ornl

OAK RIDGE NATIONAL LABORATORY

LOCKHEED MARTIN

ORNL-6895

ORTEP-III: Oak Ridge Thermal Ellipsoid Plot Program for Crystal Structure Illustrations

Michael N. Burnett
Carroll K. Johnson

MANAGED AND OPERATED BY
LOCKHEED MARTIN ENERGY RESEARCH CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL-27 (3-96)

ORNL-6895

Chemical and Analytical Sciences Division

ORTEP-III: OAK RIDGE THERMAL ELLIPSOID PLOT PROGRAM FOR CRYSTAL STRUCTURE ILLUSTRATIONS

Michael N. Burnett
Carroll K. Johnson

Date Published: July 1996

Research sponsored by the
Laboratory Directed Research and Development Program

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6285
managed by
LOCKHEED MARTIN ENERGY RESEARCH CORP.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-96OR22464

CONTENTS

ACKNOWLEDGEMENTS

From previous versions of this report:[1,2]

We are particularly indebted to our colleagues, Drs. H. A. Levy, W. R. Busing, G. M. Brown, and R. D. Ellison for many helpful discussions and to R. A. Hollister, a summer participant with the ORNL Mathematics Division, who helped plan and code several parts of the initial 1965 release of the program. The initial draft version of ORTEP was written as a subroutine for the Busing, Martin, and Levy Function and Error Program, OR FFE; and many of the concepts and several of the subroutines of OR FFE are incorporated into the present program. Several parts of EIGEN were taken from a program written by R. E. Funderlic and B. Franz from the Central Data Processing group. Subroutine AXEQB was adapted from a subroutine obtained from the Oak Ridge Central Data Processing Library.

Additional acknowledgements:

We wish to thank Oak Ridge National Laboratory for providing computer resources for our ORTEP-III (and related crystallographic topology) World Wide Web pages at

http://www.ornl.gov/ortep/ortep.html

We especially thank Dr. Martin Kroeker for setting up a European World Wide Web site to mirror Oak Ridge's ORTEP-III web site at

http://tutor.oc.chemie.th-darmstadt.de/Ortep3/ortep.html

We thank all the people who have used ORTEP over the years and who have modified the program and made it available for others to use, including those listed at

http://www.ornl.gov/ortep/kin.html

A number of people assisted with the testing of ORTEP-III before the program was released to the public. It is impossible to list everyone here, but we appreciate all their efforts. We particularly thank: Dr. John Bollinger, Karl A. Byriel, Bjorn Dalhus, Dr. Bill Harrison, Dr. John C. Huffman, Prof. Gerald G. Johnson, Jr., Dr. Martin Kroeker, Dr. Anthony Linden, Dr. Wolfgang Poll, Dr. James V. Silverton, and Dr. Beverly R. Vincent.

We also thank Dr. Tim Pearson at the California Institute of Technology for the free graphics package PGPLOT, which enabled us to create one version of ORTEP-III that would produce screen graphics on a wide range of computer hardware. Thanks to Christian T. Dum and John S. Salmento for assistance with their ports of PGPLOT for DOS and Macintosh personal computers, respectively.

Thanks go to Kate Crennell for producing an executable version of ORTEP-III for Acorn microcomputers running the RISC OS operating system, which can be found at

http://micros.hensa.ac.uk/micros/arch-riscos.html

and again to Dr. Kroeker for making a PC LINUX version available at

ftp://tutor.oc.chemie.th-darmstadt.de/pub/ortep/linux/

ABSTRACT

This report describes a computer program for drawing crystal structure illustrations. Ball-and-stick type illustrations of a quality suitable for publication are produced with either spheres or thermal-motion probability ellipsoids on the atomic sites. The program can also produce stereoscopic pairs of illustrations which aid in the visualization of complex packing arrangements of atoms and thermal motion patterns. Interatomic distances, bond angles, and principal axes of thermal motion are also calculated to aid the structural study.

1. INTRODUCTION

1.1 WHAT IS ORTEP?

Appropriate illustrations are essential in any manuscript dealing with crystallographic structures. An often quoted expression might justifiably be paraphrased to read that a well-planned figure is worth a thousand numbers. With the information explosion in the scientific literature, the author of a structure paper is obligated to provide the reader with "crystal clear" illustrations.

The Oak Ridge Thermal Ellipsoid Plot (ORTEP) program is a tool for drawing certain types of crystal structure illustrations. The program and the precision obtainable through machine plotting make feasible the production of detailed stereoscopic illustrations that are impractical to draw by conventional drafting methods. Several types of illustrations may be drawn with ORTEP. For a standard thermal motion drawing, an ellipsoid positioned on an atomic site represents a 3-D Gaussian probability density function showing the averaged atomic displacement as derived through the anisotropic temperature factor parameters for that atom. In critical net drawings, a very elongated or flattened ellipsoid of revolution, not positioned on an atomic center, represents the orientation of a saddle-point type critical point of the global density function. For simpler drawings of a crystal structure, all atoms are represented as spheres with sphere radii, or some other graphical variable, used to depict the chemical type of the atom. ORTEP cannot make van der Waal's type drawings, which require overlapping spheres or ellipsoids.

Since its inception, four major goals have driven the development of ORTEP. These are listed here in decreasing order of their assigned importance. (1) The program must produce high quality illustrations, including stereoscopic pairs of thermal-motion figures, as free as possible of visually distracting approximations. (2) The program must be general both with respect to the types of illustrations it can draw and the types of computing and plotting equipment that it can utilize. (3) The program must be easy to use, require a minimum of input, and be easy to modify. (4) The computation time should be minimized. Since generality is placed higher in this goal list than ease of use, the program originally designed in the 1960s lacks several of the user-friendly attributes of the 1990s. However, the program seems to have survived the test of time better than some of its more user-friendly competitors. There were over 1000 citations of the 1965[1] and 1976[2] versions of the program in the 1995 Science Citation Index.[3] Since ORTEP has become widely used, we decided not to make changes that would render existing ORTEP input data sets inoperative.

ORTEP-III is written in device-independent FORTRAN, and the code should compile and run on any computer system that has a FORTRAN compiler. Compiled versions of ORTEP-III are available for DOS compatible and Macintosh† compatible personal computers. ORTEP-III is available on the World Wide Web at

http://www.ornl.gov/ortep/ortep.html

or via anonymous ftp at ftp://ftp.ornl.gov/pub/ortep. Questions, comments, problems, suggestions, etc. may be sent via electronic mail to ortep@ornl.gov or via regular mail to either author of this report at P.O. Box 2008, Oak Ridge, TN 37831-6197, USA.

---

†Macintosh is a registered trademark of Apple Computer, Inc.

1.2 WHAT'S NEW IN ORTEP-III

For compatibility with existing ORTEP data sets, ORTEP-III retains ALL the functionality of OR TEP,[1] released in 1965, and OR TEP-II,[2] released in 1976. A brief description of the new capabilities in ORTEP-III is provided here. See the indicated sections of this report for more detailed information.

1.2.1 User Interface

ORTEP-III is a semi-interactive program that requests certain information from the user while the program executes. A default value for each item requested is provided in square brackets, and it will be used if the user simply hits the RETURN key (on some keyboards, the name of this key is ENTER). On systems that distinguish upper and lower case, care must be taken when entering file names to provide the correct case of the letters. (See 4.1.)

1.2.2 Screen Display

The screen drawing subroutines available in ORTEP-III use PGPLOT. PGPLOT is a free graphics library developed by T. J. Pearson at the California Institute of Technology. The package is written in FORTRAN and operates on a variety of platforms. Information about PGPLOT can be found on the World Wide Web at http://astro.caltech.edu/~tjp/pgplot or via e-mail to tjp@astro.caltech.edu. PGPLOT is **not** required to run ORTEP-III. The compiled versions of ORTEP-III for personal computers include the screen drawing capability. (See 4.2.)

1.2.3 Output Formats

Illustrations generated by ORTEP-III can be saved as Encapsulated Postscript™ or Hewlett-Packard Graphics Language (HPGL/2)™ files.† The files may be printed directly on a wide variety of printers and plotters or may be imported into a number of computer programs that accept these formats. (See 4.3.)

1.2.4 Color

By default, ORTEP-III plots its illustrations in black on a white background. The new 204 instruction allows color to be added to the illustrations if the output device supports color. Once a color is set, it remains in effect until another 204 instruction changes the color. A 204 with no parameter (or a "0") returns the plotting color to black (or pen #1). The screen and Postscript drivers built into ORTEP-III define color value 2 as red, 3 as green, 4 as blue, 5 as cyan, 6 as magenta, and 7 as yellow. (See 3.3.3.)

1.2.5 Interactive Editor

ORTEP-III provides a simple line editor for editing the input file without exiting the program. When the editor is invoked, the instruction set from the input file is displayed on the screen with line numbers (#) along with the editor commands shown below. (See 4.4.)

---

†Encapsulated Postscript is a registered trademark of Adobe Systems, Inc. Hewlett-Packard Graphics Language Version 2 (HPGL/2) is a registered trademark of Hewlett-Packard Corporation.

```
C=Change line #                    D=Delete line(s) # [#]
I=Insert line before #             T=Type line(s) [#] [#]
S=Save modified instruction set    O=Restore original instruction set
P=Save drawing as Postscript       H=Save drawing as HPGL
R=Redraw structure on screen       Q=Quit
```

1.2.6 Comments

In previous versions of ORTEP, Format No. 3 trailer cards (see 3.3.1) were the only means of placing comments in the input file. This method still works in ORTEP-III, but a new method also exists. Among the ORTEP instructions, any line beginning with # is treated as a comment and is totally ignored by the program. Such comments can only go in the instruction portion of the input data and only at places where a new instruction could begin, i.e., comments cannot go between instructions and their trailer (continuation) cards. CAUTION: These comment lines are not printed in the ORTEP output file, and they are lost if the input file is edited with ORTEP-III's line editor.

1.2.7 Alternate Formats for Atomic Parameters

ORTEP-III provides a way to input atom parameters that are available in a "nonstandard" format. If the sentinel value (column 1) on the last symmetry card (see 3.2.3) in the ORTEP input file is "2" instead of "1", the program asks the user for the name of a file containing the atom parameters and branches to subroutine READIN to read the information. This subroutine may be recoded to read any desired format. (See 4.5.)

1.2.8 Atom "Features"

In earlier versions of ORTEP, atoms could be referenced only by their numeric positions in the input file; and atom number runs (ANR) (see 3.1.4) were used to select groups of atoms to be treated in the same manner. ORTEP-III allows two optional attributes called "features" to be included with each atom, and feature number runs (FNR) can be used to select groups of atoms having particular features. To handle features, a new parameter, number run type, has been added to the 100 series, 400 series, 505, 506, 700 series, 800 series, and 1001 instructions. Features should prove especially useful for polymeric materials such as proteins or nucleic acids and for critical net illustrations. (See 4.6.)

1.2.9 Critical Net Illustrations

ORTEP-III can produce critical net illustrations that depict some canonical topological characteristics of the global ensemble of overlapping atomic-thermal-motion Gaussian density functions in a crystal. (See 7.5.)

1.2.10 Symmetry Operator Format

The symmetry operators in the ORTEP input file (see 3.2.3) may now be provided in a free format using the xyz coordinate triplet notation found in the International Tables for Crystallography.[4] ORTEP is informed that this style for the symmetry operators is being used by having a "1" in column 1 of the cell parameter card (see 3.2.2). (A "0" or blank in that position indicates the old style is being used.)

Symmetry cards using this style do not have a specific format with the following two exceptions: (1) the symmetry information on each card must not go beyond column 72, and (2) column 1 must be blank on all symmetry cards other than the last one in the set, which must have a non-zero value in column 1. Below is an example set of symmetry cards to illustrate the flexibility of this style.

```
X,Y,Z
X   -Y   Z+1/2
X+0.5,  .5+Y,  Z
x+1/2,-y+1/2,1/2+z
```

Letters may be either upper or lower case. Commas or spaces may be used to separate the components of the triplet. The three components may not have spaces within them. Decimal fractions may be used with or without an initial 0. Fractions may precede or follow the letters.

1.2.11 Miscellaneous Changes in ORTEP-III

- No longer provides a choice of centered symbols.
- Adds a parameter LOGC to the 100 and 400 series instructions to control the logic used (union or intersection) when multiple screening conditions are applied to the atoms. (See 3.1.5.)
- Makes the parameters on the Format No. 2 trailer cards of the 100 and 400 series instructions optional. (See 3.1.5.)
- Increases number of symmetry cards from 48 to 96. (See 3.2.3.)
- Increases number of atoms from 166 to 500. (See 3.2.4.)
- Makes entry of VDC_2 on Type 6 and Type 7 temperature factor cards optional. (See 3.2.4.2.)
- Increases number of Format No. 2 trailer cards per instruction from 10 to 20. (See 3.3.1.)
- Adds a 205 instruction to change the plotting pen width. (See 3.3.3.)
- Makes "no retrace" (instruction 303) the default. (See 3.3.4.3.)
- Adds a 304 instruction to control the resolution (smoothing) of the ellipsoids. (See 3.3.4.4.)
- Allows atom screening on the 403/413 and 404/414 instructions. (See 3.3.5.3.)
- Changes 600 instructions to allow input of SCAL2 or ellipsoid probability. (See 3.3.7.)
- Adds a 706/716 instruction to add another standard ellipsoid type (open octant football) for drawing atoms. (See 3.3.8.)
- Adds lower case letters for labeling. (See 3.3.10.)
- Centers titles automatically if they begin in column 1 of Format No. 3 trailer cards following instructions 902, 903, and 913. (See 3.3.10.)
- Makes instruction number 1001 an alias for the 511 instruction of OR TEP-II.[2] (See 3.3.12.)

1.3 REPORT ORGANIZATION

This report covers the following topics. First, Section 2.1 includes a summary table of the ORTEP instructions and is the part of the report to which the experienced user will routinely refer. The remainder of Section 2 provides an overview of how to program an ORTEP illustration, ending with a detailed example. Section 3 defines the terms used in this report and describes the ORTEP input in detail. Section 4 discusses the use of ORTEP-III in general terms while Section 5 looks at some of the more technical aspects. The mathematics of thermal-motion probability ellipsoids are described in Section 6. Lastly, several example structures illustrating a number of ORTEP's capabilities are provided in Section 7. The appendices provide brief descriptions of the subprograms and the more important variables that are used in ORTEP along with a listing of the entire program's FORTRAN code.

2. PROGRAMMING ORTEP

2.1 GENERAL PRINCIPLES

Originally, the input to ORTEP was a set of punched cards. Now, the ORTEP input is a computer text file containing lines of information corresponding to the cards. However, since the input must generally be very precisely formatted as required by the FORTRAN code, it is still convenient to refer to the individual lines of input as "cards," and that practice is maintained in this report.

For those who may be unfamiliar with FORTRAN input, two points are in order. First, when a card field is specified to contain a particular numeric input item, the value should be entered in the rightmost positions of the field. Secondly, the value of "0" is assigned to a numeric input item whose card field is left blank.

ORTEP input consists of five types of information.

- Title card (see 3.2.1)
- Cell parameter card (see 3.2.2)
- Symmetry cards (see 3.2.3)
- Atomic parameter cards (see 3.2.4)
- Instruction cards (see Table 2.1 and 3.3)

ORTEP applies the concept of programming to the task of drawing illustrations. Using this approach, a set of basic building block operations have been developed that are put together by the user to "program" an illustration. These operations, which are termed instructions, control six types of activities.

- setting up the graphics
- composing an illustration
- drawing the illustration
- repeating a sequence of other instructions
- calculating and printing tables of ancillary information
- terminating the process

Table 2.1 is a summary of the ORTEP instructions. Each instruction starts on a separate card and consists of an identifying number and the parameters needed for the particular instruction. The general role of these instructions is explained in the remaining parts of this section, and the individual instructions are described in detail in Section 3.3. The simplest way to construct the program is, first, to select a "template" instruction set used for some previous related ORTEP drawing and then to scan through the list of instructions in Table 2.1 and pick out the relevant parameters and new instructions to modify the template.

In order to produce high quality illustrations with ORTEP, an iterative approach is generally required; that is, the illustration must usually be computed and plotted several times before an optimal figure is produced. With each trial, as many factors as possible are optimized to give a more informative and more aesthetically pleasing result.

Table 2.1a. Summary table of Format No. 0 ORTEP-III instructions and Format No. 1 instruction continuation cards.

Function	1-3	4-9	10-18	19-27	28-36	37-45	46-54	55-63	64-72
Structure Analysis									
Distances	0, 1, or 2	101	Org. ADR (f)	Org. ADR (t)	Tar. ANR (f)	Tar. ANR (t)	D_{max} (Å)	—	—
(Format No. 1 trailer card)	2	—	—	[LOGC]	—	—	—	—	—
Distances + angles	0, 1, or 2	102	(same as 101)						
Principal axes	—	103	—	—	—	—	—	—	—
Distances single convolute	0, 1, or 2	105	Org. NR (f)	Org. NR (t)	Tar. NR (f)	Tar. NR (t)	D_{max} (Å)	NR Type	—
(Format No. 1 trailer card)	2	—	—	[LOGC]	—	—	—	—	—
Dist. reiterate convolute	0, 1, or 2	106	(same as 105)						
Plotter Control									
Initialize	—	201	—	—	—	—	—	—	—
Shift plot origin/terminate	—	202	[X (in.)]	[Y (in.)]	—	—	—	—	—
Color	—	204	ICOLOR	—	—	—	—	—	—
Pen width	—	205	WIDTH	—	—	—	—	—	—
Drawing Parameters									
Dimensions and view	—	301	X (in.)	Y (in.)	VIEW (in.)	BRDR (in.)	—	—	—
Title rotation	—	302	THETA (°)	—	—	—	—	—	—
Retrace displace	—	303	DISP (in.)	—	—	—	—	—	—
Ellipse smoothness	—	304	CHORD	—	—	—	—	—	—
ATOMS Array									
Run add	0 or 1	401	FROM (1)	(-) TO (1)	[FROM (2)	(-) TO (2)]	[FROM (3)	(-) TO (3)]	...
Run subtract	0 or 1	411	(same as 401)						
Sphere add	0, 1, or 2	402	Org. ADR (f)	Org. ADR (t)	Tar. ANR (f)	Tar. ANR (t)	D_{max} (Å)	—	—
(Format No. 1 trailer card)	2	—	—	[LOGC]	—	—	—	—	—
Sphere subtract	0, 1, or 2	412	(same as 402)						
Box add	0, 1, or 2	403	Org. ADR (f)	Org. ADR (t)	Tar. ANR (f)	Tar. ANR (t)	a/2 (Å)	b/2 (Å)	c/2 (Å)
(Format No. 1 trailer card)	2	—	—	[LOGC]	—	—	—	—	—
Box subtract	0, 1, or 2	413	(same as 403)						
Triclinic box add	0, 1, or 2	404	Org. ADR (f)	Org. ADR (t)	Tar. ANR (f)	Tar. ANR (t)	a/2 (fract.)	b/2 (fract.)	c/2 (fract.)
(Format No. 1 trailer card)	2	—	—	[LOGC]	—	—	—	—	—
Triclinic box subtract	0, 1, or 2	414	(same as 404)						
Convolute add	0, 1, or 2	405	Org. NR (f)	Org. NR (t)	Tar. NR (f)	Tar. NR (t)	D_{max} (Å)	NR Type	—
(Format No. 1 trailer card)	2	—	—	[LOGC]	—	—	—	—	—
Convolute subtract	0, 1, or 2	415	(same as 405)						
Reiterate convolute add	0, 1, or 2	406	Org. NR (f)	Org. NR (t)	Tar. NR (f)	Tar. NR (t)	D_{max} (Å)	NR Type	—
(Format No. 1 trailer card)	2	—	[ASYMUNIT]	[LOGC]	—	—	—	—	—
Reiterate convolute subt.	0, 1, or 2	416	(same as 406)						
Zero ATOMS array	—	410	—	—	—	—	—	—	—

Table 2.1a. Summary table of Format No. 0 ORTEP-III instructions and Format No. 1 instruction continuation cards.

Function	1-3	4-9	10-18	19-27	28-36	37-45	46-54	55-63	64-72
Cartesian System									
Explicit definition	—	501	ORGN	V1 (f)	V1 (t)	V2 (f)	V2 (t)	—	Type
Rotate reference	0 or 1	502	Axis No.	Rotation (°)	[Axis No.	Rotation]	...	...	...
Rotate working	—	503	Axis No.	Rotation (°)	—	—	—	—	—
Translate reference	—	504	ΔX (in.)	ΔY (in.)	ΔZ (in.)	—	—	—	—
Origin at centroid	0 or 2	505	—	—	—	—	—	—	—
Centroid org./inertial axes	0 or 2	506	—	—	—	—	—	—	—
Position and Scale									
Explicit center and scale	—	601	X0 (in.)	Y0 (in.)	SCAL1	SCAL2*	—	—	—
Explicit center and auto scale	—	602	X0 (in.)	Y0 (in.)	—	SCAL2*	—	—	—
Explicit scale and auto center	—	603	—	—	SCAL1	SCAL2*	—	—	—
Auto center and scale	—	604	—	—	—	SCAL2*	—	—	—
Incr. position and incr. scale	—	611	ΔX0 (in.)	ΔY0 (in.)	ΔSCAL1	SCAL2*	—	—	—
Incr. position and auto scale	—	612	ΔX0 (in.)	ΔY0 (in.)	—	SCAL2*	—	—	—
Incr. scale and auto center	—	613	—	—	ΔSCAL1	SCAL2*	—	—	—
Atom Plotting									
Shaded octant football	0 or 1	701	—	—	—	—	Sym. hgt. (in.)	∥ Offset (in.)	⊥ Offset (in.)
(Format No. 1 trailer card)	—	—	A0 (in.)	A1 (in.)	NR (f)	NR (t)	NR Type	—	—
Football	0 or 1	702	(same as 701)						
Open model	0 or 1	703	(same as 701)						
Boundary only	0 or 1	704	(same as 701)						
Explicit ellipsoid description	0 or 1	705	NPLANE	NDOT	NLINE	NDASH	Sym. hgt. (in.)	∥ Offset (in.)	⊥ Offset (in.)
(Format No. 1 trailer card)	—	—	A0 (in.)	A1 (in.)	NR (f)	NR (t)	NR Type	—	—
Open octant football	0 or 1	706	(same as 701)						
As above except	0 or 1	711	(same as 701)						
no printed output of	0 or 1	712	(same as 701)						
individual coordinates	0 or 1	713	(same as 701)						
	0 or 1	714	(same as 701)						
	0 or 1	715	(same as 705)						
	0 or 1	716	(same as 701)						

*Or probability (entered as negative whole number).

Table 2.1a. Summary table of Format No. 0 ORTEP-III instructions and Format No. 1 instruction continuation cards.

Function	1-3	4-9	10-18	19-27	28-36	37-45	46-54	55-63	64-72
Bond Plotting									
Explicit	[1 &] 2	801	ADC (f)	ADC (t)	[ADC (f)	ADC (t)]	[ADC (f)	ADC (t)]	...
Implicit stick	2	802	—	NR Type	—	—	—	—	—
Implicit line	2	803	—	NR Type	—	—	—	—	—
As above except	[1 &] 2	811	(same as 801)						
no printed output	2	812	(same as 802)						
	2	813	(same as 803)						
Label Plotting									
Atom label	—	901	ADC 1	[ADC 2]	X Reset (in.)	Y Reset (in.)	HGT (in.)	‖ Offset (in.)	⊥ Offset (in.)
Regular title	3	902	ADC 1	[ADC 2]	X Reset (in.)	Y Reset (in.)	HGT (in.)	‖ Offset (in.)	⊥ Offset (in.)
Normal plane vector title	3	903	ADC 1	ADC 2	X Reset (in.)	Y Reset (in.)	HGT (in.)	‖ Offset (in.)	⊥ Offset (in.)
General plane vector title	3	913	ADC 1	ADC 2	—	—	HGT (in.)	‖ Offset (in.)	⊥ Offset (in.)
Nrm. plane bond-length label									
(1 decimal place)	—	904	ADC 1	ADC 2	X Reset (in.)	Y Reset (in.)	HGT (in.)	‖ Offset (in.)	⊥ Offset (in.)
(2 decimal places)	—	905	(same as 904)						
(3 decimal places)	—	906	(same as 904)						
Gen. plane bond-length label									
(1 decimal place)	—	914	ADC 1	ADC 2	—	—	HGT (in.)	‖ Offset (in.)	⊥ Offset (in.)
(2 decimal places)	—	915	(same as 914)						
(3 decimal places)	—	916	(same as 914)						
Overlap Correction									
Atoms [and implicit bonds]	0 [or 2]	1001	0, 1, or	NR Type	—	—	—	—	—
(Format No. 1 trailer card)			OVMRGN (in.)						
Explicit bonds	[1 &] 2	821	ADC (f)	ADC (t)	[ADC (f)	ADC (t)]	[ADC (f)	ADC (t)]	...
Implicit bonds	2	822	—	NR Type	—	—	—	—	—
Save Sequence									
Start	—	1101	—	—	—	—	—	—	—
End	—	1102	—	—	—	—	—	—	—
Execute	—	1103	—	—	—	—	—	—	—
Terminate	—	-1	—	—	—	—	—	—	—
Next Structure	—	-2	—	—	—	—	—	—	—

Table 2.1b. Summary table of Format No. 2 instruction continuation cards.

				Positive Number or Blank in Col. 43-48			Negative Number in Col. 43-48	
Columns	101,102,105, 106,402/412, 405/415, 406/416	403/413, 404/414	505,506	801/811,821	802/812,822,1001	803/813	802/812, 822,1001	803/813
3	0 or 2	0 or 2	0 or 2	0 or 2	0 or 2	0 or 2	0 or 2	0 or 2
4-9	—	—	—	—	—	—	—	—
10-12	[Org. NR (f)	[Org. NR (f)	NR (f)	—	Org. NR (f)	Org. NR (f)	Org. NR (f)	Org. NR (f)
13-15	Org. NR (t)]	Org. NR (t)]	NR (t)	—	Org. NR (t)	Org. NR (t)	Org. NR (t)	Org. NR (t)
16-18	[Tar. NR (f)	[Tar. NR (f)	—	—	Tar. NR (f)	Tar. NR (f)	Tar. NR (f)	Tar. NR (f)
19-21	Tar. NR (t)]	Tar. NR (t)]	—	—	Tar. NR (t)	Tar. NR (t)	Tar. NR (t)	Tar. NR (t)
22-24	[NR type]	NR type	NR type	Bond type	Bond type	—	Bond type	—
25-30	[D_{min} (Å)	—	Weight	—	D_{min} (Å)	D_{min} (Å)	D_{min} (Å)	D_{min} (Å)
31-36	D_{max} (Å)]	—	—	—	D_{max} (Å)	D_{max} (Å)	D_{max} (Å)	D_{max} (Å)
37-42	—	—	—	Bond radius (Å)	Bond radius (Å)	—	Bond radius (Å)	—
43-48	—	—	—	Persp. label hgt. (in.)	Persp. label hgt. (in.)	—	Poly. NR (f)	Poly. NR (f)
49-54	—	—	—	⊥ displacement (in.)	⊥ displacement (in.)	—	Poly. NR (t)	Poly. NR (t)
55-60	—	—	—	Nonp. label hgt. (in.)	Nonp. label hgt. (in.)	—	Poly. D_{min} (Å)	Poly. D_{min} (Å)
61-66	—	—	—	⊥ displacement (in.)	⊥ displacement (in.)	—	Poly. D_{max} (Å)	Poly. D_{max} (Å)
67-72	—	—	—	Digits indicator	Digits indicator	—	—	—

2.2 PROGRAMMING A NONSTEREOSCOPIC ILLUSTRATION FOR ORTEP

This section describes the general stepwise procedure to follow when writing an ORTEP program to draw a single nonstereographic illustration of the contents of one unit cell. The instruction numbers used are examples only, and often other instructions may be used instead.

2.2.1 Graphics Setup

The first instruction card is instruction 201 (see 3.3.3), which initializes plotting.

Next, instruction 301 (see 3.3.4.1) is needed to set the following drawing parameters: x dimension for the plot boundary, y dimension for the plot boundary, viewing distance for perspective projection (or "0" as a signal for parallel projection), and border (or margin) dimension inside the boundary. All values are supplied in inches.

2.2.2 Composing the Illustration

This step involves specifying which atoms are to be used as the figure subject, the rotational orientation of the figure, and the scaling and positioning of the figure relative to the drawing area. These three components of composition are implemented by the 400, 500, and 600 series instructions, respectively.

Atoms can be explicitly added to the figure with a 401 instruction (see 3.3.5.1). For unit cell content drawings, the 404 instruction (see 3.3.5.3) is useful. It defines a triclinic box of enclosure, and ORTEP determines which atoms appear in the figure.

A 501 instruction (see 3.3.6.1) can be used to orient the crystal axes relative to the x and y axes of the plot. If additional adjustment of the figure orientation is necessary, a 502 instruction (see 3.3.6.2) can be used after the 501.

Scaling and positioning of the figure to fill the drawing area can be accomplished automatically with a 604 instruction (see 3.3.7.1).

2.2.3 Drawing the Illustration

Crystal structure illustrations of the ball-and-stick type are made up of three components: balls (atoms), sticks (bonds), and labels. The three components are drawn with the 700, 800, and 900 instruction series, respectively; the first two instruction series can also perform certain types of labeling. Before drawing any atoms or bonds, instruction 1001 (or 511) (see 3.3.12) should be used to calculate and store the information needed for the overlap correction.

The atom representation can be either a general ellipsoid or a boundary ellipse. In some cases, these become a sphere and a circle. Chemical symbols may be plotted simultaneously with the atoms. A 704 instruction (see 3.3.8) will draw circles for all the atoms of the subject and put the chemical symbols within the circles.

Bonds are not always necessary in a drawing; but for structures with molecules or with distinctive groupings, they are usually quite helpful. The most convenient method for describing and drawing bonds is instruction 812 (see 3.3.9.2). This instruction uses vector search codes (see 3.1.5) that reflect the user's knowledge of the structural chemistry and the interatomic distance ranges for the compound being drawn. Covalent bonds or any other desired type are found and

drawn automatically from the list of atoms that make up the subject. If desired, the interatomic distance label can also be drawn with the bond.

Various types of labeling can be done with the 900 series instruction. The one that will most often be included is a caption for the figure, provided by the 902 instruction (see 3.3.10).

2.2.4 Terminating the Drawing of the Illustration

Instruction 202 with no parameters (see 3.3.3) terminates the plotting.

To terminate ORTEP, a -1 instruction (see 3.3.13) is used as the last instruction of the input.

2.3 PROGRAMMING A STEREOSCOPIC ILLUSTRATION FOR ORTEP

A stereoscopic pair of figures is simply two perspective views of the subject as seen from two different viewpoints (which are usually 5°–6° apart). This pair is produced with ORTEP by programming for two drawings. A few instructions in addition to those outlined in Section 2.2 are needed for producing stereo figures. These are the stereoscopic rotation instruction 503 and the 1100 series of instructions that are used to repeat a series of instructions. A program to draw a stereo pair would involve the following steps.

1. set up the graphics
2. compose the subject
3. stereo rotate subject for left-eye view
4. store overlap information
5. draw the subject
6. shift plot origin for second view
7. stereo rotate subject for right-eye stereo view
8. store overlap information
9. draw the subject
10. terminate plotting
11. terminate ORTEP

2.3.1 Stereoscopic Rotations

In general, one member of a detailed stereoscopic illustration cannot be drawn independently of the other member of the pair because certain features (e.g., which octant of an ellipsoid is shaded) must be done identically in the two drawings. In ORTEP the "stereoscopically sensitive decisions" are handled by using two Cartesian coordinate systems: the reference system and the working system (see 3.1.8). The steps involved in picture composition (see 2.2.2) and the stereoscopically sensitive decisions are always based on the reference system, but the drawing of the illustration (see 2.2.3) is always based on the working system. A stereoscopic rotation is simply a rotation of the working system *from the reference system* about the axis that is vertical while viewing the final result. For example, a nominal rotation of +2.7° about the plot's y axis might be used for the left-eye view and a rotation of -2.7° about the same axis might be made before plotting the right-eye view, thus producing a total interocular angle of 5.4°.

2.3.2 Repeating a Sequence of Operations

It should be noted that steps 8 and 9 in the program to draw a stereo pair, which actually comprise several ORTEP instructions, are identical to steps 4 and 5. The program can be shortened somewhat by using the "save sequence" instructions (see 3.3.11). An 1101 instruction (start save sequence) would be placed before step 4, and an 1102 instruction (end save sequence) after 5. Then steps 8 and 9 can be replaced by a single 1103 instruction (execute save sequence).

Any sequence of instructions can be saved in this manner and repeated as many times as desired with 1103 instructions. For example, the save sequence feature can be used to produce a complete series of views of a structure at 15° intervals about an axis. Note that the instructions between the start and end instructions are *both executed and saved* the first time through.

2.4 DRAWING THE CUBANE STRUCTURE: AN EXAMPLE

The standard example for illustrating the use of ORTEP is the novel compound cubane (C_8H_8), whose structure was published by Fleischer in 1964.[5] In cubane, the carbon-carbon bonds lie along the edges of a cube within experimental error. The compound crystallizes with the trigonal symmetry of space group $R\bar{3}$. The $\bar{3}$ axis lies along a body diagonal of the molecule, and as a result the compound contains only four unique atoms. These are one carbon and its attached hydrogen in general positions off the $\bar{3}$ axis (C1 and H1) and one carbon and its hydrogen in special positions on the $\bar{3}$ axis (C2 and H2). Anisotropic temperature factor coefficients were fitted to the carbon atoms during the least-squares refinement of the structure, and isotropic temperature factors were used for the hydrogen atoms. The anisotropic temperature factors given for the carbon atoms are of the type called zero† in this report (see 3.2.4.2).

To draw the structure the following information is needed:

Cell Parameters	$a = b = c = 5.34$ Å, $\alpha = \beta = \gamma = 72.26°$
Equivalent Positions of Space Group $R\bar{3}$	$x, y, z \quad z, x, y \quad y, z, x \quad \bar{x}, \bar{y}, \bar{z} \quad \bar{z}, \bar{x}, \bar{y} \quad \bar{y}, \bar{z}, \bar{x}$
Positional Parameters	x y z
C1	-0.18711 0.19519 0.10706
C2	0.11546 0.11546 0.11546
H1	-0.32460 0.34680 0.18480
H2	0.21000 0.21000 0.21000
Anisotropic Temperature Factor Coefficients	b_{11} b_{22} b_{33} b_{12} b_{13} b_{23}
C1	0.0410 0.0425 0.0450 -0.0042 -0.0142 -0.0051
C2	0.0468 0.0468 0.0468 -0.0143 -0.0143 -0.0143

†Occasionally authors of structure papers neglect to define the equation for the anisotropic temperature factor coefficients. In the present case, the type can be determined from the comparative isotropic temperature factor listed in anisotropic form in Fleischer's Table II.[5] In other instances, one must arbitrarily choose a type (usually 0, 1, or 4 in the USA), do the principal axis transformation, and then check that the principal values are correct, or at least reasonable. In particular, the principal values must all be positive.

2.4.1 Data Input for Cubane

2.4.1.1 Title Card

The first line of input is a job title (see 3.2.1).

```
CUBANE, E.B. FLEISCHER, J. AM. CHEM. SOC., 86, 3889 (1964)
```

2.4.1.2 Cell Parameter Card

Cell parameters are provided on the second line (see 3.2.2). The "1" in the first column signals the format of the symmetry information that follows.

```
1    5.34      5.34      5.34     72.26     72.26     72.26
```

2.4.1.3 Symmetry Cards

The symmetry operators of the space group begin on line 3 of the input (see 3.2.3). The last of these has a numeric value other than "0" in column 1.

```
  x, y, z
  z, x, y
  y, z, x
  -x, -y, -z
  -z, -x, -y
1 -y, -z, -x
```

2.4.1.4 Atomic Parameter Cards

Two cards are required for each atom. The first gives the atom's positional parameters (see 3.2.4.1), and the second provides its thermal parameters (see 3.2.4.2). The last atom has a "1" in column 1 of its thermal parameter card. Atoms are frequently referenced in the ORTEP instructions by their numeric position within this list.

Atoms 1 and 2 are entered with positional parameter Type 0 and anisotropic temperature factor Type 0.

```
  C1                          -.18711   .19519   .10706         0
0  .04100    .04250    .04500  -.00420  -.01420  -.00510         0
  C2                           .11546   .11546   .11546         0
0  .04680    .04680    .04680  -.01430  -.01430  -.01430         0
```

Atoms 3 and 4 are entered with positional parameter Type 0 and with dummy sphere temperature factors (Type 7) with a radius of 0.1 Å before scaling.

```
  H1                          -.32460   .34680 .  .18480         0
0     .10                                                        7
  H2                           .21000   .21000   .21000         0
0     .10                                                        7
```

A dummy atom (atom 5) at the cell origin is also included with a blank card dummy sphere. This could also be entered with Type 7 as were atoms 3 and 4.

```
 ORGN                                    0         0         0         0
1                                                                      0
```

2.4.2 Analysis of Structure

The 100 series instructions (see 3.3.2) are neither associated with nor required for producing an illustration. They are shown here to demonstrate how they are used. A 101 instruction is used to obtain a tabulation of the atoms surrounding one atom or a series of several designated atoms. For example, to obtain a list of all atoms (hydrogen and carbon atoms) out to a distance of 3.61 Å about the two carbons C1 and C2, the following 101 instruction would be used.

```
101   155501          2         1         4      3.61
      └──────────────┘          └──────────┘      └────┘
            (a)                    (b)            (c)
```

where the parts designate:

(a) origin atoms 1 through 2 of symmetry operation 1,
(b) target atoms 1 through 4 of all symmetry and translation operations, and
(c) a distance D_{max} of 3.61 Å.

A 102 instruction gives both interatomic distances and interatomic angles. The following instruction could be used to find all covalent bonds and bond angles about the two carbons.

```
102   155501          2         1         4      1.8
```

In this case a smaller D_{max} is used so that only the distances and angles of immediate interest would be computed since there are $n(n - 1)/2$ angles for n interatomic vectors about an atom.

2.4.3 Programming the Cubane Illustration

First, plotting is initialized with a 201 instruction (see 3.3.3).

```
201
```

The two plot boundary dimensions can be equal for the present illustration since the cubane molecule is a cube. A 2.8 × 2.8 inch boundary is specified with a 0.25 inch margin to give a 2.3 × 2.3 inch square working area. A 10 inch view distance might be reasonable to use in viewing a model of this size. These are set with a 301 instruction. (See 3.3.4.1.)

```
301       2.8       2.8       10.       .25
```

The subject of the illustration is a single complete cubane molecule. Since all the atoms of the molecule were not provided in the input atoms list, the "missing" atoms may be found by using a 402 instruction to specify a sphere of enclosure, centered on the dummy atom 5, which is at the center of a cubane molecule. A radius of 3.2 Å should be adequate to find all the atoms and

isolate a single molecule. The atoms found by this instruction are stored in the ATOMS array. (See 3.3.5.2.)

```
402   555501        5        1        4      3.2
      └──────────┘          └──────┘      └──┘
          (a)                 (b)          (c)
```

where the parts designate:

(a) a run of origin atom(s) representing sphere centers from atom 5 to atom 5 in symmetry position 55501 (in this example, a single sphere),
(b) a run of target atoms from atom 1 to atom 4, and
(c) a sphere radius of 3.2 Å.

A 501 instruction is used to establish a coordinate system for orienting the molecule. In this case, the coordinate system is defined along the edges of the cubane cube. The origin is positioned on the dummy atom 5. The desired coordinate system orientation is defined by specifying two vectors from the special position atom 255501 to the two symmetry-related general-position atoms 155501 and 155502. (See 3.3.6.1.)

```
501   555501   255501   155501   255501   155502        0         0
```

A 502 instruction is used to rotate the molecule relative to the established coordinate system to optimize the appearance of the illustration. A rotation of 25° about the *y* axis (axis 2) followed by a rotation of 28° about the *x* axis (axis 1) will produce a satisfactory view of the molecule. (See 3.3.6.2.)

```
502        2      25.        1      28.
```

To position and scale the subject for projection onto the "drawing board" and to utilize all available space, a 604 instruction is used, which automatically sets the origin and drawing scale. It also sets the ellipsoid scale factor ratio SCAL2 to the value corresponding to 50% probability unless a non-zero value is supplied on the instruction card. (See 3.3.7.1).

```
604        0         0         0         0
```

The stereoscopic rotation for the left eye is specified next with a 503 instruction. A rotation of 2.7° about the *y* axis (axis 2) is used. Later in the program, a –2.7° rotation about the same axis will be made for the right-eye view to give a total interocular angle of 5.4°. (See 3.3.6.3.)

```
503        2      2.7
```

Since the structure will be drawn twice, once for each eye, the "save sequence" feature is used to shorten the program. Note that the instructions between the start (1101) and end (1102) are *both executed and saved* the first time through. They can then be re-executed as many times as desired by using the "execute save sequence" instruction 1103. (See 3.3.11.) The 1101 instruction starts the save sequence.

```
1101
```

The 1001 (or 511) instruction stores the information needed for the overlap hidden-line correction (i.e., as a projected boundary ellipse for each atom in the ATOMS array and a quadrangle approximation for each bond). The bonds are specified with a Format No. 2 trailer card identical to that used in the 812 instruction described below. (See 3.3.12.1.)

```
2 1001
        1  4  1  4  4   0.9   1.6   .04
```

The ATOMS array currently contains all the atom designators for one cubane molecule. The molecule is drawn in two separate steps so that the hydrogen and carbon atoms can be given different graphical representations. To draw the carbon atoms (ANR = 1,2), a standard model produced with the 716 instruction (new in ORTEP-III) is used. [This model is the same as that produced with a 705/715 instruction that (a) draws the three principal-plane forward traces and the boundary-plane trace (NPLANE = 4), (b) omits the reverse sides of the principal planes (NDOT = 0), (c) draws the forward principal axes without additional shading (NLINE = 1), and (d) omits the reverse principal axes (NDASH = 0).] In addition, chemical symbols are drawn with letters 0.07 in. high (before projection) and displaced from the atomic centers by 0.18 in. horizontally and 0.21 in. vertically. (See 3.3.8.)

```
1  716                                                  .07      .18      .21
     0        0        0        1        2
```

The hydrogen atoms (ANR = 3, 4) are drawn with a different standard model (instruction 712) than the carbon atoms. Chemical symbols are 0.07 in. high and offset 0.15 in. horizontally and 0.13 in. vertically.

```
1  712                                                  .07      .15      .13
     0        0        0        3        4
```

The most convenient procedure for drawing bonds is to use the implicit bond instruction 812 (see 3.3.9.2). All other information can be entered with a single Format No. 2 trailer card.

```
2  812
     1  4  1  4  4   0.9   1.6   .04
     |--|  |--|  |   |-----|  |---|
      (a)   (b)  (c)    (d)     (e)
```

where the parts designate:

(a) origin atom number run of those atoms that must be at one end of each bond,
(b) target atom number run of those atoms that must be at the other end of each bond,
(c) bond type 4,
(d) the distance range, 0.9 to 1.6 Å, that will cover all covalent bond distances, and
(e) the bond radius, 0.04 Å.

The remaining fields on the card are blank since a complete set of bond distance labels is not desired.

The last feature of the illustration is the labeling with 900 instructions (see 3.3.10). Because of the symmetry, there are only two different C–C bond lengths in cubane. These are C1–C1 and C1–C2. One example of each of these bonds is labeled. For variety, one is labeled with a normal bond-length label and the other with a perspective label. The two bonds that can be labeled most advantageously are 155504–155503 and 255504–155505. The labels will be 0.07 in. high and displaced vertically from the bond center by -0.2 in.

```
906    155504    155503        0        0       .07        0       -.2
916    255504    155505        0        0       .07        0       -.2
```

Finally, a caption for the illustration is drawn. This can conveniently be positioned by "hanging" it from the dummy atom 555501 and "bouncing" it 1.8 in. from the left x boundary and 0.3 in. from the lower y boundary. The caption is 0.15 in. high.

```
  3   902    555501        0       1.8       0.3       .15        0         0
CUBANE
```

The save sequence is now terminated.

```
1102
```

The plotting origin is then shifted 2.375 in. along x with a 202 instruction in preparation for the right-eye view.

```
202     2.375
```

The stereo rotation of -2.7° about axis 2 is now performed for the right-eye view with a 503 instruction. (Note that this rotation starts with the reference orientation, not the previous working orientation.)

```
503         2      -2.7
```

The save sequence is now repeated for the right eye. (Note that the ATOMS array contains the same information that it did when the first view was drawn.)

```
1103
```

The illustration is now complete. Plotting is terminated with a 202 instruction with no parameters.

```
202
```

Finally, a -1 instruction terminates the program. (See 3.3.13.)

```
-1
```

2.4.4 Illustration of the Example

Figure 2.1 shows a stereoscopic illustration of cubane exactly as it was produced by ORTEP. The drawing has not been retouched. Note that one bond distance label was drawn in perspective along the bond, and the other was drawn as a regular label parallel to the page.

Note , too, that while most of the atom labels are easy to read and it is clear which atoms they are labeling, some are in locations where the atom being labeled is ambiguous and some are not corrected for overlap. The first problem is the result of labeling a number of atoms with the 700 instruction that draws the atoms. More precise placement can be accomplished by labeling atoms individually (see 7.1) though this can be a time consuming process.

Overlap, particularly for chemical symbols and bond distance labels, is often one of the major problems encountered when producing an illustration. For nonstereoscopic figures, it may be better to add the lettering after the drawing is completed. However, for stereoscopic figures, adding the lettering after the illustration is produced is unsatisfactory because of the necessity for exact relative placement of the lettering on the two views to maintain good stereopsis. In this case, it is best to let ORTEP place the lettering.

The ORTEP input consisting of the crystal data and ORTEP instructions used to produce the structure illustration immediately follows the figure. The example utilizes a wide range of instructions in order to demonstrate their use. As is the case with any programming system, there are many ways of doing a given problem. The user should examine the example carefully since several useful techniques are illustrated. Additional examples are provided in Section 7.

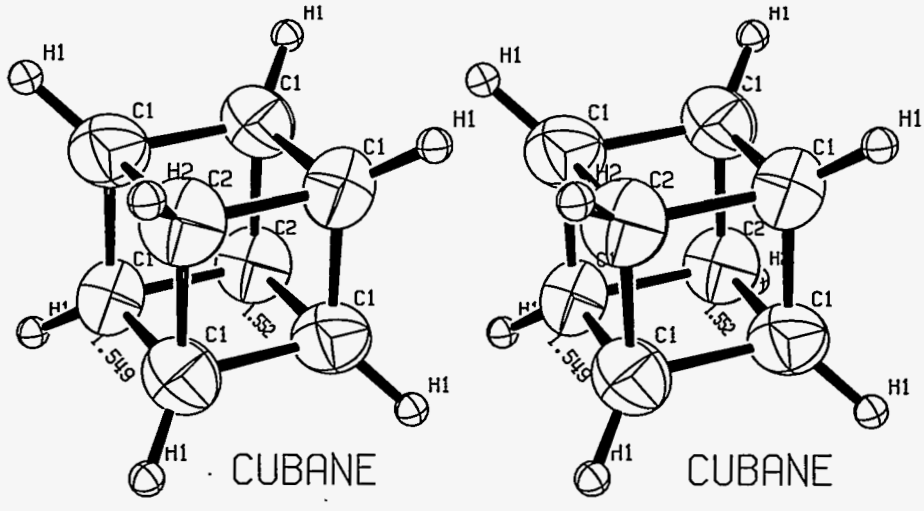

Fig. 2.1. Cubane drawn by ORTEP as a stereoscopic pair.

```
 CUBANE, E.B. FLEISCHER, J. AM. CHEM. SOC., 86, 3889 (1964)
1    5.34      5.34      5.34     72.26     72.26     72.26
  x, y, z
  z, x, y
  y, z, x
 -x, -y, -z
  -z, -x, -y
1 -y, -z, -x
  C1                            -.18711    .19519    .10706         0
   .04100    .04250    .04500  -.00420   -.01420   -.00510         0
  C2                             .11546    .11546    .11546         0
   .04680    .04680    .04680  -.01430   -.01430   -.01430         0
  H1                            -.32460    .34680    .18480         0
      .10                                                           7
  H2                             .21000    .21000    .21000         0
      .10                                                           7
 ORGN                                 0         0         0         0
1                                                                   0
# FIND ALL NEIGHBOR ATOMS AROUND C1 AND C2 OUT TO 3.61 A
      101    155501         2         1         4      3.61
# FIND COVALENT BONDS AND BOND ANGLES AROUND CARBONS
      102    155501         2         1         4       1.8
# INITIALIZE PLOTTING
      201
# 2.8x2.8 BOUNDARY, 2.3x2.3 INSIDE 0.25 MARGIN, VIEW FROM 10 INCHES
      301       2.8       2.8       10.       .25
# ENCLOSER SPHERE OF RADIUS 3.2 A ABOUT DUMMY ATOM 5,555,01
      402    555501         5         1         4       3.2
# ORIGIN ON DUMMY ATOM 555501, VECTORS ALONG 2 EDGES OF CUBANE
      501    555501    255501    155501    255501    155502                    0
# ROTATE 25 DEGREES ABOUT Y, THEN 28 DEGREES ABOUT NEW X
      502         2       25.         1       28.
# AUTOMATIC SCALE AND POSITION, 50 PERCENT PROBABILITY ELLIPSOIDS
      604         0         0         0         0
# STEREO ROTATION OF 2.7 DEGREES ABOUT Y FOR LEFT EYE
      503         2       2.7
# START SAVE SEQUENCE
     1101
# STORE PROJECTED ATOMS AND BONDS FOR OVERLAP, MARGIN SET BY DEFAULT
  2  1001         0
           1  4  1  4  4   0.9   1.6   .04
# DRAW CARBON ATOM ELLIPSOIDS, ATOM NUMBER RUN 1-2
  1   716                                                 .07       .18       .21
                                     1         2
# DRAW HYDROGEN ATOM SPHERES, ATOM NUMBER RUN 3-4
  1   712                                                 .07       .15       .13
                                     3         4
# TYPE 4 BONDS .04 A MEAN RADIUS, ATOMS 1-4 TO ATOMS 1-4, 0.9-1.6 A
  2   812
           1  4  1  4  4   0.9   1.6   .04
# LABEL BOND 155504-155503 WITH REGULAR BOND DISTANCE LABEL
      906    155504    155503         0         0       .07         0       -.2
# LABEL BOND 255504-155505 WITH PERSPECTIVE BOND DISTANCE LABEL
      916    255504    155505         0         0       .07         0       -.2
```

```
# DRAW TITLE
  3    902    555501         0       1.8       0.3       .15         0         0
CUBANE
# END SAVE SEQUENCE
     1102
# SHIFT DRAWING ORIGIN 2.375 INCHES ALONG X
      202     2.375
# STEREO ROTATION OF -2.7 DEGREES ABOUT Y FOR RIGHT EYE VIEW
      503         2      -2.7
# EXECUTE SAVE SEQUENCE FOR RIGHT EYE DRAWING
     1103
# TERMINATE PLOTTING
      202
# TERMINATE ORTEP
       -1
```

3. ORTEP INPUT

3.1 DEFINITIONS

Several terms must be defined before the instructions can be explained.

3.1.1 Atom Designator Code (ADC) and Addressable Point

A five-component *atom designator code* is used to specify a particular atom in the crystal within a reasonable distance from the crystallographically defined origin.

$$ADC = AN*10^5 + (TA + 5)*10^4 + (TB + 5)*10^3 + (TC + 5)*10^2 + SN\,,$$

where

AN = *atom number* ($0 \le AN \le NATOM \le 500$) – the numerical position of the atom in the input list of atoms in the asymmetric unit, which contains NATOM atoms. Atom 0 is not in the input atom list but refers to the crystal origin point (0.,0.,0.).

TA,TB,TC = *crystal lattice translation digits* – translations along cell edges **a**, **b**, and **c**, respectively. Each digit in an ADC can range from 1 to 9; consequently, it is possible to move up to 4 cells in any direction from the origin cell 555.

SN = *symmetry operator number* ($0 \le SN \le NSYM \le 96$) – the numerical position of the symmetry operator in the input list of symmetry operators, which contains NSYM entries. Symmetry operator number 0 is not in the input list but refers to an identity operator. However, the identity operation (corresponding to position *x, y, z*) generally must be somewhere in the input symmetry operator list and is usually the first operator.

Example: An atom designator code of 347502 refers to atom 3 moved through symmetry operation 2, then translated -1 cell translation along **a**, +2 cell translations along **b**, and 0 cell translations along **c**.

An *addressable point* in the crystal is any point for which an atom designator code exists. In general, the addressable region is approximately a $9 \times 9 \times 9$ block of unit cells.

3.1.2 Vector Designator Code (VDC)

A *vector designator code* defines a vector using two atom designator codes. The vector direction is from the first to the second.

Example: 253704 263704 is a vector along the positive **a** direction of the crystal lattice.

3.1.3 Atom Designator Run (ADR)

An *atom designator run* is a straight run sequence of atoms that is defined using two atom designator codes with a "-" preceding the second of the two. The run hierarchy is: first, atom number AN; second, symmetry operator number SN; third, **a** translation TA; fourth, **b** translation TB; and last, **c** translation TC.

Example: ADR (145502-245603) will generate the 8-atom run 145502, 245502, 145503, 245503, 145602, 245602, 145603, 245603.

The following exceptions are allowed in Org. ADRs of instructions 101, 102, 402/412, 403/413, and 404/414 **only**:

- The "-" may be omitted from the second ADC.
- If the second atom in the atom designator run has the same symmetry and translation components as the first atom, the second atom may be represented by its atom number component alone.
- If the symmetry and translation components of both atoms are 55501, both atoms may be represented by their atom number components alone.

Example: ADR (345502-745502) may also be represented as (345502 745502) or (345502 7). ADR (355501-755501) may also be represented as (355501 755501) or (355501 7) or (3 7).

3.1.4 Atom Number Run (ANR)

An *atom number run* is a subset of the atom designator run in which only the atom number AN changes. Normally, an ANR is entered by using only the atom number values for the first and last members of the sequence without a "-".

Example: (1 4) will designate atoms 1, 2, 3, and 4 of the input list.

3.1.5 Vector Search Code (VSC)

A *vector search code* consists of two number runs and a distance range. It is used for finding interatomic distances that have a particular chemical significance, such as covalent and coordination bonds.

Example: Suppose that metal atoms are numbers 1 and 2 in the atom list, oxygen atoms are 6-12, and the interatomic distance range between metals and oxygens is 1.9 Å to 2.4 Å. The metal-to-oxygen vectors can be specified by the vector search code (1 2) (6 12) (1.9 2.4). Several variations of this basic code are used in ORTEP.

The vector searches in ORTEP-III for the 100 instructions and 400 instructions have been generalized to allow both atom number runs and feature number runs (see 3.1.11) with the number run type (see 3.1.12) specified in column 24. In addition, it is no longer necessary to specify all three screening ranges found on this card (i.e., origin, target, and distance range). Any ranges not specified (i.e., zero or blank in the maximum value entry) are omitted from the screening. If there is more than one vector search card for an instruction, the parameter LOGC controls the logic of the screening. If the value is "0" (the default), an atom satisfying the screening conditions on any one of the No. 2 trailer cards will be retained (i.e., it is a logical union of the results). If the value is "1", an atom must satisfy the screening conditions on all the cards to be retained (i.e., it is an intersection of the results). If a value of "1" is needed for LOGC, a "1" is placed in column 27 of a Format No. 1 trailer card that goes between the main instruction card and the Format No. 2 VSC cards. If the value for LOGC is "0", the Format No. 1 card is not needed.

3.1.6 Sphere of Enclosure

A *sphere of enclosure* specifies some or all of the atoms lying within a sphere of radius D_{max} about a given "origin" atom without the necessity of delineating each atom individually. The sphere of enclosure is said to contain a *complete population* if all addressable atoms within the D_{max} radius are included. If the sphere of enclosure contains only certain types of atoms that are derived from a group of sequential atoms in the input list or atoms having particular features, then the sphere is said to have a *partial population.* Finally, the population (complete or partial) of the sphere of enclosure can be screened as selectively as desired through the use of vector search codes (see 3.1.5), and the resulting content is called a *vector screened population.*

A sphere of enclosure can be centered on any addressable atom, but the origin atom should not be chosen in the outermost cells because of the possibility of having nonaddressable points within the D_{max} radius.

3.1.7 Box of Enclosure

A *box of enclosure* is a parallelepiped that can be centered about any addressable point and assigned arbitrary orientation and dimensions. The orientation depends upon either the unit cell axes (triclinic box of enclosure) or the reference axes (rectangular box of enclosure). The box of enclosure can have a *complete population,* a *partial population,* or a *vector screened population* as described for the sphere of enclosure (see 3.1.6).

3.1.8 Reference, Working, and Standard Cartesian Coordinate Systems

Many of the ORTEP calculations use fractional coordinates based on the crystal axes **a**, **b**, and **c** (triclinic coordinate system); but other steps necessitate the introduction of orthonormal base vector triplets (Cartesian coordinate systems). Two Cartesian systems—reference and working—are utilized. The reference (major) system is used for all operations except plotting, where the working (minor) system is used. For a right-eye or left-eye stereo view, the working system is moved from the reference system by rotation about an axis of the reference system. However, certain decisions made while plotting must still be referred to the reference system to maintain accurate stereopsis. The user can define and orient the two Cartesian systems through the series 500 instructions. Until a 500-series instruction is given, a "standard Cartesian system" is utilized for both the reference and working systems. The orthonormal base vectors of the standard system are oriented as follows:

x axis along $\mathbf{a}$,

y axis along $(\mathbf{a} \times \mathbf{b}) \times \mathbf{a}$,

z axis along $(\mathbf{a} \times \mathbf{b}) = \mathbf{c}^*$,

where **a**, **b**, and **c** are crystal axes and × denotes the outer vector product (cross product). The symbol **c*** refers to a reciprocal axis.

3.1.9 Prime Parameters and Primer Constants

The more basic among the many settable parameters in ORTEP are the *prime parameters.* The default values assigned to these prime parameters are often similar or identical from one problem to the next. Among the first things ORTEP does is a call to subroutine PRIME, which sets as many prime parameters as possible to reasonable default *primer constant* values. For

example, the maximum plot dimensions (instruction 301) are set to 10.5 in. for x_{max} and 8.0 in. for y_{max}, and the overall scale for plotting (instruction 600 series) is set to 1.0 in./Å. If the value assigned to a particular constant by the PRIME subroutine is satisfactory, the user does not have to change the value with ORTEP instructions.

3.1.10 Atom Feature

An *atom feature* is a user-defined characteristic of a group of atoms that may be assigned to the atoms in the ORTEP input as needed for a given task. For example, the atomic number may be provided with each atom to make selecting atoms of the same element easier. Up to two features may be assigned to each atom. These are referred to as Feature #1 and Feature #2. (See 4.6.)

3.1.11 Feature Number Run (FNR)

A *feature number run* is used to identify those atoms having a particular atom feature within a specified value range.

3.1.12 Number Run (NR) and Number Run Type

A *number run* is a generic term that refers to both atom number runs (ANR) (see 3.1.4) and feature number runs (FNR) (see 3.1.11). The *number run type* identifies the number run. Number run type 0 refers to an ANR, type 1 refers to an FNR on Feature #1, and type 2 refers to an FNR on Feature #2.

3.2 CRYSTAL STRUCTURE DATA INPUT

3.2.1 Title

The first card in the ORTEP input is a title card with FORMAT (18A4), consisting of up to 72 characters of alphanumeric identification information. This will appear periodically in the output file.

3.2.2 Cell Parameters

The second input card contains the cell parameters with FORMAT (I1,F8.6,5F9.6). Any one of the following four input alternatives may be used.

Columns	Type A	Type B	Type C	Type D
1	Symmetry format indicator	Symmetry format indicator	Symmetry format indicator	Symmetry format indicator
2-9	**a** (Å)	**a** (Å)	**a*** ($Å^{-1}$)	**a*** ($Å^{-1}$)
10-18	**b** (Å)	**b** (Å)	**b*** ($Å^{-1}$)	**b*** ($Å^{-1}$)
19-27	**c** (Å)	**c** (Å)	**c*** ($Å^{-1}$)	**c*** ($Å^{-1}$)
28-36	α (°)	cos α	α^* (°)	cos α^*
37-45	β (°)	cos β	β^* (°)	cos β^*
46-54	γ (°)	cos γ	γ^* (°)	cos γ^*

No indicator is needed to specify the input type. The routine assumes that $\mathbf{a} \geq 1.0$ Å, $\mathbf{a}^* < 1.0$ Å^{-1}, α (or α^*) $\geq 1.0°$, and $|\cos \alpha|$ (or $|\cos \alpha^*|$) < 1.0. The parameters $\mathbf{a}^*$, etc., refer to the reciprocal unit cell such that $\mathbf{a} \cdot \mathbf{a}^* = 1$. All four types will be printed in the output regardless of which type was used for input.

An integer value in column 1 of the cell parameter card indicates the format used for the crystal symmetry cards that follow.

3.2.3 Symmetry

Crystal symmetry in ORTEP-III may be supplied in either of two styles. The first of these is identical to that of OR TEP-II[2] and is triggered by having a "0" or blank in column 1 of the cell parameter card. A "1" in that position indicates the symmetry operators are provided in a free format using the xyz coordinate triplet notation found in the International Tables for Crystallography.[4] These two styles are referred to as Type 0 and Type 1, respectively.

The number of symmetry cards (NSYM) may not exceed 96. At least one (the identity operator) is required. The reason for the maximum of 96 is that the *symmetry operator number* (SN) occupies only two places in the ADC (see Sect. 3.1.1). If it is not possible to supply all the symmetry operators for the space group (or if the user chooses not to supply all of them), each unique atom in the ORTEP input file will require multiple entries with those lattice centering translations added that are not provided in the symmetry cards. (See 7.5 for an example.)

A Type 0 symmetry card has FORMAT (I1,F14.10,3F3.0,2(F15.10,3F3.0)) and will be interpreted in one of two ways, depending on the value of the number in columns 70-72. If that number is < 5.0, the card is interpreted as a crystallographic symmetry operation; but if the number is ≥ 5.0, the card is interpreted as a general helix-screw symmetry operation† along the $\mathbf{c}^*$ crystal axis (third axis of the standard Cartesian system; see 3.1.8). The two symmetry types can be intermixed if desired.

---

†The general helix-screw symmetry operation is not an allowed element of a crystallographic group, so the molecular environment of the transformed unit will not in general be identical to that of the untransformed unit (unless the crystal is considered to be one dimensional). This input is simply an expedient for use in plotting helical polymer structure models with minimum input. In general, it would be possible to produce the same results by specifying the complete crystallographic asymmetric unit and normal crystallographic symmetry transformations.

This input mode is only meaningful if the cell angles α and β are 90°, so that $\mathbf{c}$ lies along $\mathbf{c}^*$ and the helix can continue uninterrupted from cell to cell along the $\mathbf{c}$ axis.

Columns	(a) Crystallographic symmetry (70-72 < 5)	(b) Helix symmetry (70-72 ≥ 5)
1	≠ 0 last card only	≠ 0 last card only
2-15	T_1	T_1
16-18	S_{11}	—
19-21	S_{12}	—
22-24	S_{13}	—
25-39	T_2	T_2
40-42	S_{21}	—
43-45	S_{22}	—
46-48	S_{23}	—
49-63	T_3	T_3
64-66	S_{31}	L
67-69	S_{32}	M
70-72	S_{33}	N

(a) Crystallographic symmetry: Transformed triclinic coordinates (X_1,Y_1,Z_1) are obtained from input triclinic coordinates (X,Y,Z) by

$$X_1 = T_1 + S_{11}X + S_{12}Y + S_{13}Z,$$
$$Y_1 = T_2 + S_{21}X + S_{22}Y + S_{23}Z,$$
$$Z_1 = T_3 + S_{13}X + S_{32}Y + S_{33}Z,$$

or in matrix notation

$$\mathbf{X}_1 = \mathbf{T} + \mathbf{SX},$$

where $\mathbf{T} = (T_1,T_2,T_3)$ as fractions of cell edges.

Only symmetry cards for *general* symmetry equivalent positions are permitted. Symmetry cards that explicitly designate special positions such as X,X,X; X,X,Z; $X,Y,0$; $1/4,Y,0$; and 1/4,3/4,0 are *not allowed*.

(b) Helix screw symmetry:

$$\mathbf{X}_1 = \mathbf{T} + \mathbf{SX},$$

where $\mathbf{T} = (T_1,T_2,T_3+L/N)$ as fractions of cell edges and $\mathbf{S}$ is a counterclockwise rotation of $L \cdot M/N$ cycles about the $\mathbf{c}^*$ axis.

Example: The Pauling and Corey right-handed alpha helix of poly-L-alanine repeats after 13 turns and 47 residues and can be represented by 47 symmetry cards with $N = 47$; $M = 13$; $L = 0, 1, \ldots, 46$; $T_1, T_2, T_3 = 0$. The input atom list then contains the contents of one residue. (See 7.2.)

Type 1 crystallographic symmetry cards do not have a specific format with the following two exceptions: (1) the symmetry information on each card must not go beyond column 72, and (2) column 1 must be "0" (or blank) on all symmetry cards other than the last one in the set, which must be non-zero. Below is an example set of Type 1 symmetry cards to illustrate the flexibility of this style.

```
 X,Y,Z
 X    -Y   Z+1/2
   X+0.5,   Y+.5,  Z
1x+1/2,-y+1/2,1/2+z
```

As shown, letters may be either upper or lower case. Commas or blanks may be used to separate the components of the triplet. The three components may not have blanks within them. Decimal fractions may be used with or without an initial 0. Fractions may precede or follow the letters.

Regardless of how the symmetry information is provided, the last card of the set must have a non-zero value in column 1 to signal the end of the symmetry cards. If the value is "1", the atom parameter information immediately follows in the ORTEP input file as described in Section 3.2.4. If the value is "2", the atom parameter information is read from a different file (see 4.5), and the ORTEP instructions follow the symmetry cards.

3.2.4 Atom Parameters

Two cards are required for each input atom. The first contains the chemical symbol , positional parameters, and feature information, if needed, The second contains temperature factor information or other information that specifies how the atom is to be represented on the drawing. Several alternate inputs are possible for each of the two cards, and the number in column 63 denotes the type used on that particular card. The number of atoms (variable NATOM) may range from 1 to 500.

3.2.4.1 Positional Parameters

The positional parameter cards have FORMAT (A6,3X,6F9.0).

Columns	Type 0	Type 1	Type 2	Type 3
1-6	Up to six alphanumeric characters centered in the six-place field			
7-9		—		
10-18	[Feature #1]	[Feature #1]	[Feature #1]	x_0 (Å, Cartesian)
19-27	[Feature #2]	[Feature #2]	[Feature #2]	y_0 (Å, Cartesian)
28-36	x (fractional, crystal)	x (Å, crystal)	x (Å, Cartesian)	r (Å, cylindrical)
37-45	y (fractional, crystal)	y (Å, crystal)	y (Å, Cartesian)	ϕ (°, cylindrical)
46-54	z (fractional, crystal)	z (Å, crystal)	z (Å, Cartesian)	z (Å, cylindrical)
63	0	1	2	3

Type 0 is the normal input based on triclinic coordinates. Coordinates in Angstroms along the unit cell vectors may be entered with Type 1. Type 2 may be used to place a model described in Cartesian coordinates onto a general triclinic lattice. The orientation of the Cartesian system *xyz* in the general lattice **abc** is the standard type described in Section 3.1.8 with x along **a** and z along **c***. Type 3 is similar to Type 2 except that cylindrical coordinates r, ϕ, z are used and the axis of the system can be displaced from zero in the xy Cartesian plane by the displacement x_0, y_0. Cylindrical coordinates are often used to describe helical structures. The x_0, y_0 translation should be zero if helical symmetry operators are used. This translation feature is meant to be used for explicitly describing the contents of a multiple helix cell.

Column fields 10-18 and 19-27 on Type 0, 1, and 2 positional parameter cards may be used in ORTEP-III to enter "feature" information about the atoms (see 3.1.10 and 4.6). Normally, these fields are ignored by ORTEP so any numeric values may be here or the fields may be blank. The information in these fields will be interpreted as atom features only if instructions are invoked that specifically look at atom features. Features cannot be entered on Type 3 positional parameter cards.

3.2.4.2 Temperature Factors

Temperature factor cards have FORMAT (I1,F8.0,5F9.0,7X,F2.0).

Columns	Type 0,1,2,3,10	Type 4,5,8,9	Type 6		Type 7	
1	A sentinel ≠ 0 for last atom only					
2-9	b_{11}	U_{11}	B	B	R	R
10-18	b_{22}	U_{22}	0	0	0	0
19-27	b_{33}	U_{33}	0	VDC_1 (from)	0	VDC_1 (from)
28-36	b_{12}	U_{12}	0	VDC_1 (to)	0	VDC_1 (to)
37-45	b_{13}	U_{13}	0	[VDC_2 (from)	0	[VDC_2 (from)
46-54	b_{23}	U_{23}	0	VDC_2 (to)]	0	VDC_2 (to)]
55-63	0,1,2,3,10	4,5,8,9	6 (or 0)	6 (or 0)	7	7

Anisotropic temperature factor Types 0, 1, 2, 3, and 10 use the following formula for the complete temperature factor.

$$\text{Base}^{-D(b_{11}h^2 + b_{22}k^2 + b_{33}l^2 + cb_{12}hk + cb_{13}hl + cb_{23}kl)}$$

The coefficients $b_{ij}(i,j = 1,2,3)$ of the various types are defined with the following constant settings.

Type 0: Base = e, $c = 2$, $D = 1$

Type 1: Base = e, $c = 1$, $D = 1$

Type 2: Base = 2, $c = 2$, $D = 1$

Type 3: Base = 2, $c = 1$, $D = 1$

Type 10: Base = e, $c = 2$, $D = 2\pi^2$

Anisotropic temperature factor Types 4, 5, 8, and 9 use the following formula for the complete temperature factor, in which $\mathbf{a}_1^*$, $\mathbf{a}_2^*$, $\mathbf{a}_3^*$ are reciprocal cell dimensions.

$$\exp[-D(\mathbf{a}_1^{*2}U_{11}h^2 + \mathbf{a}_2^{*2}U_{22}k^2 + \mathbf{a}_3^{*2}U_{33}l^2 + C\,\mathbf{a}_1^*\mathbf{a}_2^*U_{12}hk + C\,\mathbf{a}_1^*\mathbf{a}_3^*U_{13}hl + C\,\mathbf{a}_2^*\mathbf{a}_3^*U_{23}kl)]$$

The coefficients $U_{ij}(i,j = 1,2,3)$ of the various types are defined with the following constant settings.

Type 4:	$C = 2,\ D = {}^1\!/_4$
Type 5:	$C = 1,\ D = {}^1\!/_4$
Type 8:	$C = 2,\ D = 2\pi^2$
Type 9:	$C = 1,\ D = 2\pi^2$

Type 6 allows the input of the Debye-Waller isotropic temperature factor B, which is used as follows:

$$\exp(-B \sin^2 \theta/\lambda^2),$$

where λ is the wavelength and θ is the Bragg angle. The parameter B is related to mean-square displacement $\overline{\mu^2}$ of the atom from its mean position by the relation

$$B = 8\pi^2 \overline{\mu^2}.$$

When the isotropic temperature factor is used, the atom is represented as an isotropic ellipsoid (sphere) with equal principal axes of length μ. When the field in columns 19-27 is "0" or blank, the directions of the principal axes are along the standard Cartesian system axes (see 3.1.8). However, these arbitrary orthogonal vectors can be reoriented by using the two vector designator codes VDC_1 and VDC_2; then the three new principal-axis vectors will be VDC_1, ($VDC_1 \times VDC_2$), and $VDC_1 \times (VDC_1 \times VDC_2)$. This is strictly an artistic feature of no physical significance.

Type 7 allows the input of arbitrary spheres of radius $\overline{\mu} = R$ in Angstroms. The vector triplet orientation is specified as with Type 6. An additional capability allows a completely blank card (except perhaps column 1) to be used for a temperature factor card. In this case the program assumes Type 7 with an $R = 0.1$ Å.

If VDC_2 is omitted on Type 6 or Type 7 temperature factor cards, the program will choose one of the three lattice vectors for VDC_2.

A Type 10 temperature factor input card may be used to load Cartesian temperature factors having components in the standard Cartesian system (see 3.1.8). This capability complements the Type 3 Cartesian positional parameter input system (see 3.2.4.1) and is useful for plotting mean-square displacements caused by internal molecular motions as calculated from spectroscopic normal-coordinate analyses.[6,7]

New in ORTEP-III is the method for specifying the orientations and sizes of the elongated pass (cigar-shaped) and flattened pale (pancake-shaped) ellipsoids used in critical net illustrations (see 7.5) without giving their quadratic form coefficients. The temperature factor card following the atom parameter card for a pass or pale has the format shown below.

Columns	
1	A sentinel ≠ 0 for last atom only
2-9	Unique axis length (Å)
10-18	Second (and third) axis length (Å)
19-27	VDC_1 (from)
28-36	VDC_1 (to)
37-45	[VDC_2 (from)
46-54	VDC_2 (to)]
63	7

VDC_1 is a vector parallel with the unique axis of the cigar-shaped pass or pancake-shaped pale and VDC_2 is a second vector **not** parallel with VDC_1 such that $VDC_1 \times VDC_2$ is a second principal axis of that ellipsoid. If VDC_1 and VDC_2 are parallel, VDC_2 is replaced by a suitable lattice translation vector. VDC_2 may be omitted from the input if desired, and the program will choose one of the three lattice vectors for VDC_2.

3.3 INSTRUCTION INPUT

The instructions are the commands used in programming an illustration, and there is no required sequence for the instructions, except as indicated by the programming logic.

3.3.1 Instruction Format

There are four different formats used for ORTEP instructions, which are numbered 0, 1, 2, and 3. Format No. 0 is always used for a new instruction and includes the instruction number. Some instructions require trailer (continuation) cards, which use Format Nos. 1, 2, and 3. The program is informed what the format of the next card will be with the value in the "look ahead" field, column 3. The program action is also influenced by this information.

"Look ahead" on current card (column 3)	Next card will be	Program action
0 or blank	Format No. 0, a new instruction	Execute present instruction; then read next new instruction card
1	Format No. 1, continuation of present instruction	Read continuation card and then check its "Look ahead"
2	Format No. 2, continuation of present instruction	Read continuation card and then check its "Look ahead"
3	Format No. 3, alphanumeric information	Read alphanumeric information and execute instruction; then read next new instruction card

The Format No. 0 instruction card and the Format No. 1 parameter continuation card have the FORMAT (I3,I6,7F9.0), but the former includes the instruction number while the latter does not. A maximum of 19 Format No. 1 continuation cards per instruction is permitted.

Columns	**Format No. 0 Instruction Card**	**Format No. 1 Parameter Continuation Card**
3	"Look ahead" (0,1,2,3)	"Look ahead" (0,1,2,3)
4-9	Instruction number	Blank
10-18	1st parameter	8th parameter, or 15th, ..., or 134th
19-27	2nd parameter	9th parameter, or 16th, ..., or 135th
...	...	...
64-72	7th parameter	14th parameter, or 21st, ..., or 140th

The Format No. 2 trailer card is also used for parameter continuation, but it is more compact with FORMAT (I3,6X,5I3,8F6.0), allowing more information to be entered than on the Format No. 1 card. Many, but not all, Format No. 2 trailer cards include vector search codes (see 3.1.5), and they are often referred to as Vector Search Code (VSC) cards. A maximum of 20 Format No. 2 continuation cards per instruction is permitted.

Columns	**Format No. 2 Parameter Continuation Card**
3	"Look ahead" (0,1,2,3)
4-9	—
10-12	1st integer parameter
...	...
22-24	5th integer parameter
25-30	1st real parameter
...	...
67-72	8th real parameter

Format No. 3 trailer cards with FORMAT (18A4) are used for entering label information and are required for instructions 902, 903, and 913. As a trailer to these, they may contain up 72 characters of alphanumeric information in columns 1-72, centered about columns 36 and 37 or beginning in column 1 for automatic centering. These may also be used with other instructions as a device to transfer comments about the particular instruction to the ORTEP output listing. There is no "look ahead" column in Format No. 3; the next card must be a new instruction card.

3.3.2 Structure Analysis Instructions (100 Series)

This series of instructions is not connected with drawing illustrations but rather with obtaining a convenient tabulation of the chemically interesting aspects of a crystal structure, including interatomic distances and angles and principal axes of thermal motion. If the ORTEP output is omitted (see 4.1), these instructions do nothing.

3.3.2.1 Instructions 101 and 102

These instructions call subroutine SEARC, which finds all "target" atoms within a sphere of enclosure of radius D_{max} about a particular "origin" atom. The instruction card has an atom designator run (see 3.1.3) of one or several origin atoms (Org. ADR) and an atom number run of target atoms (Tar. ANR). The Org. ADR allows calculation of several spheres successively with a printout of results after each one.

Example: Suppose there are nine atoms in the input list and we want to find all atoms surrounding atoms 365502, 465502, and 565502 out to a maximum radius of 4 Å.

```
101   365502        5        1        9      4.0
```

The selection of the interatomic vectors from a particular origin atom to the target atoms can be narrowed with a vector search code on a Format No. 2 continuation card (see 3.3.1). The first line of the instruction defines a limiting sphere of vectors, and the VSC finds the subset of vectors that satisfy the additional restriction. (See 3.1.5.)

Example: Suppose we want to limit the results of the search in the preceding example to the shell of atoms that lie 2.0-2.7 Å from each of the origin atoms.

```
2   101   365502        5        1        9      4.0
          3  5  1  9  0   2.0   2.7
```

Vectors found about a particular origin atom are stored in a table of dimension 200 sorted on distance. Duplicate vectors (not duplicate distances) are eliminated. If more than 200 acceptable atoms are found about an origin atom, the 200 shortest vectors are saved. At the end of the search about each origin atom, the distances are printed out along with the atom designator codes, chemical symbols, and triclinic crystal coordinates for the origin and target atoms. If the instruction is 102, all possible interatomic angles and interatomic distances for the edges opposite the angles are also calculated and printed for the stored vectors. There will be $n(n—1)/2$ angles for n vectors. The tabulation of atom designator codes, which is obtained automatically when these instructions are given, is often useful for planning an illustration.

Columns	Instructions 101 and 102	Columns	Format No. 2 Vector Search Code (if used)
3	0, 1, or 2	3	0 or 2
7-9	101 or 102	4-9	—
10-18	Org. ADR (from)	10-12	[Org. NR (from)
19-27	Org. ADR (to)	13-15	Org. NR (to)]
28-36	Tar. ANR (from)	16-18	[Tar. NR (from)
37-45	Tar. ANR (to)	19-21	Tar. NR (to)]
46-54	D_{max} (Å)	24	[NR type]
		25-30	[D_{min} (Å)
		31-36	D_{max} (Å)]

Columns	Format No. 1 Continuation Card (if needed)
3	2
7-9	—
10-18	—
27	LOGC

3.3.2.2 Instruction 103

Principal axes of thermal motion (or arbitrary spheres, according to the temperature factor input) for all atoms in the input list are calculated. The printout contains root-mean-square amplitudes of displacement along the principal axes of the trivariate normal probability density function and direction cosines for the principal axes relative to the reference Cartesian base vectors. A symmetric covariance dispersion matrix based on the reference Cartesian system is also printed out. The diagonal elements are the mean-square displacements along the reference Cartesian axes.

Columns	Instruction 103
7-9	103
10-72	Blank

3.3.2.3 Instructions 105 and 106, Convoluting Sphere of Enclosure and Reiterative Convoluting Sphere of Enclosure

These instructions utilize the ATOMS array and thus can only be used after one or more atoms have been placed in the table by a $40n$ instruction (see 3.3.5). The contents of the array are returned to the condition of entry at the conclusion of instructions 105 and 106.

All atoms in the ATOMS array that have atom or feature numbers within the origin number run (Org. NR) of the instruction are used as origin points. Interatomic distances for all neighboring atoms (whether or not in the ATOMS array) are found out to the specified radius. Vector search codes on Format No. 2 continuation cards may be used for screening if desired. (See 3.1.5.)

Instructions 105 and 106 are similar, except that instruction 106 keeps repeating the "convolution" process until no new atoms with atom numbers within the Org. NR are found. Instruction 106 is useful for molecular structures where the atoms in the input asymmetric unit do not form an intact molecule. In a case of this nature, it is advisable to place a single atom into the ATOMS array with a 401 instruction and let a 106 instruction find the molecule. Care must be taken to specify a D_{max} that will enclose only bonded atoms. The 106 instruction should not be used for structures forming infinite chains. Instead, a short sequence of 105 instructions can be used for this case.

Columns	**Instructions 105 and 106**	Columns	**Format No. 2 Vector Search Code (if used)**
3	0, 1, or 2	3	0 or 2
7-9	105 or 106	4-9	—
10-18	Org. NR (from)	10-12	[Org. NR (from)
19-27	Org. NR (to)	13-15	Org. NR (to)]
28-36	Tar. NR (from)	16-18	[Tar. NR (from)
37-45	Tar. NR (to)	19-21	Tar. NR (to)]
46-54	D_{max} (Å)	24	[NR type]
63	NR type	25-30	[D_{min} (Å)
		31-36	D_{max} (Å)]

Columns	**Format No. 1 Continuation Card (if needed)**
3	2
7-9	—
18	—
27	LOGC

3.3.3 Plotter Control Instructions (200 Series)

The 200 series is a group of instructions that control the device, real or virtual, that receives the information to output the ORTEP illustration. This may be, for example, an actual hardware plotter, a computer monitor, or a file of page description language instructions that will produce a hard copy of the illustration when the file is downloaded to a printer/plotter. Because no single word can describe all of the possibilities, the word "plotter" will be used in this report.

The 200 instructions control plotter initialization, plot origin, color control, termination, and any other peripheral commands that are required for a particular equipment configuration or plotting package. When the program is modified for a different equipment configuration, the series 200 instructions, which are executed through subroutine F200, should be redefined to suit the user's requirements.

The 201 instruction initializes plotting and must be executed before any plotting can take place. It is a safe policy always to make this the first instruction card. If the 201 and 301 instructions are omitted, all calculations are carried out but no plotting is done. The 202 instruction with no parameters is used to terminate plotting. With parameters, the 202 instruction may be used to shift the plot origin. The 201/202 (with no parameters) form an initialize/terminate pair and must always appear in pairs in the ORTEP input file. The pair *may occur more than once* (unlike ORTEP-II[2]). Each call to 201 begins a new plot page.

Instruction 204 is used to change the color of (or plotter pen used for) subsequent plotting. The default value is 0 for black (or pen #1). The screen and Postscript drivers built into ORTEP-III define color value 2 as red, 3 as green, 4 as blue, 5 as cyan, 6 as magenta, and 7 as yellow. (The value 1 is also black or pen #1.)

Instruction 205 is used to vary the thickness of subsequently plotted lines. The unit for width is thousandths of an inch, and the default value is 5. For both the 204 and 205 instructions, if columns 10-18 contain "0" or are blank, the default settings are restored.

Columns	**Instruction 201**	**Instruction 202**		**Instruction 204**	**Instruction 205**
		Shift plot origin	Terminate plotting		
7-9	201	202	202	204	205
10-18	—	Shift along x (in.)	—	Color (or pen #)	Pen width
19-27	—	Shift along y (in.)	—	—	—

3.3.4 Drawing Parameter Instructions (300 Series)

This is a set of instructions for specifying the drawing dimensions, viewing distance, general lettering orientation, pen displacement for line retracing, and ellipse smoothness.

3.3.4.1 Instruction 301, Drawing Boundaries, Margin, and Viewing Distance

This instruction defines the maximum x and y dimensions of the drawing and the margin, all in inches. The boundary dimensions must not exceed those allowed by the plotter. The program will prevent the pen from getting closer than 0.1 in. to any boundary. The drawing margin is a constant width border inside the entire boundary. When automatic scaling is used (600 series), the center points of the atoms are prevented from falling in the margin; but the atom representation, which has a finite size, may extend into that area. To accommodate this extension into the margin, the margin width should be large when the overall drawing scale and the ellipsoid scale are expected to be large.

In addition, instruction 301 specifies the perspective viewing distance, in inches, from the plane of the drawing. An entry of "0" for view distance is used to indicate an infinite view distance, and the crystal structure is then mapped in parallel projection normal to the drawing board.

Columns	**Instruction 301**	Primer Constant
7-9	301	—
10-18	Drawing x_{max} (in.)	10.5
19-27	Drawing y_{max} (in.)	8.0
28-36	View distance (in.)	0. (parallel projection)
37-45	Margin width (in.)	0.5

3.3.4.2 Instruction 302, Title Rotation

For regular titles and chemical symbols, the title rotation is specified with instruction 302. The lettering base line for all lettering is rotated counterclockwise by an angle theta, in degrees, from the x axis of the plotter. Although any value is allowed, 0° and -90° are the values most often used so that the finished drawing has either the y plotter axis vertical or the x plotter axis vertical.

Columns	**Instruction 302**	Primer Constant
7-9	302	—
10-18	θ (°)	0.

3.3.4.3 Instruction 303, Retrace Displacement

Instruction 303 directs certain lines to be made heavier than others by retracing over the path several times with slight pen displacements (DISP) from the original path. For example, if DISP is greater than 0., the forward half of the principal plane trace of ellipsoids is drawn heavier than the hidden half so the eye does not confuse the two halves. Also, all regular lettering (but not perspective lettering) is gone over four times to give it a boldface appearance. In ORTEP-III, retracing is turned off by default with the primer constant for DISP set to 0. in. Retracing can be turned on with the 303 instruction, using a value for DISP greater than 0. (0.005 in. usually works well).

Columns	**Instruction 303**	Primer Constant
7-9	303	—
10-18	DISP (in.)	0.

3.3.4.4 Instruction 304, Ellipse Smoothness

Instruction 304 may be used to adjust the smoothness of the plotted ellipses as a function of their size. Smoother ellipses require more computational time and produce larger illustration files. (See 5.2.) Smaller smoothness factors produce smoother ellipses, and larger values produce ellipses with more "jaggies". Factors in the range 0.5–3.0 are recommended; the default value is 1. A value of 0 will draw all size ellipses with 128 points, the maximum provided in ORTEP.

Columns	**Instruction 304**
7-9	304
10-18	Smoothness factor

3.3.5 ATOMS Array Instructions (400 Series)

This series allows the user to specify which atoms are to be included in the illustration. The atom designators for the chosen atoms are stored in the ATOMS array for future use by other instructions. The ATOMS array as currently dimensioned holds 500 input atoms, but it can be changed by redimensioning the array in COMMON and setting the variable NATOM in subroutine PRIME to the new value.

Groups of atoms are added to or eliminated from the ATOMS array (which is set to zero at the start of the program) with the 40*n* and 41*n* series, respectively. The groups can be denoted by atom designator runs (see 3.1.3), spheres of atoms about any center point (see 3.1.6), and boxes of atoms centered on any point (see 3.1.7). Duplicate entries of the same atomic position are prevented by the program. The content of the ATOMS array is printed in the ORTEP output file after each 400 series instruction.

3.3.5.1 Instructions 401/411, Atom Designator Run Add/Eliminate

These instructions may contain: *(a)* atom designator codes for single atoms; (*b*) atom designator runs for several atoms in a run; (*c*) blank fields (except between the two entries of a run); and *(d)* any combinations of (*a*), (*b*), and (*c*). Since up to 19 Format No. 1 continuations are possible per instruction, up to 70 runs can be made per instruction and an unlimited number of instructions can be used. The ATOMS array and overlap calculation, however, will only accept the first 500 atoms.

Columns	**Instructions 401/411**
3	0 (or 1 if continued)
7-9	401 or 411
10-18	
.	
.	As described above
.	
64-72	

3.3.5.2 Instructions 402/412, Sphere of Enclosure Add/Eliminate

These instructions allow the user to build or modify the ATOMS array by specifying the contents (complete, partial, or vector screened) of a sphere of enclosure (see 3.1.6) about any addressable point. Instruction 402 adds the contents of the spheres to the ATOMS array omitting positional duplications. The 412 instruction eliminates atoms within the spheres from the ATOMS array if they are present in that array. The instructions call subroutine SEARC, and the instruction input details are identical to those of instructions 101 and 102 (see 3.3.2.1) except for the instruction number. In the ORTEP output file, only the ATOMS array atom designator codes are printed and not the coordinates and interatomic distances. If the origin atoms on which the spheres of enclosure are centered are to be saved, the target atom number run (Tar. ANR) must

contain those atom numbers. Vector search codes on Format No. 2 continuation cards may be used for screening if desired. (See 3.1.5.)

Columns	**Instructions 402/412**
3	0, 1, or 2
7-9	402 or 412
10-18	Org. ADR (from)
19-27	Org. ADR (to)
28-36	Tar. ANR (from)
37-45	Tar. ANR (to)
46-54	D_{max} (Å)

Columns	**Format No. 2 Vector Search Code (if used)**
3	0 or 2
4-9	—
10-12	[Org. NR (from)
13-15	Org. NR (to)]
16-18	[Tar. NR (from)
19-21	Tar. NR (to)]
24	[NR type]
25-30	[D_{min} (Å)
31-36	D_{max} (Å)]

Columns	**Format No. 1 Continuation Card (if needed)**
3	2
7-9	—
10-18	—
27	LOGC

3.3.5.3 Instructions 403/413 and 404/414, Box of Enclosure Add/Eliminate and Triclinic Box of Enclosure Add/Eliminate

The 403/413 instructions allow the user to build or modify the ATOMS array by specifying the contents of a box of enclosure (see 3.1.7) about any addressable point (or atom designator run of addressable points). The three axes of the box are parallel to the three base vectors of the reference Cartesian system, and the semidimensions of the box are specified on the instruction card. If an orientation of the box different from the standard orientation (see 3.1.8) is desired, then a 501 or a 502 instruction, or both, should be used before this instruction to reorient the reference Cartesian system. After this instruction has been executed, the reference system can undergo further reorientation as desired for plotting purposes, etc.

The 404/414 instructions are identical to the 403/413 instructions except that the triclinic box of enclosure is bounded by planes parallel to the principal planes of the crystal lattice. The semidimensions $a/2$, $b/2$, $c/2$ refer to fractional (triclinic) coordinates. To specify the contents of the conventional unit cell, one would use $a/2 = b/2 = c/2 = .5$, and the Org. ADR would refer to a point in the input atom list at (.5,.5,.5).

As in the case of the 402/412 instruction, the origin atom on which the box is centered will not be included unless the target atom number run includes the origin atom number. Subrou-

tine SEARC is used by this instruction, and the instruction input details are similar to those of instructions 402/412 except that D_{max} is replaced by the semidimension $a/2$ of the box and the following fields on the card are used to specify the other two semidimensions $b/2$ and $c/2$. The box dimensions must be chosen carefully so that the ATOMS array does not overflow. Vector search codes on Format No. 2 continuation cards may be used for screening if desired. (See 3.1.5.)

Columns	**Instructions 403/413 and 404/414**	Columns	**Format No. 2 Continuation Card (if used)**
3	0, 1, or 2	3	0 or 2
7-9	403, 404, 413, or 414	4-9	—
10-18	Org. ADR (from)	10-12	[Org. NR (from)
19-27	Org. ADR (to)	13-15	Org. NR (to)]
28-36	Tar. ANR (from)	16-18	[Tar. NR (from)
37-45	Tar. ANR (to)	19-21	Tar. NR (to)]
46-54	*a/2* (Å or fraction)	24	[NR type]
55-63	*b/2* (Å or fraction)		
64-72	*c/2* (Å or fraction)		

Columns	**Format No. 1 Continuation Card (if needed)**
3	2
7-9	—
10-18	—
27	LOGC

3.3.5.4 Instructions 405/415 and 406/416, Convoluting Sphere of Enclosure Add/Eliminate and Reiterative Convoluting Sphere of Enclosure Add/Eliminate

These instructions are used in the same manner as instructions 105 and 106 (see 3.3.2.3). Their function is to add atoms to or eliminate atoms from the ATOMS array. A valid origin atom must be placed in the ATOMS array with a 401, 402, 403, or 404 instruction before these instructions are used. All atoms in the ATOMS array that have atom or feature numbers within the origin number run (Org. NR) of the instruction are used as origin points of convolution. Vector search codes on Format No. 2 continuation cards may be used for screening if desired. (See 3.1.5.)

An example use for the 405 instruction is to complete the coordination shells around metal atoms without having to identify the coordination shell atoms individually. Another use might be to obtain a cluster of atoms out to the nth neighbor when only the distance to the first neighbor is known. This can be accomplished by using n consecutive 405 instructions with D_{max} slightly more than the first neighbor distance.

Instruction 406 is useful for describing molecular compounds where an unfortunate choice of atoms for the input asymmetric unit does not allow the molecule to be described by a run. D_{max} must be chosen judiciously so that the search does not cross molecular boundaries.

Instruction 406 has an added feature that makes it useful for limiting the ATOMS array to those atoms in an asymmetric unit of the unit cell. An example is the examination of electron density maps. When direct methods are used for solving crystal structures, stereoscopic drawings provide a rapid means for screening the E maps. If the interpolated positions of the largest peaks in the Fourier synthesis E map are taken as atom positions, then ORTEP can start at a given peak (usually the largest) and do a reiterative sphere-of-enclosure add to isolate a molecule if one is present. In order to terminate the convolution procedure when extraneous "bridging peaks" link the molecules, a feature has been added to the 406 instruction to prevent multiple instances of an atom (in any of its equivalent positions) from being entered in the ATOMS array. To invoke this feature, a Format No. 1 continuation card with a "1" in column 18 (ASYMUNIT) is added to the 406 instruction. The 406 instruction operates in its usual manner if the continuation card is omitted or has a "0" or blank in column 18.

Columns	**Instructions 405/415 and 406/416**
3	0, 1, or 2
7-9	405, 406, 415, or 416
10-18	Org. NR (from)
19-27	Org. NR (to)
28-36	Tar. NR (from)
37-45	Tar. NR (to)
46-54	D_{max} (Å)
63	NR type

Columns	**Format No. 2 Vector Search Code (if used)**
3	0 or 2
4-9	—
10-12	[Org. NR (from)
13-15	Org. NR (to)]
16-18	[Tar. NR (from)
19-21	Tar. NR (to)]
24	[NR type]
25-30	[D_{min} (Å)
31-36	D_{max} (Å)]

Columns	**Format No. 1 Continuation Card (if needed)**
3	2
7-9	—
18	ASYMUNIT (406 only)
27	LOGC

3.3.5.5 Instruction 410, Clear ATOMS Array

This instruction clears the ATOMS array to zero. When the program is first entered, the array is automatically set to zero.

Columns	Instruction 410
7-9	410

3.3.6 Orienting Instructions (500 Series)

For information on instruction 511, see Section 3.3.12.

Any Cartesian coordinate system is based on three orthonormal base vectors and an origin point. In the absence of any 500 instructions, ORTEP calculates the base vectors of the reference and working Cartesian systems (see 3.1.8) from the input cell parameters and sets the origin to (0.,0.,0.).

This series of instructions can be used to reorient the reference and working Cartesian systems. Each time the reference system is redefined or rotated, the working system is automatically made coincident with the reference system. The working system can be displaced from the reference system by rotating about the x or y axis of the reference system with a 503 instruction. The working system is always positioned from the reference system and does not depend on any previous working system orientation. After each 500 series instruction, the base vectors of the relevant Cartesian system are printed in the ORTEP output file. These vectors are based on the triclinic coordinate system. The postfactor transformation matrix for converting from triclinic coordinates to Cartesian coordinates is also printed out. The inverse transformation matrix may be formed by placing the three base vectors together in row vector form.

3.3.6.1 Instruction 501, Explicit Reference Cartesian System Assignment

Instruction 501 allows the user to define the reference Cartesian system explicitly. The origin point in the model (ORGN) is specified with an atom designator code. The three orthonormal base vectors can be described by two non-collinear vectors, and ORTEP provides the two following separate techniques for performing this operation, using vector cross products of the two vectors **u** and **v**. Type 1 produces base vectors that are roughly along the general triclinic coordinate axes of the crystal.

	Type 0	Type 1
Base vector 1 (x axis)	$\mathbf{u}$	$\mathbf{u}$
Base vector 2 (y axis)	$\mathbf{u} \times \mathbf{v}$	$(\mathbf{u} \times \mathbf{v}) \times \mathbf{u}$
Base vector 3 (z axis)	$\mathbf{u} \times (\mathbf{u} \times \mathbf{v})$	$\mathbf{u} \times \mathbf{v}$

The reference system x and y axes will parallel the plotter x and y axes, and the origin point ORGN will lie in the plane of the plotter. The viewer will be looking into the z axis vector of the coordinate system from the viewing distance set by the 301 instruction (see 3.3.4.1) directly above the origin point.

Columns	**Instruction 501**	Effective Primer Constant
7-9	501	—
10-18	ORGN (ADC)	000000
19-27	Vector **u** (from ADC)	155501
28-36	Vector **u** (to ADC)	165501
37-45	Vector **v** (from ADC)	155501
46-54	Vector **v** (to ADC)	156501
55-63	—	—
72	Type	1

3.3.6.2 Instruction 502, Reference Cartesian System Rotation

The crystal model can be given any desired orientation with a series of rotations of the model about the reference system axes. In general, three rotations (e.g., those of an Eulerian system) are sufficient to achieve any orientation, but for convenience an unlimited number of rotations are permitted by ORTEP. In addition, rotations of 120° about the body diagonal of the reference Cartesian system are permitted (achieved by a cyclic permutation of reference base vectors).

Each operation requires two fields on the instruction card. For axial rotations, the first field of each pair will have the number 1, 2, or 3 to indicate rotation about the x, y, or z axis of the reference system, respectively. The second field will have the rotation angle in degrees for a right-handed rotation of the model about the designated axis (i.e., a positive angle signifies a counterclockwise rotation of the structure with the designated axis pointing toward the observer). The body diagonal rotation is designated by either a -1 or a -2 in the first field to indicate a 120° or a 240° right-handed rotation about the body diagonal, respectively, and the second field is blank. A -3 would rotate the structure completely around and thus not change its previous orientation. If desired, each rotation can be executed with a separate 502 instruction card.

Columns	**Instruction 502**
3	0 (or 1 if continued)
7-9	502
17-18	1, 2, 3, -1, or -2
19-27	ϕ (°) (if value in previous field is positive)
28-36	[...]
37-45	
46-54	[...]
55-63	

3.3.6.3 Instruction 503, Working Cartesian System Rotation (Stereoscopic Rotation)

To define an orientation of the working system that is not coincident with the reference system, a 503 instruction may be used, which allows one rotation about one axis of the reference

system. Actually any number of successive rotations can be made, but the effect is not cumulative since the starting point for each rotation is always the reference system. Body diagonal rotations are not permitted.

A 503 rotation normally precedes each member of a stereoscopic pair of plots. (See 3.3.6.4 and 5.4.) The rotation is about axis 2 if the stereo pair is to be viewed with the x axis parallel to the observer's interocular line and about axis 1 if the y axis is to be parallel to that line.

Columns	**Instruction 503**
7-9	503
18	1 or 2
19-27	ω (°)

3.3.6.4 Instruction 504, Reference Cartesian System Origin Translation

Instruction 504 is used to translate the origin of the reference Cartesian system along the x, y, and z axes of the reference system. Stereo by translation of origin can be achieved with instruction 504, which may be used in place of the 503 instruction. However, the 504 instruction should not be used when the ellipsoids have internal structure because the octants selected for shading may not be the same on both stereo views. (See 5.4.)

Columns	**Instruction 504**
7-9	504
10-18	Translation along x axis (in.)
19-27	Translation along y axis (in.)
28-36	Translation along z axis (in.)

3.3.6.5 Instruction 505, Reference Cartesian System Origin at Centroid

This instruction finds the first moment (i.e., centroid or center of gravity) of the atoms in the ATOMS array and makes this point the origin point (ORGN) of the reference and working coordinate systems. The base vectors of the coordinate systems are unchanged from their previous values. A weighting scheme and screening may be applied to the atoms used to calculate the centroid by using Format No. 2 trailer cards. If no trailer cards are used, all atom positions are entered with unit weights.

Columns	**Instruction 505**
3	0 or 2
7-9	505

Columns	**Format No. 2 Continuation Card (if used)**
3	0 or 2
10-12	NR (from)
13-15	NR (to)
24	NR type
25-30	Weight

3.3.6.6 Instruction 506, Reference Cartesian System Origin at Centroid and Inertial Axis Coordinate System

The calculation described for instruction 505 is performed; then the second moment matrix about the centroid is calculated, and the reference and working coordinate systems are set up along the principal axes of this matrix. This principal axis system of coordinates is along the inertial axes of the configuration of atoms in the ATOMS array. The x axis is along the long axis of the configuration (i.e., the minimal axis of inertia) and the z axis is along the short axis of the configuration (i.e. the maximal axis of inertia). The overlap along the z-view direction is often minimized by this option. Furthermore, the xy plane is the least-squares best plane for the atomic configuration. Format No. 2 trailer cards may be used to supply weights and screening.

Columns	**Instruction 506**	Columns	**Format No. 2 Continuation Card (if used)**
3	0 or 2	3	0 or 2
7-9	506	4-9	—
		10-12	NR (from)
		13-15	NR (to)
		24	NR type
		25-30	Weight

3.3.7 Positioning and Scaling Instructions (600 Series)

These instructions are used to direct the placement of the origin point ORGN (specified by instruction 501, 505, or 506) onto the drawing (dimensioned by instruction 301). In addition the three-dimensional assembly of atoms (chosen by the 400 series instruction) constituting the model is scaled. The atomic centers of the model will then be hanging in space above and below the drawing board, correctly positioned to be projected from the eye point described with instruction 301.

A second scaling parameter SCAL2 scales the ellipsoid (or sphere) size. It is a dimensionless scale factor ratio used to modify all rms displacement values before plotting equi-probability ellipsoids or spheres. A listing of SCAL2 values vs. probability is given in Table 6.1. The primer constant for SCAL2 is 1.54, corresponding to 50% probability. ORTEP-III saves the user the time of looking up the SCAL2 value corresponding to a desired probability. If the value entered in columns 37-45 is a negative whole number in the range -1 to -99, it is interpreted as the probability of the ellipsoids or spheres, and the value of SCAL2 is set by the program by table lookup, using the values in Table 6.1.

3.3.7.1 Normal Modes of Positioning and Scaling

Several normal modes of operation are available to the user for positioning and scaling the model. Instruction 601 requires the user to supply a complete explicit description of position (X0,Y0) and scale (SCAL1). At the other extreme, instruction 604 automatically scales and positions the model so that the peripheral projected atom centers will touch two opposite borders and the peripheral atoms in the remaining dimensions will be centered on the drawing. An intermediate mode is available through 602, which provides automatic scaling after explicit positioning.

This usually allows only one edge of the model to touch a border. Finally, instruction 603 requires an explicit scale and does automatic centering. In general, the 604 instruction is the easiest and safest one to use, but situations arise in which the user should not relinquish control to the program.

If the instruction's entry for X0, Y0, SCAL1, or SCAL2 is "0" or blank, the primer constant value is used. This means that an X0 or Y0 cannot be entered as exactly zero. If zero is wanted, a small nonzero number should be entered.

Columns	**Instruction 601**	**Instruction 602**	**Instruction 603**	**Instruction 604**	Primer Constant
7-9	601	602	603	604	—
10-18	X0	X0	—	—	8.5
19-27	Y0	Y0	—	—	5.5
28-36	SCAL1	—	SCAL1	—	1.0
37-45	SCAL2 or probability	SCAL2 or probability	SCAL2 or probability	SCAL2 or probability	1.54

3.3.7.2 Incremental Modes of Positioning and Scaling

Additional flexibility is provided through the incremental instructions 611, 612, and 613. These allow the user to "nudge" the model or modify the scale factor SCAL1, or both, after the parameters have been initially set with a previous 600 series instruction. The 611 instruction adds ΔX0, ΔY0 to the previous X0, Y0 position for the ORGN placement and multiplies the existing SCAL1 by ΔSCAL1 (except if ΔSCAL1 = 0, SCAL1 is unmodified). Instruction 612 increments the position and then does an automatic scaling; 613 first increments the scale (by multiplying by ΔSCAL1) and then automatically repositions.

A conservative general approach is to follow a 604 with a 611 having ΔX0 = 0, ΔY0 = 0 and ΔSCAL1 = 0.9. This will simply reduce the scale 10% about the origin so that there is more space for labels, etc.

Columns	**Instruction 611**	**Instruction 612**	**Instruction 613**
7-9	611	612	613
10-18	ΔX0	ΔX0	—
19-27	ΔY0	ΔY0	—
28-36	ΔSCAL1	—	ΔSCAL1
37-45	SCAL2 or probability	SCAL2 or probability	SCAL2 or probability

3.3.8 Atom Plotting Instructions (700 Series)

These instructions produce various representations of the atom based on the familiar ball-and-stick molecular model. In the general case, the ball is an ellipsoid representing a contour surface of equal probability density of thermal motion displacement. Alternatively, when thermal

motion is not being portrayed, the ball can be a sphere of arbitrary dimension. The 700 series also has provision for labeling the atomic site with the corresponding chemical symbol. The instructions in this series draw the atoms in the ATOMS array that project onto the usable part of the drawing area, defined with the 301 instruction. Atoms found to be out of bounds are bypassed, and a Fault Message (NG = 10) (see 3.4) is printed in the ORTEP output file. An atom is out of bounds under the following conditions: (1) its z coordinate in the scaled reference Cartesian system is greater than 1/2 the viewing distance, (2) its center after projection falls outside the limiting boundary of the drawing board, or (3) its projected center falls in the outermost 3/4 of the drawing margin. (See 3.3.4.1.)

An ellipsoid, for graphical purposes in ORTEP, is considered to be composed of ellipses and straight lines. The ellipses are of two types—principal ellipses and boundary ellipses. Relative to the viewpoint, a principal ellipse is further subdivided into a front half and a back, or hidden, half. There are three principal ellipses per ellipsoid, corresponding to the three principal planes. The boundary ellipse is the edge of the ellipsoid as seen from the viewpoint. The front and back halves of the principal ellipses meet at the boundary ellipse. The straight line segments of the ORTEP ellipsoid are the forward principal axes, reverse principal axes, and octant-shading lines.

Fig. 3.1 shows various combinations of these elements along with the ORTEP instruction number and parameter values to produce each. It is obvious that certain of these combinations are better representations than others. Instructions 701, 702, 703, and 706 generate specific ellipsoid types as shown in Fig. 3.1. Instruction 704 draws the boundary ellipse alone. If an atom is entered as a sphere, the boundary will be circular before projection and slightly elliptical after perspective projection. Instruction 705 allows the user to make up any representation from the basic components.

Chemical symbols up to six alphanumeric characters in length may be included with each atom in the ORTEP input file. These symbols can be put onto the illustration with one 700 series or several 900 series instructions. The 700 series places the center of the six-character field of each atom in the same position relative to the atom center for all the drawn atoms; the 900 series allows the user to position each symbol individually. The 700 series requires only three parameters as follows: (1) symbol height in inches, (2) parallel (left/right) offset in inches, and (3) perpendicular (up/down) offset in inches. The parameters refer to the model before projection, and they will change slightly during perspective projection. The parallel and perpendicular offset refer to the exact center of the six-character input field and are relative to the lettering base line set up with the 302 instruction. A symbol height of "0" or blank will cause the symbol drawing routine to be bypassed. If NPLANE = 0 in a 705/715 instruction and the symbol height is greater than zero, then chemical symbols alone are drawn on the atomic sites.

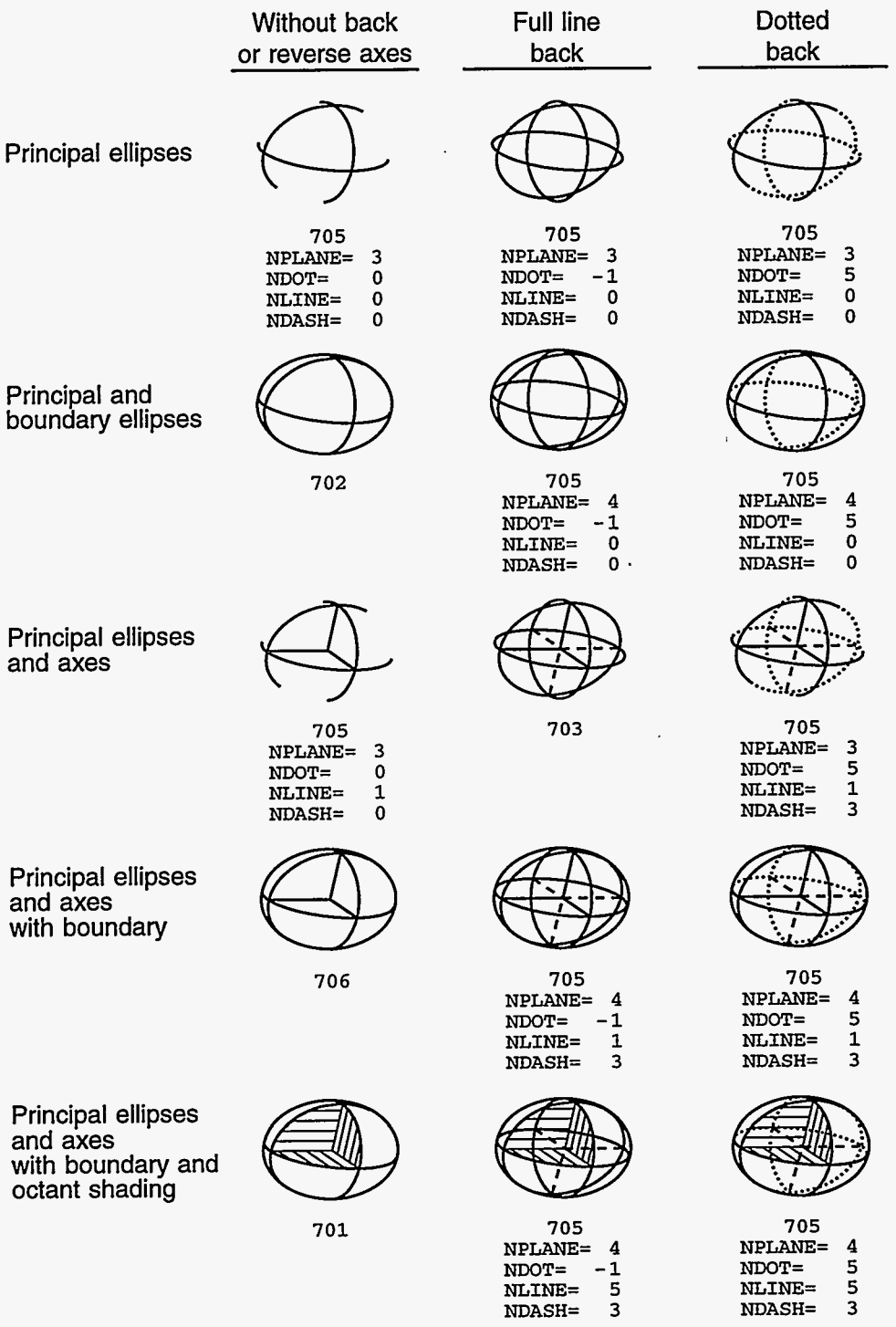

Fig. 3.1. Various combinations of ellipsoid components showing ORTEP instruction number and parameter values to produce each.

It is possible to vary the thickness of the boundary ellipse line by making it a function of z, the height of the atom from the drawing board. This option is normally used with the 704 (boundary only) instruction but will work for any 700 instruction. Entries are put in the A_0 and A_1 fields of the instruction continuation card to specify the coefficients of

$$\Delta R(z) = A_0 + A_1 z ,$$

where

ΔR is the increase in radial dimension to be added to the width of the single pen line,
A_0 is ΔR for an atom at $z = 0$, and
A_1 is the rate of increase in radial dimension with z.

Example: Assume the atoms of a scaled model range from 5 in. below to 5 in. above the drawing board and the pen width is 0.2 mm (.008 in.). If we want the closest ellipse boundary to be five times as thick as the farthest, then $\Delta R(-5 \text{ in.}) = 0$, $\Delta R(5) = 0.008 \times (5 - 1) = 0.032$ in.; thus $A_0 = 0.016$ in. and $A_1 = 0.0032$ in.

The program widens the line by stepping radially in increments of DISP, which is set by primer constant to 0. in. A 303 instruction must be used to give DISP a positive value for this line thickening process to work.

Selected types of atoms from the ATOMS array can be drawn without having to alter the contents of the array. This is accomplished by using a number run (NR) code that includes the atom types that are to be drawn with a particular 700-series instruction. This feature is particularly useful when two or more different atom representations are used, such as for the carbon and hydrogen atoms in the cubane example. If no NR is entered, then all atoms in the array are drawn.

The ORTEP file output for the 701–706 instructions consists of the following:

- x,y plotter coordinates: the coordinates, in inches, for the projected atom center on the plot, measured from the lower left-hand corner of the limiting boundary. This is the fixed plotter coordinate system with the origin set by the plotter driver.

- x,y,z working Cartesian coordinates: the coordinates, in inches, for the oriented and scaled atomic model before projection. The x and y axes parallel the plotter x and y axes, and the origin of the system is in the plane of the plotter at the point x_0,y_0 (see 3.3.7) in plotter coordinates. The point ORGN of the scaled model is at this point (see 3.3.6.1).

- x,y,z triclinic coordinates, in fractions of the unit-cell edges relative to the crystal unit-cell origin.

- Principal axes of thermal motion, consisting of (a) principal values of root-mean-square displacement and (b) direction cosine for principal vectors relative to the working Cartesian system.

- The atom designator code and chemical symbol for the atom.

Instructions 711–716 are identical to 701–706 except that the 71n series suppresses all ORTEP file output except fault messages.

Columns	Instructions 701, 702, 703, 704, 706, 711, 712, 713, 714, 716	Instructions 705 and 715
3	1 (if atom selection or boundary retracing is desired; otherwise 0)	1 (if atom selection or boundary retracing is desired; otherwise 0)
7-9	701, 702, 703, 704, 706, 711, 712, 713, 714, or 716	705 or 715
10-18	—	NPLANE = 0: no ellipsoid components = 1: boundary ellipse only = 3: principal ellipses only = 4: boundary + principal ellipses
19-27	—	NDOT (back side of principal ellipses) < 0: solid line back side = 0: back side omitted = 3: 4 dots on back side = 4: 8 dots on back side = 5: 16 dots on back side = 6: 32 dots on back side
28-36	—	NLINE (forward principal axes and shading) = 0: no forward axes or shading = 1: forward principal axes only = *N*: forward axes + (*N*–1) line shading
37-45	—	NDASH (dashed reverse principal axes) = 0: no reverse axes = *N*: dashed reverse axes with *N* dashes
46-54	Symbol height (in.)	Symbol height (in.)
55-63	Parallel offset (in.)	Parallel offset for symbols (in.)
64-72	Perpendicular offset (in.)	Perpendicular offset for symbols (in.)

Format No. 1
Parameter Continuation

Columns	(if used)
10-18	[A_0 (in.)]
19-27	[A_1 (in.)]
28-36	[NR (from)
37-45	NR (to)]
54	[NR type]

3.3.9 Bond Plotting Instructions (800 Series)

The bond plotting instructions are grouped into two general types, explicit and implicit, depending on how the bonds are specified. Explicit bonds require a vector designator code (see 3.1.2) for each bond. Implicit bonds make use of vector search codes (see 3.1.5) to find pairs of atoms from the ATOMS array set up by the 400 series instructions.

There are two types of bonds that can be drawn, line bonds and stick bonds. The line bond is a very crude, but rapid, method useful in drawing preliminary illustrations. It is constructed by placing the symbol ✳ on the two atom sites and drawing a single straight line between them. Line bonds are always specified implicitly.

The more elaborate bond is the stick bond, which could also be called a conical bond because of its accentuated perspective taper. (The accentuated taper may be increased or decreased by changing the value assigned to TAPER in subroutine PRIME.) Each end of the bond intersects either an ellipsoid or an enveloping cone (tangent cone) that has its apex at the viewpoint and is tangent to an ellipsoid. In general, the ellipsoid intersection is automatically used if the axis of the bond intersects the ellipsoid at a point that is visible to the viewpoint; otherwise, the tangent cone intersection is used, so that the bond appears to terminate at the boundary of the ellipsoid. However, the user can specify that the ellipsoid intersection always be used in order to make the skeleton type model (such as produced by the 703 instruction) appear even more transparent. The radius of the stick bond and the number of lines used to draw the bond are specified by input parameters.

Bond-distance labels can be drawn automatically with stick bonds but not with line bonds. The bond-distance label numbers are in Angstrom units with one, two, or three places past the decimal point. The bond labels on the illustration will have their base lines parallel to the stick bonds and will be right side up for the viewer. The height of the label in inches and the perpendicular offset distance for the center of the label relative to the center of the bond are parameters to be specified by the user. With the current primer constant for FORE, if the sine of the angle between the bond and the mean viewing vector is greater than 0.5, the lettering is done in perspective along the bond. When the sine of the angle is less than 0.5, the perspective lettering would be excessively foreshortened; the lettering is then made parallel to the plane of the drawing with its base line parallel to the projected bond. Different lettering heights and different perpendicular offset distances can be assigned to the perspective and nonperspective bond-distance labels.

All 800 instructions require Format No. 2 trailer cards to provide a number of bond parameters as follows:

- Bond type (for stick bonds) is designated by an integer NBOND, where $-5 \leq \text{NBOND} \leq 5$. The negative integers denote that both ends of the bonds terminate at the ellipsoids. The positive integers denote bonds ending either at the ellipsoid or the tangent cone, as described previously in this section. An entry of zero draws no bond. A magnitude of 1 for NBOND produces two lines, one for each bond edge, 180° apart in the plane normal to the bond axis. Lines are drawn 90°, 45°, 22.5°, or 11.25° apart for NBOND magnitudes of 2, 3, 4, or 5, respectively. The back side of the bond is not drawn.

- The bond radius (mean value for stick bonds) is in Angstrom units. Values between 0.01 and 0.06 Å usually give good results. Any positive value may be used as long as it is smaller than the scaled ellipsoid minimum semi-dimension. The bond radius is not changed by the ellipsoid scale factor ratio SCAL2. The bond radius should not be made "vanishingly small" (e.g., r < 0.005 Å) if the overlap feature (see 3.3.12) is used because numerical rounding may cause incorrect hidden-line elimination.

- The height of perspective labels for bond distances is entered as zero if no bond distances are to be labeled. Positive values denote the lettering height in inches before projection.

- The perpendicular offset for bond distance perspective labels (in inches) specifies the offset of the center of the distance label relative to the center of the bond.

- The height of regular labels for bond distances is entered as zero if no bond distances corresponding to foreshortened bonds are to be drawn. Positive values give the lettering height in inches before projection.

- The perpendicular offset for bond-distance regular labels (in inches) specifies the offset of the center of the distance label relative to the center of the bond.

- The significant digits indicator is -1, 0, or 1, denoting bond distance labels with one, two, or three digits, respectively, after the decimal place.

Instructions 801, 802, and 803 differ from 811, 812, and 813 only in the ORTEP output file listing. The second group has no output except error messages. The first group lists: plotter coordinates in inches, scaled Cartesian coordinates (in inches) of atoms before projection, and triclinic crystal coordinates for the atoms of each bond. The interatomic bond distance in Angstroms is also listed. If an atom of a bonded pair is out of bounds, a fault message (NG = 10) is printed in the ORTEP output file. If the bond is hidden and cannot be drawn, fault message NG = 14 is printed. Fault NG = 13 signifies that an imaginary intersection was found with a bond radius larger than the ellipsoid semidimension. (See 3.4.)

3.3.9.1 Explicit Bonds

Explicit stick bonds are produced with the 801/811 instructions. The bonds are described with two atom designator codes for each bond. The atom designator codes go on the 801/811 card and on Format No. 1 trailer cards if more than three atom pairs are needed. The two atom designator codes for a bond must be in adjacent fields, but blank fields can be inserted between the different bonds. Since there are seven fields available per card, it is a good idea to use only two, four, or six of them so that the card sequence within the instruction (other than first and last cards) will be unimportant. As mentioned earlier, a Format No. 2 trailer card is required.

Columns	Instructions 801/811	Columns	Format No. 2 Continuation Card
3	2 (or 1 if more than 3 pairs of ADCs are needed)	3	—
7-9	801 or 811	22-24	Bond type
10-18	ADC_1 (from)	37-42	Bond radius (Å)
19-27	ADC_1 (to)	43-48	Perspective label height (in.)
28-36	[ADC_2 (from)	49-54	Perpendicular displacement (in.)
37-45	ADC_2 (to)]	55-60	Nonperspective label height (in.)
46-54	[...	61-66	Perpendicular displacement (in.)
55-63	...]	67-72	Digits indicator

3.3.9.2 Implicit Bonds

Instructions 802/812 are used for implicit stick bonds, and 803/813 are used for implicit line bonds. The symbol drawn on the atomic sites by the 803/813 instructions may be made larger or smaller by redefining the SCAL2 factor, which is controlled by the 600 series of instructions (see 3.3.7). Number run type takes a non-zero (or non-blank) value only when working with atom "features" (see 3.1.10-12 and 4.6). At least one Format No. 2 trailer card is required.

The use of vector search code (VSC) cards for the implicit bond plotting instructions has been extended to include a provision for drawing coordination polyhedra while suppressing the unwanted bonds. In addition to describing the bond with origin and target number runs and the D_{min} to D_{max} range, a condition can be imposed to require that both atoms must be within a specified "polyhedral distance range" of an atom in the ATOMS array that is included in a "polyhedron" number run. This option is brought into play by a negative value in columns 43 to 48 of the VSC card.

Columns	Instructions 802/812 and 803/813
3	2
7-9	802 or 812
10-18	—
27	NR type

Columns	Instructions 802/812 Format No. 2 Vector Search Code: Positive value in col. 43-48	Instructions 802/812 Format No. 2 Vector Search Code: Negative value in col. 43-48	Instructions 803/813 Format No. 2 Vector Search Code: Positive value in col. 43-48	Instructions 803/813 Format No. 2 Vector Search Code: Negative value in col. 43-48
3	0 or 2	0 or 2	0 or 2	0 or 2
10-12	Org. NR (from)	Org. NR (from)	Org. NR (from)	Org. NR (from)
13-15	Org. NR (to)	Org. NR (to)	Org. NR (to)	Org. NR (to)
16-18	Tar. NR (from)	Tar. NR (from)	Tar. NR (from)	Tar. NR (from)
19-21	Tar. NR (to)	Tar. NR (to)	Tar. NR (to)	Tar. NR (to)
22-24	Bond type	Bond type	—	—
25-30	D_{min} (Å)	D_{min} (Å)	D_{min} (Å)	D_{min} (Å)
31-36	D_{max} (Å)	D_{max} (Å)	D_{max} (Å)	D_{max} (Å)
37-42	Bond radius (Å)	Bond radius (Å)	—	—
43-48	Perspective label hgt. (in.)	Polyhedron NR (from)	—	Polyhedron NR (from)
49-54	Perpendicular displacement (in.)	Polyhedron NR (to)	—	Polyhedron NR (to)
55-60	Nonperspective label hgt. (in.)	Polyhedron D_{min} (Å)	—	Polyhedron D_{min} (Å)
61-66	Perpendicular displacement (in.)	Polyhedron D_{max} (Å)	—	Polyhedron D_{max} (Å)
67-72	Digits indicator	—		

3.3.9.3 Bond Overlap

Instructions 821 and 822 are used for calculating bond/atom and bond/bond overlap rather than for drawing bonds. These are described in Section 3.3.12.

3.3.10 Label Plotting Instructions (900 Series)

The 900 series allows the user to plot general titles up to 72 characters in length, chemical symbols up to 6 characters long, and bond length labels. The bond length labels can have two decimal places before the decimal point and one, two, or three places after the decimal point. The 700 and 800 series instructions can plot chemical symbols and bond length labels, respectively, but it is often desirable to position certain labels individually with the 900 series.

ORTEP-III provides the following character set for labels:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789-!@#$%&* () +, - '" .
/:;<=>? space
```

General titles and bond length labels can be drawn either in perspective or parallel to the plane of the drawing (nonperspective). Chemical symbols are always drawn parallel to the plane. Instructions 913–916 produce perspective lettering, and instructions 901–906 produce regular plane lettering.

Two vectors, the upright vector and the base-line vector, are needed to describe a lettering plane. In ORTEP, the upright lettering vector is always parallel to the plane of the drawing. For perspective lettering the base-line vector is a general vector in three dimensions. In the nonperspective case, the base-line vector is either along the projection of a general vector or along the vector (in the plane of the drawing) that is oriented with a 302 title rotation instruction (theta base line). If theta is zero, then the theta base-line vector is along the plotter positive x axis.

The exact center of the label is always referred to when specifying the position of the label. The program goes through the following steps to position the center point of the label onto the drawing.

(1) A point $P1$ is found that is either the position of atom A (columns 10-18) or the mean of two atom positions (atom A and atom B) if an atom designator exists in the atom B field (columns 19-27) of the instruction card.

(2) A point $P2$ is found by (a) translating from $P1$ along the base-line vector for the distance specified by the parallel offset and then (b) translating along the upright vector by the perpendicular offset distance.

(3) A point $P3$ is found by projecting $P2$ onto the plane of the drawing.

(4) If the *x edge reset* is greater than 0, then x is reset to this value. If *x edge reset* is less than 0, x is reset to the positive x plot boundary minus |*x edge reset*|. No resetting is done if *x edge reset* equals 0. The y parameter is handled in the same manner with *y edge reset*.

Instruction 901 produces a nonperspective atom label with theta base line, using the atom label for atom A.

A nonperspective title with theta base line is drawn with the 902 instruction. The title must be provided with the instruction on a Format No. 3 trailer card. The title should be centered about columns 36-37 of that card. However, if the title begins in column 1, it will be centered automatically (a new feature in ORTEP-III).

A nonperspective general title with normal plane lettering is produced by the 903 instruction and with perspective lettering by 913. The general vector is from atom A to atom B. The title must be provided with the instruction on a Format No. 3 trailer card. The title should be centered about columns 36-37 of that card. If the title begins in column 1, it will be centered automatically.

Instructions 904, 905, 906, 914, 915, and 916 produce bond-length labels. The first three are for nonperspective lettering with one, two, and three places after the decimal point, respectively; and the last three are for the corresponding bond-length labels with perspective. The general vector is from atom A to atom B. Note that the sense of the vector is important in order to have the label right side up.

The format for the entire 900 series is as follows:

Columns	**900 Instructions**
3	0 (or 3 for 902, 903, and 913)
7-9	9*nn* (instruction number)
10-18	ADC for atom A
19-27	[ADC for atom B]
28-36	*x* edge reset (in.)
37-45	*y* edge reset (in.)
46-54	Lettering height (in.)
55-63	Parallel offset (in.)
64-72	Perpendicular offset (in.)

3.3.11 Save Sequence Instructions (1100 Series)

It is often desirable to repeat a sequence of instructions one or more times with other instructions inserted between the repetitions. The 1100 series allows the user to do this without the necessity of putting in duplicate sequences of instruction cards. It is not an elaborate looping device, but it does give additional flexibility to the system.

The three instructions in this series are 1101 to start the save sequence, 1102 to terminate the save sequence, and 1103 to execute the save sequence. All instruction cards and their trailer cards between the 1101 and 1102 instructions are *executed and saved*. Each subsequent 1103 instruction then repeats all the saved instructions. There are no parameters to be entered with the 1100 series instructions.

	Instructions
Columns	**1101, 1102, and 1103**
6-9	1101, 1102, or 1103

3.3.12 Overlap Correction Instructions (1001, 821, 822)

(Instruction 511 in OR TEP-II[2] has been renumbered to 1001 in ORTEP-III. Instruction 511 continues to work as it did in OR TEP-II,[2] but users should use 1001 in the future.)

Instruction 1001 activates the hidden line removal feature of ORTEP, which corrects the illustration for overlapping atoms and bonds. It stores the projected atom boundary ellipses for all atoms in the ATOMS array. It is important that the contents of the ATOMS array, the scaling and positioning, and the structure orienting parameters (controlled by the 400, 600, and 500 series of instructions, respectively) not be changed between the 1001 instruction and the drawing of the atoms and bonds by the 700 and 800 series of instructions. Otherwise, the projected outlines may be destroyed or become inappropriate.

The projected outlines of bonds are approximated as quadrangles. Those of implicitly specified bonds may be stored for overlap correction by adding Format No. 2 trailer cards to the 1001 instruction that are identical to those of the 802/812 instructions used subsequently for plotting the implicit bonds. Alternatively, implicitly specified bonds may be stored with one or more 822 instructions with the Format No. 2 trailer card set. A 1001 instruction must precede the first 822. If more than one 802/812 instruction is used to draw the bonds, there probably should be a corresponding 822 provided for each.

Explicit bonds may only be stored by using one or more 821 instructions following a 1001 instruction. These will be identical to the 801/811 instructions used subsequently for plotting the explicit bonds except for the instruction number.

As currently dimensioned, the maximum numbers of projected atoms and projected bonds that can be stored are 500 and 599, respectively. A list of the projected atoms and bonds is given in the ORTEP output file.

The projected outline information for atoms and bonds must be recalculated for each member of a stereo pair; consequently the 1001 and 821 or 822 instructions are usually the first instructions within the save sequence. The old overlap information is deleted whenever a new 500 or 600 series or 1001 instruction is executed.

All details inside atoms and bonds, including chemical symbols drawn with 700 series instructions and bond-distance labels drawn with 800 series instructions, will be corrected for overlap. Lettering drawn by the 700 and 800 series instructions that is outside the atom or bond boundaries will not be corrected for overlap unless the overlying atom or bond outline actually intersects (in projection) the atom or bond being drawn. Labels and symbols drawn with the 900 series instructions will not be corrected for overlap.

3.3.12.1 Instruction 1001, Projected Outline Storage

A constant width overlap margin (i.e., a blank strip at the intersection of overlapping elements) is included in the dimensions of each projected atom ellipse and projected bond quadrangle. The width of this margin (in inches) may be specified as a parameter with the 1001 instruction if desired; otherwise, the margin is set by default. The default value is calculated as follows:

$$\text{for SCAL1} < 0.25: \quad \text{OVMRGN} = \max(0.010,\ \text{SCAL1}^{1/2} \times 0.05)$$

$$\text{for SCAL1} \geq 0.25: \quad \text{OVMRGN} = \max(0.025,\ \text{SCAL1}^{1/2} \times 0.03)$$

Some users prefer an overlap margin of 0.0 for stereoscopic drawings.

Columns	**Instruction 1001**
3	2 (if bonds are to be stored; otherwise 0)
6-9	1001
10-18	Overlap Margin 0: default value (described above) 1: 0.0 in. Value in range 0-1.0 in.
27	NR type

If the value in column 3 is "2", Format No. 2 trailer cards are included. In general, all the trailer cards included with the 802/812 instructions are used. (See 3.3.9 and 3.3.12.3.)

New on this card in ORTEP-III is the second parameter, which specifies the number run type of the number runs that follow on the Format No. 2 trailer cards.

3.3.12.2 Instruction 821, Explicit Bond Outline Storage

If explicit bonds are to be stored for the overlap correction, the attached atoms must be in the ATOMS array even though this is not a requirement for the 801/811 instructions used for plotting the explicit bonds. In general, the 821 instructions will be identical to the 801/811 instructions (except for the instruction number) used subsequently for plotting the explicit bonds. (See 3.3.9.)

3.3.12.3 Instruction 822, Implicit Bond Outline Storage

Usually, the information on implicit bond outlines is stored with the 1001 card. However, if more than 20 Format No. 2 trailer cards are needed to specify all bonds, the extra ones can be entered with this instruction. Also, 822 instructions in addition to the 1001 may be used to treat different atom features in different ways. In general, the 822 instructions will be identical to the 802/812 instructions (except for the instruction number) used subsequently for plotting the implicit bonds. (See 3.3.9.)

Columns	**Instruction 822**
3	2
7-9	822
10-18	—
27	NR type

3.3.13 Termination Instructions (Negative Series)

The -1 instruction terminates ORTEP, exiting the program via subroutine EXITNG.

The -2 instruction terminates the current structure and reinitializes ORTEP to read another structure from the same input file starting with a new title card. As many structures as desired may be cascaded in the input file in this manner before exiting with a -1 instruction. The user's initial input about the destinations for the drawing and text output hold for the entire set of structures. At the end of each structure, the user is offered the opportunity to save the drawing, view it on the screen, or edit the current instruction set before proceeding to the next structure. (See 4.1.).

	Instructions
Columns	**-1 and -2**
8-9	-1 or -2

3.3.14 Supplementary Instructions

The instructions available in ORTEP-III have numbers less than 1200. Any instruction in the input file with a number greater than or equal to 1200 will cause ORTEP to branch to subroutine SPARE to execute the code found there. As distributed, this is a dummy subroutine containing the single FORTRAN statement RETURN. It is provided to give users a simple way to add their own instructions without having to understand the full program logic. Subroutine SPARE has a single parameter INST, which is the instruction number.

3.4 LIST OF FAULT INDICATORS

ORTEP checks for certain errors, and when one of these occurs, an error message, "FAULT NG = *NG ADC m*" is written in the ORTEP output file. The meanings of fault indicator number *NG* is explained in Table 3.1. *ADC* and *m* identify the atom code and the instruction involved (if these are relevant). If possible, corrective measures are made by ORTEP and the calculation proceeds; otherwise, the job is terminated by calling subroutine EXITNG.

Table 3.1. ORTEP fault indicators.

NG	Subroutine Involved	Fault	Action
1	PRELIM	No sentinel found after reading 96 symmetry cards	Tries to read parameter cards
2	PRELIM	No sentinel found after reading the parameter cards for 505 atoms	Tries to read instruction cards
3	PRELIM	Anisotropic temperature factor coefficients form a matrix which is not positive definite	EXIT after printing out all rms principal values (imaginary ones are listed negative)
4	ATOM, PAXES	Symmetry operation number is higher than the number of input operations	Omit atom
5	ATOM, PAXES	Atom number is higher than the number of input atoms	Omit atom
6	EIGEN	Null temperature factor matrix or failure in bisection routine	EXIT, after printing out all principal values
7	EIGEN	Eigenvector routine failure due to null vector	EXIT, after printing out all principal values
8	INITSC	Error initializing screen driver	EXIT
9	MAIN, SPARE	Unidentified instruction number	Omit faulty instruction
10	BOND, F700	Atom out of bounds	Omit atom
11	F800	No vector search codes	Omit instruction
12	F600, SEARC	Insufficient number of atoms in ATOMS array	EXIT
13	BOND	Imaginary bond intersection (i.e., bond larger than atom)	Omit bond
14	BOND	Hidden (end-on) bond	Omit bond
15	F900	Null vector as base line	Omit label
16	STORE	ATOMS array is full	Omit all succeeding atoms
17	LAP700, LAPAB	Maximum number of projected ellipses (20) over an atom or bond to be drawn exceeded	Additional ellipses not corrected for overlap
18	LAP800, LAPAB	Maximum number of quadrangles (30) over an atom or bond to be drawn exceeded	Additional quadrangles (bonds) not corrected for overlap

4. USING ORTEP-III

This section provides some general information on using ORTEP-III, and it describes how some aspects of the program work to help users who wish to modify the program's operation.

4.1 USER INTERFACE

ORTEP-III is a semi-interactive program that requests certain control information from the user while the program executes. A default value for each item is provided in square brackets, and it will be used if the user simply hits the RETURN key (on some keyboards, the name of this key is ENTER). When ORTEP starts, the user is asked to supply three basic items of information.

```
Enter instruction set file name or "exit" [TEP.IN]:
Drawing to (1) Screen, (2) Postscript file, (3) HPGL file, or (0) Omit [1]:
Text output to (1) File, (2) Screen, or (0) Omit [0]:
```

The first of these is the ORTEP input file name with a default name of TEP.IN. The user may supply a file name or may enter "exit" or "EXIT" to stop program execution. Care must be taken when entering a file name to match the case of the letters on those systems that distinguish upper and lower case.

The second item requested is the destination of the ORTEP drawing with the default being the screen. The alternatives of an Encapsulated Postscript (EPS) file or HPGL/2 (Hewlett-Packard Graphics Language) file are chosen if the user enters a "2" or "3", respectively. A choice of "2" or "3" causes ORTEP to ask about the orientation of the drawing.

```
(1) Portrait or (2) Landscape orientation [1]:
```

One further piece of information is needed for an EPS landscape drawing.

```
How tall is printer page in inches? [11.00]:
```

If the ORTEP illustration is saved in a file, the file will be named TEP*nnn*.PRN where *nnn* is a number starting with the value 001. The program sets the value so *old illustration files are not overwritten.* Once the file name is determined by the program, the name is displayed on the screen. If a different naming convention is preferred, the code for naming the files is in subroutines INITPS and INITHP.

The third main user inquiry is the destination of ORTEP's text output. The default value here is to omit the output. An entry of "2" displays the output on the screen. If a "1" is entered to save the output in a file, the user is asked to supply a file name.

```
Enter output file name [TEP.out]:
```

The default name is derived from the name of the input file by adding or substituting the extension ".out". A different output file name may be typed in. CAUTION: If the name chosen here is the same as an existing file, it will be overwritten.

Before the program exits, the user is given another opportunity to save the drawing or view it on the screen. In addition, an option to use ORTEP's internal editor is provided.

```
(1) Save drawing as Postscript file
(2) Save drawing as HPGL file
(3) Redraw structure on screen
(4) Edit instruction set
[Quit]:
```

The wording of the user requests and the code to handle the user input are in subroutine UINPUT. Default values are set in subroutine DFLTS and held in COMMON DFL.

4.2 SCREEN DISPLAY OF THE ORTEP ILLUSTRATION

The screen drawing subroutines available in ORTEP-III use PGPLOT. PGPLOT is a free graphics library developed by T. J. Pearson at the California Institute of Technology. The package is written in FORTRAN and operates on a variety of platforms. Information about PGPLOT can be found on the World Wide Web at http://astro.caltech.edu/~tjp/pgplot or via e-mail to tjp@astro.caltech.edu.

PGPLOT creates a window on the screen that represents an 8 1/2 × 11 inch drawing surface in landscape orientation on which ORTEP draws its illustration. After viewing the illustration, the user must click in the EXIT box with the mouse and hit RETURN to go back to the user input dialog. If the mouse does not operate, two RETURNs should work.

Users can interact with the illustration to identify atoms displayed on the screen. This is done by positioning the cursor (with a mouse or with the keyboard's arrow keys) on a displayed atom and clicking the mouse button or typing the letter "a". The atom's label (if it has one) and atom designator code are displayed. Whether a mouse will work depends on how PGPLOT operates on the particular platform. The cursor must be positioned within 1/16 inch of the atom center. If ORTEP identifies two or more atom centers within 1/16 inch of the cursor, it will provide the identification of the atom center closest to the cursor position along with question marks. Exercise caution when atoms lie directly over or very near each other.

4.3 PLOTTING THE ORTEP ILLUSTRATIONS

The destination of the ORTEP illustration is controlled by the value stored in the variable NDRAW, which is set from user input in subroutine UINPUT and subsequently held in COMMON NS. The pre-defined values for NDRAW built into ORTEP-III are listed below. Exercise caution if the illustration destination code is altered. If a new destination is needed, it is probably a good idea to create a new value for NDRAW that is different from those already programmed.

NDRAW = 0: no illustration
NDRAW = 1: screen output
NDRAW = 2: EPS file output
NDRAW = 3: HPGL file output
NDRAW = 9: RESERVED for future use

Seven subroutines control the plotting of illustrations produced by ORTEP-III—F200, PLOT, INIT*xx*, COLR*xx*, PEN*xx*, PENW*xx*, and END*xx* [*xx* refers to a specific plotting device (see below)]. These may be modified to produce the output required to plot the illustrations on a variety of devices.

Subroutine F200 responds to 200 series instructions in the ORTEP input file. A 201 instruction produces a call to INIT*xx*, which contains the specific initialization code for device *xx*. Similarly, a 202 instruction with no parameters (i.e., blanks or "0"s) produces a call to END*xx*, which contains the device's termination code. When a 202 instruction with one or two non-zero parameters is received by F200, it interprets the parameters as the *x*- and *y*-shift, respectively, in the plotting origin. These values are stored in the variables XTRANS and YTRANS, respectively, that are held in COMMON TRFAC. (These variables are both initially set to "0" when the 201 is received.) The 204 instruction produces a call to COLR*xx* to change the color of the plot. PENW*xx* is called to change the thickness of the plotted lines in response to a 205 instruction. For compatibility with OR TEP-II,[2] the 203 instruction is treated as a 201.

The illustration is produced when ORTEP calls subroutine PLOT with the arguments X, Y, and IPEN. X and Y are the *x,y* position (in inches) where the plotting device's pen should move, and IPEN is a flag that specifies whether the pen should be down (producing a line, IPEN=2) or up (not producing a line, IPEN=3) as it moves to its new position. These values are sent to PEN*xx*, which contains the specific code for device *xx* to handle the pen movements. The values of XTRANS and YTRANS are added to X and Y, respectively, in PEN*xx*.

Two functional plotting "devices" are included in the ORTEP-III distribution. These do not actually control physical devices but instead create files containing EPS and HPGL/2 descriptions of the illustrations. These files may be downloaded to printers/plotters that accept those particular page description languages. ORTEP-III itself does *not* automatically download (or spool) the files for printing/plotting. Some computer graphics programs and word processors will import these files. The EPS specific codes are in subroutines INITPS, COLRPS, PENPS, PENWPS, and ENDPS. The HPGL specific codes are in subroutines INITHP, COLRHP, PENHP, PENWHP, and ENDHP.

Since every user will not have PGPLOT, the five subroutines that control screen drawing—INITSC, COLRSC, PENSC, PENWSC, and ENDSC—are not functional in the ORTEP-III source code distribution, i.e., they are "dummy" subroutines containing only the single FORTRAN instruction RETURN. However, on systems that have PGPLOT implemented, alternate code for these five subroutines is available in the distribution and should be substituted for the dummy versions. INITSC creates the window in which the ORTEP illustration is drawn. The size and orientation of the window can be changed by modifying the code in INITSC.

The PGPLOT screen driver code includes a subroutine named CURSSC that has no counterpart in the EPS and HPGL drivers. This code allows users to identify atoms displayed on the screen (see 4.2).

4.4 INTERACTIVE EDITOR

ORTEP-III provides a simple line editor for editing the input file without exiting the program. NOTE: Comments in the input file beginning with # will not be displayed and will not be retained if the instruction set is saved. The editor cannot be used to create comments beginning with #. (See 1.2.6.)

When the editor is invoked, the instruction set from the input file is displayed on the screen with line numbers along with the editor commands.

```
C=Change line #                          D=Delete line(s) # [#]
I=Insert line before #                   T=Type line(s) [#] [#]
S=Save modified instruction set          O=Restore original instruction set
P=Save drawing as Postscript             H=Save drawing as HPGL
R=Redraw structure on screen             Q=Quit
```

(The symbol # in the editor commands stands for a line number. Brackets [ ] mean optional.)

All commands are entered as a single letter (upper or lower case) followed by 0, 1, or 2 integers, representing line numbers, separated by spaces.

The "Q" command exits the editor and terminates ORTEP.

The "C" command requires the number of the line to be changed. The line is displayed, and changes are typed on the line directly under it. Use the space bar and backspace key to position the cursor where changes are to be made and type the changes. Only non-space characters will be substituted into the original line. To substitute a space into the original line where a non-space character exists, type the @ character at that position. The substitutions are made when the RETURN key is pressed.

Similarly, the "I" command requires one line number. A blank line is provided and the user types in its contents, ending with a RETURN.

The "D" command requires the number of at least one line to delete. If two are entered, a block of lines is deleted.

With no line numbers, the "T" command types the entire instruction set on the screen. It types a single line or a block of lines if one or two line numbers, respectively, are entered.

After editing the instruction set, it may be saved in a new file with the "S" command. The user is asked to provide the name of a file that does not already exist.

If it is determined that editing has taken things too far afield, the original input instruction set may be restored with the "O" command.

The "R", "P", and "H" commands draw the ORTEP illustration on the screen, in an EPS file, and in an HPGL file, respectively. As a safety precaution, these three commands save a copy of the current instruction set in a file named TEP.NEW before their respective actions are executed.

The code that controls the editor is in subroutine EDITR.

4.5 ALTERNATE FORMATS FOR ATOMIC PARAMETERS

ORTEP-III provides a way to input atom parameters that are available in a "nonstandard" format. If the sentinel value (column 1) on the last symmetry card in the ORTEP input file is "2" instead of "1", the program asks the user for the name of a file containing the atom parameters

and branches to subroutine READIN to read the information. This subroutine may be modified and the program recompiled to read any desired format.

Subroutine READIN has 16 parameters as described below. Upon entry, only IU, the input file unit number, has been set by the calling routine. All other parameters must be set in READIN before control is returned to the calling routine. The READIN subroutine distributed with ORTEP-III may be used to read small protein fragments (500 atoms or less) in the Brookhaven Protein Data Bank format. (See 7.4.)

```
subroutine readin(iu,chem,id1,id2,x1,x2,x3,it,is,b1,b2,b3,b4,b5,b6,btype)
```

chem	6-character atom label
id1,id2	feature #1, #2
x1,x2,x3	atom x,y,z position
it	positional parameter type
is	end-of-atoms flag ("1" for last atom, "0" for rest)
b1,b2,b3,b4,b5,b6	atom temperature factors
btype	temperature factor type

4.6 ATOM "FEATURES"

In earlier versions of ORTEP, atoms could be referenced only by their numeric positions in the input file. Thus, atom number runs (ANR) (see 3.1.4) were used to select groups of atoms to be treated in the same manner. ORTEP-III allows two optional attributes called "features" (see 3.1.10) to be provided with each atom, and feature number runs (FNR) (see 3.1.11) can be used to select groups of atoms having particular features. Features should prove especially useful for polymeric materials such as proteins or nucleic acids as will be seen in the examples below.

The two atom features are stored in the INTEGER*2 variables IDENT(1,*n*) and IDENT(2,*n*) where *n* is the atom number in the input list. The first of these contains Feature #1 of the atom and the second contains Feature #2. These can be assigned values as appropriate to the required task.

For typical ORTEP input, features can be entered in columns 10-18 and 19-27 on Type 0, 1, and 2 atom positional parameter cards (see 3.2.4.1). The values (or zeroes) in these fields are read in and stored in temporary real variables. They are also truncated to integers and stored in IDENT(1,*n*) and IDENT(2,*n*). If these fields happen to contain numeric values unrelated to feature definition, they will be ignored and will not interfere in any way with the operation of the program. However, if commands are invoked that specifically look at feature values, ORTEP will assume these numbers represent features of the atoms.

To illustrate this new concept, suppose a crystal contains four elements but the user wishes to plot only the atoms of one element, say sulfur. If all the sulfur atoms are grouped together in the input file, a single ANR would select them. If the sulfur atoms do not occur together in the input file, several ANRs would be needed to select all of them. In either case, the user would have to count the atoms in the input file to determine the starting and ending position numbers of the sulfur atoms. On the other hand, if the atomic number of each atom is recorded as Feature #1 in the input list, a single FNR to find all atoms with the value 16 for Feature #1 would select all sulfur atoms regardless of their positions in the input list.

The most common use for features will probably be with macromolecules. For example, in a protein, a feature field can be a convenient place to store a residue number or a code representing a structural feature of an amino acid. Atom parameters for these large molecules are likely to be read into ORTEP with the subroutine READIN (see 4.5) In this case, code can be written in READIN that sets the values of IDENT(1,*n*) and IDENT(2,*n*) directly. Another place where features can be useful is in critical net illustrations (see 7.5) where it is advantageous to distinguish the peak, pass, pale, and pit critical points.

To handle features, a new parameter, number run type (see 3.1.12), has been added to the 100 series, 400 series, 505, 506, 700 series, 800 series, and 1001 instructions. See the instruction descriptions in Section 3 for the details.

Example: A protein contains 60 amino acids with a total of 500 atoms. Feature #1 of each atom has the sequence number of the amino acid containing the atom. Feature #2 has codes for the structural characteristics of the atom: 1 for an α-carbon, 2 for a C=O carbon in the peptide link, 3 for a nitrogen in the peptide link, 4 for all other atoms.

• To select all the atoms in amino acids 9-17:

```
2   402   155501      500        1      500      2.0
          9 17  9 17  1
```

• To select all the atoms in the protein backbone:

```
2   402   155501      500        1      500      2.0
          1  3  1  3  2
```

• To select all the backbone atoms in amino acids 9-17:

```
1   402   155501      500        1      500      2.0
2                       1
2         9 17  9 17  1
          1  3  1  3  2
```

• To draw all atoms in the protein:

```
1   712
                                 1      500
```

• To draw the atoms in the protein backbone:

```
1   712
                                 1       3       2
```

• To draw the atoms in amino acids 9-17:

```
1   712
                                 9       17      1
```

• To draw the bonds between the atoms in amino acids 9-17:

```
2   812                    1
          9 17   9 17   1    0.9    2.0    .04
```

• To correct for overlap among the atoms and bonds in amino acids 9-17:

```
2  1001                    1
          9 17   9 17   1    0.9    2.0    .04
```

4.7 MODIFYING ORTEP-III

Appendix A is a listing and brief description of the subprograms that make up ORTEP-III, and Appendix B has the same information for the major variables that are used. The entire FORTRAN program listing is in Appendix C. This information is provided to help users who wish to make changes to the program.

5. TECHNICAL DETAILS

Publications by Heading,[8] Springer,[9] Todd,[10] and Korn and Korn[11] provide background on many of the projective and analytical geometry concepts discussed in this section.

5.1 HOW ORTEP DRAWS ELLIPSOIDS

Fig. 3.1 illustrates various ellipsoid graphical representations that can be drawn with ORTEP. The major components in the representations are the three principal ellipses and the boundary (outline) ellipse. The principal ellipses have a front half and a back (hidden) half. The entire boundary ellipse is visible.

An ellipse is approximated by connecting a series of points on the ellipse with straight line segments. Points on an ellipse having a general orientation in three dimensions are computed; then each of these points is projected onto the drawing board for plotting.

The basic algorithm for finding the points along a given general ellipse utilizes the properties of conjugate diameters. Assume that we have the three principal axis vectors $\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{V}_3$ of the general ellipsoid and a vector $\mathbf{V}_4$ from the center of the ellipsoid to the viewpoint. The vector $\mathbf{V}_5$ normal to the polar plane (see Fig. 5.1), whose pole is the viewpoint, can be obtained from

$$\mathbf{V}_5 = \mathbf{A}\,\mathbf{V}_4\,, \tag{5.1.1}$$

where $\mathbf{A}$ is the matrix for the ellipsoid that is defined by

$$\mathbf{X}^{\mathrm{T}}\,\mathbf{A}\,\mathbf{X} = d\,, \tag{5.1.2}$$

where d is a constant.

The boundary ellipse is defined by two conjugate vectors, one of which is any vector $\mathbf{V}_6$ perpendicular to $\mathbf{V}_5$ and the second is $\mathbf{V}_7$, where

$$\mathbf{V}_7 = \mathbf{V}_5 \times \mathbf{A}\,\mathbf{V}_6\,. \tag{5.1.3}$$

The assumption made for this boundary ellipse derivation is that the view distance is large compared to the ellipsoid size. Therefore, the boundary ellipse defined above always lies on the diametral polar plane (see Fig. 5.1).

A principal ellipse that lies in the plane of the principal axis vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ will have the third principal axis vector $\mathbf{V}_3$ normal to the plane of the ellipse. The intersection of this principal ellipse with the boundary ellipse is along the vector $\mathbf{V}_8$ where

$$\mathbf{V}_8 = \mathbf{V}_5 \times \mathbf{V}_3\,. \tag{5.1.4}$$

This vector divides the front and back (hidden) sides of the principal ellipse. A vector conjugate to $\mathbf{V}_8$ and in the principal plane containing $\mathbf{V}_1$ and $\mathbf{V}_2$ is $\mathbf{V}_9$, where

$$\mathbf{V}_9 = \mathbf{V}_3 \times \mathbf{A}\,\mathbf{V}_8\,. \tag{5.1.5}$$

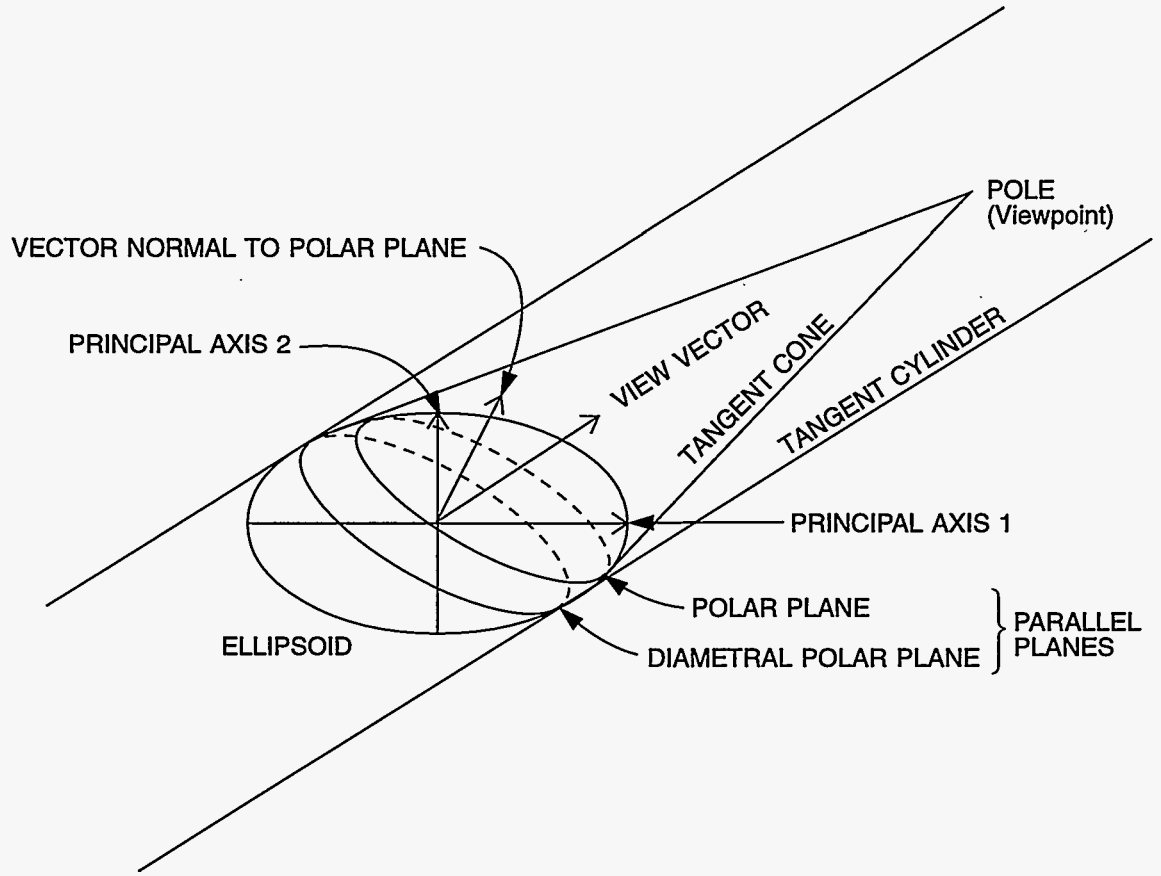

Fig. 5.1. Polar planes formed by tangent cylinder and tangent cone.

After the conjugate vectors have been found, their lengths are adjusted to make them satisfy Eq. 5.1.2 by letting $\mathbf{X} = s\,\mathbf{I}$, where $\mathbf{I}$ is a unit vector. Solving for s, we obtain

$$s = [d / (\mathbf{I}^{\mathrm{T}}\,\mathbf{A}\,\mathbf{I})]^{1/2}. \tag{5.1.6}$$

A conjugate vector pair is expanded into an ellipse by subroutine RADIAL. Since an ellipse is centrosymmetric, the two conjugate vectors and their negatives give four vectors whose endpoints lie on the ellipse. By performing a vector sum of two adjacent vectors and dividing the resultant vector components by $\sqrt{2}$, we can obtain an additional vector. After doing this for all adjacent pairs, we then have a total of eight vectors. This process can be repeated as many times as desired except that the scaling constant will be different for each cycle. The constant is described by

$$\mathrm{CONST}_i = \sqrt{2[1 + \cos(\pi/2^i)]} = 2\cos(\pi/2^{i+1}), \tag{5.1.7}$$

where i is the cycle number.

This total process may be thought of as taking a planar radial set of equally spaced unit vectors and performing a deformation and scaling on the space in which it is described. In geometry, this deformation is called an affine transformation.

Complete details on drawing ellipsoids can be obtained from the FORTRAN coding of subroutines F700 and RADIAL.

5.2 ELLIPSE RESOLUTION

With printer/plotter resolution improvements, highly complex stereo drawings with each image of the pair about 2.25 inches wide can now be produced directly without photographic reduction. As a result, it has become necessary to adjust the resolution (smoothness) of the plotted elliptical curves in ORTEP to take full advantage of the output devices' capabilities.

ORTEP produces its ellipses by "stretching" an equal-area circle to the shape of the ellipse. (Since ORTEP draws all its shapes with straight lines, the circle is, in fact, a polygon, and the smoothness of the ellipse depends on the number of vertices, or spokes, in the polygon.) ORTEP can produce circles having 16, 32, 64, and 128 spokes. The selection is a function of the radius of the circle. The default radii for changing to circles with fewer spokes are 0.09375 in. (128 → 64), 0.375 in. (64 → 32), and 0.75 in. (32 → 16). These default values are smaller than those in OR TEP-II.[2]

The smoothness of ellipses can be altered with the new 304 instruction in ORTEP-III, which allows the user to enter a smoothness factor. A factor of 1 sets the circle radii for changing to other spoke angles to the default values given above. A factor less than 1 produces smoother ellipses, and values greater than 1 produce ellipses with more "jaggies". If the factor is 0, all ellipses, regardless of their size, will be drawn from circles having 128 spokes.

Fig. 5.2 shows three groups of ellipses produced with smoothness factor settings of 3, 2, and 0. Although perfectly smooth ellipses are the ideal, it may be necessary to balance their appearance against the computational time to produce them and the sizes of the resultant files containing the ORTEP illustration. Table 5.1 shows the sizes of the EPS and HPGL files of the concentric ellipses in Fig. 5.2 as a function of the smoothness factor. Computational time was not determined for these examples, but it may become a significant factor on slower computers.

Table 5.1. Size of ORTEP illustration files as a function of ellipse smoothness factor settings.

Smoothness Factor	Size of EPS File	Size of HPGL File
0.	35544	28558
0.5	29400	23566
1.	21464	17118
2.	11992	9422
3.	8408	6510
5.	7640	5886
10.	5848	4430

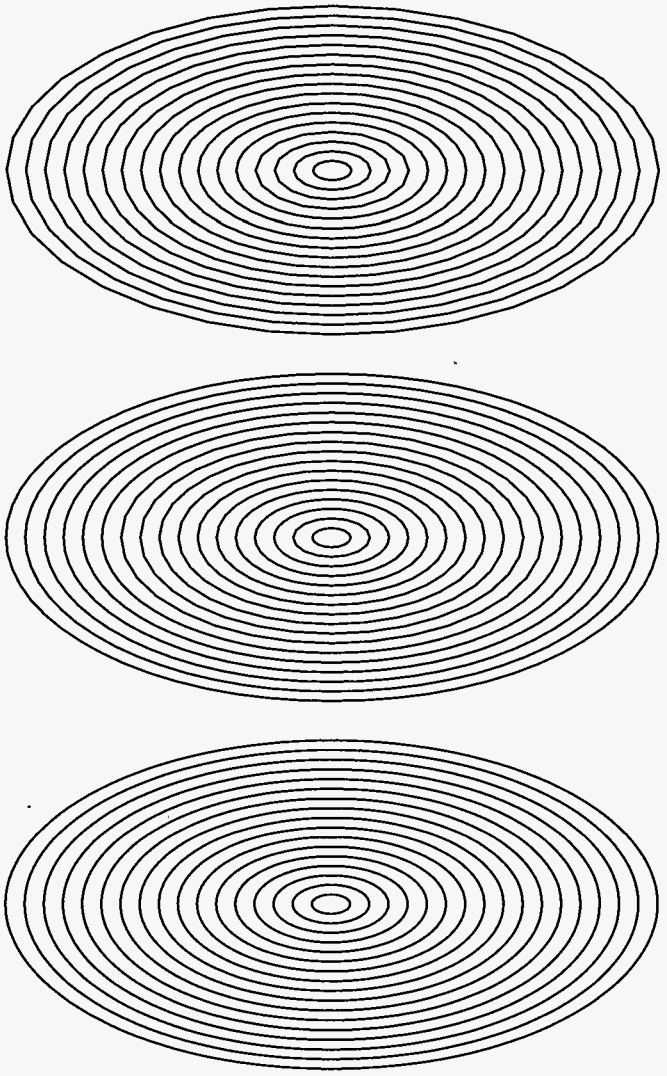

Fig. 5.2. Three sets of concentric ellipses produced by ORTEP with smoothness factor settings of 3, 2, and 0, respectively, starting from the top.

5.3 HOW ORTEP DRAWS BONDS

The major difficulty when drawing bonds is obtaining the intersection where the bond penetrates the ellipsoid. Three quadrics are used in subroutine BOND to calculate bond intersection. These three are the ellipsoid, the tangent cylinder, and the tangent cone.

The ellipsoid is described in matrix notation as

$$\mathbf{X}^{\mathrm{T}}\mathbf{A}\,\mathbf{X} = d, \tag{5.3.1}$$

where d is a constant and $\mathbf{X}$ is any vector from the center to the surface of the ellipsoid. The matrix $\mathbf{A}$ is 3×3 symmetrical with components a_{ij} $(i,j = 1, 2, 3)$.

The elliptic cylinder tangent to the ellipsoid and with its axis along z is described by

$$\mathbf{X}^T\mathbf{B}\,\mathbf{X} = d, \tag{5.3.2}$$

where

$$\mathbf{B} = \begin{pmatrix} a_{11} - \dfrac{a_{13}a_{31}}{a_{33}} & a_{12} - \dfrac{a_{23}a_{31}}{a_{33}} & 0 \\ a_{12} - \dfrac{a_{13}a_{32}}{a_{33}} & a_{22} - \dfrac{a_{23}a_{32}}{a_{33}} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{5.3.3}$$

and d is the constant used in Eq. 5.3.1. The tangent cylinder is used when it is necessary to terminate the bond at the boundary of the ellipsoid *when a parallel projection is used.*

To find the intersection of a cylindrical bond along $\mathbf{V_b}$ with radius r with either the ellipsoid or the tangent cylinder, we proceed as follows:

1. Form a radial set of vectors $\mathbf{V}r_j$ of length r normal to $\mathbf{V_b}$.

2. Take a unit vector $\mathbf{I}$ parallel to $\mathbf{V_b}$ and let

$$\mathbf{X}_j = \mathbf{V}r_j + s\,\mathbf{I}, \tag{5.3.4}$$

where s is a constant to be determined. Substituting in Eq. 5.3.1, we obtain

$$s^2\mathbf{I}^T\mathbf{A}\,\mathbf{I} + 2s\mathbf{V}r^T\mathbf{A}\,\mathbf{I} + \mathbf{V}r^T\mathbf{A}\,\mathbf{V}r - d = 0\,; \tag{5.3.5}$$

and solving for s, we get

$$s = \frac{-\mathbf{V}r^T\mathbf{A}\,\mathbf{I} + \sqrt{(\mathbf{V}r^T\mathbf{A}\,\mathbf{I})^2 - (\mathbf{I}^T\mathbf{A}\,\mathbf{I})(\mathbf{V}r^T\mathbf{A}\,\mathbf{V}r - d)}}{\mathbf{I}^T\mathbf{A}\,\mathbf{I}}\,. \tag{5.3.6}$$

The elliptic cone that is tangent to the ellipsoid and has its apex on the viewpoint can be obtained from the matrix $\mathbf{A}$ and the vector $\mathbf{V_u}$, which extends from the center of the ellipsoid to the viewpoint. This is performed in the following steps:

1. The ellipsoid is transformed with a rotation matrix to a new Cartesian frame of reference that has the z axis along the view vector $\mathbf{V_u}$.

2. The tangent cone can now be described as

$$\mathbf{Y}^T\mathbf{C}\,\mathbf{Y} = 0\,, \tag{5.3.7}$$

where $\mathbf{Y}$ is a vector originating from the vertex (viewpoint) of the cone and

$$\mathbf{C} = \begin{pmatrix} a_{11} + \dfrac{a_{13}a_{31}}{K-a_{33}} & a_{12} + \dfrac{a_{13}a_{32}}{K-a_{33}} & \dfrac{Ka_{13}}{K-a_{33}} \\ a_{21} + \dfrac{a_{23}a_{31}}{K-a_{33}} & a_{22} + \dfrac{a_{23}a_{32}}{K-a_{33}} & \dfrac{Ka_{23}}{K-a_{33}} \\ \dfrac{Ka_{31}}{K-a_{33}} & \dfrac{Ka_{32}}{K-a_{33}} & \dfrac{Ka_{33}}{K-a_{33}} \end{pmatrix}, K = d/(\mathbf{Vu}^T \mathbf{Vu}). \quad (5.3.8)$$

3. The frame of reference is rotated back to its original orientation with a rotation matrix that is the inverse of the one used in step 1. Note that the origin is now on the viewpoint rather than on the ellipsoid center.

To find the length, s, of a vector $s\mathbf{I}$ extending from any point p inside the cone to the surface of the cone, we let

$$\mathbf{Y} = \mathbf{Vp} + s\mathbf{I} \quad (5.3.9)$$

and obtain from Eq. 5.3.7

$$(\mathbf{Vp} + s\mathbf{I})^T \mathbf{C} (\mathbf{Vp} + s\mathbf{I}) = 0; \quad (5.3.10)$$

then solving for s, we obtain

$$s = \frac{-\mathbf{Vp}^T\mathbf{CI} + \sqrt{(\mathbf{Vp}^T\mathbf{CI})^2 - (\mathbf{I}^T\mathbf{CI})(\mathbf{Vp}^T\mathbf{CVp})}}{\mathbf{I}^T\mathbf{CI}} \quad (5.3.11)$$

The vector $\mathbf{Vp}$ from the vertex to p is formed by

$$\mathbf{Vp} = -\mathbf{Vu} + \mathbf{Vr}, \quad (5.3.12)$$

where $\mathbf{Vr}$ is any member of a radial set such as that described for the regular ellipsoid intersection.

5.4 OPTIMAL PARAMETERS FOR STEREOSCOPIC DRAWINGS

For optimal viewing of stereoscopic drawings, the origins of the two views should be separated by 2.2-2.4 in. and the stereo rotation between the two should be 5°-6°.

Fig. 5.3 was created when it was common practice to produce "large" ORTEP drawings that would be photographically reduced to give the optimal origin separation.[1,2] The reduced drawings would then typically be viewed with a stereoscope. Under these circumstances, Fig. 5.3 provides a picture of the relationship among the various parameters that must be taken into consideration when producing the drawing.[12,13]

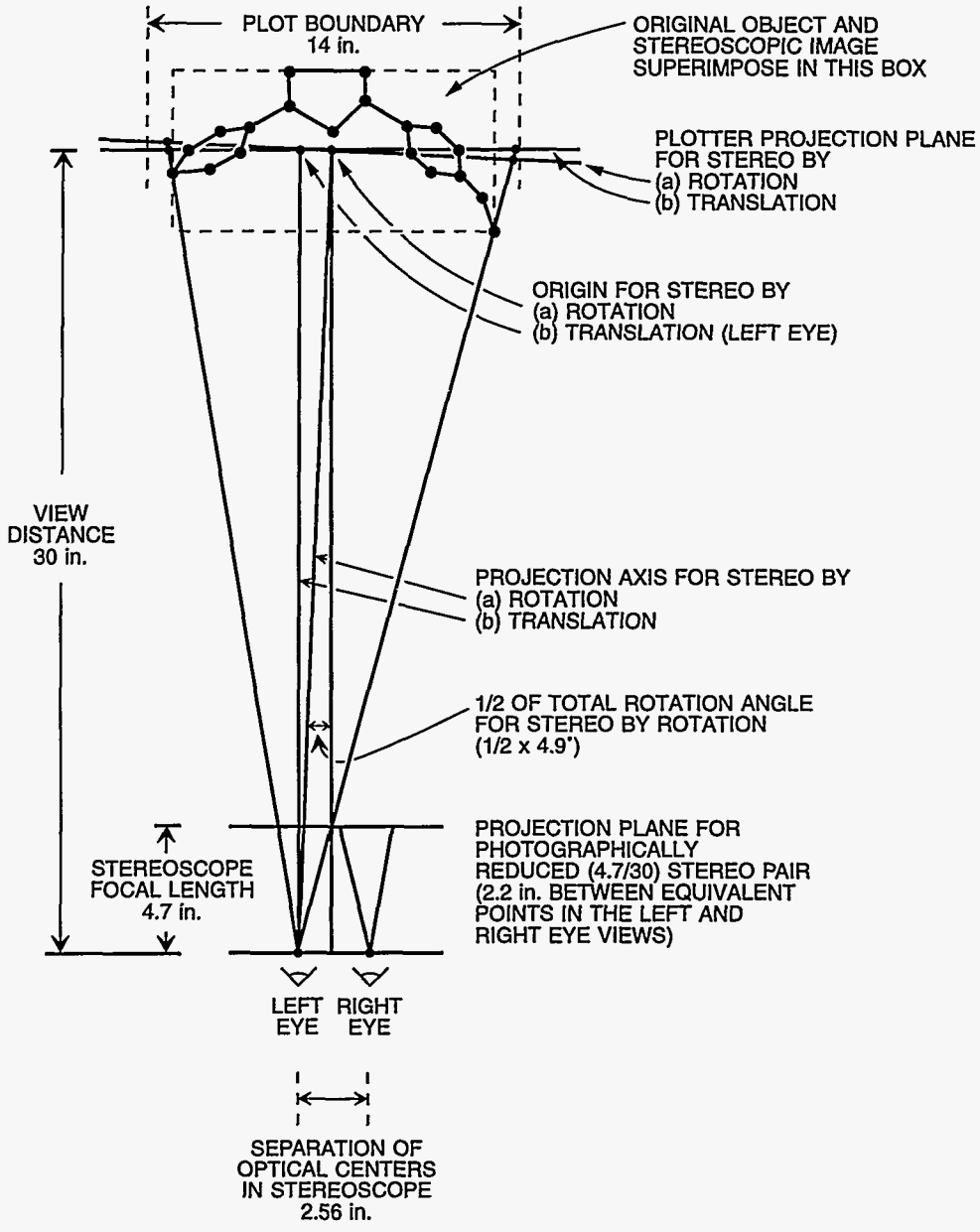

Fig. 5.3. Geometrical relations among the stereoscopic perspective projection parameters for a typical ORTEP drawing.

In Fig. 5.3, the available plotting area for each projection is assumed to be 14 in. horizontally and at least 11 in. vertically. The scaled mathematical object is within a box 12 in. wide, 9 in. (or more) high, and 6 in. deep with the plane of the plotter halfway back into the box. The stereoscopic image seen through a stereoscope with a 4.7-in. focal length and a 2.56-in. separation between optical centers should appear superimposed on the original object. The parameters for both "translation stereo" and "rotation stereo" are shown. The appropriate linear dimensions can be scaled to accommodate other plotting areas and still produce the same final stereoscopic image.

With the advent of high resolution printers/plotters, the need to reduce large drawings has diminished, and high quality drawings with the optimal origin separation can be produced directly. For those who view such drawings without the aid of a stereoscope, the viewing distance set in ORTEP should be the actual physical distance that will be used to view the pair. If a stereoscope is used for viewing, the viewing distance set in ORTEP is a function of the device's focal length and the location of the drawing's origin.

The cubane example in Section 2.4 can be used as a tool for determining the optimal view distance parameter. When the stereo cubane drawing is observed under usual viewing conditions, users should compare the front and back faces to decide if the molecule actually looks like a cube. If the faces appear out of proportion with each other, adjust the view distance parameter until the cube looks correct.

Theoretically, the best stereoscopic fidelity is obtained by translation of the origin rather than by stereo rotation of the object; however, the comparison of results given in Fig. 5.4 shows that the differences are indeed minor and nearly impossible to detect.

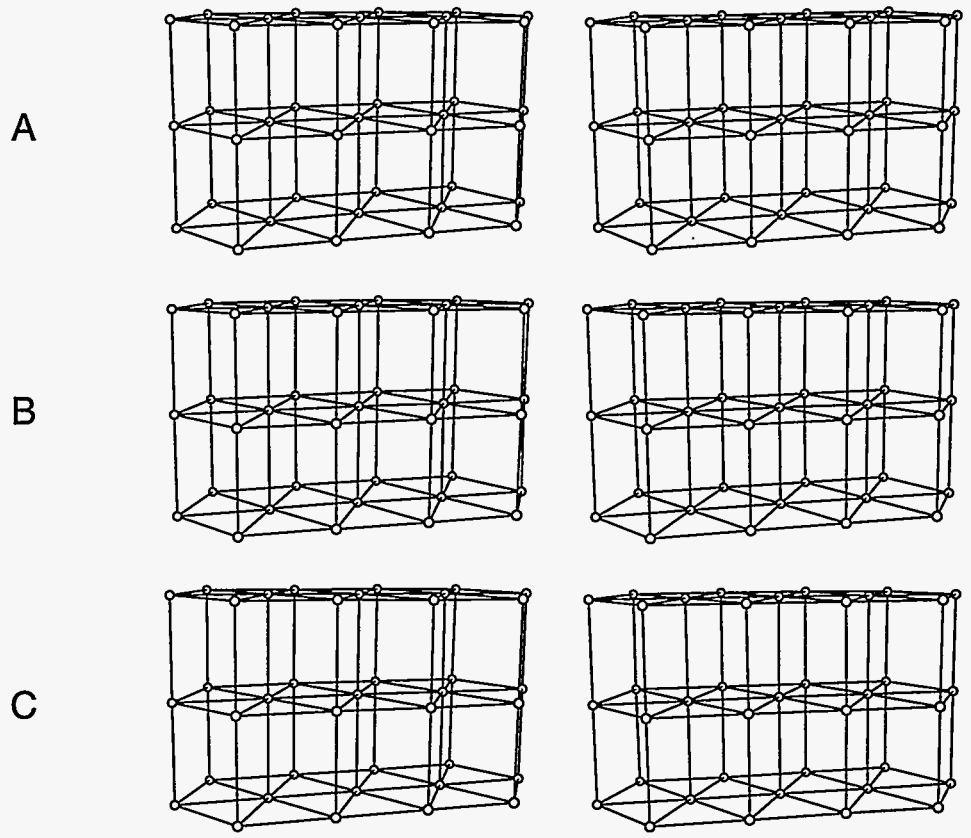

Fig. 5.4. Stereoscopic drawings of a hexagonal lattice with different stereoscopic parameters. (A) Stereo rotation of 4.9°, (B) stereo translation of 2.56 in., and (C) stereo rotation of 6.0°.

The top and middle drawings of Fig. 5.4 utilize the parameters derived in Fig. 5.3 and demonstrate that the differences predicted[13] for translation and rotation stereo are not discernible

in practice. The bottom stereo drawing illustrates the slight exaggeration in depth that occurs when a larger stereo rotation angle is used.

Fig. 5.4A was produced with the following stereo-rotation instructions for the left and right eye views,

```
503         2      2.45

503         2     -2.45
```

and stereo translation instructions were used for Fig. 5.4B.

```
504     -1.28         0         0

504      2.56         0         0
```

Remember that the 504 instruction (see 3.3.6.4) changes the origin of the *reference* Cartesian system while the 503 instruction (see 3.3.6.3) rotates the *working* Cartesian system. If additional drawings are to be made following a stereo translation, the reference system origin should be returned to its original position in order to prevent confusion.

```
504     -1.28         0         0
```

Also keep in mind that the 504 instruction (see 3.3.6.4) should not be used when the ellipsoids have internal structure because the octants selected for shading may not be the same in both views.

6. MATHEMATICS OF THERMAL-MOTION PROBABILITY ELLIPSOIDS

It is convenient to develop the physical significance of the anisotropic temperature factor with the notation and terminology of probability theory rather than with the more familiar Fourier transform theory. The results are, of course, identical regardless of the terminology used. The reason for this choice is that the literature of mathematical statistics and probability theory is somewhat neater and easier to follow. The texts by Wilks,[14] Cramer,[15] Miller,[16] Hamilton,[17] and Lukacs and Laha[18] and the handbooks by Burington and May[19] and Owen[20] are found to be particularly useful.

6.1 PROBABILITY DENSITY FUNCTION OF A TRIVARIATE NORMAL DISTRIBUTION

Given three chance variables X_1, X_2, X_3 and S, which is a region in X_1, X_2, X_3 space, the probability $P(S)$ that the point (X_1, X_2, X_3) falls in the region S is given by

$$P(S) = \int\int_S \int \phi(X_1, X_2, X_3)\, dX_1\, dX_2\, dX_3 . \tag{6.1.1}$$

If the integration is carried over all space, then

$$\int_{-\infty}^{\infty}\int\int \phi(X_1, X_2, X_3)\, dX_1\, dX_2\, dX_3 = 1 . \tag{6.1.2}$$

The function $\phi(X_1, X_2, X_3)$ is called the probability density function (pdf) for the joint distribution of X_1, X_2, X_3. Using vector notation, we can designate the pdf as $\phi(\mathbf{X})$.

When the distribution is the type said to be normal or Gaussian, the pdf is

$$\phi(\mathbf{X}) = \frac{[\det(\mathbf{M}^{-1})]^{1/2}}{(2\pi)^{3/2}} \exp[-\tfrac{1}{2}(\mathbf{X} - \hat{\mathbf{X}})^T \mathbf{M}^{-1}(\mathbf{X} - \hat{\mathbf{X}})] . \tag{6.1.3}$$

The matrix $\mathbf{M}^{-1}$ is the inverse of the symmetrical dispersion (variance-covariance) matrix $\mathbf{M}$, where

$$\mathbf{M} = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{12} & \sigma_1\sigma_3\rho_{13} \\ \sigma_1\sigma_2\rho_{12} & \sigma_2^2 & \sigma_2\sigma_3\rho_{23} \\ \sigma_1\sigma_3\rho_{13} & \sigma_2\sigma_3\rho_{23} & \sigma_3^2 \end{pmatrix} .$$

The symbols σ_i^2 represent the second moments or variance about the mean position $\hat{\mathbf{X}}$. The symbols $\sigma_i\sigma_j\rho_{ij}$ are the corresponding covariances and ρ_{ij} are the correlation coefficients.

6.2 EQUIPROBABILITY ELLIPSOIDS

For a proper normal distribution, the quadratic form $(\mathbf{X} - \hat{\mathbf{X}})^T \mathbf{M}^{-1}(\mathbf{X} - \hat{\mathbf{X}})$ is positive definite, and a principal axis transformation (see 6.4) is possible that will make the cross correlation coefficients $\rho_{ij} = 0$ $(i \neq j)$. The result of the transformation is the pdf

$$\phi(y_1, y_2, y_3) = \frac{1}{(2\pi)^{3/2}\sigma_{y_1}\sigma_{y_2}\sigma_{y_3}} e^{-Q/2}, \tag{6.2.1}$$

where

$$Q = \frac{(y_1 - \hat{y}_1)^2}{\sigma_{y_1}^2} + \frac{(y_2 - \hat{y}_2)^2}{\sigma_{y_2}^2} + \frac{(y_3 - \hat{y}_3)^2}{\sigma_{y_3}^2}. \tag{6.2.2}$$

The y_i are coordinates based on the Cartesian principal axis system and $\sigma_{y_i}^2$ are the variances along the principal axes, $i = 1, 2, 3$.

The normal probability density function is constant for points on the ellipsoid $Q = C^2$ where C is a constant. The probability that a random point (y_1, y_2, y_3) in the distribution will fall inside the ellipsoid is

$$P(C) = (2/\pi)^{1/2} \int_0^C r^2 e^{-r^2/2}\, dr. \tag{6.2.3}$$

This result is derived from Eqs. 6.1.1, 6.2.1, and 6.2.2 by transforming to spherical coordinates.

When $C = 1.5382$, $P = 0.5$ and the corresponding ellipsoid is called the 50% probability ellipsoid. Table 6.1 is a table of P versus C values that were calculated by integrating Eq. 6.2.3 using Gaussian quadrature. Quadruple precision calculations were required to match the values found on page 203 of Owen's handbook.[20]

6.3 CHARACTERISTIC FUNCTION OF A TRIVARIATE NORMAL DISTRIBUTION

The characteristic function $\Phi(\mathbf{T})$ corresponding to a trivariate distribution $\phi(\mathbf{X})$ is the expected value of $e^{i\mathbf{T}^T\mathbf{X}}$, namely,

$$\Phi(\mathbf{T}) = \int_{-\infty}^{\infty} \phi(\mathbf{X}) e^{i\mathbf{T}^T\mathbf{X}}\, d\mathbf{X}. \tag{6.3.1}$$

For the trivariate normal pdf, Eq. 6.1.3, the corresponding characteristic function is

$$\Phi(\mathbf{T}) = \exp\left[\, i\mathbf{T}^T\hat{\mathbf{X}} - \tfrac{1}{2}\mathbf{T}^T\mathbf{M}\mathbf{T}\,\right], \tag{6.3.2}$$

where $\mathbf{M}$ is the variance-covariance dispersion matrix described in Section 6.1 and $\hat{\mathbf{X}}$ is the center of mass of the distribution.

Table 6.1. Critical values for probability ellipsoids of a trivariate normal distribution.

P	C	P	C	P	C
0.01	0.3389	0.41	1.3842	0.81	2.1824
0.02	0.4299	0.42	1.4013	0.82	2.2114
0.03	0.4951	0.43	1.4183	0.83	2.2416
0.04	0.5479	0.44	1.4354	0.84	2.2730
0.05	0.5932	0.45	1.4524	0.85	2.3059
0.06	0.6334	0.46	1.4695	0.86	2.3404
0.07	0.6699	0.47	1.4866	0.87	2.3767
0.08	0.7035	0.48	1.5037	0.88	2.4153
0.09	0.7349	0.49	1.5209	0.89	2.4563
0.10	0.7644	0.50	1.5382	0.90	2.5003
0.11	0.7924	0.51	1.5555	0.91	2.5478
0.12	0.8192	0.52	1.5729	0.92	2.5997
0.13	0.8447	0.53	1.5904	0.93	2.6571
0.14	0.8694	0.54	1.6080	0.94	2.7216
0.15	0.8932	0.55	1.6257	0.95	2.7955
0.16	0.9162	0.56	1.6436	0.96	2.8829
0.17	0.9386	0.57	1.6616	0.97	2.9912
0.18	0.9605	0.58	1.6797	0.98	3.1365
0.19	0.9818	0.59	1.6980	0.99	3.3682
0.20	1.0026	0.60	1.7164	0.991	3.4019
0.21	1.0230	0.61	1.7351	0.992	3.4390
0.22	1.0430	0.62	1.7540	0.993	3.4806
0.23	1.0627	0.63	1.7730	0.994	3.5280
0.24	1.0821	0.64	1.7924	0.995	3.5830
0.25	1.1012	0.65	1.8119	0.996	3.6492
0.26	1.1200	0.66	1.8318	0.997	3.7325
0.27	1.1386	0.67	1.8519	0.998	3.8465
0.28	1.1570	0.68	1.8724	0.999	4.0331
0.29	1.1751	0.69	1.8932	0.9991	4.0607
0.30	1.1932	0.70	1.9144	0.9992	4.0912
0.31	1.2110	0.71	1.9360	0.9993	4.1256
0.32	1.2288	0.72	1.9580	0.9994	4.1648
0.33	1.2464	0.73	1.9804	0.9995	4.2107
0.34	1.2638	0.74	2.0034	0.9996	4.2661
0.35	1.2812	0.75	2.0269	0.9997	4.3365
0.36	1.2985	0.76	2.0510	0.9998	4.4335
0.37	1.3158	0.77	2.0757	0.9999	4.5943
0.38	1.3330	0.78	2.1012	0.99999	5.0894
0.39	1.3501	0.79	2.1274	0.999999	5.5376
0.40	1.3672	0.80	2.1544	0.9999999	5.9503

The crystallographic structure factor equation that incorporates general anisotropic temperature factor coefficients is

$$F(\mathbf{h}) = \sum_n f_n(\mathbf{h}) \exp(2\pi i \mathbf{h}^T \hat{\mathbf{X}}_n) \exp(-\mathbf{h}^T \mathbf{B}_n \mathbf{h}) , \tag{6.3.3}$$

where

- $\mathbf{h}$ is a vector giving the Miller indices,
- $\mathbf{X}_n$ is a vector giving the fractional unit cell coordinates of the nth atom,
- $\mathbf{B}_n$ is the anisotropic temperature factor coefficient matrix, and
- $f_n(\mathbf{h})$ is the atom form factor value for atom n.

If a change of variables $\mathbf{T} = 2\pi\mathbf{h}$ is made, then Eq. 6.3.3 can be rewritten as

$$F(\mathbf{T}) = \sum_n f_n(\mathbf{T}) \exp\left(i\mathbf{T}^T \hat{\mathbf{X}}_n - \tfrac{1}{2}\mathbf{T}^T \frac{\mathbf{B}_n}{2\pi^2} \mathbf{T} \right). \tag{6.3.4}$$

The scaled anisotropic temperature factor matrix $(1/2\pi^2)\mathbf{B}$ is seen to be identical with the variance-covariance dispersion matrix $\mathbf{M}$ in Eq. 6.3.2.

The corresponding crystal space trivariate normal pdf for any particular atom n is

$$\phi(\mathbf{X}) = \frac{[2\pi^2 \det(\mathbf{B}^{-1})]^{1/2}}{(2\pi)^{3/2}} \exp[-\pi^2(\mathbf{X}-\hat{\mathbf{X}})^T \mathbf{B}^{-1}(\mathbf{X}-\hat{\mathbf{X}})] ; \tag{6.3.5}$$

or if $\mathbf{M}^{-1} = 2\pi^2\mathbf{B}^{-1}$, then

$$\phi(\mathbf{X}) = \frac{\det(\mathbf{M}^{-1})}{(2\pi)^{3/2}} \exp[-\tfrac{1}{2}(\mathbf{X}-\hat{\mathbf{X}})^T \mathbf{M}^{-1}(\mathbf{X}-\hat{\mathbf{X}})] , \tag{6.3.6}$$

which is identical to Eq. 6.1.3.

6.4 PRINCIPAL AXIS TRANSFORMATION

The transformation of anisotropic temperature factor coefficients (for the general triclinic case) to principal axes of thermal motion is discussed by Waser,[21] Busing and Levy,[22] and Cruickshank *et al.*[23]

The principal axis transformation is necessary to find the thermal-motion probability ellipsoids discussed in Section 6.2. The principal axes of the matrix $\mathbf{M}^{-1}$ in Eq. 6.3.6 are the vectors $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$ for which the inner vector product $(\mathbf{y}_i, \mathbf{y}_i)$ has a stationary value subject to the constraint

$$(\mathbf{y}_i, \mathbf{M}^{-1}\mathbf{y}_i) = 1, \quad i = 1, 2, 3 . \tag{6.4.1}$$

For the general triclinic crystal system, this means that the quadratic form $\mathbf{y}^T\mathbf{G}^{-1}\mathbf{y}$ has a stationary value subjected to the constraint

$$\mathbf{y}^T\mathbf{G}^{-1}\mathbf{M}^{-1}\mathbf{y} = 1 \tag{6.4.2}$$

where $\mathbf{G}^{-1}$ is the metric tensor with components $\mathbf{a}_i \cdot \mathbf{a}_j$, where $\mathbf{a}_i \cdot \mathbf{a}_j$ is the scalar vector product of two of the three unit cell vectors. Introducing the Lagrange multiplier $1/\lambda$ leads to

$$\left[\mathbf{G}^{-1} - \frac{1}{\lambda_i}\mathbf{M}^{-1}\right]\mathbf{y}_i = 0 \quad (i = 1, 2, 3)\,; \tag{6.4.3}$$

premultiplying by $\mathbf{M}$ yields

$$\left[\mathbf{M}\mathbf{G}^{-1} - \frac{1}{\lambda_i}\mathbf{I}\right]\mathbf{y}_i = 0 \quad (i = 1, 2, 3)\,. \tag{6.4.4}$$

Or we can do some additional rearranging and obtain

$$[\,\mathbf{G}\mathbf{M}^{-1} - \lambda_i\mathbf{I}\,]\mathbf{y}_i = 0 \quad (i = 1, 2, 3)\,. \tag{6.4.5}$$

Eq. 6.4.4 is equivalent to one of the results derived by Busing and Levy,[22] except the λ_i obtained here are the reciprocals of their λ_i because we are doing the principal axis transformation on $\mathbf{M}^{-1}$ while their formulation performs the transformation on $\mathbf{M}$. The numerical procedure used in ORTEP finds the eigenvalues and eigenvectors of the unsymmetrical matrix $\mathbf{M}\mathbf{G}^{-1}$ in Eq. 6.4.4.

7. ORTEP EXAMPLES

This section includes several example structures that illustrate a number of the capabilities found in ORTEP-III. The ORTEP input file for each example is provided.

7.1 CELL PACKING – 5-HYDROXY-5-PHENYLNORBORNANONE

Two illustrations of 5-hydroxy-5-phenylnorbornanone are provided here. The data were obtained from a neutron diffraction study at room temperature.[24] The first illustration shows one complete molecule, and the second shows the contents of the unit cell.

The input file of the first structure illustrates ORTEP's original format for the symmetry operators. The atom parameter lines were taken directly from the output of a least squares refinement, and information is included there that is not needed by ORTEP. The extra information lies in card fields that are not required for ORTEP's operation and is ignored by the program. Note that the atoms are individually labeled with 901 instructions.

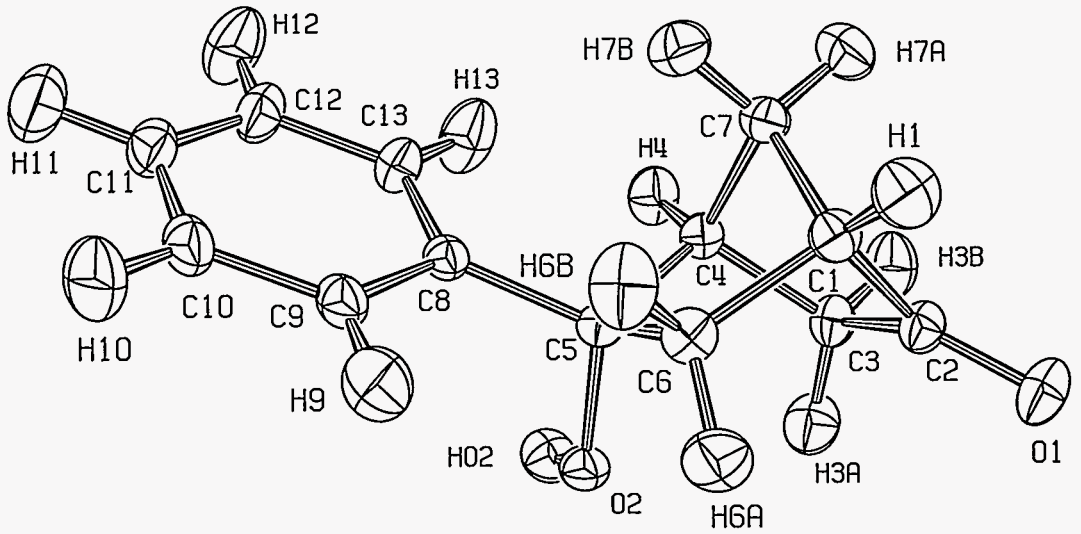

Fig. 7.1. Single molecule of 5-hydroxy-5-phenylnorbornanone.

```
 PHENYL HYDROXYL NORBORNANONE
   10.331   10.646   10.099       0.0 -.283810       0.0
              .0  1  0  0             .0  0  1  0              .0  0  0  1
              .0 -1  0  0             .0  0 -1  0              .0  0  0 -1
              .5 -1  0  0             .5  0  1  0              .5  0  0 -1
1             .5  1  0  0             .5  0 -1  0              .5  0  0  1
  C1      0.661000 1.000000 0.224802 0.001638 0.901515 0.0
 0.011930 0.006750 0.011647 0.000642 0.003989 0.000860            0  1  0
  C2      0.661000 1.000000 0.335043 0.050840 0.844329 0.0
 0.010432 0.007831 0.012112 0.001274 0.003947-0.000264            0  1  0
  C3      0.661000 1.000000 0.327553 0.192798 0.850375 0.0
 0.009129 0.007883 0.013469-0.000262 0.003784 0.000058            0  1  0
  C4      0.661000 1.000000 0.206609 0.213473 0.908171 0.0
 0.008568 0.006819 0.009641-0.000310 0.002118-0.001032            0  1  0
```

```
 C5      0.661000 1.000000 0.074610 0.177193 0.796977 0.0
0.008590 0.006732 0.008260-0.000670 0.002360 0.000103          0  1  0
 C6      0.661000 1.000000 0.091094 0.031659 0.790120 0.0
0.011368 0.006963 0.012116-0.001403 0.003143-0.001297          0  1  0
 C7      0.661000 1.000000 0.223679 0.102901 1.010406 0.0
0.011840 0.009377 0.008981 0.001423 0.002835 0.000648          0  1  0
 C8      0.661000 1.000000-0.053735 0.216149 0.834431 0.0
0.008328 0.007955 0.008995-0.000100 0.002374 0.001216          0  1  0
 C9      0.661000 1.000000-0.175896 0.157400 0.769638 0.0
0.008481 0.012396 0.011267-0.001406 0.001677 0.001394          0  1  0
 C10     0.661000 1.000000-0.296122 0.195813 0.795042 0.0
0.008526 0.016481 0.014906-0.000226 0.002743 0.004954          0  1  0
 C11     0.661000 1.000000-0.295600 0.294171 0.886394 0.0
0.011357 0.014259 0.016607 0.003003 0.006602 0.005712          0  1  0
 C12     0.661000 1.000000-0.175266 0.352517 0.950755 0.0
0.012936 0.012232 0.018491 0.002923 0.008046 0.001071          0  1  0
 C13     0.661000 1.000000-0.055392 0.314579 0.925104 0.0
0.010921 0.009871 0.014343 0.000394 0.005261-0.001410          0  1  0
 O1      0.577000 1.000000 0.414647-0.009305 0.801699 0.0
0.014915 0.010743 0.020270 0.002523 0.008903-0.001350          0  1  0
 O2      0.577000 1.000000 0.068052 0.227607 0.664516 0.0
0.010744 0.010771 0.008919-0.000196 0.003244 0.001269          0  1  0
 H1     -0.375000 1.000000 0.236869-0.096021 0.933996 0.0
0.020624 0.008781 0.019162 0.002373 0.007156 0.003208          0  1  0
 H3A    -0.375000 1.000000 0.318029 0.234292 0.749440 0.0
0.015989 0.012191 0.019604 0.000563 0.008507 0.003596          0  1  0
 H3B    -0.375000 1.000000 0.419746 0.227078 0.923493 0.0
0.011065 0.013144 0.022925-0.001035 0.003225-0.001808          0  1  0
 H4     -0.375000 1.000000 0.205933 0.307044 0.952030 0.0
0.012786 0.009537 0.016431-0.000243 0.003014-0.003491          0  1  0
 H6A    -0.375000 1.000000 0.092339 0.004576 0.685888 0.0
0.019268 0.012340 0.014229-0.000125 0.002520-0.004920          0  1  0
 H6B    -0.375000 1.000000 0.009600-0.018900 0.814691 0.0
0.014224 0.009647 0.026857-0.003061 0.006299 0.000168          0  1  0
 H7A    -0.375000 1.000000 0.316741 0.107454 1.094270 0.0
0.016598 0.016936 0.012393 0.003607 0.001040 0.000321          0  1  0
 H7B    -0.375000 1.000000 0.138340 0.092052 1.052878 0.0
0.017074 0.014780 0.014417 0.001660 0.007380 0.002011          0  1  0
 H9     -0.375000 1.000000-0.177376 0.080634 0.699229 0.0
0.014017 0.019179 0.019187-0.004154 0.001724-0.005322          0  1  0
 H10    -0.375000 1.000000-0.389010 0.146805 0.746257 0.0
0.010867 0.027857 0.025083-0.003438 0.003403 0.001062          0  1  0
 H11    -0.375000 1.000000-0.387954 0.322178 0.907661 0.0
0.015401 0.023080 0.028789 0.005406 0.011698 0.006558          0  1  0
 H12    -0.375000 1.000000-0.172867 0.427707 1.023741 0.0
0.021232 0.019665 0.030399 0.002995 0.013356-0.006345          0  1  0
 H13    -0.375000 1.000000 0.036386 0.362903 0.977271 0.0
0.015325 0.017015 0.026705-0.002320 0.007502-0.010296          0  1  0
 HO2    -0.375000 1.000000 0.071033 0.317748 0.672629 0.0
10.013588 0.011864 0.013832-0.001003 0.004119 0.003274         0  1  0
```

```
    201
    301      7.2       5.4        12       1.0
    401   155501 -2955501
    501   555501    655501    255501    255501    355501                    0
    502        2        28         1       2.7
    604                                    1.00
    601      4.0       1.5
2   1001
         1 29   1 29   3    .80    1.6    .05
    716
2   812
         1 29   1 29   3    .80    1.6    .05
    901   155501                                          .10      -.05      -.22
    901   255501                                          .10      +.10      -.22
    901   355501                                          .10      +.17      -.22
    901   455501                                          .10      +.05     -.212
    901   555501                                          .10      -.20      -.12
    901   655501                                          .10      -.17      -.22
    901   755501                                          .10      -.25      -.03
    901   855501                                          .10     -.063      -.22
    901   955501                                          .10      -.27      -.07
    901  1055501                                          .10      +.05      -.30
    901  1155501                                          .10      -.25      -.17
    901  1255501                                          .10      +.27      +.07
    901  1355501                                          .10      -.12      +.27
    901  1455501                                          .10      +.02      -.30
    901  1555501                                          .10      +.22     -.185
    901  1655501                                          .10      0.00      +.27
    901  1755501                                          .10      +.10      -.27
    901  1855501                                          .10      +.35      +.05
    901  1955501                                          .10      0.00      +.27
    901  2055501                                          .10      +.05     -.265
    901  2155501                                          .10      -.40      +.10
    901  2255501                                          .10      +.35      0.00
    901  2355501                                          .10      -.40      0.00
    901  2455501                                          .10      -.33      -.07
    901  2555501                                          .10      0.00      -.33
    901  2655501                                          .10      -.05      -.30
    901  2755501                                          .10      +.30      +.12
    901  2855501                                          .10     +.012      +.32
    901  2955501                                          .10     -.475      0.00
    202
     -1
```

This second example shows the packing of the norbornanone molecules in a unit cell. The unit cell contains four molecules, but six have been drawn. The hydrogen atoms have been omitted for clarity, and the carbon and oxygen atoms have been drawn with different representations for easier visual identification. (The oxygen atoms are the ellipsoids with the shaded octant.) The input shows the new format for the symmetry operators that is available in ORTEP-III. In the atom parameters, dummy atoms have been provided for a corner of the unit cell (atom #16 at 0.,0.,0.) and one for its center (atom #17 at .5,.5,.5).

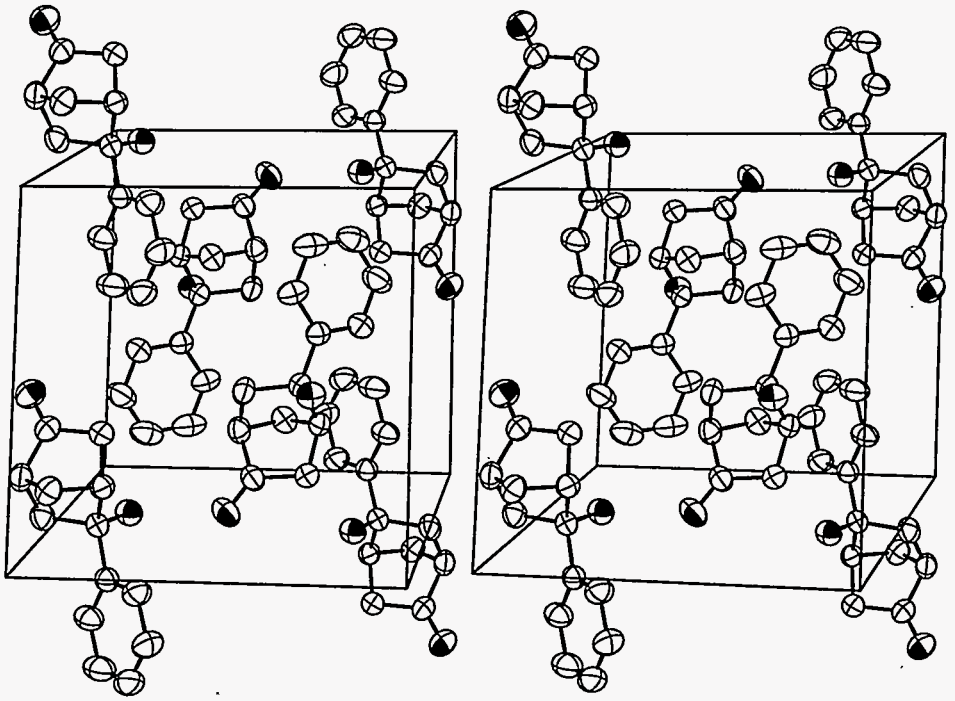

Fig. 7.2. Packing diagram of 5-hydroxy-5-phenylnorbornanone.

```
 PHENYL HYDROXYL NORBORNANONE
1  10.331    10.646    10.099       0.0 -.283810       0.0
  x,y,z
  -x,-y,-z
  1/2-x,1/2+y,1/2-z
1 1/2+x,1/2-y,1/2+z
  C1       0.661000 1.000000 0.224802 0.001638 0.901515 0.0
 0.011930 0.006750 0.011647 0.000642 0.003989 0.000860              0  1  0
  C2       0.661000 1.000000 0.335043 0.050840 0.844329 0.0
 0.010432 0.007831 0.012112 0.001274 0.003947-0.000264              0  1  0
  C3       0.661000 1.000000 0.327553 0.192798 0.850375 0.0
 0.009129 0.007883 0.013469-0.000262 0.003784 0.000058              0  1  0
  C4       0.661000 1.000000 0.206609 0.213473 0.908171 0.0
 0.008568 0.006819 0.009641-0.000310 0.002118-0.001032              0  1  0
  C5       0.661000 1.000000 0.074610 0.177193 0.796977 0.0
 0.008590 0.006732 0.008260-0.000670 0.002360 0.000103              0  1  0
  C6       0.661000 1.000000 0.091094 0.031659 0.790120 0.0
 0.011368 0.006963 0.012116-0.001403 0.003143-0.001297              0  1  0
  C7       0.661000 1.000000 0.223679 0.102901 1.010406 0.0
 0.011840 0.009377 0.008981 0.001423 0.002835 0.000648              0  1  0
  C8       0.661000 1.000000-0.053735 0.216149 0.834431 0.0
 0.008328 0.007955 0.008995-0.000100 0.002374 0.001216              0  1  0
  C9       0.661000 1.000000-0.175896 0.157400 0.769638 0.0
 0.008481 0.012396 0.011267-0.001406 0.001677 0.001394              0  1  0
  C10      0.661000 1.000000-0.296122 0.195813 0.795042 0.0
 0.008526 0.016481 0.014906-0.000226 0.002743 0.004954              0  1  0
  C11      0.661000 1.000000-0.295600 0.294171 0.886394 0.0
```

```
 0.011357 0.014259 0.016607 0.003003 0.006602 0.005712           0  1  0
  C12     0.661000 1.000000-0.175266 0.352517 0.950755 0.0
 0.012936 0.012232 0.018491 0.002923 0.008046 0.001071           0  1  0
  C13     0.661000 1.000000-0.055392 0.314579 0.925104 0.0
 0.010921 0.009871 0.014343 0.000394 0.005261-0.001410           0  1  0
  O1      0.577000 1.000000 0.414647-0.009305 0.801699 0.0
 0.014915 0.010743 0.020270 0.002523 0.008903-0.001350           0  1  0
  O2      0.577000 1.000000 0.068052 0.227607 0.664516 0.0
 0.010744 0.010771 0.008919-0.000196 0.003244 0.001269           0  1  0
ORIGIN                           0.0      0.0      0.0
      .01
CENTER                           0.5      0.5      0.5
1
      201
      301      5.4      5.4       12      1.0
# Store unit cell corners for cell outline
      401  1655501 -1666601
# Find and store all atoms within 5.5 A of unit cell center
      402  1755501       17        1       15      5.5
# Reiterative convolution around found atoms to complete molecules
      406        1       15        1       15       2.
      501  1655501  1655501  1656501  1655501  1655601                 0
      502        3      180        1       10        2      -10
      604
      503        2      2.7
     1101
  2  1001        1
  2        1 15  1 15  3   .80   1.6   .03
          16 16 16 16  1   10.   11.   .01
# Different representations for carbons (1-13) and oxygens (14-15)
  1   702
                                   1       13
  1   701
                                  14       15
  2   802
  2        1 15  1 15  3   .80   1.6   .03
          16 16 16 16  1   10.   11.   .01
     1102
      202    2.375
      503        2     -2.7
     1103
      202
       -1
```

7.2 HELICAL STRUCTURE – POLY-L-ALANINE

The structure of poly-L-alanine was published by Elliott and Malcolm in 1959.[25] The Pauling and Corey right-handed alpha helix repeats after 13 turns and 47 residues and can be represented in ORTEP by 47 symmetry cards with $N = 47$; $M = 13$; $L = 0, 1, \ldots, 46$; $T_1, T_2, T_3 = 0$. The input atom list then contains the contents of one residue. In this example there are 48 symmetry cards with operator 1 and operator 48 related by one cell translation along c.

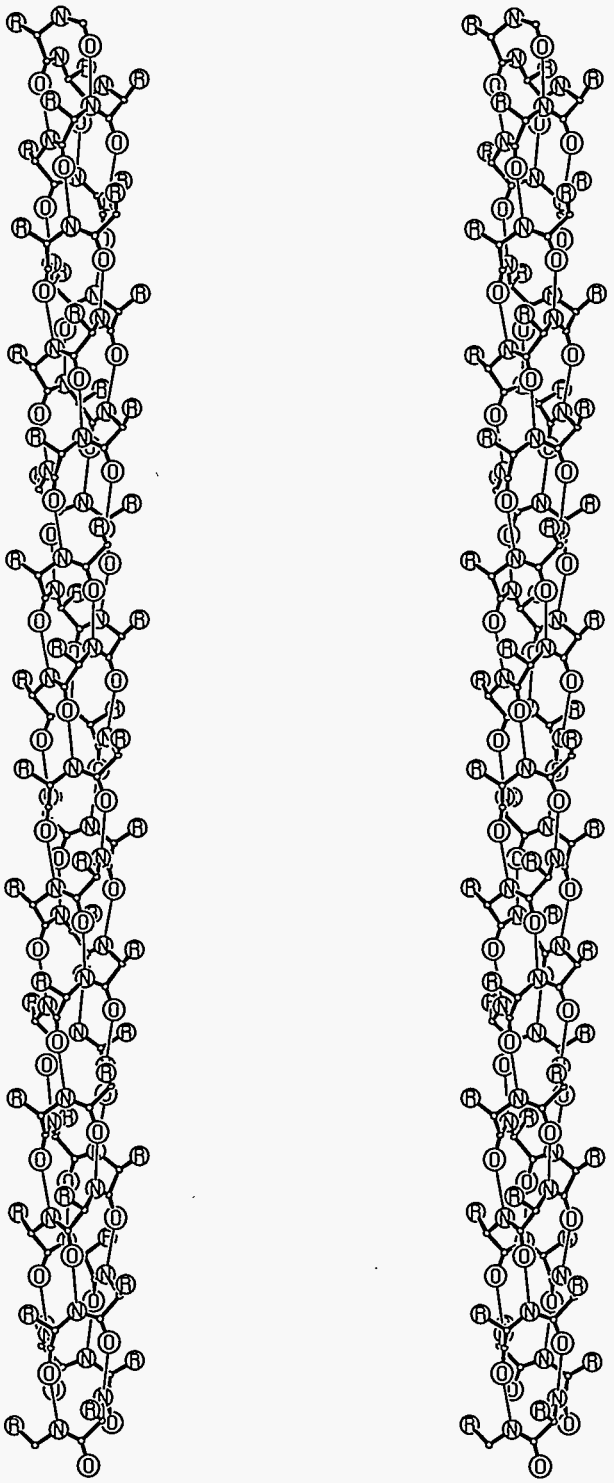

Fig. 7.3. 47/13 α-Helix of poly-L-alanine. The thin "vertical" lines between nitrogen and oxygen atoms indicate a hydrogen bond path.

```
POLY-L-ALANINE  47/13 HELIX      ELLIOTT AND MALCOLM (1959)
   8.55      8.55      70.3       90.       90.      120.
                                                          0 13 47
                                                          1 13 47
                                                          2 13 47
                                                          3 13 47
                                                          4 13 47
                                                          5 13 47
                                                          6 13 47
                                                          7 13 47
                                                          8 13 47
                                                          9 13 47
                                                         10 13 47
                                                         11 13 47
                                                         12 13 47
                                                         13 13 47
                                                         14 13 47
                                                         15 13 47
                                                         16 13 47
                                                         17 13 47
                                                         18 13 47
                                                         19 13 47
                                                         20 13 47
                                                         21 13 47
                                                         22 13 47
                                                         23 13 47
                                                         24 13 47
                                                         25 13 47
                                                         26 13 47
                                                         27 13 47
                                                         28 13 47
                                                         29 13 47
                                                         30 13 47
                                                         31 13 47
                                                         32 13 47
                                                         33 13 47
                                                         34 13 47
                                                         35 13 47
                                                         36 13 47
                                                         37 13 47
                                                         38 13 47
                                                         39 13 47
                                                         40 13 47
                                                         41 13 47
                                                         42 13 47
                                                         43 13 47
                                                         44 13 47
                                                         45 13 47
                                                         46 13 47
                                                         47 13 47
1
                                 1.63      94.9      -.40        3
       .1                                                        7
```

```
                                2.29      20.7      -.81         3
      .1                                                        7
  R                             3.17       0.        0.         3
      .3                                                        7
  N                             1.49      49.7       .06         3
      .3                                                        7
  0                             1.98      104.     -1.58         3
     .35                                                        7
 ORGN                             .0         0        .5
1
# Initialize plotting
      201
# Landscape drawing orientation
      301      8.5      2.0       15      0.5
# Rotate lettering for landscape orientation
      302      -90
# Store atoms to be drawn
      401   155501  -555548
# Define coordinate system
      501   655501   155501   155601   155501   165501
# Rotate structure for landscape orientation
      502        1      90.
# Automatic position and scale
      604
# Shift plot origin for left eye view
      202        0        5
# Stereo rotation for left eye view
      503        1      2.5
# Start save sequence
     1101
# Calculate overlap
  2  1001
  2        1  5  1  5  1   1.1   1.6  .050
           4  4  5  5  1   2.7   3.0  .010
# Draw atoms and labels
      714                                              .07      .03
# Draw covalent bonds and inter-residue hydrogen bonds
  2   812
  2        1  5  1  5  1   1.1   1.6  .050
           4  4  5  5  1   2.7   3.0  .010
# End save sequence
     1102
# Stereo rotation for right eye view
      503        1     -2.5
# Shift plot origin for right eye view (view separation = 2.375 in.)
      202        0    2.625
# Repeat save sequence
     1103
# Terminate plotting
      202
# Terminate ORTEP
       -1
```

7.3 COORDINATION POLYHEDRA – POTASSIUM PERXENATE NONAHYDRATE

The crystal structure of this hydrated ionic material was published by Zalkin *et al.* in 1964.[26] The only covalent bonds are between the xenon and oxygen atoms in the perxenate anions (the darker bonds in Fig. 7.4). To see better how the oxygens of the perxenate anions and water molecules coordinate around the potassium and atoms, lines have been drawn from the potassiums to all oxygens within a distance of 3.3 Å.

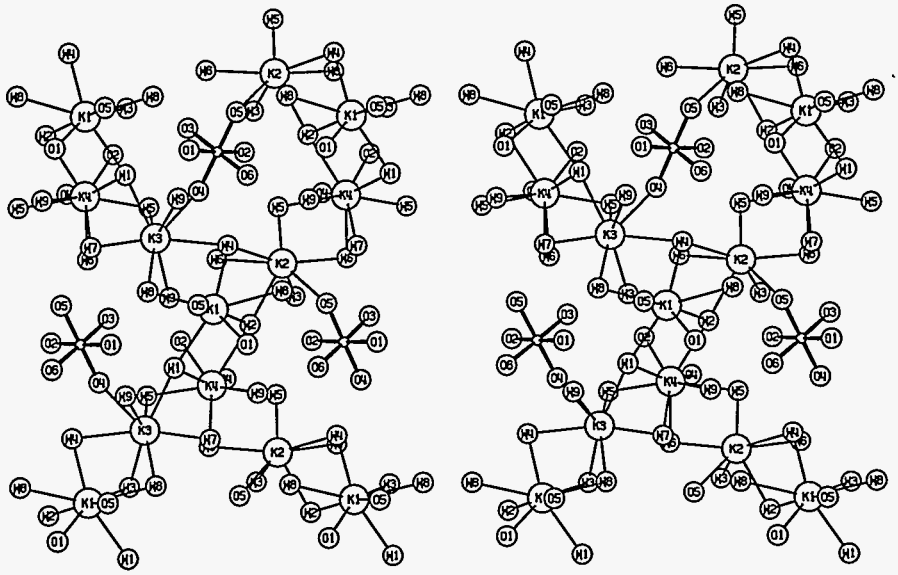

Fig. 7.4. Coordination polyhedra in potassium perxenate nonahydrate.

```
 POTASSIUM PERXENATE 9-HYDRATE/A.ZALKIN ET AL (1964) JACS 86,3569
1   9.049   10.924   15.606      90.      90.      90.
   x,y,z
  -x,-y,1/2+z
  -x,1/2+y,z
1  x,1/2-y,1/2+z
  XE                          .249     .988     .250
      .10                                                  7
  K1                          .628     .987     .339
      .30                                                  7
  K2                          .846     .238     .958
      .30                                                  7
  K3                          .307     .227     .026
      .30                                                  7
  K4                          .877     .989     .139
      .30                                                  7
  O1                          .403     .101     .251
      .20                                                  7
  O2                          .094     .878     .253
      .20                                                  7
  O3                          .138     .096     .316
      .20                                                  7
  O4                          .176     .058     .151
```

```
      .20                                                          7
  O5                          .323      .918      .351
      .20                                                          7
  O6                          .360      .881      .188
      .20                                                          7
  W1                          .654      .839      .190
      .20                                                          7
  W2                          .850      .136      .297
      .20                                                          7
  W3                          .873      .829      .369
      .20                                                          7
  W4                          .692      .046      .506
      .20                                                          7
  W5                          .997      .243      .111
      .20                                                          7
  W6                          .967      .980      .972
      .20                                                          7
  W7                          .376      .470      .002
      .20                                                          7
  W8                          .493      .262      .886
      .20                                                          7
  W9                          .606      .150      .124
      .20                                                          7
 ORGN                         .250      .500      .400
      .10                                                          7
                              .000      .000      .000
1     .03                                                         7
      201
      301      2.6      3.6       15     0.25
# Locate and store K and Xe atoms
      404       21       21        1        5      .13      .52      .57
# Convolute sphere of enclosure with each central atom
      405        1        5        1       20     3.30
# B axix horizontal, C axis vertical, viewed along -A axis
      501  2155501  2155501  2156501  2155501  2155601                  1
      604                                 2.
      503        2        3
     1101
  2  1001        1
  2        1  1  6 11  3   1.8   1.9   .04
           2  5  6 20  1   2.6   3.3   .01
  1   714                                          .04
                                  2       20
# Xe not labeled
  1   714
                                  1        1
  2   812
  2        1  1  6 11  3   1.8   1.9   .04
           2  5  6 20  1   2.6   3.3   .01
     1102
      202    2.375
      503        2       -3
```

```
1103
 202
  -1
```

In the following representation of the same structure shown in Fig. 7.4, only the xenon and potassium atoms have been explicitly drawn. (The xenon atoms are the smaller circles.) The oxygen atoms are shown implicitly as the vertices of polyhedra centered on the potassium and xenon atoms. As in the previous case, oxygens within 3.3 Å of the potassium are treated as making up the coordination polyhedron. Only the instruction portion of the input file is provided. The input lines that precede these are the same as those in the previous case.

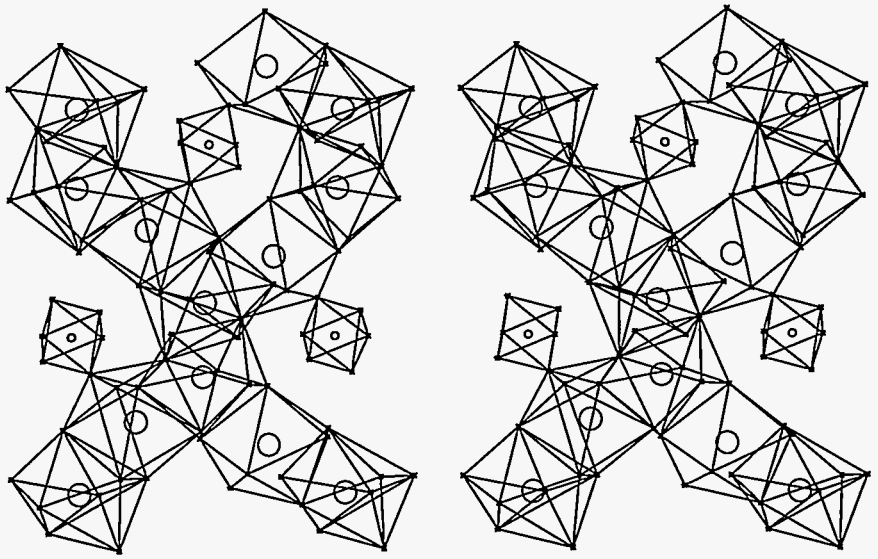

Fig. 7.5. Coordination polyhedra in potassium perxenate nonahydrate.

```
      201
      301      2.6      3.6       15     0.25
# Locate and store K and Xe atoms
      404       21       21        1        5      .13       .52       .57
# Convolute sphere of enclosure with each central atom
      405        1        5        1       20     3.30
# B axix horizontal, C axis vertical, viewed along -A axis
      501  2155501  2155501  2156501  2155501  2155601                    1
      604
      503        2        3
     1101
# Draw xenon and potassium atoms only
  1   714
                                    1        5
# Use polygon radii to limit bonds drawn
  2   813
  2         6 11  6 11      2.5   2.8          -1     1   1.8   1.9
            6 20  6 20      2.6   4.6          -2     5   2.6   3.3
     1102
      202    2.375
      503        2       -3
```

```
1103
 202
  -1
```

7.4 ATOM FEATURES – LYSOSOME MUTANT POLYPEPTIDE

The data for this example were taken from the Protein Data Bank #216L. The header information from that file is provided below.

```
HEADER    HYDROLASE(O-GLYCOSYL)                        10-MAY-94   216L
COMPND    LYSOZYME (E.C.3.2.1.17) MUTANT WITH SER 44 REPLACED BY TRP,
COMPND   2 CYS 54 REPLACED BY THR, CYS 97 REPLACED BY ALA (S44W,
COMPND   3 C54T, C97A)
SOURCE    BACTERIOPHAGE T4 (MUTANT GENE DERIVED FROM THE M13
SOURCE   2 PLASMID BY CLONING THE T4 LYSOZYME GENE)
AUTHOR    M.BLABER,B.W.MATTHEWS
REVDAT   1   31-JUL-94 216L    0
SPRSDE     31-JUL-94 216L      116L
JRNL        AUTH   M.BLABER,X.-J.ZHANG,B.W.MATTHEWS
JRNL        TITL   STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO
JRNL        TITL 2 SITES IN T4 LYSOZYME
JRNL        REF    SCIENCE                          V. 260  1637 1993
JRNL        REFN   ASTM SCIEAS  US ISSN 0036-8075                  0038
```

Only the first 63 amino acids (500 atoms) of the protein were used for this example since that is the size used in the dimension statements in ORTEP-III. The first 500 ATOM lines were extracted from the PDB file and placed unaltered in a file named ATOMS.DAT. A few of the lines are shown below.

```
ATOM      1  N   MET A   1      82.486  23.405  25.378  1.00 29.06      216L 127
ATOM      2  CA  MET A   1      81.291  22.758  24.885  1.00 15.78      216L 128
ATOM      3  C   MET A   1      80.495  23.789  24.150  1.00 33.32      216L 129
ATOM      4  O   MET A   1      80.951  24.925  24.017  1.00 29.09      216L 130
ATOM      5  CB  MET A   1      80.556  22.168  26.090  1.00 14.87      216L 131
ATOM      6  CG  MET A   1      79.353  21.283  25.811  1.00 44.92      216L 132
ATOM      7  SD  MET A   1      78.906  20.301  27.306  1.00 34.12      216L 133
ATOM      8  CE  MET A   1      80.536  19.686  27.844  1.00  7.96      216L 134
ATOM      9  N   ASN A   2      79.348  23.416  23.650  1.00  7.39      216L 135
ATOM     10  CA  ASN A   2      78.619  24.379  22.897  1.00 14.21      216L 136
  .              .               .              .              .            .
  .              .               .              .              .            .
  .              .               .              .              .            .
ATOM    491  CB  GLU A  62      69.880  12.430   7.589  1.00  6.64      216L 617
ATOM    492  CG  GLU A  62      70.251  11.994   6.135  1.00  1.34      216L 618
ATOM    493  CD  GLU A  62      69.487  10.795   5.671  1.00 27.84      216L 619
ATOM    494  OE1 GLU A  62      68.805  10.091   6.416  1.00 19.47      216L 620
ATOM    495  OE2 GLU A  62      69.547  10.652   4.368  1.00 34.69      216L 621
ATOM    496  N   ALA A  63      70.531  13.275  10.600  1.00 32.33      216L 622
ATOM    497  CA  ALA A  63      70.126  13.774  11.873  1.00  6.04      216L 623
ATOM    498  C   ALA A  63      70.877  15.054  12.241  1.00 55.04      216L 624
ATOM    499  O   ALA A  63      70.278  16.027  12.662  1.00 13.24      216L 625
ATOM    500  CB  ALA A  63      70.323  12.701  12.964  1.00 18.46      216L 626
```

Since these atom data are not in the standard format used by ORTEP, subroutine READIN was written to read this particular format. It is shown below. As each atom is read by READIN, the subroutine sets the value of FEATURE #2 (id2) for the atom to the sequence number of the amino acid containing the atom. FEATURE #1 (id1) is set to a value that indicates the type of atom:

1	peptide link N
2	alpha carbon
3	carbon of C=O in peptide link
4	oxygen of C=O in peptide link
9	all other atoms

```
      subroutine readin(iu,chem,id1,id2,x1,x2,x3,it,is,b1,b2,b3,b4,
     1                  b5,b6,btype)
      integer*2 id1,id2
      character*1 chain
      character*3 res
      character*4 atom
      character*6 rec
      character*8 chem
      b1=.1
      b2=0
      b3=0
      b4=0
      b5=0
      b6=0
      btype=7.
      id1=0
      id2=0
      it=2
      read (iu,201) rec,iserno,atom,res,chain,id2,x1,x2,x3,occ,tf
  201 format(a6,i5,1x,a4,1x,a3,1x,a1,i4,4x,3f8.0,2f6.0)
      id1=9
      if (atom.eq.' N  ') id1=1
      if (atom.eq.' CA ') id1=2
      if (atom.eq.' C  ') id1=3
      if (atom.eq.' O  ') then
         id1=4
         b1=.15
      end if
      chem=atom(2:4)//res
      is=0
c *** check if another data record is available
      read (iu,202,end=203) rec
  202 format(a6)
      backspace(iu)
      return
  203 is=1
      return
      end
```

The ORTEP input file contains the instructions for producing three different illustrations. Each begins with a 201 instruction and ends with a 202. The second and third sets make use of the assigned atom features to select particular atoms for drawing. In the input file, a "2" in column 1 of the final symmetry card tells ORTEP (1) that the atom data are in a separate file and (2) to use subroutine READIN to read the data.

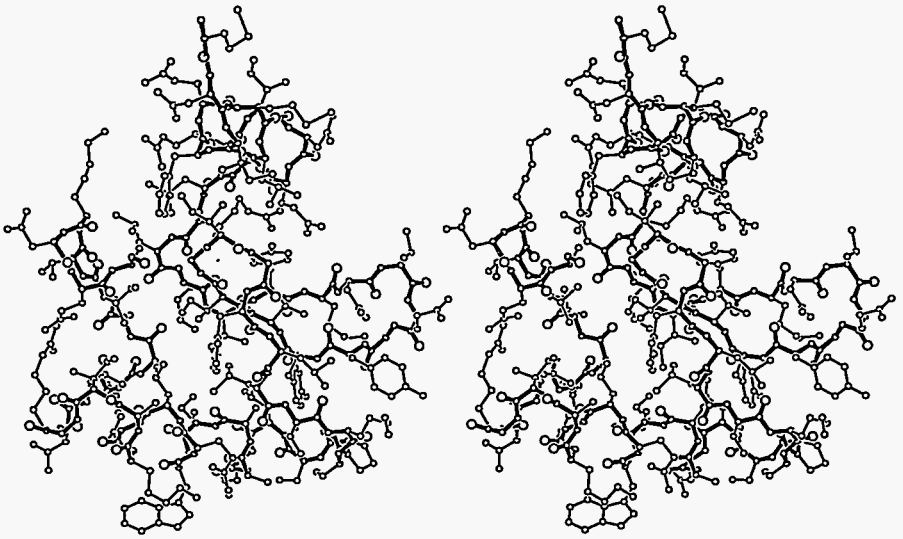

Fig. 7.6. First 63 amino acids of lysosome mutant protein.

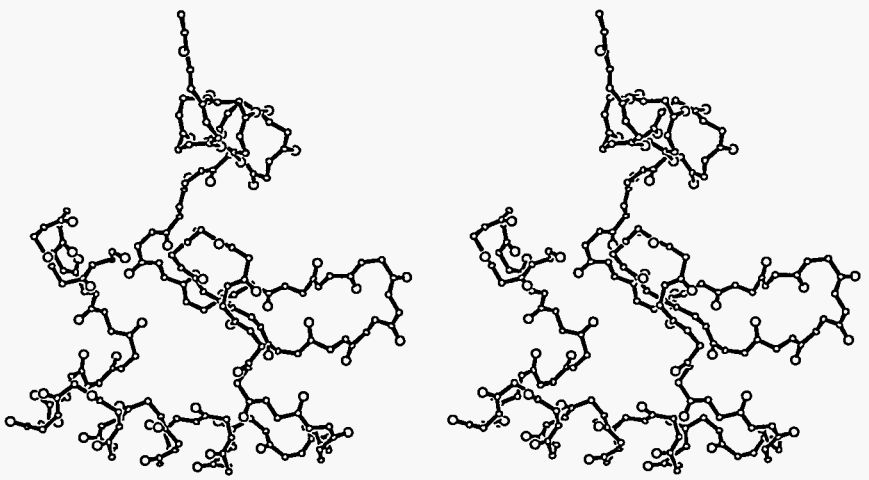

Fig. 7.7. First 63 amino acids of lysosome mutant protein with side chains eliminated.

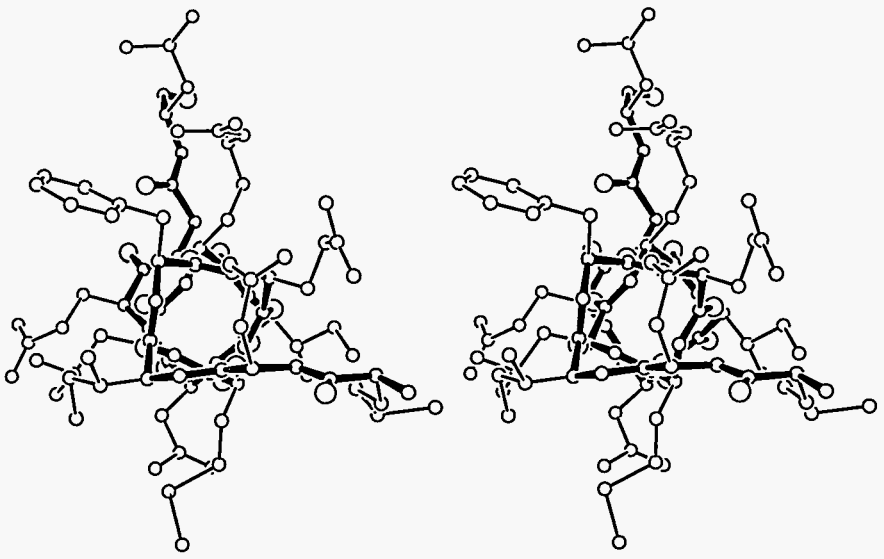

Fig. 7.8. First 13 amino acids of lysosome mutant protein, looking through α-helix.

```
 LYSOZYME MUTANT PROTEIN DATA BANK #216L - BLABER AND MATTHEWS
1 116.500   54.400   59.500    90.00   102.30    90.00
     X,Y,Z
     -X,Y,-Z
     X+1/2,Y+1/2,Z
2    -X+1/2,Y+1/2,-Z
# Polypeptide containing first 63 amino acids of protein.
      201
      301       5.0       3.5       15.        .4
      401   155501-50055501
      506
      502         3        90         1        35
      604                                     2.
      503         2       2.7
     1101
  2  1001                    1
  2         1  4  1  4  5   0.9   2.0   .08
  2         1  4  5  9  1   0.9   2.0   .02
            5  9  5  9  1   0.9   2.0   .02
  1   714
                                      1         9         1
  2   812                     1
  2         1  4  1  4  5   0.9   2.0   .08
  2         1  4  5  9  1   0.9   2.0   .02
            5  9  5  9  1   0.9   2.0   .02
     1102
      202       2.3
      503         2      -2.7
     1103
      202
# 63 amino acid polypeptide with side chains eliminated.
```

```
# The origin, axes, and scale are unchanged from above.
      201
      301       5.0       3.5       15.        .4
      410
  2   402    155501        500         1       500       2.0
                 1  4  1
      503         2       2.7
     1101
  2  1001                    1
           1  4  1  4  5    0.9    2.0    .08
  1   714
                                       1         4         1
  2   812                     1
           1  4  1  4  5    0.9    2.0    .08
     1102
      202       2.3
      503         2      -2.7
     1103
      202
# First 13 amino acids looking through alpha helix.
      201
      301       5.0       3.5       15.        .4
      410
  2   402    155501        500         1       500       2.0
           1 13  1 13  2
      506
      502         2       100         1        15
      604                                      2.
      503         2       2.7
     1101
  2  1001                    1
  2           1  4  1  4  5    0.9    2.0    .08
  2           1  4  5  9  1    0.9    2.0    .02
              5  9  5  9  1    0.9    2.0    .02
  1   714
                                       1        13         2
  2   812                     1
  2           1  4  1  4  5    0.9    2.0    .08
  2           1  4  5  9  1    0.9    2.0    .02
              5  9  5  9  1    0.9    2.0    .02
     1102
      503         2      -2.7
      202       2.3
     1103
      202
       -1
```

7.5 CRITICAL NET – SODIUM CHLORIDE

ORTEP-III can produce "critical net" illustrations that depict some canonical topological characteristics of the global ensemble of overlapping atomic-thermal-motion Gaussian density

functions in a crystal. Non-degenerate critical points occur where the first derivative of the global density is zero and the second derivative is a 3×3 symmetric matrix with a non-zero determinant. The signs of the three eigenvalues of the second derivative matrix specify the types of critical points, which are termed peak (–,–,–), pass (+,–,–), pale (+,+,–) and pit (+,+,+). Peaks correspond to density maxima, pits to density minima, and passes and pales to saddle points in the density function. The four types of critical points represent 0 (e.g., vertex), 1 (e.g., edge), 2 (e.g., face), and 3 (e.g., body) dimensional cells in the topological Morse function CW complex (i.e., C for closure finite, W for weak topology), simply called a critical net, and correspond with the number of + signs in the sign signature for each critical point. The most gradual up-density path from a pit to a peak follows the sequence pit → pale → pass → peak. A discussion of critical nets can be found on the World Wide Web at http://www.ornl.gov/ortep/topology/critnet.html.

Fig. 7.9 illustrates the critical net for NaCl with the larger corner spheres representing Cl peaks; the smaller corner spheres, Na peaks; the elongated "cigar-shaped" ellipsoids, passes; the flattened "pancake-shaped" ellipsoids, pales; and the smallest sphere in the center, a pit. The paths connecting the critical points, shown by the connection "bonds" in Fig. 7,9, are topologically unique.

New in ORTEP-III is the method for specifying the orientations and sizes of the elongated and flattened ellipsoids without giving their quadratic form coefficients. The temperature factor card following the atom parameter card for a pass or pale has the format:

Columns	
1	A sentinel $\neq 0$ if last atom
2-9	Unique axis length (Å)
10-18	Second (and third) axis length (Å)
19-27	VDC_1 (from)
28-36	VDC_1 (to)
37-45	[VDC_2 (from)
46-54	VDC_2 (to)]
55-63	7

VDC_1 is a vector parallel with the unique axis of the cigar-shaped pass or pancake-shaped pale and VDC_2 is a second vector *not* parallel with VDC_1 such that $VDC_1 \times VDC_2$ is a second principal axis of that ellipsoid. If VDC_1 and VDC_2 are parallel, VDC_2 is replaced by a suitable lattice translation vector. VDC_2 may be omitted from the input if desired, and the program will choose one of the three lattice vectors for VDC_2.

This example illustrates an important point about the relationship between the symmetry operators and atom input data unrelated to the fact that this is a critical net drawing. Sodium chloride crystallizes in space group $Fm\bar{3}m$, which has 192 symmetry operators. Of these, 48 are "unique", and the others may be obtained from these by adding the centering translations. The centering translations in this space group are (0,0,0), (0,.5,.5), (.5,0,.5), and (.5,.5,0). As discussed earlier (see 3.2.3), if all the symmetry operators are not provided in the ORTEP input file, each unique atom will require multiple entries with those centering translations added that are not provided in the symmetry cards. In this case only 48 symmetry operators have been included

(although ORTEP-III allows a maximum of 96). As a consequence, each atom has four entries, obtained by adding the centering translation values to the atom's positional coordinates. If 96 operators had been included, each atom would have required two entries. The symmetry operators are provided in ORTEP's original format.

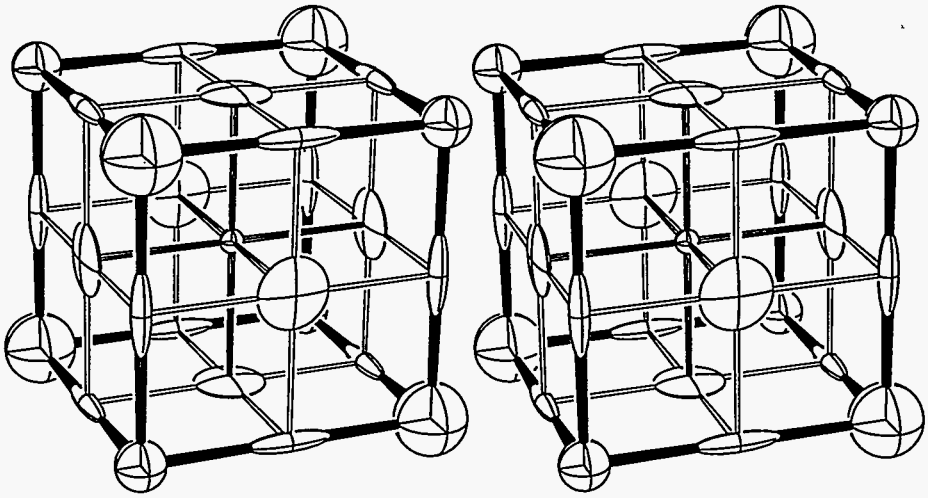

Fig. 7.9. Sodium chloride critical net.

```
NaCl Fm3m peak-a,b=m3m; pit c=4bar3m; pale-d=mmm; pass-e=4mm
  10.0000  10.0000  10.0000       0.        .0       0.
             0.  1  0  0              0.  0  1  0              0.  0  0  1
                 0  0  1                 1  0  0                 0  1  0
                 0  1  0                 0  0  1                 1  0  0
                 1  0  0                 0  0  1                 0  1  0
                 0  1  0                 1  0  0                 0  0  1
                 0  0  1                 0  1  0                 1  0  0
             0.  1  0  0              0.  0 -1  0              0.  0  0 -1
                 0  0 -1                 1  0  0                 0 -1  0
                 0 -1  0                 0  0 -1                 1  0  0
                 1  0  0                 0  0 -1                 0 -1  0
                 0 -1  0                 1  0  0                 0  0 -1
                 0  0 -1                 0 -1  0                 1  0  0
             0. -1  0  0              0.  0  1  0              0.  0  0 -1
                 0  0 -1                -1  0  0                 0  1  0
                 0  1  0                 0  0 -1                -1  0  0
                -1  0  0                 0  0 -1                 0  1  0
                 0  1  0                -1  0  0                 0  0 -1
                 0  0 -1                 0  1  0                -1  0  0
             0. -1  0  0              0.  0 -1  0              0.  0  0  1
                 0  0  1                -1  0  0                 0 -1  0
                 0 -1  0                 0  0  1                -1  0  0
                -1  0  0                 0  0  1                 0 -1  0
                 0 -1  0                -1  0  0                 0  0  1
                 0  0  1                 0 -1  0                -1  0  0
             0. -1  0  0              0.  0 -1  0              0.  0  0 -1
                 0  0 -1                -1  0  0                 0 -1  0
                 0 -1  0                 0  0 -1                -1  0  0
```

```
                    -1  0  0                    0  0 -1                    0 -1  0
                     0 -1  0                   -1  0  0                    0  0 -1
                     0  0 -1                    0 -1  0                   -1  0  0
                0. -1  0  0               0.   0  1  0              0.   0  0  1
                     0  0  1                   -1  0  0                    0  1  0
                     0  1  0                    0  0  1                   -1  0  0
                    -1  0  0                    0  0  1                    0  1  0
                     0  1  0                   -1  0  0                    0  0  1
                     0  0  1                    0  1  0                   -1  0  0
                0.   1  0  0               0.   0 -1  0              0.   0  0  1
                     0  0  1                    1  0  0                    0 -1  0
                     0 -1  0                    0  0  1                    1  0  0
                     1  0  0                    0  0  1                    0 -1  0
                     0 -1  0                    1  0  0                    0  0  1
                     0  0  1                    0 -1  0                    1  0  0
                0.   1  0  0               0.   0  1  0              0.   0  0 -1
                     0  0 -1                    1  0  0                    0  1  0
                     0  1  0                    0  0 -1                    1  0  0
                     1  0  0                    0  0 -1                    0  1  0
                     0  1  0                    1  0  0                    0  0 -1
                     0  0 -1                    0  1  0                    1  0  0
1
  Na                                          .0         .0         .0
       .10                                                                    7
  Na                                          .0         .5         .5
       .10                                                                    7
  Na                                          .5         .0         .5
       .10                                                                    7
  Na                                          .5         .5         .0
       .10                                                                    7
b Cl                                          .5         .0         .0
       .15                                                                    7
b Cl                                          .5         .5         .5
       .15                                                                    7
b Cl                                          .0         .5         .0
       .15                                                                    7
b Cl                                          .0         .0         .5
       .15                                                                    7
c Pit                                        .25        .25        .25
       .05                                                                    7
c Pit                                        .25        .75        .75
       .05                                                                    7
c Pit                                        .75        .25        .75
       .05                                                                    7
c Pit                                        .75        .75        .25
       .05                                                                    7
  Pass                                       .25        .00        .00
       .18       .04    155501    555501                                     7
  Pass                                       .75        .50        .00
       .18       .04    155501    555501                                     7
  Pass                                       .75        .00        .50
       .18       .04    155501    555501                                     7
  Pass                                       .25        .50        .50
```

```
      .18       .04    155501    555501                          7
  Pale                              .250      .250      .000
      .03       .15    155501    155601                          7
  Pale                              .750      .750      .000
      .03       .15    155501    155601                          7
  Pale                              .750      .250      .500
      .03       .15    155501    155601                          7
  Pale                              .250      .750      .500
      .03       .15    155501    155601                          7
center                              .250      .250      .250
1     .02                                                       7
      201
      301       3.2       3.2        12       0.5
      404   2155501   2155501         1        20       .26       .26       .26
      501   2155501    155501    165501    155501    156501                  1
      502         2        20         1        20
      503         2       2.5
      604        0.        0.        0.      4.00
     1101
  2  1001
  2          1  8 13 16  4   2.0   3.6   .07
  2          9 12 17 20  2   2.0   2.6   .04
            13 16 17 20  1   2.0   2.6   .03
  1   716
                                     1        12
  1   712
                                    13        20
  2   812
  2          1  8 13 16  4   2.0   3.6   .07
  2          9 12 17 20  2   2.0   2.6   .04
            13 16 17 20  1   2.0   2.6   .03
     1102
      503         2     -2.5
      202     2.375
     1103
      202
       -1
```

REFERENCES

1. C. K. Johnson, *OR TEP: A FORTRAN Thermal-Ellipsoid Plot Program for Crystal Structure Illustrations*, ORNL-3794 (Rev.), Union Carbide Corp., Oak Ridge Natl. Lab., June 1965.

2. C. K. Johnson, *OR TEP-II: A FORTRAN Thermal-Ellipsoid Plot Program for Crystal Structure Illustrations*, ORNL-5138, Union Carbide Corp., Oak Ridge Natl. Lab., March 1976.

3. *Science Citation Index*, Institute for Scientific Information, Inc., Philadelphia, 1995.

4. T. Hahn, ed., *International Tables for Crystallography* **A**, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

5. E. B. Fleischer, "X-Ray Structure Determination of Cubane," *J. Am. Chem. Soc.* **86**, 3889-3890 (1964).

6. C. K. Johnson, "Generalized Treatments for Thermal Motion," pp. 132-160 in *Thermal Neutron Diffraction*, ed. B. T. M. Willis, Oxford University Press, London, 1970.

7. R. R. Holmes and R. M. Deiters, "Anisotropic Thermal Motion of Trigonal Bipyramidal Molecules from Spectroscopic Data. Pentacoordinated Molecules. XIII," *J. Chem. Phys.* **51**, 4043-4054 (1969).

8. J. Heading, *Matrix Theory for Physicists*, Longmans, Green and Co., London, 81-106, 1958.

9. C. E. Springer, *Geometry and Analysis of Projective Spaces,* Freeman and Co., San Francisco, 1964.

10. J. A. Todd, *Projective and Analytical Geometry*, Pitman Pub. Corp., New York, 1946.

11. G. A. Korn and T. M. Korn, *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill Book Company, New York, 1961.

12. J. T. Rule, "Stereoscopic Drawings," *J. Opt. Soc. Am.* **28**, 313-322 (1938).

13. B. G. Saunders, "Stereoscopic Drawing by Computer—Is it Orthoscopic?," *Appl. Opt.* **7**, 1499-1504 (1968).

14. S. S. Wilks, *Mathematical Statistics*, Wiley, New York, 1962.

15. H. Cramer, *Random Variables and Probability Distributions*, Cambridge University Press, London, 1962.

16. K. S. Miller, *Multidimensional Gaussian Distributions*, John Wiley & Sons, New York, 1964.

17. W. C. Hamilton, *Statistics in Physical Science*, Ronald Press, New York, 1964.

18. E. Lukacs and R. G. Laha, *Applications of Characteristic Functions,* Hafner Publishing Co., New York, 1964.

19. R. S. Burington and D. C. May, *Handbook of Probability and Statistics with Tables*, 2nd ed., McGraw-Hill, New York, 1970.

20. D. B. Owen, *Handbook of Statistical Tables*, Addison-Wesley, Reading, Mass., 1962.

21. J. Waser, "The Anisotropic Temperature Factor in Triclinic Coordinates," *Acta Cryst*. **8**, 731 (1955).

22. W. R. Busing and H. A. Levy, "Determination of the Principal Axes of the Anisotropic Temperature Factor," *Acta Cryst*. **11**, 450-451 (1958).

23. D. W. J. Cruickshank *et al.*, "Crystallographic Calculations on the Ferranti Pegasus and Mark I Computers," pp. 32-78 in *Computing Methods and the Phase Problem in X-Ray Crystal Analysis*, ed. R. Pepinsky, J. M. Robertson, and J. C. Speakman, Pergamon Press, New York, 1961.

24. C. K. Johnson, unpublished data.

25. A. Elliott and B. R. Malcolm, "Chain Arrangement and Sense of the α-Helix in Poly-L-Alanine Fibres," *Proc. Roy. Soc. London* **A249**, 30-41 (1959).

26. A. Zalkin, J. D. Forrester, D. H. Templeton, S. M. Williamson, and C. W. Koch, "Potassium Perxenate Nonahydrate," *J. Am. Chem. Soc.* **86**, 3569-3571 (1964).

APPENDIX A

ORTEP-III SUBPROGRAMS

FUNCTION ARCCOS(X)	Computes θ, the arc cosine of X in degrees; $0 \le \theta \le 180°$.
ATOM(QA,Z)	Finds the triclinic coordinates **Z** for the atom described by the atom designator code QA.
AXEQB(A1,X,B1,JJJ)	Solves the matrix equation **A1 X = B1** for **X**. The matrices **B1** and **X** are (3,JJJ) and **A1** is always (3,3). To invert **A1**, make **B1** an identity matrix.
AXES(U,V,X,ITYPE)	Provides three orthogonal column vectors in **X**, each 1 Å long, from the two vectors **U** and **V**. ITYPE > 0: Cartesian system ITYPE < 0: triclinic system |ITYPE| = 1: $\mathbf{X}_1 = \mathbf{U}$; $\mathbf{X}_2 = (\mathbf{U} \times \mathbf{V})$; $\mathbf{X}_3 = \mathbf{U} \times (\mathbf{U} \times \mathbf{V})$ |ITYPE| = 2: $\mathbf{X}_1 = \mathbf{U}$; $\mathbf{X}_2 = (\mathbf{U} \times \mathbf{V}) \times \mathbf{U}$; $\mathbf{X}_3 = \mathbf{U} \times \mathbf{V}$ ITYPE = 0: same as type 2 except **U** = *a* crystal axis, **V** = *b* crystal axis.
BOND(Z1,Z2,NB,NA1,NA2)	Draws a bond, described by Format No. 2 trailer card number NB, between two atoms. Z*n* is atom designator code of atom *n*, and NA*n* is number of atom *n* in ATOMS array.
COLR*xx*(ICOLOR)	Sets plot color on "device" *xx* to ICOLOR.
CURSSC	Identifies atoms selected on screen display.
DFLTS	Sets default values for items requested from user.
DIFV(X,Y,Z)	Performs the vector subtraction **X** – **Y** = **Z**. **Z** may have the same location as **X** or **Y**.
DRAW(W,DX,DY,NPEN)	Interconnects ORTEP and the plot package. It also prevents the pen from crossing the boundaries. If the indicator ITILT in COMMON is zero, the array W contains *x* and *y* in plotter coordinates. While perspective lettering is being plotted, ITILT ≠ 0; and W contains *x*, *y*, *z* in Cartesian coordinates, which will be rotated and projected by DRAW to form plotter x,y coordinates. DX and DY are added to the plotter *x* and *y*, respectively, before the plot package is called. NPEN = 2 for pen down and 3 for pen up.

EDITR	Controls ORTEP line editor.
EIGEN(W,VALU,VECT)	Determines the three eigenvalues **VALU** and the three column eigenvectors **VECT** of the matrix **W**. Indeterminate eigenvectors are replaced by zeros and the fault indicator NG set to a negative value (eigenvectors are assigned for the indeterminate cases by PRELIM).
END*xx*	Terminates plotting on "device" *xx*.
ERPNT(TD,N)	Prints error message when a fault is found. The arguments identify the atom designator code TD and the instruction N involved in the fault. The fault indicator, NG, is in COMMON.
EXITNG(ING)	Prints fault indicator ING if abnormal termination and stops program execution.
F200	Executes the 200 series instructions.
F400	Executes the 400 series instructions.
F500	Executes all 500 series instructions.
F600	Executes all 600 series instructions.
F700	Executes all 700 series instructions.
F800	Executes all 800 series instructions. Bonds to be drawn are found by F800, then drawn by subroutine BOND.
F900	Executes all 900 series instructions.
F1000	Executes the 1001 instruction.
FUNCTION IEND(STRING)	Returns the position of the last non-space character in a character STRING.
INIT*xx*	Initializes plotting on "device" *xx*.
LAP500(NTYPE)	Sorts the ATOMS array, then calculates the projected outline ellipses for all atoms in the ATOMS array. The ellipses are stored in the CONIC array along with the minima and maxima in x and y for a rectangle enclosing each ellipse. If NTYPE < 0, previous overlap information is cleared.

LAP700(NA,ICQ)	Finds the atoms that overlap a given atom to be drawn. The routine first checks the bounding rectangles for intersections, then forms the cubic discriminant from the quadratic descriptions of the two projection ellipses. The discriminant provides a specification for complete overlap, partial overlap, or no overlap. A list of up to 20 interfering ellipses is compiled. NA is the atom to be drawn. ICQ is set > 0 if overlap exists and = -1 if not.
LAP800(NA1,NA2,ICQ)	Used in the "Projected Outline Storage Step" to store the projected quadrangles for the bonds specified by the trailer cards of the 1001, 821, and 822 instructions. The routine also is used in the "Area-Overlap Search Step" to find the projected bond quadrangles that overlap a given bond to be drawn. A list containing up to 30 interfering quadrangles is compiled. NA1 and NA2 are the two atoms of the bond. ICQ is set > 0 if overlap exists and = -1 if not.
LAPAB(IQ,IA,ICQ,ITY)	Finds the bonds that overlap an atom to be drawn and the atoms that overlap a bond to be drawn. It is used in the "Area-Overlap Search Step." ITY > 0 checks for atom, IA, over bond, IQ, and ITY < 0 checks for bond over atom. ICQ is set > 0 if overlap exists, = 0 for no overlap, and < 0 for hidden atom or bond.
LAPCON(CON1,CON,Y,OVMR)	Transforms conic, CON1, to plotter homogeneous coordinate system, CON, with center at Y. OVMR denotes overlap margin.
LAPDRW(Y,NPEN,NCQ)	Checks each line segment to be drawn for intersection with the interfering ellipses and quadrangles and compiles a list of intersections. The intersection list is sorted according to distance along the line segment, and the intersection pattern is analyzed to determine which subsegments are visible and which are hidden. The line subsegments are passed to the SCRIBE routine. Y is pen position, NPEN denotes if pen is up or down, and NCQ is set to NCOVER+NQOVER.
CHARACTER*(*) FUNCTION MAKSYM(GP)	Returns a character string representation (xyz notation) of a symmetry operator stored in ORTEP's internal representation in array GP.
MM(X,Y,Z)	Performs the matrix multiplication **XY** = **Z**. The location of **Z** must be different from **X** and **Y**.

MV(X,Y,Z)	Performs the matrix-vector multiplication **XY** = **Z**. The location of **Z** must be different from **X** and **Y**.
NORM(X,Y,Z,ITYPE)	Stores at **Z** a vector (not necessarily a unit vector) perpendicular to both **X** and **Y**. The sense of **Z** is that of the vector product **X** × **Y**. ITYPE > 0: Cartesian system ITYPE ≤ 0: triclinic system
NUMBUR(W,W2,HGT,DIST,THT,ND)	Converts number to character string for placement on the drawing. W contains coordinates of the lower left edge of the first character, W2 is unused, HGT is the height of the characters, DIST is the number to be drawn, THT is the angle by which the base line of the characters is to be rotated counterclockwise from the positive *x* axis, and ND is the number of digits to the right of the decimal point.
ORTEP	ORTEP is the MAIN program and controlling routine that decodes the ORTEP instructions. It either executes the command directly or calls the appropriate subroutine to execute the instruction.
PAXES(DCODE,ITYPE)	Stores the covariance (dispersion) matrix for the thermal ellipsoid or its inverse matrix, which is the matrix of coefficients in the quadratic form describing the ellipsoid, in COMMON at Q for the atom with atom designator code DCODE. ITYPE > 0 for covariance matrix ITYPE < 0 for ellipsoid quadratic form matrix |ITYPE| = 1 based on triclinic system |ITYPE| = 2 based on working Cartesian system |ITYPE| = 3 based on reference Cartesian system
PEN*xx*(X,Y,IPEN)	Controls pen movement on "device" *xx*. X is the abscissa and Y is the ordinate expressed in inches. IPEN=2: pen draws line as it moves IPEN=3: pen moves without drawing line
PENW*xx*(PENW)	Sets pen thickness on "device" *xx* to PENW. PENW is provided in thousandths of an inch. The default is 5.
PLOT(X,Y,IPEN)	Calls the appropriate PEN*xx* routine for drawing lines on "device" *xx*. Parameters are sent to PEN*xx*.
PLTXY(X,Y)	Calculates the plotter coordinates Y from the unscaled Cartesian coordinates X. The distance to the closest boundary of the plot is stored in the variable EDGE in COMMON.

PRELIM	Performs all calculations to process (e.g., principal axis transformations) and store the input crystallographic parameters.
PRIME	"Primes the program" by initializing all the "primer parameters".
PROJ(D,DP,X,XO,VIEW,I1,I2,I3)	Used to obtain an array, DP, of plotter coordinates from a scaled array, D, of points described in Cartesian coordinates. X, XO, and VIEW are parameters involved in the projection, and I1, I2, I3 are DO loop parameters for indexing through the array.
RADIAL(ND)	Generates a "radial" array (D in COMMON) of points lying on an ellipse, given two conjugate radius vectors of the ellipse in the array DA in COMMON. From 8 to 128 points are generated depending on the value of ND ($1 \leq ND \leq 5$).
READIN(IU,CHEM,ID1,ID2,X1, X2,X3,IT,IS,B1,B2,B3,B4,B5, B6,BTYPE)	Reads atom parameters in any format from a file. This subroutine may be modified by the user. See Section 4.5 for a description of the parameters.
RECYCLE	Returns instruction pointer to 201 instruction and zeroes ATOMS array.
SCRIBE(Y,NPEN)	Filters out the hidden line segments and passes the visible line segments to the DRAW routine.
SEARC	Conducts an exhaustive (but educated) search to find all points within a sphere or rectangular box. Interatomic distances and angles are also calculated for the 100 series.
SIMBOL(W,W2,HGT,ITXT,THT,N)	Processes character strings for placement on the drawing. W contains coordinates of the lower left edge of the first character, W2 is unused, HGT is the height of the characters, ITXT is the string to be drawn, THT is the angle by which the base line of the characters is to be rotated counterclockwise from the positive x axis, and N is the number of characters to be drawn.
SPARE(INST)	Expands the user supplied instruction set by responding to any INST $\geq$ 12. INST = instruction /100.
STOR(TD1)	Stores atom with atom designator code TD1 in (or removes atom from) the ATOMS array. Coordinates in whichever system is in use are communicated to STOR via array V1 of COMMON.

TEPSYM(TXT,NUM,KK)	Parses symmetry operator in character string representation (xyz notation), TXT, and stores the information in ORTEP's internal representation. NUM is symmetry operator number, and KK is the component number.
TMM(X,Y,Z)	Performs the matrix multiplication $(\mathbf{X}^T \mathbf{Y})^T = \mathbf{Z}$. The location of **Z** must be different from **X** and **Y**.
UINPUT(IN,NOUT)	Controls user input. IN is input file device number, and NOUT is output file device number.
UNITY(X,Z,ITYPE)	Makes the vector **Z** 1 Å long and parallel to **X**. The vectors **X** and **Z** may have the same location. ITYPE > 0: Cartesian system ITYPE < 0: triclinic system
VM(Y,X,Z)	Performs the vector-matrix multiplication $\mathbf{Y}^T\mathbf{X} = \mathbf{Z}^T$. The location of **Z** must be different from **Y** and **X**.
FUNCTION VMV(X1,Q,X2)	Performs the vector-matrix-vector multiplication $\mathbf{X1}^T \mathbf{Q}\ \mathbf{X2}$ = scalar.
FUNCTION VV(X,Y)	Performs the vector-vector multiplication $\mathbf{X}^T\mathbf{Y}$ = scalar.
XYZ(DQA,X,ITYPE)	Returns in X coordinates for atom with atom designator code DQA. ITYPE = 0: triclinic coordinates ITYPE = 1 or 2: working Cartesian system coordinates ITYPE = 3: reference Cartesian system coordinates

APPENDIX B

GLOSSARY OF VARIABLES IN ORTEP-III COMMONS

*	UNNAMED	Main Common Block
	A(9)	Direct crystal cell parameters, a, b, c, cos α, cos β, cos γ, α, β, γ.
	AA(3,3)	Metric tensor **g** where $g_{ij} = a_i \cdot a_j$.
	AAREV(3,3)	Postfactor transformation matrix to convert coordinates from triclinic to the reference Cartesian system. **AAREV = AA REFV.**
	AAWRK(3,3)	Postfactor transformation matrix to convert coordinates from triclinic to the working Cartesian system. **AAWRK = AA WRKV.**
	AID(3,3)	Identity matrix.
	REAL*8 AIN(140)	Array containing the input parameters of the current ORTEP instruction.
	REAL*8 ATOMID(500)	Atom designator codes of atoms in ATOMS array.
	ATOMS(3,500)	Temporary storage of atom coordinates in any of several coordinate systems.
	BB(3,3)	Reciprocal metric tensor. $\mathbf{BB} = \mathbf{AA}^{-1}$.
P	BRDR	Border (margin) width in inches extending inward from plot boundary.
	CD(8,20)	Holds the real values entered on a Format No. 2 trailer card. Used in conjunction with KD array.
P	CONT(5)	Constants used in subroutine RADIAL.
	D(3,130)	Array in which three-dimensional points on an ellipse are stored by RADIAL.
	DA(3,3)	Transmits conjugate vectors to RADIAL. Also used for temporary storage.
P	DISP	Displacement parameter for retracing.
	DP(2,130)	Array in which two-dimensional points for ellipse are stored after projection.
	EDGE	Distance in inches from a projected point to the closest boundary. Set in PLTXY.

P	FORE	Cosine of critical angle between bond and Cartesian z axis vectors for perspective bond distance labels. At smaller angles, the labels, produced from subroutine BOND, are drawn without perspective to prevent excessive foreshortening.
	FS(3,3,96)	Rotation matrices for input symmetry operators based on triclinic system. Used with TS array.
	IN	Logical unit number of input file.
P	ITILT	Indicator used to signal subroutine DRAW, whether or not to do perspective labeling.
	KD(5,20)	Holds the integer values entered on a Format No. 2 trailer card. Used in conjunction with CD.
P	LATM	Number of entries in ATOMS array.
	NATOM	Number of input atoms.
P	NCD	Number of Format No. 2 trailer cards for an instruction.
P	NG	Fault Indicator value.
	NJ	Instruction number/100.
	NJ2	Last two decimal digits of the instruction number (instruction = NJ × 100 + NJ2).
	NOUT	Logical unit number of text output file.
	NSR	Logical unit number of scratch file.
	NSYM	Number of input symmetry operators.
P	ORGN(3)	Triclinic coordinates for the atom that is the origin of the drawing (i.e., on the optic axis for the projection).
	PAC(3,5)	A 3 × 3 matrix produced by subroutine PAXES and made up of three orthonormal principal axis column vectors, based on either the working or reference Cartesian system. Columns 4 and 5 are used in subroutine F700 to duplicate columns 1 and 2 for ease in indexing.
	PAT(3,3)	A matrix produced by subroutine PAXES and composed of three principal axis column vectors each 1 Å long, based on the triclinic system.
	Q(3,3)	A matrix produced by subroutine PAXES. Contains either the dispersion matrix or its inverse, based on either the working or reference Cartesian systems.

	REFV(3,3)	A matrix made up of three orthogonal column vectors, each 1 Å long, based on the triclinic system. This is the base vector triplet for the reference Cartesian coordinate system. The transpose is the postfactor transformation matrix for converting coordinates from the reference orthogonal system to the triclinic system. $\mathbf{REFV}^T = \mathbf{AAREV}^{-1}$.
P	RES(4)	Regulates the resolution of the plotting of a given ellipse as a function of the longest principal axis x in the given ellipsoid of the scaled model. $x \geq$ RES(1) 128-point ellipse RES(1) $> x \geq$ RES(2) 64-point ellipse RES(2) $> x \geq$ RES(3) 32-point ellipse RES(3) $> x$ 16-point ellipse RES(4) not used
	RMS(5)	The rms displacements along the principal axes in arrays PAC and PAT.
P	SCAL1	The scale of the model in inches per Angstrom before projection.
P	SCAL2	The scale factor ratio that sets the ellipsoid scale relative to SCAL1.
P	SCL	SCL = SCAL1 × SCAL2.
P	SYMB(3,3)	A rotation matrix based on the angle THETA, which is set by instruction 302.
P	TAPER	The exaggerated bond taper parameter. The top and bottom ends of a bond have radii: RADIUS = 1. ± TAPER × T6 where T6 = lcosine of angle between bond and z axis of Cartesian systeml.
P	THETA	Angle in degrees between plot x axis and lettering base-line vector.
	CHARACTER*4 TITLE(18)	Alphanumeric job title storage.
	CHARACTER*4 TITLE2(18)	Alphanumeric information storage for Format No. 3 trailer card.
	TS(3,96)	Translation vector for each input symmetry operator. Used with FS array.
P	VIEW	Viewing distance in inches.
	VT(3,4)	Perspective title rotation matrix and translation vector. Also used for temporary storage.
	V1(4)	Array to transfer data to subroutine STORE. Also used for temporary storage.

	V2(3),V3(3),V4(3), V5(3),V6(3)	Temporary storage.
	WRKV(3,3)	Same definition as for REFV except that this one is for working Cartesian system. $\mathbf{WRKV}^{T} = \mathbf{AAWRK}^{-1}$.
P	XLNG(3)	Elements 1 and 2 are *x* and *y* plot dimensions. Element 3 is not used.
P	XO(3)	Elements 1 and 2 denote the position in plotter coordinates (in inches) where ORGN is placed. Element 3 is used to transfer *z* coordinates to subroutine DRAW when perspective lettering is used.
	XT(3)	Triclinic coordinates for an atom position are placed here by subroutine XYZ.
	DFL	**Default Values for User Input**
	CHARACTER*60 ATOMFI	Default name of file containing atom parameters used by subroutine READIN.
	CHARACTER*4 EXT	Default filename extension for ORTEP output.
	FPAPLEN	Default page length for drawing
	IDRAW	Default drawing destination indicator.
	CHARACTER*60 INFILE	Default input file name.
	IORIENT	Default orientation of drawing
	IOUT	Default ORTEP text output logical unit number.
	NS	**Output Drawing Parameters**
	NDRAW	ORTEP drawing destination indicator. NDRAW=0: none NDRAW=1: screen NDRAW=2: Postscript file NDRAW=3: HPGL file NDRAW=9: Reserved for future use
	NORIENT	Orientation of drawing.
	NPF	Logical unit number of drawing output file.
	NVAR	Temporary storage.
	OLAP	**Overlap Correction Variables**
	CONIC(7,500)	Overlap correction ellipses describing intersection of enveloping cones with drawing plane.

	COVER(6,20)	Stores up to 20 overlapping ellipses for an atom or bond being drawn.
	KC(20)	Which ellipses overlap the atom or bond being drawn.
	KQ(30)	Which quadrangles overlap the atom or bond being drawn.
	NCONIC	Total number of projected ellipses stored for overlap calculations.
	NCOVER	Number of projected ellipses over an atom or bond to be drawn.
	NQOVER	Number of quadrangles over an atom or bond to be drawn.
	NQUAD	Total number or projected bond quadrangles for overlap calculations.
	OVMRGN	Overlapping element margin.
	QOVER(3,4,30)	Stores up to 30 overlapping bond quadrangles for atom or bond being drawn.
	QUAD(9,600)	Overlap correction bond quadrangles projected onto drawing plane.
	SEGM(50,2)	Visible segments of an ellipsoid or bond element to be drawn.
	PARMS	**Input Atom Parameters**
	CHARACTER*8 CHEM(505)	Names for input atoms.
	EV(3,505)	Root-mean-square displacements for each principal axis of each input atom.
	INTEGER*2 IDENT(2,505)	Two feature identifiers for each input atom.
P	MAXATM	Array size for input atoms. (Currently 505.)
	P(3,505)	Triclinic positional coordinates for the input atoms.
	PA(3,3,505)	Matrices for each input atom made up of three orthogonal column eigenvectors each 1 Å long, based on the triclinic system (principal axis vectors).
	PS	**Encapsulated Postscript Output Parameters**
	IXMIN	Minimum x value of illustration.
	IXMAX	Maximum x value of illustration.

	IYMIN	Minimum y value of illustration.
	IYMAX	Maximum y value of illustration.
	IXT	Page translation along x.
	IYT	Page translation along y.
	QUEUE	**Editor Variables**
	CHARACTER*73 HQUE(96)	Original instruction set as read from input file.
	CHARACTER*73 INQ	Next instruction held in memory to be processed.
	NBACK	Number of lines in original instruction set as read from input file.
	NED	Logical unit number of temporary file used by editor.
	NEXT	Line number of next instruction held in memory to be processed.
	NQUE	Current number of instruction lines held in memory.
	CHARACTER*73 QUE(96)	Instruction lines held in memory.
	TRFAC	**Plot Translation Factors**
	XTRANS	Shift of plot origin along x-axis.
	YTRANS	Shift of plot origin along y-axis.

*Letter "P" indicates Prime Parameter (i.e., initialized in subroutine PRIME).

APPENDIX C

ORTEP-III FORTRAN SOURCE CODE LISTING

```
c ***************************************************************
c
c     ORTEP-III: Oak Ridge Thermal Ellipsoid Plot Program
c     Carroll K. Johnson and Michael N. Burnett
c     Oak Ridge National Laboratory
c     Version 1.0 April 1, 1996
c
c     Send comments, questions, problems, suggestions, etc. to
c     ortep@ornl.gov
c
c ***************************************************************

      PROGRAM ORTEP
      REAL*8 TD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      CHARACTER*73 INQ,QUE,hque
      common /ns/ npf,ndraw,norient,nvar
      logical tmpopn

c *** Drawing Output Options
c *** ndraw=0: no drawing output
c *** ndraw=1: screen output
c *** ndraw=2: Postscript file output
c *** ndraw=3: HPGL file output

c *** Logical Unit Numbers ***
c *** 15,16 are used in subroutine EDITR
c *** 18 (variable iu) is used in subroutine PRELIM
c *** NOUT is set in subroutine UINPUT
      IN=3
      NED=7
      NSR=8
      NPF=10

      call uinput(in,nout)

    2 CALL PRIME

c *** open ORTEP scratch file ***
c *** if already open, close it first ***
      inquire(NSR,opened=tmpopn)
      if (tmpopn) close(NSR)
      open(NSR,status='scratch',form='unformatted')

c *** open a temporary file - needed by the editor ***
c *** if already open, close it first ***
      inquire(NED,opened=tmpopn)
      if (tmpopn) close(NED)
      open(NED,status='scratch')

C     ***** READ JOB TITLE CARD *****
      READ (IN,4)(TITLE(I),I=1,18)
    4 FORMAT(18A4)
      write (NED,4)(TITLE(I),I=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,6)(TITLE(I),I=1,18)
    5 FORMAT(1H0,10X,18A4)
    6 FORMAT(1H1,10X,18A4)
      CALL PRELIM
      IF (NOUT.GE.0)
     &WRITE (NOUT,6)(TITLE(I),I=1,18)
C     ***** LOAD INSTRUCTION QUE *****
      NQUE=0
 2005 NQUE=NQUE+1
 2010 READ (IN,2012,END=2015,ERR=3000) QUE(NQUE)
      if (que(nque)(1:1).eq.'#') go to 2010
      hque(nque)=que(nque)
      if (que(nque)(4:9).eq.'    -2') go to 2020
 2012 FORMAT(A72)
      IF(NQUE.LT.96) GO TO 2005
      GO TO 2020
 2015 NQUE=NQUE-1
C     ***** REPOSITION TO POINT BEFORE EOF *****
      BACKSPACE IN
 2020 NBACK=NQUE
      NEXT=1
      ISAVE=0
      GO TO 507
    7 ISAVE=0
C     ***** ZERO AIN ARRAY *****
    8 DO 10 J=1,140
   10 AIN(J)=0.
   11 FORMAT(1H0,4X,17H((((( INSTRUCTION,I5,6H ))))))
   12 FORMAT(I3,I6,7F9.0)
   13 FORMAT(I3,I6,7D15.8)
   14 FORMAT(1H ,9X,7D15.7)
C     ***** READ NEW INSTRUCTION CARD *****
      NCD=0
      N1=-6
   16 N1=N1+7
      N2=N1+6
      IF(ISAVE)22,18,18
   18 INQ=QUE(NEXT)
      NEXT=NEXT+1
      READ (INQ,12)IIC,NF,(AIN(I),I=N1,N2)
      IF(ISAVE)24,24,20
   20 WRITE (NSR)IIC,NF,(AIN(I),I=N1,N2)
      GO TO 24
   22 READ (NSR)IIC,NF,(AIN(I),I=N1,N2)
      IF(IIC)7,24,24
   24 IF(N1-1)26,26,30
   26 IF (NOUT.GE.0)
     &WRITE (NOUT,11)NF
      NF1=NF
      IF(NF1)28,8,30
c *** run editor?
   28 call go2edtr
      if (next.lt.nque) go to 8
```

```
      IF(NF1+2)2,2,3000
   30 CONTINUE
CCC   IF (NOUT.GE.0)
CCC  &WRITE (NOUT,14)(AIN(I),I=N1,N2)
   32 IIC=IIC+1
      GO TO (90,16,38,50),IIC
   33 FORMAT(I3,6X,5I3,8F6.0)
   34 FORMAT(6I3,8E12.5)
   35 FORMAT(1H ,11X,5I3,8F11.5)
C     ***** READ FORMAT 2 TRAILER CARDS *****
   38 NCD=NCD+1
      IF(ISAVE)44,40,40
   40 INQ=QUE(NEXT)
      NEXT=NEXT+1
      READ (INQ,33)IIC,(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
      IF(ISAVE)46,46,42
   42 WRITE (NSR)IIC,(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
      GO TO 46
   44 READ (NSR)IIC,(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
   46 IF (NOUT.GE.0)
     &WRITE (NOUT,35)(KD(I,NCD),I=1,5),(CD(I,NCD),I=1,8)
      GO TO 32
C     ***** READ FORMAT 3 TRAILER CARD *****
   50 IF(ISAVE)52,54,54
   52 READ (NSR)(TITLE2(I),I=1,18)
      GO TO 55
   54 INQ=QUE(NEXT)
      NEXT=NEXT+1
      READ (INQ,4)(TITLE2(I),I=1,18)
   55 IF (NOUT.GE.0)
     &WRITE (NOUT,5)(TITLE2(I),I=1,18)
      IF(ISAVE)90,90,56
   56 WRITE (NSR)(TITLE2(I),I=1,18)
C     ***** EXECUTE INSTRUCTION *****
   90 NJ=NF1/100
      NJ2=NF1-NJ*100
      NJ3=MOD(NJ2,10)
      IF(NJ-12)98,92,92
   92 CALL SPARE(NF1)
      IF(NG)94,8,94
   94 CALL ERPNT(0.D0,NF1)
      GO TO 8
C     ******BRANCH TABLE FOR FUNCTION TYPES******
   98 GO TO(100,200,300,400,500,600,700,800,900,1000,1100),NJ
C     *******100 INSTRUCTIONS-STRUCTURE ANALYSIS FUNCTIONS*******
  100 GO TO (101,101,104,104,101,101,94),NJ2
  101 CALL SEARC
      GO TO 8
C     ***** ANISOTROPIC TEMP FACTOR OUTPUT *****
  104 DO 164 I=1,NATOM
      IF(MOD(I,14)-1)134,114,134
  114 IF (NOUT.GE.0)
     &WRITE (NOUT,6)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,129)
  129 FORMAT(1H010X,4HATOM3X,16HRMS DISPLACEMENT3X,31HROW VECTORS, BASED
     1 ON REFERENCE17X,29HPROBABILITY COVARIANCE MATRIX)
  134 TD=55501.+FLOAT(I)*100000.
      CALL PAXES(TD,-3)
      IF(NG)144,154,144
  144 CALL ERPNT(TD,104)
  149 FORMAT(1H0,10X,A6,F10.6,6X,3F12.7,10X,3F12.7)
  154 IF (NOUT.GE.0)
     &WRITE (NOUT,149)CHEM(I),RMS(1),(PAC(J,1),J=1,3),(Q(J,1)
     1,J=1,3)
  164 IF (NOUT.GE.0)
     &WRITE (NOUT,159)(RMS(K),(PAC(J,K),J=1,3),(Q(J,K),J=1,3)
     1,K=2,3)
  159 FORMAT(1H ,16X,F10.6,6X,3F12.7,10X,3F12.7)
      GO TO 8
C     *******200 INSTRUCTIONS-PLOTTER CONTROL FUNCTIONS*******
  200 CALL F200
      GO TO 8
C     *******300 INSTRUCTIONS-DRAWING CONTROL FUNCTIONS*******
  300 GO TO (301,302,303,304,94),NJ2
C     *******PLOT DIMENSIONS*******
  301 IF(AIN(1))321,321,311
  311 XLNG(1)=AIN(1)
  321 IF(AIN(2))341,341,331
  331 XLNG(2)=AIN(2)
  341 IF(AIN(3))361,351,351
  351 VIEW=AIN(3)
  361 IF(AIN(4))381,381,371
  371 BRDR=AIN(4)
  381 IF (NOUT.GE.0)
     &WRITE (NOUT,389)XLNG(1),XLNG(2),BRDR
  389 FORMAT(1H010X,11HPLOT LIMITSF6.2,3H BYF6.2,15H IN.  INCLUDINGF6.2,
     112H  IN. MARGIN)
  391 IF (NOUT.GE.0)
     &WRITE (NOUT,399)VIEW
  399 FORMAT(1H ,10X,13HVIEW DISTANCE,F7.3,7H INCHES)
      GO TO 8
C     *******LEGEND ROTATION*******
  302 THETA=AIN(1)
      T1=THETA*.01745329252
      COSTH=COS(T1)
      SINTH=SIN(T1)
      DO 312 J=1,9
  312 SYMB(J,1)=0.
      SYMB(1,1)=COSTH
      SYMB(2,2)=COSTH
      SYMB(3,3)=1.
      SYMB(2,1)=SINTH
      SYMB(1,2)=-SINTH
      IF (NOUT.GE.0)
     &WRITE (NOUT,319)THETA
  319 FORMAT(1H010X,44HREGULAR TITLE AND SYMBOL ROTATION IN DEGREESF8.2)
      GO TO 8
C     ***** RETRACE DISPLACEMENT *****
  303 DISP=AIN(1)
      IF (NOUT.GE.0)
     &WRITE (NOUT,313)DISP
  313 FORMAT(1H0,10X,22HRETRACE DISPLACEMENT =,F7.4,5H INCH)
      GO TO 8
C     ***** change resolution (smoothness) of ellipses *****
  304 res(1)=AIN(1)*.75
      res(2)=.5*res(1)
      res(3)=.25*res(2)
      GO TO 8
C     *******400 INSTRUCTIONS-ATOM LIST FUNCTIONS*******
  400 GO TO (401,401,401,401,401,401,401,490, 94,410,
     1       401,401,401,401,401,401,401, 94),NJ2
```

```
  401 CALL F400
      GO TO 490
  410 LATM=0
      DO 420 I=1,500
      ATOMID(I)=0.
      DO 420 J=1,3
  420 ATOMS(J,I)=0.
  490 IF(LATM)8,8,491
  491 IF (NOUT.GE.0)
     &WRITE (NOUT,499)(ATOMID(I),I=1,LATM)
  499 FORMAT(1H0,10X,23HCONTENTS OF ATOMS ARRAY/(15X,10F10.0))
      GO TO 8
C     *******500 INSTRUCTIONS-CARTESIAN COORDINATE SYSTEM FUNCTIONS*******
  500 CALL F500
      IF (NOUT.GE.0)
     &WRITE  (NOUT,503)(ORGN(J),J=1,3)
  503 FORMAT(1H0,10X,    44HORIGIN FOR PROJECTION AXIS IN CRYSTAL COORD.
     1,3F15.6)
      IF(NJ3-3)507,539,504
  504 IF(NJ3-6)601,507,601
  507 IF (NOUT.GE.0)
     &WRITE (NOUT,529)
      IF (NOUT.GE.0)
     &WRITE (NOUT,519)((REFV(J,I),I=1,3),(AAREV(J,I),I=1,3),J
     1=1,3)
      GO TO 8
  509 FORMAT(1H010X, 49HORTHONORMAL WORKING VECTORS BASED ON CRYSTAL AXE
     1S18X,33HPOST-FACTOR TRANSFORMATION MATRIX/16X,8HX VECTOR8X,8HY VEC
     2TOR8X,8HZ VECTOR)
  519 FORMAT(1H ,10X,3E16.7,8X,3E16.7)
  529 FORMAT(1H010X,51HORTHONORMAL REFERENCE VECTORS BASED ON CRYSTAL AX
     1ES16X,33HPOST-FACTOR TRANSFORMATION MATRIX/16X,8HX VECTOR8X,8HY VE
     2CTOR8X,8HZ VECTOR)
  539 IF (NOUT.GE.0)
     &WRITE (NOUT,509)
      IF (NOUT.GE.0)
     &WRITE (NOUT,519)((WRKV(J,I),I=1,3),(AAWRK(J,I),I=1,3),J
     1=1,3)
      GO TO 8
C     *******600 INSTRUCTIONS-PLOT CENTERING FUNCTIONS*******
  600 CALL F600
  601 IF (NOUT.GE.0)
     &WRITE (NOUT,609)XO(1),XO(2),SCAL1,SCAL2
  609 FORMAT(1H010X,31HORIGIN POINT IN PLOTTER COORD.(F6.2,2H ,F6.2,8H )
     1 IN.  / 11X,15HOVERALL SCALE =F6.3,32H INCH/ANGSTROM ELLIPSOID SCA
     2LE =F6.3)
      GO TO 391
C     *******700 INSTRUCTIONS-ELLIPSOID AND SYMBOL PLOT FUNCTIONS*******
C     ********FILL OUT DETAILS FOR SPECIAL MODELS********
  700 GO TO (701,702,704,705,709,7006,94),NJ3
 7006 AIN(3)=1.
      GO TO 703
  701 AIN(3)=8.
      GO TO 703
  702 AIN(3)=0.
  703 AIN(1)=4.
      AIN(2)=0.
      AIN(4)=0.
      GO TO 709
  704 AIN(1)=3.
      AIN(2)=-5.
      GO TO 706
  705 AIN(1)=1.
      AIN(2)=0.
  706 AIN(3)=1.
      AIN(4)=5.
  709 CALL F700
      GO TO 8
C     *******800 INSTRUCTIONS-BOND FUNCTIONS*******
  800 CALL F800
      GO TO 8
C     *******900 INSTRUCTIONS-TITLE FUNCTIONS*******
  900 CALL F900
      GO TO 8
C     *******1000 INSTRUCTIONS-OVERLAP FUNCTIONS*******
 1000 CALL F1000
      GO TO 8
C     *******1100 INSTRUCTIONS-SAVE SEQUENCE FUNCTIONS*******
 1100 IF(NJ2-2)1101,1102,1103
 1101 ISAVE=1
      GO TO 1104
 1102 ISAVE=0
      J=-1
CCC   END FILE NSR
      GO TO 1104
 1103 ISAVE=-1
 1104 REWIND NSR
      GO TO 8
 3000 CALL EXITNG(NG)
      END
      FUNCTION ARCCOS(X)
C     ARCCOS(X) IN DEGREES
      IF(1.0-ABS(X))1,2,2
    1 X=SIGN(1.0,X)
    2 IF(X)3,4,5
    3 ARCCOS=180.0+ATAN(SQRT(1.0-X*X)/X)*57.29577951
      GO TO 6
    4 ARCCOS=90.0
      GO TO 6
    5 ARCCOS=ATAN (SQRT(1.0-X*X)/X)*57.29577951
    6 RETURN
      END
      SUBROUTINE ATOM(QA,Z)
C     ATOM COORDINATE SUBROUTINE
      REAL*8 QA,TA,D100K
      DIMENSION X(3),Z(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      D100K=100000.0
      K=QA/D100K
```

```
      IF(K)109,109,117
  109 X(1)=0.0
      X(2)=0.0
      X(3)=0.0
      GO TO 125
  117 IF(K-NATOM)119,119,503
  503 NG=5
      GO TO 325
  119 DO 123 J=1,3
  123 X(J)=P(J,K)
  125 TA=DABS(QA)
      KSYM=DMOD(TA,D100K)
      KT=KSYM/100
      KS=KSYM-100*KT
      IF(KS-NSYM)203,203,403
  403 NG=4
      GO TO 325
  203 IF(KS)403,205,213
  205 Z(1)=X(1)
      Z(2)=X(2)
      Z(3)=X(3)
      GO TO 311
  213 DO 223 K=1,3
      Z(K)=TS(K,KS)
      DO 223 J=1,3
  223 Z(K)=Z(K)+FS(J,K,KS)*X(J)
  311 IF(KT)403,325,313
  313 IF(KT-555)317,315,317
  315 KSYM=KS
      GO TO 325
  317 K1=KT/100
      K=KT-100*K1
      K2=K/10
      K3=K-10*K2
      Z(1)=Z(1)+FLOAT(K1-5)
      Z(2)=Z(2)+FLOAT(K2-5)
      Z(3)=Z(3)+FLOAT(K3-5)
  325 RETURN
      END
      SUBROUTINE AXEQB(A1,X,B1,JJJ)
C     ***** SOLUTION OF MATRIX EQUATION AX=B FOR X *****
C     ***** USES METHOD OF TRIANGULAR ELIMINATION *****
C     ***** B AND X HAVE DIMENSIONS (3,JJJ),A IS ALWAYS (3,3)
C     ***** TO INVERT A MAKE B 3 BY 3 IDENITY MATRIX *****
      DIMENSION A1(3,3),A(3,3),B(3,3),B1(3,3),X(3,3)
      NV=JJJ
C     ***** TRANSFER DATA *****
      DO 2 I=1,3
      DO 2 J=1,3
      IF(NV-J)2,1,1
    1 B(I,J)=B1(I,J)
    2 A(I,J)=A1(I,J)
C     ***** TRIANGULARIZE MATRIX A *****
      DO 17 I=1,2
      S=0.0
      DO 4 J=I,3
      R=ABS(A(J,I))
      IF(R-S)4,3,3
    3 S=R
      L=J
    4 CONTINUE
      IF(L-I)5,10,5
    5 DO 6 J=I,3
      S=A(I,J)
      A(I,J)=A(L,J)
    6 A(L,J)=S
      DO 8 J=1,NV
      S=B(I,J)
      B(I,J)=B(L,J)
    8 B(L,J)=S
   10 TEM=A(I,I)
      IF(TEM)11,17,11
   11 IPO=I+1
      DO 16 J=IPO,3
      IF(A(J,I))12,16,12
   12 S=A(J,I)/TEM
      A(J,I)=0.0
      DO 13 K=IPO,3
   13 A(J,K)=A(J,K)-A(I,K)*S
      DO 15 K=1,NV
   15 B(J,K)=B(J,K)-B(I,K)*S
   16 CONTINUE
   17 CONTINUE
C     ***** MODIFY SINGULAR MATRIX *****
      DO 20 I=1,3
      IF(A(I,I))20,19,20
   19 A(I,I)=AMAX1(1.E-25,AMAX1(A(1,1),A(2,2),A(3,3))*1.E-15)
   20 CONTINUE
      DO 24 K=1,NV
      DO 24 I=1,3
      N=4-I
      M=N+1
      TEM=B(N,K)
      IF(3-M)23,21,21
   21 DO 22 J=M,3
   22 TEM=TEM-A(N,J)*B(J,K)
   23 B(N,K)=TEM/A(N,N)
   24 X(N,K)=B(N,K)
      RETURN
      END
      SUBROUTINE AXES(U,V,X,ITYPE)
C     ***** STORE THREE ORTHOGONAL VECTORS EACH 1 ANGSTROM LONG *****
C     ***** ITYPE .GT.0 FOR CARTESIAN,.LE.0 FOR TRICLINIC *****
C     *****IABS(ITYPE)=1 W(1)=U,W(2)=(UXV),W(3)=UX(UXV) *****
C     *****IABS(ITYPE)=2 W(1)=U,W(2)=(UXV)XU,W(3)=(UXV) *****
C     ***** ITYPE=0 W(1)=A,W(2)=(AXB)XA,W(3)=(AXB), ABC=CELL VECTORS ***
      DIMENSION U(3),V(3),W(3,3),X(3,3)
      IT=ITYPE
      IF(IT)115,105,115
  105 U(1)=1.
      U(2)=0.
      U(3)=0.
      V(1)=0.
      V(2)=1.
      V(3)=0.
  115 DO 125 J=1,3
  125 W(J,1)=U(J)
      IF(IABS(IT)-1)145,135,145
  135 CALL NORM(U,V,W(1,2),IT)
      CALL NORM(U,W(1,2),W(1,3),IT)
      GO TO 155
  145 CALL NORM(U,V,W(1,3),IT)
```

```
      CALL NORM(W(1,3),U,W(1,2),IT)
  155 DO 195 I=1,3
      IF(IT)165,165,175
  165 IC=-1
      GO TO 195
  175 IC=1
  195 CALL UNITY(W(1,I),X(1,I),IC)
      RETURN
      END
      SUBROUTINE BOND(Z1,Z2,NB,NA1,NA2)
      REAL*8 Z1,Z2,WD(2),TD,D100,D1000,D100K
      DIMENSION B1(3,3),E(3,3),S(3,3),U(3,3),VUE(3)
      DIMENSION V7(3),W(13,2),Z(3),RESB(2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
C     ***** OBTAIN POSITIONAL PARAMETERS *****
      DATA RESB/.2,.08/
      D100=100.
      D1000=1000.
      D100K=100000.
      NG1=0
      DO 105 J=1,26
  105 W(J,1)=0.
      WD(1)=Z1
      WD(2)=Z2
      DO 135 I=1,2
      CALL XYZ(WD(I),W(4,I),2)
      IF(NG)125,110,125
  110 DO 115 J=1,3
  115 W(J+6,I)=XT(J)
      K=WD(I)/D100K
      L=DMOD(WD(I)/D100,D1000)
      L1=DMOD(WD(I),D100)
      CALL PLTXY(W(4,I),W(2,I))
      IF(EDGE-BRDR*.25)120,128,128
  120 NG=10
  125 NG1=1
      IF (NOUT.GE.0)
     &WRITE(NOUT,136)CHEM(K),K,L,L1,(W(J,I),J=2,9)
      CALL ERPNT(WD(I),800)
      GO TO 134
  128 IF(NJ2-10)130,134,134
  130 IF (NOUT.GE.0)
     &WRITE(NOUT,136)CHEM(K),K,L,L1,(W(J,I),J=2,9)
  134 continue
  135 CONTINUE
  136 FORMAT(1H 10X,A6,3H  (I3,1H,I3,I2,4H)   2F8.2,5X,3F8.3,13X,3F8.4)
      IF(NG1)999,137,999
  137 CALL DIFV(W(7,1),W(7,2),V7)
      DIST=SQRT(VMV(V7,AA,V7))
      IF(MOD(NJ2,2).EQ.0) GO TO 143
      IF(MOD(NJ2,10).EQ.1) GO TO 143
C     ***** LINE BONDS AND CENTERED SYMBOLS (803,813) *
      HGT=SCL*.12
c *** ORTEP-II calls
c     CALL SIMBOL(W(2,1),W(3,1),HGT,MOD(IDINT(WD(1)/D100K),10),0.,-
c    11)
c     CALL SIMBOL(W(2,2),W(3,2),HGT,MOD(IDINT(WD(2)/D100K),10),0.,-
c    12)
c *** Only one centered symbol (*) is available in ORTEP-III.
c *** It is triggered by the negative value for argument 6.
c *** Argument 4 is ignored by SIMBOL.
      CALL SIMBOL(W(2,1),W(3,1),HGT,' ',0.,-1)
      CALL SIMBOL(W(2,2),W(3,2),HGT,' ',0.,-2)
      GO TO 570
C     ***** STICK BONDS FOR 801,802,811,812 *****
  143 KODE=KD(5,NB)
      IF(KODE)145,144,146
  144 NBND=0
      GO TO 148
  145 KODE=-KODE
  146 NBND=128/2**KODE
C     ***** FIND UPPERMOST ATOM PUT IN POSITION ONE *****
  148 IF(VIEW)152,150,152
  150 W(12,1)=1.
      W(12,2)=1.
      IF(W(6,1)-W(6,2))165,175,175
C     *****VECTOR FROM ATOM TO VIEWPOINT *****
  152 DO 160 I=1,2
      DO 155 J=10,12
  155 W(J,I)=-W(J-6,I)
      W(12,I)=W(12,I)+VIEW
C     ***** DISTANCE SQUARED TO VIEWPOINT *****
  160 W(13,I)=VV(W(10,I),W(10,I))
      IF(W(13,2)-W(13,1))165,175,175
C     ***** SWITCH ATOMS *****
  165 DO 170 J=1,13
      T1=W(J,1)
      W(J,1)=W(J,2)
  170 W(J,2)=T1
      TD=WD(1)
      WD(1)=WD(2)
      WD(2)=TD
C     ***** FORM IDEMFACTOR MATRIX *****
  175 DO 180 J=1,3
      E(J,J)=1.
      E(J+1,1)=0.
  180 E(J+5,1)=0.
C     ***** FORM VECTOR SET RADIAL TO BOND *****
      CALL DIFV(W(4,2),W(4,1),DA(1,3))
      CALL UNITY(DA(1,3),V3,1)
C     ***** UNIT VECTOR FROM BOND MIDPOINT TO REFERENCE VIEWPOINT *****
      DO 183 I=1,3
      V2(I)=0.0
      DO 181 J=1,3
  181 V2(I)=V2(I)+AAREV(J,3)*WRKV(J,I)
      IF(VIEW)183,183,182
  182 V2(I)=V2(I)*VIEW-0.5*(W(I+3,1)+W(I+3,2))
  183 CONTINUE
      CALL UNITY(V2,V2,1)
```

```
      T6=ABS(VV(V3,V2))
      IF(.9994-T6)185,185,187
C     ***** ALTERNATE CALC IF BOND IS ALONG REFERENCE VIEW DIRECTION ***
  185 DO 186 J=1,3
  186 V2(J)=W(J+9,1)+W(J+9,2)
      CALL UNITY(V2,V2,1)
      T6=ABS(VV(V3,V2))
      IF(.9994-T6)390,390,187
  187 CALL AXES(V3,V2,B1,1)
  188 T1=CD(3,NB)/SCAL2
      DO 190 J=1,3
      DA(J,1)=-B1(J,2)*T1
  190 DA(J,2)=-B1(J,3)*T1
      IF(NBND)500,500,195
C     ***** SET PLOTTING RESOLUTION FOR BOND *****
  195 T1=CD(3,NB)*SCL
      NRESOL=4
      NBIS=3
      DO 200 J=1,2
      IF(T1.GE.RESB(J)) GO TO 202
      IF(NBND.LE.NRESOL) GO TO 202
      NBIS=NBIS-1
  200 NRESOL=NRESOL*2
  202 NRES1=NRESOL+1
      CALL RADIAL(NBIS)
C     ***** DERIVE QUADRICS FOR EACH ATOM *****
      DO 380 II=1,2
      CALL PAXES(WD(II),2)
      IF(NG)205,210,205
  205 CALL ERPNT(WD(II),800)
      GO TO 999
C     ***** DOES BOND GO TO ELLIPSOID OR TO ENVELOPE *****
  210 T1=3-II*2
      DO 212 J=1,3
      V3(J)=V3(J)*T1
  212 VUE(J)=0.
      IF(KD(5,NB))260,260,215
  215 IF(VMV(V3,Q,W(10,II)))220,260,260
  220 IBND=0
      IF(VIEW)240,240,225
C     ***** DERIVE TANGENT CONE DIRECTLY WITHOUT ROTATING COORDINATES **
  225 T2=-(SCAL2*RMS(1)*RMS(2)*RMS(3))**2
      DO 230 J=1,3
      V1(J)=-W(J+9,II)/SCAL1
      VUE(J)=V1(J)/SCAL2
C     ***** INVERT ELLIPSOID MATRIX *****
      DO 230 K=J,3
      T1=0.0
      DO 228 I=1,3
  228 T1=T1+PAC(J,I)*PAC(K,I)*RMS(I)**2
      U(J,K)=T1
  230 U(K,J)=T1
C     *****  ADD POLARIZED COFACTOR MATRIX TO ELLIPSOID MATRIX *****
      DO 235 J=1,3
      J1=MOD(J,3)+1
      VJ1=V1(J1)
      J2=MOD(J+1,3)+1
      VJ2=V1(J2)
      DO 235 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
      S(J,K)=T2*Q(J,K)+(VJ2*(U(J1,K1)*V1(K2)-U(J1,K2)*V1(K1))
     1               + VJ1*(U(J2,K2)*V1(K1)-U(J2,K1)*V1(K2)))
  235 S(K,J)=S(J,K)
      T5=0.0
      GO TO 300
C     ***** DERIVE TANGENT CYLINDER WITH AXIS ALONG Z *****
  240 T1=-1.0/Q(3,3)
      DO 250 J=1,2
      DO 245 K=1,2
  245 S(K,J)=Q(K,J)+Q(K,3)*Q(J,3)*T1
      S(3,J)=0.0
  250 S(J,3)=0.0
      S(3,3)=0.0
      GO TO 270
C     ***** TRANSFER ELLIPSOID *****
  260 DO 265 J=1,9
  265 S(J,1)=Q(J,1)
      IBND=II
  270 T5=1.
C     ***** CHECK FOR BOND TAPER *****
  300 IF(II-2)305,310,310
  305 RADIUS=1.+T6*TAPER
      GO TO 320
  310 RADIUS=1.-T6*TAPER
  320 CALL MV(S,V3,V4)
      T2=VV(V3,V4)
C     ***** COMPUTE BOND INTERSECTION *****
      KL=5-II-II
      KSTP=NRESOL
      IF(NJ2-21)324,322,322
  322 KSTP=32
  324 DO 335 K=1,65,KSTP
      DO 325 J=1,3
      V6(J)=D(J,K)*RADIUS
  325 V5(J)=V6(J)+VUE(J)
      T3=VV(V5,V4)
      T4=T3*T3-T2*(VMV(V5,S,V5)-T5)
      IF(T4)345,330,330
  330 T4=SQRT(T4)
      T1=(T4-T3)/T2
      T3=(-T4-T3)/T2
      L=K+KL-1
      DO 335 J=1,3
      D(J,L)=(V6(J)+T1*V3(J))*SCL
  335 D(J,L+1)=(-V6(J)-T3*V3(J))*SCL
      IF(IBND+21-NJ2)360,338,360
  338 IF(KD(5,NB))360,360,340
C     ***** FOR LOCAL OVERLAP, MAKE BOND QUADRANGLE TANGENT TO ENVELOPING CONE
  340 T3=VV(VUE,V4)
      T4=T3**2-T2*(VMV(VUE,S,VUE)-T5)
      IF(T4)345,350,350
  345 NG=13
      CALL ERPNT(WD(II),800)
      GO TO 999
  350 T1=(SQRT(T4)-T3)/T2
      DO 355 J=1,3
      T4=(T1*V3(J)*SCL-0.5*(D(J,KL)+D(J,KL+64)))*1.001
      D(J,KL)=D(J,KL)+T4
  355 D(J,KL+64)=D(J,KL+64)+T4
  360 CALL PROJ(D(1,KL),DP(1,II),W(4,II),XO,VIEW,1,65,KSTP)
      IF(IBND-1)370,365,370
```

```
  365 CALL PROJ(D(1,KL+KSTP+1),DP(1,II+64+KSTP),W(4,II),XO,VIEW,1,
     & 65-KSTP,KSTP)
      GO TO 380
C     ***** RETRACE TOP HALF *****
  370 KK=64-(II-1)*KSTP
      DO 375 K=KSTP,KK,KSTP
      L=K+II
      M=L+64
      N=66-L
      DP(1,M)=DP(1,N)
  375 DP(2,M)=DP(2,N)
  380 CONTINUE
C     ***** CHECK FOR LOCAL OVERLAP OR HIDDEN BOND *****
      DO 395 K=1,65,32
      T1=0.
      T2=0.
      DO 385 J=1,2
      T1=T1+(DP(J,K)-W(J+1,1))**2
  385 T2=T2+(DP(J,K+1)-W(J+1,1))**2
      IF(T2-T1)390,390,395
  395 CONTINUE
C     ***** CALL GLOBAL OVERLAP ROUTINE *****
      ICQ=0
      CALL LAP800(NA1,NA2,ICQ)
      IF(NJ2-21)400,999,999
  400 IF(ICQ)390,405,405
C     ***** DRAW BOND OUTLINE *****
  405 CALL DRAW(DP(1,1),0.,0.,3)
      DO 415 K=NRES1,129,NRESOL
  415 CALL DRAW(DP(1,K),0.,0.,2)
      DO 420 K=2,66,NRESOL
  420 CALL DRAW(DP(1,K),0.,0.,2)
      CALL DRAW(DP(1,65),0.,0.,2)
C     ***** DRAW BOND DETAIL *****
  425 K=65
  430 K=K-NBND
      IF(K-1)500,500,435
  435 CALL DRAW(DP(1,K),0.,0.,3)
      CALL DRAW(DP(1,K+1),0.,0.,2)
      K=K-NBND
      IF(K-1)500,500,440
  440 CALL DRAW(DP(1,K+1),0.,0.,3)
      CALL DRAW(DP(1,K),0.,0.,2)
      GO TO 430
  500 HGT=CD(4,NB)
      OFF=CD(5,NB)
      IF(HGT)570,570,510
C     ***** PERSPECTIVE BOND LABEL ROUTINE *****
C     ***** BASE DECISIONS ON REFERENCE SYSTEM *****
  510 K=0
      CALL DIFV(W(7,2),W(7,1),V7)
      CALL VM(V7,AAREV,V1)
      CALL AXES(V1,E(1,3),U,1)
      DO 535 I=1,3
      T1=1.
      IF(I-2)515,515,520
  515 IF(VV(U(1,I),SYMB(1,I)))525,530,530
  520 IF(MOD(K,2))530,525,530
  525 T1=-1.
      K=K+1
  530 DO 535 J=1,3
      U(J,I)=U(J,I)*T1
  535 VT(J,I)=B1(J,I)*T1
      DO 540 J=1,3
  540 VT(J,4)=.5*(W(J+3,1)+W(J+3,2))
C     ***** CHECK FOR EXCESS FORESHORTENING *****
      IF(FORE-ABS(U(3,1)))545,550,550
  545 CALL NORM(U(1,2),SYMB(1,3),VT(1,1),1)
      VT(1,3)=SYMB(1,3)
      VT(2,3)=SYMB(2,3)
      VT(3,3)=SYMB(3,3)
      HGT=CD(6,NB)
      OFF=CD(7,NB)
      IF(HGT)550,999,550
  550 T1=CD(8,NB)
      Z(1)=VT(1,4)-HGT*(11.+3.*T1)/7.
      Z(2)=VT(2,4)+OFF-HGT*.5
      Z(3)=VT(3,4)
      XO(3)=Z(3)
      ITILT=1
      I9=T1+2.
      T9=10.**I9
      DISTR=AINT((DIST*T9)+0.5)/T9 +.0001
      CALL NUMBUR(Z(1),Z(2),HGT,DISTR,0.,I9)
  570 ITILT=0
      IF(NJ2-10)580,999,999
  580 IF (NOUT.GE.0)
     &WRITE (NOUT,571)DIST
  571 FORMAT(1H ,59X,10HDISTANCE =,F8.3/1H )
      GO TO 999
  390 NG=14
      CALL ERPNT(WD(2),800)
  999 RETURN
      END
      SUBROUTINE DIFV(X,Y,Z)
C     VECTOR - VECTOR
C     Z(3)=X(3)-Y(3)
      DIMENSION X(3),Y(3),Z(3)
      Z(1)=X(1)-Y(1)
      Z(2)=X(2)-Y(2)
      Z(3)=X(3)-Y(3)
      RETURN
      END
      SUBROUTINE DRAW(W,DX,DY,NPEN)
      DIMENSION W(3),X(3),Y(3),Z(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      Y(1)=W(1)+DX
      Y(2)=W(2)+DY
      IF(ITILT)115,140,115
C     ***** ROTATE FOR PERSPECTIVE TITLE *****
  115 Y(3)=XO(3)
      DO 120 I=1,3
  120 Z(I)=Y(I)-VT(I,4)
```

```
      DO 130 I=1,3
  130 X(I)=VT(I,1)*Z(1)+VT(I,2)*Z(2)+VT(I,3)*Z(3)+VT(I,4)
      CALL PLTXY(X,Y)
C     ***** CHECK BOUNDRY *****
  140 DO 160 J=1,2
      IF(Y(J)-XLNG(J)+.1)150,150,145
  145 Y(J)=XLNG(J)-.1
  150 IF(Y(J)-.1)155,160,160
  155 Y(J)=.1
  160 CONTINUE
C     ***** CHECK FOR OVERLAP *****
      NCQ=0
      CALL LAPDRW(Y,NPEN,NCQ)
      IF(NCQ)165,165,170
C     ***** CALL PLOTTING ROUTINE IF NO OVERLAPPING ELEMENTS ARE STORED
  165 CALL SCRIBE(Y,NPEN)
  170 RETURN
      END
      SUBROUTINE EDITR
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      common /ns/ npf,ndraw,norient,nvar
      CHARACTER*73 INQ,QUE,hque,tline
      character*80 answer,card
      CHARACTER*1 CH
   10 FORMAT(6X,'        1111111111222222222233333333334444444444555',
     *'55555556666666666777',/,1x,'LINE',1x,'12345678901234567890123345',
     *'67890123456789012345678901234567890123456789012')
      NUM=1
C *** PRINT PART OR ALL OF COMMAND QUE ***
      NUM1=MAX0(1,NUM-2)
  100 WRITE(*,103)
  103 FORMAT(1X)
      WRITE(*,111)(J,QUE(J),J=NUM1,NQUE)
  111 FORMAT(1X,I3,2X,A72)
C *** DISPLAY PROMPT ***
  115 WRITE(*,121)
  121 FORMAT(/,' C=Change line #                          D=Delete line(s
     &) # [#]',/,' I=Insert line before #                  T=Type line(s
     &) [#] [#]'
     &,/,' S=Save modified instruction set         O=Restore original in
     &struction set'
     &,/,' P=Save drawing as Postscript            H=Save drawing as HPG
     &L'
     &,/,' R=Redraw structure on screen            Q=Quit',/,' >-> ',$)
C *** READ COMMAND CHARACTER AND LINE NUMBER(S) ***
      read(*,131)answer
  131 format(a)
      if (answer(1:1).eq.' ') go to 133
      if (answer(1:1).ge.'1' .and. answer(1:1).le.'9') go to 133
      last=iend(answer)
      do 132 i=2,last
         ch=answer(i:i)
         if ((ch.ge.'a'.and.ch.le.'z') .or.
     *       (ch.ge.'A'.and.ch.le.'Z')) go to 133
         ich=ichar(ch)-48
         if ((ch.ne.' ') .and. ((ich.lt.0) .or. (ich.gt.9))) go to 133
  132 continue
      go to 135
  133 write(*,134)
  134 format (/,'***INVALID INPUT! Enter 1 letter and 0, 1, or 2 integer
     *s separated by spaces.***')
      go to 115
  135 answer=answer(1:last)//' 0 0'
C     read(answer,*)ch,num,num2
      open(15,status='scratch')
      write(15,136) answer(1:1),answer(2:75)
  136 format('''',a1,'''',1x,a)
      rewind(15)
      read(15,*)ch,num,num2
      close(15)
      numz=num
      NUM=MAX0(1,NUM)
      if (num2.gt.nque) then
         write(*,137)
  137    format (/,'*** value out of range ***')
         go to 115
      end if
      IF(NUM.GT.NUM2) NUM2=NUM
      write(6,*) ' '
      IF(CH.EQ.'T'.or.ch.eq.'t') GO TO 210
      IF(CH.EQ.'D'.or.ch.eq.'d') GO TO 240
      IF(CH.EQ.'C'.or.ch.eq.'c') GO TO 270
      IF(CH.EQ.'I'.or.ch.eq.'i') GO TO 310
      IF(CH.EQ.'O'.or.ch.eq.'o') GO TO 410
      IF(CH.EQ.'S'.or.ch.eq.'s') GO TO 420
      IF(CH.EQ.'R'.or.ch.eq.'r') GO TO 540
      IF(CH.EQ.'Q'.or.ch.eq.'q') GO TO 590
      IF(CH.EQ.'P'.or.ch.eq.'p') GO TO 510
      IF(CH.EQ.'H'.or.ch.eq.'h') GO TO 520
      GO TO 115
C *** TYPE LINES ***
  210 if (numz.eq.0) num2=nque
      WRITE(*,10)
      WRITE(*,111)(J,QUE(J),J=NUM,NUM2)
      GO TO 115
C *** DELETE LINES ***
  240 if (numz.eq.0) then
         write(6,*) ' *** Supply line number(s) with command ***'
         go to 115
      end if
      DO 260 I=NUM,NUM2
      NQUE=NQUE-1
      NEXT=NEXT-1
      DO 250 J=NUM,NQUE
  250 QUE(J)=QUE(J+1)
  260 QUE(NQUE+1)= '
      GO TO 100
C *** CHANGE AN OLD LINE ***
```

```
  270 if (numz.eq.0) then
         write(6,*) ' *** Supply line number with command ***'
         go to 115
      end if
c     NUM3=MAX0(1,NUM-3)
c     WRITE(*,111)(J,QUE(J),J=NUM3,NUM)
      write(*,271)
  271 format(' *** NOTE: Type @ to substitute a space in the original li
     &ne. ***',/)
      WRITE(*,10)
      WRITE(*,111)NUM,QUE(NUM)
      WRITE(*,276)NUM
  276 FORMAT(1X,I3,'  ',$)
      read(*,281) tline
  281 FORMAT(A72)
      do 282 i=1,72
         if (tline(i:i).ne.' ') then
            if (tline(i:i).eq.'@') then
               que(num)(i:i)=' '
            else
               que(num)(i:i)=tline(i:i)
            end if
         end if
  282 continue
      write(*,283)
  283 format(/,'      Line now reads:')
      WRITE(*,284)QUE(NUM)
  284 format(6x,a)
      write(*,285)
  285 format(/,' Hit ENTER or RETURN key ',$)
      read(*,131) CH
      GO TO 100
C *** INSERT A NEW LINE ***
  310 if (numz.eq.0) then
         write(6,*) ' *** Supply line number with command ***'
         go to 115
      end if
      IF(NQUE+1 .GT. 96) GO TO 115
      NQUE=NQUE+1
      NN=NQUE
      N=NQUE-NUM
      DO 320 J=1,N
      QUE(NN)=QUE(NN-1)
  320 NN=NN-1
      NUM4=MAX0(1,NUM-4)
      NUM1=MAX0(1,NUM-1)
      write(*,10)
      WRITE(*,111)(J,QUE(J),J=NUM4,NUM1)
      WRITE(*,276)NUM
      READ(*,281) QUE(NUM)
      NEXT=NEXT+1
      GO TO 100
C *** RETRIEVE OLD SET OF INSTRUCTIONS ***
  410 DO 415 J=1,NBACK
  415 que(j)=hque(j)
      NUM=1
      NUM1=1
      NQUE=NBACK
      NUM2=NQUE
      GO TO 100
C *** SAVE CURRENT SET OF INSTRUCTIONS ***
  420 CONTINUE
  421 write (*,422)
  422 format(' Enter file name:  ',$)
      read (*,131) answer
      open (16,file=answer,status='new',err=460)
      rewind(NED)
  430 read (NED,131,end=440) card
      write (16,131) card(1:iend(card))
      go to 430
  440 WRITE(16,450)(QUE(I),I=1,NQUE)
  450 FORMAT(A73)
      close(16)
      GO TO 115
  460 write(6,*) 'File already exists.  Choose a different name.'
      go to 421
C *** SAVE PICTURE AS POSTSCRIPT***
  510 ndraw=2
      go to 541
C *** SAVE PICTURE AS HPGL***
  520 ndraw=3
      go to 541
C *** REDRAW PICTURE ***
  540 ndraw=1
c *** default instruction save
  541 open (16,file='TEP.NEW',status='unknown')
      rewind(NED)
  542 read (NED,131,end=543) card
      write (16,131) card(1:iend(card))
      go to 542
  543 WRITE(16,450)(QUE(I),I=1,NQUE)
      close(16)
      call recycle
  590 RETURN
      END
      SUBROUTINE EIGEN (W,VALU,VECT)
C     ***** EIGENVALUES AND EIGENVECTORS OF 3X3 MATRIX *****
      DIMENSION W(3,3),VALU(3),VECT(3,3),A(3,3),B(3,3),V(3),U(3)
      COMMON NG
C     ***** STATEMENT FUNCTION *****
      PHIF(Z)=((B2-Z)*Z+B1)*Z+B0
C     ***** START OF PROGRAM *****
      ERRND=5.E-7
      SIGMA=0.
      DO 115 J=1,3
      DO 115 I=1,3
      TEM=W(I,J)
      A(I,J)=TEM
  115 SIGMA=SIGMA+TEM*TEM
C     ***** CHECK FOR NULL MATRIX *****
      IF(SIGMA)230,230,120
  120 SIGMA=SQRT(SIGMA)
C     ***** FORM CHARACTERISTIC EQUATION *****
      B2=A(1,1)+A(2,2)+A(3,3)
      B1=-A(1,1)*A(2,2)-A(1,1)*A(3,3)-A(2,2)*A(3,3)+A(1,3)*A(3,1)
     1+A(2,3)*A(3,2)+A(1,2)*A(2,1)
      B0=A(1,1)*A(2,2)*A(3,3)+A(1,2)*A(2,3)*A(3,1)+A(1,3)*A(3,2)*A(2,1)-
     1A(1,3)*A(3,1)*A(2,2)-A(1,1)*A(2,3)*A(3,2)-A(1,2)*A(2,1)*A(3,3)
C     ***** FIRST ROOT BY BISECTION *****
      X=0.
      Y=SIGMA
      TEM=PHIF(SIGMA)
```

```
      VNEW=0.0
      IF(B0)135,250,145
  135 IF(TEM)140,140,165
  140 Y=-Y
      GO TO 165
  145 Y=0.
      X=SIGMA
      IF(TEM)165,165,150
  150 X=-X
C     ***** NOW PHIF(X).LT.0.AND.PHIF(Y).GT.0. *****
  165 VNEW=(X+Y)*.5
      DO 225 I=1,40
  175 IF(PHIF(VNEW))180,250,185
  180 X=VNEW
      GO TO 200
  185 Y=VNEW
  200 VOLD=VNEW
      VNEW=(X+Y)*.5
      TEM=ABS(VOLD-VNEW)
      IF(TEM-ERRND)250,250,205
  205 IF(VOLD)210,225,210
  210 IF (ABS(TEM/VOLD)-ERRND)250,250,225
  225 CONTINUE
C     ***** DID NOT CONVERGE, SET ERROR INDICATOR *****
  230 NG=6
      GO TO 400
C     ***** STORE FIRST ROOT *****
  250 U(3)=VNEW
C     ***** DEFLATE *****
      C1=B2-VNEW
      C0=B1+C1*VNEW
C     ***** SOLVE QUADRATIC *****
      TEM=C1*C1+4.*C0
      IF(TEM)255,265,260
C     ***** IGNORE IMAGINARY COMPONENT OF COMPLEX ROOT *****
  255 TEM=0.
      GO TO 265
  260 TEM=SQRT(TEM)
  265 U(1)=.5*(C1-TEM)
      U(2)=.5*(C1+TEM)
C     ***** SORT ROOTS *****
      DO 275 J=1,2
      IF(U(J)-U(3))275,275,270
  270 TEM=U(J)
      U(J)=U(3)
      U(3)=TEM
  275 CONTINUE
      LLL=-2
      DO 375 III=1,2
C     ***** CHECK FOR MULTIPLE ROOTS *****
      TEM=ERRND*100.
      NG=0
      L=1
      DO 305 I=1,2
      IF(U(I+1)-U(I)-TEM)300,300,290
  290 IF(U(I))295,305,295
  295 IF(ABS((U(I+1)-U(I))/U(I))-TEM)300,300,305
  300 L=L-1
      NG=NG-2*I
  305 CONTINUE
      IF(LLL-L)308,400,400
  308 LLL=L
C     ***** EIGENVECTOR ROUTINE *****
      DO 375 II=1,3
      T1=U(II)
      IF(L)315,310,322
C     ***** TWO VECTORS NULL FOR DOUBLE ROOT *****
  310 IF(NG+5-II)315,322,315
C     ***** ALL VECTORS NULL FOR TRIPLE ROOT *****
  315 DO 320 J=1,3
  320 VECT(J,II)=0.0
      GO TO 375
  322 DO 325 J=1,3
  325 A(J,J)=W(J,J)-T1
      SMAX=0.0
      DO 355 I=1,3
      I1=1
      IF(I-2)335,335,340
  335 I1=I+1
  340 B(I,1)=A(I,2)*A(I1,3)-A(I,3)*A(I1,2)
      B(I,2)=A(I,3)*A(I1,1)-A(I,1)*A(I1,3)
      B(I,3)=A(I,1)*A(I1,2)-A(I,2)*A(I1,1)
      TEM=B(I,1)**2+B(I,2)**2+B(I,3)**2
      IF(TEM-SMAX)355,355,350
  350 SMAX=TEM
      IMAX=I
  355 CONTINUE
      IF(SMAX)353,353,360
  353 NG=7
      GO TO 375
  360 SMAX=SQRT(SMAX)
      DO 365 J=1,3
  365 V(J)=B(IMAX,J)/SMAX
C     ***** REFINE EIGENVECTOR *****
      CALL AXEQB(A,V,V,1)
      TEM=AMAX1(ABS(V(1)),ABS(V(2)),ABS(V(3)))
      DO 370 J=1,3
  370 V(J)=V(J)/TEM
      CALL UNITY(V,VECT(1,II),1)
C     ***** REFINE EIGENVALUE *****
      T1=VMV(VECT(1,II),W,VECT(1,II))
      U(II)=T1
  375 VALU(II)=T1
  400 RETURN
      END
      SUBROUTINE ERPNT(TD,N)
      REAL*8 TD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      character*63 msg(18)
      data msg /
     1 'No sentinel found after reading 96 symmetry cards',
     2 'No sentinel found after reading parameter cards for 100 atoms',
     3 'Aniso temp factor coefs form non-positive definite matrix',
```

```
     4 'Symmetry operation no. is higher than no. of input operations',
     5 'Atom number is higher than the number of input atoms',
     6 'Null temp factor matrix or failure in bisection routine',
     7 'Eigenvector routine failure due to null vector',
     8 'Error initializing screen driver',
     9 'Unidentified instruction number',
     a 'Atom out of bounds',
     b 'No vector search codes',
     c 'Insufficient number of atoms in ATOMS list',
     d 'Imaginary bond intersection (i.e., bond larger than atom)',
     e 'Hidden (end-on) bond',
     f 'Null vector as base line',
     g 'ATOMS array is full',
     h 'Maximum number of overlapping atoms (20) exceeded',
     i 'Maximum number of overlapping bonds (30) exceeded' /
      IF (NOUT.GE.0) then
         WRITE (NOUT,115)NG,TD,N
 115     FORMAT(1H ,10X,10HFAULT NG =,I3,F10.0,I6)
         write (nout,116) msg(ng)
 116     format(1H ,10x,a,/)
      end if
      NG=0
      RETURN
      END
      SUBROUTINE EXITNG(ING)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      character*12 routin(18)
      character*63 msg(18)
      data routin /
     1 'PRELIM', 'PRELIM', 'PRELIM', 'ATOM, PAXES', 'ATOM, PAXES',
     2 'EIGEN', 'EIGEN', 'INITSC ', 'MAIN, SPARE', 'BOND, F700',
     3 'F800', 'F600, SEARCH', 'BOND', 'BOND', 'F900', 'STORE',
     4 'LAP700, LAPAB', 'LAP800, LAPAB' /
      data msg /
     1 'No sentinel found after reading 96 symmetry cards',
     2 'No sentinel found after reading parameter cards for 100 atoms',
     3 'Aniso temp factor coefs form non-positive definite matrix',
     4 'Symmetry operation no. is higher than no. of input operations',
     5 'Atom number is higher than the number of input atoms',
     6 'Null temp factor matrix or failure in bisection routine',
     7 'Eigenvector routine failure due to null vector',
     8 'Error initializing screen driver',
     9 'Unidentified instruction number',
     a 'Atom out of bounds',
     b 'No vector search codes',
     c 'Insufficient number of atoms in ATOMS list',
     d 'Imaginary bond intersection (i.e., bond larger than atom)',
     e 'Hidden (end-on) bond',
     f 'Null vector as base line',
     g 'ATOMS array is full',
     h 'Maximum number of overlapping atoms (20) exceeded',
     i 'Maximum number of overlapping bonds (30) exceeded' /
      if (ng.gt.0) then
         if (nout.gt.0) then
            write (nout,101) ing
            write (nout,102) routin(ing)
            write (nout,103) msg(ing)
         end if
         write (*,101) ing
         write (*,102) routin(ing)
         write (*,103) msg(ing)
      end if
      IF(NOUT.GT.0) CLOSE(NOUT,STATUS='KEEP')
 101  format('Fault Indicator:  ',i2)
 102  format('Subroutine(s) Involved:  ',a)
 103  format('Fault:  ',a)
      STOP
      END
      SUBROUTINE F200
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans

c *** NO drawing
      if (ndraw.eq.0) return

      go to (201,202,201,204,205), nj2
c *** initialize plotting (201 or 203 inst) ***
  201 xtrans=0.
      ytrans=0.
      if (ndraw .eq. 1) call initsc
      if (ndraw .eq. 2) call initps
      if (ndraw .eq. 3) call inithp
      if (ndraw .eq. 9) then
         open(unit=npf,file='TEP.EDT',status='unknown')
         nvar=1
      end if
      return
c *** change origin of plotting area or terminate (202 inst) ***
  202 if (ain(1) .eq. 0. .and. ain(2) .eq. 0.) then
         if (ndraw .eq. 2) call endps
         if (ndraw .eq. 3) call endhp
         if (ndraw .eq. 1) call endsc
         if (ndraw .eq. 9) close(npf)
      else
         xtrans=ain(1)
         ytrans=ain(2)
         if (ndraw .eq. 9) write (npf,203) xtrans,ytrans
  203    format('TRN',2(1x,f10.6))
      end if
      return
c *** change plot color (204 inst) ***
  204 icolor=ain(1)
      if (ndraw .eq. 1) call colrsc(icolor)
```

```
      if (ndraw .eq. 2) call colrps(icolor)
      if (ndraw .eq. 3) call colrhp(icolor)
      if (ndraw .eq. 9) call colrsc(icolor)
      return
c *** change pen width (205 inst) ***
c *** parameter units are thousandths of an inch (default=5)
  205 penw=ain(1)
      if (ndraw .eq. 1) call penwsc(penw)
      if (ndraw .eq. 2) call penwps(penw)
      if (ndraw .eq. 3) call penwhp(penw)
      if (ndraw .eq. 9) call penwsc(penw)
      return
      end
      SUBROUTINE F400
C     ***** ATOM LIST FUNCTIONS *****
      REAL*8 D100,D1000,D100K,TD,TD1,TD2
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      D100=100.
      D1000=1000.
      D100K=100000.
      NG=0
      IF(LATM)402,402,400
  400 DO 401 I=1,LATM
  401 CALL ATOM(ATOMID(I),ATOMS(1,I))
  402 IF(MOD(NJ2,10)-1)499,404,403
  403 IF(MOD(NJ2,10)-7)406,404,499
  406 CALL SEARC
      GO TO 499
C     ***** STORES (401) OR REMOVES (411) RUNS OF ATOMS *****
C     ***** RUN HIERARCHY = ATOM NO./SYM/ A/B/C TRANS. *****
  404 II=1
C     ***** FIND RUNS IN AIN ARRAY *****
  405 TD1=AIN(II)
      IF(TD1)410,410,420
  410 II=II+1
      IF(140-II)499,405,405
  420 JJ=II
C     ***** SET INITIAL RUN VALUES *****
      M1=TD1/D100K
      M2=DMOD(TD1,D100)
      M5=DMOD(TD1/D100,D1000)
      IF(M5)422,422,423
  422 M5=555
  423 M3=M5/100
      M4=MOD(M5/10,10)
      M5=MOD(M5,10)
  425 JJ=JJ+1
      IF(140-JJ)435,430,430
  430 TD2=-AIN(JJ)
      IF(TD2)435,425,440
  435 II=JJ-1
C     ***** SET TERMINAL VALUES FOR DEGENERATE RUN *****
      N1=M1
      N2=M2
      N3=M3
      N4=M4
      N5=M5
      GO TO 450
  440 II=JJ
C     ***** SET TERMINAL RUN VALUES *****
      N1=TD2/D100K
      N2=DMOD(TD2,D100)
      N5=DMOD(TD2/D100,D1000)
      IF(N5)445,445,446
  445 N5=555
  446 N3=N5/100
      N4=MOD(N5/10,10)
      N5=MOD(N5,10)
C     ***** LOOP THROUGH ALL RUNS *****
  450 DO 490 L5=M5,N5
      DO 490 L4=M4,N4
      DO 490 L3=M3,N3
      DO 490 L2=M2,N2
      DO 490 L1=M1,N1
      TD=DBLE(L1)*D100K+DBLE(L3*10000+L4*1000+L5*100+L2)
      CALL ATOM(TD,V1(1))
      IF(NG)455,458,455
  455 CALL ERPNT(TD,401)
      GO TO 490
C     ***** CHECK IDENT CODE IF 407/417 INSTRUCTION *****
  458 IF(MOD(NJ2,10)-7)475,460,490
  460 ID1=IDENT(1,L1)
      ID2=IDENT(2,L1)
      IF(NCD)490,490,465
  465 DO 470 J=1,NCD
          if (kd(1,j).gt.0. .and. kd(3,j).gt.0.) then
             if ((id1.ge.kd(1,j) .and. id1.le.kd(2,j)) .and.
     &            (id2.ge.kd(3,j) .and. id2.le.kd(4,j))) go to 475
          else if (kd(1,j).gt.0.) then
             if (id1.ge.kd(1,j) .and. id1.le.kd(2,j)) go to 475
          else if (kd(3,j).gt.0.) then
             if (id2.ge.kd(3,j) .and. id2.le.kd(4,j)) go to 475
          end if
c     IF(ID1-KD(1,J))470,467,467
c 467 IF(KD(2,J)-ID1)470,468,468
c 468 IF(ID2-KD(3,J))470,469,469
c 469 IF(KD(4,J)-ID2)470,475,475
  470 CONTINUE
      GO TO 490
  475 CALL STOR(TD)
  490 CONTINUE
      GO TO 410
  499 RETURN
      END
      SUBROUTINE F500
      DIMENSION RM(3,3),V(3,4)
      REAL*8 TD,D100,D1000,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
```

```
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      CHARACTER*8 CHEM
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      NG=0
      D100=100.
      D1000=1000.
      D100K=100000.
      IF(NJ2-11)500,700,710
  500 IF(NJ2-1)710,501,510
  501 TD=AIN(1)
      CALL ATOM(TD,ORGN)
      IF(NG)502,504,502
  502 CALL ERPNT(TD,501)
      CALL EXITNG(NG)
  504 DO 506 K=1,4
      TD=AIN(K+1)
      CALL ATOM(TD,V(1,K))
      IF(NG)502,506,502
  506 CONTINUE
      DO 507 J=1,3
      V1(J)=V(J,2)-V(J,1)
  507 V2(J)=V(J,4)-V(J,3)
      IND=-1
      IF(AIN(7))509,509,508
  508 IND=-2
  509 CALL AXES(V1,V2,REFV,IND)
      GO TO 670
  510 IF(NJ2-4)515,511,599
C      ***** SHIFT ORIGIN FOR PROJECTION AXIS (IN INCHES) *****
  511 DO 513 J=1,3
      DO 512 K=1,3
      T1=AIN(K)
  512 ORGN(J)=ORGN(J)+REFV(J,K)*T1/SCAL1
      T2=AIN(J)
  513 XO(J)=XO(J)+T2
      GO TO 675
C      ***** FORM ROTATION MATRIX *****
  515 DO 514 J=1,3
      DO 514 K=1,3
  514 V(J,K)=REFV(J,K)
      DO 517 L=1,139,2
      I=AIN(L)
      IF(I) 532,519,516
  516 X=AIN(L+1)*1.7453293D-2
      T1=COS(X)
      T2=SIN(X)
      I3=MOD(I+2,3)+1
      I1=MOD(I3,3)+1
      I2=MOD(I1,3)+1
      RM(I1,I1)=T1
      RM(I1,I2)=T2
      RM(I1,I3)=0.0
      RM(I2,I1)=-T2
      RM(I2,I2)=T1
      RM(I2,I3)=0.0
      RM(I3,I1)=0.0
      RM(I3,I2)=0.0
      RM(I3,I3)=1.0
  517 CALL MM(V,RM,V)
  519 IF(NJ2-3)518,525,599
  518 DO 522 J=1,3
      DO 522 I=1,3
  522 REFV(I,J)=V(I,J)
      GO TO 552
  525 DO 528 J=1,3
      DO 528 I=1,3
  528 WRKV(I,J)=V(I,J)
      GO TO 552
  532 IF(NJ2-3)535,552,599
  535 I=MOD(-I,3)
      DO 542 J=1,I
      DO 542 K=1,3
      T1=REFV(K,3)
      REFV(K,3)=REFV(K,2)
      REFV(K,2)=REFV(K,1)
  542 REFV(K,1)=T1
  552 CONTINUE
      IF(NJ2-3)670,582,599
  582 CALL MM(AA,WRKV,AAWRK)
      GO TO 699
  599 IF(NJ2-5)699,600,607
  600 IF(LATM-1)605,610,610
  605 NG=12
  606 CALL ERPNT(0.D0,506)
      CALL EXITNG(NG)
  607 IF(NJ2-6)699,608,710
  608 IF(LATM-3)605,610,610
  610 DO 612 J=1,3
      V2(J)=0.0
      DO 612 I=1,3
  612 RM(I,J)=0.0
      AWT=0.0
      DO 620 K=1,LATM
      CALL ATOM(ATOMID(K),ATOMS(1,K))
      T2=1.0
      IF(NCD)618,618,613
  613 I1=ATOMID(K)/D100K
      DO 616 J=1,NCD
      if (kd(5,j).eq.1) i1=ident(1,k)
      if (kd(5,j).eq.2) i1=ident(2,k)
      T2=CD(1,J)
      IF(I1-KD(1,J))616,614,614
  614 IF(KD(2,J)-I1)616,618,618
  616 CONTINUE
      GO TO 620
  618 AWT=AWT+T2
      DO 619 J=1,3
  619 V2(J)=V2(J)+ATOMS(J,K)*T2
  620 CONTINUE
      IF(AWT)605,605,621
C     ***** PUT ORIGIN AT CENTER OF GRAVITY *****
  621 DO 622 J=1,3
  622 ORGN(J)=V2(J)/AWT
```

```
      IF(NJ2-6)699,624,710
C     ***** FORM PRODUCT-MOMENT MATRIX FOR ATOMS IN ATOM LIST *****
  624 DO 630 K=1,LATM
      T2=1.0
      IF(NCD)628,628,625
  625 I1=ATOMID(K)/D100K
      DO 627 J=1,NCD
      if (kd(5,j).eq.1) i1=ident(1,k)
      if (kd(5,j).eq.2) i1=ident(2,k)
      T2=CD(1,J)
      IF(I1-KD(1,J))627,626,626
  626 IF(KD(2,J)-I1)627,628,628
  627 CONTINUE
      GO TO 630
  628 DO 629 J=1,3
      T1=(ATOMS(J,K)-ORGN(J))*T2
      DO 629 I=1,3
  629 RM(I,J)=T1*(ATOMS(I,K)-ORGN(I))+RM(I,J)
      T1=0.03/(RM(1,1)+RM(2,2)+RM(3,3))
  630 CONTINUE
      DO 632 J=1,3
      DO 632 I=1,3
  632 RM(I,J)=RM(I,J)*T1
C     ***** TRANSFORM TO INERTIAL AXIS SYSTEM *****
      CALL MM(RM,AA,DA)
      CALL EIGEN(DA,RMS,PAT)
      IF(RMS(2))605,605,635
  635 IF(NG)640,633,606
C     ***** MAKE SURE VECTORS ARE ORTHOGONAL --> NEW REFERENCE VECTORS *
  633 CALL AXES(PAT(1,3),PAT(1,1),REFV,-1)
      GO TO 665
C     ***** TWO EQUAL EIGENVECTORS SPECIAL CASE *****
  640 IF(NG+6)665,665,645
  645 N=NG+5
      CALL UNITY(PAT(1,N),V1,-1)
      DO 650 K=1,3
      IF(ABS(VMV(V1,AA,REFV(1,K)))-.58)655,650,650
  650 CONTINUE
  655 CALL AXES(V1,REFV(1,K),DA,-1)
      DO 660 K=1,3
      L=MOD(N-K+2,3)+1
      DO 660 J=1,3
  660 REFV(J,L)=DA(J,K)
  665 NG=0
  670 CALL MM(AA,REFV,AAREV)
  675 DO 680 J=1,3
      DO 680 I=1,3
      WRKV(I,J)=REFV(I,J)
  680 AAWRK(I,J)=AAREV(I,J)
C     ***** ELIMINATE ALL PREVIOUSLY STORED OVERLAP INFORMATION ****
C     ***** (ALL INSTRUCTIONS FROM 501 THROUGH 510 DO THIS *****
  699 CALL LAP500(0)
      GO TO 710
C     ***** STORE NEW OVERLAP INFORMATION (INSTRUCTION 511) *****
  700 CALL LAP500(1)
  710 RETURN
      END
      SUBROUTINE F600
C     ***** SCALING AND CENTERING FUNCTIONS *****
      DIMENSION MAX(3),SCAL(4),X(3),XMAX(3),XMIN(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      dimension crtval(99)
      data crtval /
     1 0.3389, 0.4299, 0.4951, 0.5479, 0.5932, 0.6334, 0.6699, 0.7035,
     2 0.7349, 0.7644, 0.7924, 0.8192, 0.8447, 0.8694, 0.8932, 0.9162,
     3 0.9386, 0.9605, 0.9818, 1.0026, 1.0230, 1.0430, 1.0627, 1.0821,
     4 1.1012, 1.1200, 1.1386, 1.1570, 1.1751, 1.1932, 1.2110, 1.2288,
     5 1.2464, 1.2638, 1.2812, 1.2985, 1.3158, 1.3330, 1.3501, 1.3672,
     6 1.3842, 1.4013, 1.4183, 1.4354, 1.4524, 1.4695, 1.4866, 1.5037,
     7 1.5209, 1.5382, 1.5555, 1.5729, 1.5904, 1.6080, 1.6257, 1.6436,
     8 1.6616, 1.6797, 1.6980, 1.7164, 1.7351, 1.7540, 1.7730, 1.7924,
     9 1.8119, 1.8318, 1.8519, 1.8724, 1.8932, 1.9144, 1.9360, 1.9580,
     a 1.9804, 2.0034, 2.0269, 2.0510, 2.0757, 2.1012, 2.1274, 2.1544,
     b 2.1824, 2.2114, 2.2416, 2.2730, 2.3059, 2.3404, 2.3767, 2.4153,
     c 2.4563, 2.5003, 2.5478, 2.5997, 2.6571, 2.7216, 2.7955, 2.8829,
     d 2.9912, 3.1365, 3.3682 /
C     ***** DEL = 1. FOR INCRUMENTING FUNCTIONS *****
C     ***** DEL = 0. FOR REGULAR FUNCTIONS *****
      DEL=FLOAT(MOD(NJ2/10,2))
      NJ2=MOD(NJ2,10)
C     ***** EXPLICIT ORIGIN AND SCALE *****
      T1=AIN(1)
      IF(T1)602,604,602
  602 XO(1)=T1+XO(1)*DEL
  604 T2=AIN(2)
      IF(T2)606,608,606
  606 XO(2)=T2+XO(2)*DEL
  608 T3=AIN(3)
      IF(T3)612,612,609
  609 IF(DEL)611,611,610
  610 SCAL1=SCAL1*T3
      GO TO 612
  611 SCAL1=T3
  612 T4=AIN(4)
      IF(T4)615,616,614
C     ***** SET ELLIPSOID SCALE FACTOR *****
  614 SCAL2=T4
      go to 616
  615 t4=-t4
      if (t4.gt.0. .and. t4.lt.1.) t4=100.*t4
      it4=t4
      scal2=crtval(it4)
C     ***** AUTOMATIC ORIGIN AND/OR SCALE *****
  616 IF(NJ2-2)790,622,620
  620 XO(1)=XLNG(1)*.5
      XO(2)=XLNG(2)*.5
  622 IF(NJ2-3)625,640,625
  625 SCAL1=1.
  630 IF(LATM-1)635,635,640
  635 NG=12
      CALL ERPNT(0.D0,602)
      CALL EXITNG(NG)
  640 DO 650 J=1,3
```

```
      XMAX(J)=-1.E5
  650 XMIN(J)=1.E5
C     ***** FIT BOX AROUND SET OF ATOMS *****
      DO 670 I=1,LATM
      CALL XYZ(ATOMID(I),ATOMS(1,I),3)
      IF(NG)652,653,652
  652 CALL ERPNT(ATOMID(I),600)
      GO TO 670
  653 DO 668 J=1,3
      T1=ATOMS(J,I)
      IF(XMAX(J)-T1)655,660,660
  655 XMAX(J)=T1
      MAX(J)=I
  660 IF(T1-XMIN(J))665,668,668
  665 XMIN(J)=T1
  668 CONTINUE
  670 CONTINUE
C     ***** KM=TOP ATOM *****
      KM=MAX(3)
      SMULT=1.
      DO 780 M=1,5
      IF(M-2)740,675,678
C     ***** CHECK VIEW DISTANCE *****
  675 IF(VIEW)785,785,680
  678 IF(NJ2-3)680,785,680
  680 T1=ATOMS(3,KM)*SMULT
      IF(VIEW*.5-T1)685,690,690
C     ***** INCREASE VIEW DISTANCE *****
  685 VIEW=2.*T1
C     ***** FIND PERSPECTIVE PROJECTION LIMITS *****
  690 DO 700 J=1,2
      XMAX(J)=-1.E5
  700 XMIN(J)=1.E5
      DO 725 I=1,LATM
      DO 705 J=1,3
  705 X(J)=ATOMS(J,I)*SMULT
      T2=VIEW/(VIEW-X(3))
      DO 725 J=1,2
      T1=X(J)*T2
      IF(XMAX(J)-T1)710,715,715
  710 XMAX(J)=T1
  715 IF(T1-XMIN(J))720,725,725
  720 XMIN(J)=T1
  725 CONTINUE
C     ***** REFINE PARAMETERS *****
  740 IF(NJ2 -3)745,742,755
  742 SMUL2=1.
      GO TO 765
C     ***** AUTOMATIC SCALE ONLY *****
  745 DO 750 J=1,2
      T2=XO(J)
      SCAL(J)=(BRDR-T2)/XMIN(J)
  750 SCAL(J+2)=(XLNG(J)-BRDR-T2)/XMAX(J)
      SMUL2=AMIN1(SCAL(1),SCAL(2),SCAL(3),SCAL(4))
      GO TO 780
C     ***** AUTOMATIC SCALE AND POSITION *****
  755 DO 760 J=1,2
  760 SCAL(J)=(XLNG(J)-BRDR*2.)/(XMAX(J)-XMIN(J))
      SMUL2=AMIN1(SCAL(1),SCAL(2))
C     ***** AUTOMATIC POSITION *****
  765 DO 770 J=1,2
  770 XO(J)=.5*(XLNG(J)-SMUL2*(XMAX(J)+XMIN(J)))
  780 SMULT=SMULT*SMUL2
      VIEW=VIEW*SMUL2
  785 SCAL1=SCAL1*SMULT
  790 SCL=SCAL1*SCAL2
C     ***** ELIMINATE ALL PREVIOUSLY STORED OVERLAP INFORMATION *****
      CALL LAP500(0)
      RETURN
      END
      SUBROUTINE F700
C     ***** SUBROUTINE TO DRAW ELLIPSOIDS *****
      DIMENSION EYE(3),VIEWV(3),X(3),Z(3)
      REAL*8 TD,TD2,D100,D1000,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans
C     ***** SET ELLIPSOID GRAPHIC DETAILS *****
      D100=100.
      D1000=1000.
      D100K=100000.
      ITILT=0
      NG=0
      NFIRST=1
      NPLANE=AIN(1)
      IF(NPLANE-1)720,715,720
  715 NFIRST=4
      NPLANE=4
  720 NSOLID=AIN(2)
      NDOT=64/2**(IABS(NSOLID))
      LINES=AIN(3)
      NDASH=AIN(4)
      CHSYM=AIN(5)
      T6=AIN(6)
      DH=T6-CHSYM*17./7.
      T7=AIN(7)
      DV=T7-CHSYM*.5
C     ***** ESTABLISH REFERENCE POINT OF VIEW *****
      T1=1.E6
      IF(VIEW)740,740,735
  735 T1=VIEW/SCAL1
  740 DO 741 J=1,3
  741 EYE(J)=REFV(J,3)*T1+ORGN(J)
      LNS=-1
C     ***** LOOP THROUGH ATOM LIST *****
      DO 1105 ITOM=1,LATM
      TD=ATOMID(ITOM)
      K=TD/D100K
      IF(AIN(10))744,744,7412
```

```
 7412 IF(AIN(12)-1.0D0)742,7414,7416
 7414 TD2=IDENT(1,K)
      GO TO 7422
 7416 TD2=IDENT(2,K)
      GO TO 7422
  742 TD2=DINT(TD/D100K)
 7422 IF(TD2-AIN(10))1105,743,743
  743 IF(AIN(11)-TD2)1105,744,744
  744 CALL XYZ(TD,X,2)
      IF(NG)758,746,758
  746 CALL PLTXY(X,Z)
      L=DMOD(TD/D100,D1000)
      L1=DMOD(TD,D100)
      if (ndraw.eq.1) WRITE (npf,750) CHEM(K),K,L,L1,
     &                                Z(1)+xtrans,Z(2)+ytrans
      IF(NJ2-10)747,754,754
  747 LNS=MOD(LNS+1,18)
      IF(LNS)749,748,749
  748 IF (NOUT.GE.0)
     &WRITE (NOUT,751)(TITLE(I),I=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,752)
  749 IF (NOUT.GE.0)
     &WRITE (NOUT,750) CHEM(K),K,L,L1,Z(1),Z(2),
     1(X(I),I=1,3),(XT(I),I=1,3)
  750 FORMAT(1H 10X,A6,3H  (I3,1H,I3,I2,4H)   2F8.2,3X,3F8.3,11X,3F8.4)
  751 FORMAT(1H1,10X,18A4)
  752 FORMAT(1H010X,18HSYMBOL   ATOM CODE7X,16HPLOTTER X,Y(IN.) 3X,21HCA
     1RTESIAN X,Y,Z (IN.)15X,20HCRYSTAL SYSTEM X,Y,Z/1H 19X,45H(DIRECTIO
     2N COSINES(I,J),I=1,3),RMSD(J)),J=1,312X,42HFOR PRINCIPAL AXES BASE
     3D ON WORKING SYSTEM/1H )
  754 IF(EDGE-BRDR*.75)755,760,760
  755 NG=10
  758 CALL ERPNT(TD,700)
      GO TO 1105
C     ***** CALL OVERLAP ROUTINE *****
  760 ICQ=0
      CALL LAP700(ITOM,ICQ)
      IF(ICQ)762,764,764
C     ***** OMIT HIDDEN ATOM *****
  762 NG=14
      GO TO 758
  764 IF(CHSYM)775,775,765
C     ***** PLOT CHEMICAL SYMBOLS *****
  765 T4=1.
      IF(VIEW)767,767,766
  766 T4=VIEW/(VIEW-X(3))
  767 T3=CHSYM*T4
      T4=DISP*T4*.5
      V1(1)=X(1)+DH*SYMB(1,1)+DV*SYMB(1,2)
      V1(2)=X(2)+DH*SYMB(2,1)+DV*SYMB(2,2)
      V1(3)=X(3)
      CALL PLTXY(V1,V3)
      IF(EDGE-CHSYM)775,768,768
  768 V2(3)=0.
      DO 770 I=1,3,2
      V2(1)=V3(1)+FLOAT(I-2)*T4
      DO 770 J=1,3,2
      V2(2)=V3(2)+FLOAT(J-2)*T4
      CALL SIMBOL(V2(1),V2(2),T3,CHEM(K),THETA,6)
      IF(T4)775,775,770
  770 CONTINUE
  775 IF(NPLANE)1105,1105,780
C     ***** ELLIPSOID PRINC VECTORS TOWARD VIEWER *****
  780 CALL PAXES(TD,2)
      IF(NG)758,783,758
  783 CALL DIFV(EYE,XT,VIEWV)
      CALL UNITY(VIEWV,VIEWV,-1)
      CALL VM(VIEWV,AA,V2)
      DO 795 I=1,3
      IF(VV(V2,PAT(1,I)))785,795,795
  785 DO 790 J=1,3
      PAC(J,I)=-PAC(J,I)
  790 PAT(J,I)=-PAT(J,I)
  795 CONTINUE
      DO 800 J=1,3
      PAC(J,4)=PAC(J,1)
  800 PAC(J,5)=PAC(J,2)
      IF(NJ2-10)802,803,803
  801 FORMAT(1H ,13X,3(3X,3F8.4,F8.5)/1H )
  802 IF (NOUT.GE.0)
     &WRITE (NOUT,801) ((PAC(J,K),J=1,3),RMS(K) ,K=1,3)
C     ***** V4 = VECTOR NORMAL TO POLAR PLANE *****
  803 continue
      CALL VM(VIEWV,AAWRK,V6)
      CALL UNITY(V6,V6,1)
      CALL MV(Q,V6,V4)
      CALL UNITY(V4,V4,1)
C     ***** SET PLOTTING RESOLUTION FOR ELLIPSOID *****
      t3a=sqrt(rms(3)*rms(2))
      t3b=sqrt(rms(2)*rms(1))
      t3c=sqrt(rms(3)*rms(1))
      T3=((t3a+t3b+t3c)/3.0)*SCL
      NRESOL=1
      NBIS=5
      DO 805 J=1,3
      IF(T3-RES(J))804,810,810
  804 NBIS=NBIS-1
  805 NRESOL=NRESOL*2
  810 NRES1=NRESOL+1
C     ***** LOOP THROUGH PRINC AND POLAR PLANES *****
      DO 1100 II=NFIRST,NPLANE
      II0=MOD(II+2,3)+1
      II1=MOD(II,3)+1
      II2=MOD(II+1,3)+1
C     ***** GENERATE CONJUGATE DIAMETERS *****
      IF(.99938- ABS(VV(V4,PAC(1,II2))))820,820,830
  820 T1=RMS(II0)*SCL
      T2=RMS(II1)*SCL
      DO 825 J=1,3
      DA(J,1)=PAC(J,II0)*T1
  825 DA(J,2)=PAC(J,II1)*T2
      GO TO 850
  830 CALL NORM(PAC(1,II0),PAC(1,II1),V1,1)
      CALL NORM(V1,V4,V2,1)
      CALL UNITY(V2,V2,1)
      CALL MV(Q,V2,V3)
      IF(II-4)835,840,840
  835 CALL NORM(V3,V1,V5,1)
      GO TO 843
  840 CALL NORM(V3,V4,V5,1)
  843 CALL UNITY(V5,V5,1)
```

```
      T1=SCL/SQRT(VMV(V2,Q,V2))
      T2=SCL/SQRT(VMV(V5,Q,V5))
      DO 845 J=1,3
      DA(J,1)=V2(J)*T1
  845 DA(J,2)=V5(J)*T2
C     ***** GENERATE ELLIPSE *****
  850 CALL RADIAL(NBIS)
      IF(II-4)900,851,851
  851 IF(NSOLID)859,859,852
C     ***** PLOT DOTTED BOUNDARY ELLIPSE *****
  852 IF(NDOT-NRESOL)853,855,855
  853 CALL RADIAL(NSOLID-1)
  855 CALL PROJ(D,DP,X,XO,VIEW,1,129,NDOT)
      DO 857 J=1,129,NDOT
      CALL DRAW(DP(1,J),DISP,DISP,3)
      DO 856 I=1,3,2
      T1=FLOAT(I-2)*DISP
      DO 856 K=1,3,2
      T2=FLOAT(K-2)*DISP
      CALL DRAW(DP(1,J),T1,T2,2)
      IF(DISP)857,857,856
  856 CONTINUE
  857 CONTINUE
      GO TO 1100
C     ***** PLOT SOLID BOUNDARY ELLIPSE *****
  859 CALL PROJ(D,DP,X,XO,VIEW,1,129,NRESOL)
      CALL DRAW(DP,0.,0.,3)
      DO 860 J=NRES1,129,NRESOL
  860 CALL DRAW(DP(1,J),0.,0.,2)
      IF(DISP)1100,1100,865
C     ***** BOUNDARY ANNULUS AS A LINEAR FUNCTION OF HEIGHT *****
  865 CALL DIFV(XT,ORGN,V1)
      T5=VV(V1,AAREV(1,3))*SCAL1
      T8=AIN(8)
      T9=AIN(9)
      NCYCLE=.5+(T8+T5*T9)/DISP
      IF(NCYCLE)1100,1100,870
  870 T3=(2.*DISP)/(T1+T2)
C     ***** INCREASE ANNULAR THICKNESS *****
      DO 875 I=1,NCYCLE
      T4=T3*FLOAT(I)
      DO 875 J=1,129,NRESOL
  875 CALL DRAW(DP(1,J),D(1,J)*T4,D(2,J)*T4,2)
      GO TO 1100
  900 CALL PROJ(D,DP,X,XO,VIEW,1,65,NRESOL)
C     ***** PLOT HALF AN ELLIPSE *****
      CALL DRAW(DP,0.,0.,3)
      DO 905 J=NRES1,65,NRESOL
  905 CALL DRAW(DP(1,J),0.,0.,2)
      IF(DISP)930,930,910
C     ***** ACCENTUATE FRONT HALF *****
  910 DO 925 I=1,3,2
      T2=FLOAT(I-2)*DISP
      DO 915 J=1,65,NRESOL
      K=66-J
  915 CALL DRAW(DP(1,K),DISP,T2,2)
      DO 925 K=1,65,NRESOL
  925 CALL DRAW(DP(1,K),-DISP,-T2,2)
  930 IF(NSOLID)940,967,935
  935 L=NDOT
      IF(NDOT-NRESOL)938,945,940
  938 CALL RADIAL(NSOLID-1)
      GO TO 945
  940 L=NRESOL
  945 CALL PROJ(D(1,65),DP(1,65),X,XO,VIEW,1,65,L)
      IF(NSOLID)960,967,950
C     ***** DOTTED LINE ON REVERSE SIDE *****
  950 DO 958 J=65,129,NDOT
      CALL DRAW(DP(1,J),DISP,DISP,3)
      DO 955 I=1,3,2
      T1=FLOAT(I-2)*DISP
      DO 955 K=1,3,2
      T2=FLOAT(K-2)*DISP
      CALL DRAW(DP(1,J),T1,T2,2)
      IF(DISP)958,958,955
  955 CONTINUE
  958 CONTINUE
      GO TO 967
C     ***** SINGLE LINE ON REVERSE SIDE *****
  960 DO 965 J=65,129,NRESOL
  965 CALL DRAW(DP(1,J),0.,0.,2)
C     ***** DETAIL INTERIOR FEATURES *****
  967 T2=NDASH*2
      DO 975 J=1,3
      T1=PAC(J,II0)*RMS(II0)*SCL
      DA(J,1)=T1
      DA(J,2)=PAC(J,II1)*RMS(II1)*SCL
      DA(J,3)=0.
      IF(NDASH)975,975,970
  970 V1(J)=-T1
      V2(J)=T1/T2
  975 CONTINUE
      IF(NDASH)987,987,980
C     ***** DASHED LINE FOR REVERSE AXIS *****
  980 DO 985 J=1,NDASH
      DO 985 K=1,2
      L=4-K
      CALL PROJ(V1,DP,X,XO,VIEW,1,1,1)
      CALL DRAW(DP,0.,0.,L)
      DO 985 I=1,3
  985 V1(I)=V1(I)+V2(I)
C     ***** SOLID LINE FOR FORWARD AXIS *****
  987 IF(LINES)1100,1100,988
  988 CALL PROJ(DA,DP,X,XO,VIEW,1,3,1)
      T1=DISP*.5
      DO 990 I=1,3,2
      T2=FLOAT(2-I)*T1
      CALL DRAW(DP,T1,T2,3)
      CALL DRAW(DP(1,3),T1,T2,2)
      IF(DISP)1000,1000,989
  989 CALL DRAW(DP(1,3),-T1,T2,2)
  990 CALL DRAW(DP,-T1,T2,2)
C     ***** SHADE QUADRANT BETWEEN TWO PRINCIPAL AXES *****
 1000 L=LINES-1
      IF(L)1100,1100,1005
 1005 T2=LINES
      DO 1025 I=1,L
      T1=FLOAT(I)/T2
      T3=SQRT(1.-T1*T1)
      IF(MOD(I,2))1010,1015,1010
 1010 M=I*2
      N=M-1
```

```
      GO TO 1020
 1015 N=I*2
      M=N-1
 1020 DO 1025 J=1,3
      T4=DA(J,1)*T1
      D(J,M)=T4
 1025 D(J,N)=DA(J,2)*T3+T4
      L=L*2
      CALL PROJ(D,DP,X,XO,VIEW,1,L,1)
      DO 1030 I=2,L,2
      CALL DRAW(DP(1,I-1),0.,0.,3)
 1030 CALL DRAW(DP(1,I),0.,0.,2)
 1100 CONTINUE
 1105 CONTINUE
C     ***** ELIMINATE LOCAL OVERLAP INFORMATION BEFORE RETURNING *****
      CALL LAP500(-1)
      RETURN
      END
      SUBROUTINE F800
C     ***** SUBROUTINE FINDS ATOM PAIRS FOR BONDS *****
      DIMENSION IA(3),W1(6)
      REAL*8 TD1,TD2,TD3,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      D100K=100000.
c *** old
c     NJ4=MOD(NJ2,10)-4
c *** new
      IAN=AIN(2)
      NJ4=(MOD(NJ2,10)-4)+(IAN*2)
      if (nj.eq.10) nj4=(ian*2)-2
      LNS=-4
      IF(MOD(NJ2,10)-2)805,848,848
C     ***** EXPLICIT DESCRIPTION *****
  805 II=0
      IF(NCD)810,810,815
  810 NG=11
      CALL ERPNT(0.D0,NJ*100+NJ2)
      GO TO 980
  815 II=II+1
      IF(140-II)980,980,820
  820 TD1=AIN(II)
      IF(TD1)815,815,825
  825 II=II+1
      TD2=AIN(II)
      IF(TD2)815,815,830
  830 IF(NJ2-10)832,838,838
  832 LNS=MOD(LNS+4,56)
      IF(LNS)838,834,838
  834 IF (NOUT.GE.0)
     &WRITE (NOUT,835)(TITLE(I),I=1,18)
  835 FORMAT(1H1,10X,18A4)
      IF (NOUT.GE.0)
     &WRITE (NOUT,837)
  837 FORMAT(1H010X,18HSYMBOL   ATOM CODE6X,16HPLOTTER X,Y(IN.) 6X,21HCA
     1RTESIAN X,Y,Z (IN.)17X,20HCRYSTAL SYSTEM X,Y,Z/1H )
C     ***** CHECK IF BOND ATOMS ARE IN ATOMS LIST (FOR OVERLAP CALC) ***
  838 NA1=0
      NA2=0
      IF(LATM-2)845,839,839
  839 N2=2
      DO 844 K=1,LATM
      TD3=ATOMID(K)
      IF(TD3-TD1)841,840,841
  840 NA1=K
      GO TO 843
  841 IF(TD3-TD2)844,842,844
  842 NA2=K
  843 N2=N2-1
      IF(N2)845,845,844
  844 CONTINUE
  845 IF(NA2-NA1)846,847,847
  846 NA3=NA1
      NA1=NA2
      NA2=NA3
      TD3=TD1
      TD1=TD2
      TD2=TD3
  847 CALL BOND(TD1,TD2,1,NA1,NA2)
      GO TO 815
C     ***** IMPLICIT DESCRIPTION *****
  848 IF(LATM-2)810,850,850
  850 SCAL3=SCAL1
      SCAL1=1.
      DO 855 I=1,LATM
  855 CALL XYZ(ATOMID(I),ATOMS(1,I),2)
      SCAL1=SCAL3
      IF(NCD)810,810,860
  860 IF (NOUT.GE.0)
     &WRITE (NOUT,861)
  861 FORMAT(1H010X,20HBOND SELECTION CODES//11X,94H(SEQUENCE(A))(SEQUEN
     1CE(B)) (BOND) (DISTANCES)( BOND )(PERSP.--LABELS)(NORMAL--LABELS)(
     2DIGITS) /11X,93H(  MIN  MAX )(  MIN  MAX ) (TYPE) (MIN   MAX)(RADI
     3US)(HEIGHT  OFFSET)(HEIGHT  OFFSET)(NUMBER))
      DMAX=0.
      DO 870 I=1,NCD
      IF(DMAX-CD(2,I))865,866,866
  865 DMAX=CD(2,I)
  866 IF (NOUT.GE.0)
     &WRITE (NOUT,871)(KD(J,I),J=1,5),(CD(J,I),J=1,8)
  870 CONTINUE
  871 FORMAT(1H ,10X,I6,I5,I8,I5,I8,2F6.2,5F8.3,F7.0)
      DMAX=DMAX*DMAX
C     ***** LOOP THROUGH ATOMS ARRAY *****
      DO 977 M=1,LATM
      NA1=M
      TD1=ATOMID(M)
      MI=TD1/D100K
```

```
      IF(NJ4)8722,8724,8718
 8718 IF(NJ4-2)8724,8726,8720
 8720 IF(NJ4-4)8726,8722,8722
 8722 IA(1)=TD1/D100K
      GO TO 8728
 8724 IA(1)=IDENT(1,MI)
      GO TO 8728
 8726 IA(1)=IDENT(2,MI)
 8728 IA(3)=IA(1)
      W1(1)=ATOMS(1,M)
      W1(2)=ATOMS(2,M)
      W1(3)=ATOMS(3,M)
      L=M+1
      IF(LATM-L)977,872,872
  872 DO 975 N=L,LATM
      NA2=N
      DIST=(ATOMS(1,N)-W1(1))**2
      IF(DMAX-DIST)975,873,873
  873 DIST=DIST+(ATOMS(2,N)-W1(2))**2
      IF(DMAX-DIST)975,874,874
  874 DIST=DIST+(ATOMS(3,N)-W1(3))**2
      IF(DMAX-DIST)975,875,875
  875 DIST=SQRT(DIST)
      TD2=ATOMID(N)
      NI=TD2/D100K
C     IF(NJ4)876,877,878
c 876 IA(2)=TD2/D100K
c     GO TO 879
c 877 IA(2)=IDENT(1,NI)
c     GO TO 879
c 878 IA(2)=IDENT(2,NI)
      IF(NJ4.LT.0) IA(2)=TD2/D100K
      IF(NJ4.EQ.0.OR.NJ4.EQ.1) IA(2)=IDENT(1,NI)
      IF(NJ4.GT.1) IA(2)=IDENT(2,NI)
C     ***** SELECT BONDS ACCORDING TO CODES *****
  879 DO 950 J=1,NCD
      JB=J
      IF(DIST-CD(1,J))   950,880,880
  880 IF(CD(2,J)-DIST)   950,881,881
  881 DO 885 K=1,2
      IF(IA(K)-KD(1,J))  885,882,882
  882 IF(KD(2,J)-IA(K))  885,883,883
  883 IF(IA(K+1)-KD(3,J))885,884,884
  884 IF(KD(4,J)-IA(K+1))885,890,890
  885 CONTINUE
      GO TO 950
C     ***** CHECK FOR POLYHEDRA CODE *****
  890 IF(CD(4,J))900,955,955
  900 W1(4)=ATOMS(1,N)
      W1(5)=ATOMS(2,N)
      W1(6)=ATOMS(3,N)
      KM1=ABS(CD(4,J))
      KM2=ABS(CD(5,J))
      DSQ1=(CD(6,J))**2
      DSQ2=(CD(7,J))**2
C     ***** SEARCH FOR POLYHEDRA CENTER *****
      DO 935 IM=1,LATM
      K3=ATOMID(IM)/D100K
      if (ian.eq.1) k3=ident(1,im)
      if (ian.eq.2) k3=ident(2,im)
      IF(K3-KM1)935,905,905
  905 IF(KM2-K3)935,910,910
C     ***** CHECK POLYHEDRA DISTANCE RANGE *****
  910 DO 930 J1=1,4,3
      DSQ=(ATOMS(1,IM)-W1(J1))**2
      IF(DSQ2-DSQ)935,915,915
  915 DSQ=DSQ+(ATOMS(2,IM)-W1(J1+1))**2
      IF(DSQ2-DSQ)935,920,920
  920 DSQ=DSQ+(ATOMS(3,IM)-W1(J1+2))**2
      IF(DSQ2-DSQ)935,925,925
  925 IF(DSQ-DSQ1)935,930,930
  930 CONTINUE
      GO TO 955
  935 CONTINUE
C     ***** END OF POLYHEDRA CHECK *****
  950 CONTINUE
      GO TO 975
C     ***** PREPARE TO DRAW BOND *****
  955 IF(NJ2-10)960,970,970
  960 LNS=MOD(LNS+4,56)
      IF(LNS)965,965,970
  965 IF (NOUT.GE.0)
     &WRITE (NOUT,835)(TITLE(I),I=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,837)
  970 CALL BOND(TD1,TD2,JB,NA1,NA2)
  975 CONTINUE
  977 CONTINUE
C     ***** ELIMINATE LOCAL OVERLAP INFORMATION BEFORE RETURNING *****
  980 IF(NJ2-21)985,990,990
  985 CALL LAP500(-1)
  990 if (nj2.eq.22) then
      IF(NQUAD)993,993,991
C     ***** PRINT OUT NUMBER OF BOND QUADRANGLES STORED *****
C     ***** PRINT OUT QUADRANGLE IDENTIFICATION ARRAY *****
  991 IF (NOUT.GE.0)
     &WRITE (NOUT,992)NQUAD,(QUAD(9,J),J=1,NQUAD)
  992 FORMAT(1H0,10X,27HBOND OVERLAP ARRAY CONTAINS,I4,23H BONDS (MAXIMU
     1M IS 599)/  11X,  66HATOM-PAIR NUMBERS IN ARRAY REFER TO SEQUENCE
     2IN SORTED ATOMS ARRAY/(15X,10F10.0))
      end if
  993 RETURN
      END
      SUBROUTINE F900
      DIMENSION X(3),XW(3,5),Y(3),Z(3)
      REAL*8 D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      character*72 tmpti, tmpti2
C     ***** LABELING FUNCTION SUBROUTINE *****
```

```
      D100K=100000.
      ITILT=0
      NJ3=MOD(NJ2,10)
      TH=THETA
      SINTH=SYMB(2,1)
      COSTH=SYMB(1,1)
      ILAST=1
      T2=AIN(2)
      IF(T2-11100.)910,910,905
  905 ILAST=2
  910 DO 925 II=1,ILAST
C     ***** OBTAIN WORKING CARTESIAN COORDINATES *****
      CALL XYZ(AIN(II),XW(1,II),2)
      IF(NG)915,925,915
  925 CALL XYZ(AIN(II),XW(1,II+3),3)
      II=1
C     ***** FIND MEAN REFERENCE POINT *****
      DO 930 J=1,3
      T2=XW(J,ILAST)
      T1=XW(J,1)
      XW(J,3)=T2-T1
  930 X(J)=(T2+T1)*.5
C     ***** PERSPECTIVE SCALING FACTOR *****
      SCAL=1.
      IF(VIEW)940,940,935
  935 SCAL=VIEW/(VIEW-X(3))
  940 T1=AIN(5)
      HGT=SCAL*T1
      IF(NJ2-3)960,950,945
  945 IF(NJ2-6)950,950,960
C     ***** PROJECTED VECTOR BASELINE *****
  950 CALL PLTXY(XW(1,4),V1)
      CALL PLTXY(XW(1,5),V2)
      T1=V2(1)-V1(1)
      T2=V2(2)-V1(2)
      T3=SQRT(T1*T1+T2*T2)
      IF(T3)912,912,955
  955 COSTH=T1/T3
      SINTH=T2/T3
      TH=ARCCOS(COSTH)
      IF(SINTH)958,960,960
  958 TH=-TH
  960 IF(NJ2-13)965,985,985
C     ***** FIND CENTER OF PROJECTED LABEL *****
  965 T6=AIN(6)
      T7=AIN(7)
      Y(1)=SCAL*(X(1)+T6*COSTH-T7*SINTH)+XO(1)
      Y(2)=SCAL*(X(2)+T6*SINTH+T7*COSTH)+XO(2)
      Y(3)=0.
C     ***** CHECK FOR LEGEND RESET *****
      DO 980 J=1,2
      T1=AIN(J+2)
      IF(T1)975,980,970
  970 Y(J)=T1
      GO TO 980
  975 Y(J)=XLNG(J)+T1
  980 CONTINUE
C     ***** SET PARAMETERS FOR INDIVIDUAL FUNCTIONS *****
  985 GO TO(990,995,995,1000,1000,1000,915,1105,1105,915,915,915,1005,10
     105,1005,1005,915),NJ2
  990 T6=17.
      L=AIN(1)/D100K
      ILAST=1
      DXW=0.
      DYW=0.
      GO TO 1030
  995 T6=215.
      ILAST=18
      T1=HGT*24./7.
      DXW=COSTH*T1
      DYW=SINTH*T1
      GO TO 1030
 1000 T6=10+3*(NJ3-4)
      DIST=SQRT(VV(XW(1,3),XW(1,3)))/SCAL1
      GO TO 1030
C     ***** TRUE PERSPECTIVE LABELS *****
 1005 CALL UNITY(XW(1,3),VT(1,1),1)
      IF(ABS(VT(3,1))-.9994)1010,912,912
C     ***** FORM PERSPECTIVE ROTATION MATRIX *****
 1010 CALL NORM(AID(1,3),VT(1,1),VT(1,2),1)
      CALL UNITY(VT(1,2),VT(1,2),1)
      CALL NORM(VT(1,1),VT(1,2),VT(1,3),1)
      DO 1015 J=1,3
 1015 VT(J,4)=X(J)
      ITILT=1
      HGT=AIN(5)
      TH=0.
      Y(3)=X(3)
      TT7=AIN(7)
      Y(2)=X(2)+TT7-HGT*.5
      TT6=AIN(6)
      IF(NJ2-13)1030,1025,1020
C     ***** PERSPECTIVE BOND LABELS *****
 1020 Y(1)=X(1)+TT6-HGT*FLOAT(22+3*(6-NJ3))/7.
      DIST=SQRT(VV(XW(1,3),XW(1,3)))/SCAL1
      GO TO 1050
C     ***** PERSPECTIVE TITLES *****
 1025 Y(1)=X(1)+TT6-HGT*215./7.
      ILAST=18
      DXW=HGT*24./7.
      DYW=0.
      GO TO 1050
 1030 DH=HGT*T6/7.
      DV=HGT*.5
      Y(1)=Y(1)-DH*COSTH+DV*SINTH
      Y(2)=Y(2)-DH*SINTH-DV*COSTH
      Y(3)=0.
C     ***** PLOT VARIOUS LABELS *****
 1050 Z(3)=Y(3)
      XO(3)=Y(3)
      GO TO(1060,1060,1060,1090,1090,1090,915,1105,1105),NJ3
 1060 if (nj3 .eq. 1) go to 1061
c *** if title begins in column 1, center it
      if (title2(1)(1:1) .ne. ' ') then
         do 101 i=1,72
            tmpti(i:i) = ' '
            tmpti2(i:i) = ' '
  101    continue
         do 102 i=1,18
            tmpti(i*4-3:i*4)=title2(i)
  102    continue
         do 103 i=72,1,-1
```

```
            if (tmpti(i:i) .ne. ' ') then
               klast = i
               go to 104
            end if
  103    continue
  104    ioffset = (72 - klast) / 2
         do 105 i=1,klast
            tmpti2(i+ioffset:i+ioffset) = tmpti(i:i)
  105    continue
         do 106 i=1,18
            title2(i) = tmpti2(i*4-3:i*4)
  106    continue
       end if
 1061 DO 1085 I=1,ILAST
      DO 1075 J=1,3,2
      Z(1)=Y(1)+FLOAT(J-2)*DISP*.5
      DO 1075 K=1,3,2
      Z(2)=Y(2)+FLOAT(K-2)*DISP*.5
      IF(NJ3-2)1065,1068,1068
C     ***** PLOT CHEMICAL SYMBOL *****
 1065 CALL SIMBOL(Z(1),Z(2),HGT,CHEM(L),TH,6)
      GO TO 1070
C     ***** PLOT TITLES *****
 1068 CALL SIMBOL(Z(1),Z(2),HGT,TITLE2(I),TH,4)
 1070 IF(DISP)1080,1080,1075
 1075 CONTINUE
 1080 Y(1)=Y(1)+DXW
 1085 Y(2)=Y(2)+DYW
      GO TO 1199
C     ***** PLOT BOND DISTANCE LABELS *****
 1090 I9=NJ3-3
      T9=10.**I9
      DISTR=AINT((DIST*T9)+0.5)/T9 +.0001
      CALL NUMBUR(Y(1),Y(2),HGT,DISTR,TH,I9)
      GO TO 1199
C     ***** PLOT CENTERED SYMBOLS *****
 1105 TT8=AIN(8)
c *** ORTEP-II call
c     CALL SIMBOL(Y(1),Y(2),HGT,IFIX(TT8),TH,7-NJ3)
c *** Only one centered symbol (*) is available in ORTEP-III.
c *** It is triggered by the negative value for argument 6.
c *** Argument 4 is ignored by SIMBOL.
      CALL SIMBOL(Y(1),Y(2),HGT,' ',TH,7-NJ3)
      GO TO 1199
  912 NG=15
  915 CALL ERPNT(AIN(II),NJ*100+NJ2)
 1199 ITILT=0
      RETURN
      END
      SUBROUTINE F1000
c *** 1001 identical to 511
      CALL LAP500(1)
      RETURN
      END
      function iend(string)
c *** returns position of last non-space character in string
      character string*(*)
      do 800 i=len(string),1,-1
         if (string(i:i) .ne. ' ') then
            iend = i
            return
         end if
  800 continue
      iend = 1
      return
      end
      SUBROUTINE LAP500(NTYPE)
C     ***** STORE PROJECTED ATOM CONICS AND BOND QUADRANGLES *****
      DIMENSION QC(3,3),QD(3,3),VD1(3),VD2(3)
      REAL*8 QC,QD,VD1,VD2,TD1,TD2,TD3,TD
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
C     ***** ELIMINATE ALL PREVIOUSLY STORED LOCAL OVERLAP INFORMATION **
      NCOVER=0
      NQOVER=0
      IF(NTYPE)420,195,195
C     ***** ELIMINATE ALL PREVIOUSLY STORED GLOBAL OVERLAP INFORMATION *
  195 NCONIC=0
      NQUAD=0
      IF(NTYPE)420,420,200
C     ***** CONSTANT FOR OVERLAP MARGIN (WHITE MARGIN AT OVERLAP) *****
  200 IF(AIN(1))205,215,210
C     ***** NEGATIVE NUMBER OR POSITIVE INTEGER GIVES OVMRGN=0.0 *****
  205 OVMRGN=0.0
      GO TO 220
C     ***** SET OVERLAP MARGIN WIDTH DIRECTLY IN INCHES *****
  210 OVMRGN=AIN(1)-DINT(AIN(1))
      GO TO 220
C      ***** DEFAULT OPTION, OVERLAP MARGIN WIDTH AS A FUNCTION OF SCAL1
  215 if (scal1.lt..25) then
         OVMRGN=AMAX1(SQRT(SCAL1)*0.050,0.010)
      else
         OVMRGN=AMAX1(SQRT(SCAL1)*0.030,0.025)
      end if
  220 IF (NOUT.GE.0)
     &WRITE (NOUT,2) OVMRGN
    2 FORMAT(1H0,10X,17HOVERLAP MARGIN IS, F6.3,5H INCH)
  225 IF(LATM)230,230,235
  230 NG=12
      CALL ERPNT(0.D0,510+NJ2)
      GO TO 420
C     ***** SORT ATOMS LIST BY -VIEWDISTANCE OR BY Z PARAMETER
  235 IF(VIEW)250,250,240
C     ***** CALCULATE VIEWDISTANCES**2 *(-1) IF VIEW.GT.ZERO *****
  240 DO 245 I=1,LATM
      CALL XYZ(ATOMID(I),V3,2)
      V3(3)=V3(3)-VIEW
  245 ATOMS(3,I)=-VV(V3,V3)
      GO TO 260
C     ***** STORE CARTESIAN COORDINATES IF VIEW.EQ.ZERO *****
  250 DO 255 I=1,LATM
  255 CALL XYZ(ATOMID(I),ATOMS(1,I),2)
```

```
C     ***** SORTING PROCEDURE BY SHELL, COMM ACM 2,30 (1959) *****
  260 M=LATM
  265 M=M/2
      IF(M)300,300,270
  270 K=LATM-M
      J=1
  275 I=J
  280 IM=I+M
      IF(ATOMS(3,I)-ATOMS(3,IM))295,295,285
  285 TD=ATOMID(I)
      ATOMID(I)=ATOMID(IM)
      ATOMID(IM)=TD
      T1=ATOMS(3,I)
      ATOMS(3,I)=ATOMS(3,IM)
      ATOMS(3,IM)=T1
      I=I-M
      IF(I)295,295,280
  295 J=J+1
      IF(J-K)275,275,265
C     ***** LOOP THROUGH ALL ATOMS IN SORTED ATOMS LIST *****
  300 DO 405 IA=1,LATM
      CALL XYZ(ATOMID(IA),ATOMS(1,IA),2)
      CALL PAXES(ATOMID(IA),2)
      DO 305 J=1,3
      V1(J)=ATOMS(J,IA)
      VD1(J)=V1(J)
      DO 305 K=1,3
  305 QD(J,K)=Q(J,K)
      IF(VIEW)340,340,310
C     ***** CALCULATE ENVELOPING CONE WITH ORIGIN AT VIEWPOINT *****
  310 V1(3)=V1(3)-VIEW
      VD1(3)=V1(3)
C     ***** FORM COFACTOR MATRIX *****
      DO 315 J=1,3
      J1=MOD(J,3)+1
      J2=MOD(J+1,3)+1
      DO 315 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
      QC(J,K)= QD(J1,K1)*QD(J2,K2)-QD(J1,K2)*QD(J2,K1)
  315 QC(K,J)=QC(J,K)
C     ***** FORM POLARIZED COFACTOR MATRIX AND ADD TO ELLIPSOID MATRIX *
      TD2=-SCL**2
C     ***** TD1 IS AN ARBITRARY SCALING FACTOR *****
      TD1=VMV(V1,Q,V1)
      DO 325 J=1,3
      J1=MOD(J,3)+1
      J2=MOD(J+1,3)+1
      DO 320 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
      QD(J,K)=((VD1(J2)*(QC(J1,K1)*VD1(K2)-QC(J1,K2)*VD1(K1))
     1 +VD1(J1)*(VD1(K1)*QC(J2,K2)-VD1(K2)*QC(J2,K1)))+TD2*QD(J,K))/TD1
  320 QD(K,J)=QD(J,K)
C     ***** PROJECTED ELLIPSE IN HOMOGENEOUS COORD OF WORKING SYSTEM ***
      QD(J,3)=-QD(J,3)*VIEW
  325 QD(3,J)=-QD(3,J)*VIEW
C     ***** PROJECT CENTER OF ATOM ONTO PROJECTION PLANE *****
      TD1=-VIEW/VD1(3)
      VD2(1)=VD1(1)*TD1
      VD2(2)=VD1(2)*TD1
C     ***** TRANSFORM TO NEW ORIGIN TO IMPROVE CONDITION OF MATRIX Q ***
      DO 330 J=1,3
      DO 330 K=1,2
  330 QD(J,3)=QD(J,3)+QD(J,K)*VD2(K)
      DO 335 J=1,3
      DO 335 K=1,2
  335 QD(3,J)=QD(3,J)+VD2(K)*QD(K,J)
      V6(1)=XO(1)+VD2(1)
      V6(2)=XO(2)+VD2(2)
      GO TO 355
C     ***** CALCULATE ENVELOPING CYLINDER ALONG Z OF WORKING SYSTEM ****
  340 DO 345 J=1,2
      DO 345 K=1,2
  345 QD(J,K)=QD(J,K)-QD(J,3)*QD(K,3)/QD(3,3)
      DO 350 J=1,2
      QD(J,3)=0.0
      QD(3,J)=0.0
  350 V6(J)=XO(J)+ATOMS(J,IA)
C     ***** PROJECTED ELLIPSE IN HOMOGENEOUS COORD ABOUT CENTER OF ATOM
      QD(3,3)=-SCL**2
C     ***** FIT RECTANGLE AROUND ELLIPSE ALLOWING OVERLAP MARGIN *****
C     ***** FORM MATRIX OF COFACTORS *****
  355 DO 360 J=1,3
      J1=MOD(J,3)+1
      J2=MOD(J+1,3)+1
      DO 360 K=J,3
      K1=MOD(K,3)+1
      K2=MOD(K+1,3)+1
  360 QC(J,K)= QD(J1,K1)*QD(J2,K2)-QD(J1,K2)*QD(J2,K1)
C     ***** RESCALE MATRIX OF COFACTORS SO THAT QC(3,3)=1.0 *****
      DO 365 J=1,3
      DO 365 K=J,3
      QC(J,K)= QC(J,K)/QC(3,3)
  365 QC(K,J)=QC(J,K)
      TD2=QD(3,3)
      NDG=0
      DO 385 J=1,2
C     ***** SOLVE QUADRATIC EQUATION *****
      T1 =QC(3,J)**2-QC(J,J)
      IF(T1)370,370,375
C     ***** ROUNDOFF PROBLEMS, RESET LIMITS IN X OR Y *****
  370 NDG=1
      V5(J)=0.001+OVMRGN
      GO TO 380
  375 V5(J)=  SQRT(T1)+OVMRGN
      V6(J)=V6(J)+QC(3,J)
      TD2=TD2+QD(3,J)*QC(3,J)
  380 CONIC(2*J-1,IA)=V6(J)-V5(J)
  385 CONIC(2*J,IA)=V6(J)+V5(J)
      IF(NDG)390,390,395
  390 IF(TD2)400,395,395
C     ***** ELLIPSE IMAGINARY DUE TO ROUNDOFF, RESET TO REAL VALUE *****
  395 CONIC(5,IA)=1.0/((CONIC(2,IA)-CONIC(1,IA))*0.5)**2
      CONIC(6,IA)=0.0
      CONIC(7,IA)=1.0/((CONIC(4,IA)-CONIC(3,IA))*0.5)**2
      GO TO 405
C     ***** STORE NORMALIZED QUADRATIC COEFFICIENTS FOR ELLIPSE *****
C     ***** SCALED BY OVERLAP MARGIN PARAMETER *****
  400 TD3= -(1.0-2.0*OVMRGN/(V5(1)+V5(2)))**2 /TD2
      CONIC(5,IA)=QD(1,1)*TD3
      CONIC(6,IA)=QD(1,2)*TD3
```

```
      CONIC(7,IA)=QD(2,2)*TD3
  405 CONTINUE
      NCONIC=LATM
C     ***** PRINT OUT SORTED ATOMS ARRAY *****
      IF (NOUT.GE.0)
     &WRITE (NOUT,4)(ATOMID(J),J=1,LATM)
    4 FORMAT(1H0,10X,30HCONTENTS OF SORTED ATOMS ARRAY/(15X,10F10.0))
C     ***** STORE BOND QUADRANGLES IF SEARCH CODES ARE GIVEN *****
      IF(NCD)420,420,410
C     ***** GENERATE PSEUDO-INSTRUCTION 822 TO CALCULATE BONDS *****
  410 NJ2=22
      CALL F800
c *** the lines below have been moved to the end of F800
c       IF(NQUAD)420,420,415
c C     ***** PRINT OUT NUMBER OF BOND QUADRANGLES STORED *****
c C     ***** PRINT OUT QUADRANGLE IDENTIFICATION ARRAY *****
c   415 IF (NOUT.GE.0)
c      &WRITE (NOUT,6)NQUAD,(QUAD(9,J),J=1,NQUAD)
c     6 FORMAT(1H0,10X,27HBOND OVERLAP ARRAY CONTAINS,I4,23H BONDS (MAXIMU
c      1M IS 599)/  11X,  66HATOM-PAIR NUMBERS IN ARRAY REFER TO SEQUENCE
c      2IN SORTED ATOMS ARRAY/(15X,10F10.0))
c *** the lines above have been moved to the end of F800
  420 RETURN
      END
      SUBROUTINE LAP700(NA,ICQ)
      DIMENSION DETER(2),QA(3,3,2),QC(3,3,2),V12(3,2),YMIN(2),YMAX(2)
      DIMENSION OVMR(2)
      REAL*8 AOV3,AOV3SQ,BOV3,DETER,PI,PHI,POV3,POV3CU,QA,QC,QOV2,QOV2SQ
      REAL*8 ROOT,TD,TIDD
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      PI=3.1415926535897932
      ICQ=0
      NCOVER=0
      NQOVER=0
      OVMR(1)=OVMRGN
      OVMR(2)=0.0
      IF(NCONIC-NA)200,200,205
  200 RETURN
C     ***** ROUGH CHECK FOR OVERLAPPING ATOMS *****
  205 DO  210 J=1,2
      YMIN(J)=CONIC(2*J-1,NA)
  210 YMAX(J)=CONIC(2*J,NA)
      L=0
      DO 420 IA=NA,NCONIC
      IF(IA-NA)230,230,215
  215 DO 225 J=1,2
      IF(YMAX(J)-CONIC(2*J-1,IA))420,420,220
  220 IF(YMIN(J)-CONIC(2*J,IA))225,420,420
  225 CONTINUE
C     ***** EXACT CHECK FOR OVERLAPPING ATOMS *****
  230 IF(L-1)235,235,240
  235 L=L+1
  240 CALL LAPCON(CONIC(1,IA),DA,V12(1,L),OVMR(L))
      DO 245 J=1,3
      DO 245 K=1,3
  245 QA(J,K,L)=DA(J,K)
C     ***** CALCULATE COFACTORS AND DETERMINANTS *****
      DETER(L)=0.0
      DO 250 J=1,3
      J1=MOD(J+3,3)+1
      J2=MOD(J+1,3)+1
      DO 250 K=1,3
      K1=MOD(K+3,3)+1
      K2=MOD(K+1,3)+1
      TD=QA(J1,K1,L)*QA(J2,K2,L)-QA(J1,K2,L)*QA(J2,K1,L)
      DETER(L)=DETER(L)+TD*QA(J,K,L)
  250 QC(J,K,L)=TD
C     ***** DETER(L) IS THE DETERMINANT TIMES 3 *****
      IF(L-1)420,420,255
C     ***** FORM CHARACTERISTIC EQUATION AND EXAMINE ITS ROOTS *****
  255 AOV3=0.0
      BOV3=0.0
      DO 260 J=1,3
      DO 260 K=1,3
      AOV3=AOV3+QC(J,K,2)*QA(J,K,1)
  260 BOV3=BOV3+QC(J,K,1)*QA(J,K,2)
      AOV3=AOV3/ DETER(2)
      AOV3SQ=AOV3**2
      BOV3=BOV3/ DETER(2)
      POV3=BOV3-AOV3SQ
      QOV2=AOV3*(AOV3SQ-BOV3*1.5D0)+DETER(1)/(DETER(2)*2.0D0)
C     ***** CHECK DISCRIMINANT OF CHARACTERISTIC CUBIC EQUATION *****
      ITYPE=0
      POV3CU=POV3**3
      QOV2SQ=QOV2**2
      IF(POV3CU+QOV2SQ)270,310,265
  265 IF(POV3CU*1.00001 +QOV2SQ)310,310,400
  270 IF(POV3CU+1.00001 *QOV2SQ)275,310,310
C     ***** THREE REAL ROOTS, ALL DIFFERENT *****
  275 ITYPE=1
C     ***** NO INTERSECTION IF A/3 AND B/3 INVARIANTS ARE NEGATIVE *****
      IF(AOV3)280,285,285
  280 IF(BOV3)420,285,285
C     ***** CALCULATE ONE ROOT OF CHARACTERISTIC CUBIC EQUATION *****
  285 IF(QOV2)295,290,295
  290 PHI=PI/2.0D0
      GO TO 305
  295 PHI=DATAN(-DSQRT(-POV3CU-QOV2SQ)/QOV2)
      IF(PHI)300,305,305
  300 PHI=PHI+PI
  305 ROOT=2.0D0*DSQRT(-POV3)*DCOS(PHI/3.0D0)-AOV3
      GO TO 325
C     ***** THREE REAL ROOTS, AT LEAST TWO ARE EQUAL *****
  310 ITYPE=2
C     ***** CHECK SIGNS OF INVARIANTS A/3 AND B/3 *****
      IF(AOV3)315,320,320
  315 IF(BOV3)420,320,320
C     ***** CALCULATE REPEATED ROOT OF CUBIC EQUATION *****
  320 ROOT=DSIGN(DSQRT(-POV3),QOV2)-AOV3
C     ***** FORM DEGENERATE CONIC (LINE PAIR WHICH MAY BE COINCIDENT) **
  325 DO 330 J=1,3
```

```
      DO 330 K=1,3
  330 DA(J,K)=QA(J,K,1)+ROOT*QA(J,K,2)
C     ***** EXAMINE INVARIANTS OF THE DEGENERATE CONIC *****
      T6=DA(1,1)*DA(2,2)
      T7=DA(1,2)**2
C     ***** NEGATIVE DENOTES REAL INTERSECTING LINE PAIR *****
C     ***** POSITIVE DENOTES IMAGINARY LINES INTERSECTING AT REAL POINT
      IF(T6-T7)335,345,340
  335 IF(T6*1.0001  -T7)400,345,345
  340 IF(T6-1.0001  *T7)345,345,365
  345 T8=DA(3,3)*(DA(1,1)+DA(2,2))
      T9=DA(1,3)**2+DA(2,3)**2
C     ***** NEGATIVE DENOTES REAL PARALLEL LINE PAIR *****
C     ***** POSITIVE DENOTES IMAGINARY PARALLELS *****
C     ***** ZERO DENOTES ONE REAL LINE (COINCIDENT PARALLELS) *****
      IF(T8-T9)350,360,355
  350 IF(T8*1.0001  -T9)400,360,360
  355 IF(T8-1.0001  *T9)360,360,365
C     ***** COINCIDENT LINE PAIR FOUND FOR THE REPEATED ROOT *****
  360 ITYPE=3
C     ***** COMPARE AREAS OF CONICS *****
  365 KA=1
      KB=2
      IF(QC(3,3,KA)-QC(3,3,KB))370,375,375
  370 KA=2
      KB=1
C     ***** SEE IF ONE CONIC IS INSIDE THE OTHER CONIC *****
  375 T1=0.0
      DO 385 J=1,3
      T2=QA(J,3,KB)
      DO 380 K=1,2
  380 T2=T2+QA(J,K,KB)*V12(K,KA)
  385 T1=T1+V12(J,KA)*T2
C     ***** DISCARD IF KA IS OUTSIDE KB *****
      IF(T1)390,390,420
  390 IF(KA-1)395,395,400
C     ***** THE OVERLAPPING ATOM HIDES THE ORIGINAL ATOM *****
  395 ICQ=-1
      RETURN
C     ***** STORE OVERLAPPING ATOM *****
  400 ICQ=ICQ+1
      IF(NCOVER-20)410,405,405
  405 NG=17
      CALL ERPNT(ATOMID(IA),700)
      NCOVER=NCOVER-1
  410 NCOVER=NCOVER+1
      IJ=1
      DO 415 I=1,3
      DO 415 J=I,3
      COVER(IJ,NCOVER)=QA(I,J,2)
  415 IJ=IJ+1
      KC(NCOVER)=IA
  420 CONTINUE
C     ***** SECOND PART OF SUBROUTINE CHECKS FOR BONDS OVER THE ATOM ***
  425 IF(NQUAD)470,470,430
  430 ITY=0
C     ***** ROUGH CHECK FOR OVERLAPPING BONDS *****
      DO 465 IQ=1,NQUAD
      TID=QUAD(9,IQ)
      TIDD=TID
      NA1=TID/1000.
      NA2=AMOD(TID,1000.)
      IF(NA-NA2)435,435,465
  435 DO 445 J=1,2
      IF(YMAX(J)-AMIN1(QUAD(J,IQ),QUAD(J+2,IQ),QUAD(J+4,IQ),QUAD(J+6,IQ)
     1 ))465,465,440
  440 IF(YMIN(J)-AMAX1(QUAD(J,IQ),QUAD(J+2,IQ),QUAD(J+4,IQ),QUAD(J+6,IQ)
     1 ))445,465,465
  445 CONTINUE
C     ***** EXACT CHECK FOR OVERLAPPING BONDS *****
  450 ITY=ITY-1
      IQQ=0
      IQR=IQ
      CALL LAPAB(IQR,NA,IQQ,ITY)
      IF(IQQ)455,460,460
  455 ICQ=-1
      RETURN
  460 ICQ=ICQ+1
      IF(NQOVER-30)465,470,470
  465 CONTINUE
  470 RETURN
      END
      SUBROUTINE LAP800(NA1,NA2,ICQ)
C     ***** SUBROUTINE CHECKS FOR ATOMS AND BONDS OVERLAPPING A BOND ***
      DIMENSION FL(4,4),Y1(2),Y2(2),YMAX(2),YMIN(2),QUA(3,4)
      DIMENSION VUE(3)
      REAL*8 TIDD
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      IQ=0
      ICQ=0
      IF(NA1*NA2)245,245,195
  195 TID1=FLOAT(NA1)*1000.+FLOAT(NA2)
      IF(NCONIC)245,245,200
  200 IF(NJ2-21)250,205,205
C     ***** PART 1, CALLED FROM BOND, STORES BOND OUTLINE INFORMATION **
  205 IF(NQUAD-599)215,210,210
  210 NG=16
      CALL ERPNT(ATOMID(NA1),822)
      GO TO 245
  215 NQUAD=NQUAD+1
C     ***** CALCULATE OVERLAP MARGIN FOR BOND QUADRANGLE *****
      T1=0.0
      T2=0.0
      DO 220 J=1,2
      Y1(J)=DP(J,1)-DP(J,65)
      Y2(J)=DP(J,2)-DP(J,66)
      T1=T1+Y1(J)**2
  220 T2=T2+Y2(J)**2
      IF(T1*T2)225,225,230
  225 T1=0.0
      T2=0.0
```

```
      GO TO 235
  230 T1=OVMRGN/SQRT(T1)
      T2=OVMRGN/SQRT(T2)
C     ***** STORE BOND QUADRANGLE *****
  235 DO 240 J=1,2
      Y1(J)=Y1(J)*T1
      Y2(J)=Y2(J)*T2
      QUAD(J,NQUAD)=DP(J,1)+Y1(J)
      QUAD(J+2 ,NQUAD)=DP(J,2)+Y2(J)
      QUAD(J+4,NQUAD)=DP(J,66)-Y2(J)
  240 QUAD(J+6,NQUAD)=DP(J,65)-Y1(J)
      QUAD(9,NQUAD)=TID1
  245 RETURN
C     ***** PART 2, CALLED FROM BOND, OVERLAP CHECK FOR BOND NA1-NA2 ***
  250 NCOVER=0
      NQOVER=0
      TOL=1.E-5
      IF(NCONIC-NA1)245,245,255
C     ***** SAVE QUADRANGLE TEMPORARILY *****
  255 IQ=NQUAD+1
      DO 260 J=1,2
      QUAD(J,IQ)=DP(J,1)
      QUAD(J+2,IQ)=DP(J,2)
      QUAD(J+4,IQ)=DP(J,66)
  260 QUAD(J+6,IQ)=DP(J,65)
      QUAD(9,IQ)=TID1
C     ***** FIT RECTANGLE AROUND QUADRANGLE *****
  265 DO 270 J=1,2
      YMIN(J)=AMIN1(DP(J,1),DP(J,2),DP(J,66),DP(J,65))
  270 YMAX(J)=AMAX1(DP(J,1),DP(J,2),DP(J,66),DP(J,65))
C     ***** ROUGH CHECK FOR ATOM-OVER-BOND OVERLAP *****
      NA1P1=NA1+1
      ITY=0
      DO 305 IA=NA1P1,NCONIC
      DO 285 J=1,2
      IF(IA-NA2)275,305,275
  275 IF(YMAX(J)-CONIC(2*J-1,IA))305,305,280
  280 IF(YMIN(J)-CONIC(2*J,IA))285,305,305
  285 CONTINUE
C     ***** CHECK FOR TRUE ATOM-OVER-BOND OVERLAP *****
      ITY=ITY+1
      IAQ=IA
      CALL LAPAB(IQ,IAQ,IQQ,ITY)
      IF(IQQ)290,305,300
  300 ICQ=ICQ+1
      IF(NCOVER-20)305,310,310
  305 CONTINUE
  310 IF(NQUAD)295,295,315
C     ***** HIDDEN BOND *****
  290 ICQ=-1
  295 RETURN
C     ***** ROUGH CHECK FOR BOND-OVER-BOND OVERLAP *****
  315 CALL DIFV(ATOMS(1,NA2),ATOMS(1,NA1),V1)
      CALL UNITY(V1,V1,1)
      VUE(1)=  ATOMS(1,NA1)
      VUE(2)=  ATOMS(2,NA1)
      VUE(3)=  ATOMS(3,NA1)-VIEW
      DO 495 IB=1,NQUAD
      TID2=QUAD(9,IB)
      IF(TID1-TID2)320,495,320
  320 NB2=AMOD(TID2,1000.)
      NB1=TID2/1000.
      IF(NA1-NB2)325,495,495
  325 DO 335 J=1,2
      IF(YMAX(J)-AMIN1(QUAD(J,IB),QUAD(J+2,IB),QUAD(J+4,IB),QUAD(J+6,IB)
     1 ))495,495,330
  330 IF(YMIN(J)-AMAX1(QUAD(J,IB),QUAD(J+2,IB),QUAD(J+4,IB),QUAD(J+6,IB)
     1 ))335,495,495
  335 CONTINUE
C     ***** SET UP LINEAR FORMS FOR EDGES OF QUADRANGLE *****
      DO 345 L=1,4
      K=2*L
      K1=MOD(K,8)+2
      QUA(1,L)=QUAD(K,IB)-QUAD(K1,IB)
      QUA(2,L)=QUAD(K1-1,IB)-QUAD(K-1,IB)
      QUA(3,L)=QUAD(K-1,IB)*QUAD(K1,IB)-QUAD(K,IB)*QUAD(K1-1,IB)
C     ***** NORMALIZE LINE EQUATION COEFFICIENTS *****
      T1=SQRT(QUA(1,L)**2+QUA(2,L)**2)
      IF(T1)495,495,340
  340 DO 345 J=1,3
  345 QUA(J,L)=QUA(J,L)/T1
C     ***** EVALUATE LINEAR FORMS AND SIGNATURES FOR QUADRANGLE *****
      T3=3.0
      DO 365 K=1,4
      T2=3.0
      J=K*2
      DO 355 L=1,4
      T1=QUAD(J-1,IQ)*QUA(1,L)+QUAD(J,IQ)*QUA(2,L)+QUA(3,L)
      IF(T1)350,355,355
  350 T2=T2-1.0
  355 FL(L,K)=T1
      IF(T2)360,365,365
  360 T3=T3-1.0
  365 CONTINUE
C     ***** CHECK FOR 4 POINTS INSIDE QUADRANGLE *****
      IF(T3)370,375,375
  370 ITYPE=-1
      GO TO 415
C     ***** CHECK FOR 1 TO 3 POINTS INSIDE QUADRANGLE ****
  375 IF(T3-3.0)380,385,385
  380 ITYPE=0
      GO TO 415
C     ***** DETERMINE WHICH EDGES ARE CROSSED BY THE 4 LINE SEGMENTS ***
  385 DO 405 L=1,4
      L1=MOD(L,4)+1
C     ***** LINE SEGMENT L FROM POINT Y1 TO POINT Y2 *****
      Y1(1)=QUAD(L*2-1,IQ)
      Y1(2)=QUAD(L*2,IQ)
      Y2(1)=QUAD(L1*2-1,IQ)
      Y2(2)=QUAD(L1*2,IQ)
      DO 405 K=1,4
      T1=FL(K,L)
      T2=FL(K,L1)
      T3=T1-T2
C     ***** T1 AND T2 MUST HAVE OPPOSITE SIGNS FOR INTERSECTION TO OCCUR
      IF(T1*T2)390,390,405
C     ***** COMPONENT OF SEGMENT L PERPENDICULAR TO EDGE K OF IB IS T3
  390 IF(ABS(T3)-1.E-5)405,405,395
C     ***** CALCULATE COORDINATES OF INTERSECTION *****
  395 T4=(T1*Y2(1)-T2*Y1(1))/T3
      T5=(T1*Y2(2)-T2*Y1(2))/T3
      K0=2*K
```

```
      K1=2*(MOD(K,4)+1)
C     ***** IS INTERSECTION WITHIN QUADRANGLE IQ *****
      T6=(T4-QUAD(K0-1,IB))*(QUAD(K1-1,IB)-T4)+(T5-QUAD(K0,IB))*
     1 (QUAD(K1,IB)-T5)
      IF(ABS(T6)-1.E-4)410,410,400
  400 IF(T6)405,410,410
  405 CONTINUE
      GO TO 495
  410 ITYPE=1
C     ***** CHECK OVER/UNDER AMBIGUITY *****
  415 IF((NA1-NB1)*(NA2-NB2)*(NA2-NB1))425,420,425
C     ***** BONDS SHARE AN ATOM *****
  420 IF(NA1+NA2-NB1-NB2)465,495,495
  425 CALL DIFV(ATOMS(1,NB2),ATOMS(1,NB1),V2)
      CALL DIFV(ATOMS(1,NB1),ATOMS(1,NA1),V4)
      CALL UNITY(V2,V2,1)
      CALL UNITY(V4,V4,1)
      CALL NORM(V1,V2,V3,1)
      IF(VV(V3,V3)-TOL)430,430,435
C     ***** PARALLEL BONDS, RECALCULATE V3 *****
  430 CALL NORM(V1,V4,V5,1)
      CALL NORM(V5,V1,V3,1)
C     ***** CHECK FOR COLLINEAR BONDS *****
      IF(VV(V3,V3)-TOL)465,465,450
  435 IF(VV(V3,V4)+TOL)440,450,450
  440 DO 445 J=1,3
  445 V3(J)=-V3(J)
C     ***** V3 IS NORMAL TO BONDS IQ AND IB GOING FROM IQ TOWARD IB ***
  450 IF(VIEW)455,455,460
  455 IF(V3(3))495,495,465
  460 IF(VV(VUE,V3))465,495,495
C     ***** OVERLAPPING BOND FOUND *****
  465 ICQ=ICQ+1
      IF(ITYPE)470,475,475
C     ***** HIDDEN BOND *****
  470 ICQ=-1
      RETURN
C     ***** STORE INTERFERING QUADRANGLE *****
  475 IF(NQOVER-30)485,480,480
  480 NG=18
      TIDD=TID2
      CALL ERPNT(TIDD,800)
      RETURN
  485 NQOVER=NQOVER+1
      DO 490 K=1,4
      DO 490 J=1,3
  490 QOVER(J,K,NQOVER)=QUA(J,K)
      KQ(NQOVER)=IB
  495 CONTINUE
  500 RETURN
      END
      SUBROUTINE LAPAB(IQ,IA,ICQ,ITY)
C     ***** SUBROUTINE CHECKS FOR OVERLAP BETWEEN ATOMS AND BONDS *****
C     ***** CALLED BY SUBROUTINES LAP700 AND LAP800 *****
      DIMENSION BF(4),CON(3,3),QF(5),QUA(3,4)
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 TIDD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      TID=QUAD(9,IQ)
      NA1=TID/1000.
      NA2=AMOD(TID,1000.)
C     ***** ITY.GT.0, CHECK FOR ATOMS OVER A BOND ****
C     ***** ITY.LT.0, CHECK FOR BONDS OVER AN ATOM *****
      ICQ=0
      IF(ITY)210,200,205
  200 RETURN
  205 CALL LAPCON(CONIC(1,IA),CON,V1,0.0)
      IF(ITY-2)220,240,240
  210 IF(ITY+2)220,220,215
  215 CALL LAPCON(CONIC(1,IA),CON,V1,OVMRGN)
C     ***** SET UP LINEAR FORMS FOR EDGES OF QUADRANGLE *****
  220 DO 235 L=1,4
      K=2*L
      K1=MOD(K,8)+2
      QUA(1,L)=QUAD(K,IQ)-QUAD(K1,IQ)
      QUA(2,L)=QUAD(K1-1,IQ)-QUAD(K-1,IQ)
      QUA(3,L)=QUAD(K-1,IQ)*QUAD(K1,IQ)-QUAD(K,IQ)*QUAD(K1-1,IQ)
      T1=SQRT(QUA(1,L)**2+QUA(2,L)**2)
      IF(T1)225,225,230
  225 ITY=0
      ICQ=0
      GO TO 430
C     ***** TRANSFORM COEFFICIENTS FOR EDGES TO NORMAL FORM *****
  230 DO 235 J=1,3
  235 QUA(J,L)=QUA(J,L)/T1
C     ***** EVALUATE 4 QUADRATIC AND 4 BILINEAR FORMS *****
  240 V2(3)=1.0
      V3(3)=1.0
      T2=3.0
      DO 265 L=1,4
      L1=(MOD(L,4)+1)*2
      V2(1)=QUAD(2*L-1,IQ)
      V2(2)=QUAD(2*L,IQ)
      V3(1)=QUAD(L1-1,IQ)
      V3(2)=QUAD(L1,IQ)
      QF(L)=0.0
      BF(L)=0.0
      DO 250 K=1,3
      T1=CON(3,K)
      DO 245 J=1,2
  245 T1=T1+V2(J)*CON(J,K)
      QF(L)=QF(L)+T1*V2(K)
  250 BF(L)=BF(L)+T1*V3(K)
      IF(QF(L))260,255,265
  255 T2=T2-0.8
      GO TO 265
  260 T2=T2-1.0
  265 CONTINUE
      QF(5)=QF(1)
C     ***** CHECK FOR 4 POINTS OF QUADRANGLE INSIDE OR ON ELLIPSE *****
      IF(T2)270,275,275
  270 ITYPE=-1
```

```
      GO TO 330
C     ***** CHECK FOR 1 TO 3 POINTS OF QUADRANGLE INSIDE THE ELLIPSE ***
  275 IF(T2-2.2)280,285,285
  280 ITYPE=0
      IF(NA2-IA)340,375,335
C     ***** CHECK FOR QUADRANGLE-ELLIPSE INTERSECTION *****
  285 DO 305 K=1,4
C     ***** EVALUATE DISCRIMINANT *****
      T1=BF(K)**2-QF(K)*QF(K+1)
      IF(T1)305,305,290
  290 T1=SQRT(T1)
C     ***** IS INTERSECTION WITHIN BOUNDS OF QUADRANGLE *****
      T3=QF(K)-BF(K)
      T4=T3+QF(K+1)-BF(K)
      IF(ABS(T4)-1.E-5)305,305,295
  295 T5=(T3-T1)/T4
      IF(T5)305,280,300
  300 IF(1.0-T5)305,305,280
  305 CONTINUE
C     ***** NO VALID INTERSECTION FOUND *****
C     ***** CHECK FOR CENTER OF ELLIPSE WITHIN THE QUADRANGLE ****
      T3=3.0
      DO 320 K=1,4
      T1=QUA(3,K)
      DO 310 J=1,2
  310 T1=T1+V1(J)*QUA(J,K)
      IF(T1)315,320,320
  315 T3=T3-1.0
  320 CONTINUE
      IF(T3)325,370,370
  325 ITYPE=1
C     ***** CHECK OVER/UNDER AMBIGUITY *****
  330 IF(NA2-IA)375,375,335
  335 IF(IA-NA1)375,375,340
  340 CALL DIFV(ATOMS(1,NA2),ATOMS(1,NA1),V2)
      CALL DIFV(ATOMS(1,IA),ATOMS(1,NA1),V3)
      CALL UNITY(V2,V2,1)
      CALL UNITY(V3,V3,1)
      CALL NORM(V2,V3,V4,1)
      IF(VV(V4,V4)-1.E-5)345,345,350
C     ***** CENTER OF ATOM IQ IS ON THE BOND LINE *****
  345 IF(ITY)370,370,385
C     ***** CENTER OF ATOM IQ IS NOT ON THE BOND LINE *****
  350 CALL NORM(V4,V2,V5,1)
      T1=-V5(3)
      IF(VIEW)365,365,355
  355 T1=V5(3)*(ATOMS(3,IA)-VIEW)
      DO 360 J=1,2
  360 T1=T1+V5(J)*ATOMS(J,IA)
  365 IF(T1*FLOAT(ITY))375,375,370
C     ***** NO INTERFERENCE FOUND *****
  370 ICQ=0
      GO TO 430
C     ***** ITYPE=1 ENCLOSED ELLIPSE / ITYPE=-1 ENCLOSED QUADRANGLE ****
  375 IF(ITYPE*ITY)380,385,385
C     ***** HIDDEN ATOM OR HIDDEN BOND *****
  380 ICQ=-1
      GO TO 430
  385 ICQ=1
      IF(ITY)410,390,390
C     ***** STORE INTERFERING ELLIPSE *****
  390 IF(NCOVER-20)400,395,395
  395 NG=17
      CALL ERPNT(ATOMID(IA),800)
      NCOVER=NCOVER-1
  400 NCOVER=NCOVER+1
      IJ=1
      DO 405 I=1,3
      DO 405 J=I,3
      COVER(IJ,NCOVER)=CON(I,J)
  405 IJ=IJ+1
      KC(NCOVER)=IA
      GO TO 430
C     ***** STORE INTERFERING QUADRANGLE *****
  410 IF(NQOVER-30)420,415,415
  415 NG=18
      TIDD=TID
      CALL ERPNT(TIDD,700)
      NQOVER=NQOVER-1
  420 NQOVER=NQOVER+1
      DO 425 K=1,4
      DO 425 J=1,3
  425 QOVER(J,K,NQOVER)=QUA(J,K)
      KQ(NQOVER)=IQ
  430 RETURN
      END
      SUBROUTINE LAPCON(CON1,CON,Y,OVMR)
C     ***** TRANSFORM CONIC TO PLOTTER HOMOGENEOUS COORDINATE SYSTEM ***
C     ***** CALLED BY SUBROUTINES LAP700 AND LAPAB *****
      DIMENSION CON1(7),CON(3,3),Y(3)
      Y(1)=(CON1(1)+CON1(2))*0.5
      Y(2)=(CON1(3)+CON1(4))*0.5
      Y(3)=1.0
      CON(1,1)=CON1(5)
      CON(1,2)=CON1(6)
      CON(2,1)=CON1(6)
      CON(2,2)=CON1(7)
      T1=(CON1(2)-CON1(1)+CON1(4)-CON1(3))*0.25
      CON(3,3)=-((T1-OVMR)/T1)**2
      DO 205 K=1,2
      CON(K,3)=0.0
      DO 200 J=1,2
  200 CON(K,3)=CON(K,3)-Y(J)*CON(J,K)
      CON(3,K)=CON(K,3)
  205 CON(3,3)=CON(3,3)-CON(3,K)*Y(K)
      RETURN
      END
      SUBROUTINE LAPDRW(Y,NPEN,NCQ)
C     ***** SUBROUTINE ELIMINATES HIDDEN LINES AND DRAWS VISIBLE LINES *
      DIMENSION CB(20),CQ(50,2),QL(4,30,2),SEG(2),Y(3),YN(3),YO(3),Z(3)
      COMMON/OLAP/CONIC(7,500),COVER(6,20),KC(20),KQ(30),NCONIC,NCOVER,
     1 NQOVER,NQUAD,OVMRGN,QOVER(3,4,30),QUAD(9,600),SEGM(50,2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
```

```
      NCQ=NCOVER+NQOVER
      IF(NCQ)200,200,205
  200 RETURN
C     ***** CHECK ALL OVERLAPPING ATOMS AND BONDS *****
  205 NPM3=NPEN-3
      IF(NPM3)  210,230,230
C     ***** SAVE INFORMATION FROM LAST POINT IF PEN IS DOWN *****
  210 YO(1)=YN(1)
      YO(2)=YN(2)
      YO(3)=1.0
      NPO=NPN
      DO 215 K=1,NCQ
  215 CQ(K,1)=CQ(K,2)
      IF(NQOVER)230,230,220
  220 DO 225 K=1,NQOVER
      DO 225 J=1,4
  225 QL(J,K,1)=QL(J,K,2)
C     ***** EVALUATE CONIC QUADRATIC FORMS AT NEW POINT YN *****
  230 YN(1)=Y(1)
      YN(2)=Y(2)
      YN(3)=1.0
      NPN=NPEN
      IF(NCOVER)250,250,235
  235 DO 245 K=1,NCOVER
      Z(1)=YN(1)*COVER(1,K)+YN(2)*COVER(2,K)+COVER(3,K)
      Z(2)=YN(1)*COVER(2,K)+YN(2)*COVER(4,K)+COVER(5,K)
      Z(3)=YN(1)*COVER(3,K)+YN(2)*COVER(5,K)+COVER(6,K)
      CQ(K,2)=Z(1)*YN(1)+Z(2)*YN(2)+Z(3)
C     ***** EVALUATE CONIC BILINEAR FORM IF PEN IS DOWN *****
      IF(NPM3)240,245,245
  240 CB(K)= Z(1)*YO(1)+Z(2)*YO(2)+Z(3)
  245 CONTINUE
C     ***** EVALUATE LINEAR FORMS AND SIGNATURE FOR QUADRANGLE *****
  250 IF(NQOVER)275,275,255
  255 KCQ=NCOVER
      DO 270 K=1,NQOVER
      T2=3.0
      DO 265 J=1,4
      T1=YN(1)*QOVER(1,J,K)+YN(2)*QOVER(2,J,K)+QOVER(3,J,K)
      IF(T1)260,265,265
  260 T2=T2-1.0
  265 QL(J,K,2)=T1
      KCQ=KCQ+1
C     ***** T2=-1 INSIDE, =0 ACROSS ANY EDGE, =1 ACROSS ANY VERTEX *****
  270 CQ(KCQ,2)=T2
C     ***** IF PEN IS UP, OMIT ALL SUBSEQUENT CHECKING *****
  275 IF(NPM3)285,280,280
  280 NPN=3
      CALL SCRIBE(YN,NPN)
      RETURN
C     ***** CHECK FOR HIDDEN SEGMENT *****
  285 DO 295 K=1,NCQ
      IF(CQ(K,1))290,295,295
  290 IF(CQ(K,2))280,295,295
  295 CONTINUE
C     ***** FIND ENTRY AND EXIT POINTS ON EACH CONIC *****
      NINT=0
      IF(NCOVER)330,330,300
  300 DO 325 K=1,NCOVER
C     ***** EVALUATE DISCRIMINANT *****
      T1=CB(K)**2-CQ(K,1)*CQ(K,2)
      IF(T1)325,325,305
  305 T1=SQRT(T1)
C     ***** SOLVE QUADRATIC EQATION *****
      T2=CQ(K,1)-CB(K)
      T3=T2+CQ(K,2)-CB(K)
      IF(ABS(T3)-1.E-5)325,325,310
  310 T4=(T2-T1)/T3
      T5=(T2+T1)/T3
C     ***** VALID INTERSECTION IF T4.LT.1 AND T5.GT.0 *****
      IF(T4-1.0)315,325,325
  315 IF(T5)325,325,320
C     ***** SAVE VALID CONIC INTERSECTIONS *****
  320 NINT=NINT+1
      SEGM(NINT,1)=T4
      SEGM(NINT,2)=T5
  325 CONTINUE
  330 IF(NQOVER)425,425,335
C     ***** FIND ENTRY AND EXIT POINTS FOR EACH QUADRANGLE *****
  335 DO 420 K=1,NQOVER
      I12=0
      KCQ=NCOVER+K
C     ***** CHECK FOR SINGLE INSIDE POINT *****
      SEG(1)=CQ(KCQ,1)
      IF(SEG(1))345,340,340
  340 SEG(1)=1.0-CQ(KCQ,2)
      IF(SEG(1)-1.0)350,350,345
C     ***** INSIDE POINT FOUND, ONLY ONE INTERSECTION POSSIBLE *****
  345 I12=1
C     ***** FIND WHICH EDGES ARE CROSSED BY THE SEGMENT *****
  350 DO 410 J=1,4
      T1=QL(J,K,1)
      T2=QL(J,K,2)
      T3=T1-T2
      IF(T1*T2)355,355,410
C     ***** CHECK FOR SEGMENT ON AN EDGE *****
  355 IF(ABS(T3)-1.E-5)420,420,360
C     ***** CALCULATE COORDINATES OF INTERSECTION *****
  360 T4=(T1*YN(1)-T2*YO(1))/T3
      T5=(T1*YN(2)-T2*YO(2))/T3
      J1=2*(MOD(J,4)+1)
      IQ=KQ(K)
C     ***** IS INTERSECTION WITHIN LIMITS OF QUADRANGLE *****
      T6=(T4-QUAD(2*J-1,IQ))*(QUAD(J1-1,IQ)-T4)+(T5-QUAD(2*J,IQ))*
     1 (QUAD(J1,IQ)-T5)
      IF(ABS(T6)-1.E-4)370,370,365
  365 IF(T6)410,370,370
C     ***** CALCULATE FRACTION PARAMETER AND STORE IT *****
  370 T1=T1/T3
      IF(I12-1)375,380,395
C     ***** STORE FIRST INTERSECTION *****
  375 I12=1
      GO TO 390
C     ***** STORE SECOND INTERSECTION ****
  380 I12=2
      IF(T1-SEG(1))385,405,405
  385 SEG(2)=SEG(1)
  390 SEG(1)= T1
      GO TO 410
C     ***** MORE THAN TWO INTERSECTIONS (I.E.,QUADRANGLE DIAGONAL) *****
  395 IF(T1-SEG(1))390,410,400
  400 IF(T1-SEG(2))410,410,405
```

```
  405 SEG(2)=T1
  410 CONTINUE
      IF(I12-1)420,420,415
C     ***** STORE FRACTION PARAMETERS *****
  415 NINT=NINT+1
      SEGM(NINT,1)=SEG(1)
      SEGM(NINT,2)=SEG(2)
  420 CONTINUE
C     ***** END OF ENTRY-AND-EXIT-POINT CALCULATIONS *****
  425 IF(NINT-1)430,490,435
C     ***** NO INTERFERENCE FOUND, DRAW ENTIRE SEGMENT *****
  430 CALL SCRIBE(YN,2)
      RETURN
C     **** SORT SEGMENT INTERSECTION LIST *****
C     ***** SORTING PROCEDURE BY SHELL,D.L. COMM. ACM 2,30-32 (1959) ***
  435 M=NINT
  440 M=M/2
      IF(M)490,490,445
  445 K=NINT-M
      J=1
  450 I=J
  455 IM=I+M
      IF(SEGM(I,1))460,470,470
  460 IF(SEGM(IM,1))465,465,485
  465 IF(SEGM(I,2)-SEGM(IM,2))485,485,475
  470 IF(SEGM(I,1)-SEGM(IM,1))485,485,475
  475 DO 480 L=1,2
      T1=SEGM(I,L)
      SEGM(I,L)=SEGM(IM,L)
  480 SEGM(IM,L)=T1
      I=I-M
      IF(I)485,485,455
  485 J=J+1
      IF(J-K)450,450,440
C     ***** FIND STARTING POINT P0 AND END POINT P1 *****
  490 P0=0.0
      K=0
  495 K=K+1
      IF(K-NINT)500,500,515
  500 P1=SEGM(K,1)
      IF(P1)510,505,505
  505 IF(P1-P0)510,510,520
  510 P0=AMAX1(P0,SEGM(K,2))
      IF(P0-1.0)495,530,525
  515 P1=1.0
C     ***** DRAW SEGMENT FROM P0 TO P1 *****
  520 IF(P0)535,535,530
  525 P0=1.0
  530 Z(1)=YO(1)*(1.-P0)+YN(1)*P0
      Z(2)=YO(2)*(1.-P0)+YN(2)*P0
      NPN=3
      CALL SCRIBE(Z,NPN)
      IF(P0-1.0)535,540,540
  535 Z(1)=YO(1)*(1.-P1)+YN(1)*P1
      Z(2)=YO(2)*(1.-P1)+YN(2)*P1
      NPN=2
      CALL SCRIBE(Z,NPN)
      IF(P1-1.0)510,540,540
  540 RETURN
      END
      character*(*) function maksym(k,gp)
c *** returns character string representation of symmetry operator
      dimension gp(3,4,192)
      character*1 xyz(3)
      character*5 fract(23)
      character*12 part(3)
      data fract/'1/24','1/12','1/8','1/6','5/24','1/4','7/24','1/3',
     *          '3/8','5/12','11/24','1/2','13/24','7/12','5/8','2/3',
     *          '17/24','3/4','19/24','5/6','7/8','11/12','23/24'/
      data xyz/'x','y','z'/
      do 200 i=1,3
         part(i) = ' '
         iff = 0
         do 300 j=1,3
            if (ifix(gp(i,j,k)) .ne. 0) then
               if (ifix(gp(i,j,k)) .eq. -1)
     *            part(i) = part(i)(1:iend(part(i))) // '-' // xyz(j)
               if (ifix(gp(i,j,k)) .eq. 1 .and. iff .eq. 0)
     *            part(i) = part(i)(1:iend(part(i))) // ' ' // xyz(j)
               if (ifix(gp(i,j,k)) .eq. 1 .and. iff .eq. 1)
     *            part(i) = part(i)(1:iend(part(i))) // '+' // xyz(j)
               iff = 1
            end if
  300    continue
         gpval = gp(i,4,k)
         if(gpval.gt..01 .or. gpval.lt.-.01) then
            if (gpval.lt.0.) then
               part(i) = part(i)(1:iend(part(i))) // '-'
            else
               part(i) = part(i)(1:iend(part(i))) // '+'
            end if
            gpval=abs(gpval)
            tfour=1./24.
            do 301 mm=1,23
               tf=float(mm)*tfour
               if (gpval.gt.(tf-.01) .and. gpval.lt.(tf+.01)) iw=mm
  301       continue
            part(i) = part(i)(1:iend(part(i))) // fract(iw)
         end if
  200 continue
      maksym = part(1) // part(2) // part(3)
      return
      end
      SUBROUTINE MM(X,Y,Z)
C     MULTIPLY TWO MATRICES
C     Z(3,3)=X(3,3)*Y(3,3)
      DIMENSION X(3,3),Y(3,3),Z(3,3)
      X11=X(1,1)
      X12=X(1,2)
      X13=X(1,3)
      X21=X(2,1)
      X22=X(2,2)
      X23=X(2,3)
      X31=X(3,1)
      X32=X(3,2)
      X33=X(3,3)
      Y11=Y(1,1)
      Y12=Y(1,2)
      Y13=Y(1,3)
      Y21=Y(2,1)
      Y22=Y(2,2)
      Y23=Y(2,3)
```

```
      Y31=Y(3,1)
      Y32=Y(3,2)
      Y33=Y(3,3)
      Z(1,1)=X11*Y11+X12*Y21+X13*Y31
      Z(2,1)=X21*Y11+X22*Y21+X23*Y31
      Z(3,1)=X31*Y11+X32*Y21+X33*Y31
      Z(1,2)=X11*Y12+X12*Y22+X13*Y32
      Z(2,2)=X21*Y12+X22*Y22+X23*Y32
      Z(3,2)=X31*Y12+X32*Y22+X33*Y32
      Z(1,3)=X11*Y13+X12*Y23+X13*Y33
      Z(2,3)=X21*Y13+X22*Y23+X23*Y33
      Z(3,3)=X31*Y13+X32*Y23+X33*Y33
      RETURN
      END
      SUBROUTINE MV(X,Y,Z)
C     MATRIX * VECTOR
C     Z(3)=X(3,3)*Y(3)
      DIMENSION X(3,3),Y(3),Z(3)
      Y1=Y(1)
      Y2=Y(2)
      Y3=Y(3)
      Z(1)=X(1,1)*Y1+X(1,2)*Y2+X(1,3)*Y3
      Z(2)=X(2,1)*Y1+X(2,2)*Y2+X(2,3)*Y3
      Z(3)=X(3,1)*Y1+X(3,2)*Y2+X(3,3)*Y3
      RETURN
      END
      SUBROUTINE NORM(X,Y,Z,ITYPE)
C     ***** VECTOR PRODUCT  Z=X*Y *****
C     ***** ITYPE .GT.0 FOR CARTESIAN,.LE.0 FOR TRICLINIC *****
      DIMENSION X(3),Y(3),Z(3),Z1(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      DO 125 I=1,3
      I1=MOD(I+3,3)+1
      I2=MOD(I+1,3)+1
      T1=X(I1)*Y(I2)-X(I2)*Y(I1)
      IF(ITYPE)115,115,105
  105 Z(I)=T1
      GO TO 125
  115 Z1(I)=T1
  125 CONTINUE
      IF(ITYPE)135,135,300
  135 CALL MV(BB,Z1,Z)
  300 RETURN
      END
      SUBROUTINE NUMBUR(W,W2,HGT,DIST,THT,ND)
C-----CONVERT BOND DISTANCE FOR PLOTTING IN ORTEP
      DIMENSION W(3)
      CHARACTER*8 IFMT,ITXT
      CHARACTER*1 ITEX(8)
      EQUIVALENCE (ITEX(1),ITXT)
C-----COMPUTE NUMBER OF CHARACTERS FOR OUTPUT
      NC=ND+1
      XD=DIST
   10 IF(XD.LT.1.0) GO TO 20
      IF(NC.GE.9) GO TO 30
      NC=NC+1
      XD=XD/10.0
      GO TO 10
C-----SET UP FORMAT STATEMENT
   20 WRITE (IFMT,25) NC,ND
   25 FORMAT('(F',I1,'.',I1,')')
C-----ENCODE DISTANCE AND PUT IT OUT
      WRITE (ITXT,IFMT) DIST
      CALL SIMBOL(W,W2,HGT,ITEX,THT,NC)
   30 RETURN
      END
      SUBROUTINE PAXES(DCODE,ITYPE)
C     ***** ITYPE .LT.0 FOR COVARIANCE MATRIX IN Q *****
C     ***** ITYPE .GT.0 FOR ELLIPSOID QUADRATIC FORM IN Q *****
C     ***** XABSF(ITYPE)=1 BASED ON TRICLINIC COORDINATE SYSTEM *****
C     ***** =2 OR 3 FOR WORKING OR REFERENCE CARTESIAN SYSTEMS *****
C     ***** CONTRAVARIANT EIGENVECTORS FOR Q IN COLUMNS OF PAC *****
C     ***** CHECK ATOM CODE *****
      DIMENSION X(3)
      REAL*8 DCODE,D100,D100K
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      D100=100.
      D100K=100000.
      IT=IABS(ITYPE)-1
      KS=DMOD(DCODE,D100)
      IF(NSYM-KS)105,115,115
  105 NG=4
      GO TO 300
  115 II=DCODE/D100K
      IF(NATOM-II)125,130,130
  125 NG=5
      GO TO 300
  130 IF(II)125,125,135
C     ***** CRYSTALLOGRAPHIC SYMMETRY ROTATION *****
  135 CALL TMM(PA(1,1,II),FS(1,1,KS),PAT)
      IF(IT-1)160,145,155
C     ***** TRANSFORM TO CARTESIAN SYSTEMS *****
  145 CALL TMM(PAT,AAWRK,PAC)
      GO TO 175
  155 CALL TMM(PAT,AAREV,PAC)
      GO TO 175
  160 IF(ITYPE)162,155,170
C     ***** TRANSFORM TO TRICLINIC SYSTEM *****
  162 DO 165 J=1,9
  165 PAC(J,1)=PAT(J,1)
```

```
      GO TO 175
  170 CALL MM(AA,PAT,PAC)
C     ***** FORM DIAGONAL MATRIX OR ITS INVERSE *****
  175 DO 205 J=1,3
      T1=EV(J,II)
      IF(ITYPE)195,195,185
  185 X(J)=1./(T1*T1)
      GO TO 205
  195 X(J)=T1*T1
  205 RMS(J)=T1
C     ***** FORM QUADRATIC FORM *****
      DO 245 I=1,3
      DO 245 J=I,3
      T1=0.0
      DO 225 K=1,3
  225 T1=T1+PAC(I,K)*PAC(J,K)*X(K)
      Q(J,I)=T1
  245 Q(I,J)=T1
  300 RETURN
      END
      SUBROUTINE PLTXY(X,Y)
C     ***** PLOT COORD. AND CLOSEST EDGE AFTER PROJECTION *****
      DIMENSION X(3),Y(2)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      T4=1.
      T1=1.
      IF(VIEW)125,125,110
  110 T4=VIEW-X(3)
      IF(T4)115,115,120
  115 Y(1)=-99.
      Y(2)=-99.
      GO TO 130
  120 T1=VIEW/T4
  125 Y(1)=X(1)*T1+XO(1)
      Y(2)=X(2)*T1+XO(2)
      T1=XLNG(1)-ABS(Y(1)*2.-XLNG(1))
      T2=XLNG(2)-ABS(Y(2)*2.-XLNG(2))
      EDGE=AMIN1(T1,T2)*.5
      IF(T4-VIEW*.5)130,300,300
  130 EDGE=-99.
  300 RETURN
      END
      SUBROUTINE PLOT(x,y,ipen)
      common /ns/ npf,ndraw,norient,nvar
      if (ndraw .eq. 0) return
      if (ndraw .eq. 1) call pensc(x,y,ipen)
      if (ndraw .eq. 2) call penps(x,y,ipen)
      if (ndraw .eq. 3) call penhp(x,y,ipen)
      if (ndraw .eq. 9) call pensc(x,y,ipen)
      return
      end
      SUBROUTINE PRELIM
C     ***** DATA INPUT ROUTINE *****
      REAL*8 TD
      DIMENSION B(9)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      CHARACTER*73 INQ,QUE,hque
      character*80 card
      character*24 sympart(3)
      dimension fsym(3,4)
      character*36 maksym
C     ***** CELL DIMENSIONS *****
      D100K=100000.
  106 FORMAT(i1,f8.6,5F9.6)
      READ (IN,107)card
  107 format(a)
      write (NED,107) card
      READ (card,106)iflag,(A(I),I=1,6)
      T1=ABS(A(4))-1.
      DO 125 J=1,3
      IF(T1)115,110,110
C     ***** CELL ANGLES IN DEGREES *****
  110 A(J+6)=A(J+3)
      A(J+3)=COS(A(J+6)*1.745329E-2)
      GO TO 120
C     ***** COSINES OF CELL ANGLES *****
  115 A(J+6)=ARCCOS(A(J+3))
C     ***** STORE IDEMFACTOR MATRIX *****
  120 AID(J,J)=1.
      AID(J+1,1)=0.
      AID(J+5,1)=0.
C     ***** STORE METRIC TENSOR *****
  125 AA(J,J)=A(J)**2
      AA(1,2)=A(1)*A(2)*A(6)
      AA(1,3)=A(1)*A(3)*A(5)
      AA(2,3)=A(2)*A(3)*A(4)
      AA(2,1)=AA(1,2)
      AA(3,1)=AA(1,3)
      AA(3,2)=AA(2,3)
C     ***** INVERT METRIC TENSOR *****
      CALL AXEQB(AA,BB,AID,3)
C     ***** CALCULATE RECIPROCAL CELL PARAMETERS *****
      DO 128 J=1,3
  128 B(J)=SQRT(BB(J,J))
      B(6)=BB(1,2)/(B(1)*B(2))
      B(5)=BB(1,3)/(B(1)*B(3))
      B(4)=BB(2,3)/(B(2)*B(3))
      DO 130 J=1,3
  130 B(J+6)=ARCCOS(B(J+3))
```

```
C     ***** WAS INPUT FOR REAL OR RECIPROCAL CELL *****
      IF(A(1)-1.)135,150,150
  135 DO 140 J=1,9
      T1=AA(J,1)
      AA(J,1)=BB(J,1)
      BB(J,1)=T1
      T1=A(J)
      A(J)=B(J)
  140 B(J)=T1
C     ***** WRITE OUT CELL PARAMETERS *****
  143 FORMAT(1H010X,22HDIRECT CELL PARAMETERS/1H 15X,1HA14X,1HB14X,1HC14
     1X,5HALPHA10X,4HBETA11X,5HGAMMA)
  145 FORMAT(1H ,10X,F9.5,2F15.6,3F15.3/1H ,48X,6HCOSINE,F12.8,2F15.8)
  147 FORMAT(1H010X,26HRECIPROCAL CELL PARAMETERS/1H 15X,2HA*13X,2HB*13X
     1,2HC*13X,6HALPHA*9X,5HBETA*10X,6HGAMMA*)
  150 IF (NOUT.GE.0)
     &WRITE (NOUT,143)
      IF (NOUT.GE.0)
     &WRITE (NOUT,145)(A(I),I=1,3),(A(I),I=7,9),(A(I),I=4,6)
      IF (NOUT.GE.0)
     &WRITE (NOUT,147)
      IF (NOUT.GE.0)
     &WRITE (NOUT,145)(B(I),I=1,3),(B(I),I=7,9),(B(I),I=4,6)
C     ***** STORE STANDARD VECTORS *****
      CALL AXES(AID,AID(1,2),REFV,0)
      CALL MM(AA,REFV,AAREV)
      DO 160 I=1,3
      DO 160 J=1,3
      AAWRK(J,I)=AAREV(J,I)
      Q(J,I)=REFV(I,J)
  160 WRKV(J,I)=REFV(J,I)
C     ***** READ AND WRITE SYMMETRY TRANSFORMATIONS *****
  171 FORMAT(1H0,10X,24HSYMMETRY TRANSFORMATIONS/1H ,14X,3HNO.)
c 171 FORMAT(1H010X,24HSYMMETRY TRANSFORMATIONS/1H 14X,3HNO.12X,13HTRANS
c    1FORMED X18X,13HTRANSFORMED Y18X,13HTRANSFORMED Z)
  173 FORMAT(I1,F14.10,3F3.0,2(F15.10,3F3.0))
  175 FORMAT(1H ,13X,I2,3(F13.6,F4.0,2H X,F4.0,2H Y,F4.0,2H Z))
  176 FORMAT(1H ,13X,I2,5x,a)
  177 FORMAT(1H1,10X,18A4)
      IF (NOUT.GE.0)
     &WRITE (NOUT,171)
      LINES=14
      DO 190 I=1,96
      LINES=MOD(LINES+1,56)
      READ (IN,107)card
      write (NED,107) card
      if(iflag.eq.0)READ(card,173)IS,(TS(J,I),(FS(K,J,I),K=1,3),J=1,3)
      if (iflag.eq.1) then
         read (card,1771) is
 1771    format(i1)
         ipart=1
         do 1772 jk=1,3
         do 1772 kl=1,24
 1772    sympart(jk)(kl:kl)=' '
         jk=2
 1773    if (card(jk:jk).eq.' ') go to 1776
         lm=1
         do 1774 kl=jk,80
            if(card(kl:kl).eq.' ' .or. card(kl:kl).eq.',') then
               jk=kl
               go to 1775
            end if
            sympart(ipart)(lm:lm)=card(kl:kl)
            lm=lm+1
 1774    continue
 1775    ipart=ipart+1
 1776    jk=jk+1
         if (jk.lt.80) go to 1773
         do 1777 isymp=1,3
            call tepsym(sympart(isymp),i,isymp)
 1777    continue
      end if
      do 178 j=1,3
         fsym(j,4)=ts(j,i)
         do 178 k=1,3
  178       fsym(j,k)=fs(k,j,i)
      IF(LINES)185,180,185
  180 IF (NOUT.GE.0)
     &WRITE (NOUT,177)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,171)
c *** ORTEP II symmetry output
c 185 IF (NOUT.GE.0)
c    &WRITE (NOUT,175)I,(TS(J,I),(FS(K,J,I),K=1,3),J=1,3)
  185 if (nout.ge.0) WRITE (NOUT,176)I,maksym(1,fsym)
C     ***** NON-CRYSTALLOGRAPHIC HELIX-SYMMETRY INPUT *****
      IF(FS(3,3,I)-5.)188,186,186
  186 T1=FS(1,3,I)/FS(3,3,I)
      TS(3,I)=TS(3,I)+T1
      T1=AMOD(T1*FS(2,3,I),1.)*6.28318531
      T2=COS(T1)
      T1=SIN(T1)
      DO 187 J=1,9
  187 VT(J,1)=AID(J,1)
      VT(1,1)=T2
      VT(2,2)=T2
      VT(2,1)=-T1
      VT(1,2)=T1
      CALL MM(VT,Q,PAC)
      CALL MM(AAREV,PAC,FS(1,1,I))
  188 IF(IS)195,190,195
  190 CONTINUE
      NG=1
      CALL ERPNT(0.D0,0)
      I=96
  195 NSYM=I
  196 ISW=IS
      NATOM=0
C     ***** POSITIONAL AND THERMAL PARAMETERS *****
  207 FORMAT(11H0 NO. ATOM 8X,1HX10X,1HY10X,1HZ13X,3HB118X,3HB228X,3HB33
     18X,3HB128X,3HB138X,10HB23    TYPE)
  209 FORMAT(1H ,I3,1X,A6,3F11.6,5X,6F11.6,F5.0)
  210 FORMAT(1H ,I3,1X,A6,3F11.6,5X,2F11.6,4F11.0,F5.0)
  211 FORMAT(A6,3X,6F9.0)
  213 FORMAT(I1,F8.0,5F9.0,7X,F2.0)
      LINES=LINES+2
      IF(LINES-56)220,215,215
  215 LINES=-1
      GO TO 225
  220 IF (NOUT.GE.0)
     &WRITE (NOUT,207)
      if (isw.eq.2) then
```

```
        iu=18
        call gtafil(iu)
      end if
C     ***** MAXIMUM NUMBER OF ATOMS EQUALS MAXATM *****
  225 MATOM=NATOM+1
      DO 245 I=MATOM,MAXATM
      LINES=MOD(LINES+1,56)
      IF(ISW.EQ.1) GO TO 226
C     ***** CALL SPECIAL PURPOSE READIN ROUTINE *****
      CALL readin(iu,CHEM(I),IDENT(1,I),IDENT(2,I),P(1,I),P(2,I),P(3,I)
     1,IT1,IS,PA(1,1,I),PA(2,1,I),PA(3,1,I),PA(1,2,I),PA(2,2,I)
     2,PA(3,2,I),PA(1,3,I))
      T1=IT1
      K=IT1+1
      GO TO 229
  226 continue
      READ (IN,107)card
      write (NED,107) card
      READ (card,211)CHEM(I),V1(1),V1(2),(P(J,I),J=1,3),T1
      IDENT(1,I)=V1(1)
      IDENT(2,I)=V1(2)
      K=1.+T1
      IF(FLOAT(K-1)-T1)227,228,227
  227 K=1
  228 continue
      READ (IN,107)card
      write (NED,107) card
      READ (card,213)IS,(PA(J,1,I),J=1,7)
  229 IF(LINES)230,230,232
  230 IF (NOUT.GE.0)
     &WRITE (NOUT,177)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,207)
  232 IF(PA(3,1,I)-10000.)235,234,234
  234 IF (NOUT.GE.0)
     &WRITE (NOUT,210)I,CHEM(I),(P(J,I),J=1,3),(PA(J,1,I),J=1,7)
      GO TO 238
  235 IF (NOUT.GE.0)
     &WRITE (NOUT,209)I,CHEM(I),(P(J,I),J=1,3),(PA(J,1,I),J=1,7)
  238 GO TO (244,239,241,242,244),K
C     ***** TYPE 1 POSITIONAL PARAMETERS (ANGSTROMS) *****
  239 DO 240 J=1,3
  240 P(J,I)=P(J,I)/A(J)
      GO TO 244
C     ***** TYPE 2 POSITIONAL PARAMETERS, STANDARD CARTESIAN *****
  241 V1(1)=P(1,I)
      V1(2)=P(2,I)
      GO TO 243
C     ***** TYPE 3 POSITIONAL PARAMETERS *****
C     ***** CYLINDRICAL COORDINATES REFERRED TO STANDARD CARTESIAN *****
  242 T2=P(2,I)*.01745329252
      V1(1)=V1(1)+P(1,I)*COS(T2)
      V1(2)=V1(2)+P(1,I)*SIN(T2)
  243 V1(3)=P(3,I)
      CALL VM(V1,Q,P(1,I))
  244 IF(IS)246,245,246
  245 CONTINUE
      if (isw.eq.2) close(iu)
      NG=2
      CALL ERPNT(0.D0,0)
      I=MAXATM
  246 NATOM=I
C     ***** CONVERT TEMP FACTOR COEF TO STANDARD TYPE ZERO *****
      NG1=0
      DO 450 I=1,NATOM
      T1=PA(1,1,I)
c     interim fix for IBM AIX
      K9=7
      K=1.+PA(K9,1,I)
      IF(T1)255,250,255
  250 T1=.1
      GO TO 405
  255 T6=.0506605918
      GO TO(270,260,265,265,270,260,400,405,270,260,270,450),K
C     ***** TYPE 1 *****
  260 DO 262 J=4,6
  262 PA(J,1,I)=PA(J,1,I)*.5
      GO TO 270
C     ***** TYPES 2 AND 3 (BASE 2 SYSTEMS) *****
  265 T6=.351152464
      IF(K-4)270,260,270
C     ***** TYPES 0 THROUGH 5 *****
  270 IF(PA(2,1,I))400,400,272
  272 DO 300 J=1,3
      DO 300 L=J,3
      T2=T6
      IF(K-5)285,275,275
  275 IF(K-6)280,280,281
C     ***** TYPES 4 AND 5 *****
  280 T2=B(J)*B(L)*T2*.25
      GO TO 285
C     ***** TYPES 8 AND 9 (U(I,J) TENSOR SYSTEMS) *****
  281 T2=B(J)*B(L)
      IF(K-11)285,282,282
C     ***** TYPE 10, (CARTESIAN TENSOR SYSTEM) *****
  282 T2=1.0
  285 IF(J-L)290,287,290
  287 VT(J,J)=T2*PA(J,1,I)
      GO TO 300
  290 M=J+L+1
      VT(J,L)=T2*PA(M,1,I)
      VT(L,J)=VT(J,L)
  300 CONTINUE
C     ***** FIND PRINCIPAL AXES *****
      IF(K-11)310,305,305
  305 CALL MM(VT,Q,PAC)
      CALL MM(REFV,PAC,VT)
  310 CALL MM(VT,AA,DA)
      CALL EIGEN(DA,RMS,PAT)
C     ***** ARE EIGENVALUES POSITIVE *****
      IF(RMS(1))325,325,320
  320 IF(NG)350,360,330
  325 NG=3
  330 NG1=1
      CALL ERPNT(DBLE(I)*D100K+55501.D0,0)
C     ***** 3 EQUAL EIGENVALUES, USE REFERENCE VECTORS *****
  340 T3=SIGN(SQRT(ABS(RMS(1)+RMS(2)+RMS(3))/3.),RMS(1))
      DO 345 J=1,3
      DO 342 K=1,3
  342 PA(J,K,I)=REFV(J,K)
  345 EV(J,I)=T3
      GO TO 450
```

```
  350 IF(NG+6)340,340,352
C     ***** TWO EQUAL EIGENVALUES *****
  352 N=NG+5
      CALL UNITY(PAT(1,N),V1,-1)
      DO 354 K=1,3
      IF(ABS(VMV(V1,AA,REFV(1,K)))-.58)356,354,354
  354 CONTINUE
  356 CALL MM(AA,DA,VT)
      CALL AXES(V1,REFV(1,K),DA,-1)
      DO 359 K=1,3
      L=MOD(N+K-2,3)+1
      DO 358 J=1,3
  358 PA(J,L,I)=DA(J,K)
  359 EV(L,I)=SIGN(SQRT(ABS(VMV(DA(1,K),VT,DA(1,K)))),RMS(L))
      GO TO 450
C     ***** MAKE EIGENVECTORS 1 ANGSTROM LONG *****
  360 CALL AXES(PAT(1,1),PAT(1,3),PA(1,1,I),-1)
  370 NG=0
C     ***** SQRT EIGENVALUE = RMS DISPLACEMENT *****
      DO 375 J=1,3
      T2=RMS(J)
  375 EV(J,I)=SIGN(SQRT(ABS(T2)),T2)
      GO TO 450
C     ***** TYPE 6 (ISOTROPIC TEMP FACTOR) *****
  400 T1=SQRT(T1*1.266515E-2)
C     ***** TYPE 7 (DUMMY SPHERE OR ELLIPSOID OF REVOLUTION) *****
  405 IF(PA(2,1,I))409,409,406
C     ***** ELLIPSOID OF REVOLUTION FOR PASS OR PALE *****
  406 EV(1,I)=T1
      EV(2,I)=PA(2,1,I)
      EV(3,I)=PA(2,1,I)
      GO TO 411
C     ***** SPHERE FOR PEAK OR PIT, OR A GENERAL SPHERE ATOM *****
  409 DO 410 J=1,3
  410 EV(J,I)=T1
  411 IF(PA(3,1,I))430,430,415
C     ***** FIRST DEFINED VECTOR FOR SPHERE OR CRITICAL POINT *****
  415 DO 417 J=1,2
      TD=PA(J+2,1,I)
      CALL ATOM(TD,VT(1,J))
      IF(NG)416,417,416
  416 CALL ERPNT(TD,0)
      GO TO 430
  417 CONTINUE
      CALL DIFV(VT(1,2),VT(1,1),V1)
      T11=SQRT(VMV(V1,AA,V1))
      DO 418 J=1,3
  418 V1(J)=V1(J)/T11
C     ***** SECOND DEFINED VECTOR FOR SPHERE OR CRITICAL POINT *****
      DO 420 J=3,4
      TD=PA(J+2,1,I)
      IF(TD.EQ.0.0)GO TO 422
      CALL ATOM(TD,VT(1,J))
      IF(NG)419,420,419
  419 CALL ERPNT(TD,0)
      GO TO 430
  420 CONTINUE
      CALL DIFV(VT(1,4),VT(1,3),V2)
      T11=SQRT(VMV(V2,AA,V2))
      DO 421 J=1,3
  421 V2(J)=V2(J)/T11
C     ***** CHECK FOR NEARLY PARALLEL UNIT VECTORS *****
      IF(ABS(VMV(V1,AA,V2)).LT.0.9) GO TO 429
C     ***** SUBSTITUTE BEST REFERENCE VECTOR *****
  422 T22=1.0
      J22=0
      DO 424 J=1,3
      T11=ABS(VMV(V1,AA,REFV(1,J)))
      IF(T22.LE.T11)GO TO 424
      T22=T11
      J22=J
  424 CONTINUE
      DO 425 J=1,3
  425 V2(J)=REFV(J,J22)
  429 CALL AXES(V1,V2,PA(1,1,I),-1)
      GO TO 450
C     ***** REFERENCE VECTORS FOR SPHERE *****
  430 DO 435 J=1,9
  435 PA(J,1,I)=REFV(J,1)
  450 NG=0
C     ***** WRITE OUT RMS VALUES *****
      LINES=LINES+2
      IF(LINES-56)458,458,455
  455 LINES=-1
      GO TO 460
  458 IF (NOUT.GE.0)
     &WRITE (NOUT,461)
  460 DO 465 I=1,NATOM
      LINES=MOD(LINES+1,56)
      IF(LINES)465,462,465
  461 FORMAT(10H0NO. ATOM 8X,1HX10X,1HY10X,1HZ13X,7HRMSD 1 4X,7HRMSD 2 4
     1X,7HRMSD 3 )
  462 IF (NOUT.GE.0)
     &WRITE (NOUT,177)(TITLE(J),J=1,18)
      IF (NOUT.GE.0)
     &WRITE (NOUT,461)
  463 FORMAT(1H ,I3,1X,A6,3F11.6)
  465 IF (NOUT.GE.0)
     &WRITE (NOUT,209)I,CHEM(I),(P(J,I),J=1,3),(EV(J,I),J=1,3
     1)
      IF(NG1)999,999,470
  470 CALL EXITNG(NG)
  999 RETURN
      END
      SUBROUTINE PRIME
C     ****GENERAL INITIALIZATION OF PRIME PARAMETERS****
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      BRDR=0.5
C     ****CALCULATE CONSTANTS****
```

```
      DO 2950 I=1,5
 2950 CONT(I)=SQRT(1./(2.*(1.+COS(3.141593   /2.**I))))
c     DISP=.005
      disp=0.
      FORE=.866
      ITILT=0
      LATM=0
      MAXATM=505
      NCD=0
      NG=0
      DO 3000 J=1,3
 3000 ORGN(J) = 0.0
      RES(1)=.75
      RES(2)=.5*res(1)
      RES(3)=.25*res(2)
      SCAL1=1.0
      SCAL2=1.54
      SCL=1.54
      DO 3005 I=1,3
      SYMB(I,I)=1.
      SYMB(I+1,1)=0.
 3005 SYMB(I+5,1)=0.
      TAPER=.375
      THETA=0.0
      VIEW=0.0
      XLNG(1)=10.5
      XLNG(2)=8.0
      XO(1)=5.25
      XO(2)=4.0
      XO(3)=0.0
c     ***** INITIATE OVERLAP ROUTINES *****
      CALL LAP500(0)
      RETURN
      END
      SUBROUTINE PROJ(D,DP,X,XO,VIEW,I1,I2,I3)
c     ***** 3D CARTESIAN TO 2D PLOTTER COORDINATES *****
      DIMENSION D(3,129),DP(2,129),X(3),XO(3)
      T3=VIEW-X(3)
      DO 145 I=I1,I2,I3
      T1=D(1,I)+X(1)
      T2=D(2,I)+X(2)
      IF(VIEW)135,135,120
  120 T4=VIEW/(T3-D(3,I))
      T1=T1*T4
      T2=T2*T4
  135 DP(1,I)=T1+XO(1)
  145 DP(2,I)=T2+XO(2)
      RETURN
      END
      SUBROUTINE RADIAL(ND)
c     ***** GENERATE ELLIPSE FROM TWO CONJUGATE VECTORS *****
c     ***** ORTHONORMAL VECTORS PRODUCE 8-128 SPOKED CIRCLE *****
c     ***** ND DENOTES NUMBER OF SUBDIVISIONS (1 TO 5) *****
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      DO 115 J=1,3
      T1=DA(J,1)
      D(J,1)=T1
      D(J,129)=T1
      D(J,65)=-T1
      T1=DA(J,2)
      D(J,33)=T1
  115 D(J,97)=-T1
      DO 135 K=1,ND
      T1=CONT(K)
      KDEL=2**(6-K)
      KDEL1=KDEL+1
      KDEL2=KDEL/2
      DO 135 L=KDEL1,65,KDEL
      J=L-KDEL
      M=L-KDEL2
      DO 135 N=1,3
      T2=(D(N,L)+D(N,J))*T1
      D(N,M)=T2
  135 D(N,M+64)=-T2
      RETURN
      END
      subroutine readin(iu,chem,id1,id2,x1,x2,x3,it,is,b1,b2,b3,b4,
     1                  b5,b6,btype)
      integer*2 id1,id2
      character*1 chain
      character*3 res
      character*4 atom
      character*6 rec
      character*8 chem
      b1=.1
      b2=0
      b3=0
      b4=0
      b5=0
      b6=0
      btype=7.
      id1=0
      id2=0
      it=2
c     ***** read the pdb file *****
      read (iu,201) rec,iserno,atom,res,chain,id2,x1,x2,x3,occ,tf
  201 format(a6,i5,1x,a4,1x,a3,1x,a1,i4,4x,3f8.0,2f6.0)
      id1=9
      if (atom.eq.' N  ') id1=1
      if (atom.eq.' CA ') id1=2
      if (atom.eq.' C  ') id1=3
      if (atom.eq.' O  ') then
         id1=4
         b1=.15
      end if
      chem=atom(2:4)//res
      is=0
      read (iu,202,end=203) rec
  202 format(a6)
      backspace(iu)
      return
  203 is=1
      return
```

```
      end
      SUBROUTINE recycle
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      COMMON /QUEUE/ NED,NQUE,NEXT,NBACK,INQ,QUE(96),hque(96)
      CHARACTER*73 INQ,QUE,hque
      common /ns/ npf,ndraw,norient,nvar

      if (ndraw.eq.2 .or. ndraw.eq.3) call getpap

c *** find 201(203) instruction
      do 545 i=1,nque
         if (que(i)(7:9).eq.'201' .or. que(i)(7:9).eq.'203') then
            next=i
            go to 570
         end if
  545 continue
C *** ZERO ATOMS ARRAY AND RETURN TO EXECUTE NEXT INSTRUCTION ***
  570 LATM=0
      DO 580 I=1,500
      ATOMID(I)=0.
      DO 580 J=1,3
  580 ATOMS(J,I)=0.
      RETURN
      END
      SUBROUTINE SCRIBE(Y,NPEN)
      DIMENSION Y(2),YO(2)
C     ***** SUBROUTINE WHICH LINKS WITH THE PLOTTER-SPECIFIC SUBROUTINES
      IF(NPEN-3)210,205,205
C     ***** KEEP TRACK OF COORDINATES FOR LAST PEN-UP LOCATION *****
  205 YO(1)=Y(1)
      YO(2)=Y(2)
      NPO=0
      RETURN
C     ***** CALL MECHANICAL PLOTTER PLOTTING SUBROUTINE *****
  210 IF(NPO)225,220,225
  220 CONTINUE
      CALL PLOT(YO(1),YO(2),3)
  225 CONTINUE
      CALL PLOT(Y(1),Y(2),2)
      NPO=1
      RETURN
      END
      SUBROUTINE SEARC
      DIMENSION NW(6),DX(3),S1D(200),S2(200),U(3),V(3),W(2,4),WW(2,3)
      DIMENSION X(4),Y(3),Z(3)
      REAL*8 DZMIN,DZMAX,DZSTO,S1D,TD1,TD2,TD3,TD4,D10K,D100K
      REAL*8 TD
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      CHARACTER*8 CHEM
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      INTEGER*2 IDENT
      COMMON /PARMS/ CHEM(505),EV(3,505),P(3,505),PA(3,3,505)
     1 ,IDENT(2,505),MAXATM
      logical featur
      featur=.false.
      ifeat=ain(6)
      if ((ifeat.eq.1 .or. ifeat.eq.2) .and.
     &    (mod(nj2,10).eq.5 .or. mod(nj2,10).eq.6)) featur=.true.
C     ***** OBTAIN PROBLEM PARAMETERS *****
      D10K=10000.
      D100K=100000.
      IF (NOUT.GE.0)
     &WRITE (NOUT,20)
   20 FORMAT(1H0 9X,82H        FROM ATOMS    TO ATOMS     WITH RADIUS OR
     1, IF A BOX, WITH SEMIDIMENSIONS  /11X,46HCODE   (MIN  MAX)   (MIN
     2 MAX)                  7X,1HA8X,1HB8X,1HC)
      IF(AIN(1)-D10K)100,100,101
  100 ITOM1=AIN(1)
      SYITOM=55501.
      GO TO 103
  101 ITOM1=AIN(1)/D100K
      SYITOM=DMOD(AIN(1),D100K)
  102 IF(DABS(AIN(2))-D10K)103,103,104
  103 ITOM2 = DABS(AIN(2))
      SYITO2=SYITOM
      GO TO 105
  104 ITOM2=DABS(AIN(2))/D100K
      SYITO2=DMOD(DABS(AIN(2)),D100K)
  105 ITAR1=AIN(3)
      IF(ITAR1)108,108,110
  108 ITAR1=1
  110 ITAR2=AIN(4)
      DMAX=AIN(5)
      IF(DMAX)115,115,120
  115 DMAX=4.
      AIN(5)=DMAX
  120 DMX=DMAX*DMAX
      TEM=.01
      KFUN=NJ*100+MOD(NJ2,10)
      K=NJ*100+NJ2
      I0=SYITOM
      I02=SYITO2
      LATOM=LATM
  121 FORMAT(1H0,10X,2I3,I5,I4,I5,2I4,18X,3F9.3/1H )
      IF (NOUT.GE.0)
     &WRITE (NOUT,121)K,ITOM1,I0,ITOM2,I02,
     1ITAR1,ITAR2,(AIN(J),J=5,7)
  124 FORMAT(1H ,15X,2I5,I8,I5,2F9.3)
      IF(NCD)130,130,125
  125 IF (NOUT.GE.0)
     &WRITE (NOUT,124) ((KD(J,I),J=1,4),(CD(J,I),J=1,2) ,I=1,
     1NCD)
  130 DO 135 J=1,4
      W(1,J)=99.
  135 W(2,J)=-99.
```

```
      inum=0
      DO 153 KI=ITAR1,ITAR2
      i=ki
  136 if (featur) then
         inum=inum+1
         i=inum
         if (ident(ifeat,i).ne.ki) go to 154
      end if
      TD=FLOAT(I)*D100K
      CALL ATOM(TD,X)
      IF(NG)140,145,140
  140 CALL ERPNT(TD,KFUN)
      GO TO 600
  145 X(4)=X(1)-X(2)
      DO 155 J=1,4
      TEM=X(J)
      IF(W(2,J)-TEM)148,150,150
  148 W(2,J)=TEM
  150 IF(TEM-W(1,J))152,155,155
  152 W(1,J)=TEM
  155 CONTINUE
  154 if (featur .and. inum.lt.natom) go to 136
  153 CONTINUE
      KFUN2=MOD(KFUN,10)
      GO TO (165,165,160,156,165,165),KFUN2
C     ***** FIND PARALLELEPIPED WHICH ENCLOSES TRICLINIC BOX *****
  156 DO 158 J=1,3
  158 DX(J)=AIN(J+4)
      GO TO 170
C     ***** FIND PARALLELEPIPED WHICH ENCLOSES RECTANGULAR BOX *****
  160 DO 162 J=1,3
      DX(J)=0.
      DO 162 I=1,3
      T9=AIN(I+4)
  162 DX(J)=DX(J)+ABS(REFV(J,I)*T9)
      GO TO 170
C     ***** FIND PARALLELEPIPED WHICH ENCLOSES DMAX SPHERE *****
  165 T1=1.-A(4)*A(4)-A(5)*A(5)-A(6)*A(6)+2.*A(4)*A(5)*A(6)
      DO 168 J=1,3
  168 DX(J)=SQRT((1.-A(J+3)**2)/T1)*DMAX/A(J)
C     ***** START SEARCH AROUND REFERENCE ATOMS *****
  170 LIST=0
      LAST=0
      M1=ITOM1
      N1=ITOM2
      IF(KFUN2-5)186,172,172
C     ***** CONVOLUTE AND REITERATIVE CONVOLUTE INSTRUCTIONS *****
  172 IF(LATM)174,174,176
C     ***** FAULT, NO ENTRIES IN ATOMS LIST *****
  174 NG=12
      CALL ERPNT(0.D0,KFUN)
      GO TO 600
C     ***** CHECK FOR REFERENCE ATOMS IN ATOMS LIST *****
  176 IF(LATM-LAST)600,600,177
  177 LIST=LAST
      LAST=LATM
  178 LIST=LIST+1
      IF(LAST-LIST)505,180,180
  180 TD1=ATOMID(LIST)
      IF(LAST-LIST.LE.0 .OR. AIN(8).EQ.0.D0) GO TO 184
C     ***** FIND SMALLEST ATOM NUMBER IN REMAINDER OF ATOMS LIST *****
      LISTP1=LIST+1
      DO 182 J=LISTP1,LAST
      IF(TD1.LE.ATOMID(J)) GO TO 182
      DO 181 I=1,3
      T1=ATOMS(I,J)
      ATOMS(I,J)=ATOMS(I,LIST)
  181 ATOMS(I,LIST)=T1
      TD1=ATOMID(J)
      ATOMID(J)=ATOMID(LIST)
      ATOMID(LIST)=TD1
  182 CONTINUE
  184 ITOM=TD1/D100K
      if (featur) then
         if (ident(ifeat,itom).lt.itom1 .or.
     &       ident(ifeat,itom).gt.itom2) go to 178
      else
         IF(ITOM.LT.ITOM1 .OR. ITOM.GT.ITOM2) GO TO 178
      end if
      SYITOM=DMOD(TD1,D100K)
      SYITO2=SYITOM
      M1=ITOM
      N1=ITOM
C     ***** SET INITIAL RUN PARAMETERS *****
  186 M2=AMOD(SYITOM,100.)
      M5=AMOD(SYITOM/100.,1000.)
      M3=M5/100
      M4=MOD(M5/10,10)
      M5=MOD(M5,10)
C     ***** SET TERMINAL RUN PARAMETERS *****
      N2=AMOD(SYITO2,100.)
      N5=AMOD(SYITO2/100.,1000.)
      N3=N5/100
      N4=MOD(N5/10,10)
      N5=MOD(N5,10)
C     ***** START SEARCH AROUND REFERENCE ATOMS *****
      DO 500 L5=M5,N5
      DO 500 L4=M4,N4
      DO 500 L3=M3,N3
      DO 500 L2=M2,N2
      DO 500 ITOM=M1,N1
      TD3=DBLE(ITOM)*D100K+DBLE(L3*10000+L4*1000+L5*100+L2)
      CALL ATOM(TD3,Y)
      IF(NG)188,190,188
  188 CALL ERPNT(TD3,KFUN)
      GO TO 500
C     ***** K=SYMMETRY EQUIVALENT POSITION *****
  190 NUM=0
      DO 400 K=1,NSYM
C     ***** SUBTRACT SYMMETRY TRANSLATION FROM REFERENCE ATOM *****
      DO 192 J=1,3
  192 U(J)=Y(J)-TS(J,K)
C     ***** DETERMINE LIMITING CELLS TO BE SEARCHED *****
C     ***** FIRST,MOVE THE BOX THROUGH THE SYMMETRY OPERATION *****
      DO 200 J=1,3
      DO 200 L=1,2
      WW(L,J)=0.0
      DO 200 I=1,3
      TEM=FS(I,J,K)
      IF(TEM)194,200,196
  194 N=MOD(L,2)+1
      GO TO 198
```

```
  196 N=L
  198 WW(L,J)=WW(L,J)+W(N,I)*TEM
  200 CONTINUE
C     ***** CHECK FOR MIXED INDEX TRANSFORMATION *****
      DO 215 J=1,2
      TEM=FS(1,J,K)
      IF(TEM+FS(2,J,K))215,201,215
  201 IF(TEM)203,215,207
  203 WW(1,J)=W(2,4)*TEM
      WW(2,J)=W(1,4)*TEM
      GO TO 215
  207 WW(1,J)=W(1,4)*TEM
      WW(2,J)=W(2,4)*TEM
  215 CONTINUE
C     ***** MOVE 4 CELLS AWAY THEN MOVE BACK UNTIL PARALLELEPIPED AROUND
C         REF ATOM AND BOX AROUND TRANSFORMED ASYM UNIT INTERSECT *****
      N=0
      DO 235 J=1,3
      DO 225 I=1,2
      N=N+1
      TT=(U(J)-WW(I,J))*FLOAT(I*2-3)-DX(J)
      TEM=5.0
  221 TEM=TEM-1.0
      IF(TEM+TT)225,225,221
  225 NW(N)=TEM*FLOAT(I*2-3)+5.
C     ***** IF NO POSSIBILITY OF A HIT, GO TO NEXT SYMMETRY OPER *****
      IF(NW(N)-NW(N-1))400,235,235
  235 CONTINUE
      LL=NW(1)
      LU=NW(2)
      ML=NW(3)
      MU=NW(4)
      NL=NW(5)
      NU=NW(6)
C     ***** L CELL TRANSLATIONS IN X *****
      DO 396 L=LL,LU
      V(1)=U(1)+FLOAT(L-5)
C     ***** M CELL TRANSLATIONS IN Y *****
      DO 396 M=ML,MU
      V(2)=U(2)+FLOAT(M-5)
C     ***** N CELL TRANSLATIONS IN Z *****
      DO 396 NN=NL,NU
      V(3)=U(3)+FLOAT(NN-5)
C     ***** I = TARGET ATOM *****
      inum=0
      DO 396 KI=ITAR1,ITAR2
      i=ki
  244 if (featur) then
         inum=inum+1
         i=inum
         if (ident(ifeat,i).ne.ki) go to 395
      end if
      DO 250 J=1,3
      TEM=0.0
      DO 245 II=1,3
  245 TEM=TEM+FS(II,J,K)*P(II,I)
C     ***** SEE IF WITHIN PARALLELEPIPED*****
      TEM=TEM-V(J)
      IF(DX(J)-ABS(TEM))395,250,250
  250 X(J)=TEM
      GO TO (255,255,252,265,255,255),KFUN2
C     ***** SEE IF WITHIN MODEL BOX *****
  252 CALL VM(X,AAREV,V1(2))
      DO 253 J=2,4
      IF(AIN(J+3)-ABS(V1(J)))395,253,253
  253 CONTINUE
      GO TO 265
C     ***** SEE IF WITHIN SPHERE *****
  255 DSQ=VMV(X,AA,X)
      IF(DMX-DSQ)395,256,256
  256 IF(DSQ-.0001)258,260,260
  258 IF(KFUN-402)395,260,260
  260 TEM=SQRT(DSQ)
      IF(AIN(8))265,265,261
C     *****SELECT ONLY FIRST ASYMMETRIC UNIT ENCOUNTERED *****
  261 IF(LATM)265,265,262
  262 DZMIN=DBLE(I)*D100K
      DZMAX=DZMIN+D100K
      DO 264 J=1,LATM
      DZSTO=ATOMID(J)
      IF(DZSTO-DZMIN)264,263,263
  263 IF(DZMAX-DZSTO)264,264,395
  264 CONTINUE


C     ***** SELECT VECTORS ACCORDING TO CODES IF ANY *****
  265 if(ncd.le.0) go to 277
c     if logc=0, screening conditions are ORed
c     if logc=1, screening conditions are ANDed
      logc=ain(8)
  268 DO 275 J=1,NCD
      norg=itom
      ntar=i
      if (kd(5,j).eq.1) then
         norg=ident(1,itom)
         ntar=ident(1,i)
      end if
      if (kd(5,j).eq.2) then
         norg=ident(2,itom)
         ntar=ident(2,i)
      end if
      if (logc.eq.0) then
         if (kd(2,j).gt.0) then
            if (norg.lt.kd(1,j) .or. norg.gt.kd(2,j)) go to 275
         end if
         if (kd(4,j).gt.0) then
            if (ntar.lt.kd(3,j) .or. ntar.gt.kd(4,j)) go to 275
         end if
         if (cd(2,j).gt.0.) then
            if (tem.lt.cd(1,j)  .or. tem.gt.cd(2,j)) go to 275
         end if
         go to 277
      end if
      if (logc.eq.1) then
         if (kd(2,j).gt.0) then
            if (norg.lt.kd(1,j) .or. norg.gt.kd(2,j)) go to 276
         end if
         if (kd(4,j).gt.0) then
            if (ntar.lt.kd(3,j) .or. ntar.gt.kd(4,j)) go to 276
         end if
         if (cd(2,j).gt.0.) then
            if (tem.lt.cd(1,j)  .or. tem.gt.cd(2,j)) go to 276
```

```
      end if
      if (j.eq.ncd) go to 277
    end if
275 CONTINUE


276 GO TO 395
277 TD=D100K*DBLE(I)+DBLE((1110-L*100-M*10-NN)*100+K)
    IF(KFUN-402)278,325,325
C     ***** DETERMINE CORRECT POSITION IN SORTED VECTOR TABLE *****
278 IF(NUM)317,317,279
279 DO 315 II=1,NUM
    TT=S2(II)-TEM
    IF(ABS(TT)-0.0001)297,297,281
281 IF(TT)315,297,283
C     ***** MOVE LONGER VECTORS TOWARD END OF TABLE *****
283 IF(200-NUM)287,287,289
287 NUM=199
289 IJ=NUM
    DO 295 J=II,NUM
    S1D(IJ+1)=S1D(IJ)
    S2(IJ+1)=S2(IJ)
295 IJ=IJ-1
    GO TO 319
C     ***** CHECK FOR DUPLICATE VECTORS IF DISTANCES ARE EQUAL *****
297 CALL ATOM(S1D(II),Z)
    DO 305 J=1,3
    IF(ABS(X(J)+Y(J)-Z(J))-0.0001)305,305,315
305 CONTINUE
    GO TO 395
315 CONTINUE
    IF(200-NUM)320,320,317
C     ***** STORE THE RESULT IN VECTOR TABLE *****
317 II=NUM+1
319 NUM=NUM+1
    S1D(II)=TD
    S2(II)=TEM
320 IF(KFUN-106)395,325,325
C     ***** STORE RESULT IN ATOMS TABLE *****
325 DO 330 J=1,3
330 V1(J)=X(J)+Y(J)
    CALL STOR(TD)
395 if (featur .and. inum.lt.natom) go to 244
396 CONTINUE
400 CONTINUE
C     ***** PRINT OUT DISTANCES *****
421 FORMAT(1H010X,20HVECTORS FROM ATOM  (I3,1H,I5,1H)6X,8HTO ATOMSI4,
   18H THROUGH,I4)
    I0 = DMOD(TD3,D100K)
    IF (NOUT.GE.0)
   &WRITE (NOUT,421)ITOM,I0,ITAR1,ITAR2
    IF(NUM)500,500,423
423 DO 435 I=1,NUM
    TD2=S1D(I)
    I1=TD2/D100K
    I2=TD2-DBLE(I1)*D100K
    CALL ATOM(TD2,Z)
    IF(I-1)432,432,434
427 FORMAT(1H 13X,2(A6,1X),39X,1H(I3,1H,I5,1H)3F7.4,7X,3HD =F6.3)
429 FORMAT(1H 13X,2(A6,1X),2(3H  (I3,1H,I5,1H)3F7.4,3X),4X,3HD =F6.3)
432 IF (NOUT.GE.0)
   &WRITE (NOUT,429)CHEM(ITOM),CHEM(I1),ITOM,I0,(Y(J),J=1,3
   1),I1,I2,(Z(J),J=1,3),S2(I)
    GO TO 435
434 IF (NOUT.GE.0)
   &WRITE (NOUT,427)CHEM(ITOM),CHEM(I1),I1,I2,(Z(J),J=1,3),
   1S2(I)
435 CONTINUE
C     ***** CALCULATE ANGLES ABOUT REF ATOM IF CODE IS 102 *****
437 IF(KFUN-102)500,451,500
441 FORMAT(1H0,10X,18HANGLES AROUND ATOM,I5)
451 IF (NOUT.GE.0)
   &WRITE (NOUT,441)ITOM
    L=NUM-1
    IF(L)500,500,457
457 DO 465 I=1,L
    TD2=S1D(I)
    T3=S2(I)
    I1=TD2/D100K
    I2=TD2-DBLE(I1)*D100K
    CALL ATOM(TD2,X)
    CALL DIFV(X,Y,U)
    CALL MV(AA,U,V2)
    M=I+1
    DO 465 J=M,NUM
    TD4=S1D(J)
    J1=TD4/D100K
    J2=TD4-DBLE(J1)*D100K
    CALL ATOM(TD4,Z)
    CALL DIFV(Z,Y,V)
    F=ARCCOS(VV(V,V2)/(T3*S2(J)))
    CALL DIFV(X,Z,V3)
    F1=SQRT(VMV(V3,AA,V3))
460 FORMAT(1H 13X,3(A6,1X),7X,3(2H (I3,1H,I5,1H)),12X,3HD =F6.3,7X,3HA
   1 =F6.2)
    IF (NOUT.GE.0)
   &WRITE (NOUT,460)CHEM(I1),CHEM(ITOM),CHEM(J1),I1,I2,ITOM
   1,I0,J1,J2,F1,F
465 CONTINUE
495 CONTINUE
500 CONTINUE
    IF(LAST-LIST)505,505,178
505 IF(KFUN2-6)600,176,600
600 IF(KFUN-106)610,605,610
605 LATM=LATOM
610 RETURN
    END
    SUBROUTINE SIMBOL(W,W2,HGT,ITXT,THT,N)
    DIMENSION W(3),ITXT(72),DS(10),DC(10)
c     DIMENSION IPTR(64),NKNT(64),IXYT(349)
    DIMENSION IPTR(90),NKNT(90),IXYT(556)
    CHARACTER*1 ITXT
    common /ns/ npf,ndraw,norient,nvar
    common /trfac/ xtrans,ytrans
    DATA IPTR
   1/312, 32, 41, 54, 47, 20, 20, 54, 64, 70,103,108, 17,115,118,128
   2,140,128,140,194,166,102,208,120, 27,211,216,  0,  0,  0,  0,237
   3,  0,150,328,301,168,223, 92,328, 88,188,180,  1, 82,180, 82, 30
   4,128,239,244,252,269,277,258,284,192,289, 76, 76, 11,  6, 14,153
   5,350,360,370,378,388,398,407,421,428,436,444,450,455,465,472,484
   6,494,504,510,521,526,533,536,541,546,550/
    DATA NKNT
```

```
      1/ 16,  9, 12,  8,  7,  7,  6, 10,  6,  6,  5,  7,  3,  5,  4,  9
      2,  7, 12, 10, 12,  4,  6,  3,  5,  5,  5,  7,  0,  0,  0,  2
      3,  0,  7,  9, 11, 12, 14, 10,  4,  4,  4,  8,  5,  6,  2,  5,  2
      4,  9,  5,  8, 13,  8,  9, 11,  5, 16, 12, 11, 12,  3,  5,  3, 13
      5, 10, 10,  8, 10, 10,  9, 14,  7,  8,  8,  6,  5, 10,  7,  9, 10
      6, 10,  6, 11,  5,  7,  3,  5,  5,  4,  7/
C         @   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O
C         P   Q   R   S   T   U   V   W   X   Y   Z                   _
C             !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
C         0   1   2   3   4   5   6   7   8   9   :   ;   <   =   >   ?
C         a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p
C         q   r   s   t   u   v   w   x   y   z
       DATA IXYT
      1/44,48,46,26,66,24,64,99,66,26,68,26,64,24,66,28,29,22,62,69
      2,29,26,56,26,22,62,29,62,99,22,69,22,25,65,25,28,39,59,68,62
      3,63,65,56,26,56,67,68,59,29,22,52,63,68,68,59,39,28,23,32,52
      4,63,65,55,22,29,26,66,69,62,32,52,42,49,39,59,35,36,46,45,35
      5,99,42,32,33,43,42,31,69,58,53,62,62,37,38,49,58,25,24,33,43
      6,64,29,23,32,52,63,69,29,22,25,69,99,47,62,22,29,45,69,62,29
      7,22,46,62,69,47,69,99,68,59,39,28,23,32,52,63,68,99,44,62,22
      8,29,59,68,67,56,26,56,65,62,49,44,99,32,43,52,32,99,44,46,56
      9,67,68,59,39,28,29,69,49,42,99,23,53,64,65,56,36,27,38,68,25
      A,65,45,63,27,45,23,67,29,38,33,22,56,67,68,59,39,28,27,36,56
      B,65,63,52,32,23,25,36,29,42,69,29,47,42,47,69,29,69,22,62,99
      C,36,56,38,28,29,39,38,99,69,22,99,53,63,62,52,53,15,75,38,49
      D,42,32,52,28,39,59,68,66,24,22,62,28,39,59,68,67,56,36,56,65
      E,63,52,32,23,28,39,59,68,29,24,64,54,59,52,42,62,23,32,52,63
      F,65,56,26,29,69,68,43,42,23,32,52,63,68,59,39,28,26,35,55,66
      G,24,64,54,53,57,56,66,26,36,37,33,66,57,47,36,35,44,54,65,67
      H,58,38,27,24,33,53,64,57,49,59,57,99,37,29,39,37,22,32,12,22
      I,23,21,22,31,13,22,33,11,22
c 349
      a,62,67,66,57,37,26,23,32,52,63
      b,22,29,26,37,57,66,63,52,32,23
      c,63,52,32,23,26,37,57,66
      d,62,69,66,57,37,26,23,32,52,63
      e,63,52,32,23,26,37,57,66,65,35
      f,32,36,26,46,36,38,49,59,68
      g,64,67,66,57,37,26,24,33,53,64,62,51,31,22
c 420
      h,22,29,26,37,57,66,62
      i,32,52,42,47,37,99,48,49
      j,32,41,51,62,67,99,68,69
      k,22,29,24,57,35,62
      l,32,52,42,49,39
      m,22,27,26,37,46,42,46,57,66,62
c 458
      n,22,27,26,37,57,66,62
      o,63,52,32,23,26,37,57,66,63,99,67,45
      p,21,27,26,37,57,66,64,53,33,24
      q,61,67,66,57,37,26,24,33,53,64
      r,22,27,26,37,57,66
c 500 82
      s,23,32,52,63,64,55,35,26,37,57,66
c     t,42,48,46,36,56
      t,42,49,47,27,67
      u,67,62,63,52,32,23,27
      v,27,42,67
c     w,27,22,45,62,67
      w,27,32,46,52,67
      x,22,67,99,27,62
      y,27,43,67,31
c     y,27,44,67,21,21
c     y,27,42,31,42,67
      z,62,22,67,27,99,35,55/
       DATA RAD/0.017453329/
       if (ndraw.eq.9) then
          write (npf,21) n,nvar,w(1)+xtrans,w(2)+ytrans,8.*hgt,tht
 21       format('TXT',i2,1x,i2,' 1',4(1x,f10.6))
          write (npf,22) (itxt(k),k=1,n)
 22       format(80a1)
          return
       end if
C-----TEST FOR SPECIAL CASE OF CENTERED SYMBOL
      IF(N.LE.0) GO TO 400
C-----SET UP TABLE OF INCREMENTS BASED ON HGT AND THT
      IF(THT.EQ.0.0) GO TO 120
      TH=RAD*THT
      ST=SIN(TH)
      CT=COS(TH)
      GO TO 130
 120  ST=0.0
      CT=1.0
 130  D=HGT/7.0
      DST=D*ST
      DCT=D*CT
      DS(1)=-DST
      DC(1)=-DCT
      DO 145 I=2,10
      DS(I)=DS(I-1)+DST
      DC(I)=DC(I-1)+DCT
 145  CONTINUE
C-----START LOOP THROUGH THE N CHARACTERS OF ITXT
      XO=0.0
      YO=0.0
      DO 370 J=1,N
      ITXTJ=ICHAR(ITXT(J))
      if (itxtj.ge.97.and.itxtj.le.122) then
          ich=itxtj-32
          go to 221
      end if
C-----MASK IT TO SIX BITS AND ADD ONE. PICK UP POINTER AND COUNTER
 220  ICH=MOD(ITXTJ,64)+1
 221  IP=IPTR(ICH)
      NK=NKNT(ICH)
C-----TEST FOR SPACE OR UNDEFINED CHARACTER
      IF(NK.EQ.0) GO TO 360
C-----START LOOP THROUGH SEGMENTS OF CHARACTER. LIFT PEN INITIALLY
      IPEN=3
      DO 350 K=1,NK
      IXY=IXYT(IP)
C-----LIFT PEN IF SPECIAL INDICATOR IS FOUND
      IF(IXY.NE.99) GO TO 300
      IPEN=3
      GO TO 340
 300  IX=IXY/10
      IY=IXY-10*IX
      DX=XO+DC(IX)-DS(IY)
      DY=YO+DC(IY)+DS(IX)
      CALL DRAW(W,DX,DY,IPEN)
C-----PUT PEN DOWN TO DRAW NEXT SEGMENTS
      IPEN=2
```

```
  340 IP=IP+1
  350 CONTINUE
C-----MOVE ORIGIN TO NEXT CHARACTER POSITION
  360 XO=XO+DC(8)
      YO=YO+DS(8)
  370 CONTINUE
      RETURN
c *** Only one centered symbol (*) is available in ORTEP-III.
C-----PLOT ONE SPECIFIC CENTERED SYMBOL. SET UP TABLE OF INCREMENTS
  400 DCT=HGT/2.0
      DC(1)=-DCT
      DC(2)= 0.0
      DC(3)= DCT
C-----MOVE TO SYMBOL WITH PEN UP OR DOWN, DEPENDING ON N
      IPEN=3
      IF(N.LE.-2) IPEN=2
C-----LOOP THROUGH SEGMENTS OF CENTERED SYMBOL
      DO 440 K=337,349
      IXY=IXYT(K)
      IX=IXY/10
      IY=IXY-10*IX
      CALL DRAW(W,DC(IX),DC(IY),IPEN)
C-----PUT PEN DOWN TO DRAW REMAINING SEGMENTS
      IPEN=2
  440 CONTINUE
      RETURN
      END
      SUBROUTINE SPARE(INST)
C     ***** THIS SUBROUTINE MAY BE USED FOR NEW INSTRUCTIONS *****
      RETURN
      END
      SUBROUTINE STOR(TD1)
C     ***** STORE IN OR REMOVE FROM ATOMS ARRAY *****
      REAL*8 TD1
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      IF(LATM)481,481,450
  450 IF(500-LATM)455,455,460
  455 IF(NJ2-10)490,490,460
  460 L=LATM
C     ***** CHECK FOR POSITIONAL DUPLICATION *****
      DO 480 K=1,L
      DO 465 J=1,3
      IF(ABS(V1(J)-ATOMS(J,K))-0.001)465,465,480
  465 CONTINUE
      IF(NJ2-10)490,490,470
C     ***** ATOM REMOVAL BY TABLE PUSHDOWN *****
  470 LATM=LATM-1
      DO 475 I=K,LATM
      ATOMID(I)=ATOMID(I+1)
      DO 475 J=1,3
  475 ATOMS(J,I)=ATOMS(J,I+1)
      GO TO 490
  480 CONTINUE
  481 IF(NJ2-10)482,490,490
C     ***** STORE ATOM *****
  482 IF(499-LATM)490,483,485
  483 NG=16
      CALL ERPNT (TD1,400)
  485 LATM=LATM+1
      DO 486 J=1,3
  486 ATOMS(J,LATM)=V1(J)
      ATOMID(LATM)=TD1
  490 RETURN
      END
      subroutine tepsym(txt,num,kk)
c *** parses character string representation of symmetry operators
c *** and stores information
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      character*24 txt
      logical twodig

c *** convert txt to upper case
      do 101 i=1,24
         iascii=ichar(txt(i:i))
         if (iascii.ge.97.and.iascii.le.122) txt(i:i)=char(iascii-32)
  101 continue

c *** default value of ts if no fraction specified
  202 ts(kk,num)=0.

c *** look for and interpret a/b style fraction
      n=index(txt,'/')
      if (n.gt.0) then

c        get denominator
         k=ichar(txt(n+1:n+1))-48
         m=ichar(txt(n+2:n+2))-48
         if (m.ge.0 .and. m.le.9) then
            iden=k * 10 + m
         else
            iden=k
         end if

c        get numerator
         twodig=.false.
         ksign=1
         k=ichar(txt(n-1:n-1))-48
         if (n-2.ge.1) then
            m=ichar(txt(n-2:n-2))-48
            if (m.ge.0 .and. m.le.9) twodig=.true.
            if (txt(n-2:n-2).eq.'-') ksign=-1
            if (n-3.ge.1) then
               if (txt(n-3:n-3).eq.'-') ksign=-1
            end if
```

```
         end if
         if (twodig) then
            inum=ksign * (m * 10 + k)
         else
            inum=ksign * k
         end if

         ts(kk,num)=float(inum)/float(iden)
      end if

c *** look for and interpret decimal style fraction
      n=index(txt,'.')
      if (n.gt.0) then

c        get post decimal point portion
         k=ichar(txt(n+1:n+1))-48
         m=ichar(txt(n+2:n+2))-48
         if (m.ge.0 .and. m.le.9) then
            ts(kk,num)=float(k) * .1 + float(m) * .01
         else
            ts(kk,num)=float(k) * .1
         end if

c        get sign
         ksign=1
         if (n-1.ge.1) then
            if (txt(n-1:n-1).eq.'-') ksign=-1
         end if
         if (n-2.ge.1) then
            if (txt(n-2:n-2).eq.'-') ksign=-1
         end if

         ts(kk,num)=float(ksign) * ts(kk,num)
      end if

c *** interpret xyz portion of symmetry operation
      do 303 i=1,24
         if (txt(i:i).eq.'X') then
            fs(1,kk,num)=1.
            if (i.ge.2) then
               if(txt(i-1:i-1).eq.'-') fs(1,kk,num)=-1.
            end if
         end if
         if (txt(i:i).eq.'Y') then
            fs(2,kk,num)=1.
            if (i.ge.2) then
               if(txt(i-1:i-1).eq.'-') fs(2,kk,num)=-1.
            end if
         end if
         if (txt(i:i).eq.'Z') then
            fs(3,kk,num)=1.
            if (i.ge.2) then
               if(txt(i-1:i-1).eq.'-') fs(3,kk,num)=-1.
            end if
         end if
303   continue

      return
      end
      SUBROUTINE TMM(X,Y,Z)
C     Z = TRANSPOSED (TRANSPOSE(X) * (Y) )
C     Z(3,3)=X(3,3)*Y(3,3)
      DIMENSION X(3,3),Y(3,3),Z(3,3)
      X11=X(1,1)
      X12=X(1,2)
      X13=X(1,3)
      X21=X(2,1)
      X22=X(2,2)
      X23=X(2,3)
      X31=X(3,1)
      X32=X(3,2)
      X33=X(3,3)
      Y11=Y(1,1)
      Y12=Y(1,2)
      Y13=Y(1,3)
      Y21=Y(2,1)
      Y22=Y(2,2)
      Y23=Y(2,3)
      Y31=Y(3,1)
      Y32=Y(3,2)
      Y33=Y(3,3)
      Z(1,1)=X11*Y11+X21*Y21+X31*Y31
      Z(1,2)=X12*Y11+X22*Y21+X32*Y31
      Z(1,3)=X13*Y11+X23*Y21+X33*Y31
      Z(2,1)=X11*Y12+X21*Y22+X31*Y32
      Z(2,2)=X12*Y12+X22*Y22+X32*Y32
      Z(2,3)=X13*Y12+X23*Y22+X33*Y32
      Z(3,1)=X11*Y13+X21*Y23+X31*Y33
      Z(3,2)=X12*Y13+X22*Y23+X32*Y33
      Z(3,3)=X13*Y13+X23*Y23+X33*Y33
      RETURN
      END
      subroutine uinput(in,nout)
c *** user input routine
      common /ns/ npf,ndraw,norient,nvar
      common /dfl/ infile,idraw,iorient,iout,ext,atomfi,fpaplen
      character*60 fname,txtans,infile,atomfi
      character*4 ext
      character*1 answer

      call dflts
      iflag=0
c *** get the input file name and open the file or "exit" ***
      if (iargc().eq.1) then
         call getarg(1,fname)
         open(in,file=fname,status='old',err=110)
         go to 135
      end if
  110 fname=infile
      ipos=index(fname,' ')
      write (*,115) fname(1:ipos-1)
  115 format(' Enter instruction set file name or "exit" [',a,']:  ',$)
      read (*,120) txtans
  120 format(a)
      if (txtans(1:4).eq.'exit' .or. txtans(1:4).eq.'EXIT')
     *   call exitng(0)
      if (txtans(1:1) .ne. ' ') fname=txtans
      open(in,file=fname,status='old',err=125)
      go to 135
  125 ipos=index(fname,' ')
      fname=fname(1:ipos-1)
      write (*,130) fname(1:ipos-1)
```

```
  130 format(/' "',a,'" does not exist'/)
      go to 110
c *** determine where ortep drawing should go ***
  135 write (*,140) idraw
  140 format(' Drawing to (1) Screen, (2) Postscript file, (3) HPGL file
     *, or (0) Omit [',i1,']:  ',$)
      read (*,145) answer
  145 format(a)
      if (answer .eq. ' ') then
         ndraw=idraw
      else
         read (answer,155) ndraw
      end if
      if (ndraw.lt.0.or.(ndraw.gt.3.and.ndraw.ne.9)) then
         write(6,*) 'invalid selection'
         go to 135
      end if
      if (ndraw.eq.0.or.ndraw.eq.1.or.ndraw.eq.9) go to 149
      go to 1451
c *** need to get this information if printing from editor
      entry getpap
      iflag=1
c *** determine orientation of drawing ***
 1451 write (*,1452) iorient
 1452 format(' (1) Portrait or (2) Landscape orientation [',i1,']:  ',$)
      read (*,145) answer
      if (answer .eq. ' ') then
         norient=iorient
      else
         read (answer,155) norient
      end if
      if (norient.lt.1.or.norient.gt.2) then
         write(6,*) 'invalid selection'
         go to 1451
      end if
c *** determine paper length for postscript landscape
      if (ndraw.eq.2.and.norient.eq.2) then
         write (*,1453) fpaplen
 1453    format(' How tall is printer page in inches? [',f5.2,']:  ',$)
         read (*,120) txtans
         if (txtans(1:1) .ne. ' ') read (txtans,1454) fpaplen
 1454    format(f10.0)
         nvar=fpaplen*1000.
      end if
c *** if called from recycle, return there
      if (iflag.eq.1) return
c *** determine where ortep output should go ***
  149 write (*,150) iout
  150 format(' Text output to (1) File, (2) Screen, or (0) Omit [',i1,
     *       ']:  ',$)
      read (*,145) answer
      if (answer .eq. ' ') then
      nou=iout
      else
      read (answer,155) nou
      end if
  155 format(i1)
c *** set output unit number ***
      nout=-4
      if (nou .eq. 1) nout=4
      if (nou .eq. 2) nout=6
c *** if output goes to a file; get its name and open the file ***
      if (nout .eq. 4) then
         ipos=index(fname,'.')
         if(ipos .ne. 0) then
            fname=fname(1:ipos-1)//ext
            go to 160
         end if
         ipos=index(fname,' ')
         fname=fname(1:ipos-1)//ext
  160    ipos=index(fname,' ')
         write (*,165) fname(1:ipos-1)
  165    format(' Enter output file name [',a,']:  ',$)
         read (*,120) txtans
         if (txtans(1:1) .ne. ' ') fname=txtans
         open(nout,file=fname,status='old',err=170)
         go to 175
  170    open(nout,file=fname,status='new')
      end if
  175 continue
      return
c *** get file name of an external file containing atomic parameters
      entry gtafil(iu)
  210 fname=atomfi
      ipos=index(fname,' ')
      write (*,215) fname(1:ipos-1)
  215 format(' Enter atom parameter file name or "exit" [',a,']:  ',$)
      read (*,120) txtans
      if (txtans(1:4).eq.'exit' .or. txtans(1:4).eq.'EXIT')
     *    call exitng(0)
      if (txtans(1:1) .ne. ' ') fname=txtans
      open(iu,file=fname,status='old',err=225)
      return
  225 ipos=index(fname,' ')
      write (*,130) fname(1:ipos-1)
      go to 210
c *** ask user about using editor
      entry go2edtr
      write (*,303)
c 303 format(/,' Edit instruction set? (Y)es or (N)o [N]:  ',$)
  303 format(/,' (1) Save drawing as Postscript file',/,
     &' (2) Save drawing as HPGL file',/,
     &' (3) Redraw structure on screen',/,
     &' (4) Edit instruction set',/,
     &' [Quit]:  ',$)
      read (*,304) answer
  304 format(a)
      if (answer.eq.'1') then
         ndraw=2
         call recycle
      else
         if (answer.eq.'2')  then
            ndraw=3
            call recycle
         else
            if (answer.eq.'3') then
               ndraw=1
               call recycle
            else
               if (answer.eq.'4') call editr
            end if
         end if
```

```
      end if
      return
      end
      SUBROUTINE UNITY(X,Z,ITYPE)
      DIMENSION X(3),Y(3),Z(3)
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      Y(1)=X(1)
      Y(2)=X(2)
      Y(3)=X(3)
      IF(ITYPE)125,125,105
  105 T1=SQRT(Y(1)*Y(1)+Y(2)*Y(2)+Y(3)*Y(3))
      GO TO 145
  125 T1=SQRT(Y(1)*(Y(1)*AA(1,1)+Y(2)*(AA(1,2)+AA(2,1))+Y(3)*(AA(1,3)+A
     1A(3,1)))+Y(2)*(Y(2)*AA(2,2)+Y(3)*(AA(2,3)+AA(3,2)))+Y(3)*Y(3)*AA(3
     2,3))
  145 IF(T1)155,155,175
  155 NG=5
      GO TO 300
  175 Z(1)=Y(1)/T1
      Z(2)=Y(2)/T1
      Z(3)=Y(3)/T1
  300 RETURN
      END
      SUBROUTINE VM(Y,X,Z)
C     TRANSPOSED VECTOR * MATRIX
C     Z(3)=Y(3)*X(3,3)
      DIMENSION X(3,3),Y(3),Z(3)
      Y1=Y(1)
      Y2=Y(2)
      Y3=Y(3)
      Z(1)=X(1,1)*Y1+X(2,1)*Y2+X(3,1)*Y3
      Z(2)=X(1,2)*Y1+X(2,2)*Y2+X(3,2)*Y3
      Z(3)=X(1,3)*Y1+X(2,3)*Y2+X(3,3)*Y3
      RETURN
      END
      FUNCTION VMV(X1,Q,X2)
C     TRANSPOSED VECTOR * MATRIX * VECTOR
C     VMV=X1(3)*Q(3,3)*X2(3)    TO  EVALUATE QUADRATIC OR BILINEAR FORM
      DIMENSION X1(3),Q(3,3),X2(3)
      VMV=X1(1)*(X2(1)*Q(1,1)+X2(2)*Q(1,2)+X2(3)*Q(1,3))
     &   +X1(2)*(X2(1)*Q(2,1)+X2(2)*Q(2,2)+X2(3)*Q(2,3))
     &   +X1(3)*(X2(1)*Q(3,1)+X2(2)*Q(3,2)+X2(3)*Q(3,3))
      RETURN
      END
      FUNCTION VV(X,Y)
C     TRANSPOSED VECTOR * VECTOR
C     VV=X(3)*Y(3)
      DIMENSION X(3),Y(3)
      VV=X(1)*Y(1)+X(2)*Y(2)+X(3)*Y(3)
      RETURN
      END
      SUBROUTINE XYZ(DQA,X,ITYPE)
C     ***** ITYPE .GT.0 CART. COORD. FROM ATOM CODE WORD *****
C     ***** XABSF(ITYPE) .LE.2 FOR WORKING SYSTEM *****
C     ***** XABSF(ITYPE) .GT.2 FOR REFERENCE SYSTEM *****
C     ***** ITYPE .LE.0 USES TRICLINIC COORD. XT *****
      DIMENSION X(3)
      REAL*8 DQA
      REAL*8 AIN,ATOMID
      CHARACTER*4 TITLE,TITLE2
      COMMON NG,A(9),AA(3,3),AAREV(3,3),AAWRK(3,3),AID(3,3)
     1 ,AIN(140),ATOMID(500),ATOMS(3,500),BB(3,3),BRDR,CD(8,20)
     2 ,CONT(5),D(3,130),DA(3,3),DP(2,130),DISP,EDGE,FORE,FS(3,3,96)
     3 ,IN,ITILT,KD(5,20),LATM,NATOM,NCD,NJ,NJ2,NOUT,NSR,NSYM
     4 ,ORGN(3),PAC(3,5),PAT(3,3),Q(3,3),REFV(3,3),RES(4),RMS(5),SCAL1
     5 ,SCAL2,SCL,SYMB(3,3),TAPER,THETA,TITLE(18),TITLE2(18),TS(3,96)
     6 ,VIEW,VT(3,4),V1(4),V2(3),V3(3),V4(3),V5(3),V6(3),WRKV(3,3)
     7 ,XLNG(3),XO(3),XT(3)
      IT=IABS(ITYPE)-2
      NG1=NG
      NG=0
      IF(ITYPE)10,10,5
    5 CALL ATOM(DQA,XT)
      IF(NG)30,10,30
   10 T1=0.
      DO 15 J=1,3
      T2=XT(J)-ORGN(J)
      V1(J)=T2
   15 T1=T1+ABS(T2)
      IF(T1-.0001)20,20,40
   20 NG=NG1
   30 DO 35 J=1,3
   35 X(J)=0.
      GO TO 300
   40 IF(IT)45,45,60
C     ***** RELATIVE TO WORKING SYSTEM *****
   45 DO 55 I=1,3
      T1=0.
      DO 50 J=1,3
   50 T1=T1+V1(J)*AAWRK(J,I)
   55 X(I)=T1*SCAL1
      GO TO 300
C     ***** RELATIVE TO REFERENCE SYSTEM *****
   60 DO 70 I=1,3
      T1=0.
      DO 65 J=1,3
   65 T1=T1+V1(J)*AAREV(J,I)
   70 X(I)=T1*SCAL1
  300 RETURN
      END
c ****************************************************************
c *** DUMMY SCREEN OUTPUT (MAY BE REPLACED WITH SCREEN DRIVER CODE)
c ****************************************************************

c     subroutine initsc
c     return
c     end

c     subroutine penwsc(penw)
c     return
c     end

c     subroutine colrsc(icolor)
```

```
c     return
c     end

c     subroutine pensc(x,y,ipen)
c     return
c     end

c     subroutine endsc
c     return
c     end

c *** end of dummy screen output
c ****************************************************************

c ****************************************************************
c *** PGPLOT CODE FOR SCREEN OUTPUT
c *** if PGPLOT is implemented, use the subroutines here
c *** instead of the ones in the DUMMY SCREEN OUTPUT section
c     PGPLOT is a free graphics library developed by T. J. Pearson at
c     the California Institute of Technology. Information about PGPLOT
c     can be found on the World Wide Web at
c     http://astro.caltech.edu/~tjp/pgplot
c     or via e-mail to tjp@astro.caltech.edu.
c ****************************************************************

      subroutine initsc
      character*10 outdev
      common /ns/ npf,ndraw,norient,nvar
      integer pgbeg

      xwid=11.
      yhgt=8.5

c *** The following is for PGPLOT on an X-windows system.
      outdev = '/XWINDOW'
c *** The following is for PGPLOT on an MS-DOS system.
c     outdev = '/MS'
c *** The following is for PGPLOT on a Macintosh system.
c     outdev = '/MAC'

      open(npf,status='scratch')

      if (pgbeg(0,' ',1,1) .ne. 1) call exitng(8)

c     switch black and white
      call pgscr(0,1.,1.,1.)
      call pgscr(1,0.,0.,0.)

c     set up drawing window
      call pgpage
      call pgqch(osize)
      call pgsch(0.)
      call pgvstd
      call pgwnad(0.,xwid,0.,yhgt)
      call pgsch(osize)
      call pgbox('BCT',1.,0,'BCT',1.,0)
      call pgsci(1)
      call pgsfs(2)
      call pgrect(10.4,11.,8.2,8.5)
      call pgtext(10.5,8.3,'EXIT')
      return
      end

      subroutine colrsc(icolor)
c *** set plot color
c *** in ORTEP icolor=0 => black
c *** PGPLOT is set up for 1=black
      common /ns/ npf,ndraw,norient,nvar
      icol=icolor
      if (icol.eq.0) icol=1
      nvar=icol
      if (ndraw.eq.1) call pgsci(icol)
      if (ndraw.eq.9) write (npf,111) icol
  111 format('COL',1x,i2)
      return
      end

      subroutine penwsc(penw)
c *** change pen width
c *** PGPLOT measures pen width in 200ths of an inch
      common /ns/ npf,ndraw,norient,nvar
      ipenw=nint(.001*penw*200.)
      if (ipenw.le.0) ipenw=1
      if (ipenw.gt.200) ipenw=200
      if (ndraw.eq.1) call pgslw(ipenw)
      if (ndraw.eq.9) write (npf,111) ipenw
  111 format('WID',1x,i3)
      return
      end

      subroutine pensc(x,y,ipen)
c *** move the pen
      common /trfac/ xtrans,ytrans
      common /ns/ npf,ndraw,norient,nvar

      if (ipen.eq.2) then
         if (ndraw.eq.1) call pgdraw(x+xtrans,y+ytrans)
         if (ndraw.eq.9) write (npf,111) x+xtrans,y+ytrans
  111    format('LIN',2(1x,f10.6))
      end if
      if (ipen.eq.3) then
         if (ndraw.eq.1) call pgmove(x+xtrans,y+ytrans)
         if (ndraw.eq.9) write (npf,112) x+xtrans,y+ytrans
  112    format('MOV',2(1x,f10.6))
      end if

      return
      end

      subroutine endsc
      common /ns/ npf,ndraw,norient,nvar

      call curssc
      close(npf)

c *** tell user to hit <enter> key
      call pgsci(0)
      call pgsfs(1)
      call pgrect(7.5,11.,8.2,8.5)
      call pgsci(1)
      call pgsfs(2)
```

```
      call pgrect(7.5,11.,8.2,8.5)
      call pgsci(1)
      call pgtext(7.6,8.3,'Hit <RETURN> or <ENTER> key')

      call pgend

      return
      end

      subroutine curssc
c *** correlate screen cursor position with atom positions and display results
      character ch
      character*21 str
      integer pgcurs
      character*6 label,alabel
      character*9 tomid,atomid
      common /trfac/ xtrans,ytrans
      common /ns/ npf,ndraw,norient,nvar

      call pgsfs(1)
      call pgscf(1)
      call pgsch(1.)

c *** get cursor position
  1   junk = pgcurs(x,y,ch)

      if (ch.eq.'x' .or. ch.eq.'X') return
      if (ch.eq.'d' .or. ch.eq.'D') return
      if (x.ge.10.4 .and. x.le.11. .and. y.ge.8.2 .and. y.le.8.5) return
      if (ichar(ch).eq.13) return

c *** initial values for variables
      xpt = x
      ypt = y
      adiffx = .0625
      adiffy = .0625
      odiffx = adiffx
      odiffy = adiffy
      atomid = '
      alabel = '
      iflag = 0
      nflag = 0

      rewind(npf)

  2   read(npf,3,end=4) label,tomid,xx,yy
  3   format(11x,a6,3x,a9,4x,2f8.0)
      diffx = abs(xx-xpt)
      diffy = abs(yy-ypt)
      if (diffx.le.adiffx .and. diffy.le.adiffy) nflag=nflag+1
      if (diffx.le.odiffx .and. diffy.le.odiffy) then
         atomid = tomid
         alabel = label
         odiffx = diffx
         odiffy = diffy
      end if
      go to 2

  4   if (nflag.eq.0) write(str,5)
      if (nflag.eq.1) write(str,6) alabel,atomid
      if (nflag.gt.1) write(str,7) alabel,atomid

  5   format('Not near atom center')
  6   format(a6,1x,a9)
  7   format(a6,1x,a9,' + ??')

c *** erase rectangle
      call pgsci(0)
      call pgsfs(1)
      call pgrect(0.,2.8,8.2,8.5)
c *** redraw empty rectangle
      call pgsci(1)
      call pgsfs(2)
      call pgrect(0.,2.8,8.2,8.5)

c *** print atom information in rectangle
      call pgtext(0.1,8.3,str)

      go to 1

      end

c *** end of PGPLOT specific routines
c ********************************************************************

c ********************************************************************
c *** HPGL FILE OUTPUT
c ********************************************************************

      subroutine inithp
      common /ns/ npf,ndraw,norient,nvar
      character ESC
      character*10 fname

      do 11 i=1,999
         write (fname, 10) i
 10      format('TEP',i3.3,'.PRN')
         open(unit=npf,file=fname,status='old',err=12)
         close(npf)
 11   continue
 12   open(unit=npf,file=fname,status='new')
      write (*,13) fname
 13   format(/,' HPGL file name:  ',a)

      ESC=char(27)

      write (npf,21) ESC
      write (npf,22) ESC
      write (npf,23)
      if (norient.eq.2) write (npf,24)
 21   format(a1,'E')
 22   format(a1,'%0B')
 23   format('IN;'/'SP1;'/'PW.15;')
 24   format('RO90.;')
      return
      end

      subroutine colrhp(icolor)
      common /ns/ npf,ndraw,norient,nvar
c *** set plot color
c *** in ORTEP icolor=0 => black
c *** plotter pen 1=black
      icol=icolor
```

```
      if (icol.eq.0) icol=1
      write (npf,21) icol
   21 format('SP',i1,';')
      return
      end

      subroutine penwhp(penw)
      common /ns/ npf,ndraw,norient,nvar
      if (penw.eq.0.) then
         penw=.15
      else
         penw=penw*.0252
      end if
      write (npf,21) penw
   21 format('PW',f5.2,';')
      return
      end

      subroutine penhp(x,y,ipen)
      common /ns/ npf,ndraw,norient,nvar
      common /trfac/ xtrans,ytrans

      ix = nint((x + xtrans) * 1000.)
      iy = nint((y + ytrans) * 1000.)

      if (ipen.eq.2) write (npf,101) ix,iy
  101 format('PD',i4,',',i4,';')
      if (ipen.eq.3) write (npf,102) ix,iy
  102 format('PU',i4,',',i4,';')
      return
      end


      subroutine endhp
      common /ns/ npf,ndraw,norient,nvar
      character ESC

      ESC=char(27)
      write (npf,31)
   31 format('PU;',/,'SP0;',/,'PG;',/,'IN;')
      write (npf,34) ESC
   34 format(a1,'%0A')
      write (npf,35) ESC
   35 format(a1,'E')
      close(npf)
      return
      end

c *** end of HPGL specific routines
c ****************************************************************

c ****************************************************************
c *** POSTSCRIPT FILE OUTPUT
c ****************************************************************

      subroutine initps
      common /ns/ npf,ndraw,norient,nvar
      common /ps/ ixmin,ixmax,iymin,iymax,ixt,iyt
      character*10 fname

c *** initialize variables to calculate bounding box

      ixmin=20000
      ixmax=0
      iymin=20000
      iymax=0

      do 11 i=1,999
         write (fname, 10) i
   10    format('TEP',i3.3,'.PRN')
         open(unit=npf,file=fname,status='old',err=12)
         close(npf)
   11 continue
   12 open(unit=npf,file=fname,status='new')
      write (*,13) fname
   13 format(/,' Postscript file name:  ',a)

      ixt=0
      iyt=0
      write (npf,21)
      write (npf,22)
      write (npf,23)
      if (norient.eq.2) then
         write (npf,24)
         iyt=nvar
      else
         write (npf,25)
      end if
      write (npf,26)
      write (npf,27)
      write (npf,28)
      write (npf,29)
      write (npf,30)
      write (npf,31)
      write (npf,32) ixt,iyt
      if (norient.eq.2) write (npf,33)
      write (npf,34)
      write (npf,35)
   21 format('%!PS-Adobe-3.0 EPSF-3.0')
   22 format('%%Creator: ORTEP-III')
   23 format('%%BoundingBox: (atend)',/,'%%Pages: 1')
   24 format('%%Orientation: Landscape')
   25 format('%%Orientation: Portrait')
   26 format('%%BeginProlog')
   27 format('/m {moveto} def')
   28 format('/l {lineto} def')
   29 format('%%EndProlog',/,'%%Page: 1 1')
   30 format('%%BeginPageSetup')
   31 format('0.072 0.072 scale')
   32 format(i6,1x,i6,' translate')
   33 format('-90 rotate')
   34 format('0 setgray 1 setlinecap 5 setlinewidth')
   35 format('%%EndPageSetup')
      return
      end

      subroutine colrps(icolor)
      common /ns/ npf,ndraw,norient,nvar
      write(npf,101)
  101 format('stroke')
      if (icolor.eq.0) write(npf,1)
      if (icolor.eq.1) write(npf,1)
      if (icolor.eq.2) write(npf,2)
```

```
      if (icolor.eq.3) write(npf,3)
      if (icolor.eq.4) write(npf,4)
      if (icolor.eq.5) write(npf,5)
      if (icolor.eq.6) write(npf,6)
      if (icolor.eq.7) write(npf,7)
    1 format('0 setgray')
    2 format('1 0 0 setrgbcolor')
    3 format('0 1 0 setrgbcolor')
    4 format('0 0 1 setrgbcolor')
    5 format('0 1 1 setrgbcolor')
    6 format('1 0 1 setrgbcolor')
    7 format('1 1 0 setrgbcolor')
      return
      end

      subroutine penwps(penw)
      common /ns/ npf,ndraw,norient,nvar
      write(npf,101)
  101 format('stroke')
      if (penw.eq.0.) penw=5.
      write(npf,102) penw
  102 format(f10.2,1x,'setlinewidth')
      return
      end

      subroutine penps(x,y,ipen)
      common /trfac/ xtrans,ytrans
      common /ns/ npf,ndraw,norient,nvar
      common /ps/ ixmin,ixmax,iymin,iymax,ixt,iyt

      ix = nint((x + xtrans) * 1000.)
      iy = nint((y + ytrans) * 1000.)

      if (ipen.eq.2) write (npf,101) ix,iy
  101 format(i6,1x,i6,1x,'l')
      if (ipen.eq.3) write (npf,102) ix,iy
  102 format('stroke'/i6,1x,i6,1x,'m')

c *** variables to calculate bounding box
      if (ix.lt.ixmin) ixmin=ix
      if (ix.gt.ixmax) ixmax=ix
      if (iy.lt.iymin) iymin=iy
      if (iy.gt.iymax) iymax=iy

      return
      end

      subroutine endps
      common /ns/ npf,ndraw,norient,nvar
      common /ps/ ixmin,ixmax,iymin,iymax,ixt,iyt

      write (npf,25)
   25 format('stroke'/'showpage')

c *** calculate bounding box
      if (norient.eq.1) then
         ixmn=float(ixmin+ixt)*.072 - 2
         iymn=float(iymin+iyt)*.072 - 2
         ixmx=float(ixmax+ixt)*.072 + 2
         iymx=float(iymax+iyt)*.072 + 2
      else
         ixmn=float(ixt+iymin)*.072 - 2
         iymn=float(iyt-ixmax)*.072 - 2
         ixmx=float(ixt+iymax)*.072 + 2
         iymx=float(iyt-ixmin)*.072 + 2
      end if
      if (ixmn.lt.0) ixmn=0
      if (iymn.lt.0) iymn=0

c *** put bounding box at end of postscript file
      write (npf,26) ixmn,iymn,ixmx,iymx
   26 format('%%BoundingBox: ',4(i6,1x))

      write (npf,27)
   27 format('%%Trailer'/'%%EOF')
      close(npf)

      return
      end

c *** end of Postscript specific routines
c ********************************************************************

c ********************************************************************
c *** DEFAULT VALUES FOR ORTEP INPUT AND OUTPUT OPTIONS ARE SET HERE
c *** USED IN SUBROUTINE UINPUT
c ********************************************************************

      subroutine dflts
      common /dfl/ infile,idraw,iorient,iout,ext,atomfi,fpaplen
      character*60 infile,atomfi
      character*4 ext

c *** name of default input file
      infile='TEP.IN'

c *** where ortep drawing output should go
c *** 1: Screen, 2: Postscript file, 3: HPGL file
      idraw=1

c *** orientation of drawing
c *** 1: portrait, 2: landscape
      iorient=1

c *** height of page
      fpaplen=11.

c *** where ortep text output should go
c *** 1: file, 2: screen, 0: omit
      iout=0

c *** text output filename extension
      ext='.out'

c *** default name of external atom parameter file
      atomfi='ATOMS.DAT'

      return
      end
```

INTERNAL DISTRIBUTION

1. J. C. Bryan
2. G. J. Bunick
3-33. M. N. Burnett
34. P. D. Butler
35. B. C. Chakoumakos
36. P. Dai
37. J. A. Fernandez-Baca
38. A. A. Gakh
39. A. Habenschuss
40. W. A. Hamilton
41. J. B. Hayter
42. C. R. Hubbard
43-73. C. K. Johnson
74. S. A. McLuckey
75. H. A. Mook
76. S. E. Nagler
77. M. L. Poutsma
78. J. M. Ramsey
79. J. L. Robertson
80. S. Spooner
81. G. D. Wignall
82. M. Yethiraj
83. J. L. Zarestky
84. Central Research Library
85. Y-12 Technical Library
86. Laboratory Records - RC
87-686. ORTEP-III Document Distribution

EXTERNAL DISTRIBUTION

687-688. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831
689. B. R. Vincent, Molecular Structure Corporation, 3200 Research Forest Drive, The Woodlands, TX 77381
690. G. G. Johnson, Jr., Penn State University
691. A. Linden, Organic Chemistry Institute, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
692. J. V. Silverton, NHLBI/LBC, 10/7N-307, 10 Center Dr MSC 1676, Bethesda, MD 20892-1676
693. M. Kroeker, Inst. f. Organ. Chemie, Univ. (TH) Darmstadt, Germany
694. W. Poll, Heinrich-Heine-Universitaet Duessseldorf, Institut fuer Anorganische Chemie und Strukturchemie, Universitaetsstrasse 1, D-40225 Duesseldorf, Germany
695. K A. Byriel, Department of Chemistry, The University of Queensland, Brisbane, QLD. 4072, Australia
696. B. Dalhus, Dept. of Chemistry, University of Oslo, PO Box 1033, Blindern, N-0315 Oslo, Norway
697. Bill Harrison, Department of Chemistry, University of Western Australia, Nedlands, WA 6907, Australia
698. J. Bollinger, Department of Chemistry, Molecular Structure Center, Indiana University, Bloomington, IN 47405
699. J. C. Huffman, Department of Chemistry, Molecular Structure Center, Indiana University, Bloomington, IN 47405
700. K. M. Crennell, 'Fortran Friends', P.O.Box 64, Didcot, Oxon OX11 0TH