TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AUTOMATED DESIGN OF CONTINUOUS BRIDGES WITH PRECAST PRESTRESSED CONCRETE BEAMS, VOLUME III: SHEAR AND MOMENT ENVELOPE PROGRAM USER'S INSTRUCTIONS		5. Report Date December, 1974 6. Performing Organization Code
7. Author(s) Harry L. Jones, David Harris and Howard L. Furr		8. Performing Organization Report No. Research Report 22-1(F)
9. Performing Organization Name and Address Texas Transportation Institute Texas A&M University College Station, Texas 77840		10. Work Unit No. 11. Contract or Grant No. Study No. 2-5-73-22
12. Sponsoring Agency Name and Address State Department of Highways and Public Transportation 11th and Brazos Austin, Texas 78763		13. Type of Report and Period Covered Final - December, 1972 December, 1974 14. Sponsoring Agency Code

15. Supplementary Notes Research performed in cooperation with the FHWA, DOT.
Study Title: "Automated Design of Prestressed Concrete Beams Made Continuous Live Load"

16. Abstract

A computer program to compute shear and moment envelopes for continuous beams has been written. The program, which was originally written as a subroutine for a continuous beam design program, handles a number of different live and dead loads. Included among these is an axle train (vehicle with up to 15 arbitrarily spaced wheels) which can be used to study the effects of overloaded or unusual vehicle configurations. A standard input form is used and the degree of detail of the output is user specified.

17. Key Words	18. Distribution Statement
Automated Design, Continuous Bridges, Precast Prestressed Concrete Beams, Shear and Moment Envelopes, Computer Program.	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	57	

Form DOT F 1700.7 (8-69)

AUTOMATED DESIGN OF CONTINUOUS BRIDGES WITH PRECAST PRESTRESSED CONCRETE BEAMS
VOLUME III: SHEAR AND MOMENT ENVELOPE PROGRAM USER'S INSTRUCTIONS

by

Harry L. Jones
Assistant Research Engineer

David Harris
Research Assistant

Howard L. Furr
Research Engineer

Research Report Number 22-1(F)

Volume III

Automated Design of Prestressed Concrete Beams
Made Continuous for Live Load

Research Study Number 2-5-73-22

Sponsored by

State Department of Highways
and Public Transportation

in cooperation with

The U. S. Department of Transportation
Federal Highway Administration

Texas Transportation Institute
Texas A&M University
College Station, Texas

December 1974

ABSTRACT

A computer program to compute shear and moment envelopes for continuous beams has been written. The program, which was originally written as a subroutine for a continuous beam design program, handles a number of different live and dead loads. Included among these is an axle train (vehicle with up to 15 arbitrarily spaced wheels) which can be used to study the effects of overloaded or unusual vehicle configurations. A standard input form is used and the degree of detail of the output is user specified.

DISCLAIMER

SUMMARY

This report describes the use of a computer program to compute shear and moment envelopes for continuous beams. The program, which was originally written as a subroutine for a continuous beam design program, handles a number of different live and dead loads. Included among these is an axle train (vehicle with up to 15 arbitrarily spaced wheels) which can be used to study the effects of overloaded or unusual vehicle configurations. A standard input form is used and the degree of detail of the output is user specified.

RECOMMENDATION FOR IMPLEMENTATION

This program should be of use in overload studies. The axle train option in the program allows direct comparison between moments produced by a standard AASHTO truck and those of an overloaded, nonconventional vehicle.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

The design of new bridges or the analysis of existing bridges under overload conditions requires a knowledge of maximum and minimum values of shear force and bending moment. The computer program described below makes these computations for a variety of live load and dead load combinations. It's capabilities are comparable to the analysis segment of "B-30" (1), used by the Bridge Division of THD, with certain additional features.

Several levels of output are available, ranging from a listing of extreme values of shear and moment to a display of ordinates of the influence lines for shear and moment. One option presents a summary of extreme values of shear and moment produced by each type of live load specified. This permits a comparison of the moments produced by a standard design vehicle (e.g. an HS-20 truck) and an overloaded vehicle.

This computer program was developed as a subroutine for use in the automated design of prestressed concrete beams made continuous for live load. Reference (2) contains a development of the methodology and calculation procedures used and flow charts of program logic.

PROGRAM CAPABILITIES AND LIMITATIONS

For the array sizes currently dimensioned in the program, continuous bridge beams from two to ten spans (inclusive) can be analyzed. The program assumes that the section properties of the beam are identical for each span. The total length of the bridge (sum of the span lengths) is limited to 999 ft. The program treats any or all of the types of loads described below.

Dead Load

The uniformly distributed loads resulting from the weight of the beam and deck slab are carried by the beam acting as a continuous beam or a series of simply supported spans, according to the user's specification. The former case would arise with a continuous steel girder, while the latter could occur when simple beams are made continuous for live load by placement of the deck slab.

Live Loads

Standard AASHTO Loadings - Program options include H-15, H-20, HS-15 HS-20 and the accompanying lane loadings. The truck and lane loadings are automatically scaled by the appropriate lateral distribution factor and impact factor (see PROGRAM INPUT section).

Axle Train - The program will treat a moving string of up to 15 axles, of arbitrary weight and spacing. The length of axle string must not exceed 199 ft. The lateral distribution factor is applied to axle loads. The impact factor is also applied unless stipulated otherwise on input.

Uniform Load - A uniformly distributed load may also be input. The program assumes this load acts on all spans, and is carried by the continuous beam.

Concentrated Stationary Loads

Concentrated forces may be applied at any point in any span. They may be carried by the continous beam or by each span acting as a simple beam, depending on the user's specification. There is no limit to the number of concentrated forces which may be applied when each span acts as a simply supported beam. When concentrated forces are applied to the continuous beam, the total number of applied forces is limited to 100.

PROGRAM INPUT

A standard input form, shown in Figure 1, is available for data preparation. The cards to be input are described below.

Title Cards

The first three input cards are the title cards shown at the top of the form in Figure 1. The information preprinted on the form in various columns need not be punched on the title cards - it will be printed automatically. The information on these cards is optional, but three cards must initiate the data deck.

Beam and Load Card

The fourth card contains information on the beam and loads it is to carry. Columns 5 through 9 allow for symbols to identify the beam and may be left blank. Beam weight in kips per foot is entered in columns 13 - 17 and lateral spacing of beams in feet in columns 21 - 25. The distribution factor applied to standard AASHTO loadings is entered in columns 29 - 31. This factor is applied to axle loads and if left blank, is assumed to be S/11 where S is the lateral beam spacing. This factor is applied to AASHTO truck and lane loads, as well as axle train loads. The slab thickness is entered in columns 34 - 38. AASHTO loadings (H-15, H-20, HS-15, HS-20) are entered in columns 41 - 42 and 44 - 45, and uniform load applied to the continuous beam is entered in columns 49 - 53. If an axle train loading is used, enter Y in column 57. Otherwise, enter N or leave blank. If the beam is continuous for its own weight and the slab weight (e.g.,

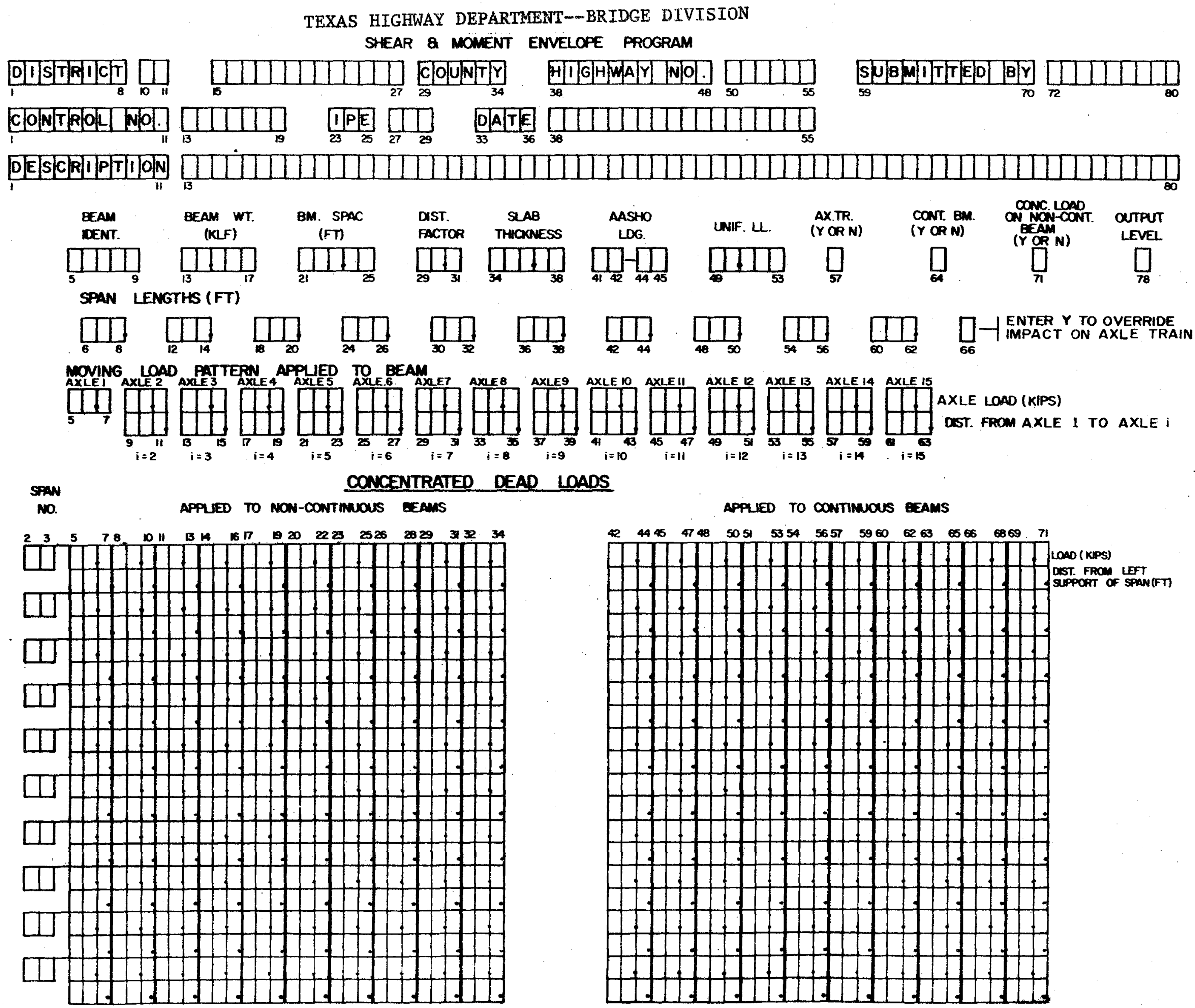

Figure 1. Input Form

diaphrams with prestressed concrete beams made continuous for live load), enter Y in column 71. The level of output (0, 1 or 2) is entered in column 78.

Span Card

The length of each span, rounded to the nearest foot, are entered where shown. If the user wishes to delete the application of an impact factor to axle train loads, he must enter Y in column 66.

Axle Train Cards (Omit if column 57 of Beam and Load card contains N or blank)

These cards describe the moving axle load string. The axle weights are entered in the appropriate columns of card 6. Card 7 contains the axle spacings. These spacings are measured in feet (round to nearest foot, if necessary), from axle number 1.

Concentrated Dead Load Cards (Omit if no concentrated loads applied)

Information concerning concentrated, non-moving dead loads are input by span number. Two cards are required for each span with concentrated loads. If the concentrated loads are applied to the continous beam, this information is entered in columns 42 - 71. If they are applied with each span simply supported, the information is entered in columns 5 -34.

The first two columns of the first card contain the span number (right justified). The remaining columns indicate the magnitude of each load in the span.

The second card indicates the position of each load in the span, measured from the left support of that span.

If more than ten concentrated forces are applied to any one span, they can be accommodated with additional card pairs, using the same span number. The total number of concentrated forces applied to the continuous beam must not exceed 100.

Multiple Problem Runs

More than one problem may be processed in a single program run. Title cards are entered with the first problem data set only. Subsequent problem data cards are loaded behind the first, but without title cards.

PROGRAM OUTPUT

The amount of output information depends upon the output level (Beam and Load card, column 78) specified. All output levels list the maximum and minimum live load moments and shears and an indication of which liveload type produced them at tenth points of each span. In the output, the tenth points are denoted as I.0, I.1,...I.10, where I is the span number. Point I.0 is a point just to the right of the left support for span I. Point I.10 is a point just to the left of the right support for span I.

The output items for each output level are:

Output Level = 0 or blank

A one page summary, listing for each design point the maximum and minimum values of live load moment and shear, total dead load plus concentrated force (if present) moments and shears, and total live load plus dead load moments and shears. For live loads, a symbol indicating which type of loading ("T" for H or HS truck, "L" for lane, "A" for axle train, "U" for uniform load) produces the maximum or minimum value is displayed. A sample of level 0 output is shown in Figure 2.

Output Level = 1

Output Level 1 produces a breakdown at each design point of the maximum and minimum moments and shears produced by each type of live loading. The positions of the wheels for an H or HS truck and the axle train which produce each extreme value of moment and shear are listed. Level 1 output is seen in Figure 3.

Output Level = 2

Output Level 2 is the same as Level 1, with the inclusion of ordinates of the influence lines for shear and moment at each design point as shown in Figure 4.

```
DISTRICT 14            TRAVIS COUNTY   HIGHWAY NO.  IH 35   SUBMITTED BY HLJ
CONTROL NO. 214-876   IPE 694   DATE DEC. 1974
DESCRIPTIONSAMPLE PROBLEM


                 PROPERTIES OF BEAM DESIGNATED  THD-1


EACH SPAN OF BEAM ACTS AS SIMPLE BEAM IN CARRYING SLAB AND BEAM WEIGHTS


BEAM WEIGHT = 0.514 (KIPS PER FOOT)
BEAM SPACING =  5.25 (FT)
SLAB THICKNESS = 7.25 (IN)


SPAN LENGTHS (FT)
L( 1)=    40.
L( 2)=    40.


                 LOADING CONDITIONS
HS 20
THE AASHO DIST FACTOR = 1.0
```

Figure 2. Sample of Level 0 Output

```
                                        SUMMARY OF CALCULATIONS

    *        *LIVE LOAD EFFECTS (WITH IMPACT)          ** DEAD LOAD EFFECTS     **   TOTAL    EFFECTS             **

DESIGN* MAXIMUM    * MINIMUM    * MAXIMUM    * MINIMUM    **               *              ** MAX(+)     * MAX(-)     **              **
POINT * MOMENT     *  MOMENT    *  SHEAR     *  SHEAR     ** MOMENT        *    SHEAR     ** MOMENT     * MOMENT     **    SHEAR     **
  .   * (KIP-FT)   *  (KIP-FT)  *   (KIPS)   *  (KIPS)    ** (KIP-FT)      *    (KIPS)    ** (KIP-FT)   * (KIP-FT)   **    (KIPS)    **


 1.   0* 0.000E-01T* 0.000E-01T* 3.228E 01T*-3.358E 00T** 0.000E-01 *-1.980E 01 ** 0.000E-01 * 0.000E-01 ** 2.315E 01 **
 1.   1* 1.086E 02T*-1.343E 01T* 2.715E 01T*-3.735E 00L** 7.126E 01 *-1.584E 01 ** 1.799E 02 * 5.783E 01 ** 1.957E 01 **
 1.   2* 1.780E 02T*-2.687E 01T* 2.225E 01T*-5.823E 00L** 1.267E 02 *-1.188E 01 ** 3.047E 02 * 9.982E 01 ** 1.770E 01 **
 1.   3* 2.115E 02T*-4.030E 01T* 1.763E 01T*-8.248E 00T** 1.663E 02 *-7.918E 00 ** 3.778E 02 * 1.260E 02 ** 1.617E 01 **
 1.   4* 2.222E 02T*-5.373E 01T* 1.337E 01T*-1.125E 01T** 1.900E 02 *-3.959E 00 ** 4.123E 02 * 1.363E 02 ** 1.520E 01 **
 1.   5* 2.154E 02T*-6.717E 01T* 9.516E 00T*-1.549E 01T** 1.980E 02 * 0.000E-01 ** 4.133E 02 * 1.308E 02 ** 1.549E 01 **
 1.   6* 1.996E 02T*-8.060E 01T* 6.169E 00T*-1.995E 01T** 1.900E 02 * 3.959E 00 ** 3.896E 02 * 1.094E 02 ** 1.599E 01 **
 1.   7* 1.579E 02T*-9.403E 01T* 4.184E 00T*-2.426E 01T** 1.663E 02 * 7.918E 00 ** 3.241E 02 * 7.225E 01 ** 1.634E 01 **
 1.   8* 9.584E 01T*-1.075E 02T* 2.255E 00L*-2.865E 01T** 1.267E 02 * 1.188E 01 ** 2.225E 02 * 1.922E 01 ** 1.677E 01 **
 1.   9* 2.694E 01L*-1.209E 02T* 9.676E-01L*-3.277E 01T** 7.126E 01 * 1.584E 01 ** 9.820E 01 *-4.964E 01 ** 1.694E 01 **
 1.  10* 0.000E-01L*-1.652E 02L* 0.000E-01L*-3.645E 01T** 0.000E-01 * 1.980E 01 ** 0.000E-01 *-1.652E 02 ** 1.980E 01 **

                                 BRIDGE IS SYMMETRICAL ABOUT THIS POINT

CORE USAGE        OBJECT CODE=    66176 BYTES,ARRAY AREA=    59820 BYTES,TOTAL AREA AVAILABLE=  131168  BYTES

DIAGNOSTICS          NUMBER OF ERRORS=         0, NUMBER OF WARNINGS=         0, NUMBER OF EXTENSIONS=        0

  TAMU/WATFIV - VER 1 LEV 3 JANUARY 1972      DATE=  75/042
```

Figure 2. Continued

```
**************************************************************DESIGN POINT   1.  6****************************************************************
*    *        * MAGNITUDE  *                    ******************************************************************************************
*    *        * (KIP-FEET) *                    *WHEEL POSTION (DISTANCE FROM LEFT END OF BRIDGE)*******************
LOAD* FORCE *      OR      ******************************************************************************************************
TYPE* TYPE  *    (KIPS)    *
**************************************************************************************************************************************
                           *WHL 1*WHL 2*WHL 3*
 HS *MAX.MOM* 1.99555E 02*   38*   24*   10*
 HS *MIN.MOM*-8.06011E 01*   78*   64*   50*
 HS *MAX.SHR* 6.16903E 00*   38*   24*    0*
 HS *MIN.SHR*-1.99457E 01*   -4*   10*   24*
LANE*MAX.MOM* 1.33959E 02*
LANE*MIN.MOM*-4.95569E 01*
LANE*MAX.SHR* 5.76974E 00*
LANE*MIN.SHR*-1.56622E 01*
 DL * MOMENT* 1.90038E 02*
 DL * SHEAR * 3.95914E 00*


**************************************************************DESIGN POINT   1.  7****************************************************************
*    *        * MAGNITUDE  *                    ******************************************************************************************
*    *        * (KIP-FEET) *                    *WHEEL POSTION (DISTANCE FROM LEFT END OF BRIDGE)*******************
LOAD* FORCE *      OR      ******************************************************************************************************
TYPE* TYPE  *    (KIPS)    *
**************************************************************************************************************************************
                           *WHL 1*WHL 2*WHL 3*
 HS *MAX.MOM* 1.57867E 02*    0*   14*   28*
 HS *MIN.MOM*-9.40346E 01*   78*   64*   50*
 HS *MAX.SHR* 4.18433E 00*  -14*    0*   28*
 HS *MIN.SHR*-2.42586E 01*   42*   28*   14*
LANE*MAX.MOM* 1.04888E 02*
LANE*MIN.MOM*-5.78164E 01*
LANE*MAX.SHR* 3.85759E 00*
LANE*MIN.SHR*-1.83470E 01*
 DL * MOMENT* 1.66283E 02*
 DL * SHEAR * 7.91826E 00*
```

Figure 3. Sample of Level 1 Output

```
*******************************************************DESIGN POINT  1. 6***************************************************
*   *       * MAGNITUDE  *            *************************************************************************************
*   *       * (KIP-FEET) *            *WHEEL POSTION (DISTANCE FROM LEFT END OF BRIDGE)******************
LOAD* FORCE *     OR     **************************************************************************************************
TYPE* TYPE  *   (KIPS)   *
***************************************************************************************************************************
                         *WHL 1*WHL 2*WHL 3*
 HS *MAX.MOM* 1.99555E 02*   38*   24*   10*
 HS *MIN.MOM*-8.06011E 01*   78*   64*   50*
 HS *MAX.SHR* 6.16903E 00*   38*   24*    0*
 HS *MIN.SHR*-1.99457E 01*   -4*   10*   24*
LANE*MAX.MOM* 1.33959E 02*
LANE*MIN.MOM*-4.95569E 01*
LANE*MAX.SHR* 5.76974E 00*
LANE*MIN.SHR*-1.56622E 01*
 DL * MOMENT* 1.90038E 02*
 DL * SHEAR * 3.95914E 00*
```

INFLUENCE LINE VALUES

DESIGN POINT 1. 6

DISTANCE FROM LEFT END OF BRIDGE (FT)	SHEAR (KIPS)	MOMENT (KIP-FT)
0(SUPPORT)	0.0000000E-01	0.0000000E-01
1	-3.1242370E-02	2.5018310E-01
2	-6.2464230E-02	5.0085440E-01
3	-9.3640680E-02	7.5260920E-01
4	-1.2474670E-01	1.0060720E 00
5	-1.5575770E-01	1.2618100E 00
6	-1.8665280E-01	1.5203240E 00
7	-2.1740740E-01	1.7822110E 00
8	-2.4799640E-01	2.0480800E 00
9	-2.7839940E-01	2.3184050E 00
10	-3.0859150E-01	2.5937950E 00
11	-3.3854770E-01	2.8748530E 00
12	-3.6824760E-01	3.1620560E 00
13	-3.9766620E-01	3.4560100E 00
14	-4.2677860E-01	3.7573110E 00
15	-4.5556440E-01	4.0664520E 00
16	-4.8399870E-01	4.3840300E 00
17	-5.1205640E-01	4.7106440E 00
18	-5.3971720E-01	5.0467850E 00
19	-5.6695620E-01	5.3930500E 00
20	-5.9374830E-01	5.7500400E 00
21	-6.2007310E-01	6.1182430E 00
22	-6.4590580E-01	6.4982590E 00
23	-6.7122140E-01	6.8906850E 00
24	3.0400360E-01	7.2960140E 00
25	2.7978510E-01	6.7148430E 00
26	2.5615710E-01	6.1477690E 00

Figure 4. Sample of Level 2 Output

REFERENCES

1. Sikes, G. H., "The Analysis of Continuous Beams for Highway Bridges," State Highway Department of Georgie, Atlanta, Georgia, July 1967.

2. Jones, H. L., Ingram, L. L., Furr, H. L. and Harris, D. W., "Automated Design of Continuous Bridges With Precast Prestressed Concrete Beams - Volume I: Design Considerations," Research Report 22-1(F), Vol. I, Texas Transportation Institute, Texas A&M University, College Station, Texas, October 1974.

3. Jones, H.L., Harris D.W. and Furr, H.L., "Automated Design of Continuous Bridges with Precast Prestressed Concrete Beams-Volume II: Program Documentation", Research Report 22-1(F), Vol. II, Texas Transportation Institute, Texas A & M University, College Station, Texas.

APPENDIX

Program Listing

```
      INTEGER POSCMP,YES
      REAL*4  INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMOM,MINSHR,L
     1,LLVIN
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,10),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCCMP(100)
      COMMON/BLK2/LMMIN(15),LMMAX(15),LVMAX(15),NWHL(14),NODDSN(11,10),P
     1OSCMP(100),LODKOD(7),LEXTRM(30),LEXTRV(30),LVMIN(15),LODCON(11,10,
     24)
      COMMON/BLK3/MINMOM,MAXMOM,MAXSHR,MINSHR,SCLLNE,SCLCOM,SCLHHS,SCLCO
     1V,UNIFLL,BMWT,S,THK
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCOUNT
      COMMON/BLK5/ITITLE(80),JTITLE(80),KTITLE(80),JBMDSG(2),KAASHO(2),K
     1AXTR,KOUTPT,KCONT,KODE,KOVRD,FACTOR
      DATA YES/4HY   /
      KKT=1
    2 CALL INPUT(KKT)
      IF(KOUTPT.EQ.0) CALL OUTKO(0)
C
C     CHECK SYMMETRY, KODE=0 SYMMETRICAL, KODE=1 UNSYMMETRICAL
      J1=N/2
      KODE=0
      DO 12 J2=1,J1
      IF(KODE.EQ.1) GO TO 12
      S1=L(J2)-L(N+1-J2)
      IF(ABS(S1).GE..1.E-02) GO TO 14
      IF(KODE.EQ.0) KODE=0
      GO TO 12
   14 KODE=1
 12   CONTINUE
C
C      TEST FOR ODD OR EVEN NUMBER OF SPANS  IEVEN=1  EVEN  IEVEN=0 ODD
      NT=N/2
      DO 10 I=1,NT
      TEST=N-(I*2)
      IF(TEST.EQ.0.)IEVEN=1
      IF(TEST.EQ.1.) IEVEN=0
   10 CONTINUE
C
C     BEGIN  COMPUTATIONS OF EFFECTS AT EACH DESIGN POINT
C
 52   CALL REACTN
      DO 1000 JSPAN=1,N
      DO 1000 JPNT=1,11
      CALL INFLNE
C
C      FIND EXTREME VALUES OF INFLM(I)
C
```

```
   53 NDISC=NODDSN(JPNT,JSPAN)
      NEXTRM=0
      NEXTRV=0
      JSTOP=199+NPNTS
      IF(JSPAN.EQ.1.AND.JPNT.EQ.1) GO TO 61
      IF(JSPAN.EQ.N.AND.JPNT.EQ.11) GO TO 61
      DO 60 J1=1,N
      JSTRT=NODDSN(1,J1)
      JSTOP=NODDSN(11,J1)
      ZMIN=0
      ZMAX=0
      DO 56 J2=JSTRT,JSTOP
      IF(INFLM(J2).GE.ZMIN) GO TO 54
      ZMIN=INFLM(J2)
      JMIN=J2
   54 IF(INFLM(J2).LE.ZMAX) GO TO 56
      ZMAX=INFLM(J2)
      JMAX=J2
   56 CONTINUE
      IF(ZMAX.LE.1.E-05) GO TO 58
      NEXTRM=NEXTRM+1
      LEXTRM(NEXTRM)=JMAX
   58 IF(ABS(ZMIN).LE.1.E-05) GO TO 60
      NEXTRM=NEXTRM+1
      LEXTRM(NEXTRM)=JMIN
   60 CONTINUE
   61 IF(NEXTRM.EQ.0) LEXTRM(1)=200
      IF(NEXTRM.EQ.0) NEXTRM=1
C
C        FIND EXTREME VALUES OF INFLV(I)
C
      NEXTRV=0
      DO 45 J1=1,N
      JSTRT=NODDSN(1,J1)
      JSTOP=NODDSN(11,J1)
      ZVIN=0
      ZVAX=0
      DO 65 J2=JSTRT,JSTOP
      IF(J2.NE.NDISC) GO TO 63
      IF(INFLV(J2)-1.GE.ZVIN) GO TO 63
      ZVIN=INFLV(J2)-1
      JVIN=J2
   63 IF(INFLV(J2).GE.ZVIN) GO TO 64
      ZVIN=INFLV(J2)
      JVIN=J2
   64 IF(INFLV(J2).LE.ZVAX) GO TO 65
      ZVAX=INFLV(J2)
      JVAX=J2
```

```
   65 CONTINUE
      IF(ZVAX.LE.1.E-05) GO TO 43
      NEXTRV=NEXTRV+1
      LEXTRV(NEXTRV)=JVAX
   43 IF(ABS(ZVIN).LE.1.E-05) GO TO 45
      NEXTRV=NEXTRV+1
      LEXTRV(NEXTRV)=JVIN
   45 CONTINUE
      IF(NEXTRM.EQ.1) GO TO 47
      DO 44 J1=2,NEXTRM
      IF(LEXTRM(J1).GT.LEXTRM(J1-1)) GO TO 44
      JT=LEXTRM(J1-1)
      LEXTRM(J1-1)=LEXTRM(J1)
      LEXTRM(J1)=JT
   44 CONTINUE
   47 IF(NEXTRV.EQ.1) GO TO 51
      DO 48 J1=2,NEXTRV
      IF(LEXTRV(J1).GT.LEXTRV(J1-1)) GO TO 48
      JT=LEXTRV(J1-1)
      LEXTRV(J1-1)=LEXTRV(J1)
      LEXTRV(J1)=JT
   48 CONTINUE
      JSTOP=NEXTRV
      DO 50 J1=2,JSTOP
      IF(LEXTRV(J1).NE.LEXTRV(J1-1)) GO TO 50
      DO 49 J2=J1,NEXTRV
   49 LEXTRV(J2-1)=LEXTRV(J2)
      NEXTRV=NEXTRV-1
      GO TO 51
   50 CONTINUE
   51 CONTINUE
   67 CALL OUTIL(0)
   69 LLMIN(JPNT,JSPAN)=10000.
      LLMAX(JPNT,JSPAN)=-10000.
      LLVAX(JPNT,JSPAN)=-10000.
      LLVIN(JPNT,JSPAN)=10000.
      NODSTP=199+NPNTS
      IF(LODKOD(1).EQ.0.AND.LODKOD(2).EQ.0) GO TO 126
      IF(LODKOD(1).NE.1) GO TO 68
      JTRIG=-1
      GO TO 70
 68   JTRIG=1
C***********************************************************************
C********H OR HS TRUCK LOADING******************************************
C***********************************************************************
C
C      H OR HS TRUCK LEFT TO RIGHT -- MOMENTS
C
```

```
  70  MAXSHR=-10000.
      MAXMOM=-10000.
      MINMOM=10000.
      MINSHR=10000.
      IF(JTRIG.EQ.-1) NMOVES=0
      IF(JTRIG.EQ.1) NMOVES=30
      LEXTRM(NEXTRM+1)=LEXTRM(NEXTRM)+NMOVES
      LEXTRV(NEXTRV+1)=LEXTRV(NEXTRV)+NMOVES
      DO 84 J1=1,NEXTRM
      JSTRT=LEXTRM(J1)
      JSTOP=JSTRT+NMOVES
      IF(JSTOP.GT.LEXTRM(J1+1)) JSTOP=LEXTRM(J1+1)
      DO 82 J2=JSTRT,JSTOP,2
      K1=J2+14
      ZMLR12=.25*INFLM(K1)+INFLM(J2)
      K3=J2-14
      IF(JTRIG) 80,80,72
   72 KSTRT=J2-30
      KSTOP=J2-14
      IF(KSTOP.LE.200) GO TO 80
      IF(KSTRT.GT.200) GO TO 78
      DO 74 J3=2,30,2
      J4=KSTOP-J3
      IF(J4.GT.200) GO TO 74
      GO TO 76
   74 CONTINUE
   76 KSTRT=J4
   78 CALL SORTIL(INFLM,ZMAX,JMAX,ZMIN,JMIN,KSTRT,KSTOP,0)
      ZMAX=ZMAX+ZMLR12
      ZMIN=ZMIN+ZMLR12
      CALL SORTHS  (MAXMOM,ZMAX,MINMOM,ZMIN,K1,J2,JMAX,JMIN,LMMAX,LMMIN)
      GO TO 82
   80 CALL SORTHS   (MAXMOM,ZMLR12,MINMOM,ZMLR12,K1,J2,K3,K3,LMMAX,LMMIN
     *)
   82 CONTINUE
   84 CONTINUE
C
C     H OR HS TRUCK RIGHT TO LEFT-MOMENTS
C
      DO 96 J1=1,NEXTRM
      JJ=NEXTRM+1-J1
      JSTRT=LEXTRM(JJ)
      JSTOP=JSTRT-NMOVES
      IF(JJ.EQ.1) GO TO 86
      IF(JSTOP.LT.LEXTRM(JJ-1)) JSTOP=LEXTRM(JJ-1)
   86 J2=JSTRT+2
   88 J2=J2-2
      K1=J2-14
```

```
      ZMRL12=.25*INFLM(K1)+INFLM(J2)
      K3=J2+14
      IF(JTRIG)94,94,90
   90 KSTRT=J2+14
      KSTOP=J2+30
      IF(KSTRT.GE.NODSTP) GO TO 94
      IF(KSTOP.GT.NODSTP)KSTOP=NODSTP
   92 CALL SORTIL(INFLM,ZMAX,JMAX,ZMIN,JMIN,KSTRT,KSTOP,0)
      ZMAX=ZMAX+ZMRL12
      ZMIN=ZMIN+ZMRL12
      CALL SORTHS  (MAXMOM,ZMAX,MINMOM,ZMIN,K1,J2,JMAX,JMIN,LMMAX,LMMIN)
      IF(J2.GT.JSTOP) GO TO 88
      GO TO 96
   94 CALL SORTHS  (MAXMOM,ZMRL12,MINMOM,ZMRL12,K1,J2,K3,K3,LMMAX,LMMIN)
   96 CONTINUE
C
C     H OR HS TRUCK LEFT TO RIGHT-SHEAR
C
      DO 110 J1=1,NEXTRV
      JSTRT=LEXTRV(J1)
      JSTOP=JSTRT+NMOVES
      IF(JSTOP.GT.LEXTRV(J1+1)) JSTOP=LEXTRV(J1+1)
      DO 108 J2=JSTRT,JSTOP,2
      K1=J2+14
      ZVLR12=.25*INFLV(K1)+INFLV(J2)
      ZVLR=ZVLR12
      IF(K1.EQ.NDISC) ZVLR=ZVLR-0.25
      IF(J2.EQ.NDISC) ZVLR=ZVLR-1.0
      K3=J2-14
      IF(JTRIG) 106,106,98
   98 KSTRT=J2-30
      KSTOP=J2-14
      IF(KSTOP.LT.200) GO TO 106
      IF(KSTRT.GE.200) GO TO 104
      DO 100 J3=2,30,2
      J4=KSTOP-J3
      IF(J4.GT.200) GO TO 100
      GO TO 102
  100 CONTINUE
  102 KSTRT=J4
  104 CALL SORTIL(INFLV,ZMAX,JMAX,ZMIN,JMIN,KSTRT,KSTOP,NDISC)
      ZMAX=ZMAX+ZVLR12
      ZMIN=ZMIN+ZVLR
      CALL SORTHS  (MAXSHR,ZMAX,MINSHR,ZMIN,K1,J2,JMAX,JMIN,LVMAX,LVMIN)
      GO TO 108
  106 CALL SORTHS  (MAXSHR,ZVLR12,MINSHR,ZVLR,K1,J2,K3,K3,LVMAX,LVMIN)
  108 CONTINUE
  110 CONTINUE
```

```
C
C       H OR HS TRUCK RIGHT TO LEFT-SHEAR
C
      DO 122 J1=1,NEXTRV
      JJ=NEXTRV+1-J1
      JSTRT=LEXTRV(JJ)
      JSTOP=JSTRT-NMOVES
      IF(JJ.EQ.1.AND.JSTOP.LT.200) JSTOP=200
      IF(JJ.EQ.1) GO TO 112
      IF(JSTOP.LT.LEXTRV(JJ-1)) JSTOP=LEXTRV(JJ-1)
  112 J2=JSTRT+2
  114 J2=J2-2
      K1=J2-14
      ZVRL12=.25*INFLV(K1)+INFLV(J2)
      ZVRL=ZVRL12
      IF(K1.EQ.NDISC) ZVRL=ZVRL-.25
      IF(J2.EQ.NDISC)ZVRL=ZVRL-1.0
      K3=J2+14
      IF(JTRIG) 120,120,116
  116 KSTRT=J2+14
      KSTOP=J2+30
      IF(KSTRT.GT.NODSTP) GO TO 120
      IF(KSTOP.GT.NODSTP) KSTOP=NODSTP+1
  118 CALL SORTIL(INFLV,ZMAX,JMAX,ZMIN,JMIN,KSTRT,KSTOP,NDISC)
      ZMAX=ZMAX+ZVRL12
      ZMIN=ZMIN+ZVRL12
      CALL SORTHS  (MAXSHR,ZMAX,MINSHR,ZMIN,K1,J2,JMAX,JMIN,LVMAX,LVMIN)
      GO TO 121
  120 CALL SORTHS  (MAXSHR,ZVLR12,MINSHR,ZVRL,K1,J2,K3,K3,LVMAX,LVMIN)
  121 IF(J2.GT.JSTOP) GO TO 114
  122 CONTINUE
C
C        FINAL CHECK OF MAXIMUM H OR HS MOMENT
C
      JSTP=3
      IF(JTRIG.EQ.-1)JSTP=1
      IF(JTRIG.EQ.-1)LMMAX(3)=LMMAX(2)+(LMMAX(2)-LMMAX(1))*30/14
      DO 123 J1=1,3
      K1=LMMAX(1)+J1-2
      K2=LMMAX(2)+J1-2
      S12 =.25*INFLM(K1)+INFLM(K2)
      DO 123 J2=1,JSTP
      K3=LMMAX(3)+J2-2
      K4=IABS(K2-K3)
      IF(K4.LT.14.OR.K4.GT.30) GO TO 123
      IF(JTRIG.EQ.-1) ZMAX=S12
      IF(JTRIG.EQ.+1) ZMAX=S12+INFLM(K3)
      CALL SORTHS  (MAXMOM,ZMAX,-10000.,ZMAX,K1,K2,K3,K3,LMMAX,LMMIN)
```

```
  123   CONTINUE
C
C        FINAL CHECK OF MINIMUM H OR HS MOMENT
C
        IF(JTRIG.EQ.-1) LMMIN(3)=LMMIN(2)+(LMMIN(2)-LMMIN(1))*30/14
        DO 125 J1=1,3
        K1=LMMIN(1)+J1-2
        K2=LMMIN(2)+J1-2
        S12 =.25*INFLM(K1)+INFLM(K2)
        DO 125 J2=1,JSTP
        K3=LMMIN(3)+J2-2
        K4=IABS(K2-K3)
        IF(K4.LT.14.OR.K4.GT.30) GO TO 125
        IF(JTRIG.EQ.-1) ZMIN=S12
        IF(JTRIG.EQ.+1) ZMIN=S12+INFLM(K3)
        CALL SORTHS  (10000.,ZMIN,MINMOM,ZMIN,K1,K2,K3,K3,LMMAX,LMMIN)
  125   CONTINUE
C
C        FINAL CHECK ON MAXIMUM H OR HS SHEAR
C
        IF(JTRIG.EQ.-1) LVMAX(3)=LVMAX(2)+(LVMAX(2)-LVMAX(1))*30/14
        DO 127 J1=1,3
        K1=LVMAX(1)+J1-2
        K2=LVMAX(2)+J1-2
        S12   =.25*INFLV(K1)+INFLV(K2)
        DO 127 J2=1,JSTP
        K3=LVMAX(3)+J2-2
        K4=IABS(K2-K3)
        IF(K4.LT.14.OR.K4.GT.30) GO TO 127
        IF(JTRIG.EQ.-1) SMAX=S12
        IF(JTRIG.EQ.+1) SMAX=S12+INFLV(K3)
        CALL SORTHS  (MAXSHR,SMAX,MINSHR,SMAX,K1,K2,K3,K3,LVMAX,LVMIN)
  127   CONTINUE
C
C        FINAL CHECK ON H OR HS MINIMUM SHEAR
C
        IF(JTRIG.EQ.-1) LVMIN(3)=LVMIN(2)+(LVMIN(2)-LVMIN(1))*30/14
        DO 227 J1=1,3
        K1=LVMIN(1)+J1-2
        K2=LVMIN(2)+J1-2
        S12   =.25*INFLV(K1)+INFLV(K2)
        IF(K1.EQ.NDISC)SMIN  =S12   -.25
        IF(K2.EQ.NDISC)SMIN  =S12   -1.0
        DO 227 J2=1,JSTP
        K3=LVMIN(3)+J2-2
        K4=IABS(K2-K3)
        IF(K4.LT.14.OR.K4.GT.30) GO TO 227
        IF(JTRIG.EQ.-1) SMIN=S12
```

```
      IF(JTRIG.EQ.+1) SMIN=S12+INFLV(K3)
      IF(JTRIG.EQ.+1.AND.K3.EQ.NDISC) SMIN=SMIN-1.0
      CALL SORTHS(MAXSHR,SMIN,MINSHR,SMIN,K1,K2,K3,K3,LVMAX,LVMIN)
C
  227 CONTINUE
C
C
C     COMPUTE WHEEL POSITIONS
C
 129  DO 124 J1=1,3
       LMMAX(J1)=LMMAX(J1)-200
       LMMIN(J1)=LMMIN(J1)-200
       LVMIN(J1)=LVMIN(J1)-200
 124   LVMAX(J1)=LVMAX(J1)-200
      RIMP=1.0
      CALL IMPACT(1,RIMP)
      MAXMOM=MAXMOM*SCLHHS*RIMP
      CALL IMPACT(2,RIMP)
      MINMOM=MINMOM*SCLHHS*RIMP
      CALL IMPACT(3,RIMP)
      MAXSHR=MAXSHR*SCLHHS*RIMP
      MINSHR=MINSHR*SCLHHS*RIMP
      IF(LLMAX(JPNT,JSPAN).GE.MAXMOM) GO TO 310
      LLMAX(JPNT,JSPAN)=MAXMOM
      LODCON(JPNT,JSPAN,1)=1
  310 IF(LLMIN(JPNT,JSPAN).LE.MINMOM) GO TO 320
      LLMIN(JPNT,JSPAN)=MINMOM
      LODCON (JPNT,JSPAN,2)=1
  320 IF(LLVAX(JPNT,JSPAN).GE.MAXSHR) GO TO 330
      LLVAX(JPNT,JSPAN)=MAXSHR
      LODCON(JPNT,JSPAN,3)=1
  330 IF(LLVIN(JPNT,JSPAN).LE.MINSHR) GO TO 340
      LLVIN(JPNT,JSPAN)=MINSHR
      LODCON(JPNT,JSPAN,4)=1
  340 CONTINUE
      CALL OUTIL(1)
C************************************************************************
C********AXLE TRAIN*******************************************************
C************************************************************************
C
C     AXLE TRAIN LEFT TO RIGHT-MOMENTS
C
  126 MAXSHR=-10000.
      MAXMOM=-10000.
      MINSHR=10000.
      MINMOM=10000.
      IF(LODKOD(3).EQ.0) GO TO 172
      LEXTRM(NEXTRM+1)=LEXTRM(NEXTRM)+NWHL(NWHEEL-1)
```

```
      LEXTRV(NEXTRV+1)=LEXTRV(NEXTRV)+NWHL(NWHEEL-1)
      DO 130 J1=1,NEXTRM
      JSTRT=LEXTRM(J1)
      JSTOP=JSTRT+NWHL(NWHEEL-1)
      IF(JSTOP.GT.LEXTRM(J1+1))JSTOP=LEXTRM(J1+1)
      DO 130 J2=JSTRT,JSTOP,2
      ZMOM=INFLM(J2)*PWHEEL(1)
      DO 128 J3=2,NWHEEL
  128 ZMOM=ZMOM+INFLM(J2-NWHL(J3-1))*PWHEEL(J3)
  130 CALL SORTAX  (MAXMOM,ZMOM,MINMOM,ZMOM,J2,-1,NWHL,NWHEEL,
     1  LMMAX,LMMIN)
C
C     AXLE TRAIN RIGHT TO LEFT-MOMENTS
C
      DO 136 J1=1,NEXTRM
      JJ=NEXTRM+1-J1
      JSTRT=LEXTRM(JJ)
      JSTOP=JSTRT-NWHL(NWHEEL-1)
      IF(JJ.EQ.1) GO TO 131
      IF(JSTOP.LT.LEXTRM(JJ-1))  JSTOP=LEXTRM(JJ-1)
  131 J2=JSTRT+2
  132 J2=J2-2
      ZMOM=INFLM(J2)*PWHEEL(1)
      DO 134 J3=2,NWHEEL
  134 ZMOM=ZMOM+INFLM(J2+NWHL(J3-1))*PWHEEL(J3)
      CALL SORTAX  (MAXMOM,ZMOM,MINMOM,ZMOM,J2,1,NWHL,NWHEEL,
     1  LMMAX,LMMIN)
      IF(J2.GT.JSTOP) GO TO 132
  136 CONTINUE
C
C     AXLE TRAIN LEFT TO RIGHT-SHEAR
C
      DO  142 J1=1,NEXTRV
      JSTRT=LEXTRV(J1)
      JSTOP=JSTRT+NWHL(NWHEEL-1)
      IF(JSTOP.GT.LEXTRV(J1+1))  JSTOP=LEXTRV(J1+1)
      DO 142 J2=JSTRT,JSTOP
      ZVAX=INFLV(J2)*PWHEEL(1)
      DO 138 J3=2,NWHEEL
  138 ZVAX=ZVAX+INFLV(J2-NWHL(J3-1))*PWHEEL(J3)
      ZVIN=ZVAX
      IF(J2.EQ.NDISC)  ZVIN=ZVIN-PWHEEL(1)
      IF(J2.EQ.NDISC)  GO TO 142
      DO 140 J3=2,NWHEEL
      J4=J2-NWHL(J3-1)
      IF(J4.NE.NDISC)  GO TO 140
      ZVIN=ZVIN-PWHEEL(J3)
      GO TO 142
```

```
  140   CONTINUE
  142   CALL SORTAX  (MAXSHR,ZVAX,MINSHR,ZVIN,J2,-1,NWHL,NWHEEL,
       1  LVMAX,LVMIN)
C
C       AXLE TRAIN RIGHT TO LEFT-SHEAR
C
        DO 152 J1=1,NEXTRV
        JJ=NEXTRV+1-J1
        JSTRT=LEXTRV(JJ)
        JSTOP=JSTRT-NWHL(NWHEEL-1)
        IF(JJ.EQ.1) GO TO 143
        IF(JSTOP.LT.LEXTRV(JJ-1)) JSTOP=LEXTRV(JJ-1)
  143   J2=JSTRT+2
  144   J2=J2-2
        ZVAX=INFLV(J2)*PWHEEL(1)
        DO 146 J3=2,NWHEEL
  146   ZVAX=ZVAX+INFLV(J2+NWHL(J3-1))*PWHEEL(J3)
        ZVIN=ZVAX
        IF(J2.EQ.NDISC) ZVIN=ZVIN-PWHEEL(1)
        IF(J2.EQ.NDISC) GO TO 150
        DO 148 J3=2,NWHEEL
        J4=J2+NWHL(J3-1)
        IF(J4.NE.NDISC) GO TO 148
        ZVIN=ZVIN-PWHEEL(J3)
        GO TO 150
  148   CONTINUE
  150   CALL SORTAX  (MAXSHR,ZVAX,MINSHR,ZVIN,J2,1,NWHL,NWHEEL,
       1  LVMAX,LVMIN)
        IF(J2.GT.JSTOP) GO TO 144
  152   CONTINUE
C
C       FINAL CHECK OF MAXIMUM AXLE TRAIN MOMENT
C
        DO 156 J1=1,3,2
        J2=J1-2
        ZMAX=0.
        DO 154 J3=1,NWHEEL
  154   ZMAX=ZMAX+INFLM(LMMAX(J3)+J2)*PWHEEL(J3)
        KDIRT=-1
        IF(LMMAX(2).GT.LMMAX(1))KDIRT=1
        J4=LMMAX(1)+J2
  156   CALL SORTAX  (MAXMOM,ZMAX, MINMOM,ZMAX,J4,KDIRT,NWHL,NWHEEL,
       1  LMMAX,LMMIN)
C
C       FINAL CHECK OF MINIMUM AXLE TRAIN MOMENT
C
        DO 160 J1=1,3,2
        J2=J1-2
```

```
      ZMIN=0.
      DO 158 J3=1,NWHEEL
  158 ZMIN=ZMIN+INFLM(LMMIN(J3)+J2)*PWHEEL(J3)
      KDIRT=-1
      IF(LMMIN(2).GT.LMMIN(1)) KDIRT=1
      J4=LMMIN(1)+J2
  150 CALL SORTAX  (MAXMOM,ZMIN,MINMOM,ZMIN,J4,KDIRT,NWHL,NWHEEL,
     1  LMMAX,LMMIN)
C
C     FINAL CHECK OF MAXIMUM AXLE TRAIN SHEAR
C
      DO 168 J1=1,3,2
      J2=J1-2
      ZVAX=0.
      DO 162 J3=1,NWHEEL
  162 ZVAX=ZVAX+INFLV(LVMAX(J3)+J2)*PWHEEL(J3)
  166 KDIRT=-1
      IF(LVMAX(2).GT.LVMAX(1)) KDIRT=1
      J4=LVMAX(1)+J2
  168 CALL SORTAX  (MAXSHR,ZVAX,MINSHR,ZVAX,J4,KDIRT,NWHL,NWHEEL,
     1   LVMAX,LVMIN)
C
C      FINAL CHECK OF MINIMUM AXLE TRAIN SHEAR
C
      DO 268 J1=1,3,2
      J2=J1-2
      ZVIN=0.
      DO 262 J3=1,NWHEEL
  262 ZVIN=ZVIN+INFLV(LVMIN(J3)+J2)*PWHEEL(J3)
      DO 264 J3=1,NWHEEL
      IF(LVMIN(J3)+J2.NE.NDISC) GO TO 264
      ZVIN=ZVIN-PWHEEL(J3)
      GO TO 266
  264 CONTINUE
  266 KDIRT=-1
      IF(LVMIN(2).GT.LVMIN(1))KDIRT=1
      J4=LVMIN(1)+J2
  268 CALL SORTAX(MAXSHR,ZVIN,MINSHR,ZVIN,J4,KDIRT,NWHL,NWHEEL,LVMAX,
     1LVMIN   )
C
C     COMPUTE WHEEL POSITIONS
C
      DO 170 J1=1, NWHEEL
       LMMAX(J1)=LMMAX(J1)-200
       LMMIN(J1)=LMMIN(J1)-200
       LVMIN(J1)=LVMIN(J1)-200
  170  LVMAX(J1)=LVMAX(J1)-200
      AXIMP=1.0
```

```
      IF(KOVRD.EQ.YES) CALL IMPACT(1,AXIMP)
      MAXMOM=MAXMOM*FACTOR*AXIMP
      IF(KOVRD.EQ.YES) CALL IMPACT(2,AXIMP)
      MINMOM=MINMOM*FACTOR*AXIMP
      IF(KOVRD.EQ.YES) CALL IMPACT(3,AXIMP)
      MAXSHR=MAXSHR*FACTOR*AXIMP
      MINSHR=MINSHR*FACTOR*AXIMP
      IF(LLMAX(JPNT,JSPAN).GE.MAXMOM) GO TO 342
      LLMAX(JPNT,JSPAN)=MAXMOM
      LODCON(JPNT,JSPAN,1)=2
  342 IF(LLMIN(JPNT,JSPAN).LE.MINMOM) GO TO 344
      LLMIN(JPNT,JSPAN)=MINMOM
      LODCON (JPNT,JSPAN,2)=2
  344 IF(LLVAX(JPNT,JSPAN).GE.MAXSHR) GO TO 346
      LLVAX(JPNT,JSPAN)=MAXSHR
      LODCON(JPNT,JSPAN,3)=2
  346 IF(LLVIN(JPNT,JSPAN).LE.MINSHR) GO TO 348
      LLVIN(JPNT,JSPAN)=MINSHR
      LODCON(JPNT,JSPAN,4)=2
  348 CONTINUE
      CALL OUTIL(2)
C*************************************************************************
C*******LANE LOADING******************************************************
C*************************************************************************
  172 IF(LODKOD(1).EQ.0.AND.LODKOD(2).EQ.0) GO TO 186
      J1=NDISC/2
      J2=J1*2
      IF(J2.EQ.NDISC) JSTRT=200
      IF(J2.LT.NDISC) JSTRT=199
      JSTOP=NODSTP+2
C
C     SELECT MAX.AND MIN.EXTREMES OF INFLM(I) AND INFLV(I)
C
      ZMAX1=0.
      ZMIN1=0.
      ZMIN2=0.
      ZVAX=0.
      ZVIN=0.
      J2=0
      DO 176 J1=1,NEXTRM
      Q=INFLM(LEXTRM(J1))
      IF(Q.LE.ZMAX1) GO TO 174
      ZMAX1=Q
  174 IF(Q.GE.ZMIN1) GO TO 176
      ZMIN1=Q
      J2=LEXTRM(J1)
  176 CONTINUE
      IF(J2.EQ.0) GO TO 180
```

```
      STORE=INFLM(J2)
      INFLM(J2)=10000.
      DO 178 J1=1,NEXTRM
      Q=INFLM(LEXTRM(J1))
      IF(Q.GE.ZMIN2) GO TO 178
      ZMIN2=Q
  178 CONTINUE
      INFLM(J2)=STORE
  180 DO 183 J1=1,NEXTRV
      J2=LEXTRV(J1)
      Q=INFLV(J2)
      IF(Q.LE.ZVAX) GO TO 182
      ZVAX=Q
  182 IF(Q.GE.ZVIN) GO TO 185
      ZVIN=Q
  185 IF(J2.NE.NDISC) GO TO 183
      Q=INFLV(J2)-1.
      IF(Q.GE.ZVIN) GO TO 183
      ZVIN=Q
  183 CONTINUE
C
C     NUMERICAL INTEGRATION OF POSITIVE AND NEGATIVE AREAS UNDER
C     INFLM(I) AND INFLV(I) CURVES
C
      MAXMOM=0.
      MINMOM=0.
      SVAX=0.
      SVIN=0.
      DO 184 J1=JSTRT,JSTOP,2
      AM=INFLM(J1)+INFLM(J1+2)
      AV=INFLV(J1)+INFLV(J1+2)
      IF(AM.LT.0.) MINMOM=MINMOM+AM
      IF(AM.GT.0.) MAXMOM=MAXMOM+AM
      IF(AV.LT.0.) SVIN=SVIN+AV
      IF(AV.GT.0.) SVAX=SVAX+AV
 184  CONTINUE
      AV=INFLV(NDISC)+INFLV(NDISC-1)
      IF(AV.GT.0.) SVAX=SVAX-AV
      IF(AV.GT.0.) SVIN=SVIN-AV
      SVIN=SVIN-1.
      AREAM=MAXMOM+MINMOM
      AREAV=SVAX+SVIN
      IF(JPNT.EQ.1.AND.JSPAN.EQ.1) AREAV=AREAV+1.0
      CALL IMPACT(1,RIMP)
      MAXMOM=MAXMOM*SCLLNE*RIMP+ZMAX1*SCLCOM*RIMP
      CALL IMPACT(2,RIMP)
      MINMOM=MINMOM*SCLLNE*RIMP+(ZMIN1+ZMIN2)*SCLCOM*RIMP
      CALL IMPACT(3,RIMP)
```

```
      MAXSHR=SVAX*SCLLNE*RIMP+ZVAX*SCLCOV*RIMP
      MINSHR=SVIN*SCLLNE*RIMP+ZVIN*SCLCOV*RIMP
      IF(LLMAX(JPNT,JSPAN).GE.MAXMOM) GO TO 350
      LLMAX(JPNT,JSPAN)=MAXMOM
      LODCON(JPNT,JSPAN,1)=3
  350 IF(LLMIN(JPNT,JSPAN).LE.MINMOM) GO TO 352
      LLMIN(JPNT,JSPAN)=MINMOM
      LODCON (JPNT,JSPAN,2)=3
  352 IF(LLVAX(JPNT,JSPAN).GE.MAXSHR) GO TO 354
      LLVAX(JPNT,JSPAN)=MAXSHR
      LODCON(JPNT,JSPAN,3)=3
  354 IF(LLVIN(JPNT,JSPAN).LE.MINSHR) GO TO 356
      LLVIN(JPNT,JSPAN)=MINSHR
      LODCON(JPNT,JSPAN,4)=3
  356 CONTINUE
      CALL OUTIL(3)
C
C*************************************************************************
C*********UNIFORMLY DISTRIBUTED LIVE LOAD*********************************
C*************************************************************************
C
  186 IF(LODKOD(4).EQ.0) GO TO 192
      IF(LODKOD(1).EQ.1.OR.LODKOD(2).EQ.1) GO TO 190
      AREAM=0.
      AREAV=0.
      J1=NDISC/2
      J2=J1*2
      IF(J2.EQ.NDISC) JSTRT=200
      IF(J2.LT.NDISC) JSTRT=199
      JSTOP=NODSTP+2
      DO 188 J1=JSTRT,JSTOP,2
      AREAM=AREAM+INFLM(J1)+INFLM(J1+2)
  188 AREAV=AREAV+INFLV(J1)+INFLV(J1+2)
      AREAV=AREAV-1.0
      IF(JPNT.EQ.1.AND.JSPAN.EQ.1) AREAV=AREAV+1.0
  190 MAXMOM=AREAM*UNIFLL
      MAXSHR=AREAV*UNIFLL
      LLMIN(JPNT,JSPAN)=LLMIN(JPNT,JSPAN)+MAXMOM
      LLMAX(JPNT,JSPAN)=LLMAX(JPNT,JSPAN)+MAXMOM
      LLVAX(JPNT,JSPAN)=LLVAX(JPNT,JSPAN)+MAXSHR
      LLVIN(JPNT,JSPAN)=LLVIN(JPNT,JSPAN )+MAXSHR
      IF(LODKOD(1).EQ.1.OR.LODKOD(2).EQ.1.OR.LODKOD(3).EQ.1) GO TO 197
      LODCON(JPNT,JSPAN,1)=4
      LODCON(JPNT,JSPAN,2)=4
      LODCON(JPNT,JSPAN,3)=4
      LODCON(JPNT,JSPAN,4)=4
  197 CONTINUE
      CALL OUTIL(4)
```

```
  192 CONTINUE
C
C
C*****************************************************************************
C********CONCENTRATED LOADS ON CONTINUOUS SPAN*************************
C*****************************************************************************
      IF(LODKOD(5).EQ.0) GO TO 900
      DO 193 KCL=1,NCOUNT
      NOD=POSCMP(KCL)
      DLMOM(JPNT,JSPAN)=INFLM(NOD)*FCOMP (KCL)+DLMOM(JPNT,JSPAN)
  193 DLSHR(JPNT,JSPAN)=INFLV(NOD)*FCOMP(KCL)+DLSHR(JPNT,JSPAN)
C
C
C
C*****************************************************************************
C******** BEAM AND SLAB WEIGHT ON CONTINUOUS BEAM *********************
C*****************************************************************************
  900 IF(LODKOD(7).EQ.0) GO TO 1001
      IF(LODKOD(1).EQ.1.OR.LODKOD(2).EQ.1.OR.LODKOD(4).EQ.1) GO TO 920
      AREAM=0.
      AREAV=0.
      J1=NDISC/2
      J2=J1*2
      IF(J2.EQ.NDISC) JSTRT=200
      IF(J2.LT.NDISC) JSTRT=199
      JSTOP=NODSTP+2
      DO 910 J1=JSTRT,JSTOP,2
      AREAM=AREAM+INFLM(J1)+INFLM(J1+2)
  910 AREAV=AREAV+INFLV(J1)+INFLV(J1+2)
      AREAV=AREAV-1.0
      IF(JPNT.EQ.1.AND.JSPAN.EQ.1) AREAV=AREAV+1.0
  920 WUNIF=BMWT+((S*THK)*.15/12.)
      DLMOM(JPNT,JSPAN)=DLMOM(JPNT,JSPAN)+AREAM*WUNIF
      DLSHR(JPNT,JSPAN)=DLSHR(JPNT,JSPAN)+AREAV*WUNIF
 1001 CONTINUE
      J1=N/2
      JP=J1+1
      IF(KODE.EQ.0.AND.IEVEN.EQ.1.AND.JSPAN.EQ.J1.AND.JPNT.EQ.11)KODE=10
      IF(KODE.EQ.0.AND.IEVEN.EQ.0.AND.JSPAN.EQ.JP.AND.JPNT.EQ.6)KODE=10
      CALL OUTIL (5)
      IF(KOUTPT.EQ.0) CALL OUTKO(1)
C
      IF(KODE.EQ.10)GO TO 369
 1000 CONTINUE
  369 WRITE(6,379)
  379 FORMAT(1H1)
      KKT=2
      GO TO 2
```

END

```
      SUBROUTINE INPUT(KKT)
      INTEGER POSCMP
      REAL*4 INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMCM,MINSHR,L
     1,LLVIN
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,10),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCCMP(100)
      COMMON/BLK2/LMMIN(15),LMMAX(15),LVMAX(15),NWHL(14),NODDSN(11,10),P
     1OSCMP(100),LODKOD(7),LEXTRM(30),LEXTRV(30),LVMIN(15),LODCON(11,10,
     24)
      COMMON/BLK3/MINMCM,MAXMCM,MAXSHR,MINSHR,SCLLNE,SCLCOM,SCLHHS,SCLCO
     1V,UNIFLL,BMWT,S,THK
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCOUNT
      COMMON/BLK5/ITITLE(80),JTITLE(80),KTITLE(80),JBMDSG(2),KAASHO(2),K
     1AXTR,KOUTPT,KCCNT,KODE,KCVRD,FACTOR
      DIMENSION KASHO(6)
      INTEGER CLLOCA,SPANNO
      DIMENSION CL(20),CLLOCA(20)
      DATA IJK/4HY    /,KASKIP/4H    /,KASHO(1),KASHO(2),KASHO(3),KASHO(4
     $),KASHO(5),KASHO(6)/4H H 1,4HH  1,4H H 2,4HH  2,4HHS 1,4HHS 2/
      IF(KKT.EQ.2)GO TO 101
      READ(5,100)(ITITLE(J1),J1=1,80)
      READ(5,100)(JTITLE(J1),J1=1,80)
      READ(5,100)(KTITLE(J1),J1=1,80)
  100 FORMAT(80A1)
  101 READ(5,104,END=1000)JBMDSG(1),JBMDSG(2),BMWT,S,FACTOR,THK,KAASHO(1
     $),KAASHO(2),UNIFLL,KAXTR,KCONT,KCCN,KOUTPT
  104 FORMAT(4X,A4,A1,3X,F5.3,3X,F5.2,3X,F3.1,2X,F5.2,2X,A4,A1,3X,F5.3,3
     $X,A1,6X,A1,6X,A1,6X,I1)
      READ(5,105)(L(J1),J1=1,10),KOVRD
  105 FORMAT(5X,10(F3.0,3X),3X,A1)
      LODKOD(3)=0
      DO 99 J1=1,10
      DO 99 J2=1,11
      DLMOM(J2,J1)=0.
   99 DLSHR(J2,J1)=0.
      IF(FACTOR.EQ.0.) FACTOR=S/11.0
      IF(KAXTR.NE.IJK) GO TO 36
      LODKOD(3)=1
      READ(5,106)(PWHEEL(J1),J1=1,15)
  106 FORMAT(4X,15(F3.1,1X))
      READ(5,108)(NWHL(J1),J1=1,14)
  108 FORMAT(8X,14(I3,1X))
C
C       COMPUTE NUMBER OF AXLES IN AXLE TRAIN
C
      NWHEEL=0
      DO 160 I=1,15
```

```
      IF(NWHEEL.NE.0) GO TO 161
      IF(NWHL(I).EQ.0) NWHEEL=I
  161 CONTINUE
  160 CONTINUE
C
C      DETERMINE NUMBER OF SPANS
C
   36 N=0
      DO 38 J1=1,10
      IF(L(J1).EQ.0.) GO TO 40
   38 N=N+1
C
C
C      COMPUTE SL(I) ARRAY
C
   40 SUM=0.
      DO 42 J1=1,N
      SUM=SUM+L(J1)
      SL(J1)=SUM
   42 CONTINUE
C           ZERO LODCON MATRIX
      DO 31 I=1,11
      DO 31 J=1,10
      DO 31 K=1,4
   31 LODCON(I,J,K)=0
C
C      WRITE OUT INPUT
C
      WRITE(6,200)
  200 FORMAT(1H1,///)
      WRITE(6,230) (ITITLE(J2),J2=9,28),(ITITLE(J2),J2=49,58),
     *(ITITLE(J2),J2=71,80)
  230 FORMAT(25X,'DISTRICT',20A1,'COUNTY   HIGHWAY NO.',10A1,
     *'SUBMITTED BY',10A1)
      WRITE(6,232) (JTITLE(J2),J2=12,22),(JTITLE(J2),J2=26,32),
     *(JTITLE(J2),J2=37,55)
  232 FORMAT(25X,'CONTROL NO.',11A1,'IPE',7A1,'DATE',
     *19A1)
      WRITE(6,234) (KTITLE(J2),J2=13,80)
  234 FORMAT(25X,'DESCRIPTION',1X,68A1)
      WRITE(6,203)JBMDSG(1),JBMDSG(2)
  203 FORMAT(///,40X,'PROPERTIES OF BEAM DESIGNATED',2X,A4,A1)
      IF(KCONT.EQ.IJK) WRITE(6,204)
  204 FORMAT(//,25X,'BEAM IS CONTINUOUS FOR SLAB AND BEAM WEIGHTS')
      IF(KCONT.NE.IJK) WRITE(6,206)
  206 FORMAT(//,25X,'EACH SPAN OF BEAM ACTS AS SIMPLE BEAM IN CARRYING S
     1LAB AND BEAM WEIGHTS')
      WRITE(6,208)BMWT,S,THK
```

```
  208 FORMAT(//,25X,'BEAM WEIGHT =',F6.3,' (KIPS PER FOOT)',/,25X,'BEAM
     1SPACING =',F6.2,' (FT)',/,25X,'SLAB THICKNESS =',F5.2,' (IN)')
      WRITE(6,210)
  210 FORMAT(//,25X,'SPAN LENGTHS (FT)')
      WRITE(6,212)(J1,L(J1),J1=1,N)
  212 FORMAT(25X,'L(',I2,')=',2X,F4.0)
      WRITE(6,214)
  214 FORMAT(//,40X,'LOADING CONDITIONS')
      WRITE(6,216)KAASHO(1),KAASHO(2)
  216 FORMAT(25X,A4,A4)
      WRITE(6,215) FACTOR
  215 FORMAT(25X,'THE LATERAL DISTRIBUTION FACTOR =',F6.3)
      IF(UNIFLL.NE.0)WRITE(6,218)UNIFLL
  218 FORMAT(25X,'UNIFORM L.L. =',F6.3,' KIPS PER FT')
      IF(KAXTR.NE.IJK) GO TO 219
      WRITE(6,220) (PWHEEL(I),I=1,15)
  220 FORMAT(25X,'AXLE TRAIN',//,30X,'LOAD ON AXLE ',/,28X,15(F5.1),//,3
     10X,'POSITION OF AXLE RELATIVE TO AXLE 1 (FT)',/)
      WRITE(6,221) (NWHL(I),I=1,14)
  221 FORMAT(33X,14(I5))
C
C        ESTABLISH NODE NUMBERS OF DESIGN POINTS
C
  219 J4=1
      DO 48 J1=1,N
      IF(J1.EQ.1) S1=-L(J1)/10.
      IF(J1.GT.1) S1=SL(J1-1)-L(J1)/10.
      DEL=L(J1)/10.
      DO 48 J2=1,11
      S1=S1+DEL
      DO 44 J3=J4,2000
      S2=FLOAT(J3)
      S2MS1=S2-S1
      IF(S2MS1.LT.0.) GO TO 44
      J5=J3
      GO TO 46
   44 CONTINUE
   46 J4=J5
      IF(S2MS1.GT.0.5) S2=S2-.99
      NODDSN(J2,J1)=200+INT(S2)
   48 CONTINUE
C
      LODKOD(5)=0
      LODKOD(6)=0
      IF(KCON.NE.IJK) GO TO 78
      NCOUNT=1
C
C        ZERO STORAGE LOCATIONS
```

```
C
      DO 63 I=1,100
      FCOMP(I)=0.
   63 POSCMP(I)=0.
   57 READ(5,112) SPANNO,(CL(J),J=1,20)
      IF(SPANNO.EQ.0) GO TO 78
C
C
      READ(5,114) (CLLOCA(J),J=1,20)
  112 FORMAT(1X,I2,1X,10F3.1,5X,10F3.1)
  114 FORMAT(4X,10I3,5X,10I3)
C
C       DETERMINE NUMBER OF CONCENTRATED FORCES APPLIED TO CONTINUOUS
C       AND NON-CONTINUOUS BEAM
C
      NLONCO=0
      ISKIP=0
      NLOCO=0
      DO 58 K=1,20
      IF(CLLOCA(K).EQ.0.AND.K.EQ.1)NLONCO=100
      IF(NLONCO.NE.0) GO TO 89
      IF(CLLOCA(K).EQ.0.AND.K.LE.10)NLONCO=K-1
   89 IF(CLLOCA(K).EQ.0.AND.K.EQ.11) ISKIP=1
      IF(ISKIP.EQ.1) GO TO 58
      IF(NLOCO.NE.0) GO TO 58
      IF(CLLOCA(K).EQ.0.AND.K.GT.11) NLOCO=K-1
   58 CONTINUE
      IF(NLONCO.EQ.100)NLONCO=0
      IF(NLONCO.EQ.0) GO TO 87
      IF(KCONT.EQ.IJK.AND.NLONCO.NE.0) WRITE(6,115)
      IF(KCONT.EQ.IJK.AND.NLONCO.NE.0) STOP
  115 FORMAT(1X,4(/),'*****CONCENTRATED LOADS FOR A SIMPLE SUPPORTED SPA
     1N HAVE BEEN ENTERED',/, 4X,'WHEN THE BEAM IS CONTINUOUS****',//,1X
     2,'*****CHECK DL INPUT CARD COLS 4 THRU 34 *****')
      WRITE(6,116)
  116 FORMAT(//,25X,'CONCENTRATED LOADS ON NON-CONTINUOUS BEAMS')
      WRITE(6,117) SPANNO
  117 FORMAT(/,28X,'SPAN NUMBER    ',I2)
      WRITE(6,118)(CL(J),J=1,10)
  118 FORMAT(///,30X,'LOAD',5X,10(F4.1,2X))
      WRITE(6,119)(CLLOCA(J),J=1,10)
  119 FORMAT(/,20X,'POSITION FROM',/,20X,'LEFT SUPPORT', 7X,10(I4,2X),/,
     120X,'OF SPAN (FT)')
      LODKOD(6)=1
C
C        COMPUTE MOMENT AND SHEARS DUE TO CONCENTRATED FORCES ON
C         NON-CONTINUOUS BEAM
C
```

```
      DO 66 JP=1,NLONCO
      LOCA=CLLOCA(JP)
      DO 66 JCOMP=1,11
      ISWIT=(JCOMP-1)*L(SPANNO)*.1
      IF(ISWIT.EQ.0) GO TO 21
      IF(ISWIT.EQ.L(SPANNO)) GO TO 23
      IF(LOCA.LE.ISWIT) GO TO 20
      IF(LOCA.GT.ISWIT) GO TO 22
   20 DLMOM(JCOMP,SPANNO)=CL(JP)*(LOCA/ISWIT)*((ISWIT*(L(SPANNO)-ISWIT))
     1/L(SPANNO))
      DLSHR(JCOMP,SPANNO)=CL(JP)*(LOCA/ISWIT)*((JCOMP-1)*.1)
      GO TO 66
   22 ISLIDE=CLLOCA(JP)-ISWIT
      DLMOM(JCOMP,SPANNO)=CL(JP)*((L(SPANNO)-ISLIDE-ISWIT)/(L(SPANNO)-IS
     1WIT))*((ISWIT*(L(SPANNO)-ISWIT))/L(SPANNO))
      DLSHR(JCOMP,SPANNO)=-CL(JP)*(1-(ISLIDE/(L(SPANNO)-ISWIT)))*(1-((JC
     1CMP-1)*.1))
      GO TO 66
   23 DLMOM(JCOMP,SPANNO)=CL(JP)*(LOCA/L(SPANNO))
      DLSHR(JCOMP,SPANNO)=CL(JP)*(LOCA/L(SPANNO))
      GO TO 66
   21 DLMOM(JCOMP,SPANNO)= CL(JP)*(1-(LOCA/L(SPANNO)))
      DLSHR(JCOMP,SPANNO)= CL(JP)*(1-(LOCA/L(SPANNO)))
   66 CONTINUE
C
C        STORE CONCENTRATED FORCES ON CONTINUOUS BEAM
C
   87 IF(ISKIP.EQ.1) GO TO 61
      SUM=0
      SPANM=SPANNO-1
      IF(SPANM.EQ.0) SUM=200
      IF(SPANM.GE.1) SUM=SL(SPANM)+200
      DO 59 K=11,NLOCO
      NODE=CLLOCA(K)+SUM
      POSCMP(NCOUNT)=NODE
      FCOMP(NCOUNT)=CL(K)+FCOMP(NCOUNT)
   59 NCOUNT=NCOUNT+1
      NCOUNT=NCOUNT-1
      WRITE(6,120)
  120 FORMAT(//,25X,'CONCENTRATED LOADS ON CONTINUOUS BEAMS')
      WRITE(6,117)SPANNO
      WRITE(6,118)(CL(J),J=11,20)
      WRITE(6,119)(CLLOCA(J),J=11,20)
      LODKOD(5)=1
   61 GO TO 57
C
C        SET UP LODKOD(I) ARRAY, I=1,2
C
```

```
   78 IF(KAASHO(1).EQ.KASKIP) GO TO 69
      DO 139 J6=1,4
      IF(KAASHO(1).EQ.KASHO(J6)) GO TO 80
  139 CONTINUE
      DO 149 J6=5,6
      IF(KAASHO(1).EQ.KASHO(J6)) GO TO 88
  149 CONTINUE
      GO TO 85
C
C       H TRUCK
C
   80 IF(KAASHO(1).EQ.KASHO(1).OR.KAASHO(1).EQ.KASHO(2)) GO TO 82
      GO TO 86
C
C       H-15 TRUCK
C
   82 LODKOD(1)=1
      LODKOD(2)=0
      SCLHHS=24.
      SCLLNE=.480
      SCLCOM=13.5
      SCLCOV=19.5
      GO TO 96
   85 WRITE(6,159)
  159 FORMAT(1X,130(1H*),/,1X,28(1H*),'UNRECOGNIZABLE AASHO TRUCK LOADIN
     1G-CHECK INPUT CARD 4, COLS. 37 THRU 41',28(1H*),/,1X,130(1H*))
      STOP
C
C       H-20 TRUCK
C
   86 LODKOD(1)=1
      LODKOD(2)=0
      SCLHHS=32.
      SCLLNE=.640
      SCLCOM=18.
      SCLCOV=26.
      GO TO 96
C
C       HS TRUCK
C
   88 IF(KAASHO(1).EQ.KASHO(5)) GO TO 90
      GO TO 94
C
C       HS-15 TRUCK
C
   90 LODKOD(1)=0
      LODKOD(2)=1
      SCLHHS=24.
```

```
      SCLLNE=.480
      SCLCOM=13.5
      SCLCOV=19.5
      GO TO 96
C
C      HS-20 TRUCK
C
   94 LODKOD(1)=0
      LODKOD(2)=1
      SCLHHS=32.
      SCLLNE=.640
      SCLCOM=18.
      SCLCOV=26.
   96 SCLHHS=SCLHHS*  FACTOR
      SCLLNE=      SCLLNE*FACTOR
      SCLCOM=      SCLCCM*FACTOR
      SCLCOV=      SCLCCV*FACTOR
      GO TO 68
   69 LODKOD(1)=0
      LODKOD(2)=0
C
C      COMPUTE SHEAR AND MOMENTS AT DESIGN POINTS DUE TO SLAB PLUS BEAM
C      WEIGHT IF BEAM IS NOT CONTINUOUS FOR DEAD LOAD
C
   68 LODKOD(4)=0
      IF(ABS(UNIFLL).GT..00099) LODKOD(4)=1
      IF(KCONT.EQ.IJK) GO TO 133
      WUNIF=BMWT+S*THK*.15/12.
      LODKOD(7)=0
      DO 98 J1=1,N
      RL=WUNIF*L(J1)/2.
      DO 98 J2=1,11
      JS=J2-1
      XARM=FLOAT(JS)*L(J1)/10.
      DLMOM(J2,J1)=DLMOM(J2,J1)+WUNIF*(XARM/2)*(L(J1)-XARM)
   98 DLSHR(J2,J1)=DLSHR(J2,J1)-RL+WUNIF*XARM
      RETURN
  133 CONTINUE
      LODKOD(7)=1
      RETURN
 1000 STOP
      END
```

```
      SUBROUTINE SORTAX      (MAXEFT,ZMAX,MINEFT,ZMIN,J1,KDIRT,NWHL,JW,
     1  LMAX,LMIN)
      REAL*4 MAXEFT,MINEFT
      DIMENSION NWHL(14),LMAX(15),LMIN(15)
C
C
C
C     COMPARE MOMENT AT CURRENT POINT TO PREVIOUS MAX. AND MIN. VALUES
C
  100 IF(ZMAX.LE.MAXEFT) GO TO 110
      MAXEFT=ZMAX
      LMAX(1)=J1
      DO 102 J2=2,JW
  102 LMAX(J2)=J1+KDIRT*NWHL(J2-1)
  110 IF(ZMIN.GE.MINEFT) RETURN
      MINEFT=ZMIN
      LMIN(1)=J1
      DO 112 J2=2,JW
  112 LMIN(J2)=J1+KDIRT*NWHL(J2-1)
      RETURN
      END
```

```
      SUBROUTINE SORTIL(ZINF,ZMAX,JMAX,ZMIN,JMIN,KSTRT,KSTOP,JJ)
      DIMENSION ZINF(1000)
      ZMIN=10000.
      ZMAX=-10000.
      DO 12 J2=KSTRT,KSTOP,2
      Z=ZINF(J2)
      IF(Z.LE.ZMAX) GO TO 10
      ZMAX=Z
      JMAX=J2
   10 IF(Z.GE.ZMIN) GO TO 12
      ZMIN=Z
      JMIN=J2
   12 CONTINUE
      IF(JJ.EQ.0) RETURN
      IF(KSTRT.LE.JJ.AND.KSTOP.GE.JJ) GO TO 14
      RETURN
   14 Z=ZINF(JJ)-1.0
      IF(Z.GE.ZMIN) RETURN
      ZMIN=Z
      JMIN=JJ
      RETURN
      END
```

```
      SUBROUTINE SORTHS     (MAXEFT,ZMAX,MINEFT,ZMIN,J1,J2,J3MAX,J3MIN,
     1 LMAX,LMIN)
      REAL*4 MAXEFT,MINEFT
      DIMENSION LMAX(15),LMIN(15)
C
C     COMPARE MOMENT AT CURRENT POINT FROM H-LOADING OR MAX. AND MIN.
C     MOMENT AT CURRENT POINT FROM HS-LOADING TO PREVIOUS
C     MAX.AND.MIN.VALUES
C
  100 IF(ZMAX.LE.MAXEFT) GO TO 110
      MAXEFT=ZMAX
      LMAX(1)=J1
      LMAX(2)=J2
      LMAX(3)=J3MAX
  110 IF(ZMIN.GE.MINEFT) RETURN
      MINEFT=ZMIN
      LMIN(1)=J1
      LMIN(2)=J2
      LMIN(3)=J3MIN
      RETURN
      END
```

```
      SUBROUTINE OUTIL (J1)
      REAL*4 INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMOM,MINSHR,L
     1,LLVIN
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,1 ),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCOMP(100)
      COMMON/BLK2/LMMIN(15),LMMAX(15),LVMAX(15),NWHL(14),NODDSN(11,10),P
     1DSCMP(100),LODKOD(7),LEXTRM(30),LEXTRV(30),LVMIN(15),LCDCON(11,10,
     24)
      COMMON/BLK3/MINMOM,MAXMOM,MAXSHR,MINSHR,SCLLNE,SCLCOM,SCLHHS,SCLCO
     1V,UNIFLL,BMWT,S,THK
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCCUNT
      COMMON/BLK5/ITITLE(30),JTITLE(80),KTITLE(80),JBMDSG(2),KAASHO(2),K
     1AXTR,KOUTPT,KCCNT,KODE
      DIMENSION ZIP(5)
      DATA ZIP(1)/4H H  /,ZIP(2)/4H HS /,ZIP(3)/4HAXTR/,ZIP(4)/4HLANE/
     1 , ZIP(5)/4HUNIF/
      IF(KOUTPT.EQ.0) RETURN
      IF(J1.NE.0) GO TO 20
      IF(KOUTPT.EQ.2) GO TO 160
      IF(JPNT.EQ.1.OR.JPNT.EQ.3.OR.JPNT.EQ.5.OR.JPNT.EQ.7.OR.JPNT.EQ.9.O
     1R.JPNT.EQ.11) WRITE(6,161)
  161 FORMAT(1H1)
  160 CONTINUE
      JJ=JPNT-1
      WRITE (6,10) JSPAN,JJ
      WRITE(6,11)
   10 FORMAT(/////,1X,54(1H*),'DESIGN POINT',1X,I2,'.',I2,58(1H*))
   11 FORMAT(1X,'*',3X,'*',7X,'*',' MAGNITUDE  *',15X,89(1H*))
      WRITE(6,12)
   12 FORMAT(1X,'*',3X,'*',7X,'*',' (KIP-FEET) *',15X,'*WHEEL POSTION (D
     1ISTANCE FROM LEFT END OF BRIDGE)',19(1H*))
      WRITE(6,13)
   13 FORMAT(1X,'LOAD* FORCE *      OR',5X,105(1H*))
      WRITE(6,14)
   14 FORMAT(1X,'TYPE* TYPE  *   (KIPS)    *')
      WRITE(6,15)
   15 FORMAT(1X,131(1H*))
      RETURN
   20 GO TO (100,200,300,400,500),J1
  100 IF (LODKOD(1).EQ.0) GO TO 150
      WRITE(6,120)
      WRITE(6,121)ZIP(1),MAXMOM,( LMMAX(J2),J2=1,2)
      WRITE(6,122)ZIP(1),MINMOM,( LMMIN(J2),J2=1,2)
      WRITE(6,123)ZIP(1),MAXSHR,( LVMAX(J2),J2=1,2)
      WRITE(6,124)ZIP(1),MINSHR,( LVMIN(J2),J2=1,2)
  120 FORMAT(26X,'*WHL 1*WHL 2*')
  121 FORMAT(1X,A4,'*MAX.MOM*',1PE12.5,'*',2(I5,'*'))
```

```
  122 FORMAT(1X,A4,'*MIN.MOM*',1PE12.5,'*', 2(I5,'*'))
  123 FORMAT(1X,A4,'*MAX.SHR*',1PE12.5,'*', 2(I5,'*'))
  124 FORMAT(1X,A4,'*MIN.SHR*',1PE12.5,'*', 2(I5,'*'))
      RETURN
  150 WRITE(6,125)
      WRITE(6,126)ZIP(2),MAXMOM,( LMMAX(J2),J2=1,3)
      WRITE(6,127)ZIP(2),MINMOM,( LMMIN(J2),J2=1,3)
      WRITE(6,128)ZIP(2),MAXSHR,( LVMAX(J2),J2=1,3)
      WRITE(6,129)ZIP(2),MINSHR,( LVMIN(J2),J2=1,3)
  125 FORMAT(26X,'*WHL 1*WHL 2*WHL 3*')
  126 FORMAT(1X,A4,'*MAX.MOM*',1PE12.5,'*',3(I5,'*'))
  127 FORMAT(1X,A4,'*MIN.MOM*',1PE12.5,'*', 3(I5,'*'))
  128 FORMAT(1X,A4,'*MAX.SHR*',1PE12.5,'*', 3(I5,'*'))
  129 FORMAT(1X,A4,'*MIN.SHR*',1PE12.5,'*', 3(I5,'*'))
      RETURN
  200 WRITE(6,130)
      WRITE(6,131)ZIP(3),MAXMOM,( LMMAX(J2),J2=1,NWHEEL)
      WRITE(6,132)ZIP(3),MINMOM,( LMMIN(J2),J2=1,NWHEEL)
      WRITE(6,133)ZIP(3),MAXSHR,( LVMAX(J2),J2=1,NWHEEL)
      WRITE(6,134)ZIP(3),MINSHR,( LVMIN(J2),J2=1,NWHEEL)
  130 FORMAT(26X,'*WHL 1*WHL 2*WHL 3*WHL 4*WHL 5*WHL 6*WHL 7*WHL 8*WHL 9
     1*WHL10*WHL11*WHL12*WHL13*WHL14*WHL15*')
  131 FORMAT(1X,A4,'*MAX.MOM*',1PE12.5,'*',15(I5,'*'))
  132 FORMAT(1X,A4,'*MIN.MOM*',1PE12.5,'*',15(I5,'*'))
  133 FORMAT(1X,A4,'*MAX.SHR*',1PE12.5,'*',15(I5,'*'))
  134 FORMAT(1X,A4,'*MIN.SHR*',1PE12.5,'*',15(I5,'*'))
      RETURN
  300 WRITE(6,301)ZIP(4),MAXMOM
      WRITE(6,302)ZIP(4),MINMOM
      WRITE(6,303)ZIP(4),MAXSHR
      WRITE(6,304)ZIP(4),MINSHR
  301 FORMAT(1X,A4,'*MAX.MOM*',1PE12.5,'*')
  302 FORMAT(1X,A4,'*MIN.MOM*',1PE12.5,'*')
  303 FORMAT(1X,A4,'*MAX.SHR*',1PE12.5,'*')
  304 FORMAT(1X,A4,'*MIN.SHR*',1PE12.5,'*')
      RETURN
  400 WRITE(6,401)ZIP(5),MAXMOM
      WRITE(6,402)ZIP(5),MAXSHR
  401 FORMAT(1X,A4,'*MOMENT *',1PE12.5,'*')
  402 FORMAT(1X,A4,'*SHEAR  *',1PE12.5,'*')
      RETURN
  500 WRITE(6,501)DLMOM(JPNT,JSPAN)
  501 FORMAT(2X,'DL * MOMENT*',1PE12.5,'*')
      WRITE(6,502)DLSHR(JPNT,JSPAN)
  502 FORMAT(2X,'DL * SHEAR *',1PE12.5,'*')
      IF(KOUTPT.EQ.1) GO TO 699
      WRITE(6,600)
  600 FORMAT(40X,'INFLUENCE LINE VALUES')
```

```
      WRITE(6,620)
  620 FORMAT(//)
      JJ=JPNT-1
      WRITE(6,640) JSPAN,JJ
  640 FORMAT(40X,'DESIGN POINT',I2,'.',I2)
      WRITE(6,620)
      WRITE(6,621)
  621 FORMAT(10X,'DISTANCE FROM LEFT',10X,'SHEAR',15X,'MOMENT',/,10X,' E
     1ND OF BRIDGE (FT)',9X,'(KIPS)',14X,'(KIP-FT)',//)
      WRITE(6,623) INFLV(200),INFLM(200)
  623 FORMAT(20X,'0(SUPPORT)',1PE14.7,11X,1PE14.7)
      DO 622 J=1,N
      JM1=J-1
      LWRIT2=SL(J)+200
      IF(JM1.EQ.0) GO TO 628
      LWRIT1=1+SL(JM1)+200
      GO TO 625
  628 LWRIT1=201
  625 DO 622 I=LWRIT1,LWRIT2
      IF(I.EQ.LWRIT2) GO TO 624
      GO TO 629
  624 IW=I-200
      WRITE(6,626)IW, INFLV(I),INFLM(I)
      GO TO 622
  629 IW=I-200
      WRITE(6,627)IW,INFLV(I),INFLM(I)
  622 CONTINUE
  626 FORMAT(18X,I3,'(SUPPORT)',1PE14.7,11X,1PE14.7)
  627 FORMAT(18X,I3,9X,1PE14.7,11X,1PE14.7)
  699 IF(KODE.EQ.10) WRITE(6,630)
  630 FORMAT(//,30X,'BRIDGE IS SYMMETRICAL ABOUT THIS POINT')
      RETURN
      END
```

```
      SUBROUTINE OUTKO (J1)
C
C     THIS SUBROUTINE IS FOR WRITING OUT THE O OUTPUT LEVEL
C
      REAL*4 INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMOM,MINSHR,L
     1,LLVIN
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,10),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCOMP(100)
      COMMON/BLK2/LMMIN(15),LMMAX(15),LVMAX(15),NWHL(14),NODDSN(11,10),P
     1OSCMP(100),LODKOD(7),LEXTRM(30),LEXTRV(30),LVMIN(15),LODCON(11,10,
     24)
      COMMON/BLK3/MINMOM,MAXMOM,MAXSHR,MINSHR,SCLLNE,SCLCOM,SCLHHS,SCLCO
     1V,UNIFLL,BMWT,S,THK
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCOUNT
      COMMON/BLK5/ITITLE(80),JTITLE(80),KTITLE(80),JBMDSG(2),KAASHO(2),K
     1AXTR,KOUTPT,KCONT,KODE
      REAL MOMT,MOMTM
      INTEGER TRUCK,AXLE,UNIF,ZERO
   30 IF(J1.NE.0) GO TO 31
      WRITE(6,706)
  706 FORMAT(1H1)
      WRITE(6,700)
  700 FORMAT(49X,'SUMMARY OF CALCULATIONS',//)
      WRITE(6,701)
      WRITE(6,705)
      WRITE(6,702)
      WRITE(6,703)
      WRITE(6,704)
      WRITE(6,705)
  705 FORMAT(/)
  701 FORMAT('          *          *LIVE LOAD EFFECTS (WITH IMPACT)          **
     *  DEAD LOAD EFFECTS      **     TOTAL    EFFECTS                 **')
  702 FORMAT('  DESIGN* MAXIMUM     * MINIMUM    * MAXIMUM    * MINIMUM     **
     $          *            **   MAX(+)    *   MAX(-)      **              **')
  703 FORMAT('  POINT * MOMENT      *  MOMENT    *   SHEAR    *   SHEAR     **
     $  MOMENT    *    SHEAR   **   MOMENT    *   MOMENT      **     SHEAR    **')
  704 FORMAT('      .     * (KIP-FT)   *  (KIP-FT) *    (KIPS)   *   (KIPS)    **
     $  (KIP-FT) *     (KIPS)  **  (KIP-FT) *   (KIP-FT) **      (KIPS)   **')
      RETURN
   31 CONTINUE
      DATA TRUCK/1HT/,AXLE/1HA/,LANE/1HL/,UNIF/1HU/,ZERO/1H /
C       DETERMINE CONTROLLING LOAD TYPE
      IF(LODCON(JPNT,JSPAN,1).EQ.0) INDI1=ZERO
      IF(LODCON(JPNT,JSPAN,2).EQ.0) INDI2=ZERO
      IF(LODCON(JPNT,JSPAN,3).EQ.0) INDI3=ZERO
      IF(LODCON(JPNT,JSPAN,4).EQ.0) INDI4=ZERO
      IF(LODCON(JPNT,JSPAN,1).EQ.1) INDI1=TRUCK
```

```
      IF(LODCON(JPNT,JSPAN,1).EQ.2) INDI1=AXLE
      IF(LODCON(JPNT,JSPAN,1).EQ.3) INDI1=LANE
      IF(LODCON(JPNT,JSPAN,1).EQ.4) INDI1=UNIF
      IF(LODCON(JPNT,JSPAN,2).EQ.1) INDI2=TRUCK
      IF(LODCON(JPNT,JSPAN,2).EQ.2) INDI2=AXLE
      IF(LODCON(JPNT,JSPAN,2).EQ.3) INDI2=LANE
      IF(LODCON(JPNT,JSPAN,2).EQ.4) INDI2=UNIF
      IF(LODCON(JPNT,JSPAN,3).EQ.1) INDI3=TRUCK
      IF(LODCON(JPNT,JSPAN,3).EQ.2) INDI3=AXLE
      IF(LODCON(JPNT,JSPAN,3).EQ.3) INDI3=LANE
      IF(LODCON(JPNT,JSPAN,3).EQ.4) INDI3=UNIF
      IF(LODCON(JPNT,JSPAN,4).EQ.1) INDI4=TRUCK
      IF(LODCON(JPNT,JSPAN,4).EQ.2) INDI4=AXLE
      IF(LODCON(JPNT,JSPAN,4).EQ.3) INDI4=LANE
      IF(LODCON(JPNT,JSPAN,4).EQ.4) INDI4=UNIF
      MOMTM=LLMIN(JPNT,JSPAN)+DLMOM(JPNT,JSPAN)
      MOMT=LLMAX(JPNT,JSPAN)+DLMOM(JPNT,JSPAN)
      SHEART=ABS(LLVAX(JPNT,JSPAN)+DLSHR(JPNT,JSPAN))
      SHEAT=ABS(LLVIN(JPNT,JSPAN)+DLSHR(JPNT,JSPAN))
      SHEART=AMAX1(SHEART,SHEAT)
      JJ=JPNT-1
      IF(LLMAX(JPNT,JSPAN).EQ.-10000)  LLMAX(JPNT,JSPAN)=0
      IF(LLMIN(JPNT,JSPAN).EQ.10000) LLMIN(JPNT,JSPAN)=0
      IF(LLVAX(JPNT,JSPAN).EQ.-10000) LLVAX(JPNT,JSPAN)=0
      IF(LLVIN(JPNT,JSPAN).EQ.10000) LLVIN(JPNT,JSPAN)=0
      IF(JPNT.EQ.1) WRITE(6,1000)
 1000 FORMAT(/)
      WRITE(6,32)JSPAN,JJ   ,LLMAX(JPNT,JSPAN),INDI1,LLMIN(JPNT,JSPAN),IN
     1DI2,LLVAX(JPNT,JSPAN),INDI3,LLVIN(JPNT,JSPAN),INDI4,DLMOM(JPNT,JSP
     2AN),DLSHR(JPNT,JSPAN),MOMT,MOMTM,SHEART
   32 FORMAT(2X,I2,'.',I3,'*',4(1PE10.3,A1,'*'),'*',2(1PE10.3,1X,'*'),'*
     $',1PE10.3,1X,'*',1PE10.3,1X,'**',1PE10.3,1X,'**')
      IF(KODE.EQ.10) WRITE(6,720)
  720 FORMAT(//,30X,'BRIDGE IS SYMMETRICAL ABOUT THIS POINT')
      RETURN
      END
```

```
      SUBROUTINE REACTN
      REAL*4 INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMCM,MINSHR,L
     1LVIN
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,10),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCCMP(100)
      COMMON/BLK2/LMMIN(15),LMMAX(15),LVMAX(15),NWHL(14),NODDSN(11,10),P
     1OSCMP(100),LODKOD(7),LEXTRM(30),LEXTRV(30),LVMIN(15),LODCON(11,10,
     224)
      COMMON/BLK3/MINMCM,MAXMOM,MAXSHR,MINSHR,SCLLNE,SCLCOM,SCLHHS,SCLCO
     1V,UNIFLL,BMWT,S,THK
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCOUNT
C
C              THIS SUBROUTINE COMPUTES INFLUENCE LINE ORDINATES FOR
C              REACTION FORCES
C
      NM1=N-1
C
C              COMPUTE ALPHA COEFFICIENTS
C
      DO 24 K=1,N
      DO 24 J=1,N
      SLKM1=0.
      SLJM1=0.
      IF(K.GT.1) SLKM1=SL(K-1)
      IF(J.GT.1) SLJM1=SL(J-1)
   24 ALPHA(K,J)=(SL(N)**3-SLKM1**3)/3.-(SLJM1+SLKM1)*(SL(N)**2-SLKM1**2
     *)/2.+SLJM1*SLKM1*(SL(N)-SLKM1)+L(N)*(SL(N)-SLJM1)*(SL(N)-SLKM1)
C
C              COMPUTE COEFFICIENT MATRIX A
C
      DO 26 K=1,NM1
      SLKM1=0.
      IF(K.GT.1) SLKM1=SL(K-1)
      DO 26 J=1,K
      SLJM1=0.
      IF(J.GT.1) SLJM1=SL(J-1)
      A(K,J)=(SL(N)-SLKM1)*(SL(N)-SLJM1)*ALPHA(N,N)/L(N)**2-(SL(N)-
     *SLKM1)*ALPHA(N,J)/L(N)-(SL(N)-SLJM1)*ALPHA(N,K)/L(N)
      A(K,J)=A(K,J)+ALPHA(K,J)
   26 A(J,K)=A(K,J)
C
C              FORM RIGHT HAND SIDE VECTORS
C
      NPNTS=0
      Z=0.
      DO 40 J=1,20000
      NPNTS=NPNTS+1
```

```
      DO 32 K=1,NM1
      SLKM1=0.
      IF(K.GT.1) SLKM1=SL(K-1)
      IF(Z.LE.SLKM1) T=SLKM1
      IF(Z.GT.SLKM1) T=Z
      IF(Z.LE.SL(NM1)) Q=SL(NM1)
      IF(Z.GT.SL(NM1)) Q=Z
   32 REACT(K,J)=(SL(N)**3-T**3)/3.-(Z+SLKM1)*(SL(N)**2-T**2)/2.+
     * Z*SLKM1*(SL(N)-T)-(SL(N)-Z)*(ALPHA(N,K)-(SL(N)-SLKM1)*ALPHA(N,N)
     */L(N))/L(N)-(SL(N)-SLKM1)*((SL(N)**3-Q**3)/3.-(Z+SL(NM1))*(SL(N)**
     *  2-Q**2)/2.+Z*SL(NM1)*(SL(N)-Q))/L(N)
      Z=Z+1.
      IF(Z.GT.SL(N)) GO TO 42
   40 CONTINUE
   42 CONTINUE
      CALL MATINV(A,NM1,REACT,NPNTS,DET)
      Z=0.
      DO 55 J=1,NPNTS
      SUM1=0.
      SUM=0.
      DO 52 K=1,NM1
      SLKM1=0.
      IF(K.GT.1) SLKM1=SL(K-1)
   52 SUM=SUM+REACT(K,J)*(SL(N)-SLKM1)
      REACT(N,J)=(SL(N)-Z-SUM)/L(N)
   55 Z=Z+1.
      RETURN
      END
```

```
      SUBROUTINE MATINV(A,N,B,M,DETERM)
      DIMENSION IPIVOT(10),A(10,10),B(10,1000),INDEX(10,2),PIVOT(10)
      EQUIVALENCE(IROW,JROW),(ICOLUM,JCOLUM),(AMAX,T,SWAP)
      DETERM=1.0
      DO 20 J=1,N
   20 IPIVOT(J) =0
      DO 550 I=1,N
      AMAX=0.0
      DO 105 J=1,N
      IF (IPIVOT(J) -1) 60,105,60
   60 DO 100 K=1,N
      IF(IPIVOT(K)-1) 80,100,740
   80 IF( ABS(AMAX)- ABS(A(J,K))) 85,85,100
   85 IROW=J
      ICOLUM = K
      AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
      IF(IROW-ICOLUM) 140,260,140
  140 DETERM=-DETERM
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(IC LUM,L)
  200 A(ICOLUM,L)=SWAP
      IF(M) 260,260,210
  210 DO 250 L=1,M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT(I)
      IF(PIVOT(I)) 330,720, 330
  330 A(ICOLUM,ICOLUM)=1.0
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
      IF(M) 380,380,360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
  380 DO 550 L1=1,N
      IF(L1-ICOLUM) 400,550, 400
  400 T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF(M) 550, 550, 460
```

```
  460 DO 500  L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
      DO 710 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
      JCOLUM = INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
      RETURN
  720 WRITE(6,730)
  730 FORMAT(20H MATRIX IS SINGULAR        )
  740 RETURN
      END
```

```
      SUBROUTINE INFLNE
      REAL*4 INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMOM,MINSHR,L
     1,LLVIN
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,10),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCCMP(100)
      COMMON/BLK2/LMMIN(15),LMMAX(15),LVMAX(15),NWHL(14),NODDSN(11,10),P
     1CSCMP(100),LODKOD(7),LEXTRM(30),LEXTRV(30),LVMIN(15),LODCON(11,10,
     24)
      COMMON/BLK3/MINMOM,MAXMOM,MAXSHR,MINSHR,SCLLNE,SCLCOM,SCLHHS,SCLCO
     1V,UNIFLL,BMWT,S,THK
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCOUNT
      NODDES=NODDSN(JPNT,JSPAN)
      J2=NODDES-200
      Z=FLOAT(J2)
      DO 22 J1=1,NPNTS
C
C      COMPUTE INFLM(J1)
C
      ZM=REACT(1,J1)*Z
      IF(JSPAN.EQ.1) GO TO 12
      DO 10 J2=2,JSPAN
   10 ZM=ZM+REACT(J2,J1)*(Z-SL(J2-1))
   12 J3=J1-1
      XARM=Z-FLOAT(J3)
      IF(XARM.LE.0.) GO TO 14
      ZM=ZM-XARM
C
C      COMPUTE INFLV(J1)
C
C
C     SUM CONTRIBUTIONS FROM REACTIONS
C
   14 XV=0.
      DO 16 J2=1,JSPAN
   16 XV=XV+REACT(J2,J1)
C
C      ADD CONTRIBUTION FROM UNIT LOAD
C
      JNODE=199+J1
      IF(JNODE.LT.NODDES) XV=XV-1.
      INFLM(J1+199) = ZM
   22 INFLV(199+J1)=XV
C
C      SET INFLM(I) AND INFLV(I) VALUES TO ZERO OVER SUPPORTS
C
      DO 30 J3=1,N
      JLFT=NODDSN(1,J3)
```

```
      INFLM(JLFT)=0.
   30 INFLV(JLFT)=0.
      JRGT=NODDSN(11,N)
      INFLM(JRGT)=0.
      INFLV(JRGT)=0.
      IF(JPNT.EQ.1) INFLV(NODDSN(1,JSPAN))=1.
      IF(JPNT.EQ.11) INFLV(NODDSN(11,JSPAN))=0.
      IF(JPNT.EQ.11.AND.JSPAN.EQ.N) GO TO 32
      GO TO 36
   32 DO 34 J3=1,NPNTS
   34 INFLM(199+J3)=0.
   36 CONTINUE
C
C       SET INFLV(I) AND INFLM(I) TO ZERO FOR ALL NODES TO LEFT OF FIRST
C       SUPPORT AND TO RIGHT OF LAST SUPPORT
C
      DO 7 I=1,199
      INFLM(I)=0.
    7 INFLV(I)=0.
      J=NPNTS+200
      K=J+200
      DO 8 I=J,K
      INFLM(I)=0.
    8 INFLV(I)=0.
      J=NPNTS+199
      RETURN
      END
```

```
      SUBROUTINE IMPACT (ITEST,RIMP)
      REAL*4 INFLM,INFLV,LLMAX,LLMIN,LLVAX,MAXSHR,MAXMOM,MINMOM,MINSHR,L
      COMMON/BLK1/L(10),SL(10),A(10,10),REACT(10,1000),ALPHA(10,10),INFL
     1M(1400),INFLV(1400),LLMAX(11,10),LLMIN(11,10),LLVAX(11,10),LLVIN(1
     21,10),PWHEEL(15),DLMOM(11,10),DLSHR(11,10),FCOMP(100)
      COMMON/BLK4/NWHEEL,N,NPNTS,JSPAN,JPNT,NEXTRM,NEXTRV,NCFCS,NCOUNT
      GO TO (100,200,300),ITEST
  100 RL=L(JSPAN)
      GO TO 400
  200 IF(LLMIN(JPNT,JSPAN).GT.0) GO TO 100
      IF(JSPAN.EQ.1) RL=(L(1)+L(2))/2.
      IF(JSPAN.EQ.1) GO TO 400
      IF(JSPAN.EQ.N) RL=(L(N)+L(N-1))/2.
      IF(JSPAN.EQ.N) GO TO 400
      IF(JPNT.GT.6) RL=(L(JSPAN)+L(JSPAN+1))/2.
      IF(JPNT.LE.6) RL =(L(JSPAN)+L(JSPAN-1))/2.
      GO TO 400
  300 IF(JPNT.GT.6) RL=(JPNT-1)*L(JSPAN)*.1
      IF(JPNT.LE.6) RL=L(JSPAN)-(JPNT-1)*L(JSPAN)*.1
  400 RI=AMIN1(.30,50./(RL+125.))
      RIMP=1.+RI
      RETURN
      END
```