Stresses and Displacements in Three-Dimensional Trusses

ROBERT L. KUSHMAN AND ELIHU GEER

A METHOD IS already available[1] for the direct solution for stresses and displacements in three-dimensional trusses. The method merits consideration by structural designers. No analogies are used. A checkerboard system of trusses is not necessary. Indeterminacy does not add in the least to the difficulty of solution, and in fact does not even have to be known. The end product is maximum stress in each member for any combination of possible loads. Moreover, the method works even better for planar trusses.

The principal disadvantage of this method is the larger number of unknowns—three for each joint which can displace. However, this feature is no handicap to the digital computer. It merely highlights the need to use the computer. A fairly sophisticated computer can solve a hundred simultaneous equations in three minutes. A modern structural design office cannot afford not to have access to a computer facility.

NOMENCLATURE

L = length of a member
x, y, z = coordinates of a joint
t = subscript indicating "this" end of member
o = subscript indicating "other" end of member
u, v, w = components of displacement of a joint in x, y and z directions respectively
P_x, P_y, P_z = x, y and z components of external force at a joint
F = axial force (commonly called stress) in a member
F_x = $(x_t - x_o)F/L$ = x-component of F
F_y = $(y_t - y_o)F/L$ = y-component of F
F_z = $(z_t - z_o)F/L$ = z-component of F
X = $x_t - x_o$
Y = $y_t - y_o$
Z = $z_t - z_o$

Joints are numbered 1, 2, 3, etc. These numbers are used as subscripts.

METHOD OF SOLUTION

$$L^2 = X^2 + Y^2 + Z^2$$

$$2L\Delta L = 2X\Delta X + 2Y\Delta Y + 2Z\Delta Z$$

$$\Delta L = FL/AE$$

$$\Delta X = u_t - u_o$$

$$\Delta Y = v_t - v_o$$

$$\Delta Z = w_t - w_o$$

Making these substitutions,

$$F = \frac{AE}{L^2}[X(u_t - u_o) + Y(v_t - v_o) + Z(w_t - w_o)]$$

$$F_x = \frac{AE}{L^3}[X^2(u_t - u_o) + XY(v_t - v_o) + XZ(w_t - w_o)]$$

$$F_y = \frac{AE}{L^3}[XY(u_t - u_o) + Y^2(v_t - v_o) + YZ(w_t - w_o)]$$

$$F_z = \frac{AE}{L^3}[XZ(u_t - u_o) + YZ(v_t - v_o) + Z^2(w_t - w_o)]$$

At every joint,

$$\sum F_{x_t} = P_{x_t}$$

$$\sum F_{y_t} = P_{y_t}$$

$$\sum F_{z_t} = P_{z_t}$$

For every joint which can displace, write F_x, F_y and F_z for every member which comes into that joint. The terms $\sum F_{x_t}$, $\sum F_{y_t}$ and $\sum F_{z_t}$ then consist of the variables u, v and w with coefficients which are combinations of AE/L^3 and X, Y and Z. There are three equations for each joint, matching the three unknowns.

It is desirable to have the computer set up the equations, as there is a considerable amount of arithmetic. For this purpose, the equations take the form:

Robert L. Kushman is Instructor in Civil Engineering, University of Detroit, Detroit, Mich.
Elihu Geer is Professor of Civil Engineering, University of Detroit, Detroit, Mich.

---

[1] *Maugh, L. C.* Statically Indeterminate Structures, *Second Edition, 1964, New York, John Wiley & Sons, pp. 405–411.*

$$u_t\sum\frac{AE}{L^3}X^2 + v_t\sum\frac{AE}{L^3}XY + w_t\sum\frac{AE}{L^3}XZ - \sum u_o\frac{AE}{L^3}X^2 - \sum v_o\frac{AE}{L^3}XY - \sum w_o\frac{AE}{L^3}XZ = P_x$$

$$u_t\sum\frac{AE}{L^3}XY + v_t\sum\frac{AE}{L^3}Y^2 + w_t\sum\frac{AE}{L^3}YZ - \sum u_o\frac{AE}{L^3}XY - \sum v_o\frac{AE}{L^3}Y^2 - \sum w_o\frac{AE}{L^3}YZ = P_y$$

$$u_t\sum\frac{AE}{L^3}XZ + v_t\sum\frac{AE}{L^3}YZ + w_t\sum\frac{AE}{L^3}Z^2 - \sum u_o\frac{AE}{L^3}XZ - \sum v_o\frac{AE}{L^3}YZ - \sum w_o\frac{AE}{L^3}Z^2 = P_{z_t}$$

The coefficients of the u_t, v_t and w_t terms are sums of the indicated properties of all the members which meet at "this" joint, whether or not the other end can displace. The u_o, v_o and w_o terms each consist of a series of subterms, one for each "other" joint which can displace and which is connected to "this" joint by a member.

Having prepared a set of equations, solve for the displacements and stresses caused by several combinations of loads and thus get the maximum values. To show how to avoid repeated solution of the many simultaneous equations in arriving at maximum values, it will be helpful to take a side trip into matrix algebra, starting with elementary methods of solving simultaneous equations.

Consider the equations:

$$\begin{array}{ll}
(1) & 2x + 3y = 9 \\
(2) & 7x - 4y = 46 \\
4(1) & 8x + 12y = 36 \\
3(2) & \underline{21x - 12y = 138} \\
 & 29x \quad = 174, \quad x = +6 \\
7(1) & 14x + 21y = 63 \\
2(2) & \underline{14x - 8y = 92} \\
 & 29y = -29, \quad y = -1
\end{array}$$

More generally,

$$\begin{array}{ll}
(1) & a_{11}x + a_{12}y = c_1 \\
(2) & a_{21}x + a_{22}y = c_2 \\
a_{22}(1) & a_{11}a_{22}x + a_{12}a_{22}y = a_{22}c_1 \\
a_{12}(2) & \underline{a_{12}a_{21}x + a_{12}a_{22}y = a_{12}c_2} \\
 & (a_{11}a_{22} - a_{12}a_{21})x = a_{22}c_1 - a_{12}c_2
\end{array}$$

$$x = \frac{a_{22}c_1}{a_{11}a_{22} - a_{12}a_{21}} - \frac{a_{12}c_2}{a_{11}a_{22} - a_{12}a_{21}}$$

$$\begin{array}{ll}
a_{21}(1) & a_{11}a_{21}x + a_{12}a_{21}y = a_{21}c_1 \\
a_{11}(2) & \underline{a_{11}a_{21}x + a_{11}a_{22}y = a_{11}c_2} \\
 & (a_{12}a_{21} - a_{11}a_{22})y = a_{21}c_1 - a_{11}c_2
\end{array}$$

$$y = -\frac{a_{21}c_1}{a_{11}a_{22} - a_{12}a_{21}} + \frac{a_{11}c_2}{a_{11}a_{22} - a_{12}a_{21}}$$

Going back to the numerical example,

$$2x + 3y = 9$$

$$7x - 4y = 46$$

$$x = \frac{(-4)(9)}{(2)(-4) - (3)(7)} - \frac{(3)(46)}{(2)(-4) - (3)(7)} = \frac{4}{29}(9) + \frac{3}{29}(46) = +6$$

$$y = \frac{(7)(9)}{(2)(-4) - (3)(7)} + \frac{(2)(46)}{(2)(-4) - (3)(7)} = \frac{7}{29}(9) - \frac{2}{29}(46) = -1$$

But also for

$$2x + 3y = 19$$

$$7x - 4y = -6$$

$$x = \frac{4}{29}(19) + \frac{3}{29}(-6) = +2$$

$$y = \frac{7}{29}(19) - \frac{2}{29}(-6) = +5$$

and for any set of constants to the right of the equal signs,

$$x = \frac{4}{29}(c_1) + \frac{3}{29}(c_2)$$

$$y = \frac{7}{29}(c_1) - \frac{2}{29}(c_2)$$

This is the principle needed for solving a three-dimensional truss problem for several sets of loads. In effect Cramer's rule has been used. The digital computer and matrix algebra make it possible to do for many equations what Cramer's rule did for a few. The set of constants, 4/29, 3/29, 7/29 and −2/29 are formed into an array, or matrix.

$$\begin{bmatrix} 4/29 & 3/29 \\ 7/29 & -2/29 \end{bmatrix}$$

which is the "inverse" of the original array, or matrix,

$$\begin{bmatrix} 2 & 3 \\ 7 & -4 \end{bmatrix}$$

We used two "vectors" or "column matrices," consisting of the constants to the right of the equal signs,

$$\begin{Bmatrix} 9 \\ 46 \end{Bmatrix} \quad \text{and} \quad \begin{Bmatrix} 19 \\ -6 \end{Bmatrix}$$

Formally written in mathematical form,

$$[S]^{-1}\ \{P\} \ = \ \{d\}$$

Broken down into numerical operations for this example,

$$\begin{bmatrix} 4/29 & 3/29 \\ 7/29 & -2/29 \end{bmatrix} \begin{Bmatrix} 9 \\ 46 \end{Bmatrix} =$$

$$\begin{Bmatrix} (4/29)(9) + (3/29)(46) \\ (7/29)(9) - (2/29)(46) \end{Bmatrix} = \begin{Bmatrix} +6 \\ -1 \end{Bmatrix}$$

and

$$\begin{bmatrix} 4/29 & 3/29 \\ 7/29 & -2/29 \end{bmatrix} \begin{Bmatrix} 19 \\ -6 \end{Bmatrix} =$$

$$\begin{Bmatrix} (4/29)(19) + (3/29)(-6) \\ (7/29)(19) - (2/29)(-6) \end{Bmatrix} = \begin{Bmatrix} +2 \\ +5 \end{Bmatrix}$$

Truss problems have many equations with many unknowns. As the number of equations increases, the difficulty of inversion increases, both as to complexity and as to the volume of arithmetic required. There is a general expression for each term of the inverse matrix which would of itself require a lengthy explanation. Each term s'_{ij} has as its denominator the determinant of the stiffness matrix, and as its numerator the minor determinant of the transpose of the stiffness matrix "co-adjoint" to the corresponding element. Fortunately, the truss matrix under consideration is symmetric and its transpose is the same as the matrix. The "co-adjoint" array is found by eliminating the entire row and entire column of the element in question.

Doubtless every computer facility has an inversion tape. A program is included in this presentation.

Following inversion of the stiffness matrix, the inverse is multiplied by the successive vectors of external forces. Again, most computer centers have subroutines for multiplying, but a program for the operation is appended. The resulting vector is a list of the u, v and w displacements of the joints. It may be printed out if desired. If E has been omitted from the computations, the output at this point is E times the displacements. If the designer wants to know the actual amounts of the displacements, he must divide the output values by E. The resulting displacements are in the same unit (feet, inches etc.) as that of the coordinates of the joints in the input data.

Having the components of the displacements, we now tell the computer to calculate stresses, using the formula:

$$F = \frac{AE}{L^2}[(x_t - x_o)(u_t - u_o) + (y_t - y_o)(v_t - v_o) + (z_t - z_o)(w_t - w_o)]$$

keeping in mind that E is already in the recorded displacements. The designer may wish to have the stresses printed out for each set of loads. This is easily done.

Finally, the computer scans the stresses for each member due to the several sets of loads, and totals and prints the maximum tensile and compressive stresses.

SUMMARY

The designer does not need to know the details of programming or of the operation of the machine. He should be trained to prepare the input data. Thereafter the computer takes over. The designer should have a general idea of the steps which are being performed. They are listed below in order to bring them together.

1. Prepare input data in the prescribed FORMAT in the program.
2. The computer computes and records elements of the stiffness matrix.
3. The computer inverts the matrix.
4. The computer multiplies the inverted matrix successively by the vectors representing sets of components of external loads. This yields components of displacements.
5. The computer uses the equation:

   $$F = \frac{AE}{L^2}[(x_t - x_o)(u_t - u_o) + (y_t - y_o) \times (v_t - v_o) + (z_t - z_o)(w_t - w_o)]$$

   to compute stresses in each member for each set of loads.
6. The computer scans the stresses for each member

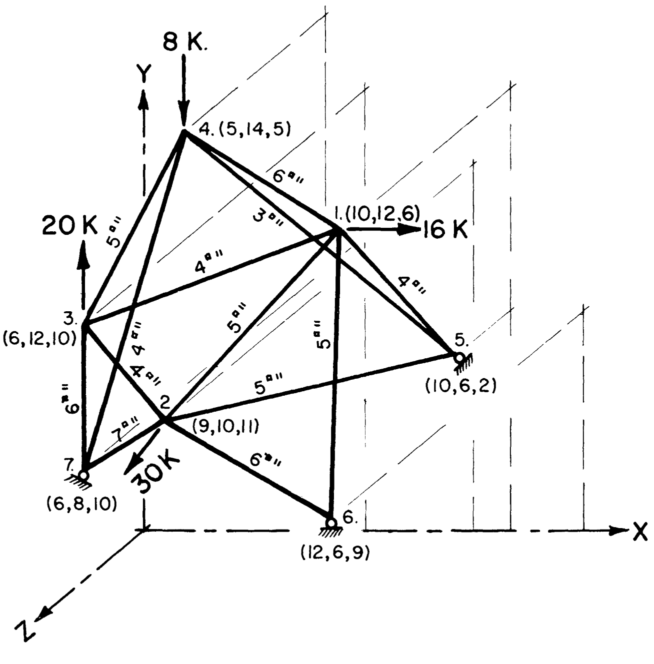
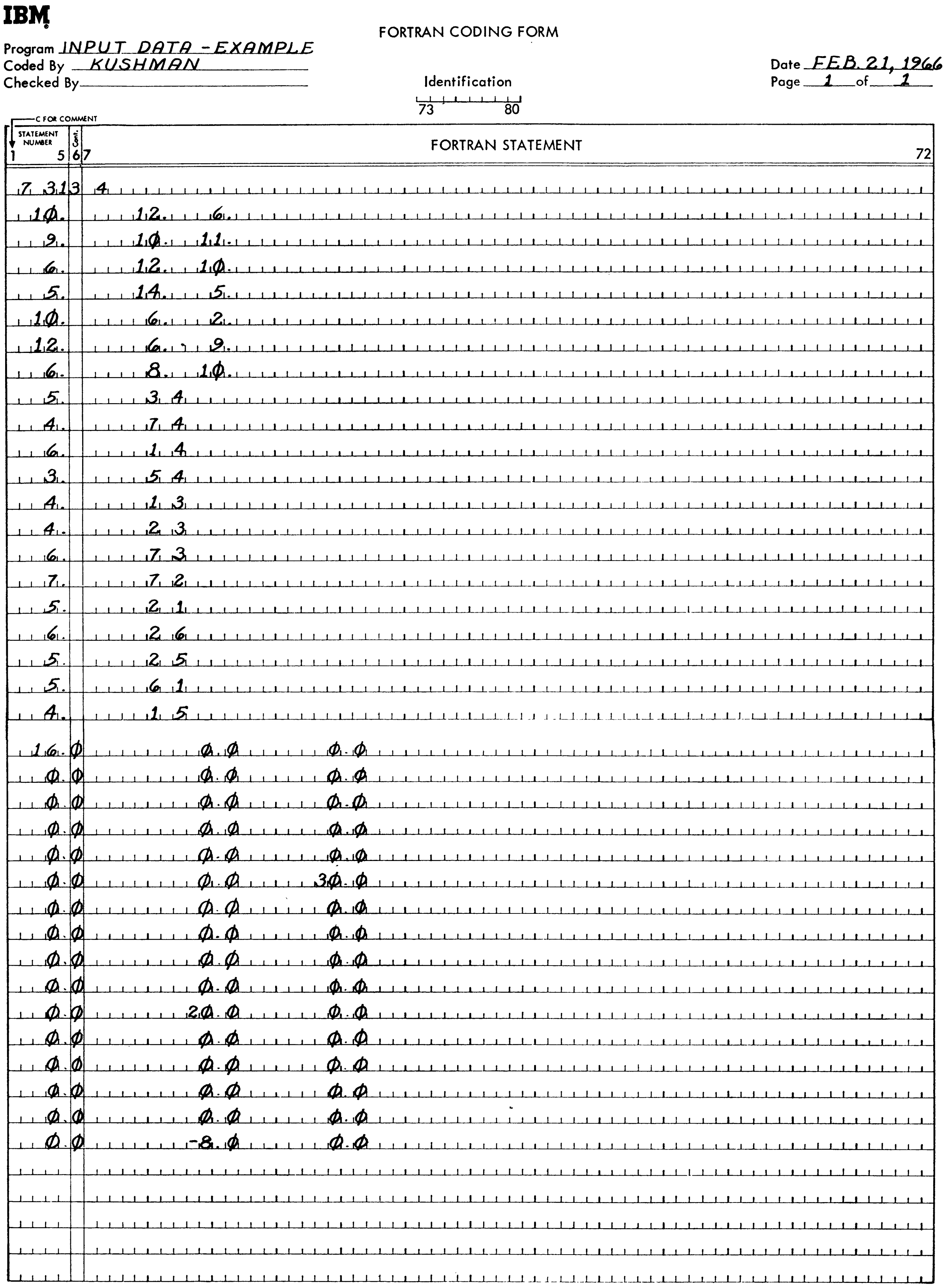


Figure 1



IBM

FORTRAN CODING FORM

Program INPUT DATA - EXAMPLE
Coded By KUSHMAN
Checked By

Date FEB. 21, 1966
Page 1 of 1

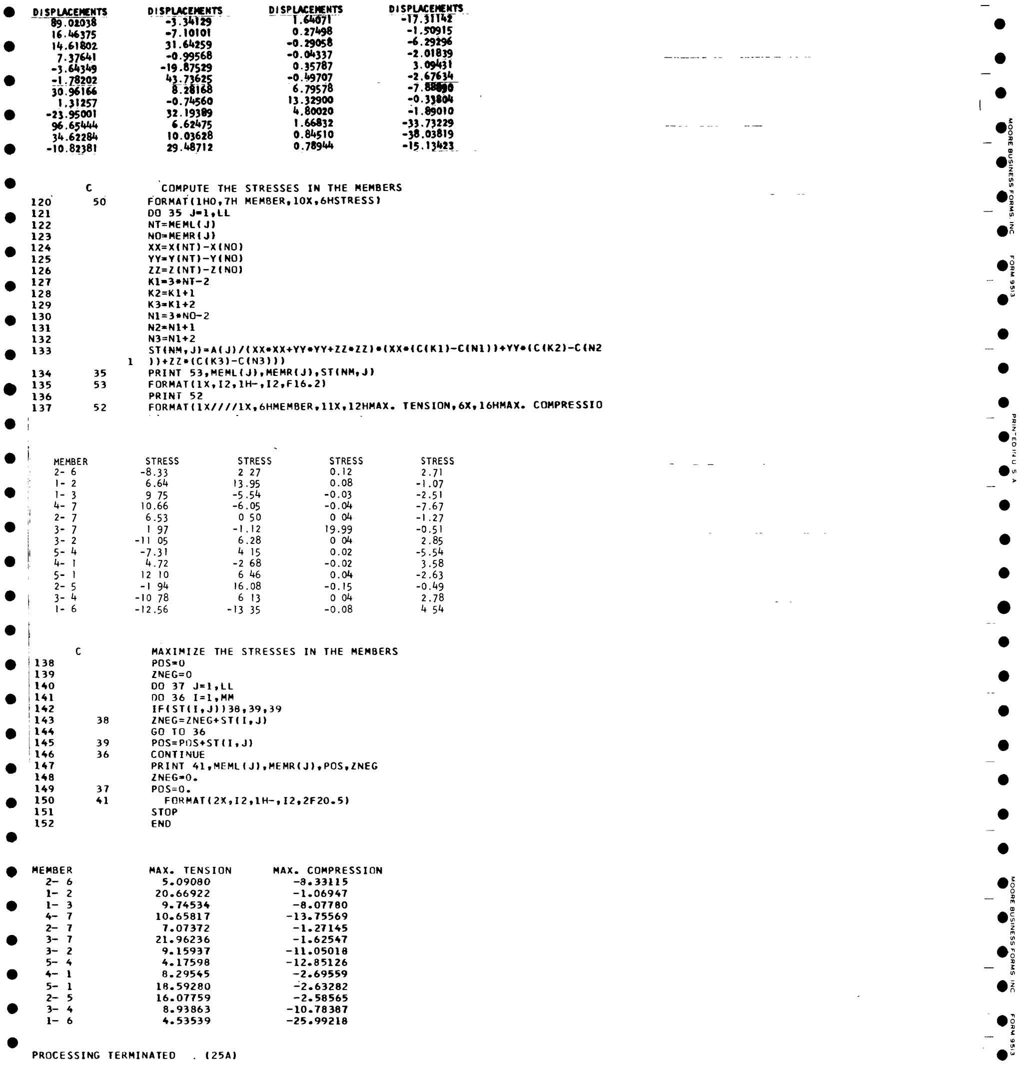
Identification 73 80

C FOR COMMENT | STATEMENT NUMBER 1 5 | Cont. 6 | 7 FORTRAN STATEMENT 72

```
 7 313 4
  10.     12.   6.
   9.     10. 11.
   6.     12. 10.
   5.     14.   5.
  10.      6.   2.
  12.      6.   9.
   6.      8. 10.
   5.      3 4
   4.      7 4
   6.      1 4
   3.      5 4
   4.      1 3
   4.      2 3
   6.      7 3
   7.      7 2
   5.      2 1
   6.      2 6
   5.      2 5
   5.      6 1
   4.      1 5
  16.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0      30.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0        20.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0         0.0       0.0
   0.0        -8.0       0.0
```

Figure 2

```
  VERSION  9/ 6/65          U N I V E R S I T Y   O F   D E T R O I T   F O R T R A N   S Y S T E M         2/28/66  - PAGE   1
  SEQ   STMNT       FORTRAN STATEMENT                                                    IDENT

        C****    *  FACULTY OR STUDENT NUMBER INVALID.
        C           SET UP THE STIFFNESS MATRIX
    1               DIMENSION X(7),Y(7),Z(7),A(13),MEML(13),MEMR(13),STIF(12,24),NO(12
                 1  ),C(21),P(12),ST(4,13)
    2               READ 20,NN,NF,LL,MM
    3       20      FORMAT(4I2)
    4               DO 2 I=1,NN
    5        2      READ3,X(I),Y(I),Z(I)
    6               N=NN-NF
    7               NK=3*N
    8               MAT=2*NK
    9               NP1=NK+1
   10               DO 8 I=1,NK
   11               DO 8 J=1,MAT
   12        8      STIF(I,J)=0
   13        3      FORMAT(3F7.0)
   14               DO 4 I=1,LL
   15        4      READ 5,A(I),MEML(I),MEMR(I)
   16        5      FORMAT(F10.0,2I2)
   17               DO 6 I=1,N
   18               K1=3*I-2
   19               K2=3*I-1
   20               K3=3*I
   21               SUM1=0
   22               SUM2=0
   23               SUM3=0
   24               SUM4=0
   25               SUM5=0
   26               SUM6=0
   27               DO 7 J=1,LL
   28               IF(MEML(J)-I)1,9,1
   29        1      IF(MEMR(J)-I) 7,11,7
   30        9      NT=MEML(J)
   31               NO=MEMR(J)
   32               GO TO 12
   33       11      NT=MEMR(J)
   34               NO=MEML(J)
   35       12      XX= X(NT)-X(NO)
   36               YY=Y(NT)-Y(NO)
   37               ZZ=Z(NT)-Z(NO)
   38               N3=3*NO
   39               N2=N3-1
   40               N1=N3-2
   41               ZL=SQRTF(XX*XX+YY*YY+ZZ*ZZ)
   42               RATIO=A(J)/(ZL**3)
   43               SUM=RATIO*XX*XX
   44               IF(N3-NK)13,13,14
   45       13      STIF(K1,N1)=-SUM
   46       14      SUM1=SUM1+SUM
   47               SUM=RATIO*YY*YY
   48               IF(N3-NK)15,15,16
   49       15      STIF(K2,N2)=-SUM
   50       16      SUM2=SUM2+SUM
   51               SUM=RATIO*ZZ*ZZ
   52               IF(N3-NK)17,17,18
   53       17      STIF(K3,N3)=-SUM
   54       18      SUM3=SUM3+SUM
   55               SUM=RATIO*XX*YY
   56               IF(N3-NK)19,19,21
   57       19      STIF(K1,N2)=-SUM
   58               STIF(K2,N1)=-SUM
   59       21      SUM4=SUM4+SUM
   60               SUM=RATIO*XX*ZZ
   61               IF(N3-NK)22,22,23
   62       22      STIF(K1,N3)=-SUM
   63               STIF(K3,N1)=-SUM
   64       23      SUM5=SUM5+SUM
   65               SUM=RATIO*YY*ZZ
   66               IF(N3-NK)24,24,25
   67       24      STIF(K2,N3)=-SUM
   68               STIF(K3,N2)=-SUM
   69       25      SUM6=SUM6+SUM
   70        7      CONTINUE
   71               STIF(K1,K1)=SUM1
   72               STIF(K2,K2)=SUM2
   73               STIF(K3,K3)=SUM3
   74               STIF(K1,K2)=SUM4
   75               STIF(K2,K1)=SUM4
   76               STIF(K1,K3)=SUM5
   77               STIF(K3,K1)=SUM5
   78               STIF(K2,K3)=SUM6
   79               STIF(K3,K2)=SUM6
   80        6      CONTINUE
   81               PRINT 2010
   82     2010      FORMAT(1H1,10X,16HSTIFFNESS MATRIX///)
   83               DO 26 I=1,NK
   84       26      PRINT 27,(STIF(I,J),J=1,NK)
   85       27      FORMAT(2X,12F10.5)

           STIFFNESS MATRIX


     1.35516  -0.47922  -0.23566  -0.03043  -0.06086   0.15215  -0.35355   0.00000   0.35355  -0.91287   0.36515  -0.18257
    -0.47922   1.17658  -0.38370  -0.06086  -0.12172   0.30429   0.00000   0.00000   0.00000   0.36515  -0.14606   0.07303
    -0.23566  -0.38370   1.45267   0.15215   0.30429  -0.76073   0.35355   0.00000  -0.35355  -0.18257   0.07303  -0.03651
    -0.03043  -0.06086   0.15215   2.27128  -0.07717   0.20092  -0.68724   0.45816  -0.22908   0.00000   0.00000   0.00000
    -0.06086  -0.12172   0.30429  -0.07717   1.65886   0.30315   0.45816  -0.30544   0.15272   0.00000   0.00000   0.00000
     0.15215   0.30429  -0.76073   0.20092   0.30315   1.54186  -0.22908   0.15272  -0.07636   0.00000   0.00000   0.00000
    -0.35355   0.00000   0.35355  -0.68724   0.45816  -0.22908   1.07123  -0.51902   0.02767  -0.03043   0.06086  -0.15215
     0.00000   0.00000   0.00000   0.45816  -0.30544   0.15272  -0.51902   1.92716  -0.45701   0.06086  -0.12172   0.30429
     0.35355   0.00000  -0.35355  -0.22908   0.15272  -0.07636   0.02767  -0.45701   1.19064  -0.15215   0.30429  -0.76073
    -0.91287   0.36515  -0.18257   0.00000   0.00000   0.00000  -0.03043   0.06086  -0.15215   1.02880  -0.59886   0.32930
     0.36515  -0.14606   0.07303   0.00000   0.00000   0.00000   0.06086  -0.12172   0.30429  -0.59886   0.76065  -0.54891
    -0.18257   0.07303  -0.03651   0.00000   0.00000   0.00000  -0.15215   0.30429  -0.76073   0.32930  -0.54891   1.02991

        C           INVERT THE STIFFNESS MATRIX
   86               DO 2014 I=1,NK
   87               J=NK+I
   88     2014      STIF(I,J)=1.
   89               DO 2060 I=1,NK
   90               IP1=I+1
   91               TEMP=1./STIF(I,I)
   92               DO 2070 J=I,MAT
   93     2070      STIF(I,J)=STIF(I,J)*TEMP
   94               DO 2060 J=1,NK
   95               IF(I-J)2059,2060,2059
   96     2059      TEMP=STIF(J,I)
   97               DO 2080 K=IP1,MAT
   98     2080      STIF(J,K)=STIF(J,K)-TEMP*STIF(I,K)
   99     2060      CONTINUE
  100               PRINT 2011
  101     2011      FORMAT(3X/////13X,15HINVERTED MATRIX///)
  102               DO 47 I=1,NK
  103       47      PRINT 27,(STIF(I,J),J=NP1,MAT)


              INVERTED MATRIX


     5.56377   1.02898   0.91363   0.46103  -0.22772  -0.11138   1.93510   0.08204  -1.49688   6.04090   2.16393  -0.67649
     1.02898   1.27437   0.28331   0.06901   0.11176  -0.23670   0.18024   0.01375  -0.36128   0.65196   0.18864  -0.25011
     0.91363   0.28331   1.97018  -0.04248  -0.67140   1.05475   0.24304  -0.01453   1.08385   1.35341   0.78662   1.03899
     0.46103   0.06901  -0.04248   0.66916  -0.13350  -0.03319   0.67719  -0.00217   0.14149   0.48143   0.25230   0.26105
    -0.22772   0.11176  -0.67140  -0.13350   1.09087  -0.66251  -0.62593   0.01789  -0.80391  -0.46127  -0.38679  -0.82231
    -0.11138  -0.23670   1.05475  -0.03319  -0.66251   1.45788   0.27606  -0.02485   1.07313   0.22082   0.33454   0.98290
     1.93510   0.18024   0.24304   0.67719  -0.62593   0.27606   2.50164   0.33979   0.19833   2.16530   0.98611   0.58777
     0.08204   0.01375  -0.01453  -0.00217   0.01789  -0.02485   0.33979   0.66645   0.24001   0.08342   0.04226   0.03947
    -1.49688  -0.36128   1.08385   0.14149  -0.80391   1.07313   0.19833   0.24001   3.51459  -1.28352   0.23626   2.88938
     6.04090   0.65196   1.35341   0.48143  -0.46127   0.22082   2.16530   0.08342  -1.28352   8.70154   4.21654  -0.11512
     2.16393   0.18864   0.78662   0.25230  -0.38679   0.33454   0.98611   0.04226   0.23626   4.21654   4.75477   1.89178
    -0.67649  -0.25011   1.03899   0.26105  -0.82231   0.98290   0.58777   0.03947   2.88938  -0.11512   1.89178   4.16004


        C           COMPUTE THE DISPLACEMENT MATRIX
  104               DO 35 NM=1,MM
  105               READ 30,(P(I),I=1,NK)
  106       30      FORMAT(3F10.0)
  107               NL=3*NN
  108               DO 40 I=1,NL
  109       40      C(I)=0
  110               DO 31 I=1,NK
  111               DO 31 J=NP1,MAT
  112               MK=J-NK
  113       31      C(I)=C(I)+STIF(I,J)*P(MK)
  114               PRINT 2012
  115     2012      FORMAT(1X/////14H DISPLACEMENTS)
  116               DO 32 I=1,NK
  117       32      PRINT 33,C(I)
  118       33      FORMAT(2X,F10.5)
  119               PRINT 50
```



```
DISPLACEMENTS       DISPLACEMENTS       DISPLACEMENTS       DISPLACEMENTS
   89.02038            -3.34129             1.64071           -17.31142
   16.46375            -7.10101             0.27498            -1.50915
   14.61802            31.64259            -0.29058            -6.29296
    7.37641            -0.99568            -0.04337            -2.01839
   -3.64349           -19.87529             0.35787             3.09431
   -1.78202            43.73625            -0.49707            -2.67634
   30.96166             8.28168             6.79578            -7.88890
    1.31257            -0.74560            13.32900            -0.33804
  -23.95001            32.19389             4.80020            -1.89010
   96.65444             6.62475             1.66832           -33.73229
   34.62284            10.03628             0.84510           -38.03819
  -10.82381            29.48712             0.78944           -15.13423

      C          COMPUTE THE STRESSES IN THE MEMBERS
          50     FORMAT(1HO,7H MEMBER,10X,6HSTRESS)
                 DO 35 J=1,LL
                 NT=MEML(J)
                 NO=MEMR(J)
                 XX=X(NT)-X(NO)
                 YY=Y(NT)-Y(NO)
                 ZZ=Z(NT)-Z(NO)
                 K1=3*NT-2
                 K2=K1+1
                 K3=K1+2
                 N1=3*NO-2
                 N2=N1+1
                 N3=N1+2
                 ST(NM,J)=A(J)/(XX*XX+YY*YY+ZZ*ZZ)*(XX*(C(K1)-C(N1))+YY*(C(K2)-C(N2
               1 ))+ZZ*(C(K3)-C(N3)))
          35     PRINT 53,MEML(J),MEMR(J),ST(NM,J)
          53     FORMAT(1X,I2,1H-,I2,F16.2)
                 PRINT 52
          52     FORMAT(1X////1X,6HMEMBER,11X,12HMAX. TENSION,6X,16HMAX. COMPRESSIO


   MEMBER          STRESS          STRESS          STRESS          STRESS
   2- 6            -8.33            2 27            0.12            2.71
   1- 2             6.64           13.95            0.08           -1.07
   1- 3             9 75           -5.54           -0.03           -2.51
   4- 7            10.66           -6.05           -0.04           -7.67
   2- 7             6.53            0 50            0 04           -1.27
   3- 7             1 97           -1.12           19.99           -0.51
   3- 2           -11 05            6.28            0 04            2.85
   5- 4            -7.31            4 15            0.02           -5.54
   4- 1             4.72           -2 68           -0.02            3.58
   5- 1            12 10            6 46            0.04           -2.63
   2- 5            -1 94           16.08           -0.15           -0.49
   3- 4           -10 78            6 13            0 04            2.78
   1- 6           -12.56          -13 35           -0.08            4 54

      C          MAXIMIZE THE STRESSES IN THE MEMBERS
                 POS=0
                 ZNEG=0
                 DO 37 J=1,LL
                 DO 36 I=1,MM
                 IF(ST(I,J))38,39,39
          38     ZNEG=ZNEG+ST(I,J)
                 GO TO 36
          39     POS=POS+ST(I,J)
          36     CONTINUE
                 PRINT 41,MEML(J),MEMR(J),POS,ZNEG
                 ZNEG=0.
          37     POS=0.
          41       FORMAT(2X,I2,1H-,I2,2F20.5)
                 STOP
                 END


MEMBER                 MAX. TENSION       MAX. COMPRESSION
  2- 6                    5.09080             -8.33115
  1- 2                   20.66922             -1.06947
  1- 3                    9.74534             -8.07780
  4- 7                   10.65817            -13.75569
  2- 7                    7.07372             -1.27145
  3- 7                   21.96236             -1.62547
  3- 2                    9.15937            -11.05018
  5- 4                    4.17598            -12.85126
  4- 1                    8.29545             -2.69559
  5- 1                   18.59280             -2.63282
  2- 5                   16.07759             -2.58565
  3- 4                    8.93863            -10.78387
  1- 6                    4.53539            -25.99218

PROCESSING TERMINATED   . (25A)
```

Figure 3

for all sets of loads and assembles them into maximum tension and maximum compression. These are printed out. There is the solution.

SAMPLE PROBLEM

The three-dimensional truss of Fig. 1 is analyzed by the method discussed. The necessary programs, written in FORTRAN II, are included.

Input Data—The input data sheet (Fig. 2) is prepared in the following form:

Line 1—The total number of joints, the number of joints restrained against displacement, the number of members, and the number of loading conditions to be investigated. The information is presented as fixed point with two columns for each piece of information.

Lines 2 to $n + 1$—For each joint, the x, y and z coordinates are given. There is a total of n joints. The coordinates are given in the same order as the joints are numbered. The numbering of the joints must be such that the joints restrained against displacement have the highest numbers; that is, joints 1 to m are free to displace while joints $m + 1$ to n are restrained. The information is presented as floating point with seven columns for each coordinate.

Lines $n + 2$ to k—For each member, the area of the member and the joint numbers corresponding to the ends of the member are given. There is a total of k-n-1 members. The area is given as floating point with ten columns allotted to it followed by the joint numbers in fixed point with two columns allotted for each. There is no particular order required for the members.

Lines k to 1—For each loading condition, the three components of the load are given at each joint. If there is no load at a particular joint, a zero is recorded for all components at that joint. If there are n number of loading conditions and m joints able to displace a total of $(3)(n)(m)$ lines are required. For each loading condition, the components of load must be presented in the same order as the joints were presented. The components of loading are given as floating point with ten columns allotted to each component.