s is generally not the

the degrees of static
acement method will
ding need to minimise
displacement method
ically within the com-
ary data necessary to
lly in a later chapter.

Automatic Analysis of Triangulated Frameworks for Multiple Load Sets and Initial Strains (The Matrix-Displacement Method) 6

6.1 Introduction

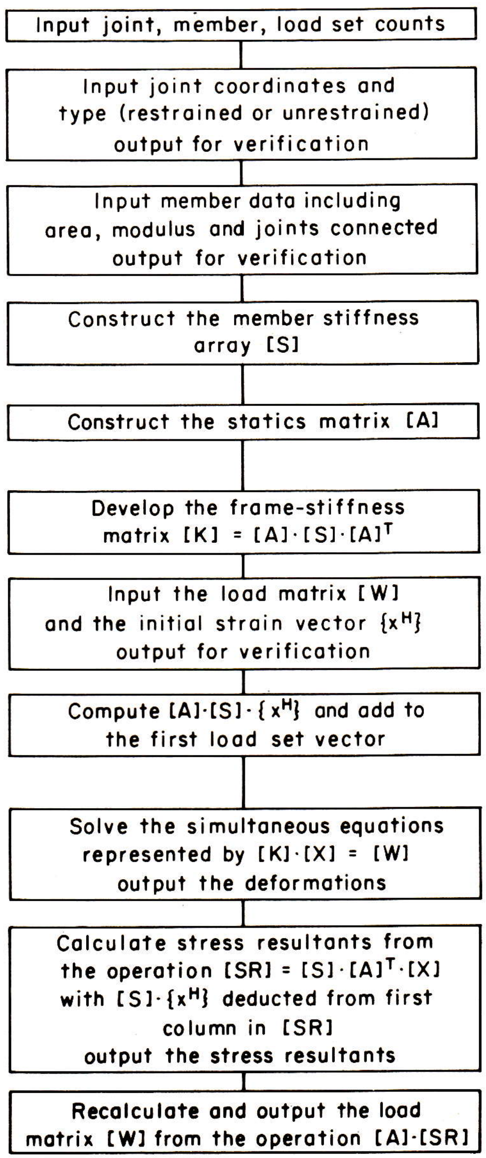
It was explained at the outset that the matrix methods had a considerable conceptual advantage over earlier manual methods of structural analysis, which was a consequence of their generality so that specific forms of structure did not require fundamentally different treatment. However, there is great advantage in treating the triangulated, simply connected framework in some detail, using these structures essentially as vehicles to the understanding of fully automatic computer methods. There is only one stress resultant and deformation per member to be considered, namely, the axial force and extension, and the joint-equilibrium equations are only two in number and are appropriately expressed as the zero summations of horizontal and vertical forces. If the analysis of these frames can be fully automated, it is then a simple step to extend the treatment to rigidly connected frameworks.

It is useful now to consider what is meant by a fully automatic analysis of a structure. Since a computer performs arithmetic accurately and quickly, and, if one is taking the trouble to use such a machine in structural analysis, it is axiomatic that it should perform all the calculations and not merely a part of them. It follows that the aim in constructing a program to analyze triangulated frameworks should include the presentation to the machine of the minimum amount of data necessary to describe a framework and its loads. Further, the output from the machine should not only include the displacements and stress resultants in an ordered and fully identified manner, but also the input data should be printed to enable ready verification. It is clear that the essential matrix required in the displacement method is the frame-stiffness matrix $[K]$, and one way of generating it is from the statics matrix $[A]$ and the member-stiffness matrix $[S]$ by the operation

$$[K] = [A]\cdot[S]\cdot[A]^{\mathrm{T}} \tag{6.1}$$

There is no problem in automating the construction of the member-stiffness matrix $[S]$ in the memory of a machine, since all that is required is member data

Fig. 6.1 General method for automatic displacement analysis of trusses



consisting of length, area, and
riate then to consider how the
if this is available, together wit
the initial-strain vector, the o
motion.

6.2 Scheme for General D

The principal stages in a c
framework should now be obv
may be described numerically
would be the count of the joint
required, so that an origin of c
found to indicate whether or n
and vertical directions. In oth
frame-support conditions. A s
of joint coordinates a pair of
depending upon whether the ho
or not restrained. The summat
of the degree of kinematic ind
the frame-stiffness matrix. The
to illustrate these concepts in
convenience it has been placed
sequentially, and the member-
However, the direction of load
rant system chosen for the fra
in Fig. 6.2, and it is obvious
member is required as well a
indicate which joints are con
sketched in the figure is fully
ditions, by the tabulated numb

In a computer program
conveniently identified as on
accordance with the rules fo
language. Within this limitati
grammer of the significance o
by the array [*CORD*] and the
stored in an array [*JTYPE*]. T
array denoted by [*MCON*] w
[*E*], respectively. These names
acceptable in other programm
NM are chosen to represent
analysis may proceed in a qu
to any plane truss.

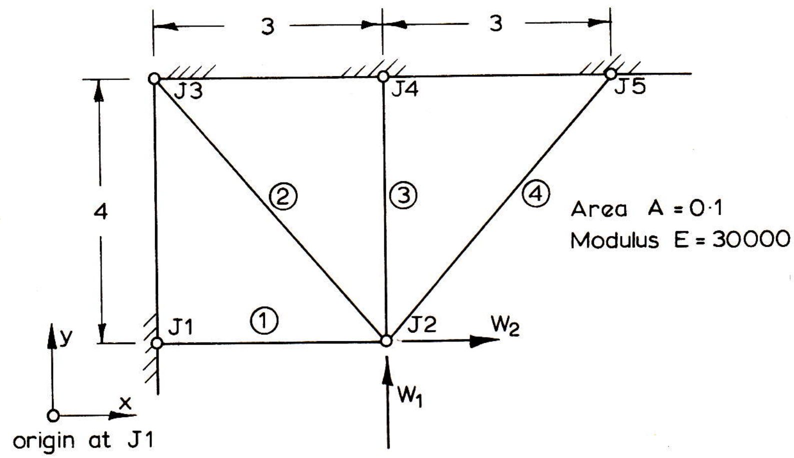
consisting of length, area, and elastic modulus of each member in turn. It is appropriate then to consider how the statics matrix $[A]$ may be automatically generated, for, if this is available, together with the load vector (or matrix for multiple load sets) and the initial-strain vector, the operations already outlined in Chapter 5 may be set in motion.

6.2 Scheme for General Displacement Analysis

The principal stages in a computer program to analyze any type of linear-elastic framework should now be obvious and are set out in Fig. 6.1. It is clear that a frame may be described numerically by a series of numbers, and the appropriate first pair would be the count of the joints and the members. The joint coordinates are obviously required, so that an origin of coordinates has to be chosen, and some means must be found to indicate whether or not each joint may deform elastically in the horizontal and vertical directions. In other words, the machine has to be informed about the frame-support conditions. A simple way of achieving this is to couple with each pair of joint coordinates a pair of integers, each of which may have the value of 0 or 1, depending upon whether the horizontal and vertical movement at the joint is restrained or not restrained. The summation of all these integers will then inform the computer of the degree of kinematic indeterminacy of the structure, which is also the order of the frame-stiffness matrix. The truss already treated in Chapters 2 and 5 has been used to illustrate these concepts in Fig. 6.2. An origin for coordinates is required, and for convenience it has been placed at joint number $J1$. All the joints have been numbered sequentially, and the member-numbering system used previously has been retained. However, the direction of load W_1 has been changed to conform with the first quadrant system chosen for the frame coordinates. The member data have been included in Fig. 6.2, and it is obvious that the area of section and elastic modulus for each member is required as well as the connection data of two integers per member to indicate which joints are connected by each member. It is clear that the frame as sketched in the figure is fully described in every aspect, except for the loading conditions, by the tabulated numbers.

In a computer program groups of numbers of the type shown in Fig. 6.2 are conveniently identified as one- or two-dimensional arrays with names chosen in accordance with the rules for variables pertinent to the particular programming language. Within this limitation, the names are usually selected to remind the programmer of the significance of the data. The coordinate data might well be identified by the array $[CORD]$ and the information about joint restraint or freedom could be stored in an array $[JTYPE]$. The member-connection data could be a two-dimensional array denoted by $[MCON]$ with the member areas and moduli held in $[AREA]$ and $[E]$, respectively. These names are acceptable FORTRAN variables but may not be acceptable in other programming languages. If the single-value variables JCT and NM are chosen to represent the count of the joints and members, respectively, the analysis may proceed in a quite general manner, applicable, in the present context, to any plane truss.

Fig. 6.2 Truss description in numeric form



Frame description in numeric form:

5	number of joints
4	number of members

$$\underset{[CORD]}{\begin{bmatrix} 0 & 0 \\ 3 & 0 \\ 0 & 4 \\ 3 & 4 \\ 6 & 4 \end{bmatrix}} \quad \underset{[JTYPE]}{\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}$$

coordinates and restraint indicators at joints $J1$ to $J5$

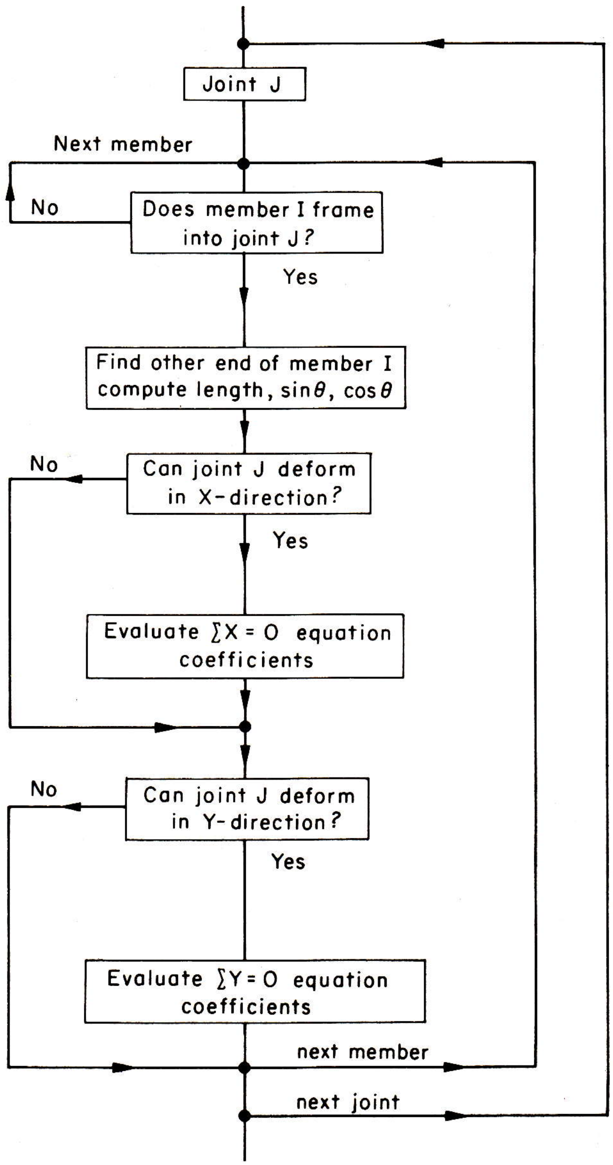
$$\underset{[MCON]}{\begin{bmatrix} 1 & 2 \\ 3 & 2 \\ 2 & 4 \\ 5 & 2 \end{bmatrix}} \quad \underset{[AREA]}{\begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}} \quad \underset{[E]}{\begin{bmatrix} 30000 \\ 30000 \\ 30000 \\ 30000 \end{bmatrix}}$$

member-connection integers with area and modulus for each member

---

With reference again to Fig. 6.1, the operations required to construct the member-stiffness array [*S*] are trivial in nature. The array dimension is the same as the number of stress resultants, which in the context of a pinned truss is the same as the number of members (*NM*). All the terms in the array would first be set to zero and then the diagonal terms evaluated from the expression $E(I)*A(I)/L(I)$, where the length of the i^{th} member $L(I)$ would be computed from the coordinates of its ends, stored in the array [*CORD*], and identified from the array [*MCON*].

Fig. 6.3 Construction of statics matrix [A]



The generation of the statics matrix [A] will require the automation of the manual operations already shown for the simple truss in Fig. 5.1. The end result shown in this figure would be achieved in the following stages, with reference now to Fig. 6.3. Every joint must be examined in turn, and, for each joint, every member is studied in turn to find the ones framing into the joint. Such information is contained in the array [*MCON*]. If a member frames into a joint, it is then necessary to locate the far end of the member so that the length and inclination may be found using the coordinate data stored in the array [*CORD*]. Finally, it is necessary to study the array [*JTYPE*] to see if a static equation needs to be generated. This will be the case whenever the integer 1 is found in the array, and, for the frame in Fig. 6.2, it is clear that the only equations to be generated will correspond with the two possible movements of the joint *J*2. The sequence of rows in the statics array will correspond to the sequence in which these unit integers are encountered in the array [*JTYPE*], whereas the column number is determined by the identification number of the member under consideration.

It is interesting to compare the relative amounts of work involved in generating the statics array for the frame in Fig. 5.1 by hand and by the computer-programmed system just described. The experienced analyst recognizes immediately from the diagram that only two static equations are required and that these express horizontal and vertical equilibrium at the loaded joint. The computer, on the other hand, has to study each of the five joints with a search made through all four members every time to find the ones framing into a joint. It goes through the motions of establishing 10 equations before it finally determines the coefficients of the relevant pair.

With the statics and member-stiffness arrays now available, the analysis can proceed following the theory outlined in Chapter 5. The frame-stiffness matrix [*K*] is generated as in equation (6.1), and the load vector $\{W\}$ or the load matrix [*W*] input and stored as a column-wise extension to the stiffness matrix. The load matrix would take the form of a collection of the multiple load vectors if analysis for more than one load set is required. The initial-strain vector $\{x^H\}$ would also be input, premultiplied by the product $[A]\cdot[S]$ and added to the first load vector following the theory expressed in equation (5.11). In this way, when initial strain effects alone are of interest, the first load vector would be input with zero terms throughout. With the coefficients and right-hand sides for the frame-stiffness equations now assembled, an appropriate equation solver such as the Gauss-Jordan scheme in Fig. 4.12 may be used to produce the joint deformations which will appear in that part of the augmented [*K*] array where initially the load matrix was located. Stress resultants would then be computed from the additional operation,

$$[SR] = [S]\cdot[A]^{\mathrm{T}}\cdot[X] \tag{6.2}$$

and, since initial-strain effects were added only to the first load set, it will be necessary, following equation (5.13), for the product $[S]\cdot\{x^H\}$ to be deducted from the first column only of the array [*SR*]. The stress-resultant array has been indicated as two dimensional rather than as a vector, since multiple load sets are involved. Stress resultants produced by each load set will appear in the corresponding column of the [*SR*] array.

Finally, as a check on accumulated round-off errors, if not of data inaccuracies, the stress-resultant array [*SR*] may be premultiplied by the statics array [*A*], and the result, according to equation (5.14), should be the initial load matrix.

6.3 Program for the Analysis of Plane and Space Trusses

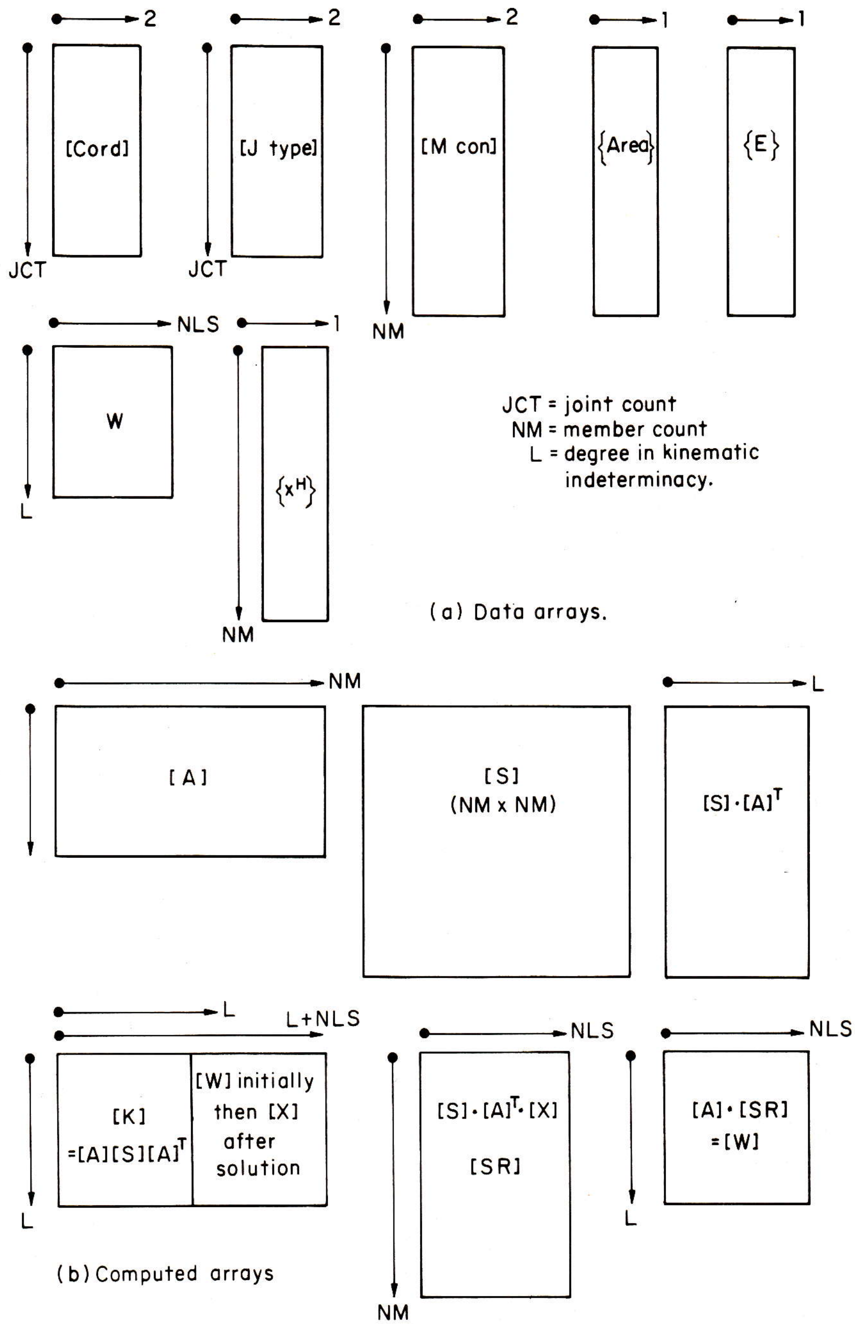
If the steps outlined in the previous section were programmed for a machine, it would be inevitable that the programmer, with increasing skill and confidence, would soon begin to effect improvements with the general aim of increasing the speed of the calculation and reducing the sizes of the arrays needed to hold the results of intermediate calculations. Equally inevitable would be his realization that a little extra effort would make the program suitable for space as well as plane, simply connected frameworks. The statics-matrix generator would need to deal with three rather than two equilibrium equations per joint and the coordinate and joint-type arrays would have three rather than the two columns, as shown in Fig. 6.4(a), where these arrays have been sketched diagramatically.

The computed arrays in Fig. 6.4(b) would soon receive close attention with a view to conserving storage space, and the large array [*S*] consisting of nonzero terms only down the diagonal would be replaced by a single-column array or vector containing only these nonzero terms. Other improvements may well be to conserve storage space at the expense of increased calculation if the result were to make the program suitable for considerably larger frames when using a particular machine. In the program listed in Appendix 6.1, all the computed arrays are eliminated except for the statics array [*A*], which is overwritten by the subsequently developed sets of numbers, at the price of regenerating it when it is further needed to produce the final two matrices. The progressive overwriting of the statics array by the rows of the frame-stiffness matrix [*K*] will introduce errors into the latter, which are subsequently eliminated through making use of the theoretical knowledge that this array must be symmetrical about the leading diagonal.

The programmer would next study the load matrix [*W*], which would be tedious to prepare by hand when load cases only involve loads at a few joints so that zeros have to be entered for all the remaining unloaded joints where elastic movements can occur. Again at the small price of extra computing time, he would think first of the simplest way of informing a machine of a load set and then arrange additional instructions for the load vector to be automatically generated and placed directly beside the augmented [*K*] array, which itself, by this time, would have overwritten the statics array. Hence, at the cost of a small amount of extra calculation, the data array [*W*] may be completely eliminated. Automating the construction of the initial-strain vector is not possible in the general case, where it may include lack-of-fit terms, but, if the assumption is made that only temperature effects are involved and that all members have the same coefficient of expansion, the construction of $\{x^H\}$ can be automated, and the only data needed at the input stage would be the coefficient of expansion and the temperature rise.

All these improvements have been made in the program listed in Appendix 6.1 and are the explanation of why a step-by-step examination of such a listing can be a puzzling experience for a relative novice in programming. Additional improvements are possible and are associated with taking advantage of the symmetry of the [*K*] array to cut its size almost in half at the price of more complexity in the equation-

Fig. 6.4 Data and computed arrays in the displacement analysis of plane trusses



solving routine. The p
zero terms well away f
taneous saving in stor
will be examined furth

No attempt will b
listing in Appendix 6.1
ments is to provide a p
complexity of a progr
handle any determinat
The size of frame that c
in which the arrays are
able memory of any p
the program does not i
and of recalculating loa
brief notes have been a
blocks in the program
redundant truss in Fig.
compared with the pre
and 2.11, where the for
method was used.

Appendix 6.1 FORTRA

General Notes

1. The frame-identificatio or 3 for a plane or a sp
2. Member and joint dat
3. The statics matrix is ge
4. The initial-strain vecto
5. The frame-stiffness ma lines 0690–0810.
6. Load sets are input an first set only by statem
7. The stiffness equations in statement lines 117
8. Titles and deformation
9. Stress resultants are cal lines 1740–1910, and s 1920–2001.

solving routine. The possibility that this array may be "sparse" or may have mainly zero terms well away from the diagonal could be examined with a view to a simultaneous saving in storage and calculating time; this subject of band-width control will be examined further in a later chapter.

No attempt will be made here to give a step-by-step explanation of the program listing in Appendix 6.1; the principal purpose in attaching the list of some 210 statements is to provide a practical demonstration of the physical size and relative order of complexity of a program in unsophisticated FORTRAN, which can theoretically handle any determinate or redundant plane or space truss with simple connections. The size of frame that can be analyzed is of course determined by the initial statements in which the arrays are dimensioned, and these dimensions must not exceed the available memory of any particular machine. To keep the listing as concise as possible, the program does not include the desirable features of outputting data for verification and of recalculating loads as a final check on both data and accumulated errors. Some brief notes have been added at the beginning of Appendix 6.1 to show the principal blocks in the program, and the results of the application of the program to the redundant truss in Fig. 6.2 have been set out in Appendix 6.2. The answers may be compared with the previous results in influence-coefficient form in Figs. 2.8, 2.10, and 2.11, where the force method was used, and in Fig. 5.5, where the displacement method was used.

Appendix 6.1 FORTRAN LISTING OF PROGRAM *CEST*

General Notes

1. The frame-identification number is the value of the variable *JJ*, whereas *JJJ* is read as 2 or 3 for a plane or a space truss, respectively.
2. Member and joint data are input in the general region of statement lines 0030–0110.
3. The statics matrix is generated by statement lines 0330–0610.
4. The initial-strain vector for temperature effects is assembled by statement lines 0630–0680.
5. The frame-stiffness matrix overwrites the statics array as a consequence of statement lines 0690–0810.
6. Load sets are input and the load matrix assembled with temperature effects added to the first set only by statement lines 0820–1160.
7. The stiffness equations are solved using Gauss-Jordan elimination with pivotal selection in statement lines 1170–1400.
8. Titles and deformations are output by statement lines 1410–1730.
9. Stress resultants are calculated, requiring the regeneration of the statics array, in statement lines 1740–1910, and stress resultants are output as a consequence of statement lines 1920–2001.

Appendix 6.1 FORTRAN LISTING OF PROGRAM *CEST* (Contd.)

Program *CEST*

```
COMMON CORD(20,3),JTYPE(20,3),AREA(27),A(27,30),CSAT(27),
+DEFX(27,4),LSN(4),SR(27,4),MCON(27,2),OLEN(27)
$FILE CEFIL*
10 READ(1),JJ,JJJ,JCT,NM,TR,ALPHA,E
IF(JJ)20,30,30
20 STOP
30 IF(E)40,40,50
40 E=207000.0
50 DO 60 I=1,JCT
60 READ(1),(CORD(I,J),J=1,JJJ),(JTYPE(I,J),J=1,JJJ)
DO 70 I=1,NM
70 READ (1),(MCON(I,J),J=1,2),AREA(I)
L=0
DO 80 I=1,JCT
DO 80 J=1,JJJ
80 L=L+JTYPE(I,J)
DO 100 I=1,NM
J1=MCON(I,1)
J2=MCON(I,2)
X=CORD(J1,1)-CORD(J2,1)
Y=CORD(J1,2)-CORD(J2,2)
IF(JJJ-2)95,95,96
95 OLEN (I)=SQRT(X*X+Y*Y)
GO TO 100
96 Z=CORD(J1,3)-CORD(J2,3)
OLEN(I)=SQRT(X*X+Y*Y+Z*Z)
100 CONTINUE
IF(ALPHA) 105,104,105
104 ALPHA=.000011
105 ISW=1
110 NJ=0
NK=0
DO 90 I=1,L
DO 90 J=1,NM
90 A(I,J)=0.
DO 270 J=1,JCT
DO 260 M=1,NM
NA=NJ
IF(J-MCON(M,1))130,120,130
120 JF=MCON(M,2)
GO TO 150
130 IF(J-MCON(M,2))260,140,260
140 JF=MCON(M,1)
150 X=CORD(JF,1)-CORD(J,1)
Y=CORD(JF,2)-CORD(J,2)
IF(JJJ-2)170,170,160
160 Z=CORD(JF,3)-CORD(J,3)
```

Appendix 6.1 FORTRAN LISTING OF PROGRAM *CEST* (Contd.)

```
170 IF(JTYPE(J,1))190,190,180
180 NA=NA+1
A(NA,M)=-X/OLEN(M)
190 IF(JTYPE(J,2))210,210,200
200 NA=NA+1
A(NA,M)=-Y/OLEN(M)
210 IF(JJJ-2)240,240,220
220 IF(JTYPE(J,3))240,240,230
230 NA=NA+1
A(NA,M)=-Z/OLEN(M)
240 IF(NA-NK)260,260,250
250 NK=NA
260 CONTINUE
NJ=NK
270 CONTINUE
IF(ISW) 890,890,280
280 IF(TR)282,289,282
282 DO 283 I=1,NM
283 CSAT(I)=E*AREA(I)*ALPHA*TR
DO 284 I=1,L
DEFX(I,2)=0.
DO 284 J=1,NM
284 DEFX(I,2)=DEFX(I,2)+A(I,J)*CSAT(J)
289 DO 320 J=1,L
DO 290 I=1,NM
290 CSAT(I)=E*AREA(I)*A(J,I)/OLEN(I)
DO 310 I=J,L
TEMP=0.
DO 300 K=1,NM
300 TEMP=TEMP+A(I,K)*CSAT(K)
310 DEFX(I,1)=TEMP
DO 320 I=1,L
320 A(J,I)=DEFX(I,1)
DO 330 I=1,L
DO 330 J=1,L
330 A(J,I)=A(I,J)
LSC=L
340 READ(1),KK,LN
LSC=LSC+1
IF(KK) 470,470,350
350 DO 360 I=1,L
360 A(I,LSC)=0.
I=LSC-L
LSN(I)=KK
DO 460 I=1,LN
READ(1), JN,XF,YF
IF(JJJ-2) 380,380,370
370 READ(1),ZF
380 LL=0
```

Appendix 6.1 FORTRAN LISTING OF PROGRAM *CEST* (Contd.)

```
LJ=JN-1
IF(LJ) 395,395,385
385 DO 390 J=1,LJ
DO 390 K=1,JJJ
390 LL=LL+JTYPE(J,K)
395 IF(JTYPE(JN,1)) 410,410,400
400 LL=LL+1
A(LL,LSC)=XF
410 IF(JTYPE(JN,2)) 430,430,420
420 LL=LL+1
A(LL,LSC)=YF
430 IF(JJJ-2) 460,460,440
440 IF(JTYPE(JN,3)) 460,460,450
450 LL=LL+1
A(LL,LSC)=ZF
460 CONTINUE
GO TO 340
470 IF(TR) 480,500,480
480 LS=L+1
DO 490 I=1,L
490 A(I,LS)=A(I,LS)+DEFX(I,2)
500 KJ=LSC-1
NLS=KJ-L
DO 680 I=1,L
IP1=I+1
TEMP=ABS(A(I,I))
K=I
DO 590 J=I,L
IF(ABS(A(J,I))-TEMP) 590,590,585
585 K=J
TEMP=ABS(A(J,I))
590 CONTINUE
IF(K-I) 600,620,600
600 DO 610 J=I,KJ
TEMP=A(I,J)
A(I,J)=A(K,J)
610 A(K,J)=TEMP
620 IF(A(I,I)) 640,998,640
640 TEMP=1./A(I,I)
DO 650 J=I,KJ
650 A(I,J)=A(I,J)*TEMP
DO 680 J=1,L
IF(I-J) 660,680,660
660 TEMP=A(J,I)
DO 670 K=IP1,KJ
670 A(J,K)=A(J,K)-TEMP*A(I,K)
680 CONTINUE
PRINT 700,JJ
```

Appendix 6.1 FORTRAN LISTING OF PROGRAM *CEST* (Contd.)

```
700 FORMAT(37H LINEAR ELASTIC ANALYSIS OF TRUSS NO.,I5///)
PRINT 730,ALPHA,TR
730 FORMAT(16H COEFF.EXPANSION,F11.8,10H TEMP.RISE,F7.2//)
PRINT 740
740 FORMAT(23H THE JOINT DEFORMATIONS//)
DO 870 I=1,NLS
KL=L+I
PRINT" "
PRINT 750,LSN(I)
750 FORMAT(37H DISPLACEMENTS CAUSED BY LOAD SET NO., I3//)
LL=0
PRINT"     JOINT   X-DEFLECTION Y-DEFLECTION Z-DEFLECTION"
DO 870 J=1,JCT
IF(JTYPE(J,1)) 780,780,770
770 LL=LL+1
AA=A(LL,KL)
GO TO 790
780 AA=-0.
790 IF(JTYPE(J,2)) 810,810,800
800 LL=LL+1
AB=A(LL,KL)
GO TO 820
810 AB=-0.
820 IF(JJJ-2) 825,825,830
825 AC=0.
GO TO 860
830 IF(JTYPE(J,3)) 850,850,840
840 LL=LL+1
AC=A(LL,KL)
GO TO 860
850 AC=-0.
860 PRINT 861,J,AA,AB,AC
861 FORMAT(I8,F14.5,2F13.5)
870 CONTINUE
DO 880 I=1,L
DO 880 J=1,NLS
K=L+J
880 DEFX(I,J)=A(I,K)
ISW=-1
GO TO 110
890 DO 900 I=1,NM
TEMP=E*AREA(I)/OLEN(I)
DO 900 J=1,L
900 A(J,I)=A(J,I)*TEMP
DO 920 I=1,NM
DO 920 J=1,NLS
SR(I,J)=0.
DO 920 K=1,L
```

Appendix 6.1 FORTRAN LISTING OF PROGRAM *CEST* (Contd.)

```
920 SR(I,J)=SR(I,J)+A(K,I)*DEFX(K,J)
IF(TR) 930,950,930
930 DO 940 I=1,NM
940 SR(I,1)=SR(I,1)-E*AREA(I)*ALPHA*TR
950 PRINT" "
PRINT 960
960 FORMAT(24H AXIAL FORCES IN MEMBERS//)
DO 990 I=1,NLS
PRINT" "
PRINT 970,LSN(I)
PRINT" "
970 FORMAT(38H AXIAL TENSIONS CAUSED BY LOAD SET NO., I3//)
PRINT"    MEMBER   AXIAL   TENSION   STRESS"
DO 990 J=1,NM
STRS=SR(J,I)/AREA(J)
990 PRINT 991,J,SR(J,I),STRS
991 FORMAT(I8,F14.5,F11.3)
PRINT" "
GO TO 999
998 PRINT" ZERO DIVISION IN EQUATION SOLUTION"
999 PRINT 1000,JJ
1000 FORMAT(33H ANALYSIS COMPLETED FOR TRUSS NO.,I5////)
GO TO 10
END
```

Appendix 6.2 ANALYSIS OF TRUSS EXAMPLE BY PROGRAM *CEST*

The essential data are as set out in Fig. 6.2. Four load sets were considered with no external loads in set 1, so that the results for this set include only the deformations and stress resultants associated with the nominated temperature rise of 100°. Load set 2 involved only a horizontal load of unit value at joint 2, and a downward vertical load of unit value comprised load set 3. Load set 4 was taken as a combination of sets 2 and 3. The entire data list in the free-formatted sequence required by the program for the frame is set out below.

2	2	5	4			JJ, JJJ, JCT, NM
100	0.0000059	30000				TR, α, E
0	0	0	0			Joint 1 data
3	0	1	1			Joint 2 data
0	4	0	0			Joint 3 data
3	4	0	0			Joint 4 data
6	4	0	0			Joint 5 data
1	2	0.1	2	3	0.1	Members 1 and 2
2	4	0.1	5	2	0.1	Members 3 and 4
1	1	2	0	0		Load set 1
2	1	2	1	0		Load set 2
3	1	2	0	−1		Load set 3
4	1	2	1	−1		Load set 4
−1	0					(End of load sets)

Appendix 6.2 ANALYSIS OF TRUSS EXAMPLE BY PROGRAM *CEST* (Contd.)

Truss Analysis by Program *CEST*

LINEAR ELASTIC ANALYSIS OF TRUSS NO. 2

COEFF.EXPANSION .00000590 TEMP.RISE 100.00

THE JOINT DEFORMATIONS

DISPLACEMENTS CAUSED BY LOAD SET NO. 1

JOINT	X-DEFLECTION	Y-DEFLECTION	Z-DEFLECTION
1	.00000	.00000	.00000
2	.00124	−.00303	.00000
3	.00000	.00000	.00000
4	.00000	.00000	.00000
5	.00000	.00000	.00000

DISPLACEMENTS CAUSED BY LOAD SET NO. 2

JOINT	X-DEFLECTION	Y-DEFLECTION	Z-DEFLECTION
1	.00000	.00000	.00000
2	.00070	.00000	.00000
3	.00000	.00000	.00000
4	.00000	.00000	.00000
5	.00000	.00000	.00000

DISPLACEMENTS CAUSED BY LOAD SET NO. 3

JOINT	X-DEFLECTION	Y-DEFLECTION	Z-DEFLECTION
1	.00000	.00000	.00000
2	.00000	−.00066	.00000
3	.00000	.00000	.00000
4	.00000	.00000	.00000
5	.00000	.00000	.00000

DISPLACEMENTS CAUSED BY LOAD SET NO. 4

JOINT	X-DEFLECTION	Y-DEFLECTION	Z-DEFLECTION
1	.00000	.00000	.00000
2	.00070	−.00066	.00000
3	.00000	.00000	.00000
4	.00000	.00000	.00000
5	.00000	.00000	.00000

AXIAL FORCES IN MEMBERS

AXIAL TENSIONS CAUSED BY LOAD SET NO. 1

MEMBER	AXIAL STRESS	STRESS
1	−.53397	−5.340
2	.13015	1.301
3	.50372	5.037
4	−.75979	−7.598

Appendix 6.2 ANALYSIS OF TRUSS EXAMPLE BY PROGRAM *CEST* (Contd.)

AXIAL TENSIONS CAUSED BY LOAD SET NO. 2

MEMBER	AXIAL TENSION	STRESS
1	.69832	6.983
2	.25140	2.514
3	.00000	.000
4	−.25140	−2.514

AXIAL TENSIONS CAUSED BY LOAD SET NO. 3

MEMBER	AXIAL TENSION	STRESS
1	.00000	.000
2	.31621	3.162
3	.49407	4.941
4	.31621	3.162

AXIAL TENSIONS CAUSED BY LOAD SET NO. 4

MEMBER	AXIAL TENSION	STRESS
1	.69832	6.983
2	.56760	5.676
3	.49407	4.941
4	.06481	.648

ANALYSIS COMPLETED FOR TRUSS NO. 2