LARGEST PRINCIPAL STRESS
SLAB MOMENTS ARE PER UNIT WIDTH

BEAM MOMENTS ARE TOTAL PER BEAM

X	Y	DEFL	SLAB X MOMENT BEAM X MOMENT	SLAB Y MOMENT BEAM Y MOMENT	SLAB X TWISTING MOMENT	LARGEST PRINCIPAL SLAB STRESS	BETA X TO LARGEST STRESS	SUPPORT REACTION
0	20	7.237E-05	-3.027E-15	-1.065E-14	3.874E-02	-8.328E-01	-45.0	0
			0	0				
1	20	-2.067E-07	2.368E-02	3.538E-03	8.585E-02	2.151E+00	41.7	2.067E+01
			6.056E-01	-8.646E-03				
2	20	-2.069E-05	1.112E-02	-1.368E-14	8.862E-02	2.028E+00	43.2	0
			5.818E-01	0				
3	20	-1.616E-05	5.073E-03	-3.723E-14	7.680E-02	1.707E+00	44.1	0
			2.654E-01	0				
4	20	-2.120E-07	-8.472E-03	-1.266E-03	4.807E-02	-1.160E+00	-42.9	2.120E+01
			-4.333E-01	3.493E-03				
5	20	-2.898E-06	1.194E-03	-1.758E-14	1.527E-02	3.413E-01	43.9	0
			6.245E-02	0				
6	20	-2.898E-06	1.194E-03	4.613E-15	-1.527E-02	3.413E-01	-43.9	0
			6.245E-02	0				
7	20	-2.120E-07	-8.472E-03	-1.266E-03	-4.807E-02	-1.160E+00	42.9	2.120E+01
			-4.333E-01	3.093E-03				
8	20	-1.616E-05	5.073E-03	-1.297E-14	-7.680E-02	1.707E+00	-44.1	0
			2.654E-01	0				
9	20	-2.069E-05	1.112E-02	-3.463E-15	-8.862E-02	2.028E+00	-43.2	0
			5.818E-01	0				
10	20	-2.067E-07	2.368E-02	3.538E-03	-8.585E-02	2.151E+00	-41.7	2.067E+01
			6.056E-01	-8.646E-03				
11	20	7.237E-05	-2.576E-15	9.812E-15	-3.874E-02	8.328E-01	-45.0	0

			0	0				
0	19	-4.527E-03	3.814E-14	5.371E-02	8.110E-02	2.414E+00	54.2	0
			0	0				
1	19	-4.829E-03	3.286E-02	9.858E-02	1.768E-01	5.279E+00	50.3	0
			0	5.429E+01				
2	19	-5.111E-03	5.214E-02	9.717E-02	1.790E-01	5.483E+00	48.6	0
			0	0				
3	19	-5.351E-03	7.234E-02	9.862E-02	1.528E-01	5.136E+00	47.5	0
			0	0				
4	19	-5.525E-03	9.098E-02	1.028E-01	9.812E-02	4.197E+00	46.7	0
			0	5.285E+01				
5	19	-5.615E-03	9.486E-02	1.004E-01	3.143E-02	2.773E+00	47.4	0
			0	0				
6	19	-5.615E-03	9.486E-02	1.000E-01	-3.143E-02	2.773E+00	-47.4	0
			0	0				
7	19	-5.525E-03	9.098E-02	1.028E-01	-9.812E-02	4.197E+00	-46.7	0
			0	5.285E+01				
8	19	-5.351E-03	7.234E-02	9.862E-02	-1.528E-01	5.136E+00	-47.5	0
			0	0				
9	19	-5.111E-03	5.214E-02	9.717E-02	-1.790E-01	5.483E+00	-48.6	0
			0	0				
10	19	-4.829E-03	3.286E-02	9.858E-02	-1.768E-01	5.279E+00	-50.3	0
			0	5.429E+01				
11	19	-4.527E-03	-5.324E-14	5.371E-02	-8.110E-02	2.414E+00	-54.2	0
			0	0				
0	18	-8.980E-03	-4.635E-14	9.663E-02	8.798E-02	3.196E+00	59.4	0
			0	0				
1	18	-9.531E-03	2.904E-02	1.850E-01	1.855E-01	6.627E+00	56.4	0
			0	1.047E+02				
2	18	-1.008E-02	8.429E-02	1.875E-01	1.824E-01	6.997E+00	52.9	0
			0	0				
3	18	-1.057E-02	1.375E-01	1.928E-01	1.537E-01	6.997E+00	50.1	0
			0	0				
4	18	-1.093E-02	1.919E-01	2.002E-01	9.934E-02	6.352E+00	46.2	0
			0	1.016E+02				
5	18	-1.111E-02	1.911E-01	1.977E-01	3.282E-02	4.888E+00	47.9	0
			0	0				
6	18	-1.111E-02	1.911E-01	1.977E-01	-3.282E-02	4.888E+00	-47.9	0
			0	0				
7	18	-1.093E-02	1.919E-01	2.002E-01	-9.934E-02	6.352E+00	-46.2	0
			0	1.016E+02				
8	18	-1.057E-02	1.375E-01	1.928E-01	-1.537E-01	6.997E+00	-50.1	0
			0	0				
9	18	-1.008E-02	8.429E-02	1.875E-01	-1.824E-01	6.997E+00	-52.9	0
			0	0				
10	18	-9.531E-03	2.904E-02	1.850E-01	-1.855E-01	6.627E+00	-56.4	0
			0	1.047E+02				
11	18	-8.980E-03	-6.584E-14	9.663E-02	-8.798E-02	3.196E+00	-59.4	0
			0	0				
0	17	-1.317E-02	-1.883E-13	1.310E-01	9.138E-02	3.825E+00	62.8	0
			0	0				
1	17	-1.399E-02	3.263E-02	2.656E-01	1.914E-01	8.023E+00	60.7	0
			-9.431E-02	1.511E+02				
2	17	-1.481E-02	1.196E-01	2.738E-01	1.869E-01	8.573E+00	56.2	0
			2.054E+00	0				
3	17	-1.555E-02	2.043E-01	2.814E-01	1.566E-01	8.688E+00	51.9	0
			4.239E+00	0				
4	17	-1.610E-02	2.896E-01	2.904E-01	1.013E-01	8.411E+00	45.1	0
			6.435E+00	1.463E+02				

5	17	-1.638E-02	2.893E-01	2.892E-01	3.367E-02	6.942E+00	45.0	0
			6.432E+00	0				
6	17	-1.638E-02	2.893E-01	2.892E-01	-3.367E-02	6.942E+00	-45.0	0
			6.432E+00	0				
7	17	-1.610E-02	2.896E-01	2.904E-01	-1.013E-01	8.411E+00	-45.1	0
			6.435E+00	1.463E+02				
8	17	-1.555E-02	2.043E-01	2.814E-01	-1.566E-01	8.688E+00	-51.9	0
			4.239E+00	0				
9	17	-1.481E-02	1.196E-01	2.738E-01	-1.869E-01	8.573E+00	-56.2	0
			2.054E+00	0				
10	17	-1.399E-02	3.263E-02	2.656E-01	-1.914E-01	8.023E+00	-60.7	0
			-9.431E-02	1.511E+02				
11	17	-1.317E-02	-1.887E-14	1.310E-01	-9.138E-02	3.825E+00	-62.8	0
			0	0				
0	16	-1.700E-02	-4.007E-14	1.709E-01	9.471E-02	4.578E+00	66.0	0
			0	0				
1	16	-1.809E-02	2.994E-02	3.342E-01	1.972E-01	9.268E+00	63.8	0
			0	1.911E+02				
2	16	-1.919E-02	1.526E-01	3.464E-01	1.913E-01	9.075E+00	58.4	0
			0	0				
3	16	-2.019E-02	2.712E-01	3.616E-01	1.596E-01	1.037E+01	52.9	0
			0	0				
4	16	-2.093E-02	3.042E-01	3.784E-01	1.030E-01	1.048E+01	43.4	0
			0	1.895E+02				
5	16	-2.131E-02	3.883E-01	3.768E-01	3.426E-02	8.970E+00	40.2	0
			0	0				
6	16	-2.131E-02	3.883E-01	3.768E-01	-3.426E-02	8.970E+00	-40.2	0
			0	0				
7	16	-2.093E-02	3.042E-01	3.784E-01	-1.030E-01	1.048E+01	-43.4	0
			0	1.895E+02				
8	16	-2.019E-02	2.712E-01	3.616E-01	-1.596E-01	1.037E+01	-52.9	0
			0	0				
9	16	-1.919E-02	1.526E-01	3.464E-01	-1.913E-01	9.075E+00	-58.4	0
			0	0				
10	16	-1.809E-02	2.994E-02	3.342E-01	-1.972E-01	9.268E+00	-63.8	0
			0	1.911E+02				
11	16	-1.700E-02	-7.883E-14	1.709E-01	-9.471E-02	4.578E+00	-66.0	0
			0	0				
0	15	-2.037E-02	2.003E-13	1.998E-01	1.006E-01	5.196E+00	67.4	0
			0	0				
1	15	-2.174E-02	3.341E-02	3.964E-01	2.043E-01	1.050E+01	65.8	0
			0	2.269E+02				
2	15	-2.314E-02	1.896E-01	4.166E-01	1.945E-01	1.136E+01	60.1	0
			0	0				
3	15	-2.439E-02	3.396E-01	4.379E-01	1.609E-01	1.197E+01	53.5	0
			0	0				
4	15	-2.533E-02	4.896E-01	4.597E-01	1.038E-01	1.246E+01	40.9	0
			0	2.289E+02				
5	15	-2.580E-02	4.881E-01	4.592E-01	3.464E-02	1.099E+01	33.7	0
			0	0				
6	15	-2.580E-02	4.881E-01	4.592E-01	-3.464E-02	1.099E+01	-33.7	0
			0	0				
7	15	-2.533E-02	4.896E-01	4.597E-01	-1.038E-01	1.246E+01	-40.9	0
			0	2.289E+02				
8	15	-2.439E-02	3.396E-01	4.379E-01	-1.609E-01	1.197E+01	-53.5	0
			0	0				
9	15	-2.314E-02	1.896E-01	4.166E-01	-1.945E-01	1.136E+01	-60.1	0
			0	0				
10	15	-2.174E-02	3.341E-02	3.964E-01	-2.043E-01	1.050E+01	-65.8	0
			0	2.269E+02				

11	15	-2.037E-02	1.533E-13 0	1.998E-01 0	-1.006E-01	5.196E+00	-67.4	0

0	14	-2.320E-02	7.572E-14 0	2.058E-01 0	9.418E-02	5.211E+00	68.8	0
1	14	-2.486E-02	4.191E-02 -3.387E-01	4.521E-01 2.584E+02	1.966E-01	1.142E+01	68.1	0
2	14	-2.655E-02	2.296E-01 4.084E+00	4.809E-01 0	1.941E-01	1.276E+01	61.9	0
3	14	-2.806E-02	4.084E-01 8.668E+00	5.132E-01 0	1.635E-01	1.360E+01	53.9	0
4	14	-2.921E-02	5.884E-01 1.329E+01	5.347E-01 2.645E+02	1.042E-01	1.439E+01	37.8	0
5	14	-2.978E-02	5.879E-01 1.326E+01	5.393E-01 0	3.358E-02	1.301E+01	27.1	0
6	14	-2.978E-02	5.879E-01 1.326E+01	5.393E-01 0	-3.358E-02	1.301E+01	-27.1	0
7	14	-2.921E-02	5.884E-01 1.329E+01	5.347E-01 2.645E+02	-1.042E-01	1.439E+01	-37.8	0
8	14	-2.806E-02	4.084E-01 8.668E+00	5.132E-01 0	-1.635E-01	1.360E+01	-53.9	0
9	14	-2.655E-02	2.296E-01 4.084E+00	4.809E-01 0	-1.941E-01	1.276E+01	-61.9	0
10	14	-2.486E-02	4.191E-02 -3.387E-01	4.521E-01 2.584E+02	-1.966E-01	1.142E+01	-68.1	0
11	14	-2.320E-02	-4.152E-14 0	2.058E-01 0	-9.418E-02	5.211E+00	-68.8	0

0	13	-2.546E-02	-2.156E-13 0	2.417E-01 0	8.564E-02	5.782E+00	72.3	0
1	13	-2.737E-02	1.758E-02 0	4.867E-01 2.806E+02	1.841E-01	1.183E+01	70.9	0
2	13	-2.934E-02	2.487E-01 0	5.292E-01 0	1.884E-01	1.341E+01	63.3	0
3	13	-3.113E-02	4.724E-01 0	5.728E-01 0	1.616E-01	1.487E+01	53.6	0
4	13	-3.247E-02	7.002E-01 0	6.139E-01 3.015E+02	1.017E-01	1.650E+01	33.5	0
5	13	-3.314E-02	6.814E-01 0	6.155E-01 0	3.141E-02	1.492E+01	21.8	0
6	13	-3.314E-02	6.814E-01 0	6.155E-01 0	-3.141E-02	1.492E+01	-21.8	0
7	13	-3.247E-02	7.002E-01 0	6.139E-01 3.015E+02	-1.017E-01	1.650E+01	-33.5	0
8	13	-3.113E-02	4.724E-01 0	5.728E-01 0	-1.616E-01	1.487E+01	-53.6	0
9	13	-2.934E-02	2.487E-01 0	5.292E-01 0	-1.884E-01	1.341E+01	-63.3	0
10	13	-2.737E-02	1.758E-02 0	4.867E-01 2.806E+02	-1.841E-01	1.183E+01	-70.9	0
11	13	-2.546E-02	-1.443E-13 0	2.417E-01 0	-8.564E-02	5.782E+00	-72.3	0

0	12	-2.707E-02	-1.483E-13 0	2.509E-01 0	8.540E-02	6.039E+00	73.1	0
1	12	-2.923E-02	1.046E-03 0	5.106E-01 2.982E+02	1.767E-01	1.224E+01	72.7	0
2	12	-3.147E-02	2.708E-01 0	5.732E-01 0	1.707E-01	1.404E+01	65.4	0
3	12	-3.350E-02	5.330E-01 0	6.335E-01 0	1.407E-01	1.593E+01	54.3	0
4	12	-3.505E-02	8.079E-01	6.868E-01	9.678E-02	1.852E+01	29.0	0

			0	3.351E+02				
5	12	-3.580E-02	7.676E-01	6.780E-01	3.103E-02	1.671E+01	17.3	0
			0	0				
6	12	-3.580E-02	7.676E-01	6.780E-01	-3.103E-02	1.671E+01	-17.3	0
			0	0				
7	12	-3.505E-02	8.079E-01	6.868E-01	-9.678E-02	1.852E+01	-29.0	0
			0	3.351E+02				
8	12	-3.350E-02	5.330E-01	6.335E-01	-1.497E-01	1.593E+01	-54.3	0
			0	0				
9	12	-3.147E-02	2.708E-01	5.732E-01	-1.747E-01	1.404E+01	-65.4	0
			0	0				
10	12	-2.923E-02	1.046E-03	5.146E-01	-1.767E-01	1.224E+01	-72.7	0
			0	2.982E+02				
11	12	-2.707E-02	5.451E-13	2.549E-01	-8.540E-02	6.039E+00	-73.1	0
			0	0				
0	11	-2.798E-02	2.489E-13	2.574E-01	8.081E-02	6.034E+00	73.9	0
			0	0				
1	11	-3.038E-02	1.973E-04	5.366E-01	1.566E-01	1.245E+01	74.9	0
			0	3.111E+02				
2	11	-3.287E-02	2.969E-01	6.206E-01	1.477E-01	1.457E+01	68.8	0
			0	0				
3	11	-3.513E-02	5.833E-01	6.979E-01	1.320E-01	1.686E+01	56.7	0
			0	0				
4	11	-3.685E-02	8.965E-01	7.514E-01	1.070E-01	2.049E+01	27.9	0
			0	3.655E+02				
5	11	-3.769E-02	8.490E-01	6.859E-01	4.565E-02	1.851E+01	14.6	0
			0	0				
6	11	-3.769E-02	8.490E-01	6.859E-01	-4.565E-02	1.851E+01	-14.6	0
			0	0				
7	11	-3.685E-02	8.965E-01	7.514E-01	-1.070E-01	2.049E+01	-27.9	0
			0	3.655E+02				
8	11	-3.513E-02	5.833E-01	6.979E-01	-1.320E-01	1.686E+01	-56.7	0
			0	0				
9	11	-3.287E-02	2.969E-01	6.206E-01	-1.477E-01	1.457E+01	-68.8	0
			0	0				
10	11	-3.038E-02	1.973E-04	5.366E-01	-1.566E-01	1.245E+01	-74.9	0
			0	3.111E+02				
11	11	-2.798E-02	-4.083E-13	2.574E-01	-8.081E-02	6.034E+00	-73.9	0
			0	0				
0	10	-2.820E-02	-6.279E-13	2.139E-01	4.503E-02	4.795E+00	78.6	0
			4	0				
1	10	-3.081E-02	1.997E-02	5.534E-01	9.048E-02	1.222E+01	84.6	0
			-8.245E-01	3.191E+02				
2	10	-3.349E-02	3.203E-01	6.912E-01	9.402E-02	1.535E+01	76.4	0
			5.665E+00	0				
3	10	-3.592E-02	6.039E-01	7.881E-01	1.174E-01	1.817E+01	64.1	0
			1.270E+01	0				
4	10	-3.781E-02	9.040E-01	7.992E-01	1.982E-01	2.271E+01	37.6	0
			2.051E+01	3.932E+02				
5	10	-3.882E-02	9.740E-01	5.137E-01	1.304E-01	2.168E+01	14.8	0
			2.346E+01	0				
6	10	-3.882E-02	9.740E-01	5.137E-01	-1.304E-01	2.168E+01	-14.8	0
			2.346E+01	0				
7	10	-3.781E-02	9.040E-01	7.992E-01	-1.982E-01	2.271E+01	-37.6	0
			2.051E+01	3.932E+02				
8	10	-3.592E-02	6.039E-01	7.881E-01	-1.174E-01	1.817E+01	-64.1	0
			1.270E+01	0				
9	10	-3.349E-02	3.203E-01	6.912E-01	-9.402E-02	1.535E+01	-76.4	0
			5.665E+00	0				
10	10	-3.081E-02	1.997E-02	5.534E-01	-9.048E-02	1.222E+01	-80.6	0

			-8.245E-01	3.191E+02				
11	10	-2.820E-02	3.024E-13	2.139E-01	-4.503E-02	4.795E+00	-78.6	0
			0	0				
0	9	-2.783E-02	-9.133E-13	2.595E-01	8.884E-03	5.585E+00	88.0	0
			0	0				
1	9	-3.008E-02	-4.625E-04	5.426E-01	2.547E-02	1.169E+01	87.3	0
			0	3.146E+02				
2	9	-3.323E-02	2.755E-01	6.169E-01	4.034E-02	1.336E+01	83.4	0
			0	0				
3	9	-3.577E-02	4.905E-01	6.974E-01	7.899E-02	1.557E+01	71.3	0
			0	0				
4	9	-3.788E-02	5.882E-01	8.040E-01	2.123E-01	2.008E+01	58.5	0
			0	4.241E+02				
5	9	-3.946E-02	1.575E+00	1.131E+00	1.583E-01	3.495E+01	17.7	0
			0	0				
6	9	-3.946E-02	1.575E+00	1.131E+00	-1.583E-01	3.495E+01	-17.7	0
			0	0				
7	9	-3.788E-02	5.882E-01	8.040E-01	-2.123E-01	2.008E+01	-58.5	0
			0	4.241E+02				
8	9	-3.577E-02	4.905E-01	6.974E-01	-7.899E-02	1.557E+01	-71.3	0
			0	0				
9	9	-3.323E-02	2.755E-01	6.169E-01	-4.034E-02	1.336E+01	-83.4	0
			0	0				
10	9	-3.008E-02	-4.625E-04	5.426E-01	-2.547E-02	1.169E+01	-87.3	0
			0	3.146E+02				
11	9	-2.783E-02	-7.128E-14	2.595E-01	-8.884E-03	5.585E+00	-88.0	0
			0	0				
0	8	-2.675E-02	-1.370E-13	2.558E-01	1.635E-03	5.499E+00	89.6	0
			0	0				
1	8	-2.942E-02	-1.069E-02	5.251E-01	3.782E-03	1.129E+01	89.6	0
			0	3.053E+02				
2	8	-3.219E-02	2.554E-01	5.844E-01	5.513E-03	1.256E+01	89.0	0
			0	0				
3	8	-3.477E-02	4.374E-01	6.568E-01	-3.654E-04	1.412E+01	-89.9	0
			0	0				
4	8	-3.696E-02	3.755E-01	7.938E-01	-3.636E-02	1.713E+01	-85.1	0
			0	4.369E+02				
5	8	-3.887E-02	1.946E+00	1.652E+00	-3.340E-02	4.191E+01	-6.4	0
			0	0				
6	8	-3.887E-02	1.946E+00	1.652E+00	3.340E-02	4.191E+01	6.4	0
			0	0				
7	8	-3.696E-02	3.755E-01	7.938E-01	3.636E-02	1.713E+01	85.1	0
			0	4.369E+02				
8	8	-3.477E-02	4.374E-01	6.568E-01	3.654E-04	1.412E+01	89.9	0
			0	0				
9	8	-3.219E-02	2.554E-01	5.844E-01	-5.513E-03	1.256E+01	-89.0	0
			0	0				
10	8	-2.942E-02	-1.069E-02	5.251E-01	-3.782E-03	1.129E+01	-89.6	0
			0	3.053E+02				
11	8	-2.675E-02	-5.085E-14	2.558E-01	-1.635E-03	5.499E+00	-89.6	0
			0	0				
0	7	-2.498E-02	3.109E-15	2.333E-01	-8.394E-03	5.022E+00	-87.9	0
			0	0				
1	7	-2.764E-02	-1.729E-02	5.000E-01	-2.578E-02	1.078E+01	-87.2	0
			0	2.914E+02				
2	7	-3.004E-02	2.742E-01	5.973E-01	-3.823E-02	1.294E+01	-83.3	0
			0	0				
3	7	-3.296E-02	5.188E-01	6.989E-01	-7.805E-02	1.565E+01	-69.5	0
			0	0				

4	7	-3.505E-02	7.417E-01 0	8.006E-01 4.084E+02	-2.167E-01	2.128E+01	-48.9	0
5	7	-3.644E-02	1.358E+00 0	8.102E-01 0	-1.625E-01	3.016E+01	-15.3	0
6	7	-3.644E-02	1.358E+00 0	8.102E-01 0	1.625E-01	3.016E+01	15.3	0
7	7	-3.505E-02	7.417E-01 0	8.006E-01 4.084E+02	2.167E-01	2.128E+01	48.9	0
8	7	-3.296E-02	5.188E-01 0	6.989E-01 0	7.805E-02	1.565E+01	69.5	0
9	7	-3.040E-02	2.742E-01 0	5.973E-01 0	3.823E-02	1.294E+01	83.3	0
10	7	-2.764E-02	-1.729E-02 0	5.000E-01 2.914E+02	2.578E-02	1.078E+01	87.2	0
11	7	-2.498E-02	2.536E-13 0	2.333E-01 0	8.394E-03	5.022E+00	87.9	0
0	6	-2.257E-02	-1.904E-13 0	1.560E-01 0	-5.633E-02	3.745E+00	-72.1	0
1	6	-2.517E-02	-3.472E-03 -9.666E-01	4.696E-01 2.725E+02	-1.152E-01	1.067E+01	-77.0	0
2	6	-2.786E-02	3.117E-01 5.630E+00	6.432E-01 0	-1.142E-01	1.459E+01	-72.7	0
3	6	-3.030E-02	6.070E-01 1.289E+01	7.604E-01 0	-1.188E-01	1.774E+01	-61.4	0
4	6	-3.219E-02	9.198E-01 2.103E+01	7.716E-01 3.755E+02	-1.480E-01	2.174E+01	-31.7	0
5	6	-3.318E-02	9.567E-01 2.291E+01	5.398E-01 0	-8.502E-02	2.093E+01	-11.2	0
6	6	-3.318E-02	9.567E-01 2.291E+01	5.398E-01 0	8.502E-02	2.093E+01	11.2	0
7	6	-3.219E-02	9.198E-01 2.103E+01	7.716E-01 3.755E+02	1.480E-01	2.174E+01	31.7	0
8	6	-3.030E-02	6.070E-01 1.289E+01	7.604E-01 0	1.188E-01	1.774E+01	61.4	0
9	6	-2.786E-02	3.117E-01 5.630E+00	6.432E-01 0	1.142E-01	1.459E+01	72.7	0
10	6	-2.517E-02	-3.472E-03 -9.666E-01	4.696E-01 2.725E+02	1.152E-01	1.067E+01	77.0	0
11	6	-2.257E-02	2.609E-13 0	1.560E-01 0	5.633E-02	3.745E+00	72.1	0
0	5	-1.974E-02	-2.740E-13 0	1.853E-01 0	-1.064E-01	5.024E+00	-65.5	0
1	5	-2.207E-02	-3.236E-02 0	4.105E-01 2.498E+02	-2.111E-01	1.064E+01	-68.2	0
2	5	-2.451E-02	2.601E-01 0	5.264E-01 0	-1.950E-01	1.353E+01	-62.2	0
3	5	-2.674E-02	5.359E-01 0	6.347E-01 0	-1.512E-01	1.600E+01	-54.0	0
4	5	-2.848E-02	8.494E-01 0	7.132E-01 3.471E+02	-7.541E-02	1.898E+01	-24.0	0
5	5	-2.938E-02	9.008E-01 0	6.590E-01 0	-1.438E-02	1.938E+01	-3.4	0
6	5	-2.938E-02	9.008E-01 0	6.590E-01 0	1.438E-02	1.938E+01	3.4	0
7	5	-2.848E-02	8.494E-01 0	7.132E-01 3.471E+02	7.541E-02	1.898E+01	24.0	0
8	5	-2.674E-02	5.359E-01 0	6.347E-01 0	1.512E-01	1.600E+01	54.0	0
9	5	-2.451E-02	2.601E-01 0	5.264E-01 0	1.950E-01	1.353E+01	62.2	0

10	5	-2.207E-02	-3.236E-02 0	4.105E-01 2.408E+02	2.111E-01	1.064E+01	68.2	0
11	5	-1.974E-02	-4.141E-14 0	1.853E-01 0	1.064E-01	5.024E+00	65.5	0
0	4	-1.640E-02	9.059E-14 0	1.598E-01 0	-1.222E-01	4.855E+00	-61.6	0
1	4	-1.840E-02	-2.874E-02 0	3.477E-01 2.040E+02	-2.605E-01	1.034E+01	-62.9	0
2	4	-2.049E-02	2.070E-01 0	4.404E-01 0	-2.575E-01	1.304E+01	-57.2	0
3	4	-2.242E-02	4.262E-01 0	5.383E-01 0	-2.043E-01	1.492E+01	-52.7	0
4	4	-2.396E-02	6.628E-01 0	6.324E-01 3.158E+02	-9.100E-02	1.592E+01	-40.3	0
5	4	-2.487E-02	9.095E-01 0	7.102E-01 0	-9.317E-03	1.956E+01	-2.7	0
6	4	-2.487E-02	9.095E-01 0	7.102E-01 0	9.317E-03	1.956E+01	2.7	0
7	4	-2.396E-02	6.628E-01 0	6.324E-01 3.158E+02	9.100E-02	1.592E+01	40.3	0
8	4	-2.242E-02	4.262E-01 0	5.383E-01 0	2.043E-01	1.492E+01	52.7	0
9	4	-2.049E-02	2.070E-01 0	4.404E-01 0	2.575E-01	1.304E+01	57.2	0
10	4	-1.840E-02	-2.874E-02 0	3.477E-01 2.040E+02	2.605E-01	1.034E+01	62.9	0
11	4	-1.640E-02	-2.797E-13 0	1.598E-01 0	1.222E-01	4.855E+00	61.6	0
0	3	-1.263E-02	-4.529E-13 0	1.007E-01 0	-1.403E-01	4.470E+00	-54.3	0
1	3	-1.425E-02	4.496E-03 -4.925E-01	2.810E-01 1.625E+02	-3.137E-01	1.044E+01	-56.9	0
2	3	-1.592E-02	1.637E-01 2.876E+00	3.587E-01 0	-3.125E-01	1.265E+01	-53.7	0
3	3	-1.746E-02	3.077E-01 6.346E+00	4.339E-01 0	-2.662E-01	1.385E+01	-51.7	0
4	3	-1.872E-02	4.523E-01 9.723E+00	5.372E-01 2.781E+02	-1.792E-01	1.460E+01	-51.7	0
5	3	-1.957E-02	8.725E-01 1.974E+01	7.845E-01 0	-6.312E-02	1.946E+01	-27.6	0
6	3	-1.957E-02	8.725E-01 1.974E+01	7.845E-01 0	6.312E-02	1.946E+01	27.6	0
7	3	-1.872E-02	4.523E-01 9.723E+00	5.372E-01 2.781E+02	1.792E-01	1.460E+01	51.7	0
8	3	-1.746E-02	3.077E-01 6.346E+00	4.339E-01 0	2.662E-01	1.385E+01	51.7	0
9	3	-1.592E-02	1.637E-01 2.876E+00	3.587E-01 0	3.125E-01	1.265E+01	53.7	0
10	3	-1.425E-02	4.496E-03 -4.925E-01	2.810E-01 1.625E+02	3.137E-01	1.044E+01	56.9	0
11	3	-1.263E-02	-2.115E-14 0	1.007E-01 0	1.403E-01	4.470E+00	54.3	0
0	2	-8.587E-03	5.490E-14 0	9.196E-02 0	-1.713E-01	4.801E+00	-52.5	0
1	2	-9.724E-03	4.414E-03 0	1.949E-01 1.126E+02	-3.569E-01	1.008E+01	-52.5	0
2	2	-1.089E-02	1.162E-01 0	2.496E-01 0	-3.583E-01	1.177E+01	-50.3	0
3	2	-1.197E-02	2.203E-01	3.116E-01	-3.195E-01	1.266E+01	-49.1	0

			0	0				
4	2	-1.285E-02	3.558E-01	3.730E-01	-2.488E-01	1.319E+01	-46.0	0
			0	1.894E+02				
5	2	-1.339E-02	5.414E-01	3.850E-01	-1.034E-01	1.274E+01	-26.4	0
			0	0				
6	2	-1.339E-02	5.414E-01	3.850E-01	1.034E-01	1.274E+01	26.4	0
			0	0				
7	2	-1.285E-02	3.558E-01	3.730E-01	2.488E-01	1.319E+01	46.0	0
			0	1.894E+02				
8	2	-1.197E-02	2.203E-01	3.116E-01	3.105E-01	1.266E+01	40.1	0
			0	0				
9	2	-1.089E-02	1.162E-01	2.496E-01	3.583E-01	1.177E+01	59.3	0
			0	0				
10	2	-9.724E-03	4.414E-03	1.949E-01	3.569E-01	1.008E+01	52.5	0
			0	1.126E+02				
11	2	-8.587E-03	6.573E-14	9.196E-02	1.713E-01	4.801E+00	52.5	0
			0	0				
0	1	-4.291E-03	3.081E-14	5.619E-02	-1.695E-01	4.298E+00	-49.7	0
			0	0				
1	1	-4.931E-03	3.741E-02	1.062E-01	-3.704E-01	9.539E+00	-47.7	0
			0	5.830E+01				
2	1	-5.546E-03	7.771E-02	1.320E-01	-3.890E-01	1.064E+01	-47.0	0
			0	0				
3	1	-6.095E-03	1.224E-01	1.628E-01	-3.506E-01	1.061E+01	-46.6	0
			0	0				
4	1	-6.535E-03	1.878E-01	1.913E-01	-2.533E-01	9.520E+00	-45.2	0
			0	9.667E+01				
5	1	-6.795E-03	2.582E-01	1.766E-01	-9.355E-02	6.867E+00	-33.2	0
			0	0				
6	1	-6.795E-03	2.582E-01	1.766E-01	9.355E-02	6.867E+00	33.2	0
			0	0				
7	1	-6.535E-03	1.878E-01	1.913E-01	2.533E-01	9.520E+00	45.2	0
			0	9.667E+01				
8	1	-6.095E-03	1.224E-01	1.628E-01	3.506E-01	1.061E+01	46.6	0
			0	0				
9	1	-5.546E-03	7.771E-02	1.320E-01	3.890E-01	1.064E+01	47.0	0
			0	0				
10	1	-4.931E-03	3.741E-02	1.062E-01	3.704E-01	9.539E+00	47.7	0
			0	5.830E+01				
11	1	-4.291E-03	3.036E-14	5.619E-02	1.695E-01	4.298E+00	49.7	0
			0	0				
0	0	1.567E-04	-2.648E-15	-3.972E-16	-8.305E-02	-1.785E+00	45.0	0
			0	0				
1	0	-2.160E-07	5.420E-02	8.096E-03	-1.854E-01	4.685E+00	-41.5	2.160E+01
			1.386E+00	-1.979E-02				
2	0	-3.794E-05	2.113E-02	3.830E-15	-1.995E-01	4.521E+00	-43.5	0
			1.105E+00	0				
3	0	-2.813E-05	7.973E-03	-4.531E-15	-1.812E-01	3.982E+00	-44.4	0
			4.171E-01	0				
4	0	-3.831E-07	-1.790E-02	-2.674E-03	-1.260E-01	-2.934E+00	43.3	3.831E+01
			-9.154E-01	6.535E-03				
5	0	-1.201E-05	5.167E-03	-7.931E-15	-4.356E-02	9.936E-01	-43.3	0
			2.703E-01	0				
6	0	-1.201E-05	5.167E-03	-1.065E-15	4.356E-02	9.936E-01	43.3	0
			2.703E-01	0				
7	0	-3.831E-07	-1.790E-02	-2.674E-03	1.260E-01	-2.934E+00	-43.3	3.831E+01
			-9.154E-01	6.535E-03				
8	0	-2.813E-05	7.973E-03	6.030E-15	1.812E-01	3.982E+00	44.4	0
			4.171E-01	0				
9	0	-3.794E-05	2.113E-02	-3.241E-15	1.995E-01	4.521E+00	43.5	0

```
                          1.105E+00            0
10    0 -2.160E-07  5.420E-02  8.096E-03  1.854E-01  4.685E+00  41.5  2.160E+01
                          1.386E+00 -1.979E-02
11    0  1.567E-04 -3.155E-15 -4.733E-16  8.305E-02 -1.785E+00 -45.0          0
                                  0            0


STATICS CHECK:          SUMMATION OF REACTIONS =     2.035E+02

                        MAXIMUM STATICS CHECK ERROR AT STA   7  3 = -3.162E+01


                         TIME FOR THIS PROBLEM =      0 MINUTES     5.101 SECONDS


                         ELAPSED      TIME =      0 MINUTES   50.847 SECONDS
```

```
PROGRAM SLAB 49  -DEVELOPMENT  DECK-  MATLOCK,PANAK, ENDRES  REV DATE 13 JUL 71

        . . . . . . . . . . . . . . . . . . . . . . . . . . . .
        .                                                     .
        .   THIS PROGRAM IS BEING USED AT YOUR OWN RISK.      .
        .   CHANGES MAY OCCUR AFTER THE ABOVE REVISION DATE.  .
        .   PLEASE REPORT DIFFICULTIES TO THE ABOVE PEOPLE    .
        .   AT THE CENTER FOR HIGHWAY RESEARCH, UT AT AUSTIN. .
        .                                                     .
        . . . . . . . . . . . . . . . . . . . . . . . . . . . .

EXAMPLE PROBLEM ; RWS.CPJ ; 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER ;  TWO LOAD CASES


                        ELAPSED     TIME =     0 MINUTES   50.847 SECONDS


KEEP RUN TIME RECORDS FOR FUTURE ESTIMATES OF PARENT AND OFFSPRING RUN TIMES
```

```
02 OCT 80 UNIVERSITY OF TEXAS 170/750B UT2D-215

CEIZ551-022

19.23.34  OUTPUT DISPOSED.
19.23.34 +   3000CM     1.046CP  3022MS      0MT
19.23.34 REWIND PLOT
19.23.34 +  12300CM     1.047CP  3029MS      0MT
19.23.34 COPYZ PLOT PLOTR
19.23.43 -   1100CM     8.344CP  3554MS      0MT
19.23.43 DISPOSE PLOTR/57
19.23.43  PLOTR DISPOSED.
19.23.43 +   3000CM     8.347CP  3677MS      0MT
19.23.43 REWALLX
19.23.43 +  20000CM     8.350CP  3679MS      0MT
19.23.43 EXECPF,1503,HWYLIB,X/SLAB49,TAPE13.
19.23.43 -  10000CM     8.385CP  3819MS      0MT
19.23.43 +  20000CM     8.432CP  3868MS      0MT
19.23.44 -  10000CM     8.461CP  3994MS      0MT
19.23.44 +  20000CM     8.500CP  4036MS      0MT
19.23.47 + 124000CM     9.042CP  5460MS      0MT
19.23.48 SLAB49,TAPE13.
19.24.32  ..END -- SB498
19.24.32 0 124000CM    11.812CP 10007MS      0MT
19.24.32  ..XTIME =   18.980 SECS. FL =  124000B
19.24.32  ** PROGRAM SLAB49 EXECUTED WITH PARAMETERS TAPE13.
19.24.32  ** BINARY STORED IN FILE SLAB49   -  HWYLIB
19.24.32 -  10000CM    11.812CP 10015MS      0MT
19.24.32 PRINT OUTPUT ID=57 PR=57
19.24.33  OUTPUT PRINTED.
```

APPENDIX K

OUTPUT FROM THE POST-PROCESSOR (SEARCH)

```
EXAMPLE PROBLEM : RWS.CPJ : 10/02/80
PRESTRESSED GIRDER BRIDGE , TYPE C GIRDER :  TWO LOAD CASES


PROBLEM
   01      TWO STANDARD AASHTO H15 TRUCKS


CONTROL DATA

   NUMBER OF PROBLEMS INPUT (LOAD CASES)                          2
   NUMBER OF GIRDER OUTPUTS                                       2
   DATA OUTPUT (1 = DATA ONLY,2 = DATA+PLOT)                      2
   PRINT SLAB49 RESULTS (1 = YES)                                 0


CONSTANTS

   NUMBER OF INCREMENTS ACROSS BRIDGE                            11
   NUMBER OF INCREMENTS ALONG BRIDGE                             20
   INCREMENT LENGTH ACROSS BRIDGE          (FT)          2.4444
   INCREMENT LENGTH ALONG BRIDGE           (FT)          2.6875
   SLAB THICKNESS                          (IN)          6.3400
   NUMBER OF GIRDERS IN BRIDGE                                    4
   GIRDER TYPE                                          PRES
   GIRDER NAME                                          TYPE C
   GIRDER - TOTAL DEPTH                    (FT)          2.8333
   GIRDER - CENTROID TO BOTTOM             (FT)          1.2500
   GIRDER - MOMENT OF INERTIA            (FT-4)          2.0882
   COMPOSITE SECT - CENTROID TO BOTTOM     (FT)          2.3717
   COMPOSITE SECT - MOMENT OF INERTIA    (FT-4)          7.2834
   MODULUS OF ELASTICITY - CONCRETE       (PSI)       3.000E+06
   MODULUS OF ELASTICITY - STEEL          (PSI)       2.900E+07
```

COMPUTED RESULTANT STRESSES

THESE STRESSES ARE COMPUTED BY METHODS DESCRIBED IN
CHFR REPORT 56 - 25 FOR SLAB49

THESE STRESSES DO NOT INCLUDE THE STRESSES INDUCED BY PRESTRESSING

STRESSES ALONG GIRDER NUMBER = 2

X	Y	DEFLECTION (IN)	TRANSVERSE SLAB STRESS (PSI)	LONGITUDINAL SLAB STRESS (PSI)	GIRDER TOP STRESS (PSI)	GIRDER BOTTOM STRESS (PSI)
4	0	-,000	1,253	,701	-,001	,007
4	1	-,075	-13,145	-68,637	-28,565	146,760
4	2	-,148	-27,436	-134,085	-55,326	284,248
4	3	-,218	-40,885	-195,282	-80,326	412,692
4	4	-,283	-55,558	-255,584	-104,578	537,290
4	5	-,341	-68,843	-311,579	-127,202	653,522
4	6	-,392	-74,695	-360,190	-148,328	762,066
4	7	-,433	-47,871	-390,509	-169,499	870,836
4	8	-,463	31,556	-387,376	-188,866	970,334
4	9	-,482	152,405	-352,191	-204,095	1048,576
4	10	-,487	-56,514	-465,523	-202,202	1038,852
4	11	-,480	-99,339	-465,467	-190,978	981,188
4	12	-,462	-97,832	-438,504	-178,786	918,550
4	13	-,432	-84,457	-402,032	-165,274	849,127
4	14	-,391	-62,201	-357,225	-150,265	772,016
4	15	-,341	-31,600	-298,489	-130,855	672,291
4	16	-,283	-39,631	-253,571	-107,835	554,024
4	17	-,219	-40,228	-200,261	-82,835	425,581
4	18	-,149	-28,764	-138,337	-56,955	292,615
4	19	-,075	-13,915	-70,819	-29,371	150,898
4	20	-.000	1.269	.710	-.001	.007

STRESSES ALONG GIRDER NUMBER = 3

X	Y	DEFLECTION (IN)	TRANSVERSE SLAB STRESS (PSI)	LONGITUDINAL SLAB STRESS (PSI)	GIRDER TOP STRESS (PSI)	GIRDER BOTTOM STRESS (PSI)
7	6	-,392	-74,695	-360,190	-148,328	762,066
7	7	-,433	-47,871	-390,509	-169,499	870,836
7	8	-,463	31,556	-387,376	-188,866	970,334
7	9	-,482	152,405	-352,191	-204,095	1048,576
7	10	-,487	-56,514	-465,523	-202,202	1038,852
7	11	-,480	-99,339	-465,467	-190,978	981,188
7	12	-,462	-97,832	-438,504	-178,786	918,550
7	13	-.432	-84.457	-402.032	-165.274	849.127

MAXIMUM STRESSES COMPUTED

X	Y	MAXIMUM STRESS (PSI)	LOCATION IN GIRDER	DISTANCE ALONG GIRDER (FT)	GIRDER NUMBER
1	10	-184.50	TOP	26.88	1
1	10	870.04	BOTTOM	26.88	1
4	9	-204.09	TOP	24.19	2
4	9	1048.58	BOTTOM	24.19	2
7	9	-204.09	TOP	24.19	3
7	9	1048.58	BOTTOM	24.19	3
10	10	-184.50	TOP	26.88	4
10	10	870.04	BOTTOM	26.88	4

TRANSVERSE SLAB STRESS MAXIMUM

X	Y	STRESS (PSI)
5	9	-285.40

PRINCIPAL SLAB STRESS MAXIMUM

X	Y	STRESS (PSI)
6	9	-337.99

MIDSPAN STRESSES INDUCED BY PRESTRESSING AND GIRDER DEAD WEIGHT

THESE STRESSES ARE NOT INCLUDED IN THE MAXIMUMS THAT ARE COMPUTED ABOVE AND MUST BE ADDED INTO THE SOLUTION

STRESS AT GIRDER TOP	(PSI)	313.64
STRESS AT GIRDER BOTTOM	(PSI)	-1623.35

```
PROBLEM
   02      ONE STANDARD AASHTO H20 FROM THE DATA BASE

CONTROL DATA

   NUMBER OF PROBLEMS INPUT (LOAD CASES)                          2
   NUMBER OF GIRDER OUTPUTS                                       2
   DATA OUTPUT (1 = DATA ONLY,2 = DATA+PLOT)                      2
   PRINT SLAB49 RESULTS (1 = YES)                                 0

CONSTANTS

   NUMBER OF INCREMENTS ACROSS BRIDGE                            11
   NUMBER OF INCREMENTS ALONG BRIDGE                             20
   INCREMENT LENGTH ACROSS BRIDGE             (FT)           2.4444
   INCREMENT LENGTH ALONG BRIDGE              (FT)           2.6875
   SLAB THICKNESS                             (IN)           6.3400
   NUMBER OF GIRDERS IN BRIDGE                                    4
   GIRDER TYPE                                         PRES
   GIRDER NAME                                         TYPE C
   GIRDER - TOTAL DEPTH                       (FT)           2.8333
   GIRDER - CENTROID TO BOTTOM                (FT)           1.2500
   GIRDER - MOMENT OF INERTIA               (FT-4)           2.0882
   COMPOSITE SECT - CENTROID TO BOTTOM        (FT)           2.3717
   COMPOSITE SECT - MOMENT OF INERTIA       (FT-4)           7.2834
   MODULUS OF ELASTICITY - CONCRETE          (PSI)        3.000E+06
   MODULUS OF ELASTICITY - STEEL             (PSI)        2.900E+07
```

COMPUTED RESULTANT STRESSES

THESE STRESSES ARE COMPUTED BY METHODS DESCRIBED IN
CHFR REPORT 56 - 25 FOR SLAB49

THESE STRESSES DO NOT INCLUDE THE STRESSES INDUCED BY PRESTRESSING

STRESSES ALONG GIRDER NUMBER = 2

X	Y	DEFLECTION (IN)	TRANSVERSE SLAB STRESS (PSI)	LONGITUDINAL SLAB STRESS (PSI)	GIRDER TOP STRESS (PSI)	GIRDER BOTTOM STRESS (PSI)
4	0	-.000	2.672	1.496	-.003	.015
4	1	-.078	-28.033	-107.011	-42.549	218.602
4	2	-.154	-53.110	-208.652	-83.363	428.295
4	3	-.225	-67.515	-300.503	-122.404	628.874
4	4	-.288	-98.936	-353.757	-138.997	714.126
4	5	-.342	-126.790	-398.955	-152.774	784.905
4	6	-.386	-137.299	-431.624	-165.274	849.127
4	7	-.421	-110.714	-447.846	-179.755	923.525
4	8	-.444	-56.051	-444.042	-192.299	987.972
4	9	-.455	-87.801	-449.748	-186.665	959.027
4	10	-.454	-134.940	-447.063	-173.064	889.152
4	11	-.442	-133.821	-420.324	-160.872	826.514
4	12	-.421	-120.595	-384.188	-147.492	757.769
4	13	-.390	-104.519	-343.408	-132.703	681.789
4	14	-.351	-87.831	-299.105	-116.418	598.120
4	15	-.304	-73.083	-257.151	-100.749	517.617
4	16	-.251	-58.245	-211.672	-83.407	428.521
4	17	-.193	-43.229	-162.446	-64.393	330.832
4	18	-.131	-28.645	-111.989	-44.719	229.750
4	19	-.066	-13.581	-57.505	-23.262	119.511
4	20	-.000	1.265	.708	-.001	.007

STRESSES ALONG GIRDER NUMBER = 3

X	Y	DEFLECTION (IN)	TRANSVERSE SLAB STRESS (PSI)	LONGITUDINAL SLAB STRESS (PSI)	GIRDER TOP STRESS (PSI)	GIRDER BOTTOM STRESS (PSI)
7	0	-.000	2.672	1.496	-.003	.015
7	1	-.078	-28.033	-107.011	-42.549	218.602
7	2	-.154	-53.110	-208.652	-83.363	428.295
7	3	-.225	-67.515	-300.503	-122.404	628.874
7	4	-.288	-98.936	-353.757	-138.997	714.126
7	5	-.342	-126.790	-398.955	-152.774	784.905
7	6	-.386	-137.299	-431.624	-165.274	849.127
7	7	-.421	-110.714	-447.846	-179.755	923.525
7	8	-.444	-56.051	-444.042	-192.299	987.972
7	9	-.455	-87.801	-449.748	-186.665	959.027

7	10	-.454	-134.940	-447.063	-173.064	889.152
7	11	-.442	-133.821	-420.324	-160.872	826.514
7	12	-.421	-120.595	-384.188	-147.492	757.769
7	13	-.390	-104.519	-343.408	-132.703	681.789
7	14	-.351	-87.831	-299.105	-116.418	598.120
7	15	-.304	-73.083	-257.151	-100.749	517.617
7	16	-.251	-58.245	-211.672	-83.407	428.521
7	17	-.193	-43.229	-162.446	-64.393	330.832
7	18	-.131	-28.645	-111.989	-44.719	229.750
7	19	-.066	-13.581	-57.505	-23.262	119.511
7	20	-.000	1.265	.708	-.001	.007

MAXIMUM STRESSES COMPUTED

X	Y	MAXIMUM STRESS (PSI)	LOCATION IN GIRDER	DISTANCE ALONG GIRDER (FT)	GIRDER NUMBER
1	10	-154.16	TOP	26.88	1
1	10	726.97	BOTTOM	26.88	1
4	8	-192.30	TOP	21.50	2
4	8	987.97	BOTTOM	21.50	2
7	8	-192.30	TOP	21.50	3
7	8	987.97	BOTTOM	21.50	3
10	10	-154.16	TOP	26.88	4
10	10	726.97	BOTTOM	26.88	4

TRANSVERSE SLAB STRESS MAXIMUM

X	Y	STRESS (PSI)
5	8	-290.48

PRINCIPAL SLAB STRESS MAXIMUM

X	Y	STRESS (PSI)
6	8	-291.04

MIDSPAN STRESSES INDUCED BY PRESTRESSING AND GIRDER DEAD WEIGHT

THESE STRESSES ARE NOT INCLUDED IN THE MAXIMUMS THAT ARE COMPUTED ABOVE AND MUST BE ADDED INTO THE SOLUTION

STRESS AT GIRDER TOP	(PSI)	313.64
STRESS AT GIRDER BOTTOM	(PSI)	-1623.35

```
02 OCT 80 UNIVERSITY OF TEXAS 170/750A UT2D-215

19.24.33 + 20000CM   11.915CP 10405MS     0MT
19.24.33 RENAME OUTPUT SCHDAT
19.24.33 -   3000CM   11.916CP 10441MS     0MT
19.24.33 REWALLX
19.24.33 +  24000CM   11.922CP 10443MS     0MT
19.24.33 SETCORE
19.24.33 SCHR,SCHDAT
19.24.33 +  50000CM   11.927CP 10496MS     0MT
19.24.33 + 100000CM   11.931CP 10496MS     0MT
19.24.34 + 150000CM   11.952CP 10547MS     0MT
19.24.35   FL USED 116300.
19.24.35   LOAD TIME      1.538 TM SEC.
19.24.35 - 104400CM   12.087CP 10880MS     0MT
19.24.36    20 PLOTTER RECORDS
19.24.36   340 SQUARE INCHES PLOT
19.24.36    70 PLOTTER SECONDS
19.24.36    20 PLOTTER RECORDS
19.24.36   340 SQUARE INCHES PLOT
19.24.37    70 PLOTTER SECONDS
19.24.38    20 PLOTTER RECORDS
19.24.38   340 SQUARE INCHES PLOT
19.24.38    70 PLOTTER SECONDS
19.24.38    20 PLOTTER RECORDS
19.24.38   340 SQUARE INCHES PLOT
19.24.38    70 PLOTTER SECONDS
19.24.38   .. STOP
19.24.38 0 104400CM   13.725CP 11728MS     0MT
19.24.38   .. XTIME =    5.334 TM SECS.   FL USED =  104400B
19.24.38 -   3000CM   13.725CP 11736MS     0MT
19.24.38 REWIND PLOT
19.24.38 +  12300CM   13.727CP 11743MS     0MT
19.24.38 COPYZ PLOT PLOTR
19.24.53 -   1100CM   25.739CP 12673MS     0MT
19.24.53 DISPOSE PLOTR/57
19.24.53  PLOTR DISPOSED.
19.24.53 +   3000CM   25.741CP 12889MS     0MT
19.24.53  FILES ENTERED IN SYSTEM QUEUES:
19.24.53   OUTPUT
19.24.53  JOB COMPLETED.

        USER NUMBER CEIZ551 JOB 022
   12917 DISK PRUS AT .004 SEC./PRU =    51.667 SEC.
                                       ---------------
                        I/O TIME =    51.667 SEC.  *    .8 =    41.333 SEC.
          CYBER 170/750 CPU TIME =    25.751 SEC.  *   1.6 =    41.201 SEC.
                                                               --------------
                                                  TM TIME =    82.534 SEC.

          TM CHARGE RATE: $230/HOUR               TM COST = $    5.27
```

APPENDIX L

CAL-COMP PLOTS FROM PRE- AND POST-PROCESSORS

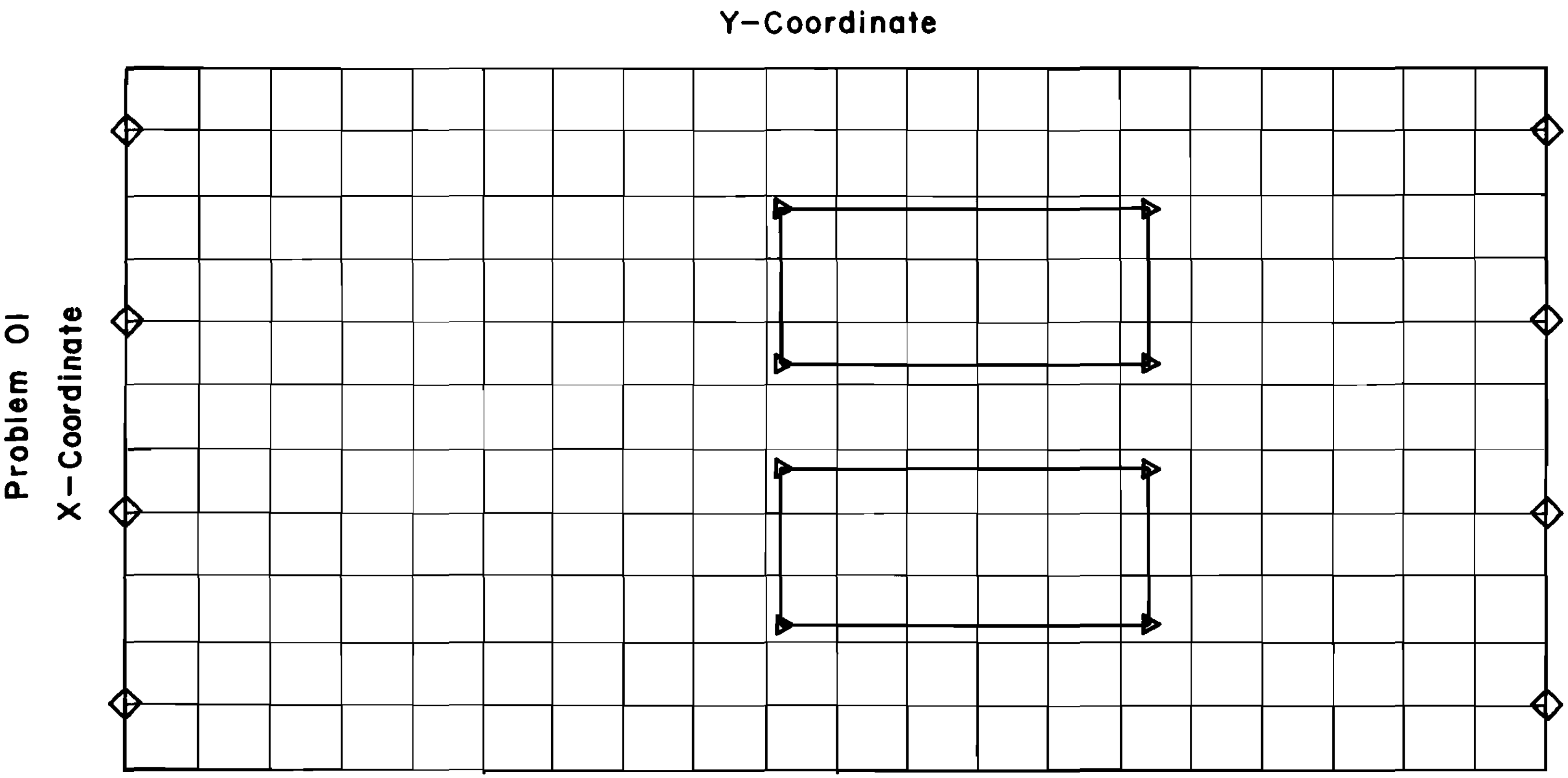


Fig L.1. Truck position for load case 1.

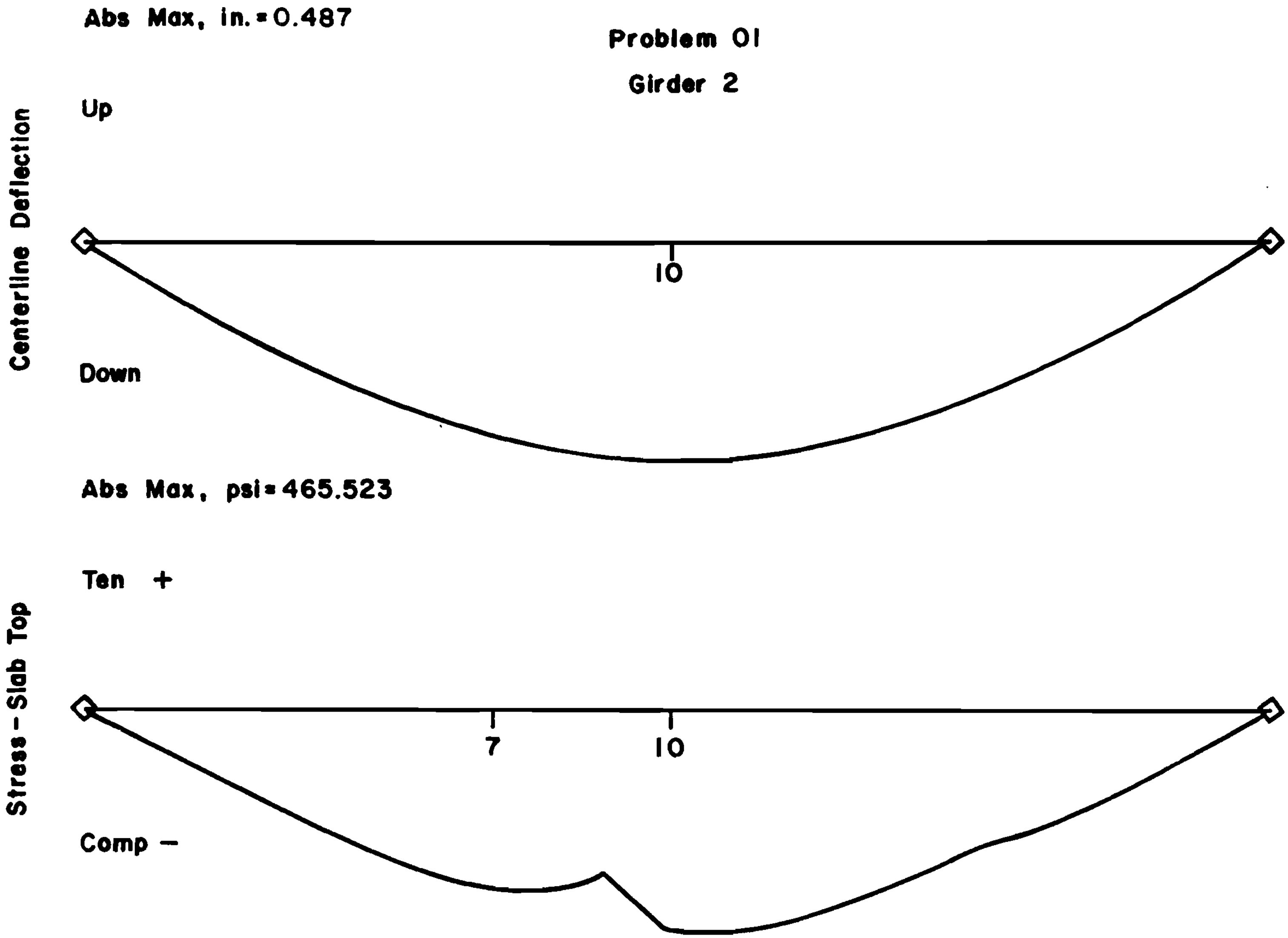


Fig L.2. Deflection and stress (Girder 2) for load case 1.

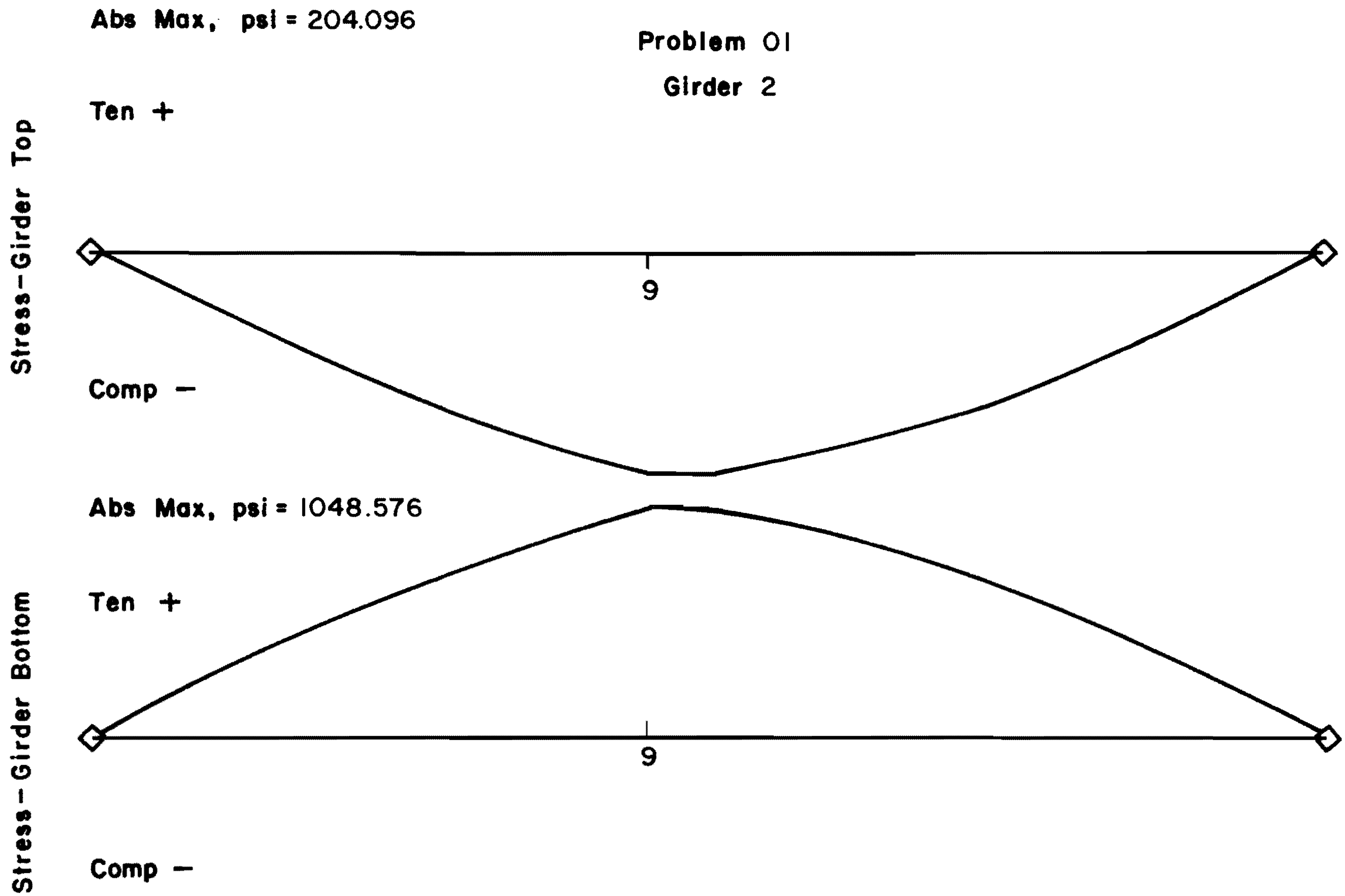


Fig L.3. Stresses for Girder 2 load case 1.

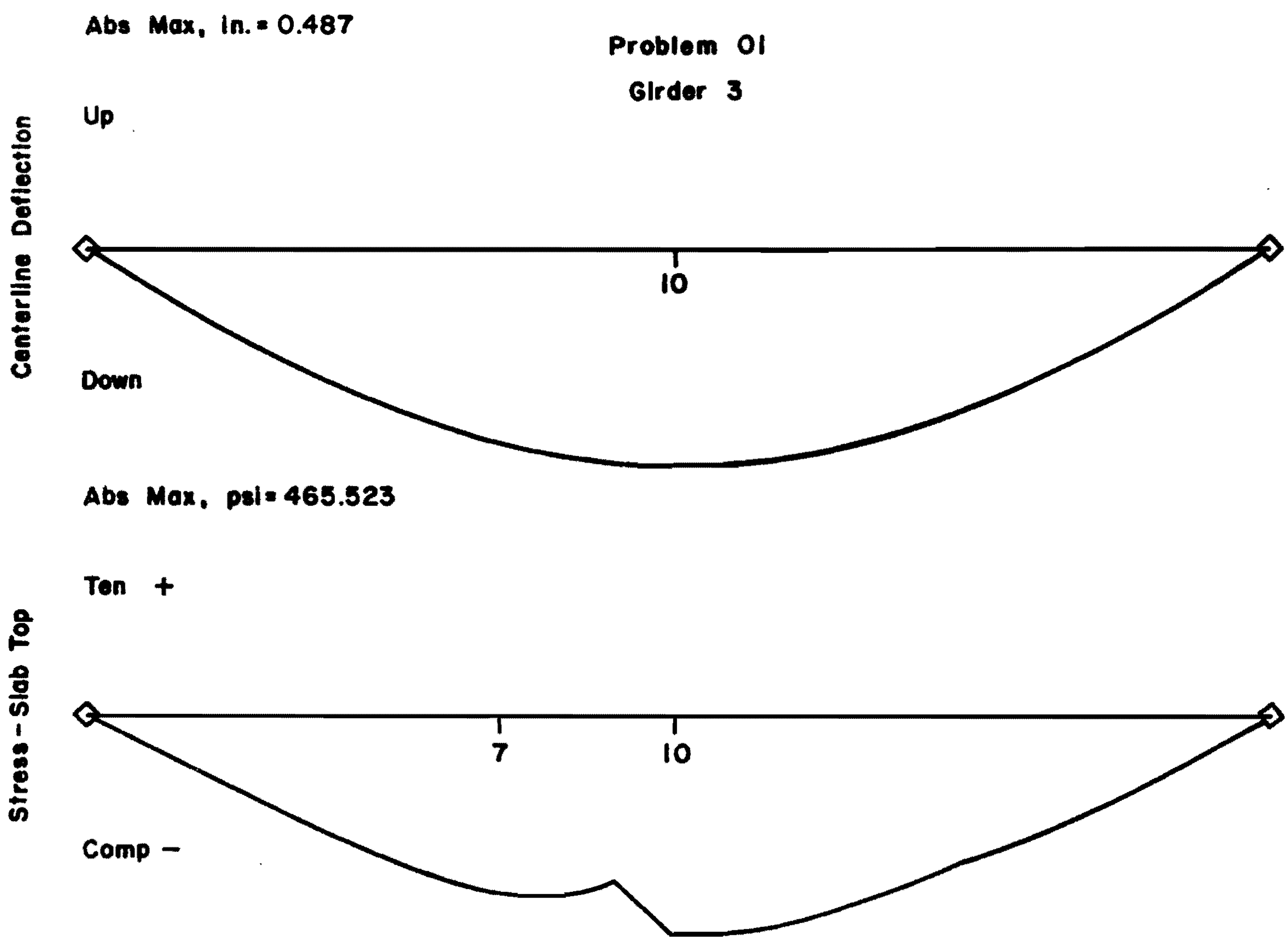


Fig L.4. Deflection and stress (Girder 3) for load case 1.

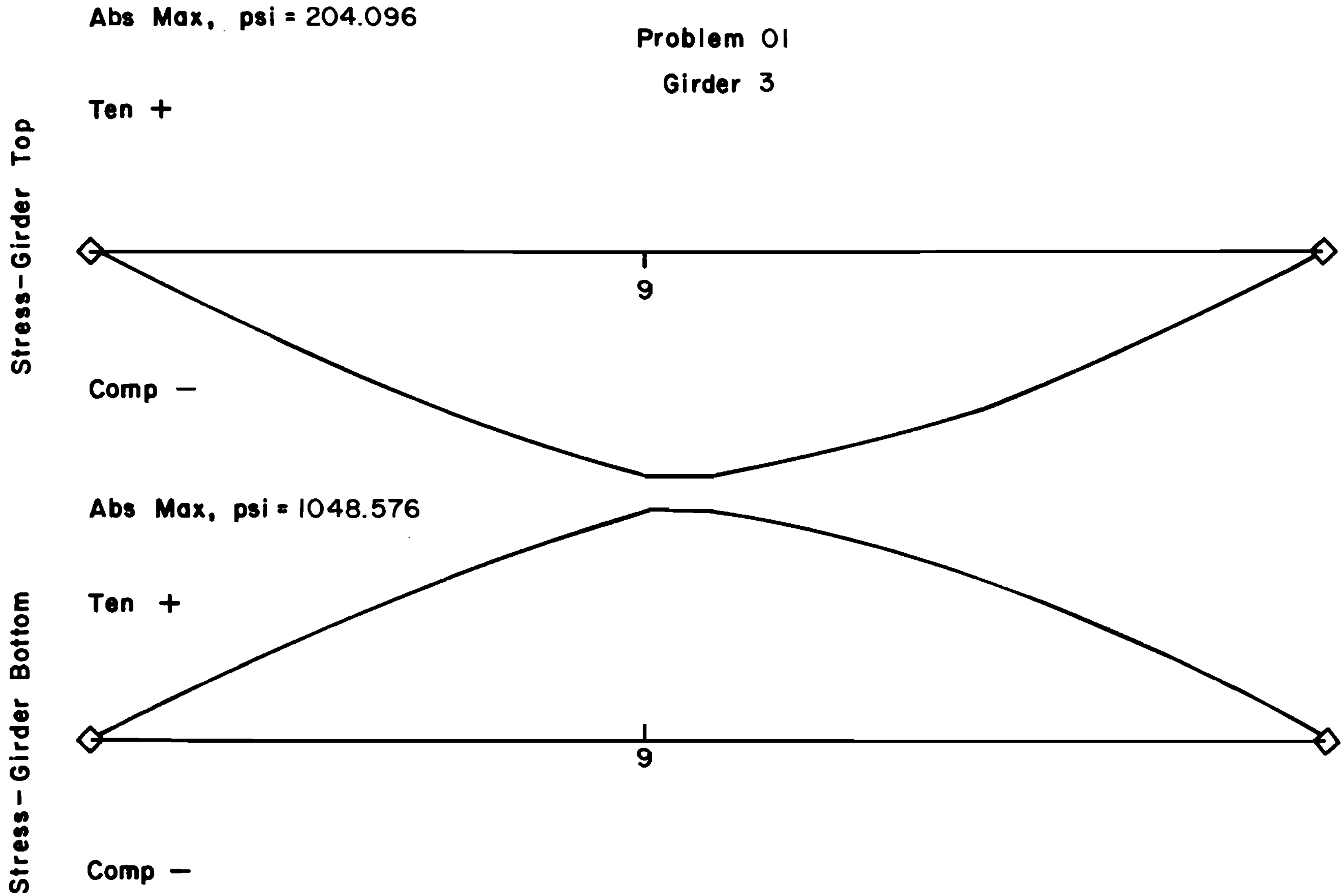


Fig L.5. Stresses for Girder 3 load case 1.

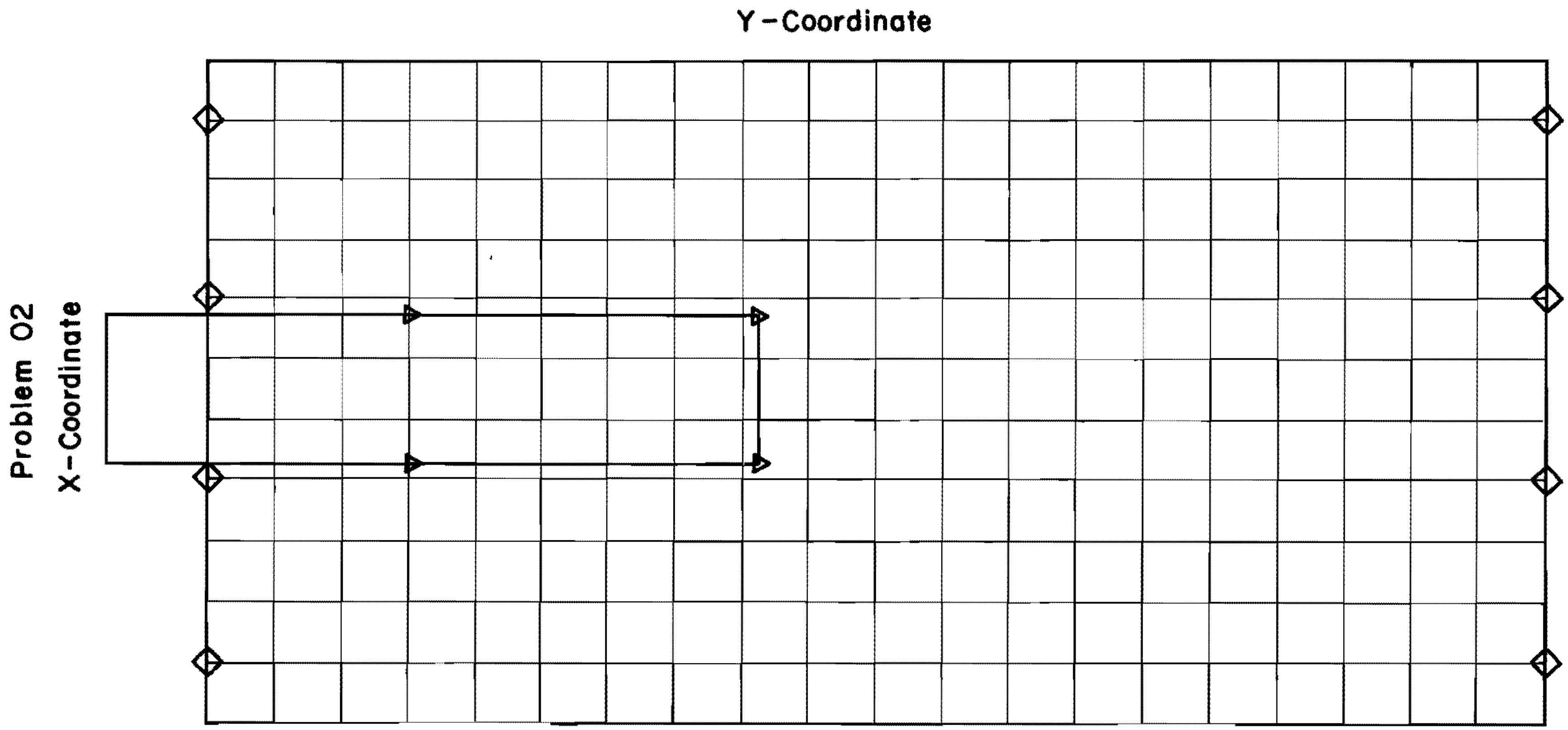


Fig L.6. Truck position for load case 2.

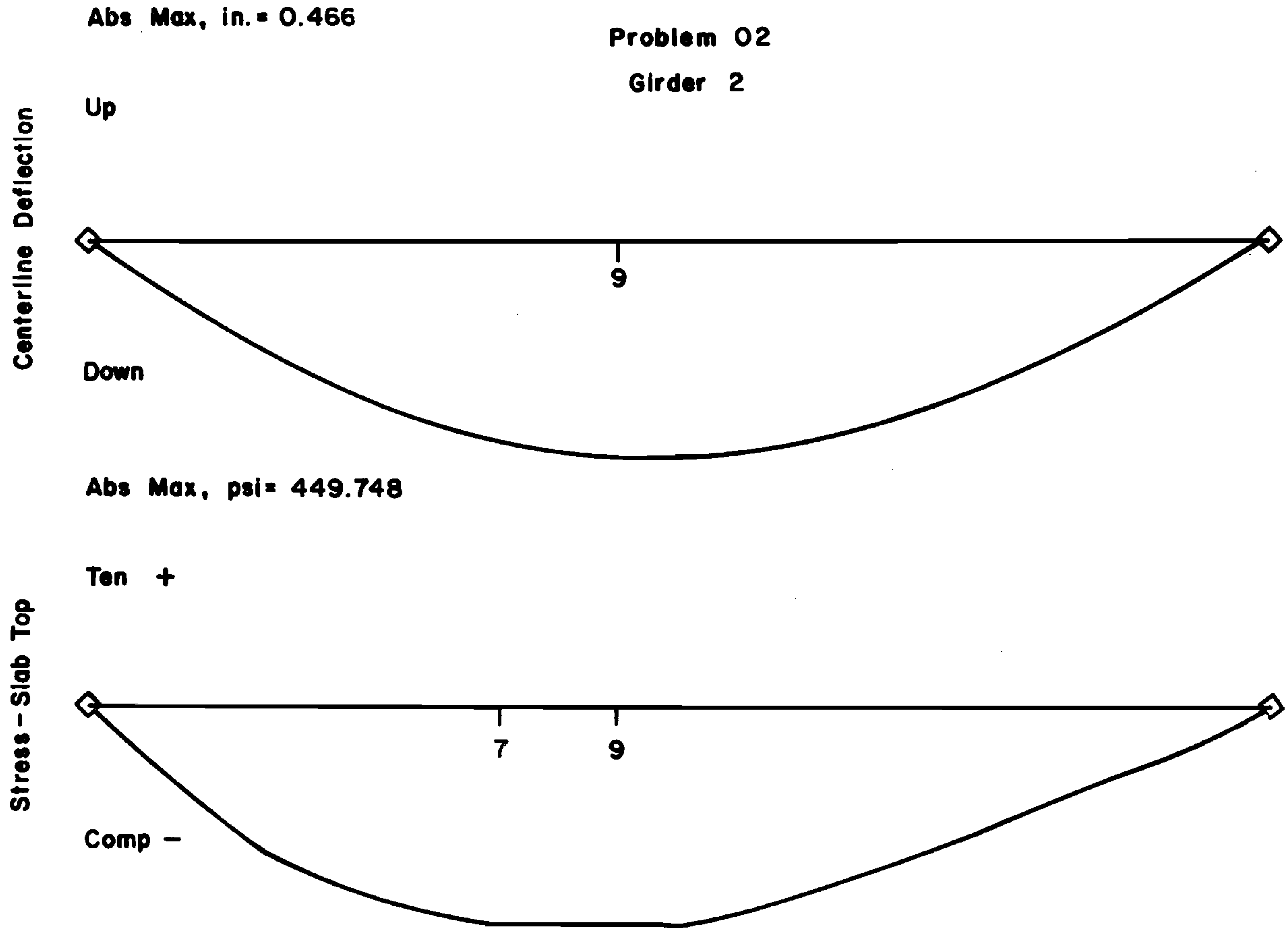


Fig L.7. Deflection and stress (Girder 2) for load case 2.

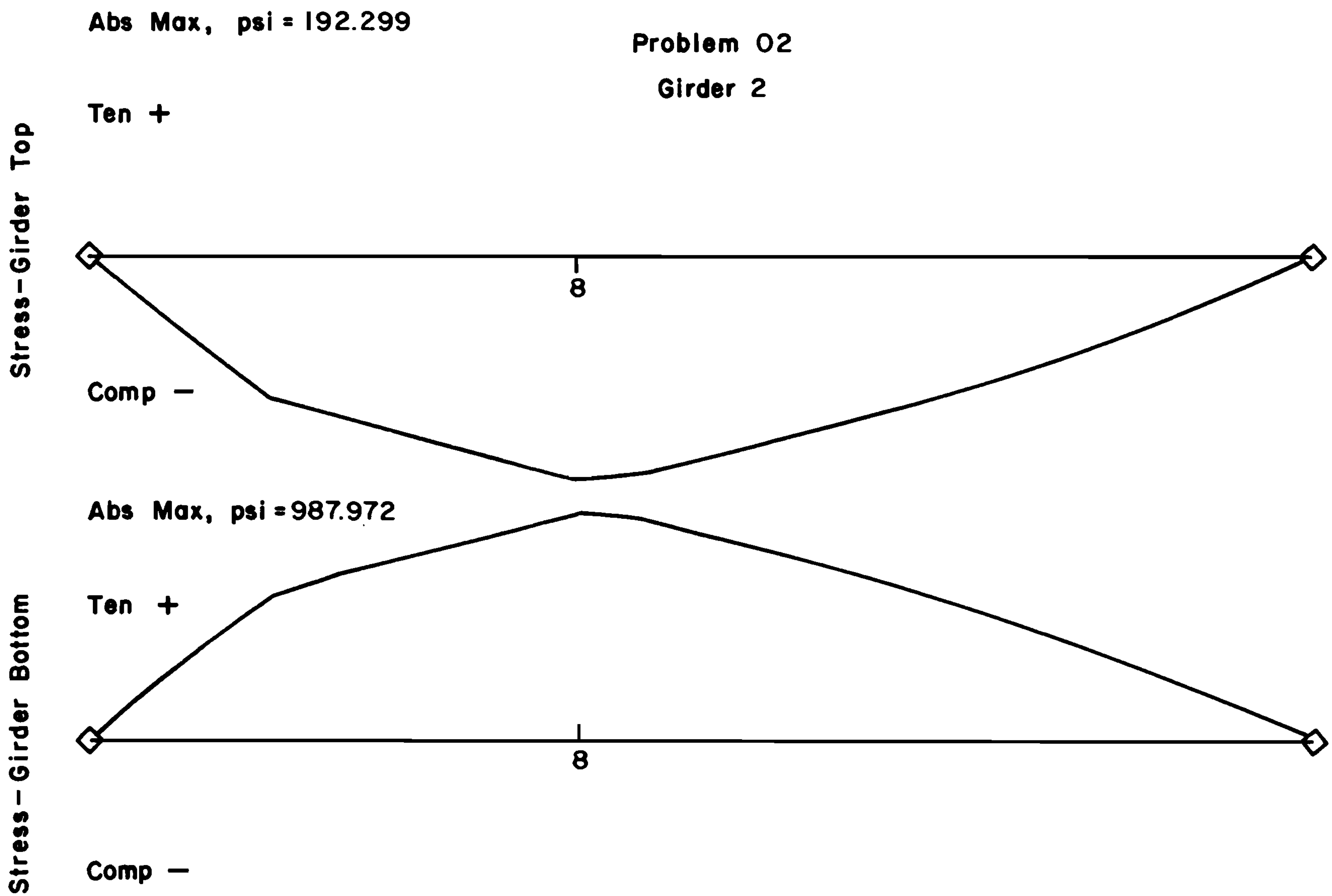


Fig L.8. Stresses for Girder 2 load case 2.

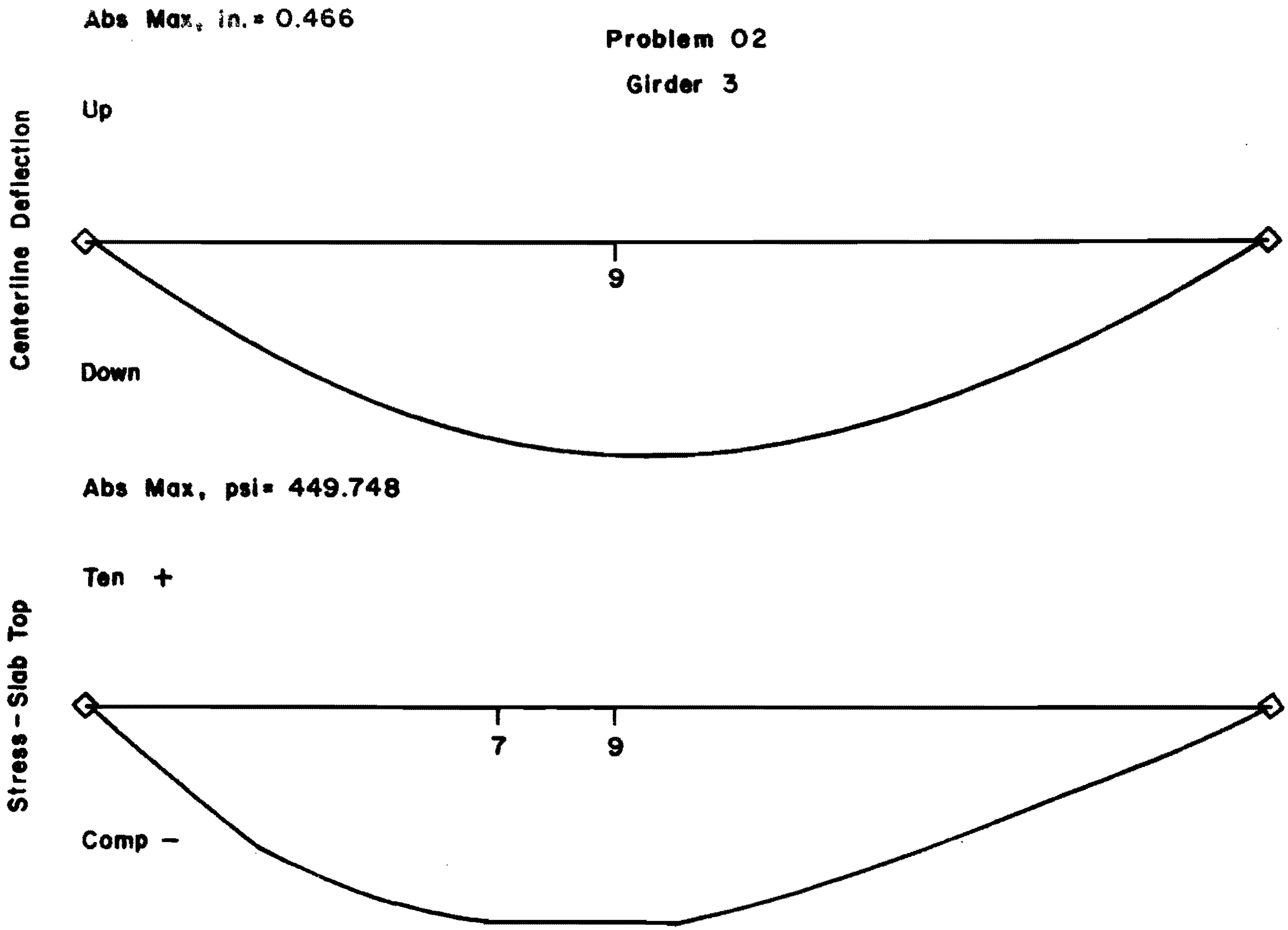


Fig L.9. Deflection and stress (Girder 3) for load case 2.

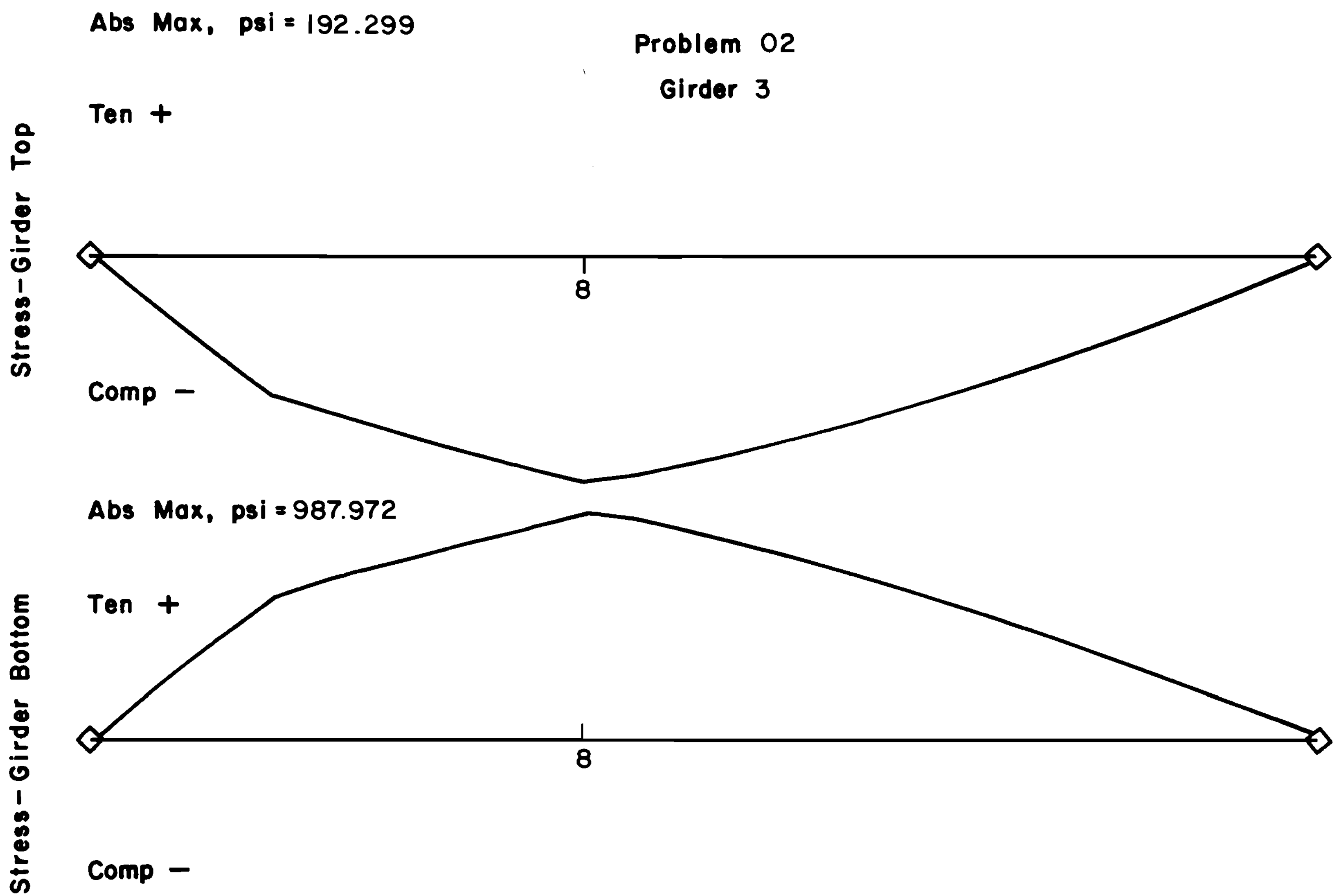


Fig L.10. Stresses for Girder 3 load case 2.