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.