GRAPHIC DISPLAY OF LATTICE ATOMS

Supervised By:
Prof. T.H. Merrett

Godfrey Iloabachie
School of Computer Science
McGill University
December, 1976.

This report describes a conversational program which uses graphic routines to plot on a Tektronix Screen, a projection of the atomic positions in a given crystal lattice.

Most often a research worker, whether a crystallographer or a physicist, wishes to visualize the arrangement of the atoms in certain crystalline structures. Figure 1 shows an artistic representation of a packing drawing of the cubic arrangements of atoms in Na Pb_3 compound. An ideal situation will be to reproduce such artistic drawings in the computer, emphasizing the bonding arrangements, interatomic distances and distinguishing (probably by shaded lines) which atoms lie above others and which are hidden behind the direction of view. To produce such packing drawings on the line printer or on a display screen will not go without undue programming difficulties.

The present program does not attempt to reproduce a packing drawing of atomic arrangements but instead produces a projection of the atom positions on planes normal to the direction of view. Figure 2 shows a cube face projection of the same Na Pb_3 structure. The larger circles represent Na atoms.

The program however, represents each atom position by a symbol (one character symbol) which the user has to supply as part of the input. Atom positions are generated in 3-space in planes normal to the direction of view. Such planes are represented by integer numbers 0 through 9. (No more than 10 planes deep are displayed). The plane closest to the viewer is assigned the number 0 while the farthest plane from the viewer is plane

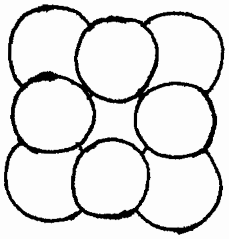


FIGURE 1

Packing Drawing of Na Pb_3

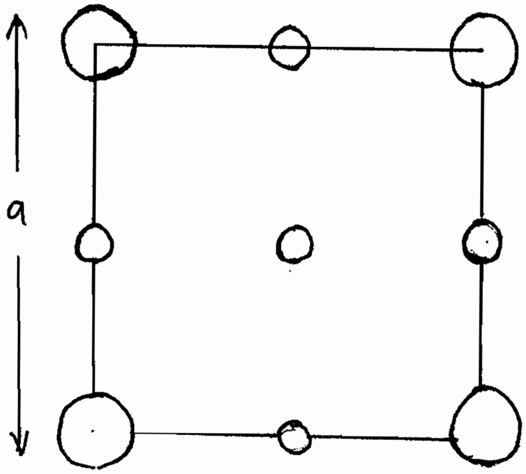


FIGURE 2

Projection on a cube face

number 9. Hence an atom position represented by P, lies in a plane nearer to the viewer than an atom position represented by P_3.

INPUT DATA

Presently the program is designed to display only lattice structures which have a cubic symmetry e.g. face centred cubic (f.c.c.) or body centred cubic Structures. This is important because the cell dimension A, asked for in the program is the length of a cubic face of the unit cell. If we consider the vectors $\underset{\sim}{u} = (1,0,0)$; $\underset{\sim}{v} = (0,1,0)$; $\underset{\sim}{w} = (0,0,1)$ as unit vectors in the cartesian coordinate system, we obtain a unit cell of the crystal by translating points a distance A units along each of the three directions $\underset{\sim}{u}$, $\underset{\sim}{v}$, $\underset{\sim}{w}$.

In the language of crystallography a unit cell may be defined as the smallest volume from which the entire space grouping can be built by repetition (translation) along the coordinate axes. The following example gives the necessary crystallographic data required for input into the program:

Crystal Name : Cu_2 Mg
Type Structure : face centred cubic
Equiv. Positions: (0,0,0);(0,1/2,1/2);(1/2,0,1/2);(1/2,1/2,0)
Cell Dimension : 8
Centring Points :

Mg : (0,0,0);(1/4,1/4,1/4)
Cu : (5/8,5/8,5/8);(5/8,7/8,7/8);(7/8,5/8,7/8);
(7/8,7/8,5/8).

The equivalent positions give coordinates of points of symmetry in the structure. The coordinates of the centring points will therefore be added in turn to the coordinates of the equivalent positions to obtain all the atom positions in the unit cell. In the above example there are two centring atoms of Mg located at coordinates (0,0,0) and (1/4,1/4,1/4). These in turn will be added to the four equiv positions to obtain eight Mg atoms in the unit cell. Similarly, the four centring Cu atoms will be added in turn to the equivalent positions to give 16 Cu atoms. This gives a total of 24 atoms in the unit cell of Cu_2 Mg structure. By translating each of the above 24 atoms a distance of 8 units each time, along the directions $\underset{\sim}{u}$, $\underset{\sim}{v}$, $\underset{\sim}{w}$, the program generates the Space lattice--a three-dimensional pattern of points in Space. The transformation routines then project these lattice points into planes normal to the direction of view.

An option in the program therefore asks for the view direction. If we may write $[x,y,z]$ to represent the direction from the origin (0,0,0) to the lattice point (x,y,z), then the user merely types in the coordinates x,y,z seperated by at least one space between them. This is interpreted to mean the observer's eye is at (x,y,z) looking towards the origin.

OPTIONS AT DISPLAY

Figure 3 shows the program flow indicating the input sequence and the options available after each display.

VIEW: Typing this word after each display will enable the user view the same structure from a different direction.

MORE: To vary the number of atoms displayed each time, the user types in the word 'MORE'. The same structure will be displayed corresponding to the number of atoms entered by user. An approximation to the number of atoms displayed on Screen is computed as follows:-

Let number of atoms in unit cell = SUMA

Number of atoms required = NA

Number of unit cells required = NA/SUMA

Let D = $\left\lceil (NA/SUMA)^{1/3} \right\rceil$

Number of atom positions actually generated = $SUMA * D^3$

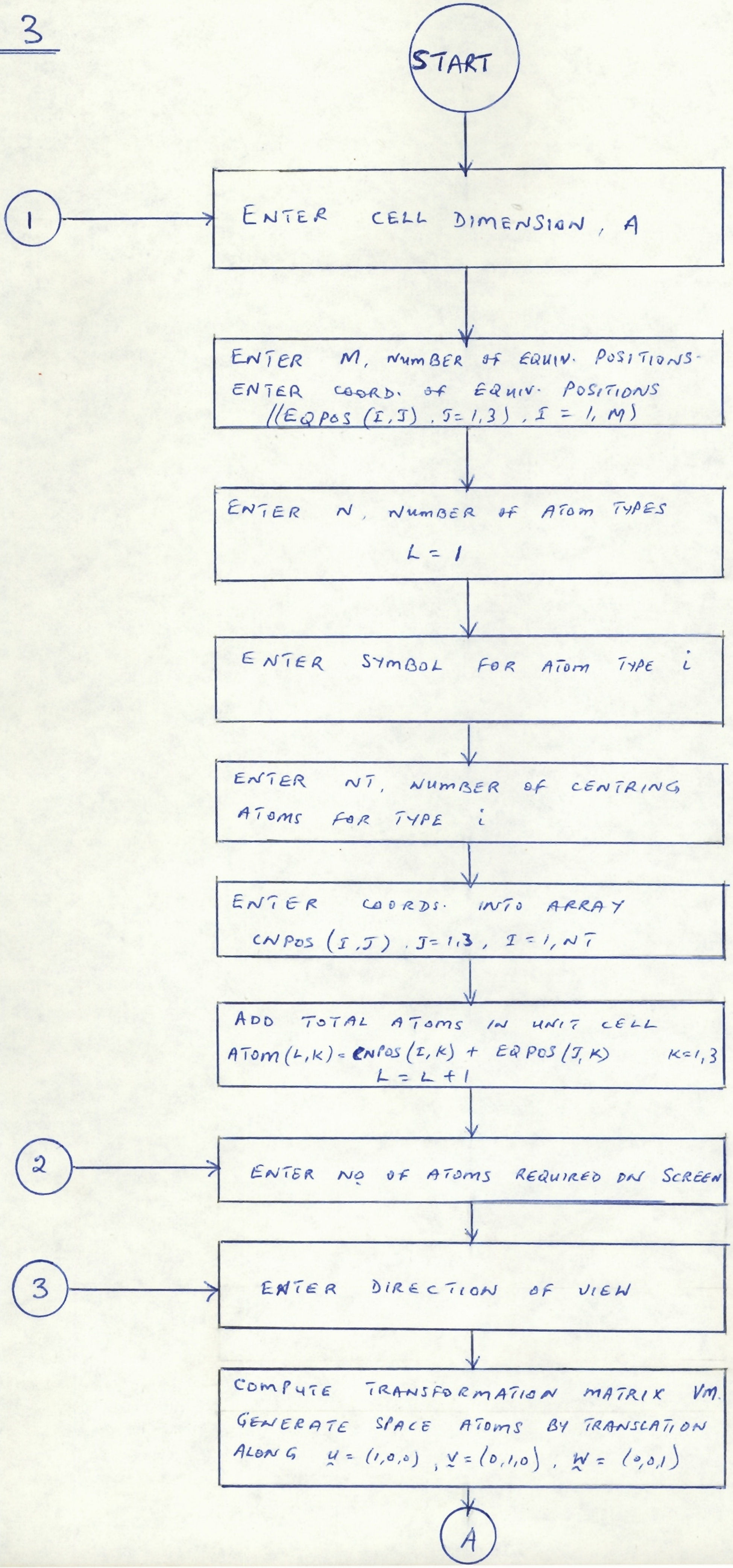
Fraction of this number to be displayed = f = NA/SUMA

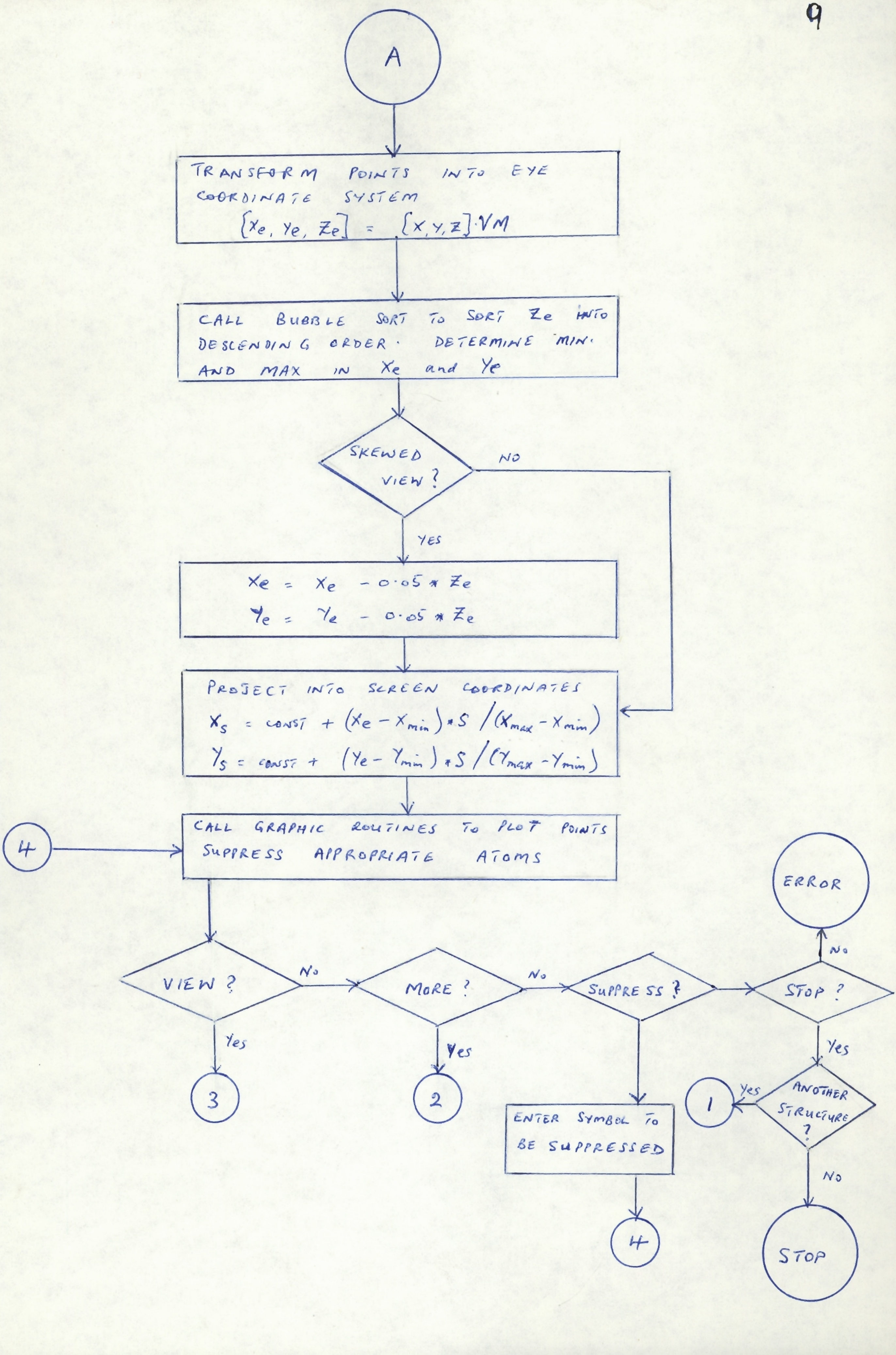
SUPPRESS: This option will enable the user to suppress the printing of any atom type. Only one type of atom can be suppressed during each display. The same structure will be displayed as before but leaving out appropriate atoms.

SKEW: There is an option to skew the view by about 5%. Atom positions in different planes which project on top one another are separated. This is accomplished by subtracting 5% of the Z_e from the corresponding X_e and Y_e values.

STOP: This option will eventually terminate the program but the user is first given an opportunity to view a different structure if he wishes.

FIG 3


START
1
ENTER CELL DIMENSION, A
ENTER M, NUMBER OF EQUIV. POSITIONS.
ENTER COORDS. OF EQUIV. POSITIONS
((EQPOS (I,J), J=1,3), I = 1, M)
ENTER N, NUMBER OF ATOM TYPES
L = 1
ENTER SYMBOL FOR ATOM TYPE i
ENTER NT, NUMBER OF CENTRING
ATOMS FOR TYPE i
ENTER COORDS. INTO ARRAY
CNPOS (I,J), J=1,3, I=1,NT
ADD TOTAL ATOMS IN UNIT CELL
ATOM(L,K) = CNPOS (I,K) + EQPOS (J,K) K=1,3
L = L+1
2
ENTER NO OF ATOMS REQUIRED ON SCREEN
3
ENTER DIRECTION OF VIEW
COMPUTE TRANSFORMATION MATRIX VM.
GENERATE SPACE ATOMS BY TRANSLATION
ALONG u = (1,0,0), v = (0,1,0), w = (0,0,1)
A


A
TRANSFORM POINTS INTO EYE COORDINATE SYSTEM
[Xe, Ye, Ze] = [X, Y, Z]·VM
CALL BUBBLE SORT TO SORT Ze INTO DESCENDING ORDER. DETERMINE MIN. AND MAX IN Xe and Ye
SKEWED VIEW ?
NO
YES
Xe = Xe - 0·05 * Ze
Ye = Ye - 0·05 * Ze
PROJECT INTO SCREEN COORDINATES
Xs = CONST + (Xe - Xmin) * S / (Xmax - Xmin)
Ys = CONST + (Ye - Ymin) * S / (Ymax - Ymin)
4
CALL GRAPHIC ROUTINES TO PLOT POINTS SUPPRESS APPROPRIATE ATOMS
VIEW ?
No
Yes
3
MORE ?
No
Yes
2
SUPPRESS ?
ENTER SYMBOL TO BE SUPPRESSED
4
STOP ?
No
ERROR
Yes
ANOTHER STRUCTURE ?
Yes
1
NO
STOP

COMPUTATIONS AND TRANSFORMATIONS

The Eye Coordinate System

In order to calculate the position of a point on the display screen which corresponds to an atom position, we must first transform the point from object space into the eye coordinate system. The eye coordinate system has its origin fixed at the viewpoint and its Z-axis (Z_e), points in the direction of view. See Figure 4. A transformation which we shall call the view transformation V, is used to convert points in object space with coordinates (X,Y,Z) to points in the eye coordinate system with corresponding coordinates (Xe,Ye,Ze).

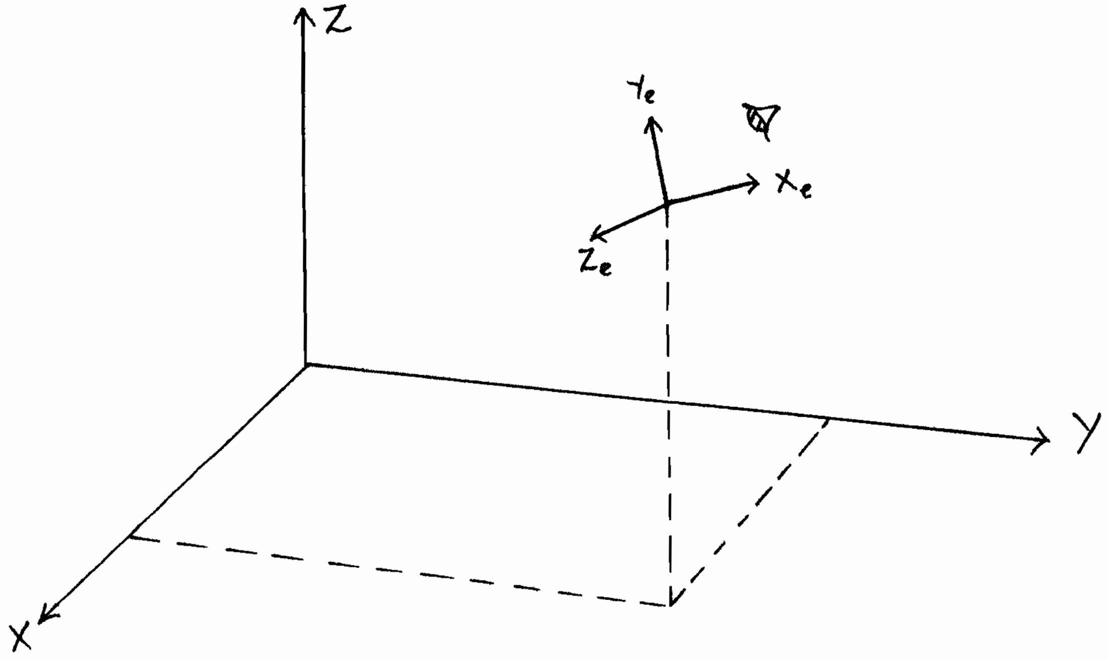


FIGURE 4

Thus we may write (Xe,Ye,Ze,1) = (X,Y,Z,1)V. The parameters of this transformation completely determine the view. The transformation matrix itself is derived from a combination of several rotations and translations as shown in next section. The eye coordinate system is a left-handed cartesian coordinate system. The Ze axis points forward from the viewpoint, the X_e to the right and the Ye axis points up. These conventions are essential so that the Xe and Ye axis will align with the X and Y axes (X_s,Y_s) of the display screen. See Figure 5.

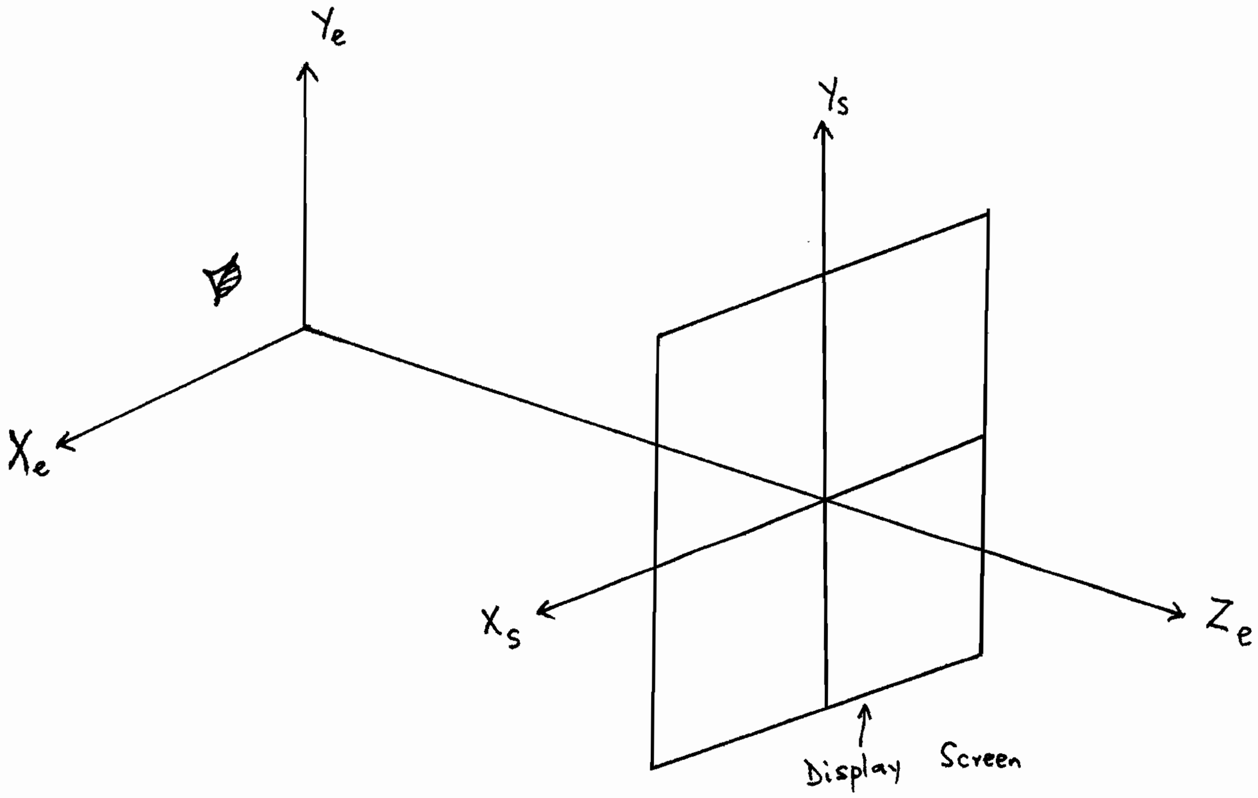


FIGURE 5

Viewing Transformation

Define (X,Y,Z) as a point from which the observer wishes to view the atoms. The viewing transformation V is derived from combinations of the following primitive transformations.

1. Translation: First translate the points into a new coordinate system where (X,Y,Z) maps into the origin (0,0,0). See Figure 6. The matrix T_1 will accomplish this result.

$$T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X & -Y & -Z & 1 \end{pmatrix}$$

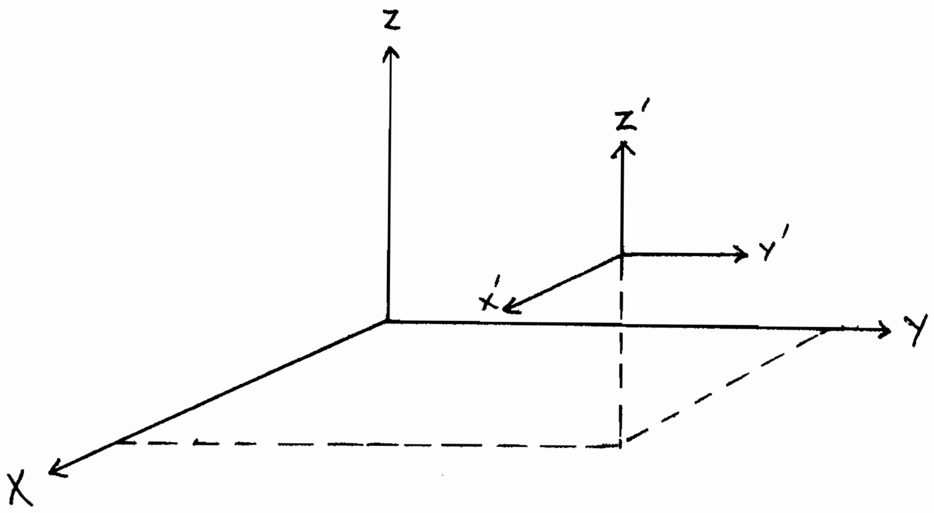


FIGURE 6

2. Next we rearrange the axes so that we have a left-handed system. See Figure 7. Multiplying each coordinate point by matrix T_2 will achieve this rearrangement.

$$T_2 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

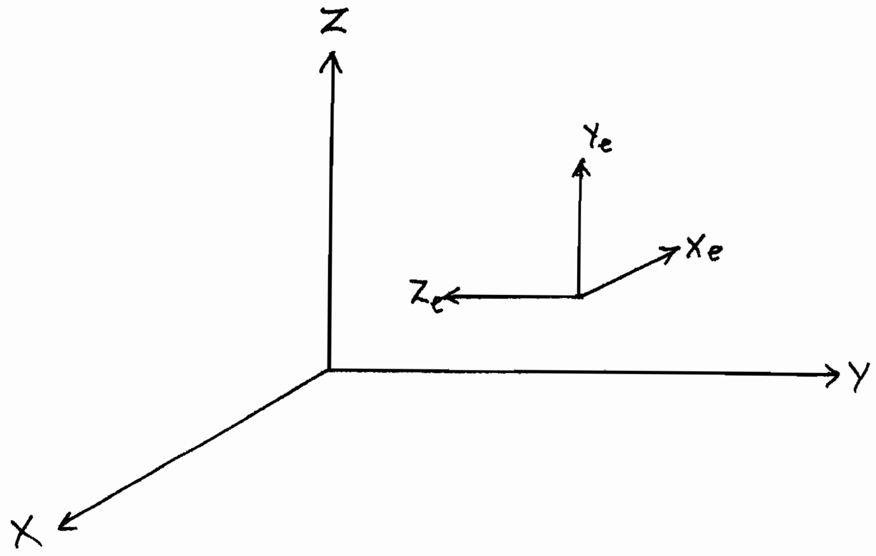


FIGURE 7

3. Rotation. Next we rotate about the Ye axis through an angle θ such that Z_e axis points toward the point (0,0,Z). The angle θ is then given by $\cos\theta = \frac{Y}{\sqrt{X^2 + Y^2}}$ and $\sin\theta = \frac{X}{\sqrt{X^2 + Y^2}}$. See Figure 8.

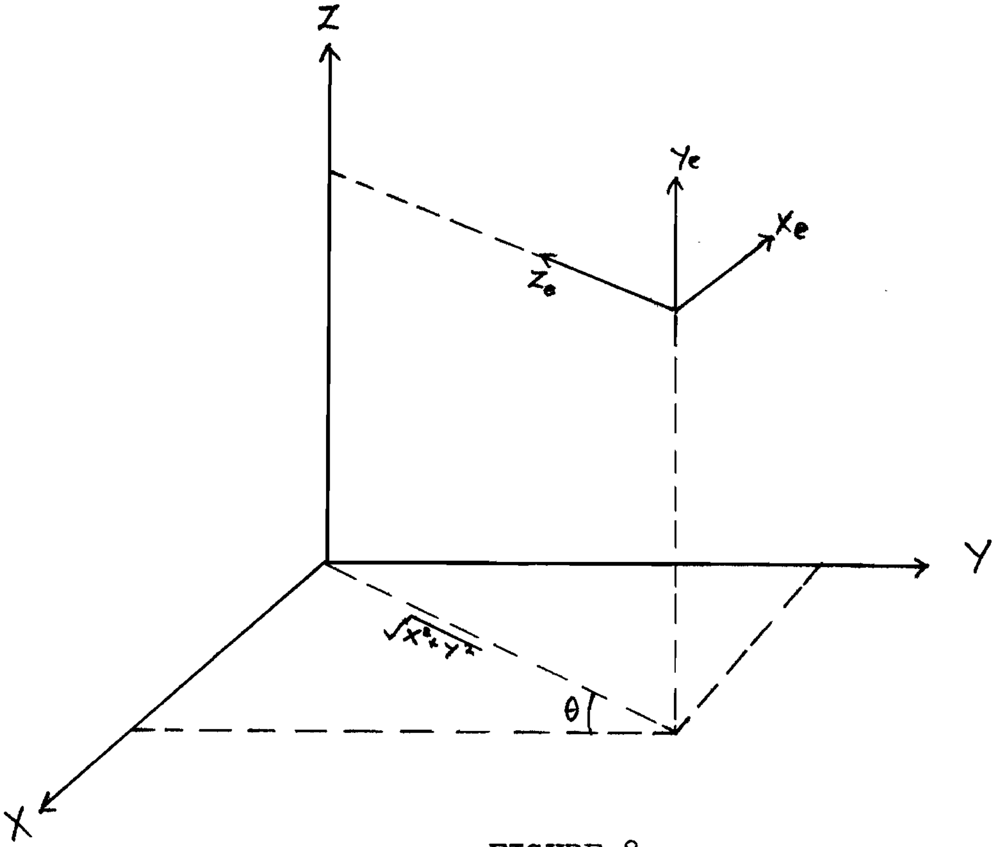


FIGURE 8

$$\text{Transformation Matrix } T_3 = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4. Rotation about X. Finally we rotate about the X_e axis so that the Ze axis now points at the origin in object space. This rotation will be through an angle $\emptyset$ given by

$$\text{Cos } \emptyset = \frac{X^2 + Y^2}{\sqrt{X^2 + Y^2 + Z^2}} \quad \text{and Sin } \emptyset = \frac{Z}{\sqrt{X^2 + Y^2 + Z^2}}$$

See Figure 9.

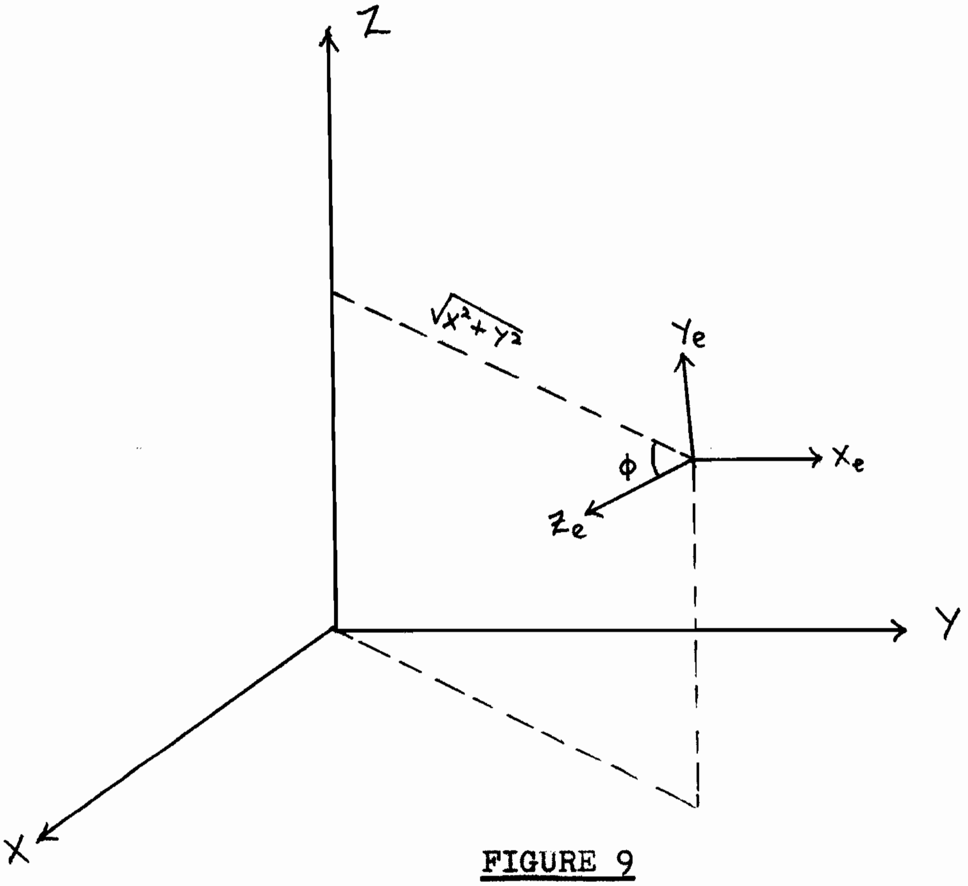


FIGURE 9

Transformation Matrix T_4 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos } \emptyset & -\text{Sin } \emptyset & 0 \\ 0 & \text{Sin } \emptyset & \text{Cos } \emptyset & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This completes the four simple transformations needed to establish the viewing transformation. Hence $V = T_1T_2T_3T_4$.

Given the view direction (X,Y,Z), the program will determine the view transformation V from the product of the four matrices. The coordinates of each atom generated in space is then multiplied by V to obtain (Xe, Ye, Ze) for that atom. The X_e and Y_e are now properly aligned with the X_s and Y_s of the display screen. The Ze values (with appropriate sign changes) are used to determine the distance of the planes from the observer. No perspective transformation is therefore involved in the display.

To go from Xe and Ye axes to screen coordinates we need some simple scaling. Let XMIN and XMAX represent minimum and maximum values of all Xe. YMIN and YMAX the minimum and maximum of all Ye generated. Further, suppose the coordinate system of the screen runs from 0 to 1023 and we wish to fill the screen, then the screen coordinates of each atom will be given by:

$$X_s = 0 + 1023* (X_e - Xmin) / (XMAX - XMIN)$$
$$Y_s = 0 + 1023* (Ye - Ymin) / (YMAX - YMIN)$$

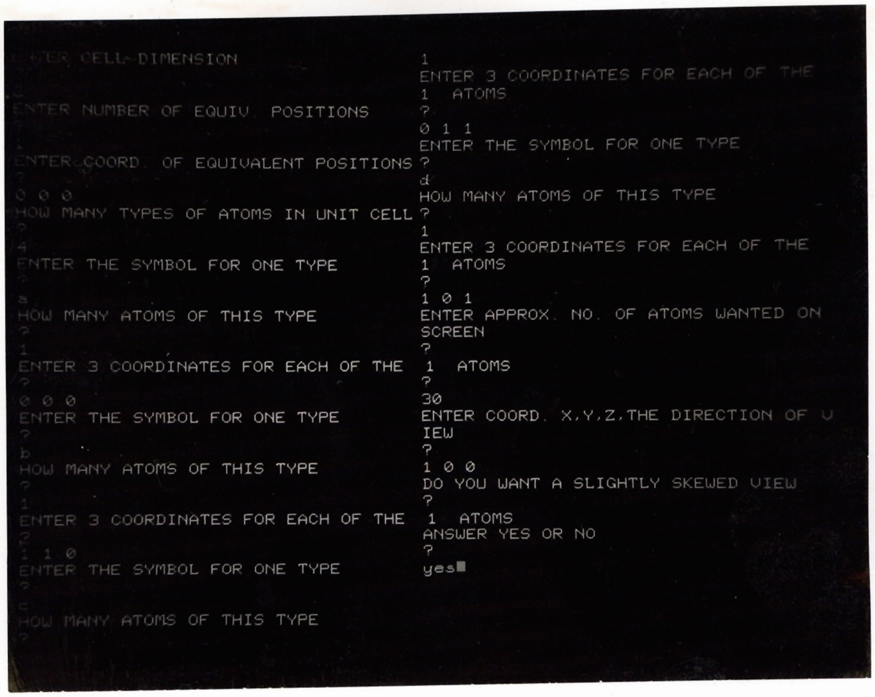
The three cases in which the observer wishes to view along any of the coordinate axes need special mention. The above transformation may not be meaningful in some of these cases. The program thus distinguishes between viewing along (X,0,0) , (0,Y,0) or (0,0,Z) and viewing along a general direction (X,Y,Z). Indeed, viewing along a coordinate axis does not require any transformations at all. As an example, to view along (X,0,0) direction, we merely reassign the axes such that $X_e = Y$; $Y_e = Z$ and $Z_e = X$.

Atom positions are thus projected in planes parallel to the Y - Z plane.

OUTPUT

The attached pictures show sample output from the display screen. For this illustration, a simple hypothetical F.C.C. (face centred cubic) structure with four atoms per unit cell was chosen. The crystallographic input data are as follows:-

Cell Dimension		=	2
Equiv. Positions		=	(0,0,0)
Centring Points:			
Atom	A	:	(0,0,0)
	B	:	(1,1,0)
	C	:	(0,1,1)
	D	:	(1,0,1)

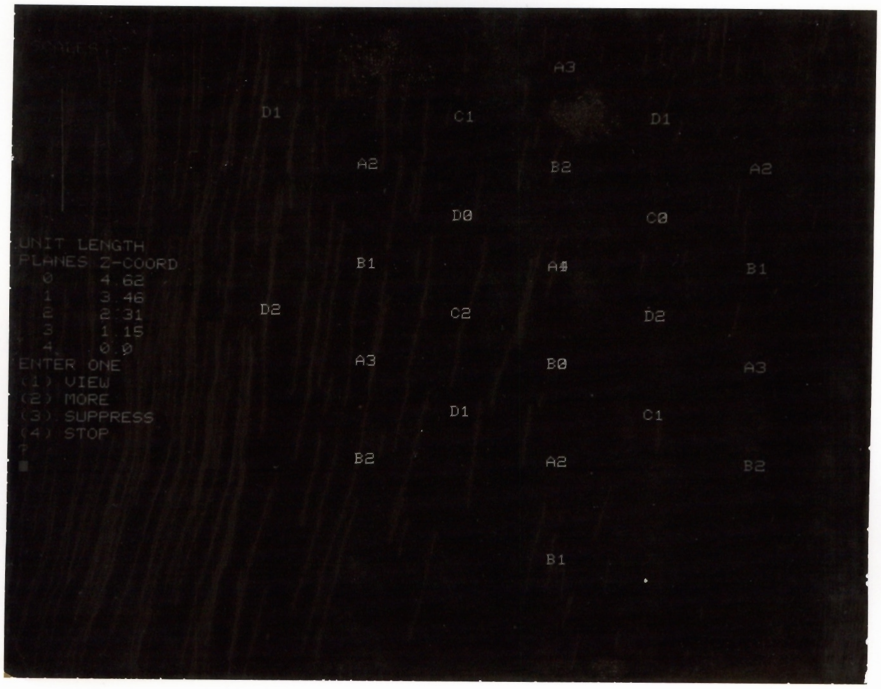


```
ENTER CELL DIMENSION
?
[illegible]
ENTER NUMBER OF EQUIV. POSITIONS
?
1
ENTER COORD. OF EQUIVALENT POSITIONS
?
0 0 0
HOW MANY TYPES OF ATOMS IN UNIT CELL
?
4
ENTER THE SYMBOL FOR ONE TYPE
?
a
HOW MANY ATOMS OF THIS TYPE
?
1
ENTER 3 COORDINATES FOR EACH OF THE
?
0 0 0
ENTER THE SYMBOL FOR ONE TYPE
?
b
HOW MANY ATOMS OF THIS TYPE
?
1
ENTER 3 COORDINATES FOR EACH OF THE
?
1 1 0
ENTER THE SYMBOL FOR ONE TYPE
?
c
HOW MANY ATOMS OF THIS TYPE
?
1
ENTER 3 COORDINATES FOR EACH OF THE
1  ATOMS
?
0 1 1
ENTER THE SYMBOL FOR ONE TYPE
?
d
HOW MANY ATOMS OF THIS TYPE
?
1
ENTER 3 COORDINATES FOR EACH OF THE
1  ATOMS
?
1 0 1
ENTER APPROX. NO. OF ATOMS WANTED ON
SCREEN
?
 1  ATOMS
?
30
ENTER COORD. X,Y,Z,THE DIRECTION OF V
IEW
?
1 0 0
DO YOU WANT A SLIGHTLY SKEWED VIEW
?
 1  ATOMS
ANSWER YES OR NO
?
yes
```

SAMPLE TERMINAL SESSION EXECUTING 'CUBIC'

INPUT DATA ARE BEING ENTERED IN A

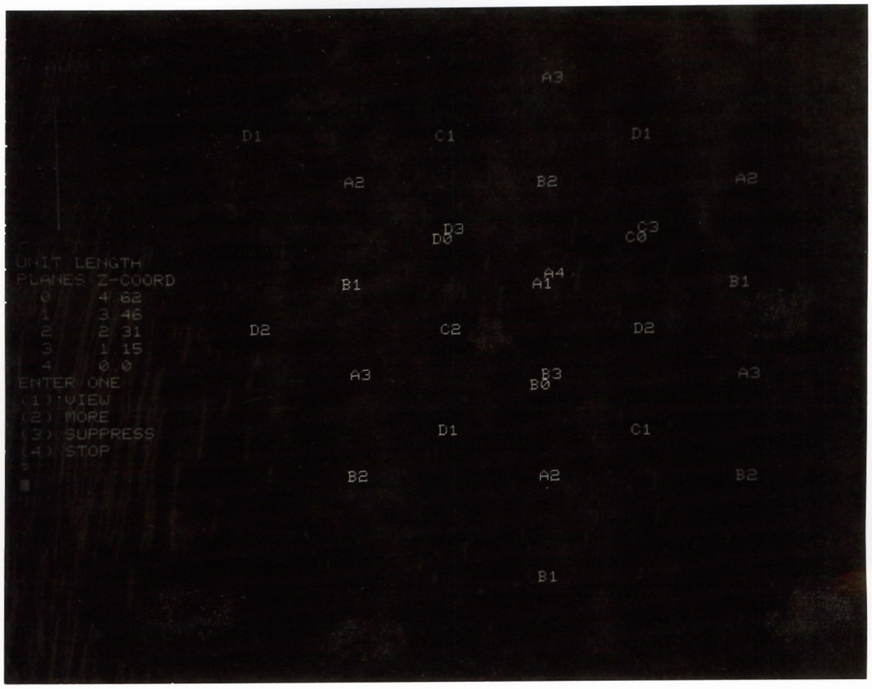
CONVERSATIONAL MODE.



```
UNIT LENGTH
PLANES Z-COORD
  Ø     4.62
  1     3.46
  2     2.31
  3     1.15
  4     Ø.Ø
ENTER ONE
(1) VIEW
(2) MORE
(3) SUPPRESS
(4) STOP
?
■
```

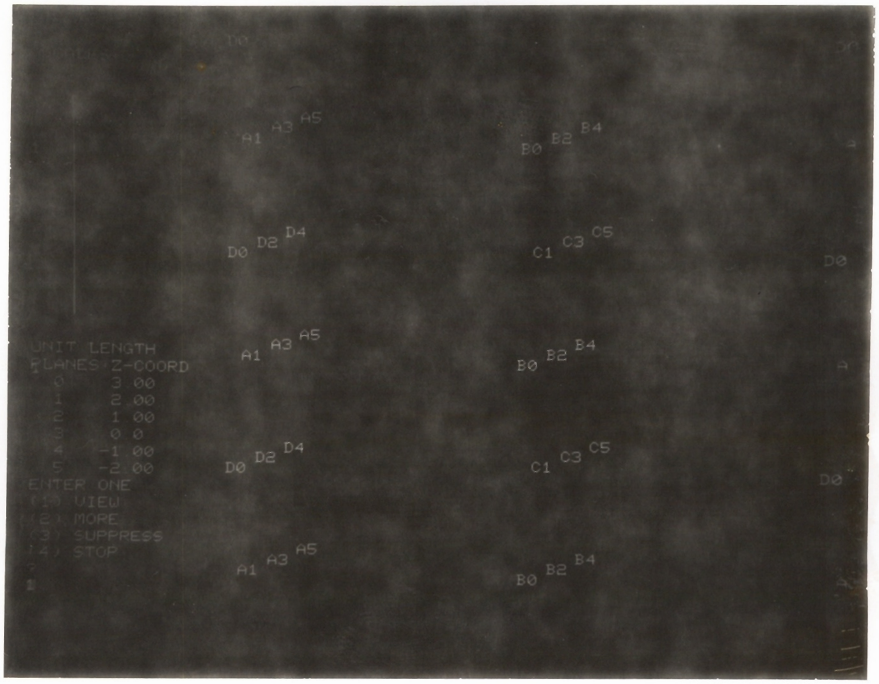
SAMPLE DISPLAY ON THE SCREEN

Structure:	Hypothetical F.C.C. structure.
	Four atoms per unit cell
	Atom : A (0,0,0) B (1,1,0) C (0,1,1) D (1,0,1)
View Direction:	Along diagonal (1,1,1)
Display Option:	No skew effect. Atoms project on top each other.
No. of Atoms Displayed:	30
Cell Dimension:	2



```
UNIT LENGTH
PLANES Z-COORD
  0     4.62
  1     3.46
  2     2.31
  3     1.15
  4     0.0
ENTER ONE
(1) VIEW
(2) MORE
(3) SUPPRESS
(4) STOP
?
■
```

Structure:	Hypothetical F.C.C. four atoms per unit cell. A : (0,0,0) B : (1,1,0) C : (0,1,1) D : (1,0,1)
View Direction:	Along diagonal (1,1,1)
No. of Atoms Displayed:	30
Cell Dimension:	2
Display Option:	SKEW



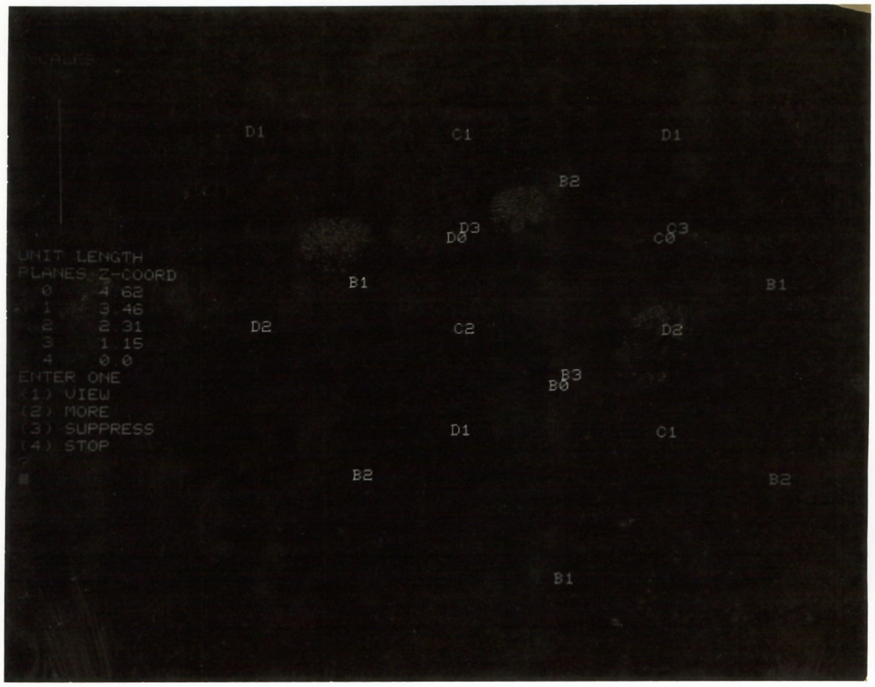
```
UNIT LENGTH
PLANES Z-COORD
  0     3.00
  1     2.00
  2     1.00
  3     0.0
  4    -1.00
  5    -2.00
ENTER ONE
(1) VIEW
(2) MORE
(3) SUPPRESS
(4) STOP
?
1
```

Structure: Simple F.C.C.

View Direction: Along X-axis (1,0,0)

Display Option: MORE and SKEWED

No. of Atoms Displayed: 40



```
SCALES

UNIT LENGTH
PLANES Z-COORD
  0     4.62
  1     3.46
  2     2.31
  3     1.15
  4     0.0
ENTER ONE
(1) VIEW
(2) MORE
(3) SUPPRESS
(4) STOP
?
```

Structure: Simple F.C.C.

View Direction: Along Diagonal (1,1,1)

Display Option: Suppress. Type 'A' atoms are dropped in the display.

No. of Atoms Displayed: 30

Cell Dimension: 2

OPERATING INSTRUCTIONS FOR USER

Signing On: The program 'CUBIC' is run under MUSIC and signing on with the Tektronix is the same as signing on with an IBM 2741 typewriter terminal. After the system types *MUSIC sign on, the following dialogue should ensue:-

```
USER RESPONSE:      / ID  tn,uuuu
                    Where tn is terminal identification (any
                    two decimal digits e.g. 23 will suffice),
                    uuuu is the user's MUSIC code.
SYSTEM RESPONSE:    *PASSWORD ?
                    ■■■■■
USER RESPONSE:      Types in his password on the blacked out
                    squares above.
SYSTEM RESPONSE:    * IN PROGRESS
                    * CODE LAST USED TIME....DATE, MONTH, YR
                    * SIGN ON DAY MONTH DATE, YR,......RESTART=
                    * GO
```

The system is now ready to accept commands to execute the program.

```
USER RESPONSE:      / INPUT
                    / INCLUDE CUBIC
                    / INCLUDE TEKØ12          TEK012 not [illegible]
                    / ENDRUN
SYSTEM RESPONSE:    IN PROGRESS
```

The system will then indicate that execution has begun by displaying the load time, quickly clears the screen and the following dialogue should ensue:-

(1) SYSTEM RESPONSE: ENTER CELL DIMENSION

?

USER RESPONSE: Types in a numeric value to represent unit cell length. This length could be in any units.

SYSTEM RESPONSE: ENTER NUMBER OF EQUIV. POSITIONS

?

USER RESPONSE: Types in an integer value representing number of equivalent positions in the structure.

SYSTEM RESPONSE: ENTER COORDINATES OF EQUIV. POSITIONS

?

USER RESPONSE: Types in 3 coordinates (X, Y, Z numeric values) for each of the above equiv. positions. May type all data in one single line but separating data items by blank space.

SYSTEM RESPONSE: HOW MANY TYPES OF ATOMS IN UNIT CELL

?

USER RESPONSE: Types in an integer value for number of different atom types.

SYSTEM RESPONSE: * ENTER THE SYMBOL FOR ONE TYPE

?

USER RESPONSE: Types in one character symbol for one atom type.

SYSTEM RESPONSE: HOW MANY ATOMS OF THIS TYPE

?

USER RESPONSE: Types in an <u>integer</u> value n_t indicating how many centring atoms for the above type.

SYSTEM RESPONSE: * ENTER 3 COORD. FOR EACH OF THE n_t ATOMS
?

USER RESPONSE: Types in <u>numeric</u> values for the X, Y, Z coords. Separate items by blank space.

(2) SYSTEM RESPONSE: ENTER APPROX. NUMBER OF ATOMS WANTED ON SCREEN
?

USER RESPONSE: Types in an <u>integer</u> value. NB: The larger the number specified, the more crowded the screen will be.

(3) SYSTEM RESPONSE: ENTER COORD. X, Y, Z. THE DIRECTION OF VIEW
?

USER RESPONSE: Types in 3 coordinates specifying the eye position and direction of view.

SYSTEM RESPONSE: DO YOU WANT A SLIGHTLY SKEWED VIEW
?

USER RESPONSE: Types in "YES" or "NO"

SYSTEM RESPONSE: Clears the screen and displays the atom positions. The bottom left corner of the screen will request the user to type in one of the following four options
VIEW, MORE, SUPPRESS, STOP.

<u>VIEW</u>

USER RESPONSE: Types in 'VIEW'

SYSTEM RESPONSE: Loops back to (3). Displays the same structure seen from a different view.

MORE

USER RESPONSE: Types in 'MORE'

SYSTEM RESPONSE: Loops back to (2)

SUPPRESS

USER RESPONSE: Types in 'SUPPRESS'

SYSTEM RESPONSE: ENTER SYMBOL TO BE SUPPRESSED

?

USER RESPONSE: Types in one character symbol representing atom type to be dropped from the display.

SYSTEM RESPONSE: Loops back to display atom positions.

STOP

USER RESPONSE: Types in 'STOP'

SYSTEM RESPONSE: DO YOU WISH TO VIEW ANOTHER LATTICE STRUCTURE

?

USER RESPONSE: Types in 'YES' or 'NO'

SYSTEM RESPONSE: If 'YES', loops back to (1)

If 'NO', execution stops.

NB Dialogue between two asterisks '*' will be repeated as necessary.

SIGN OFF: / OFF

```
d cubic
*IN PROGRESS
/LOAD FORTG
        DIMENSION EQPOS(10,3),ATOM(500,3),EYE(500,3),ZCOD(10)
        DIMENSION UCEL(100,3),CNPOS(10,3),CUBE(3,3),SCREEN(2),VM(4,4)
        INTEGER SYMBOL(10),MESS(5,3),NAME(4,2),KOMD(2),MSUM(10)
        INTEGER KP,KZ,NPRNT,ANSW(2),TYPE(500)
        DATA ANSW/'YE','NO'/
        DATA BLANK/'    '/
        DATA CUBE/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
        DATA SCREEN/1024.0,780.0/
        DATA NAME/'VIEW','MORE','SUPP','STOP',
       1          '    ','    ','RESS','    '/
        DATA MESS/'ENTE','(1) ','(2) ','(3) ','(4) ',
       1          'R ON','VIEW','MORE','SUPP','STOP',
       1          'E   ','    ','    ','RESS','    '/
C
C     THIS PROGRAM USES GRAPHIC ROUTINES TO PROJECT ATOM POSITIONS FROM A CRYSTAL
C     STRUCTURE ONTO A TEKTRONIX SCREEN. NO PERSPECTIVE PROJECTIONS ARE INVOLVED.
C     ARRAY DECLARATIONS ASSUME A MAX. OF 500 ATOMS DURING EACH DISPLAY. THE
C     STRUCTUERS TO BE VIEWED ARE ASSUMED TO HAVE LESS THAN 100 ATOMS PER
C     UNIT CELL,AND A MAX. OF 10 DIFFERENT TYPES OF ATOMS. ONLY STRUCTURES WITH
C     CUBIC SYMMETRY NEED BE EXAMINED USING THIS PROGRAM.
C     THE ARRAY 'CNPOS' CONTAINS COORDINATES OF CENTRING ATOMS FOR EACH TYPE.
C     THE ARRAY 'EQPOS' CONTAINS COORDINATES OF EQUIV. POSITIONS.
C     THE ARRAY 'UCEL' CONTAINS TOTAL NUMBER OF ATOMS PER UNIT CELL.
C     THE ARRAY  'ATOM' CONTAINS COORDINATES OF GENERATED SPACE LATTICE ATOMS.
C     THE ARRAY 'EYE' CONTAINS COORD. OF ATOMS AFTER TRANSFORMATION INTO
C             EYE COORDINATE SYSTEM.
C     THE ARRAY 'TYPE' STORES SYMBOLS FOR EACH ATOM POSITION FOUND IN EYE.
C     THE ARRAY 'ZCOD' STORES COORDS. (Z EYE COORDINATES) OF PLANES OF LATTICE
C             ATOMS.THE LATTICE PLANES ARE NORMAL TO DIRECTION OF VIEW
C
C
2       CALL TEK001(0,0,0)
        CALL TEK001(0,766,16)
        WRITE(6,9)
9       FORMAT(' ENTER CELL DIMENSION')
        READ(9,*) A
        WRITE(6,7)
7       FORMAT(' ENTER NUMBER OF EQUIV. POSITIONS')
        READ(9,*) N
        WRITE(6,10)
10      FORMAT(' ENTER  COORDINTES OF EQUIV. POSITONS')
        READ(9,*) ((EQPOS(I,J),J = 1,3),I = 1,N)
        WRITE(6,12)
12      FORMAT(' HOW MANY ATOM TYPES IN UNIT CELL')
        READ(9,*) M
        L = 1
        SUMA = 0.0
        DO 20 I = 1,M
        WRITE(6,15)
15      FORMAT(' ENTER SYMBOL FOR ONE TYPE')
        READ(9,17) SYMBOL(I)
17      FORMAT(A1)
        WRITE(6,18)
18      FORMAT(' HOW MANY ATOMS OF THIS TYPE')
        READ(9,*) NT
        WRITE(6,19) NT
19      FORMAT('ENTER ALL COORDS. FOR THE',I3,2X,'ATOMS')
        READ(9,*) ((CNPOS(J,K),K = 1,3),J = 1,NT)
        MSUM(I) = N*NT
C      ADD UP TOTAL NUMBER OF ATOMS IN UNIT CELL
        SUMA = SUMA + MSUM(I)
        DO 24 NN = 1,NT
        DO 23 J = 1,N
        DO 22 K = 1,3
22      UCEL(L,K)=CNPOS(NN,K) + EQPOS(J,K)
23      L = L + 1
24      CONTINUE
20      CONTINUE
30      WRITE(6,5)
5       FORMAT(' ENTER APPROX. NO. OF ATOMS WANTED ON SCREEN')
        READ(9,*) NA
C      TOTAL SPACE ATOMS TO BE GENERATED IS ISQ*SUMA. ISQ IS THE CUBE OF THE
C           CEILING OF FU (FRACTION OF UNIT CELL).
C      COMPUTE FRACTION F = FU/ISQ REQUIRED TO GIVE NA ATOMS ON THE SCREEN.
C
        FU = NA/SUMA
        IW = FU**(1.0/3) + 1
        ISQ = IW**3
        F = FU/ISQ
        IND = (IW - 1)/2 + 1
26      WRITE(6,25)
25      FORMAT(' ENTER COORD. X,Y,Z, THE DIRECTION OF VIEW')
        READ(9,*) X,Y,Z
        WRITE(6,27)
27      FORMAT(' DO YOU WANT A SKEWED VIEW')
28      READ(9,340) KZ
        IF(KZ.EQ.ANSW(1).OR.KZ.EQ.ANSW(2)) GO TO 29
        WRITE(6,350)
        GO TO 28
C
C      CHECK FOR VIEW DIRECTIONS ALONG THE COORD. AXES.
C
29      IF((X.EQ.0.AND.Y.EQ.0).OR.(X.EQ.0.AND.Z.EQ.0).OR.
       1(Y.EQ.0.AND.Z.EQ.0)) GO TO 50
C
C      OTHERWISE COMPUTE VIEW TRANSFORMATION MATRIX FOR A GENERAL VIEW DIRECTION.
C
        ASQ = SQRT(X**2 + Y**2)
        BSQ = SQRT(X**2 + Y**2 + Z**2)
        VM(1,1) = -2*Y/ASQ
        VM(2,1) = 2*X/ASQ
        VM(3,1) = 0.0
        VM(4,1) = 0.0
        VM(1,2) = -2*X*Z/(ASQ*BSQ)
        VM(2,2) = -2*Y*Z/(ASQ*BSQ)
        VM(3,2) = 2*ASQ/BSQ
        VM(4,2) = 0.0
        VM(1,3) = -X/BSQ
        VM(2,3) = -Y/BSQ
        VM(3,3) = -Z/BSQ
        VM(4,3) = -BSQ
        VM(1,4) = 0.0
        VM(2,4) = 0.0
        VM(3,4) = 0.0
        VM(4,4) = 1.0
C
C      GENERATE OTHER ATOM POSITIONS IN 3-SPACE BY TRASLATION, A UNITS ALONG
C      THE COORDINATE AXES U=(1,0,0),V=(0,1,0),W=(0,0,1)
C
50      L = 1
        LN = 0
        NN = 1
        DO 100 II = 1,M
        LN = LN + MSUM(II)
        DO 90 JJ = NN,LN
        DO 80 I = 1,IW
        DO 70 J = 1,IW
        DO 60 K = 1,IW
        DO 55 KL = 1,3
55      ATOM(L,KL) = UCEL(JJ,KL) +A*(I-IND)*CUBE(1,KL) + A*
       1(J-IND)*CUBE(2,KL) +A*(K-IND)*CUBE(3,KL)
        TYPE(L) = SYMBOL(II)
60      L = L + 1
70      CONTINUE
80      CONTINUE
90      CONTINUE
        NN = NN + MSUM(II)
C
C      ARRAY MSUM CONTAINS NUMBER OF ATOMS IN UNIT CELL FROM EACH TYPE PRESENT.
C
100     CONTINUE
C
C      CLEAR THE SCREEN AND GO TO GRAPHIC MODE. CALL GRAPHIC ROUTINES.
C
        NUM = L - 1
        CALL TEK001(0,0,0)
        CALL TEK001(0,766,16)
        XMIN = 1000.0
        XMAX = -999.0
        YMIN = 780.0
        YMAX = -999.0
        IF(X.EQ.0.AND.Y.EQ.0) GO TO 110
        IF(X.EQ.0.AND.Z.EQ.0) GO TO 120
        IF(Y.EQ.0.AND.Z.EQ.0) GO TO 130
C
C      DO A MATRIX MULTIPLICATION TO TRANSFORM EACH ATOM COORD. INTO EYE COORD.
C
        DO 300 I = 1,NUM
        DO 290 K = 1,3
        SUM = 0.0
        DO 280 J = 1,3
        SUM = SUM + ATOM(I,J)*VM(J,K)
280     CONTINUE
290     EYE(I,K) = SUM
300     CONTINUE
C
C      CHANGE THE SIGNS ON Z AXIS FOR CONSISTENCY WITH DRECTION OF VIEW
C
        DO 170 I = 1,NUM
170     EYE(I,3) = -EYE(I,3)
        CALL BBSORT(EYE,TYPE,NUM)
        CALL SORT(XMIN,YMIN,XMAX,YMAX,EYE,NUM)
        GO TO 310
C
C      VIEW DIRECTION IS ALONG THE Z AXIS. NO REARRANGEMENT OF AXES IS NECESSARY.
C
110     DO 115 I = 1,NUM
        EYE(I,1) = ATOM(I,1)
        EYE(I,2) = ATOM(I,2)
115     EYE(I,3) = ATOM(I,3)
        CALL BBSORT(EYE,TYPE,NUM)
        CALL SORT(XMIN,YMIN,XMAX,YMAX,EYE,NUM)
        GO TO 310
C
C      VIEW DIRECTION IS ALONG THE Y-AXIS. REARRANGE THE AXES SUCH THAT
C      Y-AXIS(OBJECT SPACE) CORRESPONDS WITH Z-AXIS (EYE COORD.).
C
120     DO 200 I = 1,NUM
        EYE(I,1) = ATOM(I,1)
        EYE(I,2) = ATOM(I,3)
200     EYE(I,3) = ATOM(I,2)
        CALL BBSORT(EYE,TYPE,NUM)
        CALL SORT(XMIN,YMIN,XMAX,YMAX,EYE,NUM)
        GO TO 310
C
C      VIEW DIRECTION IS ALOG X-AXIS. REARRANGE AXES SUCH THATX-AXIS CORRESPONDS
C      WITH Z-AXIS(EYE COORD SYSTEM).
C
130     DO 220 I = 1,NUM
        EYE(I,1) = ATOM(I,2)
        EYE(I,2) = ATOM(I,3)
220     EYE(I,3) = ATOM(I,1)
        CALL BBSORT(EYE,TYPE,NUM)
        CALL SORT(XMIN,YMIN,XMAX,YMAX,EYE,NUM)
C
C      CHECK WHETHER SKEWED VIEW IS ASKED FOR.VARIABLE KZ CONTAINS 'YE' OR 'NO'.
C
310     IF(KZ.EQ.ANSW(2)) GO TO 307
        DO 305 I = 1,NUM
        EYE(I,1) = EYE(I,1) - 0.05*EYE(I,3)
305     EYE(I,2) = EYE(I,2) - 0.05*EYE(I,3)
307     KOUNT = 0
C      COMPUTE UNIT CELL LENGTH IN SCREEN COORDINATES.
        LENT = 700*A/(YMAX-YMIN)
C
C      SELECT THE APPRIOPRIATE FRACTION OF TOTAL ATOMS GENERATED SUCH THAT THIS
C      FRACTION WILL GIVE APPROX. NA ATOMS ON THE SCREEN.SCALE TO FILL SCREEN.
        TEMP = EYE(1,3)
        ZCOD(1) = TEMP
        XMAX = XMIN +F*(XMAX - XMIN)
        DO 440 I = 1,NUM
        IX = 300 + (EYE(I,1) - XMIN)*700/(XMAX - XMIN)
        IY = 100 + (EYE(I,2) - YMIN)*700/(YMAX - YMIN)
        IF(IX.GT.SCREEN(1).OR.IY.GT.SCREEN(2)) GO TO 440
        IF(ABS(TEMP-EYE(I,3)).LE.0.005) GO TO 303
        TEMP = EYE(I,3)
        KOUNT = KOUNT + 1
        ZCOD(KOUNT+1) = TEMP
C
C      CHECK ATOM TYPE TO BE SUPPRESSED.
C
 303    IF(TYPE(I).EQ.NPRNT) GO TO 440
        CALL TEK001(IX,IY,4)
        CALL TEK001(IX,IY,16)
        IF(KOUNT.GT.9) GO TO 304
        WRITE(6,301) TYPE(I),KOUNT
        GO TO 440
304     KOUNT = KOUNT - 1
        WRITE(6,308) TYPE(I)
308     FORMAT(1X,A1)
301     FORMAT(1X,A1,I1)
440     CONTINUE
        NPRNT = BLANK
        KK = KOUNT + 1
        CALL TEK001(0,766,16)
        WRITE(6,410)
        CALL TEK001(50,700,4)
        CALL TEK001(50,700-LENT,12)
        CALL TEK001(0,700-LENT-50,16)
        WRITE(6,420)
        WRITE(6,430)
410     FORMAT(2X,'SCALES')
420     FORMAT('UNIT LENGTH')
430     FORMAT('PLANES:Z-COORD')
        DO 460 I = 1,KK
        I1 = I - 1
        WRITE(6,450) I1,ZCOD(I)
450     FORMAT(3X,I1,3X,F5.2)
460     CONTINUE
        WRITE(6,315) (MESS(1,J),J = 1,3)
        WRITE(6,315) (MESS(2,J),J = 1,3)
        WRITE(6,315) (MESS(3,J),J = 1,3)
        WRITE(6,315) (MESS(4,J),J = 1,3)
        WRITE(6,315) (MESS(5,J),J = 1,3)
315     FORMAT(3(A4))
320     READ(9,180) (KOMD(J),J = 1,2)
180     FORMAT(2(A4))
        DO 190 I = 1,4
        DO 185 J = 1,2
        IF(NAME(I,J).NE.KOMD(J)) GO TO 190
185     CONTINUE
        GO TO (26,30,390,400),I
190     CONTINUE
        WRITE(6,165) (KOMD(K),K = 1,2)
165     FORMAT(' INVALID COMMAND',2X,2(A4),2X,'TRY AGAIN')
        GO TO 320
390     WRITE(6,395)
395     FORMAT('ENTER THE SYMBOL YOU WISH TO SUPPRESS')
        READ(9,17)NPRNT
        CALL TEK001(0,0,0)
        GO TO 307
400     WRITE(6,330)
330     FORMAT(' DO YOU WISH TO VIEW A DIFFERENT STRUCTURE')
335     READ(9,340) KP
        IF(KP.EQ.ANSW(1)) GO TO 2
        IF(KP.EQ.ANSW(2)) GO TO 500
        WRITE(6,350)
        GO TO 335
340     FORMAT(A2)
350     FORMAT(' ANSWER YES OR NO')
500     STOP
        END
        SUBROUTINE SORT(XMIN,YMIN,XMAX,YMAX,TEMP,N)
C
C      DETERMINE THE MINIMUM AND MAXIMUM VALUES IN THE X AND Y AXES TO PLOTTED.
C
        REAL XMIN,YMIN,XMAX,YMAX,TEMP(500,3)
        DO 60 I = 1,N
        IF(TEMP(I,1).GT.XMIN) GO TO 30
        XMIN = TEMP(I,1)
30      IF(TEMP(I,2).GT.YMIN) GO TO 40
        YMIN = TEMP(I,2)
40      IF(TEMP(I,1).LE.XMAX) GO TO 50
        XMAX = TEMP(I,1)
50      IF(TEMP(I,2).LE.YMAX) GO TO 60
        YMAX = TEMP(I,2)
60      CONTINUE
        RETURN
        END
        SUBROUTINE BBSORT(A,LOC,N)
C
C      SORT THE Z-COORDINATES IN DESCENDING ORDER
C
        DIMENSION A(500,3)
        INTEGER ITEM,LOC(500)
        LOGICAL FLAG
        N1 = N - 1
        N2 = N1
        DO 50 I = 1,N1
        FLAG = .FALSE.
        DO 40 J = 1,N2
        IF(A(J,3).GE.A(J+1,3)) GO TO 40
        TEMP1 = A(J,1)
        TEMP2 = A(J,2)
        TEMP3 = A(J,3)
        ITEM = LOC(J)
        A(J,1) = A(J+1,1)
        A(J,2) = A(J+1,2)
        A(J,3) = A(J+1,3)
        LOC(J) = LOC(J+1)
        A(J+1,1) = TEMP1
        A(J+1,2) = TEMP2
        A(J+1,3) = TEMP3
        LOC(J+1) = ITEM
        FLAG = .TRUE.
40      CONTINUE
        IF(.NOT.FLAG) GO TO 60
50      N2 = N2 - 1
60      RETURN
        END
*END

*GO
```