Unrestricted distrib.
UTIG Tech. Rep.[1] # 128

Users Manual for *TexFlex-0.5:* the Texas Flexible, Practical Program Package for Locating Seismic Events

by:

Cliff Frohlich
cliff@utig.ig.utexas.edu
Institute for Geophysics
8701 North Mopac
University of Texas at Austin
Austin, Texas 78759

TexFlex is a software package developed as part of a research project administered by Phillips Laboratory, Hanscom Air Force Base under Contract No. F19628-91-K-0026.

University of Texas Institute for Geophysics Technical Report No. 128

September 21, 1993

How to Obtain the TexFlex Relocation Software from the University of Texas Using Anonymous FTP:

1. Change to a directory where you will store this program within a subdirectory named "TexFlex"
2. Type "ftp utig.ig.utexas.edu"
3. Login as "anonymous" with the password being your email address
4. Change directory to pub - type "cd pub"
5. Type "prompt", then "binary", then "!mkdir TexFlex"
6. Type "mget TexFlex/*"
7. Type "quit"
8. Now, the program, sample input, and sample output are in your directory.

In summary, examples of the screen commands are:

```
cd /utig/usr3/cliff
ftp utig.ig.utexas.edu
anonymous
cliff@utig.ig.utexas.edu
cd pub
prompt
binary
!mkdir TexFlex
mget TexFlex
quit
```

TABLE OF CONTENTS

5.7.4**function **AreaTri(***rA, rB, rC***)**

5.7.5**subroutine **fndPlane(***r1, r2, 43, p, c***)**

5.7.6**subroutine **edgePlane(***p, c, kbeg, kend, nF, kchange***)**

5.7.7**subroutine **newPlane(***k1, k2, k3, nF, new***)**

5.7.8**subroutine **prntStuff(***norder, p, c, View***)**

5.7.9**subroutine **vecdif(***r1, r2, r12***)**

5.7.10**subroutine **vecassign(***k1, k2, k3, r1, r2, r3***)**

5.7.11**subroutine **rePoly(***np***)**

5.7.12**subroutine **indexx(***n, kcoord, indx***)**

5.7.13**subroutine **intord(***j1, j2, j3, k1, k2, k3***)**

5.7.14**subroutine **ran1(***idummy***)**

SECTION 1: Introduction and Brief Overview of the **TexFlex** Programs

1.1 What is **TexFlex**?

TexFlex is a computer program written in Fortran language and designed to find improved locations for seismic events. Basically, the user provides preliminary estimates of event locations and observed arrival times of seismic phases, and the program finds improved locations.

1.2 Why Another Hypocenter Determination Program?

In the development of **TexFlex**, the primary design objective has been to build a computer program which meets all of or at least most of the needs of the typical research seismologist. The intent has been to design a program which is flexible yet simple and modular, i. e., flexible enough so that it performs most of the relocation tasks that a typical project in seismology requires, yet simple enough so that inexperienced users can implement it without modification. At the same time, **TexFlex** is modular enough so that more knowledgeable users can easily modify it by adding subroutines, yet simple enough so that the inexperienced or busy seismologist doesn't have to understand the whole program to use it successfully.

A typical research effort involving the observation and location of seismic events consists of several steps. For example, these steps might be:

- Data collection
- Initial catalog preparation
- Determination of station corrections, "best" events, and a preferred velocity model
- Final catalog preparation
- Error analysis and comparison with previous event locations

Each of these phases requires software with somewhat different properties. During the data collection phase it is desirable to perform single event locations to identify inconsistent arrival times or other blunders, and to determine whether there are data gaps, *i. e.*, an absence of phase arrivals from particular important azimuths. When a catalog of all phase arrivals and preliminary locations is available, repeated application of joint hypocenter determination (JHD) methods to a smaller subset of especially well-recorded events recorded at the most reliable stations helps to determine a set of station corrections and a preferred velocity model.

Finally, after routine locations are complete there is often a phase where the researcher compares the routine locations to historical events, locations reported by the ISC, results of previous research, etc. This phase often involves relocating seismic events using both regional and teleseismic phase arrivals, it may include analysis using phases other than P or S, and generally involves a careful studies of both systematic and statistical errors.

Thus, a typical research project will require relocation software that will allow:

- both flat-earth and round earth locations
- both single-event and JHD relocations
- incorporation of P, S, as well as "exotic" phases such as PKP, etc.
- error analysis, and the preparation of various tables
- event-by-event data summary for catalog preparation.

Moreover, this software will be most useful if:

- it requires a minimum of file editing, etc. when it is applied to different subsets of the data
- it allows comments to be a routine part of data input and output files, so that the user can "remember" what he has learned during each tedious step of the relocation process
- it is possible to "group" station corrections so that several stations within a geographic region or in similar tectonic environments can receive the same station correction.

1.3 Special Features of **TexFlex**

Section 1.3 briefly describes a number of specific design features of the **TexFlex** program.

1.3.1 Input file structure

The author's experience is that editing input files is the most time-consuming part of most projects requiring seismic event relocation. To reduce this "wasted" time, **TexFlex** calls up several 'archive" files which contain information about phase arrival times, trial hypocenters, and choices concerning station correction variables. Moreover, a special "driver" file selects which events are to be relocated and also sets up the architecture of the relocation process. Thus, the researcher can use these "archive" files repeatedly without editing regardless of whether he or she wishes to relocate only a select few events or a large number, using either single-event, JED, or JHD relocation methods.

1.3.2 A comment about comments

A useful practical feature of **TexFlex** is that it ignores lines beginning with the character 'c' in both the archive and the driver files. This permits the user to liberally add comments to all these files. In the examples supplied with the **TexFlex** package, these comments are mostly advice helping the user to remember file formats, etc. However, in practice most users will want to add information about the ongoing research process. For example, comments in the trial hypocenter file might contain information concerning locations of hypocenters from previous runs of **TexFlex**, or locations reported by other institutions. In the phase file comments might contain information about the quality of phases which the user has read several times, or about the presence or absence of key data which may affect the location.

1.3.3 Similar input/output file structure

TexFlex produces output files which have formats which are identical or compatible with the "archive" input files. This is useful because it is often desirable to have input files which contain information determined during a previous program run. For example, one may wish to alter the trial hypocenter input file so that it uses the most recent relocations as trial hypocenters for a subsequent run. Or, one want to alter the phase input file to include information about the distance, azimuth, or travel time residual of phase arrivals.

The output files for **TexFlex** not only contain information about the just-completed run, but they also retain all comment lines and all information about seismic events which were not relocated during the just-completed run. This means that the phase, hypocenter, and station correction files can be renamed and used for a subsequent program run, even if the subsequent run involves events not relocated during the just-completed run.

1.3.4 Convenience of grouping station correction variables

A unique feature of **TexFlex** is that it allows the user to group station correction information in a straightforward fashion. When performing JED or JHD relocations, it is often desirable to assign the same station correction to more than one station. For example, if a station's name changes because it is moved a short distance, the user may with to apply the same station correction to both names. Similarly, the user may with to group station corrections so that all stations within a particular geographic region or tectonic regime receive the same station correction. Finally, one may wish to have the same station correction apply to different phases from the same station, except that the correction is "adjusted" by a different multiplicative constant for each different phase, e. g., S station corrections might be a factor 1.71 larger than P station corrections.

TexFlex allows the user considerable flexibility for handling all the above situations concerning the grouping of station corrections. The user specifies the grouping of station correction information in the station correction input file. For each station correction variable, the file specifies all the station/phase combinations in the group, the initial station correction, and the desired multiplicative constants. Then, for relocation and matrix manipulation, the program assigns the same station correction variable to all station/phase combinations in the group. Yet, on data output **TexFlex** seamlessly handles the necessary bookkeeping so that the user can identify residuals, etc., for individual station/phase combinations.

1.3.5 Choices for weighting arrival times

Most seismic event relocation programs, including **TexFlex**, use a least squares method, i. e., they attempt to find the hypocentral location which minimizes the sum of squares of travel time residuals. However, unless the accuracy of all phase arrival data is the same, and unless there is a nearly uniform azimuthal distribution of observing stations, it is almost always desirable to use a weighted least squares method. This mean that each travel time residual is multiplied by the assigned weight, and the program finds the location which minimizes the weighted sum of squares.

Because the choice of a weighting scheme depends on the particular idiosyncrasies of the data, **TexFlex** offers a range of different options. For example, the user may with to simply assign arrivals a integer quality between 0 and 9, then, set the program to assign different user-specified weights for each quality. This is especially useful if there is a good azimuthal distribution of stations, if the user has personally read all the phase arrival data, and thus has a good idea of the relative accuracy of readings of different assigned quality.

On the other hand, if the accuracy of the readings is poorly known or if most of the stations lie in a particular azimuthal sector, the user may wish to choose a different scheme. **TexFlex** allows the user to design and specify trapezoidal or Gaussean weighting functions, and to adjust weights to balance the sum of weights of phases from each azimuthal sector.

1.3.6 Choice for different travel time models

Phase arrival data for the **TexFlex** program may originate either from teleseismic stations or from a regional station network. The only choice the user must make is to specify the travel time model used to calculate travel time residuals. For teleseismic data, the user will generally specify IASPEI91 travel times [Kennett, 1991], especially if the arrivals include exotic phases such as SKP, ScP, etc. However, if all the data are P phases the user may choose Jeffreys-Bullen [1970] tables.

For data from a regional station network **TexFlex** will calculate P and S travel times using layered flat-earth models. In this case, the user can either specify a preferred flat-layered model, or use the program-supplied default models. The first program-supplied default model is a "continental" model which is essentially P-velocity model EU2 from Lerner-Lam and Jordan [1987] and a V_P/V_S ratio of 1.77; this model has layer of velocity 2.98, 6.08, and 6.78 km/sec and thicknesses 3, 17, and 16 km overlying a mantle of velocity 8.19 km/sec. The second default model is based on an "oceanic" model from the review of Purdy and Ewing [1986] for structures determined in areas not disrupted by fracture zones. This model has velocities of 5.5 and 7.0 km/sec in layers of thickness 7 and 5 km thick overlying a mantle of velocity 8.0 km/sec. Here, the top layer is about 5 km thicker than that of Purdy and Ewing [1986] as we imagine that the crust is overlain by 5 km of water, however, we seldom wish to locate seismic events using rays that traverse the water column.

1.3.7 Single-event, JED, or JHD relocation

TexFlex will perform either single-event or JHD relocations. Currently, for JHD relocation the only program option is the "fast JHD" method described by Frohlich [1979]. For both single-event and joint locations, the user can specify "fixed-depth" relocation so that focal depth is not one a free variable. If the user specifies both fixed-depth and joint relocation, this is JED relocation.

1.3.8 Option to find volume of a group of hypocenters

If the user so chooses, **TexFlex** will calculate the volume of the smallest convex polyhedron that encloses a group of relocated hypocenters. For this purpose **TexFlex** uses the algorithm described by Frohlich [1992]. This volume may be measure of location quality in situations where all members of the group are thought to originate from nearly the same focus. Or, knowledge of the volume can be useful for using seismic moment to estimate strain release rates using Kostrov's equation [see Kostrov, 1974; or Frohlich and Apperson, 1992].

1.3.9 Modular subroutine structure

The program is designed so that a moderately knowledgeable user can modify **TexFlex** and add subroutines which extend the capabilities of the present package. For example, to add a new option for calculating travel times the user could add a subroutine as well as a few calling lines in the present subroutine **TravT**. Without undue effort it should be possible for a user to change the weighting scheme by modifying function **WtPha**, or to implement a different JHD relocation method with subroutines called from **reloc**.

1.4 Discussion

1.4.1 Some practical notes about seismic event location

A good program for locating seismic events helps when you are trying to find accurate locations using phase arrivals for seismic events of unknown location. However, often the locations found will be bad even though the program is good. This section gives some practical advice to help the user recognize certain pathological situations where reliable locations are difficult or impossible to obtain.

There has been no attempt to write **TexFlex** so that it specifically identifies problems, or so that it is very robust with respect to any of the problems described below. In practice, the only clue may be that the program will find unusually large error ellipsoids, or even bomb because certain matrices become very singular. Thus, the program relies on the user to be aware when the input data may be inadequate for the relocation task at hand, and to adjust the data or the program defaults accordingly.

1.4.1.1 Depth/origin time tradeoff for teleseismic relocations

In the location of shallow seismic events using teleseismic P phases there is a tradeoff between focal depth and origin time. This occurs because at teleseismic distances P leaves the source region nearly vertically downward. Thus for any location that fits the observed arrival times, one can obtain a different location that fits equally well simply by changing both the depth and the origin time. Thus, if the crustal velocity is 6 km/sec, the residuals are the almost exactly same if one adds 30 km to the depth and makes the origin time 5 seconds later.

For this reason one must always structure the relocation problem so that there is information that constrains either the focal depth or the origin time. The easiest approach is to arbitrarily fix the focal depth and relocate only the epicenter. Alternatively, one can use S phases or other phases to effectively fix the origin time. Or, one can fix the depth using other phases such as pP, etc.

1.4.1.2 Station correction/origin time constraint for JHD

For N seismic events and M station correction variables there are 4N+M unknowns in the JHD relocation problem, so it would appear that the problem is at least mathematically solvable if there are significantly more than 4N+M observations. However, this is untrue, as the problem requires an additional constraint on the station corrections or the origin times. For example, if one has any set of locations for the N events and any set of M station corrections, the travel time residuals will be the same if one adds 60 seconds to all M station corrections and subtracts 60 seconds from all N origin times.

There are various ways to constrain this problem. For example, Frohlich [1979] requires the sum of the station corrections to be a constant. A second common strategy is the "master event" approach, where one fixes the location of one of the events.

1.4.1.3 Events occurring outside a regional network

There are a host of difficulties obtaining accurate locations for events occurring at some distance beyond of the perimeter of a local or regional station network. For such events there is virtually no constraint on focal depth. Moreover, differences in P arrival times constrain the direction with respect to the network which the event occurred, whereas the difference between S and P times constrains the origin time and the distance from the network. Local network P times alone are seldom sufficient to determine an accurate location. A surprising number of seismologists seem unaware of these assertions even though an understanding of them is essential for understanding the limitations of locations determined from observations at local networks.

It is straightforward to recast the least-squares problem to explicitly distinguish the relative contributions of P arrivals and (S-P) time differences. This can be especially instructive for evaluating location errors determined from local network observations. However, the author has not implemented such an option in the current version of **TexFlex**.

1.4.1.4 Convergence to more than one location

Because travel times are not strictly linear with respect to changes in focal depth, etc., occasionally the iterative least squares calculation will converge to a location corresponding to a local minimum in the sum of squares of residuals, rather than the optimum. In fact, this situation occurs quite commonly for shallow earthquakes recorded by a local network. Often one of the locations will possess a focal depth placing the event above the surface of the earth.

There are several approaches for handling this problem. Options in **TexFlex** allow the user to prevent focal depths from being above the surface, or, to specify that the location can move at most a specified distance on each iteration. Alternatively, sometimes it makes more sense simply to fix the depth at some reasonable value.

1.4.2 About statistics

Among seismologists there is an absence of agreement and much confusion about the appropriate statistical methods for evaluating the quality of seismic event locations. A statistic is just a number, and this number is only useful for evaluating a location if it is a reliable indicator of the accuracy of the location or the stability of the location process. The most common statistics used to evaluate locations are:

- the root mean square (RMS) of the travel time residuals
- the azimuthal gap, or the largest gap in the azimuths of directions of phases used to located the seismic event
- the error ellipsoid, characterized by the lengths and orientations of its three axes, determined from the eigenvectors and eigenvalues of the matrix used to determine the optimum least squares location.

For most purposes a location is usually good if all of these statistics are favorable. Nevertheless, in pathological cases any one of these statistics may give misleading information about the quality of the location. For example, a small RMS residual indicates that the location fits the data. But this may occur either because the location is good or because the data places so few constraints on the location that it can move great distances till the fit is good. The data for a location with a small azimuthal gap should provide good constraints on the location and prevent it from moving freely, however, this may be untrue if some of the "key" phase arrivals are of poor quality. Also, neither the RMS nor the azimuthal gap gives much information about whether the focal depth is accurate.

The error ellipsoid represents an attempt to combine information about the model/data fit with three dimensional information about the directional constraint provided by the weighted phase data. The relative shape and orientation of the ellipsoid depends only on the directions and assigned weights of arriving phases, while the absolute size of the ellipsoid is proportional to the RMS fit. Different authors disagree about the appropriate constant of proportionality [see Boyd and Snoke, 1984], so for simplicity **TexFlex** uses 1.0, allowing the user to apply any additional factor depending on sum of phases weights, etc., if desired.

However, to be safe the wary user should take care not to place undue trust in error ellipsoids, especially if the RMS residual is unreasonably small. Seismic data nearly always violates the statistical assumptions underlying the interpretation of error ellipsoids in terms of confidence limits. Moreover, the actual errors or uncertainties in event locations are nearly always larger than those suggested by these statistical models. The user should remember that in the largest sense the quality of any particular location depends on all the data used to determine the location, and thus any particular statistic can only give a partial indication of the quality.

One direct approach for evaluating location uncertainties is to implement so-called "jack-knife" or "bootstrap" methods [e. g., see Tichelaar and Ruff, 1989]. In this approach the user systematically leaves out data from individual stations or stations groups and statistically evaluates how seriously the missing data affects the final location. This quite directly shows how individual data influence the result, and in principal gives an indication of how additional data might affect the location, if such additional data were available. At present we have not specifically designed **TexFlex** to implement a jackknife or bootstrap analysis. However, since the essence of the method is to perform locations while assigning certain input data zero weight, it would not be difficult to change the program to do this.

1.4.4 A word about bugs and program support

Nearly every computer program has bugs. **TexFlex** is a computer program. So **TexFlex** probably has bugs, too, especially since the author of **TexFlex** is not a professional programmer. Presently there are no formal plans to provide software support for **TexFlex**, however, the author welcomes users to send email which makes suggestions and provides details about problems.

References

Boyd, T. M. and J. A. Snoke. Error estimates in some commonly used location programs, *Earthquake Notes, 55*, 3-6, 1984.

Buland, R. and C. H. Chapman. The computation of seismic travel times, *Bull. Seism. Soc. Amer., 73*, 1271-1302, 1983.

Bullen, K. E. *An Introduction to the Theory of Seismology*, Cambridge University Press, 381 p., Cambridge, 1965.

Dewey, J. W. Seismicity and tectonics of western Venezuela, *Bull. Seismol. Soc. Amer., 62*, 1711-1751, 1972.

Doornbos, D. J.. Asphericity and ellipticity corrections, *Seismological Algoithms*, 75-85., ed. D. J. Doornbos, Academic Press, London.

Douglas, A. Joint epicenter determination, *Nature 215*, 47-48, 1967.

Dziewonski, A. M. and F. Gilbert. The effect of small, aspherical perturbations on travel times and a re-examination of the correction for ellipticity, *Geophys. J. R. Astr. Soc., 44*, 7-17, 1976.

Flinn, E. A. Confidence regions and error determination for seismic event location, *Rev. Geophys., 3*, 157-185, 1965.

Frohlich, C. An efficient method for joint hypocenter determination for large groups of earthquakes, *Comput. Geosci., 5*, 387-389, 1979.

Frohlich, C. *An Integrated Approach to Seismic Event Location: I. Evaluating How Method of Location Affects the Volume of Groups of Hypocenters*, Phillips Laboratory, Hanscom Air Force Base, Technical Report PL-TR-92-2323, 30 p., 1992.

Frohlich, C. and K. D. Apperson. Earthquake focal mechanisms, moment tensors, and the consistency of seismic activity near plate boundaries. *Tectonics, 11*, 279-296, 1992.

Frohlich, C., S. Billington, E. R. Engdahl and A. Malahoff. Detection and location of earthquakes in the central Aleutian subduction zone using island and ocean bottom seismograph station. *J. Geophys. Res., 87*, 6853-6864, 1982.

Herrmann, R. B., S-K Park and C-Y Wang. The Denver earthquakes of 1967-1968, *Bull. Seismol. Soc. Amer., 71*, 731-745, 1981.

Jeffreys, H. and K. E. Bullen. *Seismological Tables.* Grey Milne Trust, London, 50 p. 1970.

Jordan, T. H. and K. A. Sverdrup. Teleseismic location techniques and their application to earthquake clusters in the south-central Pacific, *Bull. Seismol. Soc. Amer., 71*, 1105-1130, 1981.

Kennett, B. L. N. (editor). *IASPEI 1991 Seismological Tables*, Australian National University, 167 p., 1991.

Kostrov, V. V. Seismic moment and energy of earthquakes, and seismic flow of rock. *Izv. Acad. Sci. USSR Phys. Solid Earth,* Engl. Transl. *1*, 23-44, 1974.

Lerner-Lam, A. L. and T. H. Jordan. How thick are the continents? *J. Geophys. Res.*, 92, 14007-14026, 1987.

McLaren, J. P. and C. Frohlich. Model calculations of regional network locations for earthquakes in subduction zones, *Bull. Seismol. Soc. Amer., 75*, 397-413, 1985.

Pavlis, G. L. Appraising relative earthquake location errors, *Bull. Seismol. Soc. Amer., 82*, 836-859, 1992.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes - The Art of Scientific Computing (FORTRAN Version)*, Cambridge Univ. Press, Cambridge, 702 p., 1989.

Pujol, J. Comments on the joint determination of hypocenters and station corrections, *Bull. Seismol. Soc. Amer., 78*, 1179-1189, 1988.

Purdy, G. M. and J. Ewing. Seismic structure of the ocean crust, in, *The Geology of North America, Volume M, The Western North Atlantic Region*, ed. by P. R. Vogt and B. E. Tucholke, Geological Society of America, 313-330, 1986.

Tichelaar, B. W. and L. J. Ruff. How good are our best models? Jackknifing, bootstrapping, and earthquake depth, *Eos, 70*, 593-596, 1989.

Willemann, R. J. and C. Frohlich. Spatial patterns of aftershocks of deep earthquakes, *J. Geophys. Res., 92*, 13927-13943, 1987.

SECTION 2: How To Use **TexFlex**

2.1 How to get started - the simplest possible trial run

Section 2.1 assumes that you are a potential user of **TexFlex** who has obtained copies of the program files, sample input data files, and other useful files. This section assumes that you wish to run the program to make sure it works properly on your workstation, and to help decide whether it would be useful for your research purposes.

If the compiled version of **TexFlex** which is a file named 'jhd' is in your computer directory, and IF various necessary input files supplied in the program package are in that directory, then you can run **TexFlex** simply by typing:

```
jhd
```

and the program will run and produce various output files. Thus one way to learn about the program is to try this and then inspect various default output files, especially the summary file 'example.sumO' and the event-by-event result file 'example.hypS'. Other default names for output files are: 'example.hypO', example.phaO', and example.corO'.

If this works, try changing the input line in file 'driver' to read 'example.2.DVR', and then run the program again. The file *dvrfile* = 'example.2.DVR' uses almost as many default values as the file 'example.DVR' except it specifies a flat-earth velocity model, and relocates events recorded by a regional OBS-land network, rather than teleseismically observed events. Finally, to learn how to change several different default values, change 'driver' to read 'example.3.DVR' and rerun the program a third time.

However, if none of this works, or if you would like to know more about TexFlex before you try to run the program, read some or all of the following section-- section 2, then, come back and try the program runs suggested in this section again.

2.2.1 Are the necessary files available?

2.2.1.1 File of notes describing **TexFlex**

The file named 'TexFlex.NOTES' briefly describes the various files in the **TexFlex** package. This information is mostly repeated in section 2.2.1.

2.2.1.2 Files containing the Fortran program **TexFlex**

Presently there are five files containing Fortran programs that make up the **TexFlex** package. These are:

- 'TexFlex.Main.0.5.f' - This includes the main program and a subroutine **SetDef** which sets default parameters and tells the main program the names of various input files. Section 5.1 describes the subroutines in this file.

- 'TexFlex.InOut.0.5.f' - This includes subroutines for reading in data and for displaying and formatting output. Sections 5.2 and 5.5 describe the subroutines in this file.

- 'TexFlex.RelocMatV.0.5.f' - This includes subroutines for relocating hypocenters, and for performing certain routine vector and matrix operations. Sections 5.3 and 5.6 describe the subroutines in this file.

- 'TexFlex.TravT.0.5.f' - This includes subroutines for calculating travel times. Currently, the only options are JB-P times, IASPEI91 travel times for a variety of phases, or flat-earth travel times for a user-supplied or a default earth model.

- 'TexFlex.VolFind.0.5.f' - This includes subroutines for calculating the volume of the smallest convex polyhedron surrounding a set of points, here a group of seismic events. Section 5.7 describes the subroutines in this file.

2.2.1.3 File for compiling/linking **TexFlex** on a unix-based Sun workstation

On a unix-based Sun workstation, one can compile and link **TexFlex** using the file 'TexFlex.make.0.5' and the following "make" command:

```
make -f TexFlex.make.0.5
```

For convenience and completeness I reproduce this "make file" in its entireity:

file 'TexFlex.make.0.5':

```
FFLAGS = -sun4 g

jhd: TexFlex.Main.0.5.o TexFlex.InOut.0.5.o TexFlex.TravT.0.5.o \
       TexFlex.RelocMatV.0.5.o TexFlex.VolFind.0.5.o
       f77 $(FFLAGS) -o jhd TexFlex.Main.0.5.o TexFlex.InOut.0.5.o \
       TexFlex.TravT.0.5.o TexFlex.RelocMatV.0.5.o TexFlex.VolFind.0.5.o
TexFlex.Main.0.5.o: TexFlex.Main.0.5.f
       f77 $ (FFLAGS) -c TexFlex.Main.0.5.f
TexFlex.InOut.0.5.o: TexFlex.InOut.0.5.f
TexFlex.TravT.0.5.o: TexFlex.TravT.0.5.f
TexFlex.RelocMatV.0.5.o: TexFlex.RelocMatV.0.5.f
TexFlex.VolFind.0.5.o: TexFlex.VolFind.0.5.f
```

2.2.1.4 Necessary input files which the **TexFlex** user doesn't change

The **TexFlex** program package expects to find several input files which are tables for determining travel times, etc. These are:

- File 'jbp.tbl' - A file containing Jeffreys-Bullen [1970] P and pP-P travel times as well as tables of ellipticity corrections.

- Files 'iasp91.tbl' and iasp91.hed' - These are lookup tables for determining IASPEI91 travel times [Kennett, 1991]. If these tables are unavailable, they can be generated by other software obtainable from IRIS and described briefly in comments near the subroutine **TravTIASP91** in file 'TexFlex.TravT.0.5.f'.

- File 'ttlim.inc' - This is a file of parameters called using and include statement in several of the IASPEI91 travel time subroutines.

2.2.1.5 Input files which the **TexFlex** user can change

The author's experience is that during a research project involving the location of seismic events, usually different combinations of the same events are located over and over again, with each iteration involving different assumptions about event grouping, phase weighting, choice of station corrections etc. The design structure of the input files described in this section should make this process as simple and efficient as possible. In this section, specific file names are in quotations (' '), while file names in italics specify character-string variables which must be specified by the users.

- File 'driver' - This file contains only a single line. This line is the name of the driver file *dvrfile* which specifies initialization parameters and designates the seismic events to be relocated.

- File *dvrfile* - If the user chooses to use default values for all file names and relocation parameters, this file contains only the identification numbers of the events to be relocated (e. g., see the default file *dvrfile* = 'example.DVR'). The program then uses these event identification numbers to find trial hypocenters and phase arrival information in files *hypfile* = 'example.HYP' and *phafile* = 'example.PHA'. However, more commonly *dvrfile* will contain information designating other file names for the input data, as well as resetting various defaults affecting phase weighting, relocation method, choice of travel time models, etc. The **TexFlex** package also includes the file 'example.3.DVR', which is a more typical driver file, and which contains extensive comments describing the options for various default parameters.

- File *stafile* - This is an archive file containing information about seismic stations, namely, their character names, latitude, longitude, and elevation about sea level. The default for *stafile* is 'world.STN' (see section 3.3). This is a file supplied with **TexFlex** which contains station information for about 2000 stations. For specific regional applications the program will run slightly faster if the user instead substitutes a shorter file containing information about only the stations of interest for a particular project.

- File *hypfile* - This is the file containing trial hypocenters, ordered from smallest event identification number to largest. Each line possesses an event identification number, a character name for the event, an origin time, a latitude, a longitude, and a depth. The file must contain such a line for each seismic event to be relocated on a particular program run, however, *hypfile* may also contain hypocenter information for other events as well. The default is *hypfile* = 'example.HYP' (see section 3.4).

- File *phafile* - This is the file containing phase arrival information for all seismic events, ordered from smallest event identification number to largest. Each phase arrival line contains an arrival time, a station name, a phase name, and (if desired) a user-specified quality integer between 0 and 9. The file must contain such a phase information for each seismic event to be relocated on a particular program run, however, it may also contain phase information for other events as well. The default is *phafile* = 'example.PHA' (see section 3.5).

- File *corfile* - Essentially, this is a file which specifies how station correction information will be processed during the relocation. However, the file is important even if the user plans to perform single-event relocations and set all station corrections to zero. For each "station correction variable" in the file, the user can designate up to 9 station/phase combinations, all of which will utilize the same station correction. For single-event relocation, **TexFlex** only considers information for the designated phases at the designated stations, and ignores all other station/phase combinations.

For JED or JHD relocation, this file allows the user to group station correction variables in a convenient way. For example, the user can choose to have several different stations all receive the same station correction. Or alternatively, the user could have different phases at the same station, such as P, S, and PKP, each receive a different station correction. Finally, the user could have P and S associated with the same station correction variable, but with the S correction always being a user-designated amount (e. g., 1.71) larger than the P correction. The default is *corfile* = 'example.COR' (see section 3.6).

2.3 Description of the Input for **TexFlex** version 0.5.

2.3.1 Absolute necessities

The compiled version of **TexFlex** expects to find several files, and will not run without them. These are:

- 'iasp91.tbl' - a table necessary for finding IASPEI91 travel times
- 'iasp91.hed' - a table necessary for finding IASPEI91 travel times
- 'ttlim.inc' - a table necessary for finding IASPEI91 travel times
- 'jbp.tbl' - a table necessary for finding JB P times

For descriptive information about these files see section 2.2.1.4.

TexFlex also expects to find a file named 'driver'. This file contains a single line, and on this line is the name of the file *dvrfile* described below.

2.3.2 The driver file *dvrfile*

If desired, this file allows the user to make changes in program input and output file name, other program defaults. In addition, at the end of *dvrfile* the user lists the identification numbers of all events to be relocated.

- Please note that the program ignores lines beginning with the character 'c', treating them as comments. In the file 'example.3.DVR', there are about 150 of these comment lines defining many of the available options described below. Many users may wish to include these comment lines in all versions of *dvrfile* as a guide for making changes in default values.

- All lines in *dvrfile* beginning with the characters '`Comment: `' define comments which will be printed at the beginning of the program summary output file *statfil.*

- All lines in *dvrfile* which define or change program defaults begin with the 9 characters '`default:`', followed by a blank character, followed by a six-character default input code *defin* designating the default to be changed, and finally by additional integers, real numbers, or characters which are new default value.

- After the last '`default:`' line, the program expects a blank line. Then the user should designate a list of identification numbers of seismic events targeted for relocation, with one event identification number per line, in format (i5). The event identification numbers do not have to be in sequence but they must be in ascending order.

Following the last event identification number there is a line containing blanks ('   ').

- In the examples that follow, the program-set default value will be followed by '[**DF**]'. That is, the program sets any default to the value so specified if the user does not change it in file *dvrfile.*

2.3.2.1 Defaults concerning file names

An example of the structure of a default line describing a file name is:

```
        default: fiPhaI /usr/home/example.PHA
```

Here, the default input code *defin* is 'fiPhaI', indicating that the following file name describes for the input phase data file *phafile*, giving it value /usr/home/example.PHA'. For phafile, the program-supplied default value is:

```
        default: fiPhaI example.PHA      [**DF**]
```

Other options of *defin* describing file names are:

- *defin* = 'fiHypI' - Following file name is for trial hypocenter input file *hypfile.* The default is:
  ```
  default: fiHypI example.HYP      [**DF**]
  ```

- *defin* = 'fiCorI' - Following file name is for trial station correction variable input file *corfile.* The default is:
  ```
  default: fiCorI example.COR      [**DF**]
  ```

- *defin* = 'fiStns' - Following file name is for station coordinate input file *stafile.* The default is:
  ```
  default: fiStns world.STN        [**DF**]
  ```

- *defin* = 'fiPhaO' - Following file name is for phase output file *phafOut* The default is:
  ```
  default: fiPhaO example.phaO     [**DF**]
  ```

- *defin* = 'fiHypO' - Following file name is for one-line-per-hypocenter output file *hypfOut.* The default is:
  ```
  default: fiHypO example.hypO     [**DF**]
  ```

- *defin* = 'fiHypS' - Following file name is for more detailed hypocenter output file *hypfSum.* The default is:

```
default: fiHypS example.hypS
```
[**DF**]

- *defin* = 'fiCorO' - Following file name is for station correction variable output file *corfOut*. The default is:
  ```
  default: fiCorO example.corO
  ```
  [**DF**]

- *defin* = 'fiStat' - Following file name is for summary information file *statfil*. The default is:
  ```
  default: fiStat example.sumO
  ```
  [**DF**]

- *defin* = 'fiVMfl' - Following file name gives information about the velocity structure to be used for calculating flat-earth travel times. The default is:
  ```
  default: fiVMfl example.flvm
  ```
  [**DF**]

2.3.2.2 Defaults concerning integer variables

All default changes involving integers must appear in format (i4) in columns 17-20, preceded by the characters '`default:`' in columns 1-9, and the six character default input code *defin* in columns 11-16. As above, the program-set default values are denoted by '[**DF**]'.

Free or fixed focal depths: For example, to change the default so that the program relocates events with fixed focal depths the line reads:

```
default: DepthF   1
```

For variable focal depth relocations the line reads:

```
default: DepthF   0
```
[**DF**]

Other integer defaults:

- Number of iterations desired:
  ```
  default: NumbIt  10
  ```
  [**DF**]

- Single-event or JHD relocation:
  ```
  default: RelocX   0
  ```
  [**DF**] – single-event relocation
  ```
  default: RelocX   1
  ```
  – Frohlich's [1979] "fast JHD" relocation

- Negative focal depths allowed:
  ```
  default: DepFly   0
  ```
  – negative focal depths allowed
  ```
  default: DepFly   1
  ```
  [**DF**] - focal depths

constrained to be positive. If the focal depth is H and then an iteration determines a negative focal depth, the program instead chooses the next trial focal depth as 0.7*H

• Method of travel time calculation:

`default: TravTm     1` – JB P times

`default: TravTm     2` [**DF**] – IASPEI91 travel times [see Kennett, 1991]

`default: TravTm     3` – Flat-earth travel times.

`default: TravTm     4` – User-specified travel times. If this option is in effect the program uses the model specified for each station-phase combination in corfile, and uses IASPEI91 travel times when no model is so specified.

For all flat-earth travel times, a default flat-earth model is used unless the model is changed by:

`default: NewFVM     1` – In this case flat-earth times are calculated from one model supplied in a file named following default code *defin* = 'fiVMfl' as described above

`default: NewFVM     2` – In this case flat-earth times are calculated for two models supplied in a file named following default code *defin* = 'fiVMfl' as described above

• Method of distance/azimuth calculation:

`default: DeltAz     1` – spherical earth assumed

`default: DeltAz     2` [**DF**] – geocentric coordinates used with method suggested by Bullen [19xx] page 155.

• Ellipticity corrections:

`default: EllipC     0` – no ellipticity correction

`default: EllipC     1` [**DF**] – Jeffreys-Bullen ellipticity correction applied

• Finds volume of grouped hypocenters:

`default: VolFnd     0` [**DF**] – volume not found

`default: VolFnd     1` – volume found by subroutine **VolumeFind** prior to first relocation and after last relocation

- Method of weighting travel time arrivals:
  `default: WtType    0` [**DF**] – Uniform weights, without residual-dependent ramps or Gaussean weight functions
  `default: WtType    1` – Trapezoidal weight function applied, with flat part having half-width set by default code 'WtWidt' and the ramp having width set by default code 'WtRamp' (see below)
  `default: WtType    2` – Gaussean weight function applied, with half-width set by default code 'WtWidt' (see 2.3.2.3 below)

- Weight factor includes user-supplied quality dependence:
  `default: WtUser    0` [**DF**] – Uniform weights, independent of phase quality
  `default: WtUser    1` – Weight function multiplied by user-supplied factor depending on phase quality factors set by WtUkth (see 2.3.2.4 below)

- Weight factor iteration dependent
  `default: WtItDp    0` [**DF**] – Weights are iteration independent
  `default: WtItDp    1` – After half of iterations are complete, the ramp or Gaussean widths specified by default codes 'WtRamp' or 'WtWidt' (see 2.3.2.3 below) are reduced by a factor of two

- Weight factor azimuthally dependent
  `default: WtAzSc    0` [**DF**] – phase weights do not depend on azimuth of recording station
  `default: WtAzSc    1` – phase weights adjusted so that sum of weights is the same for arrivals from each azimuthal sector
  `default: WtNazS    4` [**DF**] – if the default code 'WtAzSc' is set to 1, this is the number of azimuthal sectors desired
  `default: WtBazS  15.` – if default code 'WtAzSc' is set to 1, this is the azimuth of the boundary between the first two azimuthal sectors. This allows the user to adjust the sector boundaries so that they possess a convenient geometry. The default value is 0.0 degrees.

- Printing option:
  `default: NoPrin    0` – Some output information suppressed, e. g., file *phafOut* not printed

  `default: NoPrin    1` [**DF**] – All output information such as *phafOut* produced

- Number of columns in file *statfil*:
  `default: NumCol 119` [**DF**] – width in characters of event-station residual table in file *statfil*. Changing this may make the table more readable if one actually prints it out on paper

2.3.2.3 Defaults concerning real variables

Real variable information appears in columns 17-24, preceded by the characters '`default:`', followed by a blank, followed by a six-character default input code *defin*.

- Phase Weighting - half-width in seconds of Gaussean function, or width of flat part of trapezoidal weights:
  `default: WtWidt     4.0` [**DF**]

- Phase Weighting - width in seconds of ramp for trapezoidal weights
  `default: WtRamp     4.0` [**DF**]

- Distance moved/iteration - maximum distance that a hypocenter can move during one individual iteration
  `default: DMvMax    30.0` [**DF**]

2.3.2.4 Other defaults

- Four-character identification code - format (a4) following six-character default input code 'RunNam':
  `default: RunNam Xx3` – here, ' Xx3' will appear on each line of hypfOut, identifying locations from this program run. The default is the blank characters '   '.

- Choice of phases for IASPEI91 travel times - format (a8) following six-character default input code 'IASPph':
  `default: IASPph ALL` [**DF**] – gives all available IASPEI phases

`default: IASPph P` - gives P-up, P, P_{diff}, PKP, and PKiKP
`default: IASPph P+` - gives P-up, P, P_{diff}, PKP, and PKiKP, PcP, pP, pP_{diff}, pPKP, pPKiKP, sP, sP_{diff}, sPKP, and sPKiKP
`default: IASPph S` - gives S-up, S, S_{diff}, and SKS
`default: IASPph S+` - gives S-up, S, S_{diff}, SKS, sS, sS_{diff}, sSKS, pS, pS_{diff}, and pSKS
`default: IASPph basic` - gives P+ and S+ as well as ScP, SKP, PKKP, SKKP, PP, and PKPPKP

• User-defined, quality-dependent weights - User assigns integer quality factors and desires these to correspond to real weights. The assigned quality/weight correspondence are in format (i4,f8.3) following '`default:  WtUkth`' . For example, to assign weights to qualities 0 through 4:

```
default: WtUkth   0  1.00 [**DF**]
default: WtUkth   1  1.00 [**DF**]
default: WtUkth   2  0.75 [**DF**]
default: WtUkth   3  0.50 [**DF**]
default: WtUkth   4  0.00 [**DF**]
```

Note, however, that these default weights are not applied unless the user explicitly implements them by setting the default input code 'WtUser' to 1.

2.3.3 The trial hypocenter file *hypfile*

In this file the user sets trial hypocenters for the seismic events to be relocated. Each trial hypocenter occupies a single line, and this line must specify an event identification number, a character name, origin time in hours, minutes, and seconds, latitude (degrees), longitude (degrees), and focal depth (km), in format (i5,a12,1x,i2,1x,i2,1x,f5.2,f8.3,f9.3,f7.3). The user can assign event identification numbers arbitrarily. However, in *hypfile, dvrfile*, and *phafile* the trial hypocenters must appear in order of increasing event identification number.

After the last trial hypocenter *hypfile* should have line beginning with an 'e'. Also, the program ignores all lines beginning with a 'c', allowing the user to add comments to *hypfile.*

2.3.4 The phase file *phafile*

In this file the user supplies basic information about phase arrival times for each of the seismic events. For each event the first line begins with an asterisk ('*') and the event identification number in format ('*',5x,i4). Then, each subsequent line is the information about one phase arrival time-- hypocenter name, phase arrival time in hours, minutes, and seconds, station name, phase name, and (if desired) phase quality assignment in format (a6,1x,i2,i2,f5.2,1x,a4,1x,a8,i3). The phase information must be ordered so that event identification numbers increase throughout *phafile*, and they must correspond to the proper trial hypocenter in file *hypfile*.

After the phase information for the last event the last line in *phafile* must begin with an 'e'. Also, the program ignores all lines beginning with a 'c', allowing the user to add comments to *phafile*.

2.3.5 The station correction variable file *corfile*

In this file the user sets up program structure determining how station corrections will be applied and/or calculated. This file allows the user to assign the same station correction or (for JED or JHE) the same station correction variable to several different stations and/or to several different phases at the same station. It also permits the user to multiply station correction variables for selected station-phase combinations by a constant. This allows the user to assign the same correction variable to P and S, for example, but the S correction is 1.71 larger than the P correction. Information in this file also permits the user to use different travel time models for calculating travel times for different station-phase combinations, if desired.

In *corfile*, each group of station-phase combinations to be assigned to the same station correction variable begins with a line having the characters 'sc' in columns 1 and 2 and the initial station correction in columns 4-11. Then, the subsequent lines contain the station name, phase name, multiplier, and an integer identifying the travel time model, in format (a4,1x,a8,f6.2,i2). Typically, blanks for station and phase names are at the end of the character field, and the multiplier is 1.00 for most relocation applications. The travel time model identifier is: 1 for JB P times; 2 for IASPEI91 times, and either 3 or 4 for flat-earth travel times.

After the phase information for the last event the last line in *corfile* must begin with an 'e'. Also, the program ignores all lines beginning with a 'c', allowing the user to add comments to *corfile*.

2.3.6 The station location file *stafile*

This file contains station names, latitudes, longitudes, and elevations in format (a4,f8.3,f9.3,f6.0) for all stations. After the phase information for the last event the last line in *corfile* must begin with an 'e'. Also, the program ignores all lines beginning with a 'c', allowing the user to add comments to *stafile*.

File 'world.STN' is supplied with **TexFlex**. This file contains station information for about 2000 commonly reporting seismic stations, as well as for some OBS stations used in the examples (see section 2.1).

2.3.7 The flat-earth velocity model file *flvmfil*

If the user specifies 1 for the default input code 'NewFVM' (see 2.3.2.2 above), the program opens file *flvmfil* to allow the user to specify two flat-earth velocity models for calculating travel times designated as methods 3 and 4 (see default identifier code 'TravTM' in 2.3.2.2 above). In *flvmfil*, the program utilizes free-format read statements to find values for the number of layers, the velocity in each layer, and thickness of each layer. Then, this information is repeated for the second flat-earth travel time model. Thus, one acceptable format would place the number layers for the first model in format (i3), and the velocities and thicknesses each in format (6f10.3).

The two models used as defaults for flat-earth travel times are:

A "continental" model based on the review of Lerner-Lam and Jordan [1987]:

layer	velocity (km/sec)	thickness (km)	V_P/V_S
1	2.98	3.0	1.77
2	6.08	17.0	
3	6.78	16.0	
4	8.19	-	

An oceanic" model based on the review of Purdy and Ewing [1986]:

layer	velocity (km/sec)	thickness (km)	V_P/V_S
1	5.50	7.0	1.77
2	7.00	5.0	
3	8.00	-	

SECTION 3: Sample Input Data for **TexFlex** version 0.5

3.1 The file 'driver'

This file always consists of a single line, which specifies the name of the file *dvrfile*. For example, if you are just beginning with **TexFlex** you might have file 'driver' contain the following:

```
example.1.DVR
```

3.2 The driver file *dvrfile*

The first file that follows is 'example.1.DVR', and it is just about the simplest possible file for running **TexFlex**. Here, all the program options are left as defaults, and *dvrfile* consists simply of a blank line, a list of event identification numbers specifying events to be relocated, and a line containing five blanks.

```
   42
  151
  360
  422
  502
  690
  696
  706

c Note: last non-comment line
c   must contain blanks as
c   first five characters
```

The next file that follows is file 'example.DVR'. Here, only three default options have been changed. In particular, the file specifies a different file for *corfile*, which defines the station correction variable architecture. It also specifies that the travel times should be calculated using the models specified in *corfile* (`TravTm = 40),` and that relocations should be determined with depths fixed at the trial values (`DepthF = 1`).

```
c *** Trial station corrections
default: fiCorI /usr2/cliff/JHD/example.2.COR
c*********************************************
default: TravTm    4
default: DepthF    1

 1011
 1012
 1013

c*****************************************
```

Finally, the following file is 'example.3.DVR' in which there are extensive comments which describe all of the options which can be changed in **TexFlex**. This particular run specifies that travel times be calculated only for JB P times (`TravTm = 1`), and that the program determine the convex polyhedron bounding the trial and relocated hypocenters (`VolFnd = 1`). It also renames most of the output files and routes them to directories other than the default directory.

```
c Driver file for relocation program TexFlex
c********************************************
c ** 14 June 1992 **
c NOTE: in this driver file this and all lines beginning
c  with a 'c' are ignored
c*************************************************
c The driver file begins by resetting some
c   defaults, if this is desired
c**************************************************
c Defaults: all lines beginning with 'default: ' and a six
c  letter default code are for changing defaults.  The
c  defaults changes can be in any order, and need not
c  be changed from the values set in the program.
c Numerical defaults which are integers must be in
c  a format i4 (ending in column 20)
c  following the six letter code and one blank.
c There must be a blank line following the last default
c  which is changed
c***************************************************
c Some default parameters are integer flags that
c  make choices for the calculation:
c***************************************************
c *** Distance and azimuth calculations
c DeltAz = 1 - Round earth used for calculating distance
c               and azimuth for station-quake pairs
c        = 2 - Geocentric coordinates used with method
c               suggested by Bullen, page 155
c***************************************************
c *** Fixed or free depth used during relocation
c DepthF = 0 - Depth allowed to vary during relocation
c        = 1 - Depth fixed during relocation
c***************************************************
c *** Ellipticity corrections used in travel time
c EllipC = 0 - no ellipticity correction made
c          1 - JB ellipticity correction included
c***************************************************
c *** Number of iterations
c NumbIt = number of iterations during relocation
c***************************************************
c *** Method of relocating hypocenters
c RelocX = 0 - Single event relocation
c          1 - JHD relocation, using method of Frohlich
c               from Computers & Geosciences 1979
c***************************************************
c *** Maximum distance moved each iteration ***
c DMvMax = maximum distance hypocenter can
c           move on each iteration
c***************************************************
```

```
c *** Negative focal depths allowed? ***
c DepFly = 0 - negative focal depths allowed
c        = 1 - focal depths constrained to be positive
c*************************************************
c *** Find Volume of Hypocentral group? ***
c VolFnd = 0 - don't find volume
c        = 1 - find volume
c*************************************************
c *** Travel time calculation
c TravTm  = 1 - JB P tables used for travel times
c         = 2 - IASPEI 91 program used for travel times
c         = 3 - Flat-earth travel times
c NewFVM  = 0 -  default flat-earth model used
c         = 1 -  flat-earth model from user-supplied file
c*************************************************
c *** Number of columns
c NumCol = number of columns in a page of output
c*************************************************
c RunNam = 4-character identification for this Run
c*************************************************
c *** Print Flag
c NoPrin = 0 - phase information not printed
c        = 1 - phase information is reprinted
c              in a file along with various statistics
c              such as delta, azimuth, etc.
c*************************************************
c *** Weighting of Travel Times
c WtType = 0 - No ramps, Gaussean weights
c        = 1 - Trapezoid weight function, with the flat
c               part having half-width WtWid and the ramp
c               having half-width WtRamp
c        = 2 - Gaussean weight function, with width
c WtWidt = if WtType = 1 or 2, this is the half-width of
c          the flat part of the trapezoid, or the
c          Gaussean
c WtRamp = if WtType = 1, this is the width of the
c          ramp
c*************************************************
c WtUser = 1 - Weight function multiplied by user-supplied
c               factor depending on phase quality
c        = 0 - No quality factor
c WtUkth - if WtUser = 1, the user resets the value wtU
c          for assigned quality k
c*************************************************
c WtItDp = 1 - Ramp or Gaussean widths are reduced
c               by a factor of two after half of
c               the iterations are complete
c        = 0 - Weights are iteration independent
c*************************************************
c WtAzSc = 1 - Weights are normalized within each
c               azimuthal sector, so no particular
c               sector dominates calculation
c        = 0 - Weights are independent of Azimuth
c WtNazS - if WtAzSc = 1, WtNazS is the number of
c          of azimuthal sectors chosen
c WtBazS = if WtAzSc = 1, WtBazS is the azimuth
c          of the boundary of the first sector
c*************************************************
```

```
c************************************************
default: TravTm    1
default: DeltAz    2
default: DepthF    0
default: RelocX    0
default: VolFnd    1
default: EllipC    0
default: NumbIt    9
default: NumCol   80
default: NoPrin    1
c************************************************
c Some defaults are file names for input files:
c *** Phase file
default: fiPhaI example.PHA
c *** Trial hypocenters
default: fiHypI example.HYP
c *** Trial station corrections
default: fiCorI example.COR
c *** Locations of seismograph stations
default: fiStns world.STN
c *** Information for flat-earth velocity model
default: fiVMfl example.flvm
c************************************************
c Some defaults are file names for output files
c *** New station corrections
default: fiCorO /utig/usr2/cliff/JHD/example.cor.out
c *** Relocated hypocenters
default: fiHypO /utig/usr2/cliff/JHD/example.hyp.out
c *** Relocated hypocenters - event-by-event list
default: fiHypS /utig/usr2/cliff/JHD/example.hyp.sum
c *** Residuals in same format as Phase file
default: fiPhaO /utig/usr2/cliff/JHD/example.pha.out
c *** Various summary statistics
default: fiStat /utig/usr2/cliff/JHD/example.Stat.out
c************************************************
c Some defaults are names for initializing tables, etc.
c *** Choice of phases for IASPEI 91 tables
c  The phase codes or keywords for the required branches
c    must be specified
c  ALL will give all available branches
c  P  gives P-up,P,Pdiff,PKP, and PKiKP
c  P+ gives P-up,P,Pdiff,PKP,PKiKP,PcP,pP,pPdiff,pPKP,
c           pPKiKP,sP,sPdiff,sPKP, and sPKiKP
c   S  gives S-up,S,Sdiff, and SKS
c   S+ gives S-up,S,Sdiff,SKS,sS,sSdiff,sSKS,pS,pSdiff,
c            and pSKS
c   basic gives P+ and S+ as well as
c            ScP, SKP, PKKP, SKKP, PP, and PKPPKP
c
c   or give a generic phase name
c**
default: IASPph ALL

c************************************************
c next: are an indeterminate number of lines
c with identification numbers of hypocenters to be
c  relocated in format i5.  There must be a blank
c ('     ') after last trial hypocenter. The
```

```
c hypocenter numbers must be an ascending order
c*******************************************
   42
  151
  360
  422
  502
  690
  696
  706

c*******************************************
```

3.3 The station location file *stafile*

This is file 'world.STN'. Note the extensive comments at the beginning of the file.

```
c 29 January 1992: This is a station file
c  copied from the file sent by Doug Wiens,
c  and augmented to include stations
c  used during 1989 ORSTOM-UTIG OBS
c  experiment
c******************************************
c Formats for the station file:
c First column = 'c'
c   This line is a comment line - ignored by the program,
c     except possibly printed out
c First column = 'e'
c   This is last line read by program, no more station data
c First column otherwise:
c   This is data for a particular station
c   example:
c AAA    43.272    76.947   800
c Name     lat       long  ht(m)
c*******************************************

c ORSTOM-UTIG VANUATU 1989 OBS EXPERIMENT
c   3 August 1992 station file
DVP  -17.730   168.190    77
EME  -17.091   168.350   527
EPI  -16.695   168.116   188
LMP  -16.470   167.820   138
MBV  -17.660   168.310   471
NGA  -17.450   168.340   593
PVC  -17.740   168.310    77
CAV  -17.740   168.310    77
RTV  -17.790   168.420    99
SWB  -16.509   167.420   305
OB1  -17.383   167.846 -3605
OB2  -17.405   168.020 -1620
OB3  -17.474   167.889 -3320
.
.
.
O34  -16.740   167.723 -1240
O35  -16.983   167.472 -3130
```

```
c
c Stations From World List
AAA   43.272    76.947   800
AAB   43.267    77.383   850
AAC   50.783     6.083   179
AAE    9.029    38.766  2442
AAI   -3.700   128.167    80
AAM   42.300   -83.656   254
AAS  -62.160   -58.463    15
ABA   36.801     3.035   332
ABC   13.928    20.858   550
ABE   57.167    -2.100    12
.
.
.
.
ZGN   36.375    10.112   720
ZLA   47.482     8.389   780
ZLP  -16.270   -68.118  4398
ZOBO -16.270   -68.125  4450
ZON  -31.546   -68.679   730
ZSC   31.097   121.187   100
ZST   48.196    17.103   250
ZUL   47.481     8.390   740
ZUR   47.369     8.580   604
end
```

3.4 The trial hypocenter file *hypfile*

This is file 'example.HYP'. Note the extensive comments at the beginning of the file.

```
c 14 June 1993:
c*****************************************************
c Formats for epicenter file:
c quake identification number, quake name, origin time,
c    trial latitude, longitude, depth
c  in: (i5,a12,1x,i2,1x,i2,1x,f5.2,f8.3,f9.3,f7.2)
c
c On output, there are also fields for RMS residual,
c   lengths of two most horizontal error ellipsoid axes,
c   angle N of E for the first error ellipsoid axis,
c   sum of weights, and the azimuthal gap
c*****************************************************
c    #      name      origin       lat      long   depth
    42      2/26/65 23:36:12.7   6.770  -73.200  158.0
   151      7/29/67 10:24:24.7   6.640  -73.090  160.0
   360      4/ 1/76 19:21:15.9   6.819  -72.993  192.0
   422      7/ 2/78  2:49:18.1   6.551  -73.019  164.0
   502      8/15/80 21:30:46.3   6.809  -73.222  168.0
   690     12/ 3/85 17:52:24.2   6.779  -72.972  145.0
   696      3/31/86  1:19:24.6   6.917  -73.047  165.0
   706      6/29/86 20:11:50.9   6.760  -73.204  185.0
  1011 72189   8  8 18 26 50.0  -17.50   167.50   30.0
  1012 72389 12 12  7 19 50.0  -17.50   167.50   30.0
  1013 72489 11 11  8 56 50.0  -17.50   167.50   30.0
```

```
c After last data there must be an 'e' in column 1
end
```

3.5 The phase arrival time file *phafile*

This following file is 'example.PHA'. Note the extensive comments at the beginning of the file.

```
c 22 August 1993: Data read by Scott Davis
c  for 8 Bucaramanga earthquakes and data
c  read by Dan Olson for three Vanuatu earthquakes
c*****************************************************
c Formats for the phase file:
c First column = 'c'
c   This line is a comment line - ignored by the program,
c     except possibly printed out
c First column = '*'
c   This is first line of phase data for an earthquake
c First column = 'e'
c   This is last line read by program, no more phase data
c First column otherwise:
c   This is data for a particular phase
c   example:
c 800815 215019.   KOD  PKP      4
c yrmoda hrmisec   stn   ph      Q
c*****************************************************
c**********
******  42
c   42     2/26/65 23:36:12.2   6.900  -73.000  146.0
c     Times verified  7/19/93
65 226 235531.1  SHL  PKP      0    c
65 226 235612.5  SHL  pPKP     0    .
c  Note:  CC on BHP reads "?"
65 226 233748.7  BHP  P        0    c
65 226 233754.5  CAR  P        0    c
c 65 226 233825.9  QUI  P        2    . [Reread]
65 226 233825.4  QUI  P        2    .
c 65 226 234011.1  QUI  S        3    . [Reread]
65 226 234006.7  QUI  S        3    .
65 226 233859.9  TRN  P        2    .
65 226 234113.8  TRN  S        3    .
65 226 233910.5  SJG  P        1    .
65 226 234007.6  LPS  P        0    d
65 226 234023.8  NNA  P        0    d
65 226 234355.2  NNA  S        2    .
65 226 234157.0  ATL  P        0    d
65 226 234227.9  GEO  P        1    d
65 226 234306.0  GEO  pP       3    .
65 226 234318.4  GEO  sP       3    .
65 226 234246.7  SCP  P        0    d
65 226 234335.5  SCP  sP       2    .
65 226 234347.5  ALQ  P        0    d
65 226 234519.1  PDA  P        2    .
65 226 234752.0  COL  P        0    d
65 226 234831.7  COL  pP       2    .
65 226 234846.2  COL  sP       2    .
65 226 234842.6  KEV  P        2    .
```

```
65 226 234941.5   KEV  sP         2    .
65 226 234847.2   NUR  P          2    c
65 226 234927.3   NUR  pP         3    .
.
.
.
.
******1013
890724 0857 0.29 O24  P         2 ip-O  -0.01
890724 0857 4.70 O24  S         2 ip-O
890724 0857 0.57 O23  P         2 ip-O  -0.02
890724 0857 5.26 O23  S         2 ip-O
890724 0857 1.33 O25  P         2 ip-O  -0.01
890724 0857 7.17 O25  S         2 ip-O
89 724  857 3.37 DVP  P         1  P+O   0.00
89 724  857 8.93 DVP  S         2  P+O  -0.41
890724 0857 3.22 O28  P         2 ep-O  -0.06
890724 085710.14 O28  S         2 ep-O   0.02
890724 0857 3.77 O27  P         2 ep-O   0.02
890724 085711.38 O27  S         2 ep-O   0.18
89 724  857 4.42 NGA  P         1  P-O   0.00
89 724  85710.71 NGA  S         2  P-O
890724 0857 4.52 O30  P         3 ep O  -0.71
890724 085713.36 O30  S         2 ep O   0.14
89 724  857 5.50 PVC  P         4  P O   0.63
89 724  85712.00 PVC  S         4  P O  -0.04
89 724  857 7.31 EME  P         2  P O   0.23
89 724  85715.98 EME  S         3  P O   0.33
89 724  857 7.07 RTV  P         2  P O   0.04
89 724  85715.02 RTV  S         2  P O
c After last data there must be an 'e' in column 1
end
```

3.6 The station correction variable file *corfile*

The following file is 'example.cor', supplied as sample data with this program. Note the extensive comments at the beginning of the file; then, the information defining the station correction variable architecture, set off by asterisks. In this example note that the initial station corrections are nonzero.

```
c** 19 July 1993
c*********************************************************
c**
c This file sets up station corrections for JHD or single
c  event location.   A separate station correction will
c  be applied and/or calculated for each line
c  beginning with 'sc'.  Following each sc-line, there
c  will be up to nine different station-phase combinations
c  for which this station correction applies.
c  This allows flexibility in several ways:
c   1. The station name can change, but keep the
c        same correction.
c   2. The same correction can be applied to two nearby
c        stations
c   3. One can apply a multiplicative factor and use
```

```
c        the same correction for two phases, e. g., with
c        the correction for S being 1.7 time the correction
c        for P.
c*****************************************************
c First column = 'c'
c   This line is a comment line - ignored by the program,
c     except possibly printed out
c**
c First column = 'e'
c   This is end of data
c**
c Format for the sc-line:
c  on input: initial station correction in ('sc',1x,f8.3)
c  on output: new station correction, uncertainty,
c   number of data, sum of data weights
c    in ('sc',1x,f8.3,f8.3,i3,f8.2)
c Format for the station-phase combination line:
c  on input: station name, phase name, multiplier
c   in (a4,1x,a8,f6.2)
c  on output: station name, phase name, multiplier, number of data
c   in (a4,1x,a8,f6.2,i3)
c*********************************************************
*
sc  +01.000
ALQ  P          1.00
ALQ  pP         1.00
ALQ  sP         1.00
*
sc  +02.000
ATL  P          1.00
ATL  pP         1.00
ATL  sP         1.00
*
sc  +00.000
BAO  P          1.00
BDF  P          1.00
BDF  pP         1.00
BDF  sP         1.00
*
.
.
.
.
sc  -5.900
SJG  P          1.00
*
sc  +00.700
SHL  PKP        1.00
SHL  pPKP       1.00
*
sc  +00.000
SPA  P
*       1.00
sc  +00.000
TRN  P          1.00
*
sc  +00.000
UNM  P          1.00
```

```
*
sc   +00.000
TPM  P          1.00
*
sc   +01.800
WES  P          1.00
WES  pP         1.00
WES  sP         1.00
*
sc   +01.000
WIN  P          1.00
WIN  pP         1.00
WIN  sP         1.00
*
end
```

3.7 The flat-earth velocity model file *flvmfil*

The following file is 'example.flvm', supplied as sample data with this program. For each new velocity model the file lines are the number of layers, the layer velocities (in km/sec), the layer thicknesses (in km), and the ratio of the P and S velocities.

```
 5
1.5 3.5 5.5 7.5 8.0
5.0 1.0 3.0 5.0 1000.
1.95
 2
5.5 8.0
9.0 1000.
1.71
```

SECTION 4: Sample Output Data for **TexFlex** version 0.5

4.1 The program summary information file *statfil*

This is the primary summary output file for **TexFlex** program runs. The file below is 'example.Stat.out', generated by running the program with 'example.3.DVR'. The file begins by summarizing the defaults specified in file *dvrfile*.

```
*************************************************
*** Output from program TexFlex Version 0.5 ***
*************************************************
*************************************************
*** Default Summary: ***
*************************************************
  DeltAz =  2: 1 is round earth
               2 is geocentric
  DepthF =  0: 0 is free depth
```

```
                1 is fixed depth
  EllipC =   0: 0 is no ellipticity correction applied;
                1 is ellipticity correction applied
  NumbIt =   9: Number of Iterations
  NumCol =  80: Number of Columns for Residual Output
  RelocX =   0: 0 is single-event relocation
                1  is JHD by Frohlich 1979 method
  DMvMax =    30.: Maximum distance moved/iteration
  DepFly =   1: 0 allows negative focal depths
  VolFnd =   1: 1 means volume of hypocenters found
  TravTm =   1: 1 is JB P tables for travel times
                2 is IASPEI 91 tables
                3 is flat-earth travel times
                4 is station-dependent times
  WtType =   0: 1 is equal weight function for phase weighting
                2 is ramp  weight function
                3 is Gaussean  weight function
                   flat width, ramp width =   4.0  4.0
  WtUser =   0: 0 phase weights are independent of user-assigned quality
                1 weights vary depending on user assigned function
  WtItDp =   0: 0 weights are iteration independent
                1 weight functions change for latter iterations
  WtAzDp =   0: 0 weight functions are azimuthally independent
                1 weight functions depend on azimuth
**************************************************
** subroutine readep - searching for  8 epicenters
**************************************************
** open trial epicenter file: example.HYP
**   11 epicenters checked in file example.HYP
**subroutine readep complete**    8 hypocenters found
** close trial epicenter file
**************************************************
```

In the following section the program finds the convex polyhedron bounding the starting hypocenters. The output specifies the event identification numbers for the each of the triangular faces, and also specifies whether the face is visible when viewed from the north or south, east or west, and top or bottom. For example, the first face found by the program is formed by events 42, 151, and 690 in 'example.HYP', and is visible when viewed from the north, from the west, or from above. The bounding polyhedron has 12 faces and 18 edges connecting 8 hypocenters, and it has volume 12,913.6 km^3 and surface area 3,363 km^2.

```
**************************************************
***1st Face*** Face   1:    42  151  690 **nP =  3 NWT
***Add Edge*** Face   2:    42  690  696 **nP =  4 SWT
***Add Edge*** Face   3:   151  422  690 **nP =  5 NWT
***ReduPoly*** Face   4:    42  151  422 **nP =  4 NWT
***Add Edge*** Face   5:   360  422  690 **nP =  5 NEB
***ReduPoly*** Face   6:   360  690  696 **nP =  4 SET
***Add Edge*** Face   7:    42  422  706 **nP =  5 NWT
***ReduPoly*** Face   8:   360  422  706 **nP =  4 NWB
***ReduPoly*** Face   9:   360  696  706 **nP =  3 SWB
***Add Edge*** Face  10:    42  502  696 **nP =  4 SWT
***ReduPoly*** Face  11:    42  502  706 **nP =  3 NWT
```

```
***ReduPoly*** Face   12:  502  696  706 **nP =  2 SWB
***************************************
***norder,xjiggle = 1 0.0000
***nFaces =   12
***nEdges =   18
***Number Edge Points =   8
***Number Unpaired edges =  0
***volume of polyhedron =  12913.60
***surface area of polyhedron =    3363.18
***Shape Factor =  0.889
**************************************************
** open station correction file: example.COR
** subroutine readcor finds 27 independent station corrections
** close station correction file
** open station location file: world.STN
**2084 stations read in file world.STN
**************************************************
** subroutine readpha - read phase file **
```

The next section of this file reports information concerning the locations at each iteration. Information for the first iteration is for the trial hypocenters. In addition to the hypocenters, for each event the program reports the current rms residual (sec), the sum of the phase weights, the azimuthal gap (degrees), and the distance moved (km) since the previous iteration.

```
**************************************************
***** Iteration  1 ** Single-Event: Fixed Station Corrections**
   42     2/26/65 23:36:12.70   6.770  -73.200 158.00   3.09  15.00 119.91    0.0
  151     7/29/67 10:24:24.70   6.640  -73.090 160.00   1.90  21.00  71.36    0.0
  360     4/ 1/76 19:21:15.90   6.819  -72.993 192.00   2.50  11.00  82.54    0.0
  422     7/ 2/78  2:49:18.10   6.551  -73.019 164.00   2.21  10.00 109.12    0.0
  502     8/15/80 21:30:46.30   6.809  -73.222 168.00   2.09  13.00 116.34    0.0
  690    12/ 3/85 17:52:24.20   6.779  -72.972 145.00   2.10   8.00 121.90    0.0
  696     3/31/86  1:19:24.60   6.917  -73.047 165.00   3.43  10.00 108.48    0.0
  706     6/29/86 20:11:50.90   6.760  -73.204 185.00   2.27  11.00 108.96    0.0
***** Iteration  2 ** Single-Event: Fixed Station Corrections**
   42     2/26/65 23:36:15.09   6.934  -73.013 163.92   2.10  15.00 119.90   28.2
  151     7/29/67 10:24:25.70   6.820  -73.069 171.42   1.10  21.00  70.49   23.1
  360     4/ 1/76 19:21:16.59   6.761  -73.052 178.42   1.65  11.00  83.82   16.4
  422     7/ 2/78  2:49:19.20   6.732  -73.006 176.40   1.33  10.00 110.73   23.6
  502     8/15/80 21:30:47.45   6.709  -72.976 172.30   1.39  13.00 109.85   29.6
  690    12/ 3/85 17:52:25.43   6.745  -72.950 174.67   1.20   8.00 122.22   30.0
  696     3/31/86  1:19:26.04   6.678  -73.085 178.23   2.64  10.00 109.48   29.9
  706     6/29/86 20:11:52.15   6.834  -73.021 174.77   0.97  11.00 108.88   24.1
.
.
.
.***** Iteration  9 ** Single-Event: Fixed Station Corrections**
**************************************************
** subroutine readpha - read phase file **
   42     2/26/65 23:36:14.83   6.912  -73.015 163.42   2.09  15.00 120.00    0.0
  151     7/29/67 10:24:25.76   6.822  -73.067 172.01   1.10  21.00  70.47    0.0
  360     4/ 1/76 19:21:16.52   6.760  -73.057 177.72   1.65  11.00  83.93    0.0
  422     7/ 2/78  2:49:19.28   6.735  -73.000 177.01   1.33  10.00 110.65    0.0
  502     8/15/80 21:30:47.36   6.709  -72.977 171.35   1.39  13.00 109.88    0.0
  690    12/ 3/85 17:52:25.25   6.726  -72.945 176.39   1.17   8.00 122.38    0.0
  696     3/31/86  1:19:26.05   6.678  -73.085 178.24   2.64  10.00 109.48    0.0
  706     6/29/86 20:11:52.07   6.832  -73.021 174.16   0.97  11.00 108.89    0.0
```

The program now reports information about the convex bounding polyhedron for the relocated hypocenters. Then, it reports all phase arrivals which have unusually large residuals, and summarizes the azimuthal distribution of the observing stations for all of the hypocenters.

```
*************************************************
***1st Face*** Face    1:   42  151  706 **nP =  3 SWB
***Add Edge*** Face    2:   42  151  696 **nP =  4 NWT
***Add Edge*** Face    3:   42  690  706 **nP =  5 SEB
***Add Edge*** Face    4:   42  502  690 **nP =  6 NET
***ReduPoly*** Face    5:   42  502  696 **nP =  5 NWT
***ReduPoly*** Face    6:  502  690  696 **nP =  4 NEB
***Add Edge*** Face    7:  151  360  706 **nP =  5 SWB
***ReduPoly*** Face    8:  151  360  696 **nP =  4 SWB
***ReduPoly*** Face    9:  360  690  706 **nP =  3 SEB
***ReduPoly*** Face   10:  360  690  696 **nP =  2 SEB
**************************************
***norder,xjiggle = 1 0.0000
***nFaces =    10
***nEdges =    15
***Number Edge Points =   7
***Number Unpaired edges =  0
***volume of polyhedron =    858.03
***surface area of polyhedron =     640.69
***Shape Factor =  0.825
*************************************************
** subroutine readpha - read phase file **
* LARGE RESIDUAL *: R =   -5.8 sec, Wt = 1.00 for #  696:     3/31/86 at TRN  P       *
LARGE RESIDUAL *: R =    6.6 sec, Wt = 1.00 for #  696:     3/31/86 at SJG  P
*************************************************

***Azimuthal Distribution of Data***
                         NE          SE           SW           NW
  Earthquake       *0-30* -60* -90*-120*-150*-180*-210*-240*-270*-300*-330*-360*  Gap
  42      2/26/65*    6    3    1    0    0    0    1    1    0    2    1    9   120.0
 151      7/29/67*    9    2    1    2    0    0    3    1    3    5    1    7    70.5
 360      4/ 1/76*    4    3    1    0    2    0    2    0    2    1    3    5    83.9
 422      7/ 2/78*    5    2    0    0    2    0    2    0    0    0    3    2   110.6
 502      8/15/80*    4    1    1    0    1    0    3    0    0    0    3    8   109.9
 690     12/ 3/85*    5    1    0    0    0    0    2    0    1    1    1    2   122.4
 696      3/31/86*    8    2    2    0    0    0    2    0    0    1    3    0   109.5
 706      6/29/86*    9    1    1    0    0    0    2    0    0    1    3    3   108.9
*************************************************
** open station correction file: example.COR
** close station correction file
*************************************************
** open trial epicenter file: example.HYP
** close trial epicenter file
*************************************************
```

Finally, the program reports individual residuals for each hypocenter and station, separating the data as specified in *corfile*. It also provides means, standard deviations, and sums of weights for each event and for each station.

```
          **** Residuals ****

  Earthquake     *SC Var  1*SC Var  2*SC Var  3*SC Var  4*SC Var  5
                 *ALQ P    *ATL P    *BDF P    *BEC P    *BHP P    *UPA P
```

```
 42      2/26/65*  -0.91  *  -0.97  *         *         *  -0.73
151      7/29/67*  -0.16  *  -1.33  *         *  -0.05  *  -0.67
360      4/ 1/76*   0.51  *  -1.71  *   0.16  *         *  -0.79
422      7/ 2/78*  -0.28  *         *   1.12  *         *
502      8/15/80*  -0.99  *   0.12  *   0.61  *         *
690     12/ 3/85*  -0.07  *         *         *         *
696      3/31/86*  -0.61  *         *         *         *            -1.33
706      6/29/86*  -0.17  *         *         *  -0.67  *            -0.98

      Mean         -0.33     -0.97      0.63     -0.36     -0.90
      St. Dev.      0.46      0.68      0.39      0.31      0.24
      Sum Wts. *     8.0  *    4.0  *    3.0  *    2.0  *    5.0


         **** Residuals ****

 Earthquake      *SC Var  6*SC Var  7*SC Var  8*SC Var 10*SC Var 12*SC Var 13
                 *BOG P    *CAR P    *COL P    *GEO P    *KEV P    *KIP P

 42      2/26/65*         *  -1.27  *  -0.84  *  -0.26  *  -0.13  *
151      7/29/67*   0.76  *         *  -0.36  *   0.09  *   1.01  *   0.82
360      4/ 1/76*   0.60  *  -2.61  *   0.09  *         *         *
422      7/ 2/78*   0.24  *  -2.42  *  -0.71  *         *         *
502      8/15/80*   0.31  *  -2.03  *  -1.49  *   1.89  *         *
690     12/ 3/85*   0.21  *  -1.76  *  -0.02  *         *         *
696      3/31/86*   1.11  *  -4.42  *         *         *  -0.24  *
706      6/29/86*   0.71  *         *   0.11  *         *   0.97  *

      Mean          0.56     -2.42     -0.46      0.57      0.40      0.82
      St. Dev.      0.31      0.99      0.55      0.94      0.59      0.00
      Sum Wts. *     7.0  *    6.0  *    7.0  *    3.0  *    4.0  *    1.0


         **** Residuals ****

 Earthquake                *SC Var 14*SC Var 15*SC Var 16*SC Var 17*SC Var 18
                 *KIP P    *LPS P    *NNA P    *NUR P    *PDA P    *QUI P

 42      2/26/65          *  -0.67  *  -3.47  *   0.74  *   4.63  *   3.96
151      7/29/67    0.82  *  -0.32  *   0.18  *   0.20  *  -0.64  *  -0.51
360      4/ 1/76          *         *   0.36  *  -0.33  *         *
422      7/ 2/78          *         *  -0.25  *  -0.55  *         *
502      8/15/80          *         *  -0.18  *  -0.43  *         *
690     12/ 3/85          *  -0.46  *         *         *         *
696      3/31/86          *         *         *  -0.33  *         *
706      6/29/86          *         *         *  -0.31  *         *

      Mean                   -0.48     -0.67     -0.14      2.00      1.73
      St. Dev.                0.14      1.42      0.42      2.63      2.24
      Sum Wts.            *    3.0  *    5.0  *    7.0  *    2.0  *    2.0


         **** Residuals ****

 Earthquake      *SC Var 19*SC Var 20*SC Var 22*SC Var 23*SC Var 24*SC Var 26
                 *SCP P    *SJG P    *SPA P    *TRN P    *UNM P    *WES P

 42      2/26/65*  -0.01  *   2.26  *         *  -2.32  *         *
151      7/29/67*  -1.36  *   3.16  *  -0.54  *  -2.42  *   1.07  *   0.10
360      4/ 1/76*         *   4.31  *         *  -0.55  *         *
422      7/ 2/78*  -0.90  *   2.80  *         *         *         *   0.95
502      8/15/80*   2.85  *         *  -0.40  *   1.46  *         *  -1.77
690     12/ 3/85*         *   2.54  *   0.51  *         *         *  -0.95
```

```
696     3/31/86*          *  6.63  *  0.94  * -0.45  *         *  -1.31
706     6/29/86*          *  2.03  *  0.05  * -1.78  *         *   0.09

      Mean        0.14       3.18     0.11     -1.70     1.07     -0.48
      St. Dev.    1.64       1.50     0.55      2.10     0.00      0.94
      Sum Wts. *   4.0  *     8.0  *   5.0  *    7.0  *   1.0  *    6.0


        **** Residuals ****

 Earthquake     *SC Var 27   Mean     St. Dev.  Sum Wts.
                *WIN P

 42      2/26/65*              0.00      2.09  *   15.0
151      7/29/67*  0.95        0.00      1.10  *   21.0
360      4/ 1/76*              0.14      1.65  *   12.0
422      7/ 2/78*              0.00      1.33  *   10.0
502      8/15/80*              0.00      1.39  *   13.0
690     12/ 3/85*              0.00      1.17  *    8.0
696      3/31/86*             -0.53      3.03  *   11.0
706      6/29/86*              0.00      0.97  *   11.0

      Mean        0.95
      St. Dev.    0.00                   1.70
      Sum Wts. *   1.0
```

4.2 The one-line-per hypocenter output file *hypfOut*

This is file 'example.hyp.out', generated by running the program with driver file 'example.3.DVR'. In addition to the relocated hypocenters, the file includes the root-mean square of the residuals (sec). As useful information for plotting error ellipsoids on maps the file includes the dimensions of the two epicentral axes of the error ellipsoid (in km), the azimuth of the first error ellipsoid axis (in degrees E of N), the sum of the weights of the observations used in the relocation, and the azimuthal gap (in degrees) of the stations used in the relocation. Note that if `Noprin` is 1 in 'example.3.DVR', then the comments in the file 'example.HYP' will be repeated in 'example.hyp.out'.

```
c    #        name       origin        lat      long     depth    RMS   ElA1   ElA2   ANoE    SumW    Gap
    42      2/26/65 23 36 14.83   6.912  -73.015  163.42  2.09    24.    30.    41.   15.00 120.00
   151      7/29/67 10 24 25.76   6.822  -73.067  172.01  1.10    15.    19.    17.   21.00  70.47
   360      4/ 1/76 19 21 16.52   6.760  -73.057  177.72  1.65    22.    27.    43.   12.00  83.93
   422      7/ 2/78  2 49 19.28   6.735  -73.000  177.01  1.33    17.    30.    20.   10.00 110.65
   502      8/15/80 21 30 47.36   6.709  -72.977  171.35  1.39    20.    28.    17.   13.00 109.88
   690     12/ 3/85 17 52 25.25   6.726  -72.945  176.38  1.17    14.    19.    30.    8.00 122.38
   696      3/31/86  1 19 26.05   6.678  -73.085  178.24  3.03    52.    35.   318.   11.00 109.48
   706      6/29/86 20 11 52.07   6.832  -73.021  174.16  0.97    12.    16.    31.   11.00 108.89
  1011 72189  8  8 18 26 50.0  -17.50   167.50    30.0
  1012 72389 12 12  7 19 50.0  -17.50   167.50    30.0
  1013 72489 11 11  8 56 50.0  -17.50   167.50    30.0
end
```

4.3 The detailed hypocenter-residual output file *hypfSum*

This is part of file 'example.hyp.sum', generated by running the program with driver file 'example.3.DVR.' For each phase arrival for each seismic event, this file summarizes the arrival time, phase quality, residual, assigned weight, distance, and azimuth. Then, it reports the eigenvalues and eigenvectors of the covariance matrix for the statistical uncertainty in the relocation.

```
 .
 .
***************************************************
 id #            hr mi sec      lat      lon      dep      mved
  706     6/29/86 20:11:52.07   6.832  -73.021 174.16    0.00
***************************************************
 stn phase         time           qual    res     wt     dist   az
 UPA  P         86 629 201325.7    0    -0.98  1.00    6.80 288.4
 BOG  P         86 629 201233.8    0     0.71  1.00    2.43 204.9
 TRN  P         86 629 201437.8    0    -1.78  1.00   12.09  71.1
 SJG  P         86 629 201448.1    0     2.03  1.00   13.06  30.3
 BEC  P         86 629 201717.1    2    -0.67  1.00   26.58  15.9
 ALQ  P         86 629 201924.9    0    -0.17  1.00   41.57 316.9
 WES  P         86 629 201835.1    2     0.09  1.00   35.44   2.2
 COL  P         86 629 202329.4    0     0.11  1.00   77.40 335.1
 KEV  P         86 629 202420.1    2     0.97  1.00   87.10  20.1
 NUR  P         86 629 202422.5    1    -0.31  1.00   87.87  29.4
 SPA  P         86 629 2025 4.0    2     0.05  1.00   96.79 180.0
                              ave res    0.00 11.00         108.9 gap
                              rms res    0.97 sum wts

 1.00-0.06 0.06 0.05     0.97 time - sec
 0.08 0.85-0.51 0.00    12.05 dist - km:   31. deg E of N   0. from vertical
-0.02 0.52 0.86 0.00    16.04 dist - km:  301. deg E of N   0. from vertical
-0.05 0.01 0.00 1.00    16.12 dist - Z:    34. deg E of N  90. from vertical
```

4.4 The phase arrival time output file *phafOut*

The following file is part of 'example.pha.out', produced by running the program with 'example.3.DVR' when NoPrin is 1. Note that the file includes information about the distance and azimuth (in degrees) of stations for all observations used, as well as the observed travel time, the calculated travel time, and the residual (in seconds). If the reported station-phase combination does not match any designated in *corfile*, then the program reports "no sta or pha." If the station-phase combination matches but the program cannot calculate a travel time for this phase, then the program reports "TT - phase." In this case, the program so reports because file 'example.DVR' designates that travel times be JB P times. thus, often the program will not calculate times for many legitimate phases like pP and sP. Note that when `NoPrin` is 1, 'example.pha.out' includes all comments from the file 'example.PHA'.

```
c**********
****** 42
c   42     2/26/65 23:36:12.2   6.900  -73.000  146.0
c      Times verified  7/19/93
65 226 235531.1  SHL  PKP      0 144.53  23.9 oTT1156.3                  TT - phase
65 226 235612.5  SHL  pPKP     0 144.53  23.9 oTT1197.7                  TT - phase
c  Note:  CC on BHP reads "?"
65 226 233748.7  BHP  P        0   6.79 287.5 oTT  93.9 cTT  98.2  -0.7
65 226 233754.5  CAR  P        0   7.00  59.2 oTT  99.7 cTT 100.9  -1.3
c 65 226 233825.9  QUI  P        2     . [Reread]
65 226 233825.4  QUI  P        2   8.94 217.8 oTT 130.6 cTT 126.6   4.0
c 65 226 234011.1  QUI  S        3     . [Reread]
65 226 234006.7  QUI  S        3     .                             no sta or pha
65 226 233859.9  TRN  P        2  12.06  71.5 oTT 165.1 cTT 167.4  -2.3
65 226 234113.8  TRN  S        3     .                             no sta or pha
65 226 233910.5  SJG  P        1  12.99  30.5 oTT 175.7 cTT 179.3   2.3
65 226 234007.6  LPS  P        0  17.47 296.0 oTT 232.8 cTT 234.7  -0.7
65 226 234023.8  NNA  P        0  19.16 191.5 oTT 249.0 cTT 252.9  -3.5
65 226 234355.2  NNA  S        2     .                             no sta or pha
65 226 234157.0  ATL  P        0  28.40 339.8 oTT 342.2 cTT 341.1  -1.0
65 226 234227.9  GEO  P        1  32.05 354.0 oTT 373.1 cTT 373.3  -0.3
65 226 234306.0  GEO  pP       3  32.05 354.0 oTT 411.2               TT - phase
65 226 234318.4  GEO  sP       3  32.05 354.0 oTT 423.6               TT - phase
65 226 234246.7  SCP  P        0  34.02 353.4 oTT 391.9 cTT 390.1   0.0
65 226 234335.5  SCP  sP       2  34.02 353.4 oTT 440.7               TT - phase
65 226 234347.5  ALQ  P        0  41.52 316.9 oTT 452.7 cTT 452.6  -0.9
65 226 234519.1  PDA  P        2  52.69  47.2 oTT 544.3 cTT 539.6   4.6
65 226 234752.0  COL  P        0  77.33 335.0 oTT 697.2 cTT 698.0  -0.8
65 226 234831.7  COL  pP       2  77.33 335.0 oTT 736.9               TT - phase
65 226 234846.2  COL  sP       2  77.33 335.0 oTT 751.4               TT - phase
65 226 234842.6  KEV  P        2  87.02  20.1 oTT 747.8 cTT 747.9  -0.1
65 226 234941.5  KEV  sP       2  87.02  20.1 oTT 806.7               TT - phase
65 226 234847.2  NUR  P        2  87.80  29.4 oTT 752.4 cTT 751.6   0.7
65 226 234927.3  NUR  pP       3  87.80  29.4 oTT 792.5               TT - phase
```

4.5 The station correction output file *corfOut*

This is file 'example.cor.out', generated when the program is run with driver file 'example.3.DVR'. For each station correction variable, the file reports the mean of and standard deviation the residuals and the sum of the weights. For example, for this program run at station ALQ the mean is -0.335, the standard deviation of 0.456, and the sum of the weights is 8.0. Note that the comments at the beginning of the file 'example.COR' will be repeated in 'example.cor.out' if `NoPrin` is 1.

```
*
sc    1.000      -0.335 +-  0.456    8.0
ALQ  P         1.00
ALQ  pP        1.00
ALQ  sP        1.00
*
sc    2.000      -0.973 +-  0.681    4.0
ATL  P         1.00
ATL  pP        1.00
ATL  sP        1.00
*
sc    0.000       0.628 +-  0.394    3.0
BAO  P         1.00
```

```
BDF  P        1.00
BDF  pP       1.00
BDF  sP       1.00
*
.
.
.
.
sc   -5.900       3.178 +-  1.502    8.0
SJG  P        1.00
*
sc    0.700       0.000 +-  0.000    0.0
SHL  PKP      1.00
SHL  pPKP     1.00
*
sc    0.000       0.111 +-  0.552    5.0
SPA  P
*       1.00
sc    0.000      -1.701 +-  2.105    7.0
TRN  P        1.00
*
sc    0.000       1.067 +-  0.000    1.0
UNM  P        1.00
*
sc    0.000       0.000 +-  0.000    0.0
TPM  P        1.00
*
sc    1.800      -0.483 +-  0.938    6.0
WES  P        1.00
WES  pP       1.00
WES  sP       1.00
*
sc    1.000       0.954 +-  0.000    1.0
WIN  P        1.00
WIN  pP       1.00
WIN  sP       1.00
*
end
```

<u>SECTION 5</u>: Description of Individual **TexFlex** Subroutines

This section describes individual subroutines in the **TexFlex** program package as of July 1993. In this section, we organize the various routines into several groups:

5.1. Main Program and Subroutines for Setting Initial Parameters
5.2. Subroutines for Reading Input Data
5.3. Subroutines for Relocating Seismic Events
5.4. Subroutines for Calculating Travel Times
5.5. Subroutines Which Manipulate Formats or Display Output Data
5.6. Additional Ancillary Subroutines for Matrix and Vector Manipulation, Error Analysis, etc.

5.7. Subroutines for Determining the Volume of a Group of Hypocenters

In this section, for reader convenience all program and subroutine names are in bold face type, and all names for variables within routines are in italics.

A difficulty in writing program description is that some readers are interested only in learning enough to use the program, whereas others want to understand the inner workings of the program so that they can modify or debug it. The approach in this section has been to attempt the middle ground, and to describe generally what each subroutine does, without becoming bogged down in tedious technical detail.

5.1 Main Program and Subroutines for Initializing Parameters

5.1.1**program **TexFlex**

This is the main program which calls all essential subroutines. As currently configured it:

- opens a file named 'driver' which reads a character string named *dvrfile* which is the name of the file containing all information about the proposed program run.
- calls subroutine **SetDef** to read file *dvrfile* setting initialization parameters controlling phase weights, number of iterations, choice of fixed depth or free depth relocation, choice of single-event or joint relocation, etc. This file also specifies file names *phafile*, *hypfile*, *stafile*, and *corfile* for files containing data input, file names *phafOut*, *hypfOut*, *hypfSum*, *corfOut*, and *statfil* for files containing data output, and closes with a list specifying identification numbers of the events selected for relocation.
- calls subroutine **readep** which reads trial hypocenters for all events from *hypfile*, including those selected for relocation.
- calls subroutine **readcor** which reads file *corfile* that specifies station corrections and grouping of station correction variables, and file *stafile* which contains station locations.
- calls subroutine **readpha** which reads phase information from *phafile* for all events, including those selected for relocation

- calls subroutine **reloc**, which relocates the selected events.

- again calls subroutines **readpha, readcor**, and **readep**, this time writing new versions of the input files *phafOut*, *hypfOut*, *hypfSum*, and *corfOut*, including information determined during relocation, such as relocated hypocenters, station-event distances and azimuths, station corrections, etc.

- calls subroutine **display**, which formats and writes a file *statfil* summarizing useful information about the relocation, including relocations from each iteration, tables of azimuthal distribution of stations for each event, and a table of travel time residuals for each station and each event.

- calls subroutine **VFnd** to calculate the volume of the smallest convex polyhedron enclosing the hypocenters. This subroutine call takes place only if the user so chooses by making the appropriate default change in *dvrfile*.

5.1.2**subroutine **SetDef**(*hypfile, phafile, corfile, stafile*)

This routine sets default parameters, then, reads *dvrfile* to reset them if desired. It returns the names of the four data input files to the main program.

- In *dvrfile*, if the first nine characters of any line are 'Comment: ' then the routine reads the line and ignores it.

- If the first nine characters are 'default: ' then the routine reads the next six characters as variable defin. The variable defin is a code specifying which default parameter is to be changed, and the remaining 40 characters reset the value of this parameter.

- For example, if the line is:
  ```
  default: NumbIt  20
  ```
  then the default for the number of iterations is reset to 20 iterations.

- Finally, the subroutine initializes routines for calculating travel times in both round- and flat-earth cases by calling subroutines **TravT** and **VMflat**.

5.2. Subroutines for Reading Input Data

5.2.1**subroutine **readep**(*hypfile, nq*)

The main program calls this subroutine twice-- once before relocation begins to find trial hypocenter information, and once after relocation is complete to print out hypocenter information in a convenient format. A 1-character print flag *p* controls printing. The print flag *p* = ' ' before relocation, and *p* = 'P' afterwards.

Prior to relocation, this subroutine reads the tail end of the file dvrfile to find all the identification numbers of events which will undergo relocation, and returns *nq*, the number of events. Then, it opens the file *hypfile* containing trial hypocenters, and finds the hypocenters for the specified events.

After relocation, the routine prints out trial hypocenter information in file *hypfOut* in a format virtually identical to that of *hypfile*, except that it also includes various statistics such as rms residual, azimuthal gap, sum of phase weights, etc., which may be useful for evaluating the quality of the relocated hypocenter.

5.2.2**subroutine **readcor**(*corfile, nsc, stafile*)

The main program calls this subroutine twice. Before relocation begins the subroutine reads in station locations from file *stafile*, then it sets up station corrections and the station correction variables using information in file *corfile*. After relocation is complete it prints out station correction information in file *corfOut* using a format which is identical to that in *corfile*. As in **readep**, the 1-character print flag *p* controls printing. The print flag *p* = ' ' before relocation, and *p* = 'P' afterwards.

The most delicate part of this routine concerns the determination and assignment of the *nsc* station correction variables. A unique feature of **TexFlex** is that any station correction variable may apply to several different stations or station-phase combinations. To take care of bookkeeping for this the subroutine constructs three arrays, *stnam(j,ksc)*, *phnam(j, ksc)* and *nspcomb(ksc)*. For the *ksc*th station correction variable, these arrays give the station name and the phase name for station-phase combination *j*, and total number of such station-phase combinations to be associated with that the *ksc*th station correction variable.

5.2.3**subroutine **readpha**(*phafile, nq, nsc, kit*)

The main program calls this subroutine twice, and in addition the relocation routines **reloc** and **relocSng** call it during each iteration of the relocation process. Before relocation begins the subroutine reads in phase arrival times from file *phafile*, during relocation upon each iteration it rereads these arrival times and calculates residual, then after relocation it prints out arrival times, residual information, and certain useful statistics in a file *phafOut* using a format which is identical to that in *phafile.*

As in **readep** and **readcor**, the 1-character print flag *p* controls printing. The print flag *p* = ' ' before relocation, and *p* = 'P' afterwards. The routine does not change the variables *nq*, *nsc*, and *kit*, which are the number of events, the number of station corrections, and the number of the current iteration, respectively.

5.2.4-5**functions **WtPha(***Res, rAveS, kit, kQual, kq, Az***)**, **kWhSec(***Az***)**

Function **WtPha** assigns a weight to each phase arrival time. The mathematical function which controls this weight can be either extremely simple or quite complex, depending on the options chosen by the user in file *dvrfile* and set in subroutine **SetDef**. See Section 2 for details of the weighting function, however, the weight may depend on one or all of the following:

- The travel time residual *Res*, if the user chooses either a boxcar with ramp or a Gaussean weighting scheme
- The average residual *rAveS* for arrivals at the particular station where the phase arrives, if the user chooses either a boxcar with ramp or a Gaussean weighting scheme
- The number of the iteration *kit*, if the user chooses an iteration dependent weighting scheme
- The user-assigned quality of the phase *kQual*
- The azimuth of the station where the phase arrives, if the user chooses an azimuthally dependent weighting scheme

When the user chooses azimuthally dependent weighting, the function **kwhSec(***Az***)** requires the internal event number *kq* to determine the azimuthal sector where the quake lies.

5.3. Subroutines for Relocating Seismic Events

5.3.1**subroutine **reloc**(*nq, nsc, phafile*)

This is the main umbrella program for relocating *nq* seismic events, regardless of whether by a single-event, JED, or JHD method. Here *nsc* is the number of station correction variables, and the method of relocation is controlled by variable *kRelocM*, which is contained in a common block.

The intent is that almost any relocation method can be attached to this program as a modular subroutine. However, as presently configured the relocation method requires that the file *phafile* is opened and read upon each iteration, as residuals are recalculated and accounted for in subroutine **readpha.** The matrix *Bnq* contains the 4X4 least squares matrices for each of the nq events.

5.3.2**subroutine **relocSng**(*nq, nsc, kflag, phafile, kit*)

For *nq* seismic events and *nsc* station correction variables, this routine performs event-by-event relocations using weighted least squares to find latitude, longitude, origin time, and focal depth (if focal depth is not fixed). The variable *kflag* = 0 for plain vanilla single-event relocations, while *kflag* = 1 for the "fast" JHD relocations described by Frohlich [1979]. At each iteration *kit* the routine calls **readpha** to read file phafile to determine residuals and do bookkeeping.

The relocation scheme itself is quite crude, as the routine solves the 4X4 least squares matrix *B* by the method of determinates. Thus, occasionally if the data is poorly conditioned *B* is singular, and the program crashes.

5.4. Subroutines for Calculating Travel Times

5.4.1**subroutine**TravT**(*method, del, depth, phase, elat, slat, selev, NoYes, Time, gradd, gradz*)

This is the main umbrella program for calculating all travel times, regardless of whether for a flat or round earth. The intent is that almost any method can be attached to this program as a modular subroutine as long as:

- units of travel time *Time* are seconds

- units for distances *del* are degrees
- units for depths *depth* are km
- the subroutine recognizes when it is asked to find a travel time for an unrecognizable phase. In this case, *NoYes* = 1
- the subroutine recognizes when it is asked to find a time outside the range of occurrence for the specified phase. In this case, *NoYes* = 2.

The other specified input variables are:

- *method* - a number for identifying the table or routine which determines the travel time.

method =	0 for initializing the subroutine. The routine should be initialized in case it is necessary to read travel time tables, velocity models, etc.
=	1 for JB P times
=	2 for IASPEI times for a variety of phases
=	3 for flat-earth times, calculated for a default model or for a model specified by the user.

- *elat, slat,* and *selev* are the longitude for the event and for the station, and the elevation of the station. If desired, this is used for calculating ellipticity corrections.
- *gradd* and *gradz* are the derivatives of travel times with respect to distance and depth, in units of sec/degree and sec/km, respectively.

5.4.2.1**subroutine **TravTJB**(*del, depth, phase, elat, slat, selev, NoYes, Time, gradd, gradz*)

This subroutines calculates JB P times by interpolating from a table.

5.4.2.2**subroutine **JBP**

This subroutine reads in the JB P table. The table includes distances from 0° to 100°, and 14 depths from 0 to 794 km. The depths are: 0 km, 33 km, and 12 intervals at 33 + k*63.38 km where k is 1 to 12.

5.4.3.1**subroutine**TravTIASP91**(*del, depth, phase, elat, slat, selev, NoYes, Time, gradd, gradz*)

This subroutine is a simple passthrough to the routines written by Ray Buland for calculating the IASPEI travel times. These were supplied to Cliff by IRIS. For additional information about the IASPEI times, see Kennett [1991] and Buland and Chapman [1983].

- These routines will not run unless the file 'iasp91.hed' and 'iasp91.tbl' are available. These are supplied with the **TexFlex** package, and are calculated using the IASPEI routines obtained from IRIS.

- Tests by Cliff indicate that the travel times calculated by **TexFlex** are identical to those published in the tables of Kennett [1991].

5.4.3.2**subroutine **phMatch**(*phase, phlist, n, kph, NoYes*)

Given an 8-character string *phase*, this routine finds an identical character string in position *kph* of the array *phlist*, which contains *n* phase names of available IASPEI phases. *NoYes* = 0 if such a phase exists, otherwise *NoYes* = 1.

5.4.3.3**subroutine **shortPh**(*phnamIn, phnamSh*)

This subroutine translates a name as used by the IASPEI routines into the shorthand form more used by observers. It does this by removing characters-- for example, if *phnamIn* = 'PKiKP ' or 'PKPbc ', *phnamSh* = 'PKP '; if *phnamin* = 'Pn ', *phnamSh* = 'P '.

5.4.3.4-5.4.3.27**subroutines **asnag1, assign, bkin, brnset, depcor, depset, findtt, fitspl, iupcor, pdecu, query, r4sort, retrns, spfit, tabin, tauin, tauspl, trtm, uctolc, umod, warn, tnoua, dasign, vexit**

These are all routines to calculate IASPEI travel times. They are supplied with TexFlex but written by Ray Buland.

5.4.4.1**subroutine **TravTflat**(*del, depth, phase, elat, slat, selev, NoYes, Time, gradd, gradz*)

For a flat earth model, this subroutine calculates the first arriving travel time and its derivatives for either P or S phases for a seismic event at a specified depth and distance, and a station of specified elevation.

- The routine requires that the earth model have at least two layers, and also that the velocity increases monotonically as depth increases (low velocity zones not allowed).
- The routine assumes that P and S velocities are simply proportional, thus calculating S times by multiplying the P times by a constant *vPS*.
- This subroutine was pirated from a program written by Cliff in about 1979 to locate earthquakes recorded jointly by OBS and land station networks [Frohlich, et al., 1982]. Thus, in principal it admits the existence of two different flat-earth velocity models and permits the user to specify which model is desirable for each station. However, presently **TexFlex** only allows the user to specify a single flat-earth model.

5.4.4.2**subroutine **VMflatSetup**(*nl1, vel1, th1, vpovs1, nl2, vel2, th2, vpovs2*)

This routine calculates several tables which are useful for calculating flat-earth travel times. As mentioned above, the routine admits of different two velocity models, although only the first is currently used by **TexFlex**. The number of layers, velocities (in km/sec), thicknesses (in km), and ratios of P to S velocity for the kth model are given by *nlk*, *velk, thk*, and *vpovsk.*

5.4.4.3-5.4.3.4**functions **cn**(*s,v1,v2*), **sn**(*s,v1,v2*)

These are utility functions called only by the routine **TravTflat**. If *s* is the sine of the angle of incidence in a layer with velocity *v1*, these functions give the cosine and sines of the angle of incidence in a layer with velocity *v2*.

5.5. Subroutines Which Manipulate Formats or Display Output Data

5.5.1-2**subroutine **display**(*nsc,nq*), **pageMake**(*kWhat,nChar,nq*)

Subroutine **display** formats a table of residuals, with each line representing the residuals at each station for one of the *nq* seismic events.

This is slightly tricky because each of the *nsc* station correction variables may include several stations and several different phases which we would like to group conveniently but keep separate within the table. Moreover, we would like this information to fit within a specified number *ncPage* of columns in case we wish to print out a hard copy of the table. This routine worries about all this nasty stuff, puts the table information in a large character array named *page*, then calls subroutine **pageMake** to print out the table in file *statfil.*

Subroutine **pageMake** basically just prints out *page*. In *page* each element is a line of text of width *nChar*, and there are something like *nq*+ 3 lines-- one for each seismic event and several header lines. Initialization occurs when *kWhat* = 0, and the routine fills *page* with blanks; when *kWhat* = 1 the routine prints out the completed page in file *statfil.*

5.5.3**subroutine **stringrd**(*kfin,string,c1st,kfout*)

Subroutines **readep, readcor,** and **readpha** call this subroutine repeatedly to process information from the input files *hypfile, stafile, corfile,* and *phafile.* **stringrd** reads a string of 95 characters named *string* from input device *kfin*, and it returns the first character *c1st* in case it is an indicator of some kind. However, if *c1st* = 'c', the routine recognizes that the string is a comment line. In this case the routine ignores *string* and reads the next line, or; if the print flag *p* = 'P' the routine first prints out *string* on device *kfout* before proceeding to read the next line.

5.5.4**subroutine **starline**(*kf*)

This routine prints out a line of asterisks ('*') on output device *kf*. Its sole purpose is to provide a distinctive marker to separate output lines as TexFlex proceeds through different input, relocation, and output sequences.

5.6. Additional Ancillary Subroutines for Matrix and Vector Manipulation, Error Analysis, etc.

5.6.1.1**subroutine **ErrAnal**(*kq,Ste*)

This subroutine uses the approach described by Willemann and Frohlich [1987] to calculate a classical single-event error ellipsoid for event *kq*, given the RMS residual of travel times *Ste*, and the covariance matrix *Bnq* found by subroutines **reloc** or **relocSng**. This subroutine calls the subroutine **jacobi** to find the eigenvalues and eigenvectors of *Bnq*.

5.6.1.2**subroutine **DirPrinAx**(*evec, az2, dp2, az3, dp3, az4, dp4*)

This routine takes the 4X4 matrix of normalized eigenvectors and returns the azimuths and dips of the three principal spatial eigenvectors.

5.6.2**subroutine **DelAz**(*kdelaz, xlat, xlon, slat, slon, del, az*)

This subroutine returns the distance *del* in degrees and azimuth *az* in degrees given the latitudes *xlat, slat* and longitudes *xlon, slon* of the seismic event and observing station. The variable *kdelaz* selects the method for determining distant azimuth-- if *kdelaz* = 1 the routine assumes that the earth is round and determines distance using cross products; if *kdelaz* = 2 the routine finds geocentric latitudes and applies Turner's method [see Bullen, 1965, pages 155, 181]

5.6.3**function **gap**(*azArr,naz*)

Given an array *azArr* containing *naz* elements, this routine finds the largest azimuthal gap between the elements.

5.6.4**function **hmstosec**(*nh, nm, sec*)

This function converts a time in hours, minutes and seconds to seconds after midnight.

5.6.5**subroutine **sectohms**(*tsec, nh, nm, sec*)

Given the number of seconds tsec since midnight, this subroutine converts it to the more usual notation with hours *nh*, minutes *nm*, and seconds *sec*.

5.6.6**subroutine **lltoRNE**(*xlat, xlon, R, Reast, Rnort*)

Given latitude *xlat* and longitude *xlon* of a point on the earth's surface, this routine returns three unit vectors: *R* extending from the earth's center to the given point, *Reast* extending from the point parallel to the earth's surface and directed eastward, and *Rnort* extending from the point parallel to the earth's surface and directed northward. Presently, this routine is called only by **DelAz** and only when *kdelaz* = 1.

5.6.7**subroutine **CROSSp**(*y, z, YcZ*)

Given 3-vectors *y* and *z*, this routine finds the cross product vector *YcZ* .

5.6.8**subroutine **XNORM**(*x*)

Given a 3 vector *x*, this routine normalizes *x* and returns it as a unit vector (unless $x = 0$).

5.6.9**function **DOTP**(*x, y*)

Given two 3 vectors *x* and *y*, this function returns their dot product.

5.6.10**subroutine **SolveDet**(*B, x*)

Given a 4X5 matrix *B* which represents the set of equations $Ax = R$ this routine finds the solution *x* by the method of determinates. It is only effective if the system is well behaved, with *A* nonsingular.

5.6.11**function **DET**(*a11, a21, a31,......a33, a43*)

Given the sixteen elements *aik* of a 4X4 matrix *A*, this function finds the determinate of *A*.

5.6.12**subroutine **jacobi**(*h, eigen, nn, mm, err*)

This routine finds eigenvalues and eigenvectors for a real symmetric matrix. At present this is called only by subroutine **ErrAnal** and used only to determine error ellipsoids from the 4X4 least squares matrices **Bnq**. Cliff pirated this routine from Mark Riedesel, who obtained it from Bob Parker.

For a symmetric matrix *h* of order *nn* (dimensioned *mm* in the calling program) this routine returns the eigenvalues of *h* along the diagonal of *h*, and it returns matrix *eigen* which contains the normalized eigenvectors of *h* as columns. The variable *err* is set by the user, and the routine proceeds till the largest off-diagonal term in *h* is at least *err* times smaller than the sum of the magnitudes of the eigenvalues.

5.7. Subroutines for Determining the Volume of a Group of Hypocenters

If the user sets the identifier code 'VolFnd' to 1 in the driver file *dvrfile*, **TexFlex** finds the volume of the smallest convex polyhedron which encloses a set of points, here, a group of seismic hypocenters. For this purpose the program uses the algorithm described in Frohlich [1992]. The principal subroutine that accomplishes this is subroutine **VolumeFind** which is called only the umbrella routine **VFnd**, which in turn is called only by the main program. This section of the manual describes **VolumeFind** and the subroutines called exclusively by **VolumeFind** (i.e., not called by other subroutines in **TexFlex**). Thus, the user would remove these subroutines and the calls to **VolumeFind** if he or she wished to implement a version of **TexFlex** without the volume-finding capability.

VolumeFind shares only two subroutines with other elements in **TexFlex**. These are **crossP** and **dotP**. Thus, if the user wished to add volume-finding capability to another Fortran program, he or she would append the subroutines in this section along with subroutines **crossP** and **dotP** to the candidate program.

5.7.1**subroutine **VFnd**(*nPoints*)

Subroutine **VFnd** is called only by the main program, and its sole purpose is to put the data in a fairly general Cartesian format for **VolumeFind** and its subroutines. In particular, **VFnd** creates a common block containing an array *r(3,500)* of three vectors of Cartesian coordinates of points in space, and an array *neqID(500)* which is the event identification numbers of the points.

5.7.2**subroutine **fixC**(*k, xlat, xlon, dep, r1, r2, r3*)

This subroutine, called only within VFnd, changes format for the kth seismic event location from a latitude, longitude depth representation to Cartesian coordinates, with units of km.

5.7.3**subroutine **VolumeFind**(*nPoints, norder, xjiggle, Vtot, SAtot, ShapeF, nUE*)

This is the principal subroutine for finding the volume *Vtot* in km^3 of the smallest convex polyhedron bounding a set of *nPoints* points in space. The routine also finds the surface area SAtot in km^2 of this volume, and the "shape factor" *ShapeF*, which essentially is the ratio of the volume to the surface area, normalized so that the *ShapeF* is 1.0 for perfect spheres, and less for all other shapes.

Essentially, the strategy in **VolumeFind** is to find triangular faces of the bounding polyhedron one by one, starting with the lowestmost face and working upward [see Frohlich, 1992]. The routine considers triples of points to define planes, and knows that such a triple defines a polyhedron face if all other points lie on one side of the plane. The search for polyhedron faces gains efficiency because the program searches triples of points going from lowest to highest points, and does not need to search points lower than the "bounding polygon" formed by the faces found so far.

The routine is more general and more complicated than would be required for **TexFlex**, but this might be useful if a user wished to pirate it for other uses. In particular, if *norder* is 2 or 3, the routine searches points from west to east, or north to south, rather than lowest or highest. Furthermore, if *xjiggle* is nonzero the routine adds a random component of length *xjiggle* to each point. This is useful if many of the points lie in or nearly within a plane, and thus they are too regular for **VolumeFind** to handle smoothly in its effort to find bounding triangles. If the program has worked properly, the edge of each triangular face is matched with the edge of another triangular faces, and the number of unmatched edges *nUE* is zero.

5.7.4**function **AreaTri**(*rA, rB, rC*)

Given three three-vectors *rA*, *rB*, and *rC* designating Cartesian points, this routine finds the area of the triangle formed by the points.

5.7.5**subroutine **fndPlane**(*r1, r2, r3, p, c*)

Given three three-vectors *r1*, *r2*, and *r3* designating Cartesian points, this routine finds the three-vector *p* and scalar *c* defining the plane $x \cdot p = c$ which contains the points.

5.7.6**subroutine **edgePlane**(*p, c, kbeg, kendc, nF, kchange*)

Given the plane defined by the three-vector *p* and the scalar *c*, this routine checks the *kbeg*th to the *kend*th points in the Cartesian array to find if all the points lie on the same side of the plane. If so, *kchange* = 0, and they form a new face for the bounding polyhedron.

5.7.7**subroutine **newPlane(*k1, k2, k3, nF, new*)**

Given a new face defined by points *k1, k2,* and *k3,* this routine determines whether the new face is distinct from all of the *nF* previously found faces. If so, *new* = 1, otherwise new is the number of the identical face.

5.7.8**subroutine **prntStuff(*norder, p, c, View*)**

Given a face defined by the three-vector *p* and scalar *c*, this routine determines whether the face faces up or down ('T' or 'B'), east or west ('E' or 'W'), or north or south ('N' or 'S') and forms a three character variable *View* to store this information. *View* is useful if one wishes to plot graphs of the bounding polyhedron. However, this information is not used in any significant way in **TexFlex** and thus **prntStuff** could be removed from the program. Unsurprisingly, determining view depends on the *norder*, the order of the coordinates in the Cartesian array.

5.7.9**subroutine **vecdif(*r1, r2, r12*)**

The three-vector *r12* is the difference between *r1* and *r2*.

5.7.10**subroutine **vecassign(*k1, k2, k3, r1, r2, r3*)**

This routine forms Cartesian three-vectors *r1, r2*, and *r3* for the points numbered *k1, k2,* and *k3* in array *r(3,500).*

5.7.11**subroutine **rePoly(*np*)**

As described in 5.7.3 above, **VolumeFind** keeps track of the *np* points in the "bounding polygon" formed by all the polyhedron faces found so far. After finding a new face and forming a new bounding polygon, this routine reorders the integers identifying the points making up the polygon. This reordering simplifies subsequent calculations.

5.7.12**subroutine **indexx(*n, kcoord, indx*)**

This routine indexes the *kcoord*th coordinate of the *n* points in array *r(3,500)* and stores the order in integer array *indx*., smallest to largest.

5.7.13**subroutine **intord**(*j1, j2, j3, k1, k2, k3*)

This routine reorders integers *j1, j2,* and *j3* from smallest to largest, returning *k1, k2,* and *k3*.

5.7.14**subroutine **ran1**(*idummy*)

Upon subsequent calls this routine generates repeatable pseudorandom numbers, and *idummy* is a dummy variable. It was pirated from Press et al. [1989]. This routine is not really necessary in the current version of **TexFlex**, since *xjiggle* is set to 0.0 in **VFnd** (see 5.7.3).