UNITED STATES
DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

CAL NET PROCEDURES FOR PROCESSING PHASE DATA AND DETERMINING HYPOCENTERS OF LOCAL EARTHQUAKES WITH UNIX

by

Jerry Eaton
Rick Lester
Rob Cockerham

345 Middlefield Road
Menlo Park, CA 94025

Open-File Report 81-110

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards (and stratigraphic nomenclature). (Any use of trade names is for descriptive purposes only and does not imply endorsement by the USGS.)

CONTENTS

PART I: OVERVIEW OF THE SYSTEM

Introduction

This paper is intended to fill several related purposes:

1) provide an overview of the Cal Net seismic data processing system for those with a need to learn and use the system,
2) provide a clear description of how the system is used to process Cal Net data,
3) provide an outline of the purpose and features of the programs employed in data processing, and
4) provide a set of FORTRAN listings of those programs to encourage users to learn how they work and to facilitate resolution of problems that arise in their routine use.

In processing Cal Net seismic data "by hand", the 16 mm Develocorder films are projected onto a digitizing table and "read" by means of a cursor that determines the x-y coordinates of selected points on the projected seismograms. A procedure for reading seismograms in this manner and a computer program for interpreting the character stream produced by the digitizer card punch were developed by Peter Ward, John Lahr, and Bill Ellsworth. The interpretive program (digi3) was subsequently amended and converted for use on MULTICS. This program has now been converted to run on the PDP 11/70 under UNIX, and it has been renamed digix to distinguish it from earlier versions. digix produces a phase-card list in the format required by HYPO71.

HYPO71 was trimmed down and modified to run on UNIX by Sam Stewart to provide a means of locating events detected by the Allan/Ellis Real-Time Earthquake Analysis System. This diminished version of HYPO71 has been modified further to simplify its use for daily processing of Cal Net events. The resulting program has been renamed hypo71l to distinguish it from earlier versions.

A number of problems arising from the relatively small size of the 11/70 memory or from differences between UNIX FORTRAN77 and MULTICS FORTRAN66 were encountered during the conversion of digix and hypo71l. These problems were solved by several auxiliary programs for error-checking, for presorting phase lists, for completing work left undone by hypo71l, etc.

To provide a convenient environment to permit a group of analysts to work on Cal Net data processing, it was necessary to create an appropriate set of directories in UNIX, to develop procedures to simplify the use of the programs employed in the data processing, and to establish conventions for naming data files whereby the file name specifies the nature and contents of the file. The system that was adopted is similar to the one that evolved over several years for processing Cal Net data in MULTICS.

It is assumed here that the reader has at least an elementary acquaintance with the UNIX operating system, such as is provided by "UNIX for Beginners--Second Edition" by Brian W. Kernigan. That short paper (14 pages) is reproduced as section UB2a (vol. 3) of the UNIX manual.

UNIX Directories for Cal Net Earthquake Data Processing

The directories and subdirectories for Cal Net data processing are specified in the following pathname/structure diagram:

```
               Directories      Subdirectories                          Groups

             ┌ /oldcal                                                  oldcal
/db/ncal     │ /netlib         [ /fortprog     [ /hypofort              netlib
             └ /calnet         ┌ /mmmyy     monthly subdirectories      calnet
                               │    .           e.g. may80                "
                               │    .                                     "
                               │    .                                     "
                               │  /rick         private                   "
                               │  /rob          working                   "
                               │  /carol        subdirectories            "
                               │  /sharon                                 "
                               │    .                                     "
                               │    .                                     "
                               └    .                                     "
```

Cal Net processing is carried out in various subdirectories of the directory /db/ncal/calnet. The cards from the table-top digitizer for a given day are read into the appropriate monthly subdirectory and named d3.mmmdda, where "mmmdd" is the film "day off" (e.g., may23 for the film day May 22-May 23) and the suffix "a" indicates the initial "pass" in analysis. The complete first pass analysis, resulting in a hypo71l printer output augmented by a list of missing stations, is carried out in the monthly directory. Gross errors such as mixed events, large timing errors, misidentified traces, etc., are corrected as part of the first pass analysis. The phase-card list file, mmmdda, which is produced by program digix and is corrected for large errors, is preserved in the monthly directory until all additional work on day mmmdd is complete. A list of preliminary locations, also, is saved in file "suma", which is generated by adding each day's summary-card list to suma when it is obtained.

Subsequent analysis, including rereads, addition of new stations, etc., is carried out in the appropriate analyst's "private" subdirectory. As successive groups of corrections are made to the phase-card list (in the private subdirectory) it is renamed mmmddb, mmmddc, etc., to distinguish the newly corrected list from earlier ones. When work on a particular day's data is complete and the phase-card list is in its final corrected form (e.g., mmmddg), the final phase-card list is copied back into the monthly subdirectory. Both the original and final phase-card lists are retained until data for the entire month have been processed to completion.

Naming of Data Files

The phase-card list for each day is designated by the film "day off", e.g., may23 for the film day May 22-May 23, plus a one-letter suffix to distinguish a current corrected version of the list from previous versions of the list. Thus, the first version of the phase-card list is named mmmdda (e.g., may23a) and subsequent versions are named mmmddb, etc. This root is retained in the names of all (nontransient) files that are based on the data contained in this phase-card list.

Names of these files and descriptions of their contents are as follows:

d3.mmmdda = digitizer card list read into UNIX

mmmdda.dig = diagnostic file from program digix

mmmdda = original phase-card list, produced by program digix

mmmdd (α=a,b,c,...) = phase-card list, original or corrected

f.mmmdd.α = sorted phase-card list, produced by program d80p

mmmddα.smp = hypo71l hypocenter summary-card list

mmmddα.pch = hypo71l "punch-card output" list: epicenter and station/phase data summary

mmmddα.prt = hypo71l printer output list: epicenter and station/phase data summary

mmmddα.pchm = mmmddα.pch augmented by missing station lists

mmmddα.prtm = mmmddα.prt augmented by missing station list

Several transient files are created and then destroyed during a "normal" processing run. If processing is interrupted, some of these files may remain in the working directory. They should be removed before a rerun is attempted. Such files are:

calstn	=	station list used by hypo71, etc.
data.err	=	list of errors detected by digchk
data.c	=	corrected phase list produced by digchk
hypo.input	=	hypo711 input file
hypo.smp	=	" summary card output
hypo.pch	=	" punch-card output list
hypo.prt	=	" printer output list

When day-by-day processing is complete on all days of a month, the appropriate monthly directory contains the set of corrected daily phase-card lists for the whole month (mmmddα) plus the summary of preliminary hypocenters (suma) for the month. The daily phase lists are then combined in proper order into a single monthly phase-card list called:

MMMYY (e.g. SEP80).

When this phase card list is processed by "runhypoplus", there results a set of monthly outputs:

MMMYY.smp	monthly hypocenter summary list
MMMYY.pch	monthly hypo711 "punch card" output
MMMYY.prt	" " printer "
MMMYY.prtm	monthly hypo711 printer output augmented by lists of missing stations.

netlib

To conserve file space and to insure uniformity in Cal Net data processing, the programs, station lists, parameter lists, etc., that are required for processing are maintained in directory netlib (/db/ncal/netlib). The programs, etc., in netlib are invoked by means of executable files that reference their absolute addresses in netlib. FORTRAN texts of the programs are stored in fortprog, a subdirectory of netlib, and in hypofort, a subdirectory of fortprog.

The current contents of netlib, fortprog, and hypofort are listed below.

```
/db/ncal/netlib
total 1571
-rwxrwxr-x 1 eaton       187 Oct  6 16:12 RMLN
-rwxrwxr-x 1 eaton       569 Oct  3 23:25 RUNLN
-rwxrwxr-x 1 lester      186 Oct  6 09:47 catparam
-rwxrwxr-x 1 eaton     48568 Oct  3 22:42 catprog
-rwxrwxr-x 1 eaton        45 Oct  3 22:30 cvc.lc
-rwxrwxr-x 1 eaton        45 Oct  3 22:30 cvc.uc
-rwxrwxr-x 1 eaton     44568 Oct  3 22:42 d80p
-rwxrwxr-x 1 eaton     32494 Oct  7 10:43 digchk
-rwxrwxr-x 1 eaton     62872 Oct  3 22:42 digix
-rwxrwxr-x 1 eaton        70 Oct  3 22:30 f
-rw-rw-r-- 1 eaton         0 Oct 13 10:15 file
drwxrwxr-x 3 eaton       464 Oct 13 10:01 fortprog
-rwxrwxr-x 1 eaton       810 Oct  9 13:11 hypar0
-rw-rw-r-- 1 lester      810 Oct  7 15:43 hypar0.a
-rwxrwxr-x 1 eaton       810 Oct  3 22:26 hypar1
-rwxrwxr-x 1 eaton     99342 Oct  3 22:45 hypo711
-rwxrwxr-x 1 eaton     33968 Oct  3 22:43 lerck
-rwxrwxr-x 1 lester    50846 Oct  6 10:18 miving
-rwxrwxr-x 1 eaton     50672 Oct  3 22:43 mizing
-rwxrwxr-x 1 lester    30294 Oct  7 15:48 nocast0
-rwxrwxr-x 1 eaton     29565 Oct  3 22:27 nocast0.2
-rw-rw-r-- 1 lester    26001 Oct  9 13:23 nocast0.fh
-rw-rw-r-- 1 lester    30131 Oct  7 15:43 nocast0.fh.1
-rwxrwxr-x 1 eaton     22194 Oct  3 22:27 nocast1
-rwxrwxr-x 1 eaton     18468 Oct  3 22:27 nocast77
-rwxrwxr-x 1 eaton     29160 Oct  3 22:27 nocasto
-rwxrwxr-x 1 eaton     40286 Oct  3 22:43 pltfm
-rwxrwxr-x 1 lester       98 Oct  6 09:56 pltfmparam
-rwxrwxr-x 1 eaton     35386 Oct  3 22:43 qrychk
-rwxrwxr-x 1 lester      693 Oct  6 09:47 qrylist
-rwxrwxr-x 1 lester    54680 Oct  6 09:47 quadlist
-rwxrwxr-x 1 eaton       146 Oct  3 22:51 runcatprog
-rwxrwxr-x 1 eaton        84 Oct  3 22:55 rundigchk
-rwxrwxr-x 1 eaton        78 Oct  3 22:57 rundigix
-rwxrwxr-x 1 eaton        41 Oct  7 07:56 rundl
-rwxrwxr-x 1 eaton       269 Oct  3 22:25 runerrchk
-rwxrwxr-x 1 eaton       543 Oct  3 23:08 runhypoplus
-rwxrwxr-x 1 eaton       112 Oct  6 16:00 runinchk
-rwxrwxr-x 1 eaton        38 Oct  3 23:12 runlerck
-rwxrwxr-x 1 eaton        84 Oct  3 23:33 runmising
-rwxrwxr-x 1 eaton        84 Oct  3 23:10 runmizing
-rwxrwxr-x 1 eaton       110 Oct  3 23:11 runpltfm
-rwxrwxr-x 1 eaton        96 Oct  3 23:15 runqrychk
-rwxrwxr-x 1 eaton        52 Oct  3 23:17 runsrthyp
-rwxrwxr-x 1 eaton     44046 Oct  3 22:43 srthyp

/db/ncal/netlib/fortprog
total 163
-rwxrwxr-x 1 eaton      6291 Oct  3 22:25 catprog.f
-rwxrwxr-x 1 eaton       260 Oct  3 22:25 d80.f
-rwxrwxr-x 1 eaton      3752 Oct  3 22:25 d80p.f
-rw-r--r-- 1 eaton      1745 Oct  7 10:46 digchk.f
-rwxrwxr-x 1 eaton     41301 Oct  3 22:25 digix.f
-rwxrwxr-x 1 eaton       625 Oct  3 22:25 dstaz.f
-rwxrwxr-x 1 eaton        70 Oct  3 22:25 f
drwxrwxr-x 2 eaton       416 Oct 13 09:56 hypofort
-rwxrwxr-x 1 eaton      1738 Oct  3 22:25 lerck.f
-rwxrwxr-x 1 eaton      4617 Oct  3 22:25 mising.f
-rwxrwxr-x 1 eaton      4469 Oct  3 22:25 mizing.f
-rwxrwxr-x 1 eaton      7788 Oct  3 22:25 pltfm.f
-rwxrwxr-x 1 eaton      3031 Oct  3 22:25 qrychk.f
-rwxrwxr-x 1 eaton      3150 Oct  3 22:25 srthyp.f


/db/ncal/netlib/fortprog/hypofort
total 226
-rw-r--r-- 1 eaton      3349 Oct 13 09:54 hypind.f
-rw-r--r-- 1 eaton      7011 Oct 13 09:54 hypo71.f
-rw-r--r-- 1 eaton     11965 Oct 13 09:54 input1.f
-rw-r--r-- 1 eaton      9736 Oct 13 09:54 input2.f
-rw-r--r-- 1 eaton       259 Oct 13 09:54 makefile
-rw-r--r-- 1 eaton     22835 Oct 13 09:54 output.f
-rw-r--r-- 1 eaton     37285 Oct 13 09:54 single.f
-rw-r--r-- 1 eaton      2106 Oct 13 09:54 sort.f
-rw-r--r-- 1 eaton     16396 Oct 13 09:54 sumreg.f
-rw-r--r-- 1 eaton     13957 Oct 13 09:54 trvdrv.f
-rw-r--r-- 1 eaton      7974 Oct 13 09:55 xfmags.f
```

Executable Files

UNIX employs a control language, called the "shell", which permits very flexible manipulation of files, direction of input and output, invocation of user's programs, etc. The set of instructions that would be "input" at a terminal to carry out some procedure can be written onto a file and stored in the directory. This file can then be given "execute" status (by the command, without quotes, "chmod +x filename"). If the name of the file is typed on the terminal and the "return" struck, the set of statements contained in the file are then executed by the shell.

Most of the executable files that we shall use have names beginning with "run", such as "rundigix".

The shell provides for passing parameters through the statement that invokes an executable file into the file that is being invoked. These parameters, which are typed after the name of the executable file to be invoked are represented in the file by $1, $2, etc., meaning the first, second, etc., parameter to be passed.

In the shell language, the symbols <, >, and >> are used to direct input and output to and from files. The command "prog <input >output" invokes the (c-compiled) program "prog" and instructs it to attach a file named "input" to the standard system input (unit 5) and to write its output from the standard system output (unit 6) into a file called "output". The commands cat file1, >input; cat file2 >>input cause a file called "file1" to be written into a file called "input" and then a file called "file2" to be added to the end of "input".

The simple commands like

"rm file1" to remove a file called "file1",

"mv file1 file2" to change the name of a file from "file 1" to "file 2", and

"cp /db/ncal/netlib/file1 file1" to copy a file named "filel" from netlib to a host directory,

have the same effect when they appear in an executable file as when they are given directly through the terminal.

Let us use the command "runprog input output" to invoke an executable file called "runprog" that consists of the single line of text:

```
prog <$1 >$2.
```

The shell would execute the single command "prog <input >output" having passed (or substituted) the first and second parameters (following the command "runprog") to their symbolic counterparts in the text of the executable file.

Some of the programs used in processing Cal Net data read input from more than one input file and write output to more than one output file. These programs contain file "open" statements that associate particular "logical units" with particular file names. On input, the named files (which must exist) are attached to the indicated logical unit; and on output, the output from the indicated logical units are written into the named files. Such files are read and written directly by the program without intervention by an executable file, which might be used to invoke the program and to direct input or output to or from the standard logical units (units 5 and 6).

The executable file can be used, of course, to manipulate the program-generated files (like changing names, deleting, etc.) once they exist.

netlib contains a number of executable files that invoke appropriate station lists, parameter lists, and data lists as well as the programs used for data processing. References to station lists, programs, etc., in these executable files employ their absolute addresses; so these executable files can be used in any of the working directories. The special executable file RUNLN can be copied into a host directory and can then be run to copy (from netlib) the entire set of executable files, etc., needed to process data in the host directory. Another special executable file, RMLN, can be copied from netlib to the host directory to provide a simple means of deleting the entire set of files copied by RUNLN.

RUNLN

```
    cp /db/ncal/netlib/runinchk runinchk

    cp /db/ncal/netlib/rundigix rundigix

    cp /db/ncal/netlib/runerrchk runerrchk

    cp /db/ncal/netlib/rundigchk rundigchk

    cp /db/ncal/netlib/runhypoplus runhypoplus

    cp /db/ncal/netlib/runmizing runmizing

    cp /db/ncal/netlib/runpltfm runpltfm

    cp /db/ncal/netlib/runlerk runlerk

    cp /db/ncal/netlib/runqrychk runqrychk

    cp /db/ncal/netlib/runcatprog runcatprog

    cp /db/ncal/netlib/runsrthyp runsrthyp

    cp /db/ncal/netlib/rundl rundl

    cp /db/ncal/netlib/hypar0 hypar0

    cp /db/ncal/netlib/pltfmparam pltfmparam

    cp /db/ncal/netlib/catparam catparam
```

RMLN

```
rm runinchk
rm rundigix
rm runerrchk
rm rundigchk
rm runhypoplus
rm runmizing
rm runpltfm
rm runlerck
rm runqrychk
rm runcatprog
rm runsrthyp
rm rundl
rm hypar0
rm pltfmparam
rm catparam
```

The executable files in netlib that are copied and erased by RUNLN and RMLN are listed below. Absolute pathnames are used for the programs and for several data lists that are stored in netlib. Note, however, that the three files that are copied into the host directory by RUNLN (i.e., hypar0, pltfmparam, and catparam) are not taken from netlib at execution time by the run... executable files. Thus, these files can be modified in the host directory for flexible control of the processes that they govern.

In the executable files listed below, the argument passed as $1 is the name (date) of the phase-list file being processed, unless otherwise specified. For the daily lists, $1 = mmmdd (e.g., May 28). For the monthly list $1 = MMMYY (e.g., MAY80).

runinchk ($1)

```
     grep -n y '\;' d3.$1
     grep -n y '\∿' d3.$1
     grep -n y '\w' d3.$1
     grep -n y '\,' d3.$1
```

This program checks the file d3.$1 for the occurrence of the following characters: ";", "∿", "w", and ",". The output appears on the terminal and consists of the lines containing the search symbols. The argument -n causes the line number (in d3.$1) to be printed and the argument -y causes a match to be made for either lower- or upper-case characters.

If any of the search symbols occurs and indeed represents an error in d3.$1, the error should be corrected before running program digix.

```
rundigix ($1)
     /db/ncal/netlib/cvc.uc<d3.$1
     /db/ncal/netlib/digix <d3.$1> $1.dig
     mv phcrd $1
```

This executable file first converts the digitizer card file to upper case and then runs program digix with the converted file as input. Digix outputs a diagnostic file, $1.dig, and a phase-card list file, phcrd, which is renamed $1 by rundigix.

```
runerrchk ($1)

     grep -n -y warning $1.dig

     grep -n -y accurate $1.dig

     grep -n -y slash $1.dig

     grep -n -y expected $1.dig

     grep -n -y assume $1.dig

     grep -n -y illegal $1.dig

     grep -n -y letters $1.dig

     grep -n -y error $1.dig

     grep -n -y delete $1.dig

     grep -n -y identifier $1.dig
```

This program checks the digix diagnostic file for the occurrence of warnings that indicate possible inaccuracies or errors in the phase-card list (phcrd or $1) produced by digix. The output appears on the computer terminal. These possible errors should be checked (and corrected) before further processing of the data is undertaken.

```
rundigchk ($1)

     /db/ncal/netlib/digchk <$1

     /db/ncal/netlib/d80p <data.c> f.$1

     rm data.c

     rm data.err
```

This executable file first calls digchk to verify that phase cards in the phase-card list produced by digix contain only digits or a decimal in columns 10 through 24 (read by hypo711 under I and F formats). It then calls d80p to replace embedded blanks in the integer time field (columns 10 through 19) of phase cards with zero's and to sort the phase cards in order of increasing time. It then removes two files generated by digchk. digchk flags errors with a message to the terminal. If any are found, digchk should be run separately (by the command digchk <$1) to regenerate the data.err file, which indicates which lines in $1 contain errors.

The input to rundigchk is the daily phase-card list ($1) produced by digix.

The output of rundigchk is the sorted phase-card list (f.$1) produced by d80p.

```
runhypoplus ($1)
     /db/ncal/netlib/cvc.uc<$1
     echo 'digchk running'
     db/ncal/netlib/digchk <$1
     /db/ncal/netlib/d80p <data.c> f.$1
     rm data.c
     rm data.err
     echo 'digchk completed'
     cat hypar0 >hypo.input
     cat f.$1 >>hypo.input
     cp /db/ncal/netlib/nocast0 calstn
     echo 'hypo71l running'
     /db/ncal/netlib/hypo71l
     mv hypo.print $1.prt
     mv hypo.pch $1.pch
     mv hypo.smp $1.smp
     rm hypo.input
     echo 'hypo71l completed'
     echo 'mizing running'
     /db/ncal/netlib/mizing <$1.prt >$1/prtm
     rm calstn
     echo 'mizing completed'
     cat $1.smp >>$1.prtm
     echo 'hypoplus completed'
```

This executable file carries processing of earthquake data through several steps and employs several programs. First, it invokes digchk and d80p to check for format errors, replace embedded blanks in the integer time field with zeros, and sort the phase card list according to increasing time. It then sets up the input file, hypo.input, and station list file, calstn, that are required by hypo71l. Next, it invokes hypo71l. Then it renames the three hypo71l output files to conform with the file name system outlined earlier. Next it invokes mizing to examine the hypo71l print output file ($1.prt) and to identify needed missing stations, whose names are included in an augmented version of the hypo71l print output file ($1.prtm). Finally, it appends the short hypocenter summary file $1.smp to $1.prtm. In addition, it deletes a number of temporary, no-longer-needed files to avoid clutter in the directory and it sends messages to the terminal to indicate the stage of processing currently underway.

The input files are:

$1 the corrected earthquake phase-card list, produced by digix

hyparO the parameter list containing the reset tests, model, and control card required by hypo71l (station delay mode)

nocast0 the station list (station delay format);

and the output files are:

f.$1	the ordered phase list generated by d80p
$1.smp	the hypo711 hypo.smp summary card output
$1.pch	the hypo711 punch-card list output
$l.prt	the hypo711 printer output
$1.prtm	the hypo711 printer output augmented by lists of missing stations and the summary card list.

runmizing ($l)

```
     cp /db/ncal/netlib/nocast0 calstn
     /db/ncal/netlib/mizing <$1.prt >$1.prtm
     rm calstn
```

This executable file invokes program mizing to examine the hypo711 print output file ($1.prt) to determine which missing stations should be read, and it adds these names in the appropriate places to $1.prt and writes out the new list as $1.prtm. It also sets up the station list file required by mizing before invoking the program and removes it after execution is completed.

runpltfm ($1)

```
     cat pltfmparam >pltfm.input
     cat $1.pch >>pltfm.input
     /db/ncal/netlib/pltfm <pltfm.input >$1.fm
     rm pltfm.input
```

This executable file sets up the input file required by pltfm (plot first motion), invokes the program, and then removes the input file. Inputs to runpltfm are 1) the list of parameters (pltfmparam) that govern the action of the program and 2) the hypo71l "punch card" output file ($1.pch) for one or more quakes. The output ($1.fm) is the $1.pch file, augmented by a list of plotted symbols, plus a printer plot of first motions on the lower half of the focal sphere.

runlerck ($1)

```
     /db/ncal/netlib/lerck <$1.smp >$1.ler
```

This executable file invokes program lerck (large error check) to examine the monthly summary card list ($1.smp - e.g., MAY80.smp) for large errors. Information on errors is output to file $1.ler (e.g., MAY80.ler).

runqrychk ($1)

```
     cp /db/ncal/netlib/qrylist qrylst
     /db/ncal/netlib/qrychk <$1.smp >$1.qrt
     mv qrycds $1.qch
     rm qrylst
```

This executable file invokes program qrychk to screen the monthly hypocenter summary list for quarry shots. It sets up the file qrylst (list of quarries and their coordinates) before invoking the program and deletes it after execution. The input to runqrychk is the monthly summary card list ($1.smp - e.g., MAY80.smp). The output consists of two files: $1.qrt, which identifies quarries and points out misidentifications, and $1.qch, a list of corrected "cards" for substitution in $1.smp to correct errors in quarry identification.

```
runcatprog ($1)

     cp /db/ncal/netlib/quadlist quadlst

     cat catparam >cat.input

     cat $1.smp >>cat.input

     /db/ncal/netlib/catprog <cat.input >$1.cat

     rm cat.input

     rm quadlst
```

This executable file invokes program catprog to prepare a standard format catalog from monthly (or yearly) lists of earthquake summaries in hypo711 summary card format. It first prepares the input files required by catprog:

1) quadlst, a list of map quadrangles is copied from file quadlist in netlib

2) the parameter file catparam and the summary card file, $1.smp (e.g., MAY80.smp), are combined into cat.input.

Next it invokes catprog, which ouputs the catalog on file $1.cat (e.g., MAY80.cat). Finally, it removes the files cat.input and quadlst.

runsrthyp ($1)

```
/db/ncal/netlib/srthyp <$1  >sumaa
rm $1
mv sumaa $1
```

This executable file invokes srthyp to sort a set of hypo711 hypocenter summary cards in order of increasing time. The sorted output file has the same name as the original input file. Normally, this program is used to sort a set of concatenated daily summary-card files, suma.

rundl ($1)

```
rm $1.dig
rm $1.pch
rm $1.prt
rm $1.prtm
```

This executable file deletes the set of output files that are generated by runhypoplus. It is normally run after a daily phase card file has been corrected for errors and before the corrected phase card list is rerun by runhypoplus.

PART II: PROCESSING OF DATA

The procedures described in this write-up carry processing of Cal Net data from Develocorder film "readings" in the form of card decks produced by the table-top digitizer to finished monthly catalogs of earthquakes and monthly archive tapes of phase card lists and hypocenter summary card lists. The work is divided into two stages: production of corrected phase card lists for each day of the month, and production of final monthly catalogs of hypocenters and monthly archive tapes of phase card lists and hypocenter lists. Most of the work is associated with the first stage, primarily with checking and correcting the daily phase card lists. Stage 1 processing, itself, is divided into two parts: first pass processing of daily phase card lists from the digitizer cards to hypocenter locations (carried out in the appropriate monthly directories), and error detection, seismogram rereading, reprocessing, etc., of daily phase card lists to produce the final corrected phase card lists (carried out in the "private" directories of the analysists).

The first pass processing is accomplished in five steps, each carried out in the computer by means of its own executable file run... command. These steps are:

1. read the digitizer card deck into the appropriate monthly subdirectory,
2. check the digitizer card deck file for obvious, frequently occurring errors,
3. convert the digitizer card deck file into a phase-card list file,

4. check the output of program digix for warning messages that indicate possible errors, omissions, etc., in the phase-card list file, and
5. process the phase card list to obtain hypocenter solutions.

These five steps are summarized in the following table, which indicates the command that executes the step, the programs that carry it out, and the input and output files that are employed with it.

Outline of Commands to Process One Day's Data

tep	(command)	programs	input and output files	
1	(cat/dev/crr d3.mmmdda)*		Read card deck from digitizer into appropriate monthly subdirectory	
2	(runinchk mmmdda)	grep	d3.mmmdda	input
			list of probable punch errors, output to terminal	output
			d3.mmmdda	input
3	(rundigix mmmdda)	cvc.uc	mmmdda.dig diagnostic file	output
		digix	mmmdda phase-card file	"
			mmmdda.dig	input
4	(runerrchk mmmdda)	grep	list of lines containing warning messages, output to terminal	output
		digchk	mmmddα corrected phase card list	input
		d80p	mmmddα.smp hypo711 summary card list	
5	(runhypoplus mmmddα)	hypo711	mmmddα.pch hypo711 punch card output	
		mizing	mmmddα.prt hypo711 printer output	
			mmmddα.prtm hypo711 printer output plus lists of missing stations and summary card list	

[*] mmmdd = film day off (e.g., may23)

To complete the first pass analysis the file mmmdd.prtm is examined for evidence of gross errors: "mixed" events, large timing errors, misidentified stations, etc. The phase card list is then corrected to resolve such errors, and it is then reprocessed to obtain new hypocenter solutions.

Reprocessing is accomplished by the commands:

(rundl mmmddα)	remove mmmddα.* files
(rundl mmmddβ)	process updated phase list.

The second part of Stage 1 processing requires skill in detecting phase card errors by the consequences they produce in the station residuals of the hypocenter solution. It also requires rereading (from film) suspected stations, requesting and analyzing Siemens playbacks for missing stations, etc. Corrections or additions to the phase card lists required by rereading some stations or adding others are accomplished in the private directory of the analyst. The reprocessing commands listed above are then executed to remove unnecessary old files and to obtain a new set of hypocenter solutions. These new solutions are then examined to determine whether they meet acceptable standards. When this process is complete, the final corrected phase card list (mmmddg, say) is copied back into the monthly directory.

Short Description of Programs Used to Process One Day's Data

Program digix

This program is a UNIX version of the program digi3 (MULTICS), written by Peter Ward, John Lahr, and Bill Ellsworth for the CDC1700, originally. It converts the cards output by the digizer table into standard HYPO71 phase-card format.

The primary conversion problems stemmed from the program's use of "R" format for alphameric characters. In this format, the high order byte of all alpha characters is NUL, not BLANK as in A format. Alpha characters encoded in R format can be read and manipulated as integers. When printed under A format, the alpha characters appear; but when printed under I format, the integer "value" of the characters appear. By utilizing R format, digi3 is able to store both alpha and integer characters in the same integer array.

In digi3, the input stream of characters is read in R format. In digix, the input stream of characters is read in A format and then converted to R. Both alpha (R representation) and numeric characters are treated as integers thereafter until the output "WRITE" statements. The write FORMAT specifies the integers in I format and the alpha characters in A format.

Two output files are produced by digix: the first is the phase-card file (phcrd) and the second is the file (on unit 6, standard output) that traces the work of the program and presents diagnostics to flag problems.

digix is invoked by the executable file:

```
rundigix ($1)


     cvc.uc >d3.$1

     digix <d3.$1 >$1.dig

     mv phcrd $1
```

Program digchk

The present configuration and performance of the table-top digitizer and its card punch and of the UNIX card reader lead to a significant number of errors in the phase card list produced by digix. If any character other than integer or blank occurs in the integer fields of the phase card or other than integer, blank, or decimal occurs in the fixed point real fields of the phase card, the program fails on execution when these fields are read under the appropriate I or F formats. Program digchk examines each character in columns belonging to I or F fields of the phase cards to verify that it is an integer, blank, or decimal (for F fields). If an inappropriate character is found, it is replaced with a blank.

digchk uses the standard system input file.

The "corrected" file (data.c) and a file that identifies the errors and where they occur (data.err) are output by the program. digchk also returns a message to the terminal if errors are detected.

digchk is invoked, along with d80p, by the executable file rundigchk.

Program d80p

This program operates on the phase-list file being prepared for hypo711. It modifies the phase cards, but not other cards, by changing all blanks in columns 10-19 (date:year-min) to zero, and by inserting a zero in column 80. hypo711 reads the date (year-hr) in I8 format; but the FORTRAN77 compiler, with the default option for treatment of imbedded blanks in integer fields, eliminates the blanks by shifting all nonblank characters to the right to close up the blanks under this reading format. It appears that the file "open" option that should cause blanks in integer fields to be read as zeros is defective in the old FORTRAN routines that must be used to compile hypo71.

The program also sorts the stations for each event on the phase list in order of increasing arrival time. Ordering stations by arrival time is necessary because hypo711 uses no more than the first 100 arrivals on the phase list for each event; the rest are simply skipped.

d80p outputs a formatted, sorted phase list. It uses the standard system input (unit 5) and output (unit 6) files.

d80p is invoked, along with digchk, by the executable file:

```
rundigchk ($1)
          digchk <$1
          d80p <data.c >f.$1
          rm data.err
          rm data.c
```

Program d80

This program inserts a "0" in the 80th column of each line of a file. It is used to fill out the lines of the station list invoked by hypo711. The station list file is read as a formatted direct access file (by hypo711) with record length 81 (80 characters plus end file). If the original lines do not contain the full count of 80 characters plus end file, the read/write formats in hypo711 don't match the "station cards" and the program fails with a format error when the direct access file is read during execution

d80 uses the standard system input and output files and is invoked directly.

```
d80 <oldlist >newlist
```

Program hypo71l

This program is a variant of the version of HYPO71 that was adapted to run on UNIX by Sam Stewart. It consists of the following:

hypo71	main program
hypind	parameter input
input1	input of model and station list, etc.
input2	input of phase cards, etc.
output	output of "print", "punch", and "summary punch" files
single	calculates solution for one event
swmreg	stepwise multiple regression
trvdrv	traveltime and derivatives
xfmags	computes amplitude and duration magnitudes
sort	sorts output files.

In this shortened version of HYPO71 the following subroutines of the original program were deleted:

AZWTOS	azimuth weighting by quadrants
FMPLOT	first motion plot
MISING	identify needed missing stations
SUMOUT	output summary of time and magnitude residuals.

Subroutines MISING and FMPLOT have been modified to run as independent programs, mizing or mising and fmplt, that use the hypo71l outputs hypo.print and hypo.punch as input. It is convenient to run mizing immediately after hypo71l by use of an appropriate executable file.

Besides its shorter length, hypo711 differs from HYPO71 principally by its treatment of the "station"list: in hypo711 this file is read as a direct-access formatted disc file; and a station name (match list) array that is indexed to the disc file "records" is used to identify the station data required for a particular earthquake.

Because of limited core space, the number of stations used for a single earthquake is limited to 100. Since the arrivals are ordered according to increasing time (by d80p) on the phase list, the 100 stations retained are the earliest ones.

The main program and subroutines were compiled with the aid of program makefile:

makefile

```
OBJECTS     hypo71.o input1.o input2.o output.o
            single.o sort.o swmreg.o trvdrv.o
            xfmags.o hypind.o

FFLAGS      -i -I2 -0

LFLAGS      -loI77 -loF77 -lm

hypo71:     $(OBJECTS)
            cc -i $(OBJECTS) $(LFLAGS) -o hypo711
            size hypo711
            @echo "hypo711 is ready"
```

The finished product is named:

hypo711.

Executable file runhypol0 (and runhypol1)

HYPO71 was originally written to run in batch mode and to take its input from cards: reset parameters, station list, model, control card, and phase lists. hypo71l requires two input files with prescribed names: "calstn" for the station list and "hypo.input" for the rest of the input. It is convenient, also, to separate the phase lists from the other lists in hypo.input and to establish naming conventions and procedures so that the names of the hypo71l output files identify the content and character of the data they contain. These objectives are met with the executable file runhypol0.

The station list is maintained on a file named "nocast0", which has been processed by program d80 to insert a 0 in column 80 of each "card". The parameter list, which consists of the reset parameters, the station-card format code, the model and the control card, is maintained on a file named "hypar0". The phase-card list, which has been processed by digix, digchk, and d80p, is named f.$1. runhypol0 appends f.$1 to hypar0 and names the combined file hypo.input. It then copies nocast0 into calstn. Next, it invokes hypo71l to operate on hypo.input and calstn. Next it renames the three output files: hypo.print becomes $1.prt; hypo.punch becomes $1.pch; and hypo.smp becomes $1.smp. Finally, it removes files calstn and hypo.input.

The foregoing setup runs hypo71l in the "station delay model" mode, for which the station-card format code is 0. To accommodate the "variable layer model" mode, for which the station-card format code is 1, analogous files runhypol1, hypar1, and nocast1 are employed.

The executable file runhypol0 (and runhypol1) is used to run hypo711 by itself.

runhypol0 ($1)

```
cat hypar0 >hypo.input
cat $1 >>hypo.input
cp nocast0 calstn
echo "Now executing hypo711"
hypo711
echo "Finished with hypo711"
mv hypo.print $1.prt
mv hypo.punch $1.pch
mv hypo.smp $1.smp
rm hypo.input
rm calstn
```

Program mizing

Program mizing checks each station on the station list against the set of stations that recorded an event and determines whether stations that were not read should be read on the basis of:

1) magnitude of the earthquake and epicentral distance of the station

2) reduction in "gap" that would result from the inclusion of the station.

Inputs to the program are the station list and the hypo71l hypo.print output file.

The output of the program is the hypo.print file augmented by the list of additional stations that should be read for each earthquake and the tape channel and Develocorder assignments of the missing stations.

Parameters that determine the operation of the program are contained in a data statement. They are:

1) tdz and tde, for the calculation of the test distance; tdel=tdz*XMAG**tde

2) gptst, the "gap reduction" required for inclusion of a station in the "missing" list.

The executable file runmizing is used to run mizing by itself.

runmizing ($1)

```
cp nocast0 calstn
mizing <$1.prt >$1.prtm
rm calstn
```

where the input files are

nocast0, the station list, and
$1.prt, the hypo.print output of hypo71l;

and the output file is

$1.prtm, the $1.prt file augmented by the list of missing stations.

Program <u>mising</u>

Program mising is the same as mizing with one exception: mising uses the hypo.punch file as its input instead of the hypo.print file; so its output is much abbreviated and consists only of the hypo.punch file augmented with the list of missing stations and their tape track and Develocorder assignments.

The executable file runmising is used to run mising by itself.

runmising ($1)

```
        cp nocast0 calstn
        mising <$1.pch >$1.pchm
        rm calstn
```

where the input files are

nocast0, the station list, and

$1.pch, the hypo.punch output of hypo711;

and the output file is

$1.pchm, the $1.pch file augmented by the list of missing stations.

Program pltfm

Program pltfm is modified substantially from the original FMPLOT to permit control of the plotted symbols on the basis of a number of test parameters, including:

1) type of phase onset, e or i
2) phase weight,
3) phase residual, and
4) epicentral distance to the recording station.

pltfm also accepts a list of "reversed" stations, for which the signs of onsets are corrected (reversed again) before plotting.

As input, this program requires:

1) a list of test parameters
2) a list of reversed stations
3) hypo711 hypo.punch file for one or more earthquakes.

As output, this program produces:

1) hypo.punch file augmented by the symbols actually plotted
2) equal area projection (lower hemisphere) of the first-motion plot on the focal sphere.

The executable file runpltfm is used to run pltfm.

```
runpltfm ($1)

        cat pltfmparam >pltfm.input
        cat $1.pch >>pltfm.input
        pltfm <pltfm.input >$1.fm
        rm pltfm.input
```

where the input files are

pltfmparam, the parameter file governing the action of pltfm and the list of reversed stations, and $1.pch, the hypo.punch output of hypo71l;

and the output file is

$1.fm, the augmented $1.pch file with appended first-motion plot.

Completion of Processing of One Month's Data

When day-by-day processing of a month's data has been completed and the monthly subdirectory contains the corrected phase lists for each day of the month, several more steps are required to complete work on the month's data. These steps are:

1) combine the daily phase lists into a single list for the entire month (MMMYY),

2) run the location program (under runhypoplus) to get a complete monthly printout of the solutions (MMMYY.prtm) and a complete monthly hypocenter summary-card list (MMMYY.smp),

3) review the monthly hypocenter list to detect large errors (and correct the offending quakes) [program lerck],

4) review the monthly hypocenter list to identify and label probable quarry blasks [program qrychk],

5) prepare a monthly catalog of earthquakes [program catprog],

6) prepare a tar tape of the monthly phase list and the monthly hypocenter list, and

7) prepare a monthly plot of earthquake epicenters.

The set of programs used for the foregoing tasks in MULTICS have been adapted for use in UNIX and placed in directory netlib. Several additional files required by these programs are also stored in netlib. The programs are invoked (in the monthly directory) by a set of executable files in netlib that are copied into the monthly subdirectories by the special executable file RUNLN. RUNLN must be copied from netlib to the monthly subdirectory and then run in the subdirectory to copy the other executable files.

Programs, Auxiliary Files, and Executable Files Used to Complete Work on Data for an Entire Month

Program lerck (large error check)

This program checks the monthly hypocenter list for possible errors revealed by:

1) an event out of chronological order

2) unusual or extreme values of latitude, longitude, depth, magnitude, number of stations, gap, minimum distance, error in epicenter or depth, rms of solution, quarry flag.

The program reproduces the hypocenter "cards" augmented by a set of codes indicating suspicious conditions.

lerck is invoked by the executable file:

```
runlerck ($1)

    (lerck <$1.smp >$1.ler)
```

Program qrychk (quarry check)

This program compares the location of each event on the monthly hypocenter summary list with the locations of a number of known quarries. The identification and location of the quarries is stored in file qrylist in netlib. If an event falls within the range of latitudes and longitudes corresponding to a quarry and occurs during working hours, the event is flagged as a possible quarry blast and a duplicate hypocenter "card" with the appropriate designation is produced. If an event that is already flagged as a probable quarry does not meet the above criteria, it is flagged as a possible misidentified quarry blast, and a duplicate hypocenter "card" with the appropriate designation is produced.

qrychk is invoked by the executable file:

```
runqrychk ($1)

     qrychk <$1.smp >$1.qrt
     mv qrycds $1.qch
```

Program catprog

This program prepares a standard hypocenter catalog from the monthly hypocenter summary list. Two auxiliary files are required to run this program: catparam, a set of parameters that govern the action of the program, and quadlist, a list of map quadrangles with names and coordinates. The files quadlst and catparam are stored in directory netlib.

catprog is invoked by the executable file:

```
runcatprog ($1)

    cat catparam >cat.input
    cat $1.smp >>cat.input
    catprog <cat.input >$1.cat
    rm cat.input
```

Program sorthyp

This program is used to sort events in a hypocenter list according to time of occurrence. Processing of data for individual days in a given month is not accomplished in chronological order. It is therefore necessary to sort the hypocenter summary file, suma, when it is augmented.

sorthyp is invoked by the executable file

```
runsorthyp ($1)

     sorthyp <$1  >sumaa
     rm $1
     mv sumaa $1
```

Reading and Writing Tar Tapes on UNIX and MULTICS

Procedures for reading and writing tar tapes on UNIX are described under UNIX commands: TAR(UA1). Procedures for reading and writing tar tapes on MULTICS have been set up by Peter Ward and described by him in "The MULTICS UNIX Tape Connection". Copies of the descriptions of both of these procedures are appended.

To write a tape in UNIX that will contain the monthly phase list MMMYY:

1) mount the tape, with a write ring, on tape drive 0 (instructions inside tape-drive door)

2) log into UNIX and get into the monthly subdirectory, mmmyy, that contains the file MMMYY,

3) write (without quotes)

   a) "all 0" (i.e., allocate tape drive 0)

   b) "tar c MMMYY" (i.e., create a tape containing MMMYY). If the file is to be added to a tape already containing data (e.g., previous months' phase data), the b) command above should be:

   b') "tar r MMMYY"

4) After the tape is written and removed from the tape drive, deallocate the tape drive so that it is available to other users:

   "deall 0".

The same procedure should be used to write a tar tape containing the monthly hypocenter summary list, MMMYY.smp.

Pending completion of programs in UNIX that will permit plotting of epicenter maps on the Calcomp plotter, monthly hypocenter summary-card lists are transferred to MULTICS by means of a tar tape or by means of the T.I. cassette terminal. Plots are then made on MULTICS.

NAME

tar - tape archiver

SYNOPSIS

tar [key] [name ...]

DESCRIPTION

tar saves and restores files on magtape. Its actions are controlled by *key* *key* is a string of characters containing at most one function letter and possibly one or more function modifiers. Other arguments to the command are file or directory names specifying which files are to be dumped or restored. In all cases, appearance of a directory name refers to the files and (recursively) subdirectories of that directory.

The function portion of the key is specified by one of the following letters:

r The named files are written on the end of the tape. The c function implies this.

x The named files are extracted from the tape. If the named file matches a directory whose contents had been written onto the tape, this directory is (recursively) extracted. The owner, modification time, and mode are restored (if possible). If no file argument is given, the entire content of the tape is extracted. Note that if multiple entries specifying the same file are on the tape, the last one overwrites all earlier.

t The names of the specified files are listed each time they occur on the tape. If no file argument is given, all of the names on the tape are listed.

u The named files are added to the tape if either they are not already there or have been modified since last put on the tape.

c Create a new tape; writing begins on the beginning of the tape instead of after the last file. This command implies **r.**

The following characters may be used in addition to the letter which selects the function desired.

0,...,7 This modifier selects the drive on which the tape is mounted. The default is **1.** (0 is physical drive 0, 800 bpi, 1 is drive 0, 1600 bpi, 2 is drive 1, 800 bpi, 3 is drive 2, 1600 bpi, etc.).

v Normally *tar* does its work silently. The **v** (verbose) option causes it to type the name of each file it treats preceded by the function letter. With the **t** function, **v** gives more information about the tape entries than just the name.

w causes *tar* to print the action to be taken followed by file name, then wait for user confirmation. If a word beginning with 'y' is given, the action is performed. Any other input means don't do it.

f causes *tar* to use the next argument as the name of the archive instead of /dev/mt?. If the name of the file is '- ', tar writes to standard output or reads from standard input, whichever is appropriate. Thus, *tar* can be used as the head or tail of a filter chain. *tar* can also be used to move hierarchies with the command

```
cd fromdir; tar cf - . | (cd todir; tar xf -)
```

b causes *tar* to use the next argument as the blocking factor for tape records. The default is 1, the maximum is 20. This option should only be used with raw magnetic tape archives (See **f** above). The block size is determined automatically when reading tapes (key letters 'x' and 't').

l tells *tar* to complain if it cannot resolve all of the links to the files dumped. If this is not specified, no error messages are printed.

m tells *tar* to not restore the modification times. The mod time will be the time of extraction.

FILES

/dev/mt?
/tmp/tar*

DIAGNOSTICS

Complaints about bad key characters and tape read/write errors.
Complaints if enough memory is not available to hold the link tables.

SEE ALSO

tp(UA1),all(UA1)

BUGS

There is no way to ask for the *n*-th occurrence of a file.
Tape errors are handled ungracefully.
The **u** option can be slow.
The **b** option should not be used with archives that are going to be updated. The current magtape driver cannot backspace raw magtape. If the archive is on a disk file the **b** option should not be used at all, as updating an archive stored in this manner can destroy it.
The current limit on file name length is 100 characters.
It has not been interfaced to *all*(UA1) yet, thus you must manually allocate a tape.

The MULTICS UNIX Tape Connection

Peter Ward

Programs now exist on Multics to read, write, or list standar UNIX TAR tapes. Thus ascii files on either machine can be readily transfered to the other.

HOW TO USE ON MULTICS

1. Define three abbreviations by typing the following (without the quotations):

```
".ab writeu   ec >udd>WARD>PWard>UNIX>writeunix.ec"
".ab readu    ec >udd>WARD>PWard>UNIX>readunix.ec"
".ab printu   ec >udd>WARD>PWard>UNIX>printunix.ec"
```

2. Deliver a blank or (1600 bpi) TAR tape to the multics computer room. The tape must have your name on it and any unique character or number label written in large letters on the front of th tape and on the side of the hang strap or cartridge.

3. To write a TAR tape:

   a. Move all files you wish to put on the tape to the sar directory.

   b. Make a file that lists the names of all files you wis to write on the tape. For example, if you want to transfer all fortran files in this directory type:

```
"fo list ; ls -name -pri *.fortran ; co"
"ted,-pn list"
"1,3d"
"w"
"q"
```

   c. type: "writeu label list" where label is the tape label of 1 to 6 characters or numbers and list is the segment name containing a list of all segments to be written. Of course list can be any name.

   d. Multics will type back:

```
"assign_resource tape_drive"
"Device tape_0x assigned"         (where x is some
                                    integer)
">udd>WARD>PWard>UNIX>writeunix_tape label list"
"Tape label, den=1600, blk=2800 will be mounted with
    write ring."
```

   e. Typically after a minute or three multics will then type:

```
"Tape label, den 1600, blk=2800 mounted on drive x
    with a write ring."
"Files being written:  Starts at block:"
```

When all the files are listed, you should get the Multics command level prompt. Then you can logout and go get your tape.

f. Problems: After typing "assign_resource tape_drive". Multics may say "assign_resource: No appropriate resource available. Unable at this time to assign tape_drive device." So you should try again later. If you want to keep trying each minute, go into Geolab and type the following:

```
"op unix (exec 'writeu label list')"
"20 do (unix 60 delay)"
```

Another problem may be that after Multics types "Tape label, den=1600, blk=2800 will be mounted with no write ring." nothing may happen forever. There are two possibilities: Either the operator is not available because he/she stepped out for a few minutes (often a problem after 5 pm) or the operator loaded the tape but multics did not get the message. There is a bug in the tape_nonstandard software that is used for writeunix where for some tape drives under some special conditions the software hangs at this point. Mike Auerbach has not been able to locate the bug. If this does happen to you (and it is rare) push the break key and type "ur all" and then try again.

g. File names on multics ending in ".fortran" are loaded on the tape ending in ".f" as used in unix.

4. To read a tar tape:

a. type "readu label"
b. all files on the tape will be dumped into the working directory
c. Files ending in ".f" will be entered as ending in ".fortran".
d. A list of files transferred will be given.

5. To print the contents of a tar tape:

a. type "printu label"
b. A segment called "label.output" will be created, the tape read into it, dprinted and deleted. If the tape is longer than one segment, more segments called "label.output.output" etc will be created along with a message that you must dprint and delete these segments yourself.

6. The source for all these ecs and programs is in the directory >udd>WARD>PWard>UNIX and you may get a listing by typing "ec >udd >WARD>PWard>UNIX>tape.ec". To use tape.ec you need an abbreviation LO. If you do not have one type ".ab LO ec >udd>WARD>PWard>Loff.ec". Tape.ec will delete any file called "output" in the working directory.

7. A similar facility exists for reading and writing tar tapes on the ecclipse. See Jeff Hobson.

TO USE ON UNIX

1. Use the tar command. For example, to read the tape into the working directory, load it on physical drive 0 and type "tar X"

PART III: FORTRAN listing of the principal programs used in processing Cal Net data

digix

```
c      digi3

c-----this program converts data on cards punched by datagrid digitiser
       implicit integer*2 (i-n)
       external time, itsum
       common /blkda / ichar(50),ista
       common       ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18)
      1,xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact
      1,terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82)
      1,at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
       dimension  kard(19,15),lcard(36),idate(6),istat(76),
      1stacor(19),xkard(19,7)
       dimension xfill(7),yfill(7)
       dimension index1(100),index2(100)
       dimension fmt(2)
       character*40 fmt
       data tcor/0.0/,ntm,ntg/0,0/
       data istat(73),istat(74),istat(75),istat(76)/42,42,42,42/
       data xfill(1),xfill(2),xfill(3),xfill(4),xfill(5),xfill(6),xfill(7
      1)/-600.,-600.,-600.,-600.,-600.,-600.,-600./
       data yfill(1),yfill(2),yfill(3),yfill(4),yfill(5),yfill(6),yfill(7
      1)/-800.,-800.,-800.,-800.,-800.,-800.,-800./
       save xfill,yfill,istat,tcor,ntm,ntg
       open(7,file='phcrd',form='formatted', status='new')
       open(8,file='dscfl',form='unformatted',status='new',
      1access='direct',recl=66)
       open(unit=5,access='sequential',form='formatted',blank='zero')
       rewind 5
c
c   ... some definitions:
c
c      ichar index      ichar value      corresponding symbol
c        1-10             48-57                0-9
c         11               43                   +
c         12               45                   -
c         13               47                   /
c         14               84                   T
c         15               73                   I
c         16               69                   E
c         17               68                   D
c         18               85                   U
c         19               83                   S
c         20               77                   M
c         21               72                   H
c         22               80                   P
c         23               78                   N
c         24               70                   F
c         25               66                   B
c         26               65                   A
c         27               67                   C
c         28               89                   Y
c         29               58                   :
c         30               59                   ;
c         31               82                   R
c         32               71                   G
c         33               74                   J
c         34               75                   K
c         35               76                   L
c         36               90                   Z
c         37               81                   Q
c         38               86                   V
c         39               87                   W
c         40               36                   $
```

```
c         41               32              (BLANK)
c         42               88                 X
c         43               46                 .
c         44               44                 ,
c         45               79                 O
c         46               63                 ?
c         47               42                 *
c
c ...   values to be output on phase cards:
c
c       istat(i), i=1,4  =  station name
c
c       kard(i,1) = p descriptor
c       kard(i,2) = first motion descriptor
c       kard(i,3) = p-wt
c       kard(i,4) = refracted layer index
c       kard(i,5) = yr
c       kard(i,6) = mo
c       kard(i,7) = da
c       kard(i,8) = hr
c       kard(i,9) = min
c       kard(i,10) = s descriptor
c       kard(i,11) = s first motion descriptor
c       kard(i,12) = s-wt
c       kard(i,13-15) = rmk
c
c       xkard(i,1) = p-sec
c       xkard(i,2) = s-sec
c       xkard(i,5) = cal amp
c       xkard(i,6) = time corr
c       xkard(i,7) = f-p time
c ...
c       idate(1-6) = window time: yr,mo,da,hr,min,sec
c....
      ichar( 1)=48
      ichar( 2)=49
      ichar( 3)=50
      ichar( 4)=51
      ichar( 5)=52
      ichar( 6)=53
      ichar( 7)=54
      ichar( 8)=55
      ichar( 9)=56
      ichar(10)=57
      ichar(11)=43
      ichar(12)=45
      ichar(13)=47
      ichar(14)=84
      ichar(15)=73
      ichar(16)=69
      ichar(17)=68
      ichar(18)=85
      ichar(19)=83
      ichar(20)=77
      ichar(21)=72
      ichar(22)=80
      ichar(23)=78
      ichar(24)=70
      ichar(25)=66
      ichar(26)=65
      ichar(27)=67
      ichar(28)=89
      ichar(29)=58
      ichar(30)=59
      ichar(31)=82
      ichar(32)=71
      ichar(33)=74
      ichar(34)=75
      ichar(35)=76
      ichar(36)=90
      ichar(37)=81
      ichar(38)=86
      ichar(39)=87
      ichar(40)=36
      ichar(41)=32
      ichar(42)=88
      ichar(43)=46
      ichar(44)=44
      ichar(45)=79
      ichar(46)=63
      ichar(47)=42
      m2=2
      m3=3
      m4=4
      m6=6
      m8=8
      m10=10
      m11=11
      m12=12
      m24=24
      m48=48
      m49=49
      do 5 i=1,4
    5 istit(i)=ichar(41)
      do 11 i=1,82
      stcor(i)=0.0
   11 istcor(i)=0
      jstcor=0
c*********************************************************************
    9 kstat=0
      ipunch=0
      tcor=0.0
      kfilm=0
      ind=0
      jstat=3
      ista=ichar(19)+ichar(14)+ichar(26)
c-----zero the array of station time corrections.-----------------------
      do 10 i=1,19
   10 stacor(i)=0.0
c-----read title card.-------------------------------------------------
      read(5,2) lcard
    2 format(8x,36a2)
c-----read the number of second marks in grid and factor by which to
c     multiply p-p amplitude measured in inches.-----------------------
c-----projector magnification is about 30.3267.-------------------------
c-----put 1 in,column 80 to print format 9080 near statement 80 for
c     debugging and tracing input of characters.-----------------------
      read(5,1) xmin,ampfac,iunit,ibug
    1 format(8x,f2.0,f10.5,8x,i2,49x,i1)
      write(6,8)(lcard(i),i=1,36),xmin,ampfac
    8 format(1h1,36a2,/,43h distance between grid marks in seconds is ,
     1f3.0,44h factor multiplied by amplitude in inches is ,f10.5)
      if (iunit.eq.0) iunit=7
      write(6,30) iunit
   30 format(36h phase cards will be output on unit ,i2)
      ib=1
      il=0
c-----read stream of characters.---------------------------------------
```

```
      kchr=1
   50 call charac
      go to ( 54, 80, 52), jstat
   52 write(6,3)
    3 format(40h station list not read in when expected.,
     124h looking for a new list.    )
      go to 1059
c-----update station list.--------------------------------------------
   54 call newlst(kfilm,index1,ind,lcard,stacor,istat)
      if (ierror.eq.1) goto 1058
c-----zero output arrays.----------------------------------------------
   70 do 74 i=1,19
      do 71 j=1,4
   71 kard(i,j)=ichar(41)
      kard(i,3)=ichar(5)
      do 72 j=5,9
   72 kard(i,j)=0
      do 73 j=10,15
   73 kard(i,j)=ichar(41)
      xkard(i,1)=9999.
      xkard(i,2)=9999.
      do 774 j=3,7
  774 xkard(i,j)=0.0
   74 continue
      izfix=ichar(41)
      inos=ichar( 2)
      isav(1)=ichar(15)
      isav(2)=ichar(22)
      isav(3)=ichar(41)
      isav(4)=ichar(41)
      i2deck=0
      write(6,77) kcard
   77 format(32h the next event begins near card  ,i4)
      go to 80
   76 ierror=0
   75 ib=ib+1
      kchr=2
   80 if (ib.ge.80) call charac
c
c-----go to appropriate part of program to operate on character
c     ichar(ic(ib)). statement numbers generally are ic(ib)x10 ---------
c
      j=ic(ib)/12+1
      ij=ic(ib)
      if(ibug.eq.1) write(6,9080) ichar(ij),ij,ib,kstat
 9080 format(20h line 80  character ,a1,14h  array index=,i3,5h  ib=,i4,
     118h last line number=  ,i3)
      ipunch=1
      if (j.gt.4) go to 500
      go to (98,119 ,239 ,359 ),j
   98 if (ic(ib).eq.11)go to 120
c-----enter single numbers as weights.---------------------------------
      j=ic(ib)
      isav(4)=ichar(j)
      go to 75
  119 j=ic(ib)-11
      go to (120, 75,140,150,150,170,180,190,200,210,220,230),j
  120 if(itsum(m11).eq.10) go to 121
      go to 180
c-----read xy coordinates + calculate p time.--------------------------
  121 call xypont(u,v)
      if (ierror.eq.1) go to 76
      call numlin(kstat)
      kskip=0
  122 ist=1
      i=1
      if(isav(2).ne.ichar(19)) go to 123
c----- or calculate s time.----------------------------------------------
      ist=2
      i=10
  123 if (kskip.eq.1) go to 124
      timme=time(kstat)
      xkard(kstat,ist)=timme      +idate(6)-seccor(timme)
      xkard(kstat,6)=tcor+stacor(kstat)
      if (ist.eq.2) go to 126
 1124 do 125 j=1,5
      ix=j+4
  125 kard(kstat,ix )=idate(j)
  124 kard(kstat,i  )=isav(1)
      i=i+1
      kard(kstat,i  )=isav(3)
      i=i+1
      kard(kstat,i  )=isav(4)
      isav(1)=ichar(15)
      isav(2)=ichar(22)
      isav(3)=ichar(41)
      isav(4)=ichar(41)
      go to 80
c------be sure grid is same for s as p or correct for change.------------
  126 if (kard(kstat,5).eq.0) go to 1124
      if (kard(kstat,9).eq.idate(5)) go to 127
      xkard(kstat,ist)= xkard(kstat,ist)+( idate(5)-kard(kstat,9))*60.
  127 do 128 j=1,4
      jj=j+4
      if (kard(kstat,jj) .ne.idate(j)) go to 129
  128 continue
      go to 124
  129 write(6,1129) (kard(kstat,i),i=5,8),(idate(i),i=1,4)
 1129 format(30h0*warning* time for p-wave is  ,4i2,
     124h but time for s-wave is  ,4i2)
      go to 124
c------read time grid----------------------------------------------------
  140 call ptdist
      isav(1)=ichar(15)
      isav(2)=ichar(22)
      isav(3)=ichar(41)
      isav(4)=ichar(41)
      istcor(1)=0
      stcor(1)=0.0
      jstcor=0
      if (factor.lt. -9998.0) go to 1059
      ntg=ntg+1
      go to 80
c------enter e or i phase descriptor.------------------------------------
  150 j=ic(ib)
      isav(1)=ichar(j)
      go to 75
c------enter day etc or down first motion.-------------------------------
  170 if (itsum(m8).ne.0) go to 180
      k=3
  171 do 172 j=k,6
      idate(j)=ic(ib+1)*10+ic(ib+2)-11
  172 ib=ib+2
      write(6,174) (idate(j),j=1,6)
  174 format(16h grid starts at  ,6i2)
      ntm=ntm+1
c------enter first motions.----------------------------------------------
  180 j=ic(ib)
```

```
      isav(3)=ichar(j)
      go to 75
c-----enter second or specify s-wave.---------------------------------------
  190 if (itsum(m2).ne.0) go to 220
      k=6
      go to 171
c-----enter month etc.------------------------------------------------------
  200 if (itsum(m10).ne.0) go to 201
      k=2
      go to 171
  201 k=5
      if (itsum(m4).eq.0) go to 171
      j=20
  202 write(6,203) ichar(j)
  203 format(4h ** ,a1," not follow by proper number of numbers. ignor",
     1 "ed.")
      go to 75
c-----enter hour etc.-------------------------------------------------------
  210 k=4
      if (itsum(m6).eq.0) go to 171
      j=21
      go to 202
c-----specify p-wave.-------------------------------------------------------
  220 j=ic(ib)
      isav(2)=ichar(j)
      go to 75
c
c-----punch arrival time cards and procede to next event.---------------
  230 write(6,231)
  231 format(//)
      if (ntg.eq.ntm) go to 3230
      write(6,2230) ntm,ntg
 2230 format(1h0,31x,"*warning* you changed the time ",i3,
     1 " times and read a time grid in",i3," times for this earthquake",
     1 ".",/)
 3230 ntg=0
      ntm=0
      ns=0
      do 238 i=1,nstat
      if (kard(i,5).eq.0) go to 238
      ika=xkard(i,4)*100.+0.5
      iamp=xkard(i,3)+0.5
      m=i*4-3
      i4=m+3
      if (xkard(i,2).lt.9998.) go to 234
c-----write out p-wave data only---------------------------------------------
 1231 continue
      if(xkard(i,1).le.99.99) go to 1235
      xkard(i,1) = xkard(i,1) - 60.
      kard(i,9) = kard(i,9) + 1
 1235 write(6,233)  (istat(j),j=m,i4), (kard(i,j),j=1,9), xkard(i,1),
     1iamp,       ika,xkard(i,5),(kard(i,j),j=13,15), (xkard(i,j),j=6,7)
  233 format(1x,  5a1,1hP,3a1,5i2,f5.2,19x,i4  ,i3,f4.1,8x,3a1,f5.2,
     1f5.0)
      go to 2238
  234 if (xkard(i,2).lt.100.0) go to 235
      if (xkard(i,1) .lt. 50.01) go to 235
      xkard(i,1)=xkard(i,1)-60.0
      xkard(i,2)=xkard(i,2)-60.0
      kard(i,9)=kard(i,9)+1
  235 if (xkard(i,1).lt.9938.) go to 237
c-----write out s-wave data only---------------------------------------------
      write(6,236)  (istat(j),j=m,i4), (kard(i,j),j=4,9), xkard(i,2),
     1(kard(i,j),j=10,12),iamp       ,ika,xkard(i,5),(kard(i,j),j=13,15),
     1 (xkard(i,j),j=6,7)
  236 format(1x,4a1,4x,a1,5i2,12x,f5.2,a1,1hS,2a1,3x,i4  ,i3,f4.1,8x,
     13a1,f5.2,f5.0)
      ns=ns+1
      go to 2238
c-----write out p and s wave data.-------------------------------------
  237 fmt(1)="(1x,5a1,1hP,3a1,5i2,f5.2,7x,f5.2,a1,1hS,"
      fmt(2)="2a1,3x,i4,i3,f4.1,8x,3a1,f5.2,f5.0)     "
      if (xkard(i,2) .gt.99.99)
     1  fmt(1)="(1x,5a1,1hP,3a1,5i2,f5.2,7x,f5.1,a1,1hS,"
      write(6,fmt) (istat(j),j=m,i4), (kard(i,j),j=1,9),(xkard(i,j),j=1
     1,2),
     1(kard(i,j),j=10,12),iamp,       ika,xkard(i,5),(kard(i,j),j=13,15),
     1 (xkard(i,j),j=6,7)
      ns=ns+1
 2238 ind=ind+1
      write(8,rec=ind)(istat(j),j=m,i4),(kard(i,j),j=1,15),(xkard(i,j),
     1j=1,7)
  238 continue
c-----punch instruction card after each set of arrival time cards.------
      if (ns.eq.0) inos=ichar(41)
      write(6,1239) inos,izfix
 1239 format(18x,2a1)
      do 1237 j=3,15
 1237 kard(19,j)=0
      ind=ind+1
      write(8,rec=ind)(istat(j),j=m,i4),inos,izfix,i2deck,(kard(19,j),
     1j=4,15),(yfill(j),j=1,7)
      inos=ichar(2)
      izfix=ichar(41)
      id=ib
      ib=ib+1
      ipunch=0
      index2(kfilm)=ind
      if (ic(id).eq.23) go to 70
      if (ic(id).eq.25) go to 54
      if (ic(id).eq.30) go to 900
      go to 80
c
  239 j=ic(ib)-23
      go to (240,250,260,270,280,290,230,310,320,330,340,350),j
c-----determine coda length---------------------------------------------
  240 if (itype(ib+1).ne.10) go to 248
      ib=ib+1
      call xypont(u,v)
      call numlin(kstat)
      if (xkard(kstat,1).gt. 9998.) go to 245
      xkard(kstat,7)=time(kstat)+idate(6)-xkard(kstat,1)+(idate(5)-kard(
     1kstat,9))*60. + (idate(4) - kard(kstat,8))*3600.
      do 241 j=1,4
      jj=j+4
      if (kard(kstat,jj) .ne.idate(j)) go to 242
  241 continue
      go to 80
  242 write(6,243)  (kard(kstat,i),i=5,8),(idate(i),i=1,4)
  243 format(30h0*warning* time for p-wave is  ,4i2,
     1 24h but time for f-mag  is  ,4i2)
      go to 80
  245 write(6,246) kstat
  246 format(1h ,"p reading not read for line ",i2," so coda length ig",
     1 "nored.")
      go to 80
  248 j=24
      if( itsum(m3).ne.0) go to 202
      xkard(kstat,7)=100.*ic(ib+1)+10.*ic(ib+2)+ic(ib+3)-111.
      ib=ib+4
      go to 80
  250 ipunch=0
      ib=ib+1
      go to 54
c----calculate amplitude. ----------------------------------------------
  260 ib=ib+1
      call xypont(u1,v1)
      call xypont(u,v)
      call numlin(kstat)
c----calculate period of maximum amplitude------------------------------
      u=u1
      v=v1
      xkard(kstat,4)=time(i)
      call xypont(u,v)
      xkard(kstat,4)=2.*abs(time(i)-xkard(kstat,4))
      xkard(kstat,3)=sqrt((u1-u)**2+(v1-v)**2)*ampfac
      if(xkard(kstat,1).lt.9998.0) go to 80
      xkard(kstat,1)=0.0
      kard(kstat,1)=ichar(41)
      kard(kstat,2)=ichar(41)
      kard(kstat,3)=ichar(5)
      do 262 i=1,5
      ii=i+4
  262 kard(kstat,ii)=idate(i)
      go to 80
c-----enter time corrections--------------------------------------------
c-----corrections are algebraic
c-----time correction,if 2 c's in a row, read relative delays.----------
  270 if(ic(ib+1).eq.27) go to 274
      if (itsum(m4).ne.10) go to 180
      if (itsum(m12).eq.20) go to 180
c-----input single correction (c) for all stations.---------------------
      tcor=ic(ib+2)*1.+ic(ib+3)*0.1+ic(ib+4)*0.01-1.11
      if (ic(ib+1).eq.12) tcor=-tcor
      ib=ib+5
      go to 80
c-----if 3 c's in a row, zero relative delays
  274 if (ic(ib+2).eq.27) goto 278
      ib=ib+2
c-----calcuate time correction for each trace---------------------------
      call xypont(u,v)
      timcor=time(0)
      do 276 i=1,nstat
      kchr=3
      if(ib.gt.80) call charac
      if (itsum(m11).ne.10) goto 80
      call xypont(u,v)
      call numlin(kstat)
      stacor(kstat)=timcor - time(kstat)
      m=i*4-3
      i4=m+3
      write(6,277)(istat(j),j=m,i4),stacor(kstat)
  277 format(19h delay for station ,4a1,4h is ,f10.2, 5h sec.)
  276 continue
      go to 80
c-----zero relative delays
  278 ib=ib+3
      do 279 i=1,nstat
      stacor(i)=0.0
  279 write(6,277)(istat(j),j=m,i4),stacor(kstat)
      goto 80
c-----enter year--------------------------------------------------------
```

```
  280 k=1
      if(itsum(m12).eq.0) go to 171
      j=28
      go to 202
c-----enter station trace number.-------------------------------------
  290 kstat=ic(ib+1)*10+ic(ib+2)-11
      ib=ib+3
      go to 80
c-----enter remark----------------------------------------------------
  310 ib=ib+1
      i3=ib+2
      k=13
      do 312 i=ib,i3
      j=ic(i)
      kard(kstat,k)=ichar(j)
  312 k=k+1
      ib=ib+3
      go to 80
c-----enter amplitude p-p cal signal----------------------------------
  320 j=32
      if(itsum(m3).ne.0) go to 202
      xkard(kstat,5)=ic(ib+1)*10.+ic(ib+2)*1.+ic(ib+3)*0.1-11.1
      ib=ib+4
      go to 80
c-----enter period of maximum amplitude.------------------------------
  330 j=33
      if (itsum(m2).ne.0) go to 335
      xkard(kstat,4)=ic(ib+1)*0.1+ic(ib+2)*0.01-0.11
      ib=ib+3
      go to 80
  335 if (itsum(m24).ne.40) go to 202
      ib=ib+1
      call xypont(u,v)
      xkard(kstat,4)=time(i)
      call xypont(u,v)
      xkard(kstat,4)=abs(time(i)-xkard(kstat,4))
      go to 80
c-----enter refraction layer interface -------------------------------
  340 ip=ic(ib+1)
      kard(kstat,4)=ichar(ip)
      ib=ib+2
      go to 80
c-----put trace identifier with last trace picked.--------------------
  350 kskip=1
      ib=ib+1
      go to  122
  359 j=ic(ib)-35
      if (j.gt 11)go to 500
      go to (360,370,380,390,400,500,500,500,440,500,460), j
c-----for fixed depth solution----------------------------------------
  360 izfix=ichar(2)
      go to 75
  370 isav(3)=ichar(23)
      go to 75
c-----do not use s in the solution.-----------------------------------
  380 inos=ichar(41)
      go to 75
c-----punch deck with s and one without s.----------------------------
  390 i2deck=1
      go to 75
c-----punch calibration card------------------------------------------
  400 j=40
      if (itsum(m4).ne.0) go to 202
      isr=ic(ib+1)-1
      fcal=ic(ib+2)*10.+ic(ib+3)*1.+ic(ib+4)*0.1-11.1
      ib=ib+4
      i1=kstat*4-3
      i3=i1+3
      write(6,403)(istat(j),j=i1,i3), (kard(kstat,j),j=5,9),isr,fcal
  403 format(1x,4a1,5x, 5i2,1x,i1,37x,f4.1,3hcal)
      ind=ind+1
      write(8,rec=ind)(istat(j),j=i1,i3),(kard(kstat,j),j=1,9), isr ,
     1(kard(kstat,j),j=11,15), (xfill(j),j=1,6),fcal
      go to 75
c-----read change in xmin----------------------------------------------
  440 j=44
      if(itsum(m2).ne.0) go to 202
      xmin=ic(ib+1)*10.0+ic(ib+2)*1.0-11.0
      ib=ib+3
      write(6,441) xmin
  441 format(54h distance between grid marks in seconds is changed to ,
     1f3.0)
      go to 80
c-----calculate time corrections for second marks across page.----------
  460 ib=ib+1
      istcor(1)=0
      stcor(1)=0.0
      jstcor=1
  462 if(( ic(ib).eq.11).or.( ic(ib).eq.12)) go to 464
  463 write(6,469) (istcor(i),stcor(i),i=1,jstcor)
  469 format(29h film distortion corrections=  ,8(i6,f6.3))
      if(ic(ib).eq.46) go to 75
      go to 80
  464 if(itsum(m11).ne.10) go to 463
      if(jstcor.gt.80) go to 465
      call xypont(u,v)
      u1=time(kstat)
      jstcor=jstcor+1
      istcor(jstcor)=u1+0.5
      stcor(jstcor)=u1-istcor(jstcor)
      kchr=4
      if(ib.ge.80) call charac
      if(istcor(jstcor).ne.istcor(jstcor-1)) go to 462
      jstcor=jstcor-1
      istcor(jstcor)=istcor(jstcor+1)
      stcor(jstcor)=stcor(jstcor+1)
      go to 462
  465 write(6,466)
  466 format(39h more than 81 time corrections read in.
     141h assume future xy points are p-readings.    )
      go to 121
  500 ip=ic(ib)
      write(6,501)ichar(ip)
  501 format(30h illegal character identifier ,a1,9h ignored.)
      go to 75
 1058 write(6,59)
   59 format(53h coordinates of lines for new station list are not in
     1 48h expected format. looking for new station list.   )
 1059 ib=ib+1
      if (ic(ib).eq.25) go to 250
      if (ic(ib).eq.30) go to 900
      if ((ic(ib).eq. 23).and.(factor.lt.-9998.0)) go to 75
      kchr=5
      if (ib.gt.80) call charac
      go to 1059
  900 call phzout(kfilm,index1,index2,iunit)
  985 write(6,986)
  986 format(/,1x,15hkeep on truckin  ,/)
```

```
      close(7,status='keep')
      close(8,status='delete')
      end
      subroutine phzout(kfilm,index1,index2,l)
c
c
      integer*2 kard,istat,kfilm,index1,index2,l
      integer*4 jdy,idy
      dimension lastp(100),ida(12),kard(1,15),istat(4),index1(100),
     1index2(100),pfirst(100),plast(100),xkard(1,7)
c     kfilm           number of films
c     index1          top of current stack
c     index2          bottom of current stack
c     lastp           last card for current event
c     pfirst          earliest p time for current event
c     plast           latest p time for current event
c     pearly          earliest current p time
c     plate           latest current p time
      character*40 fmt
      dimension fmt(2)
      data ida(1),ida(2),ida(3),ida(4),ida(5),ida(6),ida(7),ida(8),
     1ida(9),ida(10),ida(11),ida(12)/0,31,59,90,120,151,181,212,243,
     1273,304,334/
c*****************************************************************
c     nxrec           index of next record to write out to disc
c     disk file 998 was already opened in digit1, to hold p-phase cards.
c     note well.. long hollerith blank fields are written out to disc
c     in order to read the p-phase cards in again with earthquake
c     location program.   be careful about format changes when writing
c     out to disc file 998.         sam stewart....may, 1974
c*****************************************************************
      data nxrec/1/
      save ida,nxrec
      nxrec=nxrec+1
   10 format(/)
 1001 pearly=10.e+30
      plate=-10.e+30
      xplate=10.e+30
      inst1=32
      inst2=32
      ind=1
      read(8,rec=ind)(istat(j),j=1,4),(kard(1,j),j=1,15),
     1(xkard(1,j),j=1,7)
      imo=kard(1,6)
      jdy=365.*kard(1,5)+ifix(kard(1,5)/4.)-25203.5
      jdy=jdy+ida(imo)+kard(1,7)
      mod=kard(1,5)-ifix(kard(1,5)/4.)*4
      if((imo.le.2).and.(mod.eq.0)) jdy=jdy-1
c      initalize stack for each film
      jump=1
      jf=0
   15 jf=jf+1
      if(jf .gt. kfilm) go to 25
      ifilm=jf
      go to 30
   25 jump=2
      goto 100
c     find earliest and latest p time for earthquake at top of stack
   30 ind=index1(ifilm)
      pfirst(ifilm)=10.e+30
      plast(ifilm)=-10.e+30
      nread=0
      temp1=10.e+30
      temp2=10.e+30
   40 read(8,rec=ind)(istat(j),j=1,4),(kard(1,j),j=1,15),
     1(xkard(1,j),j=1,7)
      test1=xkard(1,1)+xkard(1,2)
      if(test1.lt.-1599.) goto 50
c     compute p-arrival time in seconds
      imo=kard(1,6)
      idy=365.*kard(1,5)+ifix(kard(1,5)/4.)-jdy-25202.5
      idy=idy+ida(imo)+kard(1,7)
      mod=kard(1,5)-ifix(kard(1,5)/4.)*4
      if((imo.le.2).and.(mod.eq.0)) idy=idy-1
      time=xkard(1,1)+60.*(kard(1,9)+60.* kard(1,8))+idy*86400.
      if(xkard(1,1).gt.9998.) goto 42
      if(kard(1,3).eq.52) goto 45
      nread=1
      if(pfirst(ifilm).gt.time)  pfirst(ifilm)=time
      if(test1.lt.-1199.) goto 50
      if(plast(ifilm).lt.time) plast(ifilm)=time
      goto 48
c     no p time was read
   42 temp1=time-9939.
      goto 48
c     p time was 4-weighted
   45 temp2=time
   48 ind=ind+1
      goto 40
c     instruction card or calibration card terminates event
   50 if(nread.ne.0) goto 55
      if(temp1.lt.temp2) temp2=temp1
      pfirst(ifilm)=temp2
      plast(ifilm)=temp2
   55 lastp(ifilm)=ind

      goto (15,100),jump
c
c     find earliest event and punch it out
c
  100 jfilm=0
      pearly=10.e+30
      ipunch=0
      do 110 i=1,kfilm
c     skip ith stack if exhausted
      if(index1(i).gt. index2(i)) goto 110
      jfilm=1
      if(pfirst(i).gt.pearly) goto 110
      pearly=pfirst(i)
      plate=plast(i)
      ifilm=i
  110 continue
c     if pearly=plate either only one p time or all p times 4-weighted
      if(pearly.eq.plate) plate=plate+60.
c     check if event is within 10 sec. of previously output event
      if(pfirst(ifilm).le.(xplate+10)) ipunch=1
c     check if event is out of order
      if(xplate.eq.10.e+30) goto 115
      if(plast(ifilm).gt.(xplate-120)) goto 115
      write(6,112)
  112 format(/40h *****warning -- event out of order*****)
      ipunch=0
c
c     punch the earliest event
c
c     first punch an instruction card if ipunch=0 and previous event
c     was not a calibration (xplate=0)
  115 xplate=plate
```

```
      if((test2.gt.-1599 ).and.(test2.lt.-1199)) goto 125
      if(ipunch.ne.0) goto 125
      if(iswave.ne.1) go to 1015
      write(7,120)  inst2
      nxrec=nxrec+1
 1015 continue
      write(7,121)  inst1,inst2
      nxrec=nxrec+1
  120 format(5x,2h**,11x,a1,
     161h                                                             )
  121 format(17x,2a1,
     161h                                                             )
 1003 inst1=32
      inst2=32
c      if all data has been punched exit
  125 if(jfilm.eq.0) goto 200
c     punch phase lists
      ind=index1(ifilm)
  130 read(8,rec=ind)(istat(j),j=1,4), (kard(1,j),j=1,15),
     1(xkard(1,j),j=1,7)
      test2=xkard(1,1)+xkard(1,2)
      if(test2.lt.-1599.) goto 150
      if(test2.lt.-1199.) goto 160
      if(kard(1,3).eq.32) kard(1,3)=48
      iamp=xkard(1,3)+0.5
      ika=xkard(1,4)*100.+0.5
      if(xkard(1,2).lt.9998.) goto 134
c     write out p-wave data only
      write(7,132) (istat(j),j=1,4),
     1(kard(1,j),j=1,9),xkard(1,1),iamp,
     1ika,xkard(1,5),(kard(1,j),j=13,15),(xkard(1,j),j=6,7)
      nxrec=nxrec+1
  132 format(5a1,1hP,3a1,5i2,f5.2,19x,i4,i3,f4.1,8x,3a1,f5.2,f5.0,
     15h       )
      goto 170
  134 if(xkard(1,1).lt.9998.) goto 137
c     write out s wave data only
      write(7,135) (istat(j),j=1,4),
     1(kard(1,j),j=4,9),xkard(1,2),
     1(kard(i,j),j=10,12),iamp,ika,xkard(1,5),
     1(kard(1,j),j=13,15),(xkard(1,j),j=6,7)
      nxrec=nxrec+1
  135 format(4a1,4x,a1,5i2,12x,f5.2,a1,1hS,2a1,3x,i4,i3,f4.1,8x,
     13a1,f5.2,f5.0, 5h       )
      goto 170
c     write out p and s wave data
  137 continue
      fmt(1)="(5a1,1hP,3a1,5i2,f5.2,7x,f5.2,a1,1hS,2a1"
      fmt(2)=",3x,i4,i3,f4.1,8x,3a1,f5.2,f5.0,5h      )"
      if(xkard(1,2) .gt. 99.99)
     1fmt(1)="(5a1,1hP,3a1,5i2,f5.2,7x,f5.1,a1,1hS,2a1"
      write(7,fmt) (istat(j),j=1,4),
     1(kard(1,j),j=1,9),(xkard(1,j),j=1,2),
     1(kard(1,j),j=10,12),iamp,ika,xkard(1,5),
     1(kard(1,j),j=13,15),(xkard(1,j),j=6,7)
      nxrec=nxrec+1
      goto 170
c     handle instruction card
  150 if(kard(1,1).eq.49) inst1=49
      if(kard(1,2).eq.49) inst2=49
      if(kard(1,3).eq.1) iswave=1
      goto 180
c     handle calibration card
  160 continue
```

```
      write(7,162) (istat(j),j=1,4),
     1(kard(1,j),j=5,10),xkard(1,7)
      nxrec=nxrec+1
  162 format(4a1,5x,5i2,1x,i1,37x,f4.1,3hcal,15h                    )
      goto 180
  170 ind=ind+1
      goto 130
c     reinitalize the stack for ifilm
  180 index1(ifilm)=lastp(ifilm)+1
      if(index1(ifilm).gt.index2(ifilm)) goto 100
      goto 30
c     exit routine
  200 continue
      write(7,210)
      nxrec=nxrec+1
  210 format(40hzz                                        ,
     140h                                        )
 1013 return
      end
      function seccor(xtime1)
      implicit integer*2 (i-n)
      common        ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18)
     1,xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact
     1,terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82)
     1,at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      seccor=0
      if(jstcor.eq.0) return
      if(xtime1.lt.0) go to 302
      do 100 i=1,jstcor
      if(istcor(i).ge.xtime1) go to 120
  100 continue
      if(xtime1.gt.xmin) go to 302
      ixmin=xmin+0.5
      seccor=stcor(jstcor)-stcor(jstcor)*(xtime1-istcor(jstcor))/(ixmin-
     1istcor(jstcor))
      return
  120 i=i-1
      seccor=(stcor(i+1)-stcor(i))*(xtime1-istcor(i))/(istcor(i+1)-
     1istcor(i))+stcor(i)
      return
  302 write(6,301)
  301 format(1h ," ***caution***second occurs outside time grid so fi",
     1 "lm di",41hstortion correction may not be accurate.   )
      return
      end
      subroutine ptdist
c-----initialize constants for distance measurements.------------------
      implicit integer*2 (i-n)
      common        ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18)
     1,xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact
     1,terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82)
     1,at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      ib=ib+1
      ierro=0
      call xypont(x1,y1)
      if (ierror.eq.1) ierro=1
      call xypont(x2,y2)
      if (ierror.eq.1) ierro=1
      call xypont(x3,y3)
      if (ierror.eq.1) ierro=1
      call xypont(x4,y4)
      if (ierror.eq.1) go to 500
      if (ierro.eq.1) go to 500
      dtop=sqrt((y2-y1)**2+(x2-x1)**2)
      dbot=sqrt((y4-y3)**2+(x4-x3)**2)
      dlef=sqrt((y3-y1)**2+(x3-x1)**2)
      drig=sqrt((y4-y2)**2+(x4-x2)**2)
      if (abs(dlef-drig).lt. 0.1) go to 87
      write(6,88) dlef,drig
   88 format(52h *caution* distance between time grid points differs
     120h by more than 0.1   ,/,10x," distance at left is",f10.3,/,
     1 10x,24h distance at right  is   ,f10.3)
   87 if( abs(dtop-dbot).lt. 0.1) go to 90
      write(6,89) dtop,dbot
   89 format(52h *caution* distance between time grid points differs
     120h by more than 0.1   ,/,10x," distance at top  is",f10.3,/,
     1 10x,24h distance at bottom is   ,f10.3)
   90 terp  =(dbot-dtop) /((dlef+drig)*0.5)
      slope=20000
      if (abs(x1-x3).lt.0.001) go to 100
      slope =(y1-y3)/(x1-x3)
  100 bconst=y1-slope*x1
      pslope=(y1-y2)/(x1-x2)
      pconst=y1-pslope*x1
      fact=1/sqrt(pslope**2+1)
      factor=xmin/ sqrt(slope**2+1)
      at=(x2*y1-x1*y2)/(x2-x1)
      bt=(y2-y1)/(x2-x1)
      ab=(x4*y3-x3*y4)/(x4-x3)
      bb=(y4-y3)/(x4-x3)
      if (abs(bb-bt).lt.0.00001) go to 200
      xi=(at-ab)/(bb-bt)
      yi=(at*bb-ab*bt)/(bb-bt)
      go to 205
  200 xi=0.0
      yi=0.0
  205 if (abs(x3-x1).lt.0.0001) go to 210
      al=(x3*y1-x1*y3)/(x3-x1)
      bl=(y3-y1)/(x3-x1)
      go to 220
  210 al=0.0
      bl=0.0
  220 if (abs(x4-x2).lt.0.0001) go to 230
      ar=(x4*y2-x2*y4)/(x4-x2)
      br=(y4-y2)/(x4-x2)
      go to 240
  230 ar=0.0
      br=0.0
  240 continue
      return
  500 write(6,501)
  501 format(1h ,"time grid data contained letters.  look for new event",
     1 ".")
      factor=-9999.0
      end
      subroutine xypont(x,y)
      implicit integer*2 (i-n)
      external itsum
      dimension iout(12)
      common /blkda / ichar(50),ista
      common     ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18),
     1 xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact,
     1 terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82),
     1 at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      m11=11
      ierror=0
      sum=0
      if(itsum(m11).eq.10) go to 150
      ib11=ib+11
      do 100 i=1,12
      in=ib+i-1
      ij=ic(in)
  100 iout(i)=ichar(ij)
      write(6,1) (iout(i),i=1,12),itsum(m11)
    1 format(2x,'expected xy coordinates but found',1x,12a1,2x,
     1'itsm11=',i3)
      ierror=1
      return
  150 x=ic(ib+1)*10.0+ic(ib+2)*1.0+ic(ib+3)*0.1+ic(ib+4)*0.01+ic(ib+5)*
     10.001-11.111
      if (ic(ib).eq.12) x=-x
      y=ic(ib+7)*10.0+ic(ib+8)*1.0+ic(ib+9)*0.1+ic(ib+10)*0.01+ic(ib+11)
     1*0.001-11.111
      if (ic(ib+6).eq.12) y=-y
      ib=ib+12
      return
      end
      subroutine charac
      implicit integer*2 (i-n)
      external itsum
      dimension jstit(10,76),mstits(10)
      dimension icc(80)
      common     ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18),
     1 xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact,
     1 terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82),
     1 at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      common /blkda / ichar(50),ista
      data ibl,ncurd/0,-3/
      data jstn/0/
      save ncurd,jstn,ibl
c-----ib is the index of the next available character.------------------
c-----il is the index of the last available character.------------------
c-----shift remaining data to beginning of character register.----------
      m3=3
      if (il.eq.0) go to 100
c-----if jstat is 4 update station name array istit only
      if (jstat.ne.4) goto 40
      if (jstn.eq.0) goto 160
      iretrn=1
      goto 71
   40 n=ib
      ib=1
      do 50 i=n,il
      ic(ib)=ic(i)
      itype(ib)=itype(i)
   50 ib=ib+1
      j=ib-1
      if (ic(j).eq.30) go to 149
c-----read a new card
  100 il=ib+79
      read(5,1) (icc(i),i=1,80)
    1 format(80a1)
      call a to r (icc,80)
      ib1=ib-1
      do 55 i=1,80
      ic(ib1+i)=icc(i)
   55 continue
c-----if this is a station name card fill array istit.------------------
c-----istit will become istat when b is encountered in character stream
c-----in main program.--------------------------------------------------
      if ((ic(ib)+ic(ib+1)+ic(ib+2)).ne.ista) go to 102
      iretrn=2
      jstn=jstn+1
      if (jstn.lt.11) goto 90
      write(6,60)
   60 format(1h ,"***** error - charac reading over 10 station lists ",
     1 "ahead.")
      jstn=10
   90 do 101 i=1,72
      ibb=ib+i+7
      jstit(jstn,i)=ic(ibb)
      if (jstit(jstn,i).ne.ichar(41)) mstits(jstn)=i
  101 continue
c------count the number of stations.------------------------------------
      mstits(jstn)=(mstits(jstn)+3)/4
c------if istit is blank first station list in jstit is next
      do 70 i=1,4
      if (istit(i).ne.ichar(41)) goto 80
   70 continue
   71 nstits=mstits(1)
      do 72 i=1,72
   72 istit(i)=jstit(1,i)
      if (jstn.eq.1) goto 76
      do 74 ibb=2,jstn
      mstits(ibb-1)=mstits(ibb)
      do 74 i=1,72
   74 jstit(ibb-1,i)=jstit(ibb,i)
   76 jstn=jstn-1
   80 goto (160,81), iretrn
   81 jstat=1
c------if a b was not read in before the station list, put one there.---
      kchr=6
      if (ib.le.1) go to 100
      if (ic(ib-1).eq.25) go to 100
      ic(ib)=25
      itype(ib)=1
      ib=ib+1
      go to 100
c------find x's and change appropriate characters to blanks.-------------
  102 ifi=0
      i=ib
  103 if (ifi.eq.0) go to 106
      if (ic(i).eq.ichar(42)) go to 107
  104 k1=2*ifi-i
      ii=i-1
      do 105 k=k1,ii
  105 ic(k)=ichar(41)
      ifi=0
      if (k1.lt.ib) ib=k1
  106 if (ic(i).eq.ichar(42)) ifi=i
  107 i=i+1
      if (i.le.il) go to 103
      if (ifi.ne.0) go to 104
c------delete blanks. if whole card blank prepare to terminate program.--
      nblank=0
      i=ib
  118 if(ic(i).ne.ichar(41)) go to 120
      nblank=nblank+1
      if (nblank.eq.80) go to 149
      il=il-1
      do 119 j=i,il
  119 ic(j)=ic(j+1)
      if (il-i) 130,118,118
c------replace character by character array index.------------------------
  120 do 122 j=1,47
      if(ic(i).eq.ichar(j)) go to 125
  122 continue
      write(6,3) ic(i),ic(i),kchr
    3 format(3x,'char',1x,a1,2x,'value',1x,i6,3x,'do not match:delete',
     13x,i2)
      il=il-1
      do 123 j=i,il
  123 ic(j)=ic(j+1)
      if (il-i) 130,118,118
  125 ic(i)=j
c-----decide on type of character. numbers=0,signs=10,others=1.---------
      itype(i)=1
      if(j.le.10) itype(i)=0
      if ((j.eq.11).or. (j.eq.12)) itype(i)=10
      if (j.eq.30) go to 150
      i=i+1
      if (i.le.il) go to 118
c-----look for and check card numbers...-------------------------------
  130 ib=ib-ibl
      if (ibl.ne.0) ibl=0
  131 if (ic(ib).ne.13) go to 140
      if ((ib+3).le.il) go to 132
      ibl=il-ib+1
      ib=ib+ibl
      go to 141
  132 if(itsum(m3).eq.0) go to 135
      write(6,133)
  133 format(44h slash not followed by 3 numbers so ignored.)
      m=1
      il=il-1
      go to 9139
  135 j=ic(ib+1)*100+ic(ib+2)*10+ic(ib+3)-111
      if (j.eq.ncurd) go to 138
      if (ncurd.ne.-3) go to 137
      write(6,9137)j
 9137 format(24h0first card of deck is    ,i4)
      go to 138
  137 kk=ncurd-1
      write(6,136) j,kk
  136 format(7h *card ,i4,15h  follows card ,i4)
  138 ncurd=j+1
      kcard=j-2
      il=il-4
      m=4
      if(ib.gt.il) go to  141
 9139 do 139 i=ib,il
      imm=i+m
      ic(i)=ic(imm)
  139 itype(i)=itype(imm)
  140 ib=ib+1
      if (ib.le.il) go to 131
      kchr=7
  141 if (il.le.160) go to 100
      ib=1
      return
c-----if end of data is encountered, fill rest of ic array with ;-------
  149 i=ib
  150 do 155 k=i,240
      itype(k)=1
  155 ic(k)=30
      ib=1
      il=240
  160 return
      end
      subroutine newlst(kfilm,index1,ind,lcard,stacor,istat)
```

```
c-----subroutine to update station list and station position information
      implicit integer*2 (i-n)
      dimension index1(100),lcard(36),stacor(19),istat(76)
      common /blkda / ichar(50),ista
      common      ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18),
     1 xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact,
     i terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82),
     1 at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      kfilm=kfilm+1
      if (kfilm.gt.100) write(6,1054) kfilm
 1054 format(19h warning -- kfilm= ,i3,25h  too many station lists.)
      index1(kfilm)=ind+1
      if (kfilm.eq.1) goto 53
      if (index1(kfilm).ne.index1(kfilm-1)+1)  goto 53
      kfilm=kfilm-1
      ind=ind-1
   53 nstat=nstits*4
      do 55 i=1,nstat
      istat(i)=istit(i)
   55 istit(i)=ichar(41)
      nstat=nstits
      jstat=4
c-----charac called to update station list only
      kchr=8
      call charac
      jstat=2
      write(6,4) lcard,nstat,kfilm
    4 format(1h1,36a2,//, 34h number of stations available is  ,i3 ,
     1      5x,20hstation list number ,i3,/)
c-----calculate distances of each trace from upper time line.-----------
      if (ic(ib).eq.14) goto 62
      call xypont(u,v)
      do 60 i=i,nstat
      kchr=9
      if (ib.gt.80) call charac
      call xypont(x,y)
      if (ierror.eq.1) goto 70
   60 dist(i)=sqrt((x-u)**2+(y-v)**2)
      n=nstat-1
      do 65 i=1,n
   65 dist(i)=(dist(i)+dist(i+1))/2
      dist(nstat)=2*dist(nstat)-dist(n)
      if (dist(nstat)-dist(n).lt.1.0) dist(nstat)=dist(n)+1.0
      goto 64
   62 write(6,63)
   63 format(51h trace distances same as for previous station list.)
   64 do 66 l=i,nstat
      i4=l*4
      i=i4-3
   66 write(6,67) (istat(j),j=i,i4),dist(l),stacor(l)
   67 format(10h  station  ,4a1,4h is ,f8.3 ," inches from top time ",
     1 "line. ",10x,19hstation correction ,f10.2,5h sec.)
      write(6,68)
   68 format(////)
      return
   70 kfilm=kfilm-1
      ind=ind-1
      return
      end
      subroutine numlin(n)
      implicit integer*2 (i-n)
      common       ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18)
     1,xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact
     1,terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82)
     1,at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      d=abs(fact*(v-pslope*u-pconst))
      do 100 n=1,nstat
      if (d.lt.dist(n))go to 150
  100 continue
      write(6,1) d
    1 format(51h distance for trace identifier is too large. it is  ,
     1f10.3)
      n=19
  150 return
      end
      function time(ifake)
      implicit integer*2 (i-n)
      common       ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18)
     1,xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact
     1,terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82)
     1,at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
      if ((abs(xi).lt.0.00001).and.(abs(yi).lt.0.00001)) go to 100
      am=(u*yi-xi*v)/(u-xi)
      bm=(v-yi)/(u-xi)
      go to 110
  100 am=v-bt*u
      bm=bt
  110 if ((abs(al).gt.0.0001).or.(abs(bl).gt.0.0001)) go to 120
      xl=x1
      yl=am+bm*x1
      go to 130
  120 xl=(al-am)/(bm-bl)
      yl=(al*bm-am*bl)/(bm-bl)
  130 if  ((abs(ar).gt.0.0001).or.(abs(br).gt.0.0001)) go to 140
      xr=x2
      yr=am+bm*x2
      go to 150
  140 xr=(ar-am)/(bm-br)
      yr=(ar*bm-am*br)/(bm-br)
  150 diss=sqrt((xl-u)**2+(yl-v)**2)
      dmid=sqrt((xl-xr)**2+(yl-yr)**2)
      time=xmin*diss/dmid
      xslo=20000.
      if(abs(x1-u).lt.0.000001) go to 170
      xslo=(y1-v)/(x1-u)
  170 if (u.lt.xl) time=-time
      return
      end
      function itsum(k)
      implicit integer*2 (i-n)
      common       ic(240),itype(240),istit(76),ib,il,u,v,nstat,dist(18)
     1,xmin,slope,bconst,factor,pslope,pconst,ierror,nstits,jstat,fact
     1,terp,d,dtop,x1,y1,kcard,jstcor,istcor(82),stcor(82)
     1,at,bt,ab,bb,xi,yi,al,bl,ar,br,x2,y2,isav(4)
c-----find sum of types of next k characters.---------------------------
      j=ib+1
      jj=ib+k
      itsum=0
      do 10 i=j,jj
   10 itsum=itsum+itype(i)
      return
      end
        subroutine a to r (ary,n)
c       subroutine to convert an array (input param#1)
c       from a1 format to r1 format. The length of that
c       array is parameter #2.
c  use:
c       call a to r (array,number)
```

```
c  where:
c        array is the input in a1 format
c                It is converted in place
c        number is the length of the array
         integer*2 ary(n),blan
         blan = 8192
 1       do 2 i=1,n
 2       ary(i)=ary(i)-blan
         return
c        do the reverse of the above
         entry rtoa(ary,n)
         blan= -8192
         go to 1
         end
```

digchk

```
      program digchk
      character*1 icard,ifip,ibk,izero,ip,icp
      dimension icard(80),ifip(12)
      data ifip/' ','0','1','2','3','4','5','6','7','8','9','.'/
      data ibk/' '/,izero/'0'/,ip/'p'/,icp/'P'/
      open(2,file='data.c',status='new',form='formatted',blank='zero')
      open(unit=5,access='sequential',form='formatted',blank='zero')
      open(unit=3,file='data.err',access='sequential',
     1form='formatted',blank='zero')
      rewind 3
      rewind 2
      itrp=0
      kr=0
      i=0
   10 i=i+1
      ierror=1
      il=i
      if(i .gt. 3000) go to 400
      read(5,100,end=400) icard
  100 format(80a1)
   13 do 20 j=8,36
      do 15 k=1,12
      if(icard(j) .eq. ifip(k)) go to 20
   15 continue
      ierror=2
      write(3,125)icard,il
      kr=kr+1
      icard(j)=ibk
   20 continue
      do 30 j=40,62
      do 25 k=1,12
      if(icard(j) .eq. ifip(k)) go to 30
   25 continue
      ierror=2
      write(3,125)icard,il
      kr=kr+1
      icard(j)=ibk
   30 continue
      do 40 j=66,80
      do 35 k=1,12
      if(icard(j) .eq. ifip(k)) go to 40
   35 continue
      ierror=2
      write(3,125)icard,il
      kr=kr+1
      icard(j)=ibk
   40 continue
      if(icard(6) .ne. ip .and. icard(6) .ne. icp) go to 60
      do 50 j=10,19
      do 45 k=1,11
      if(icard(j) .eq. ifip(k)) go to 50
   45 continue
      ierror=2
      write(3,125)icard,il
      kr=kr+1
      icard(j)=ibk
   50 continue
      go to 70
   60 if(icard(18) .eq. ibk) icard(18)=izero
      if(icard(19) .eq. ibk) icard(19)=izero
   70 if(ierror .eq. 1)  itrp=1
  125 format(80a1,/,5x,'error on line',i5,/)
  290 write(2,100)icard
      if(kr .gt. 100) go to 400
```

d80p

```
      program d80p
      double precision tm
      character*80 a
      character*4 last
      character*1 ip,icp,a1,a2,a6
      character*1 icard(80),izero,iblnk,ione
      character*1 last1,last2
      dimension ida(12),tm(500),idx(500)
      data ip/"p"/,icp/"P"/
      data last/'ZZZZ'/,ione/'1'/
      data ida/0,31,59,90,120,151,181,212,243,273,304,334/
      data izero/"0"/,iblnk/" "/
      data last1/"Z"/,last2/"z"/
      open(8,status='scratch',access='direct',
     1form='formatted',recl=80,blank='zero')
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
      iph=0
 310  i=0
 315  read(5,101)a,a1,a2,a6
 101  format(a80,t1,2a1,t6,a1)
 317  if((a1.eq.last1 .or. a1.eq.last2) .and.
     1(a2.eq.last1 .or. a2.eq.last2)) go to 400
      if(a6 .eq. ip .or. a6 .eq. icp) go to 319
      if(iph .eq. 0) go to 355
      write(6,102)a
 102  format(a80)
      go to 310
 319  iph=0
      i=i+1
      ind=i
      write(8,102,rec=ind) a
      read(a,320)kyr,kmo,kdy,khr,kmn,sec
 320  format(9x,5i2,f5.2,56x)
 322  if(kyr.eq.0) go to 355
      jdy=365.*kyr+ifix((kyr-1)/4.)-29219
      jdy=jdy+ida(kmo)+kdy
      mod=kyr-ifix(kyr/4.)*4
      if((kmo.gt.2) .and.(mod.eq.0)) jdy=jdy+1
      ktm=86400*jdy+3600*khr
      if(i .gt. 1) go to 352
      ktm1=ktm
 352  tm(i)=float(ktm-ktm1)+float(60*kmn)+sec
      go to 315
 355  iph=1
      i=i+1
      ind=i
      write(8,102,rec=ind) a
      ii=i
      i1=i-1
      call sort(tm,idx,i1)
      idx(ii)=ii
      do 360 k=1,ii
      ind=idx(k)
      read(8,150,rec=ind) icard
 150  format(80a1)
      if(icard(6) .ne. ip .and. icard(6) .ne. icp) go to 359
      do 358 kj=10,19
      if(icard(kj) .eq. iblnk) icard(kj)=izero
 358  continue
      icard(80)=izero
 359  write(6,150) icard
 360  continue
      go to 310
 400 write(6,410) last
 410 format(a4,75x,'0')
     stop
     end
     SUBROUTINE SORT(X,KEY,NO)
     DOUBLE PRECISION X
     DIMENSION X(NO),KEY(NO)
     DO 1 I=1,NO
1    KEY(I)=I
     MO=NO
2    IF (MO-15) 21,21,23
21   IF (MO-1) 29,29,22
22   MO=2*(MO/4)+1
     GO TO 24
23   MO=2*(MO/8)+1
24   KO=NO-MO
     JO=1
25   I=JO
26   IF (X(I)-X(I+MO)) 28,28,27
27   TEMP=X(I)
     X(I)=X(I+MO)
     X(I+MO)=TEMP
     KEMP=KEY(I)
     KEY(I)=KEY(I+MO)
     KEY(I+MO)=KEMP
     I=I-MO
     IF (I-1) 28,26,26
28   JO=JO+1
     IF (JO-KO) 25,25,2
29   RETURN
     END
```

d80

```
      program d80
      character*1 icard(80),izero
      data izero/'0'/
      l=0
    1 l=l+1
      read(5,701,end=900)icard
  701 format(80a1)
      icard(80)=izero
      if(l .gt. 2000) go to 900
      write(6,701) icard
      go to 1
  900 stop
      end
      PROGRAM HYPO71
C------- PROGRAM: HYPO71 (DEC. 21, 1971; REVISED NOV. 25, 1973) --------
      CHARACTER*4 MSTA,MBK,MDOL,BLANK,MSTAR,DOT,STAR4
      CHARACTER*4 MCENT,WRK,AZRES,PRMK
      CHARACTER*3 CRMK,RMK
      CHARACTER*1 QUES,ISTAR,INS,IEW,IW
      INTEGER*2 SYM
      REAL*8 TIME1,TIME2
      REAL LATEP,LONEP,MAG,LATR,LONR
      COMMON /A3/ NRES(2,102),NXM(102),NFM(102),SR(2,102),SRSQ(2,102),
     1       SRWT(2,102),SXM(102),SXMSQ(102),SFM(102),SFMSQ(102),QNO(4)
      COMMON /A5/ ZTR,XNEAR,XFAR,POS,IQ,KMS,KFM,IPUN,IMAG,IR,QSPA(9,40)
      COMMON /A6/ NMAX,LMAX,NS,NL,MMAX,NR,FNO,Z,X(4,101),ZSQ,NRP,DF(101)
      COMMON /A7/ KP,KZ,KOUT,WT(101),Y(4),SE(4),XMEAN(4)
      COMMON /A8/ CAL(101),XMAG(101),FMAG(101),NM,AVXM,SDXM,NF,AVFM,
     1       SDFM,MAG,KDX(101),AMX(101),PRX(101),CALX(101),FMP(101)
      COMMON /A12/ MSTA(101),PRMK(101),W(101),JMIN(101),P(101),
     1       WRK(101),TP(101),DT(101),RMK(101)
      COMMON /A14/ MBK,MDOL,BLANK,MSTAR,DOT,STAR4,QUES,CRMK,MCENT,ISTAR
      COMMON /A15/ M,L,J,ORG,JAV,PMIN,AZRES(101),NEAR,IDXS,LATEP,LONEP
      COMMON /A17/ TIME1,TIME2,LATR,LONR,KTEST,KAZ,KSORT,KSEL,XFN
      COMMON /A19/ KNO,IELV(102),TEST(15),FLT(2,102),MNO(102),IW(102)
      COMMON /A21/ KSMP(102),FMO,ONF,B(4),IPH,KF,AVRPS,IEXIT
      COMMON /A23/ AIN(101),RMS,ADJ,SYM(101)
      COMMON /A25/ INS(102),IEW(102),JPH
C-----------------------------------------------------------------------
      OPEN(UNIT=8,FILE='hypo.input',BLANK='ZERO')
      OPEN(UNIT=9,FILE='hypo.print')
      REWIND 8
      REWIND 9
      OPEN(UNIT=2,FILE='hypo.smp')
      REWIND 2
      OPEN (UNIT=3,FILE='calstn',ACCESS='direct',FORM='formatted',
     X RECL=81,BLANK='ZERO')
      REWIND 3
      OPEN(UNIT=4,FILE="hypo.punch")
      REWIND 4
      FNULL=9.9
   30 M=0
C------- INPUT STATION LIST, CRUSTAL MODEL, & CONTROL CARD -------------
   40 CALL INPUT1
      IF(IPUN .EQ. 0) GO TO 44
      WRITE(4,41)
      WRITE(2,41)
   41 FORMAT(" DATE    ORIGIN     LAT \    LONG      DEPTH     MAG",
     1" NO GAP DMIN  RMS  ERH  ERZ QM")
C-------- INITIALIZE SUMMARY OF RESIDUALS ------------------------------
   44 DO 48 L=1,NS
      NRES(1,L)=0
      NRES(2,L)=0
      NXM(L)=0
      NFM(L)=0
      SR(1,L)=0.
      SR(2,L)=0.
      SRSQ(1,L)=0.
      SRSQ(2,L)=0.
      SRWT(1,L)=0.
      SRWT(2,L)=0.
      SXM(L)=0.
      SXMSQ(L)=0.
      SFM(L)=0.
      SFMSQ(L)=0.
   48 CONTINUE
      DO 49 I=1,4
   49 QNO(I)=0.
      XFN=XFAR-XNEAR+0.000001
      TIME1=0 D+00
   50 CALL INPUT2
C------- TO PROCESS ONE EARTHQUAKE -------------------------------------
      IF (M .EQ. 1) GO TO 900
      IF (NR .GE. 1) GO TO 100
      WRITE(9,55)
   55 FORMAT( ///," ***** EXTRA BLANK CARD ENCOUNTERED *****")
      GO TO 50
  100 CALL SINGLE
      IF (IEXIT .EQ. 1) GO TO 50
C------- COMPUTE SUMMARY OF MAGNITUDE RESIDUALS -----------------------
  110 IF (JAV .GT. IQ) GO TO 50
      DO 150 I=1,NRP
      IF  (XMAG(I) .EQ. FNULL) GO TO 120
      JI=KDX(I)
      DXMAG=XMAG(I)-AVXM
      NXM(JI)=NXM(JI)+1
      SXM(JI)=SXM(JI)+DXMAG
      SXMSQ(JI)=SXMSQ(JI)+DXMAG**2
  120 IF  (FMAG(I)  EQ. FNULL) GO TO 150
      JI=KDX(I)
      DFMAG=FMAG(I)-AVFM
      NFM(JI)=NFM(JI)+1
      SFM(JI)=SFM(JI)+DFMAG
      SFMSQ(JI)=SFMSQ(JI)+DFMAG**2
  150 CONTINUE
      GO TO 50
  900 CONTINUE
C------- END OF ONE DATA SET: PRINT SUMMARY OF RESIDUALS & RETURN ------
      IF (MSTA(NR+1) .EQ. MSTAR) GO TO 30
      M=1
      IF (MSTA(NR+1) .EQ. MDOL) GO TO 40
      M=2
      IF (MSTA(NR+1) .EQ. MCENT) GO TO 40
       WRITE(9,130)
  130  FORMAT('\f')
      STOP 22
      END
      BLOCK DATA
C------- INITIALIZE CONSTANTS IN COMMON STATEMENTS --------------------
      INTEGER*4 IBLANK
      REAL RBLANK
      CHARACTER*1 CLASS,QUES,ISTAR
      CHARACTER*1 IB1,KN1,KW1,KS1
      CHARACTER*3 CRMK
      CHARACTER*4 MBK,DOT,MSTAR,MDOL,MCENT,ISTTT,BLANK,STAR4,IONE,AHEAD
      CHARACTER*4 ZDOT,IPRO,CBLANK,IDSTA,LAZT
      COMMON /A6/ NMAX,LMAX,NS,NL,MMAX,NR,FNO,Z,X(4,101),ZSQ,NRP,DF(101)
      COMMON /A10/ ANIN(101),AZ(101),TEMP(101),CA(71),CB(71)
      COMMON /A13/ JDX(102),LDX(101),KEY(101),CLASS(4)
      COMMON /A14/ MBK,MDOL,BLANK,MSTAR,DOT,STAR4,QUES,CRMK,MCENT,ISTAR
      COMMON /A22/ F( 9, 9),G(4, 9),H( 9),DEPTH( 9),IONE
      COMMON /A24/ FLTEP,IPRO,ISTTT,ISKP(4),AHEAD(12),FLIM,AF(3),NDEC
      COMMON /S25/ ZDOT,CBLANK,IBLANK,RBLANK,LAZT
      COMMON /S26/ IDSTA(501),NMAXX,IB1,KN1,KW1,KS1,NSMAX
      DATA CA/  1. 855365,1. 855369,1. 855374,1. 855383,1. 855396,1. 855414,
     1 1. 855434,1. 855458,1. 855487,1. 855520,1. 855555,1. 855595,1. 855638,
     2 1. 855683,1. 855733,1. 855786,1. 855842,1. 855902,1. 855966,1. 856031,
     3 1. 856100,1. 856173,1. 856248,1. 856325,1. 856404,1. 856488,1. 856573,
     4 1. 856661,1. 856750,1. 856843,1. 856937,1. 857033,1. 857132,1. 857231,
     5 1. 857331,1. 857435,1. 857538,1. 857643,1. 857750,1. 857858,1. 857964,
     6 1. 858074,1. 858184,1. 858294,1. 858403,1. 858512,1. 858623,1. 858734,
     7 1. 858842,1. 858951,1. 859061,1. 859170,1. 859276,1. 859384,1. 859488,
     8 1. 859592,1. 859695,1. 859798,1. 859896,1. 859995,1. 860094,1. 860187,
     9 1. 860279,1. 860369,1. 860459,1. 860544,1. 860627,1. 860709,1. 860787,
     A 1. 860861,1. 860934/
      DATA CB/  1. 842808,1. 842813,1. 842830,1. 842858,1. 842898,1. 842950,
     1 1. 843011,1. 843085,1. 843170,1. 843265,1. 843372,1. 843488,1. 843617,
     2 1. 843755,1. 843903,1. 844062,1. 844230,1. 844408,1. 844595,1. 844792,
     3 1. 844998,1. 845213,1. 845437,1. 845668,1. 845907,1. 846153,1. 846408,
     4 1. 846670,1. 846938,1. 847213,1. 847495,1. 847781,1. 848073,1. 848372,
     5 1. 848673,1. 848980,1. 849290,1. 849605,1. 849922,1. 850242,1. 850565,
     6 1. 850890,1. 851217,1. 851543,1. 851873,1. 852202,1. 852531,1. 852860,
     7 1. 853188,1. 853515,1. 853842,1. 854165,1. 854487,1. 854805,1. 855122,
     8 1. 855433,1. 855742,1. 856045,1. 856345,1. 856640,1. 856928,1. 857212,
     9 1. 857490,1. 857762,1. 858025,1. 858283,1. 858533,1. 858775,1. 859008,
     A 1. 859235,1. 859452/
      DATA MBK,DOT,MSTAR,MDOL,MCENT/"    ","  .  "," ***"," $$$"," %%%"/
      DATA ISTTT/" ** "/
      DATA BLANK,STAR4,CLASS/"    ","****","A","B","C","D"/,QUES/"?"/
      DATA LMAX,MMAX,NMAX/ 9,101,501/,CRMK,ISTAR,IONE/"CAL","*","1   "/
      DATA AHEAD/"    ","    ","    ","    ","    ","    ","    ",
     1 "    ","    ","    ","    ","    "/
      DATA ZDOT/"0.  "/,CBLANK/"    "/,IBLANK/"    "/,RBLANK/"    "/
      DATA NSMAX/102/,LAZT/"ZZZZ"/
      END
```

input 1

70 83

```
      SUBROUTINE INPUT1
C------- INPUT STATION LIST, CRUSTAL MODEL, AND CONTROL CARD -------------
      CHARACTER*1 IB1, KN1, KW1, INS, IEW, KS1
      CHARACTER*1 QUES, ISTAR, IW, NEWLYN
      CHARACTER*3 CRMK
      CHARACTER*4 MDOL, BLANK, MSTAR, DOT, STAR4, MCENT, AZRES, IPRO, ISTTT
      CHARACTER*4 HEAD, AHEAD, BHEAD, XEMP
      CHARACTER*4 ISW, IONE, MBK, NSTA, IDSTA
      REAL*8 TIME1, TIME2
      REAL LAT, LON, LAT2, LON2, LATR, LONR
      REAL LATEP, LONEP
      COMMON /A1/ NSTA(102), DLY(2, 102), FMGC(102), XMGC(102), KLAS(102),
     1        PRR(102), CALR(102), ICAL(102), IS(102)
      COMMON /A2/ LAT(102), LON(102), DELTA(101), DX(101), DY(101), T(101)
      COMMON /A5/ ZTR, XNEAR, XFAR, POS, IQ, KMS, KFM, IPUN, IMAG, IR, QSPA(9, 40)
      COMMON /A6/ NMAX, LMAX, NS, NL, MMAX, NR, FNO, Z, X(4, 101), ZSQ, NRP, DF(101)
      COMMON /A14/ MBK, MDOL, BLANK, MSTAR, DOT, STAR4, QUES, CRMK, MCENT, ISTAR
      COMMON /A15/ M, L, J, ORG, JAV, PMIN, AZRES(101), NEAR, IDXS, LATEP, LONEP
      COMMON /A16/ KLSS(102), CALS(102), IPRN, ISW
      COMMON /A17/ TIME1, TIME2, LATR, LONR, KTEST, KAZ, KSORT, KSEL, XFN
      COMMON /A19/ KNO, IELV(102), TEST(15), FLT(2, 102), MNO(102), IW(102)
      COMMON /A20/ V( 9), D( 9), VSQ( 9), THK( 9), TID( 9,  9), DID( 9,  9)
      COMMON /A22/ F( 9,  9), G(4,  9), H( 9), DEPTH( 9), IONE
      COMMON /A24/ FLTEP, IPRO, ISTTT, ISKP(4), AHEAD(12), FLIM, AF(3), NDEC
      COMMON /A25/ INS(102), IEW(102), JPH
      COMMON /S26/ IDSTA(501), NMAXX, IB1, KN1, KW1, KS1, NSMAX
      DIMENSION BHEAD(12),  ATEST(15)
      DATA HEAD/"HEAD"/
C-------------------------------------------------------------------------
      DO 350 I=1, 15
      ATEST(I) = 1.23456
  350 CONTINUE
      WRITE(9, 300)
  300 FORMAT('\f')
      IF (M-1) 1, 100, 200
C------- INITIALIZE TEST VARIABLES ----------------------------------------
    1 TEST(1)=0.10
      TEST(2)=10.
      TEST(3)=2.
      TEST(4)=0.05
      TEST(5)=5.
      TEST(6)= 4.
      TEST(7)=-0.87
      TEST(8)=+2.00
      TEST(9)=+0.0035
      TEST(10)=100.
      TEST(11)=8.0
      TEST(12)=0.5
      TEST(13)= 1.
      IFLAG=0
C------- INPUT RESET TEST-VARIABLE CARDS AND SELECTION CARD ------------
      DO 5 I=1, 16
      READ(8, 4 ) ISW, J, TESTJ, BHEAD
    4 FORMAT(A4,  7X,  I2,  2X, F9.4, 12A4)
   11 IF ((ISW.EQ.MBK).OR.(ISW.EQ.IONE)) GO TO 6
      IF(ISW .NE. HEAD) GO TO 12
      DO 13 II=1, 12
      AHEAD(II)= BHEAD(II)
   13 CONTINUE
      GO TO 5
   12 IFLAG=1
      ATEST(J)=TESTJ
    5 CONTINUE
    6 WRITE(9, 14) AHEAD
   14 FORMAT(40X,12A4)
      WRITE(9,2)
    2 FORMAT(///," ********** PROGRAM: HYPO71 REVISED ( 8-30-79) *******
     1",    ///,13X,"TEST(1)  TEST(2)  TEST(3)  TEST(4)  TEST(5)  TEST(6
     2)  TEST(7)  TEST(8)  TEST(9) TEST(10) TEST(11) TEST(12) TEST(13)")
      WRITE(9,3) (TEST(I),I=1,13)
    3 FORMAT(" STANDARD ",13F9.4)
      IF (IFLAG .EQ. 0) GO TO 8
      DO 16 I = 1,15
      IF(ATEST(I) .NE. 1.23456) TEST(I)=ATEST(I)
   16 CONTINUE
      WRITE(9,7) (TEST(I),I=1,13)
    7 FORMAT(" RESET TO ",13F9.4)
C------- SQUARE SOME TEST-VARIABLES FOR LATER USE ----------------------
    8 TEST(1)=TEST(1)**2
      TEST(2)=TEST(2)**2
      TEST(4)=TEST(4)**2
C------- INPUT STATION LIST --------------------------------------------
      DO 50 L=1,NMAX
      IF(ISW.EQ.IONE) GO TO 30
      READ(3,25,REC=L) XEMP,NEWLYN
25    FORMAT(2X,A4,74X,A1)
27    IDSTA(L)=XEMP
      IF(XEMP.EQ.MBK) GO TO 60
      GO TO 50
30    CONTINUE
      READ(3,35,REC=L) XEMP,NEWLYN
35    FORMAT(A4,76X,A1)
      GO TO 27
50    CONTINUE
      WRITE(9,55) NMAX
   55 FORMAT(///," ***** ERROR: STATION LIST EXCEEDS ARRAY DIMENSION
     xOF ",I4)
      STOP 1
60    NMAXX=L-1
C-----NMAXX is actual no. of stations in the complete station list
C------- INPUT CRUSTAL MODEL ------------------------------------------
  100 WRITE(9,105)
  105 FORMAT(///,7X,"CRUSTAL MODEL 1",/,5X,"VELOCITY     DEPTH")
      DO 130 L=1,LMAX
      READ(8,115 ) V(L),D(L)
  115 FORMAT(2F7.3)
      IF (V(L) .LT. 0.01) GO TO 140
      WRITE(9,125) V(L),D(L)
  125 FORMAT(3X,2F10.3)
      DEPTH(L)=D(L)
      VSQ(L)=V(L)**2
  130 CONTINUE
      WRITE(9,135)
  135 FORMAT(///," ***** ERROR: CRUSTAL MODEL EXCEEDS ARRAY DIMENSION")
      STOP 2
  140 NL=L-1
      N1=NL-1
C-----LAYER THICKNESS THK,F & G TERMS
      DO 145 L=1,N1
      THK(L)=D(L+1)-D(L)
  145 H(L)=THK(L)
C---- COMPUTE TID AND DID
      DO 150 J=1,NL
      G(1,J)=SQRT(ABS(VSQ(J)-VSQ(1)))/(V(1)*V(J))
      G(2,J)=SQRT(ABS(VSQ(J)-VSQ(2)))/(V(2)*V(J))
      G(3,J)=V(1)/SQRT(ABS(VSQ(J)-VSQ(1))+0.000001)
      G(4,J)=V(2)/SQRT(ABS(VSQ(J)-VSQ(2))+0.000001)
      IF (J .LE. 1) G(1,J)=0.
      IF (J .LE. 2) G(2,J)=0.
      IF (J .LE. 1) G(3,J)=0.
      IF (J .LE. 2) G(4,J)=0.
      DO 150 L=1,NL
      F(L,J)=1.
      IF (L .GE. J) F(L,J)=2.
  150 CONTINUE
      DO 165 J=1,NL
      DO 165 M=1,NL
      TID(J,M)=0.
  165 DID(J,M)=0.
      DO 170 J=1,NL
      DO 170 M=J,NL
      IF (M .EQ. 1) GO TO 170
      M1=M-1
      DO 160 L=1,M1
      SQT=SQRT(VSQ(M)-VSQ(L))
      TIM=THK(L)*SQT/(V(L)*V(M))
      DIM=THK(L)*V(L)/SQT
      TID(J,M)=TID(J,M)+F(L,J)*TIM
  160 DID(J,M)=DID(J,M)+F(L,J)*DIM
  170 CONTINUE
C------- INPUT CONTROL CARD -------------------------------------------
  200 WRITE(9,205)
  205 FORMAT(///," ZTR XNEAR XFAR  POS   IQ  KMS  KFM IPUN IMAG   IR"
     1," IPRN CODE   LATR       LONR")
      READ(8,215) ZTR,XNEAR,XFAR,POS,IQ,KMS,KFM,IPUN,IMAG,IR,IPRN
     1,KTEST,KAZ,KSORT,KSEL,LAT1,LAT2,LON1,LON2
  215 FORMAT(3F5.0,F5.2,7I5,1X,4I1,2(I4,F6.2))
      WRITE(9,215) ZTR,XNEAR,XFAR,POS,IQ,KMS,KFM,IPUN,IMAG,IR,IPRN
     1,KTEST,KAZ,KSORT,KSEL,LAT1,LAT2,LON1,LON2
      LATR=60.*LAT1+LAT2
      LONR=60.*LON1+LON2
      IF(IPUN.EQ.0) GOTO 220
220   CONTINUE
      IF (IR .EQ. 0) RETURN
      DO 240 I=1,IR
      READ(8,225) (QSPA(I,J),J=1,40)
  225 FORMAT(20F4.2)
      WRITE(9,235) I,(QSPA(I,J),J=1,40)
  235 FORMAT(/," QSPA(",I1,"): ",20F5.2,/,10X,20F5.2)
  240 CONTINUE
      RETURN
      END
```

p. 88 followed

input2

```
      SUBROUTINE INPUT2
C------- INPUT PHASE LIST ----------------------------------------------------
      REAL RBLANK
      INTEGER*4 JTIME, KTIME, KDATE, IBLANK
      INTEGER*2 SYM
      CHARACTER*4 RMK1, RMK2, DOT, STAR4, AZRES, ISW, ISTTT, AHEAD, IONE
      CHARACTER*4 IPRO, AZRES, BLANK, WRK, PRMK, SRMK, ZDOT, CBLANK
      CHARACTER*4 NSTA, MSTA, MSTAR, MDOL, MCENT, MBK, AS, IDSTA, LAZT
      CHARACTER*3 CRMK, RMK
      CHARACTER*2 Q, QS, QD
      CHARACTER*1 ICARD, IW, INS, IEW, KN1, KW1, KS1
      CHARACTER*1 CLASS, QUES, ISTAR, IB1, NEWLYN
      REAL*8 TIME1, TIME2
      REAL LAT2, LON2, LATEP, LONEP, MAG, LATR, LONR, LAT, LON
      DIMENSION ICARD(80)
      COMMON /A1/ NSTA(102), DLY(2, 102), FMGC(102), XMGC(102), KLAS(102),
     1       PRR(102), CALR(102), ICAL(102), IS(102)
      COMMON /A2/ LAT(102), LON(102), DELTA(101), DX(101), DY(101), T(101)
      COMMON /A6/ NMAX, LMAX, NS, NL, MMAX, NR, FNO, Z, X(4, 101), ZSQ, NRP, DF(101)
      COMMON /A8/ CAL(101), XMAG(101), FMAG(101), NM, AVXM, SDXM, NF, AVFM,
     1       SDFM, MAG, KDX(101), AMX(101), PRX(101), CALX(101), FMP(101)
      COMMON /A10/ ANIN(101), AZ(101), TEMP(101), CA(71), CB(71)
      COMMON /A11/ KDATE, KHR, KMIN, SEC, LAT1, LAT2, LON1, LON2, RMK1, RMK2,
     1       IGAP, DMIN, RMSSQ, ERH, Q, QS, QD, ADJSQ, INST, AVR, AAR, NI, KNST, JHR
      COMMON /A12/ MSTA(101), PRMK(101), W(101), JMIN(101), P(101),
     1       WRK(101), TP(101), DT(101), RMK(101)
      COMMON /A13/ JDX(102), LDX(101), KEY(101), CLASS(4)
      COMMON /A14/ MBK, MDOL, BLANK, MSTAR, DOT, STAR4, QUES, CRMK, MCENT, ISTAR
      COMMON /A15/ M, L, J, ORG, JAV, PMIN, AZRES(101), NEAR, IDXS, LATEP, LONEP
      COMMON /A16/ KLSS(102), CALS(102), IPRN, ISW
      COMMON /A17/ TIME1, TIME2, LATR, LONR, KTEST, KAZ, KSORT, KSEL, XFN
      COMMON /A18/ S(101), SRMK(101), WS(101), TS(101), NOS, QRMK(101)
      COMMON /A19/ KNO, IELV(102), TEST(15), FLT(2, 102), MNO(102), IW(102)
      COMMON /A20/ V( 9), D( 9), VSQ( 9), THK( 9), TID( 9, 9), DID( 9, 9)
      COMMON /A21/ KSMP(102), FMO, ONF, B(4), IPH, KF, AVRPS, IEXIT
      COMMON /A22/ F( 9, 9), G(4, 9), H( 9), DEPTH( 9), IONE
      COMMON /A23/ AIN(101), RMS, ADJ, SYM(101)
      COMMON /A24/ FLTEP, IPRO, ISTTT, ISKP(4), AHEAD(12), FLIM, AF(3), NDEC
      COMMON /A25/ INS(102), IEW(102), JPH
      COMMON /S25/ ZDOT, CBLANK, IBLANK, RBLANK, LAZT
      COMMON /S26/ IDSTA(501), NMAXX, IB1, KN1, KW1, KS1, NSMAX
C----------------------------------------------------------------------------
C---for variable first layer:
      VC=V(1)*V(2)/SQRT(VSQ(2)-VSQ(1))
   10 PMIN=9999.
      IDXS=0
      NS=0
      DO 20 I=1, NSMAX
      KSMP(I)=0
   20 JDX(I)=0
   25 L=1
      KMES=0
30    CONTINUE
      READ(8, 35, END=300) MSTA(L), PRMK(L), W(L), JTIME, JMIN(L),
     1P(L), S(L), SRMK(L),
     2 WS(L), AMX(L), PRX(L), CALP, CALX(L), RMK(L), DT(L), FMP(L), AZRES(L),
     3 SYM(L), AS, QRMK(L), IPRO, ICARD
35    FORMAT (A4, A4, TL1, F1. 0, TR1, I8, I2, F5. 2, TR7, F5. 2, A4, TL1, F1. 0, TR3,
     1 F4. 0, F3. 2, F4. 1, TR4, F4. 1, A3, F5. 2, F5. 0, TL55, A4, TL18, A1, TR24,
     2 A4, TR27, A1, TL59, A4, TL8, 80A1)
      IF(L .GE.  MMAX) GO TO 47
      IF ((MSTA(L). EQ. MSTAR). OR. (MSTA(L). EQ. MDOL). OR. (MSTA(L). EQ. MCENT)
     1. OR.  (MSTA(L) . EQ.  LAZT)) GO TO 300
      IF (MSTA(L). EQ. MBK) GO TO 350
      KSMP(L)=1
      IF(TP(L).GE.PMIN) GO TO 95
      PMIN=TP(L)
      NEAR=L
      GO TO 95
   89 IF (TP(L) .GE. PMIN) GO TO 90
      PMIN=TP(L)
      NEAR=L
   90 IF (AS .EQ.CBLANK) GO TO 100
C------- S DATA ---------------------------------------------------------
      IDXS=1
      LDX(L)=1
      WS(L)=(4.-WS(L))/4.
      IF (IW(L) .EQ. ISTAR) WS(L)=0.
   95 TS(L)=60.*JMIN(L)+S(L)+DT(L)
  100 L=L+1
      IF (L .LT. MMAX) GO TO 30
      WRITE(9,105)
  105 FORMAT(///," ***** ERROR: PHASE LIST EXCEEDS ARRAY DIMENSION; EXTR
     1A DATA TREATED AS NEXT EARTHQUAKE")
      GO TO 350
  300 M=1
      NR=L-1
      RETURN
  350 M=0
  400 NR=L-1
      RETURN
C-----FOLLOWING CODE ADDED BY SAM STEWART, MAY 30, 1977.
  450 CONTINUE
      WRITE(9,456) ICARD
  456 FORMAT (//," ***** PHASE CARD READ ERROR *****",2x,80A1,//)
      GO TO 30
      END
      SUBROUTINE OUTPUT                                                 738.
C------- OUTPUT HYPOCENTER ---------------------------output----------- 739.
      INTEGER*2 SYM
      INTEGER*4 KDATE, KKYR, KKMO, KKDAY, IBLANK
      CHARACTER*4 RMKO, RMK1, RMK2, RMK3, RMK4, RMK5, AZRES, NSTA, MSTA, LAZT
      CHARACTER*4 FMT1, FMT2, FMT3, FMT4, WRK, PRMK, SRMK
      CHARACTER*4 F1, F2, G1, G2, F4, F5, F6, G3, G4
      CHARACTER*4 MBK, MDOL, MSTAR, DOT, ZDOT, STAR4, MCENT
      CHARACTER*4 IPRO, ISTTT, AHEAD, ISW, IONE, CBLANK, BLANK
      CHARACTER*3 CRMK, RMK
      CHARACTER*1 QUES, ISTAR, CLASS, IW
      CHARACTER*2 Q, QS, QD
      CHARACTER*1 SYM1, SYM2, SYM3, SYMBOL, KS1, KW1, INS, IEW, QRMK
      REAL*8 TIME1, TIME2                                               741.
      REAL LAT, LON, LAT2, LON2, LATEP, LONEP, MAG                      742.
      REAL LATR, LONR, RBLANK
      COMMON /A1/ NSTA(102), DLY(2, 102), FMGC(102), XMGC(102), KLAS(102),    743.
     1       PRR(102), CALR(102), ICAL(102), IS(102)
      COMMON /A2/ LAT(102), LON(102), DELTA(101), DX(101), DY(101), T(101)    745.
      COMMON /A5/ ZTR, XNEAR, XFAR, POS, IQ, KMS, KFM, IPUN, IMAG, IR, QSPA(9, 40)  746.
      COMMON /A6/ NMAX, LMAX, NS, NL, MMAX, NR, FNO, Z, X(4, 101), ZSQ, NRP, DF(101) 747.
      COMMON /A7/ KP, KZ, KOUT, WT(101), Y(4), SE(4), XMEAN(4)          748. 1
      COMMON /A8/ CAL(101), XMAG(101), FMAG(101), NM, AVXM, SDXM, NF, AVFM,   749.
     1       SDFM, MAG, KDX(101), AMX(101), PRX(101), CALX(101), FMP(101)     750.
      COMMON /A10/ ANIN(101), AZ(101), TEMP(101), CA(71), CB(71)        751.
      COMMON /A11/ KDATE, KHR, KMIN, SEC, LAT1, LAT2, LON1, LON2, RMK1, RMK2,  752.
     1       IGAP, DMIN, RMSSQ, ERH, Q, QS, QD, ADJSQ, INST, AVR, AAR, NI, KNST, JHR  753.
      COMMON /A12/ MSTA(101), PRMK(101), W(101), JMIN(101), P(101),     754.
     1       WRK(101), TP(101), DT(101), RMK(101)                       755.
      COMMON /A13/ JDX(102), LDX(101), KEY(101), CLASS(4)               756.
      COMMON /A14/ MBK, MDOL, BLANK, MSTAR, DOT, STAR4, QUES, CRMK, MCENT, ISTAR  757.
      COMMON /A15/ M, L, J, ORG, JAV, PMIN, AZRES(101), NEAR, IDXS, LATEP, LONEP  758.
      COMMON /A16/ KLSS(102), CALS(102), IPRN, ISW
      COMMON /A17/ TIME1, TIME2, LATR, LONR, KTEST, KAZ, KSORT, KSEL, XFN    760.
      COMMON /A18/ S(101), SRMK(101), WS(101), TS(101), NOS, QRMK(101)  761.
      COMMON /A19/ KNO, IELV(102), TEST(15), FLT(2, 102), MNO(102), IW(102)   762.
      COMMON /A21/ KSMP(102), FMO, ONF, B(4), IPH, KF, AVRPS, IEXIT     763.
      COMMON /A22/ F( 9, 9), G(4, 9), H( 9), DEPTH( 9), IONE            764.
      COMMON /A23/ AIN(101), RMS, ADJ, SYM(101)                         765.
      COMMON /A24/ FLTEP, IPRO, ISTTT, ISKP(4), AHEAD(12), FLIM, AF(3), NDEC  766.
      COMMON /A25/ INS(102), IEW(102), JPH                              767.
      COMMON /S25/ ZDOT, CBLANK, IBLANK, RBLANK, LAZT
      DIMENSION FMT1(32), FMT2(24), FMT3(32)
      DIMENSION FMT4(16), DEMP(101), SYMBOL(5)
C=    DATA FMT1/"(1x, ", "i6, a", "1, 2i", "2, f6", ". 2, i", "3, a1", ", f5. ", "2, i4",  769.
C=   1         ", a1, ", "f5. 2", ", a1, ", "f6. 2", ", a1, ", "f6. 2", ", 2i3", ", i4, ",  770.
C=   2         "i2, f", "5. 2, ", "f5. 1", ",    ", "f5. 1", ", 2(1", "x, a1", "), 2a",  771.
C=   2         "1, f5", ". 2, 2", "i3, 2", "f5. 2", ", 2(i", "3, 2f", "5. 1)", ", i2)"/  772.
C=    DATA FMT2/"(I6, ", "1X, 2", "I2, F", "6. 2, ", "I3, A", "1, F5", ". 2, I", "4, A1",  773.
C=   1         ", F5. ", "2, A1", ", F6. ", "2, A1", ",    ", "F6. 2", ", I3, ", "I4, F",  774.
C=   2         "5. 1, ", "F5. 2", ",    ", "F5. 1", ",    ", "F5. 1", ", 3A1", ")    "/  775.
C=    DATA FMT3/"(1X, ", "A4, F", "6. 1, ", "2I4, ", "1X, A", "4, 1X", ", 2I2", ", 4F6",  776.
C=   1         ". 2,  ", "F6. 2", ", A2, ", "F4. 2", ", I4, ", "I3, F", "6. 2, ", "I2,  ",  777.
C=   2         "F4. 1", ", A1, ", "1X, A", "3,   ", "I4,  ", "F4. 1", ", A1, ", "1X, A",  778.
C=   3         "4,  3", "F6. 2", ", A2, ", "F4. 2", ",    ", "F6. 2", ", T6, ", "A1) "/  779.
C=    DATA FMT4/"(A4, ", "3F6. ", "1, 1X", ", A4, ", "2F6. ", "2, F5", ". 1,  ", "F6. 2",  780.
C=   1         ", 1X, ", "A3,  ", "F6. 2", ", I7, ", "    2", "I2, 2", "I4, A", "1)   "/  781.
      DATA SYM1, SYM2, F1, F2, G1, G2/"-", "!", "F6. 2", "F5. 1", "A6  ", "  A5"/   782.
      DATA F4, F5, F6, G3, G4/"F4. 1", "I4,  ", "F4. 2", "A4  ", "A4,  "/   783.
      DATA SYMBOL/" ", "1", "2", "Q", "*"/, KS1, KW1/"S", "W"/          784. 00
C---------------------------------------------------------------------- 785.
      IF ((IPRN. GE. 2) . OR.  (KP. EQ. 1)) CALL XFMAGS                 786.
      LAT1=LATEP/60.                                                    787.
      LAT2=LATEP-60. *LAT1                                              788.
      LON1=LONEP/60.
      LON2=LONEP-60.*LON1
      ADJ=SQRT(ADJSQ)
      RMS=SQRT(RMSSQ)
      JHR=KHR
      OSAVE = ORG
      IF (ORG .GE. 0.) GO TO 5
      ORG=ORG+3600.
      KHR=KHR-1
    5 KMIN=ORG/60.0
      SEC=ORG-60.0*KMIN
      ERH=SQRT(SE(1)**2+SE(2)**2)
      IF(ERH .GT. 999.8) ERH=999.89
      IF( SE(3) .GT. 999.8) SE(3)= 999.89
      NO=FNO
      RMK1=CBLANK
      RMK2=CBLANK
      RMKO=CBLANK
C---- KZ=1 FOR FIXED DEPTH; ONF=0 FOR ORIGIN TIME BASED ON SMP"S
      IF (ONF .EQ. 0.) RMKO=STAR4
      IF (KZ .EQ. 1) RMK2=STAR4
      J=0
      DO 10 I=1,NRP
      DXI=DX(I)
      DYI=DY(I)
      IF ((DXI.EQ.0.).AND.(DYI.EQ.0.)) GO TO 6
      JI=KDX(I)
      IF (INS(JI) .EQ. KS1) DYI=-DYI
      IF (IEW(JI) .EQ. KW1) DXI=-DXI
      AZ(I)=AMOD(ATAN2(DXI,DYI)*57.29578 + 360., 360.)
      GO TO 7
    6 AZ(I)= 999.
    7 CONTINUE
      AIN(I)=ASIN(ANIN(I))*57.29578
      IF (AIN(I) .LT. 0.) AIN(I)=180.+AIN(I)
      AIN(I)=180.-AIN(I)
      SWT=0.
      IF (LDX(I) .EQ. 0.) GO TO 8
      KK=LDX(I)
      SWT=WT(KK)
    8 IF ((WT(I).EQ.0.).AND.(SWT.EQ.0.)) GO TO 10
      J=J+1
      TEMP(J)=AZ(I)
   10 CONTINUE
      CALL SORT(TEMP,KEY,J)
      GAP=TEMP(1)+360.-TEMP(J)
      DO 20 I=2,J
      DTEMP=TEMP(I)-TEMP(I-1)
      IF (DTEMP .GT. GAP) GAP=DTEMP
   20 CONTINUE
      IGAP=GAP+0.5
      DO 25 I=1,NRP
   25 DEMP(I)=DELTA(I)
      CALL SORT(DEMP,KEY,NRP)
      DO 27 I=1,NRP
      K=KEY(I)
      SWT=0.
      IF (LDX(K) .EQ. 0.) GO TO 26
      KK=LDX(K)
      SWT=WT(KK)
   26 IF ((WT(K).GT.0.).OR.(SWT.GT.0.)) GO TO 28
   27 CONTINUE
   28 DMIN=DEMP(I)
      IDMIN=DMIN+0.5
      OFD=Z
      TFD=2.*Z
      IF (OFD .LT.  5.) OFD=5.
      IF (TFD .LT.  10.) TFD=10.
      JS=4
      IF ((RMS.LT.0.50).AND.(ERH.LE.5.0)) JS=3
      IF ((RMS.LT.0.30).AND.(ERH.LE.2.5).AND.(SE(3).LE.5.0)) JS=2
      IF ((RMS.LT.0.15).AND.(ERH.LE.1.0).AND.(SE(3).LE.2.0)) JS=1
      JD=4
      IF (NO .LT.  6) GO TO 30
      IF ((GAP.LE.180.).AND.(DMIN.LE.50.)) JD=3
      IF ((GAP.LE.135.).AND.(DMIN.LE.TFD)) JD=2
      IF ((GAP.LE.  90.).AND.(DMIN.LE.OFD)) JD=1
   30 JAV=(JS+JD+1)/2
      Q=CLASS(JAV)
      QS=CLASS(JS)
      QD=CLASS(JD)
   50 TIME2=SEC+1.D+02*KMIN+1.D+04*KHR+1.D+06*KDATE
      IF(IPRN .EQ.  0) GO TO 52
      IF(NI .NE.  1) GO TO 60
      IF(NDEC .GE.  1) GO TO 60
      IF (JPH .EQ.  1) GO TO 60
   52 KKYR=KDATE/10000
      KKMO=(KDATE-KKYR*10000)/100
      KKDAY=(KDATE-KKYR*10000-KKMO*100)
      JPH=1
      IF(KSEL) 501,501,505
  501 WRITE(9,502)
  502 FORMAT(///)
      GO TO 535
  505 WRITE(9,506)
  506 FORMAT('\f')
   51 WRITE(9,53) AHEAD,KKYR,KKMO,KKDAY,KHR,KMIN
   53 FORMAT(/,30X,12A4,T112,I2,"/",I2,"/",I2,4X,I2,":",I2)
  535 IF( TIME2 - TIME1 .GT.  -20.)GO TO 60
      WRITE(9,54)
   54 FORMAT(" ***** FOLLOWING EVENT IS OUT OF ORDER *****")
   60 IF ((KP.EQ.1) .AND.  (IPRN.EQ.0)) GO TO 67
      IF (IPH .EQ.  1) GO TO 62
      WRITE(9,61) INS(1),IEW(1)
   61 FORMAT(/,59X,"  ADJUSTMENTS (KM)  PARTIAL F-VALUES  STANDARD ERROR
     1S  ADJUSTMENTS TAKEN",/,"  I  ORIG  LAT ",A1
     2,"    LONG ",A1,          "   DEPTH  DM  RMS AVRPS SKD   CF    DLA
     3T  DLON     DZ  DLAT  DLON     DZ  DLAT  DLON     DZ  DLAT  DLON     D
     4Z")
      IF (IPRN .EQ.  1) IPH=1
   62 WRITE(9,63) NI,SEC,LAT1,LAT2,LON1,LON2,Z,RMK2,IDMIN,RMS,AVRPS,
     1 QS,KF,QD,FLIM,B(2),B(1),B(3),AF(2),AF(1),AF(3),SE(2),SE(1),
     2 SE(3),Y(2),Y(1),Y(3)
   63 FORMAT(I3,F6.2,I3,"-",F5.2,I4,"-",F5.2,F6.2,A1,I3,F5.2,F6.2,
     1 1X,A1,I1,A1,13F6.2)
      IF (KP .EQ.  0) GO TO 100
   67 JNST=KNST*10+INST
      IF (NM .EQ.  0) AVXM=0.
      IF (NF .EQ.  0) AVFM=0.
      FMT1(14)=F1
      FMT1(19)=F2
      FMT1(21)=F2
      FMT2(14)=F1
      FMT2(20)=F2
      FMT2(22)=F2
      IF(MAG.NE.RBLANK) GOTO 68
      FMT1(14)=G1
      FMT2(14)=G1
```

```
68     IF(SE(3).NE. 0.) GOTO 70
       SE(3)=RBLANK
       FMT1(21)=G2
       FMT2(22)=G2
70     IF(ERH.NE. 0.)   GOTO 72
       ERH=RBLANK
       FMT1(19)=G2
       FMT2(20)=G2
   72 WRITE(9,75) INS(1),IEW(1)
   75 FORMAT(//, "  DATE    ORIGIN    LAT ",A1,"    LONG ",A1,"    DEPTH
      1    MAG NO DM GAP M  RMS  ERH  ERZ Q SQD  ADJ IN NR  AVR  AAR NM AV
      2XM SDXM NF AVFM SDFM I")
3333   FORMAT(1x,i6,a1,2i2,f6.2,i3,a1,f5.2,i4,a1,f5.2,a1,f6.2,a1,f5.2,
      1 2i3,i4,i2,f5.2,f5.1,f5.1,2(1x,a1),2a1,f5.2,2i3,2f5.2,
      2 2(i3,2f5.1),i2)
80     CONTINUE
       WRITE(9,3333)KDATE,RMKO,KHR,KMIN,SEC,LAT1,SYM1,LAT2,LON1,SYM1,LON2
      1,RMK1,Z,RMK2,MAG,NO,IDMIN,IGAP,KNO,RMS,ERH,SE(3),Q,QS,SYM2,QD,ADJ
      2,JNST,NR,AVR,AAR,NM,AVXM,SDXM,NF,AVFM,SDFM,NI
       IF (IPUN .EQ. 0) GO TO 100
       IF ((QRMK(1).NE.SYMBOL(4)).AND.(QRMK(1).NE.SYMBOL(5)))
      1QRMK(1)=SYMBOL(1)
       SYM3=SYMBOL(KNO+1)
       WRITE(4,3020) KDATE,KHR,KMIN,SEC,LAT1,SYM1,LAT2,LON1,SYM1,LON2
      1,RMK1,Z,RMK2,MAG,NO,IGAP,DMIN,RMS,ERH,SE(3),QRMK(1),Q,SYM3
       WRITE(2,3020) KDATE,KHR,KMIN,SEC,LAT1,SYM1,LAT2,LON1,SYM1,LON2
      1,RMK1,Z,RMK2,MAG,NO,IGAP,DMIN,RMS,ERH,SE(3),QRMK(1),Q,SYM3
3020   FORMAT(i6,1x,2i2,f6.2,i3,a1,f5.2,i4,a1,f5.2,a1,f6.2,a1,
      1 f6.2,i3,i4,f5.1,f5.2,f5.1,f5.1,3a1)
  100 IF (KP .EQ. 1) GO TO 105
       IF(IPRN .LE. 1) GO TO 300
  105 WRITE(9,110)
  110 FORMAT(/," STN  DIST AZM AIN PRMK HRMN P-SEC TPOBS TPCAL DLY/H1 P
      1-RES P-WT AMX PRX CALX K XMAG RMK FMP FMAG SRMK S-SEC TSOBS S-RES
      2 S-WT   DT")
       DO 200 I=1,NRP
       K=I
       IF (KSORT .EQ. 1) K=KEY(I)
       KJI=KDX(K)
       TPK=TP(K)-ORG
       IF (TPK .LT. 0.) TPK=TPK+3600.
       FMT3(10)=F1
       IF ((AZRES(K).NE.DOT).AND.(AZRES(K).NE.CBLANK).AND.
      1(AZRES(K).NE.ZDOT)) GO TO 114
       X(4,K)=RBLANK
       FMT3(10)=G1
  114 RMK3=BLANK
       IF (XMAG(K) .EQ.RBLANK) GO TO 115
       IF (ABS(XMAG(K)-AVXM) .GE. 0.5) RMK3=STAR4
  115 RMK4=BLANK
       IF (FMAG(K) .EQ.RBLANK) GO TO 130
       IF (ABS(FMAG(K)-AVFM) .GE. 0.5) RMK4=STAR4
  130 FMT3(17)=F4
       FMT3(21)=F5
       FMT3(22)=F4
       FMT4(8)=F1
       FMT4(11)=F1
       IF (XMAG(K) .NE.RBLANK) GO TO 160
       FMT3(17)=G3
       FMT4(8)=G1
  160 IF (FMAG(K) .NE.RBLANK) GO TO 162
       IFMP=IBLANK
       FMT3(21)=G4
       FMT3(22)=G3
      FMT4(11)=G1
  162 FMT3(26)=F1
      FMT3(28)=F6
      IAZ=AZ(K)+0.5
      IAIN=AIN(K)+0.5
      IAMX=AMX(K)
      IPRX=100.*PRX(K)+0.5
      IFMP=IBLANK
      IF(FMP(K).NE.RBLANK) IFMP=FMP(K)
      IF (LDX(K) .NE. 0) GO TO 163
C-----CHECK FOR SMP DATA
      IF (KSMP(K) .EQ. 0) GO TO 165
      SRES=X(4,K)
      RMK5=BLANK
      SWT=11111.
      TSK=S(K)-P(K)
      GO TO 168
  163 KK=LDX(K)
      SRES=X(4,KK)
      RMK5=WRK(KK)
      SWT=WT(KK)
  164 TSK=TS(K)-ORG
      GO TO 168
  165 S(K)=RBLANK
      TSK=RBLANK
      SRES=RBLANK
      RMK5=BLANK
      SWT=RBLANK
      FMT3(26)=G1
      FMT3(28)=G3
  168 FMT3(30)=F1
      DLYK=DLY(KNO,KJI)
      IF (ISW .EQ. IONE) DLYK=FLT(KNO,KJI)
      DTK=DT(K)
      IF (DTK .NE. 0.) GO TO 170
      DTK=RBLANK
      FMT3(30)=G1
      IF(FMAG(K) .EQ.RBLANK) IFMP=IBLANK
C== following statement clears up the station output, but i dont get
       IF(IFMP.EQ.8224) IFMP=0
C== where the integer value comes from.  sws  sept.79.
3030  FORMAT(1x,a4,a1,f5.1,2i4,1x,a4,1x,2i2,4f6.2,f6.2,a2,f4.2,i4,i3,
     1 f6.2,i2,f4.1,a1,1x,a3,i4,f4.1,a1,1x,a4,3f6.2,a2,f4.2,f5.2)
  170 WRITE(9,3030) MSTA(K),IW(KJI),DELTA(K),IAZ,IAIN,PRMK(K),JHR,
     1 JMIN(K),P(K),
     1        TPK,T(K),DLYK,X(4,K),WRK(K),WT(K),IAMX,IPRX,CAL(K)
     2,KLAS(KJI),XMAG(K),RMK3,RMK(K),IFMP,FMAG(K),RMK4,SRMK(K),S(K)
     3,TSK,SRES,RMK5,SWT,DTK
      IF (IPUN .NE. 2) GO TO 200
      ISEC = 100.*SEC + 0.5
      WRITE(4,3040) MSTA(K),DELTA(K),AZ(K),AIN(K),PRMK(K),TPK,X(4,K)
     1,WT(K),XMAG(K),RMK(K),FMAG(K),KDATE,KHR,KMIN,ISEC,KJI,SYM3
3040  FORMAT(a4,3f6.1,1x,a4,2f6.2,f5.1,f6.2,1x,a3,f6.2,
     1 i7,2i2,2i4,a1)
  200 CONTINUE
      WRITE(9,3032)
 3032 FORMAT(1X,' $$$')
      IF (IPUN .NE. 2) GO TO 300
      WRITE(4,205)
  205 FORMAT(" $$$")
  300 KHR = JHR
      ORG = OSAVE
      RETURN
      END
```

single

```
      SUBROUTINE SINGLE
C-------- SOLUTION FOR A SINGLE EARTHQUAKE ----------------------------
      INTEGER*4 KDATE
      INTEGER*2 SYM
      CHARACTER*4 MBK, MDOL, BLANK, MSTAR, DOT, STAR4, MCENT
      CHARACTER*4 NSTA, MSTA, RMK1, RMK2, PRMK, SRMK, AHEAD
      CHARACTER*4 WRK, AZRES, CHECK, IPRO, ISTTT, ISW, IONE
      CHARACTER*3 CRMK, RMK
      CHARACTER*1 QUES, ISTAR, IW, CLASS, INS, IEW
      CHARACTER*2 Q, QS, QD
      REAL*8 TIME1, TIME2
      REAL LATRT,  LONRT,  LATSV,  LONSV
      REAL LAT, LON, LAT2, LON2, LATEP, LONEP, MAG, LATR, LONR
      COMMON /A1/ NSTA(102), DLY(2, 102), FMGC(102), XMGC(102), KLAS(102),
     1       PRR(102), CALR(102), ICAL(102), IS(102)
      COMMON /A2/ LAT(102), LON(102), DELTA(101), DX(101), DY(101), T(101)
      COMMON /A3/ NRES(2, 102), NXM(102), NFM(102), SR(2, 102), SRSQ(2, 102),
     1       SRWT(2, 102), SXM(102), SXMSQ(102), SFM(102), SFMSQ(102), QNO(4)
      COMMON /A5/ ZTR, XNEAR, XFAR, POS, IQ, KMS, KFM, IPUN, IMAG, IR, QSPA(9, 40)
      COMMON /A6/ NMAX, LMAX, NS, NL, MMAX, NR, FNO, Z, X(4, 101), ZSQ, NRP, DF(101)
      COMMON /A7/ KP, KZ, KOUT, WT(101), Y(4), SE(4), XMEAN(4)
      COMMON /A8/ CAL(101), XMAG(101), FMAG(101), NM, AVXM, SDXM, NF, AVFM,
     1       SDFM, MAG, KDX(101), AMX(101), PRX(101), CALX(101), FMP(101)
      COMMON /A10/ ANIN(101), AZ(101), TEMP(101), CA(71), CB(71)
      COMMON /A11/ KDATE, KHR, KMIN, SEC, LAT1, LAT2, LON1, LON2, RMK1, RMK2,
     1       IGAP, DMIN, RMSSQ, ERH, Q, QS, QD, ADJSQ, INST, AVR, AAR, NI, KNST, JHR
      COMMON /A12/ MSTA(101), PRMK(101), W(101), JMIN(101), P(101),
     1       WRK(101), TP(101), DT(101), RMK(101)
      COMMON /A13/ JDX(102), LDX(101), KEY(101), CLASS(4)
      COMMON /A14/ MBK, MDOL, BLANK, MSTAR, DOT, STAR4, QUES, CRMK, MCENT, ISTAR
      COMMON /A15/ M, L, J, ORG, JAV, PMIN, AZRES(101), NEAR, IDXS, LATEP, LONEP
      COMMON /A16/ KLSS(102), CALS(102), IPRN, ISW
      COMMON /A17/ TIME1, TIME2, LATR, LONR, KTEST, KAZ, KSORT, KSEL, XFN
      COMMON /A18/ S(101), SRMK(101), WS(101), TS(101), NOS, QRMK(101)
      COMMON /A19/ KNO, IELV(102), TEST(15), FLT(2, 102), MNO(102), IW(102)
      COMMON /A20/ V( 9), D( 9), VSQ( 9), THK( 9), TID( 9,  9), DID( 9,  9)
      COMMON /A21/ KSMP(102), FMO, ONF, B(4), IPH, KF, AVRPS, IEXIT
      COMMON /A22/ F( 9,  9), G(4,  9), H( 9), DEPTH( 9), IONE
      COMMON /A23/ AIN(101), RMS, ADJ, SYM(101)
      COMMON /A24/ FLTEP, IPRO, ISTTT, ISKP(4), AHEAD(12), FLIM, AF(3), NDEC
      COMMON /A25/ INS(102), IEW(102), JPH
      DIMENSION SUM(5), WF(41), ALZ(10), LA(10), LO(10)
      DATA WF/.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.94, 0.94, 0.94, 0.93,
     1       0.92, 0.92, 0.91, 0.90, 0.88, 0.87, 0.85, 0.83, 0.80, 0.77,
     2       0.73, 0.69, 0.64, 0.59, 0.53, 0.47, 0.41, 0.34, 0.28, 0.23,
     3       0.18, 0.14, 0.11, 0.08, 0.06, 0.04, 0.03, 0.02, 0.01, 0.01, 0. /
      DATA LA/1, 1, 1, 1, 0, 0, -1, -1, -1, -1/,
     1     LO/+1, -1, +1, -1, 0, 0, +1, -1, +1, -1/,
     2     ALZ/-1.0, -1.0, +1.0, +1.0, -1.732, +1.732, -1.0, -1.0, +1.0, +1.0/
C---------------------------------------------------------------------
      AVRPS = 0.0
      IEXIT=0
      LATRT=0.
      ZRES=P(NR+1)
      KNST=JMIN(NR+1)/10
      INST=JMIN(NR+1)-KNST*10
      NRP=NR
   30 IF (IDXS .EQ. 0) GO TO 80
C------- TREAT S DATA BY AUGMENTING P DATA ----------------------------
      NOS=0
      DO 65 I=1, NRP
      IF (LDX(I) .EQ. 0) GO TO 65
      NOS=NOS+1
      NRS=NRP+NOS
      TP(NRS)=TS(I)
      W(NRS)=WS(I)
      KSMP(NRS)=0
      IF ((KNST.NE.1).AND.(KNST.NE.6)) W(NRS)=0.
      KDX(NRS)=KDX(I)
      LDX(I)=NRS
      WRK(NRS)=BLANK
   65 CONTINUE
      NR=NRP+NOS
C------- INITIALIZE TRIAL HYPOCENTER -----------------------------------
   80 K=KDX(NEAR)
      SVY1 = 0.0
      SVY2 = 0.0
      SVY3 = 0.0
      ERLMT = 0.
      DO 25 I = 1,3
      ISKP(I)=0
   25 CONTINUE
      IF (INST .NE. 9) GO TO 90
      READ(8,85) ORG1,ORG2,LAT1,LAT2,LON1,LON2,Z
   85 FORMAT(F5.0,F5.2,I5,F5.2,I5,2F5.2)
      ORG=60.*ORG1+ORG2
      LATEP=60.*LAT1+LAT2
      LONEP=60.*LON1+LON2
      GO TO 105
   90 IF (NR .GE. 3) GO TO 100
   96 WRITE(9,97)
   97 FORMAT(///," ***** INSUFFICIENT DATA FOR LOCATING THIS QUAKE:")
      IF( NRP .EQ. 0 ) NRP = 1
      DO 98 L=1,NRP
   98 WRITE(9,99) MSTA(L),PRMK(L),KDATE,KHR,JMIN(L),P(L),S(L)
   99 FORMAT(5X,2A4,1X,I6,2I2,F5.2,7X,F5.2)
      IEXIT=1
      IF (NRP .EQ. 1) RETURN
      GO TO 575
  100 Z=ZTR
      IF (AZRES(NRP+1).NE. BLANK) Z=ZRES
      ORG=PMIN-Z/5.-1.
      IF(LATRT.EQ.0.) GO TO 102
      LATEP=LATRT
      LONEP=LONRT
      GO TO 105
  102 IF (LATR .EQ. 0.) GO TO 104
      LATEP=LATR
      LONEP=LONR
      GO TO 105
  104 LATEP=LAT(K)+0.1
      LONEP=LON(K)+0.1
  105 ADJSQ=0.
      IPH=0
      NDEC=0
      PRMSSQ=100000.
      KNO=1
      IF (ISW .EQ. IONE) KNO=MNO(K)
      IF(ISW .EQ. IONE) FLTEP=FLT(KNO,K)
      NIMAX=TEST(11)+.0001
C------- GEIGER"S ITERATION TO FIND HYPOCENTRAL ADJUSTMENTS ------------
  109 NI = 1
      IF (INST .EQ. 9) NI=NIMAX
  111 IF(ERLMT .EQ. 0.) GO TO 110
      LATEP = LATSV + LA(NA)*DELAT
      LONEP = LONSV + LO(NA)*DELON
      Z = ZSV + ALZ(NA)*DEZ
      IF(Z .LT. 0.) Z=0.
  110 FMO=0.
      FNO=0.
      DO 112 I=1,5
  112 SUM(I)=0.
C------- CALCULATE EPICENTRAL DISTANCE BY RICHTER"S METHOD -------------
      DO 120 I=1,NR
      JI=KDX(I)
      AVL=(LAT(JI)+LATEP)/120.
      M1=AVL+1.5
      M2=AVL*10.+1.5
      DX(I)=(LON(JI)-LONEP)*CA(M1)*COS((M2-1)*.0017453)
      DY(I)=(LAT(JI)-LATEP)*CB(M1)
      DELTA(I)=SQRT(DX(I)**2+DY(I)**2)+0.000001
      WT(I)=W(I)
      IF (NI .LE. 1) GO TO 115
C------- DISTANCE WEIGHTING -------------------------------------------
      IF (DELTA(I) .LE. XNEAR) GO TO 115
      WT(I)=W(I)*(XFAR-DELTA(I))/XFN
      IF (WT(I) .LT. 0.005) WT(I)=0.
  115 IF (WT(I) .EQ. 0.) GO TO 120
      IF (KSMP(I) .EQ. 1) FMO=FMO+1.
      FNO=FNO+1.
      SUM(4)=SUM(4)+WT(I)
  120 CONTINUE
      IF (FNO .LT. 3.) GO TO 96
      AVWT=SUM(4)/FNO
C------- NORMALIZE DISTANCE WEIGHTS ------------------------------------
      SUM(4)=0.0
      DO 122 I=1,NR
  122 WT(I)=WT(I)/AVWT
      IF ((NI.LE.2).OR.(KAZ.EQ.0)) GO TO 130
C------- AZIMUTHAL WEIGHTING -------------------------------------------
C------- COMPUTE TRAVEL TIMES & DERIVATIVES ----------------------------
  130 ZSQ=Z**2
      CALL TRVDRV
      FDLY=1.
      IF (ISW .EQ. IONE) FDLY=0.
C------- CALCULATE TRAVEL TIME RESIDUALS X(4,I) & MODIFY THE DERIV"S ---
  140 DO 150 I=1,NR
      JI=KDX(I)
      IF (I .LE. NRP) GO TO 145
C------- S PHASE DATA --------------------------------------------------
      T(I)=POS*T(I)
      X(1,I)=POS*X(1,I)
      X(2,I)=POS*X(2,I)
      X(3,I)=POS*X(3,I)
      X(4,I)=TP(I)-T(I)-ORG-POS*DLY(KNO,JI)*FDLY
      GO TO 150
  145 IF (KSMP(I) .EQ. 0) GO TO 146
C------- S-P DATA ------------------------------------------------------
      X(1,I)=(POS-1.)*X(1,I)
      X(2,I)=(POS-1.)*X(2,I)
      X(3,I)=(POS-1.)*X(3,I)
      X(4,I)=TS(I)-TP(I)-(POS-1.)*(DLY(KNO,JI)*FDLY+T(I))
      GO TO 150
C------- P TRAVEL TIME RESIDUAL ----------------------------------------
  146 X(4,I)=TP(I)-T(I)-ORG-DLY(KNO,JI)*FDLY
  150 CONTINUE
C------- COMPUTE AVR, AAR, RMSSQ, & SDR --------------------------------
      ONF=0.0
      DO 152 I=1,NR
      ONF = ONF + WT(I)*(1-KSMP(I))
      XWT = X(4,I)*WT(I)
      SUM(1)=SUM(1)+XWT
      SUM(2)=SUM(2)+ABS(XWT)
      SUM(3)=SUM(3)+X(4,I)*XWT
      SUM(5)=SUM(5)+XWT*(1-KSMP(I))
  152 CONTINUE
      IF(FNO .GT. FMO) AVRPS=SUM(5)/(ONF)
      AVR=SUM(1)/FNO
      AAR=SUM(2)/FNO
      RMSSQ=SUM(3)/FNO
      SDR=SQRT(ABS(RMSSQ-AVR**2))
      DO 153 I=1,5
      SUM(I)= 0.0
  153 CONTINUE
      IF (RMSSQ .GE. TEST(1)) GO TO 154
      IF(ERLMT .EQ. 1.) GO TO 167
      IF(INST.EQ.9) GO TO 501
      IF(NI .GE. 2) GO TO 167
      GO TO 165
C------- JEFFREYS" WEIGHTING --------------------------- ---------------
  154 FMO=0.
      FNO=0.
      DO 160 I=1,NR
      WRK(I)=BLANK
      IF (WT(I) .EQ. 0.) GO TO 160
      K=10.*ABS(X(4,I)-AVR)/SDR+1.5
      IF (K .GT. 41) K=41
      WT(I)=WT(I)*WF(K)
      IF (K .GT. 30) WRK(I)=STAR4
      IF (WT(I) .LT. 0.005) WT(I)=0.
      IF (WT(I) .EQ. 0.) GO TO 160
      IF (KSMP(I) .EQ. 1) FMO=FMO+1.
      FNO=FNO+1.
      SUM(4)=SUM(4)+WT(I)
  160 CONTINUE
      IF (FNO .LT. 3.) GO TO 96
      AVWT=SUM(4)/FNO
      SUM(4)=0.0
      ONF=0.0
      DO 164 I=1,NR
      WT(I)=WT(I)/AVWT
      ONF = ONF + WT(I)*(1-KSMP(I))
      XWT=X(4,I)*WT(I)
      SUM(5)=SUM(5)+XWT*(1-KSMP(I))
  164 CONTINUE
C------- RECALCULATE AVRPS ---------------------------------------------
      IF(ERLMT .EQ. 1.) GO TO 163
      IF(INST .NE. 9) GO TO 163
      AVRPS = 0.0
      IF(FNO .NE. FMO) AVRPS = SUM(5)/ONF
      GO TO 501
  163 IF(FNO.EQ.FMO) AVRPS=0.0
      IF(FNO.EQ.FMO) GO TO 167
      AVRPS=SUM(5)/(ONF)
      SUM(5)=0.0
      IF(ERLMT .EQ. 1.) GO TO 167
C------- RESET FIRST ORIGIN TIME ---------------------------------------
      IF(NI.GE. 2) GO TO 167
  165 ORG=ORG+AVRPS
      DO 166 I=1,NR
      IF(KSMP(I) .EQ. 0) X(4,I)=X(4,I)-AVRPS
      XWT=WT(I)*X(4,I)
      SUM(5)=SUM(5)+XWT*(1 - KSMP(I))
      SUM(2)=SUM(2)+ABS(XWT)
      SUM(3)=SUM(3)+X(4,I)*XWT
  166 CONTINUE
```

```
      IF(FNO .GT. FMO) AVRPS=SUM(5)/(ONF)
      AAR=SUM(2)/FNO
      RMSSQ = SUM(3)/FNO
      GO TO 169
C------- FOR NI>1, COMPUTE AAR, & RMSSQ AS IF AVRPS=0. -----------------
  167 DO 168 I=1,NR
      XWT=WT(I)*(X(4,I)-AVRPS*(1-KSMP(I)))
      SUM(2)=SUM(2)+ABS(XWT)
      SUM(3)=SUM(3)+(X(4,I)-AVRPS*(1-KSMP(I)))*XWT
  168 CONTINUE
      AAR=SUM(2)/FNO
      RMSSQ=SUM(3)/FNO
      IF(ERLMT .EQ. 0.) GO TO 169
C------- OUTPUT RMS ERROR OF AUXILIARY POINTS --------------------------
      L = LATEP/60.
      ALA = LATEP - 60.*L
      L = LONEP/60.
      ALO = LONEP - 60.*L
      RMSX= SQRT(RMSSQ)
      DRMS = RMSX - RMSSV
      GO TO (1,2,3,4,5,6,1,2,3,4), NA
    1 WRITE(9,801) ALA,ALO,Z,AVRPS,RMSX,DRMS
  801 FORMAT(5F10.2,10X,F6.2)
      GO TO 174
    2 WRITE(9,802) ALA,ALO,Z,AVRPS,RMSX,DRMS
  802 FORMAT(5F10.2,28X,F6.2)
      GO TO 174
    3 WRITE(9,803) ALA,ALO,Z,AVRPS,RMSX,DRMS
  803 FORMAT(5F10.2,13X,"(",F6.2,")")
      GO TO 174
    4 WRITE(9,804) ALA,ALO,Z,AVRPS,RMSX,DRMS
  804 FORMAT(5F10.2,31X,"(",F6.2,")")
      IF(NA .EQ. 10) GO TO 550
      GO TO 174
    5 WRITE(9,805) ALA,ALO,Z,AVRPS,RMSX,DRMS
  805 FORMAT(/5F10.2,19X,F6.2)
      WRITE(9,807) RMSSV
  807 FORMAT(40X,F10.2,23X,"0.00")
      GO TO 174
    6 WRITE(9,806) ALA,ALO,Z,AVRPS,RMSX,DRMS
  806 FORMAT(5F10.2,22X,"(",F6.2,")"/)
  174 NA = NA + 1
      GO TO 111
C------- CHECK IF SOLUTION IS BETTER THAN PREVIOUS ONE -----------------
  169 IF((NI .EQ. 1) .AND. (NDEC .EQ. 0)) GO TO 170
      IF(PRMSSQ.GE.RMSSQ) GO TO 170
      NDEC = NDEC +1
      IF(NDEC .GT. 1) GO TO 175
      DO 177 I= 1,3
      B(I) = 0.0
      AF(I)=-1.0
      SE(I) = 0.0
  177 CONTINUE
      NI = NI -1
      BM1=Y(1)
      BM2=Y(2)
      BM3=Y(3)
      BMAX = ABS(Y(1))
      IIMAX = 1
      DO 176 I = 2,3
      IF(ABS(Y(I)).LE.BMAX) GO TO 176
      BMAX = ABS(Y(I))
      IIMAX = I
  176 CONTINUE
```

```
      ISKP(IIMAX)=1
      Y(1)=-BM1/5.
      Y(2)=-BM2/5.
      Y(3)=-BM3/5.
      Y(4)=-Y(1)*XMEAN(1)-Y(2)*XMEAN(2)-Y(3)*XMEAN(3)
      XADJSQ=Y(1)**2+Y(2)**2+Y(3)**2
      KP=0
      IF(XADJSQ .LT. 4.*TEST(4)/25.) GO TO 170
  175 IF(NDEC .EQ. 5) GO TO 170
      GO TO 325
C------- STEPWISE MULTIPLE REGRESSION ANALYSIS OF TRAVEL TIME RESIDUALS-
  170 IF(NDEC .GE. 1) NI = NI + 1
      IF (INST.EQ.1) GO TO 250
      IF(ISKP(3) .EQ. 1) GO TO 250
      IF (INST .EQ. 9) GO TO 501
      IF ((FNO.EQ.3) .AND. (FMO.LT.3)) GO TO 250
C---- FREE SOLUTION
  200 KZ=0
      KF=0
      CALL SWMREG
C------- AVOID CORRECTING DEPTH IF HORIZONTAL CHANGE IS LARGE ----------
      IF (Y(1)**2+Y(2)**2 .LT. TEST(2)) GO TO 300
C---- FIXED DEPTH SOLUTION
  250 KZ=1
      KF=0
      CALL SWMREG
C------- LIMIT FOCAL DEPTH CHANGE & AVOID HYPOCENTER IN THE AIR --------
  300 DO 275 I= 1,3
      ISKP(I)=0
  275 CONTINUE
      OLDY1=Y(1)
      OLDY2=Y(2)
      OLDY3=Y(3)
      ABSY1=ABS(Y(1))
      ABSY2=ABS(Y(2))
      ABSY3=ABS(Y(3))
      IF(ABSY1.GT.ABSY2) GO TO 305
      ABSGR=ABSY2
      GO TO 308
  305 ABSGR=ABSY1
  308 IF(ABSY3.LE.TEST(5)) GO TO 310
      I=ABSY3/TEST(5)
      Y(3)=Y(3)/(I+1)
  310 IF((Z+Y(3)).GT. 0.0) GO TO 315
      Y(3)=-Z*TEST(12)+.000001
      ISKP(3) = 1
C------- LIMIT HORIZONTAL ADJUSTMENT OF EPICENTER ----------------------
  315 IF(ABSGR.LE.TEST(10)) GO TO 320
      I=ABSGR/TEST(10)
      Y(1)=Y(1)/(I+1)
      Y(2)=Y(2)/(I+1)
  320 Y(4)=Y(4)-(Y(3)-OLDY3)*XMEAN(3)-(Y(1)-OLDY1)*XMEAN(1)
     1 -(Y(2)-OLDY2)*XMEAN(2)
      XADJSQ=Y(1)**2+Y(2)**2+Y(3)**2
      KP=0
      NDEC=0
      JPH=0
  325 IF (IPRN .GE. 1) CALL OUTPUT
      IF(NDEC .GE. 1) GO TO 330
C------- TERMINATE ITERATION IF HYPOCENTER ADJUSTMENT < TEST(4) --------
      IF (XADJSQ .LT. TEST(4)) GO TO 500
  330 IF(NI .EQ. NIMAX) GO TO 500
C------- ADJUST HYPOCENTER ---------------------------------------------
  350 AVL=LATEP/60.
  375 M1=AVL+1.5
      M2=AVL*10.+1.5
      DY1 =Y(1)/(CA(M1)*COS((M2-1)*.0017453))
      DY2 =Y(2)/CB(M1)
      LATEP=LATEP+DY2
      LONEP=LONEP+DY1
      Z=Z+Y(3)
      ORG=ORG+Y(4)
      SVY1 = Y(1)
      SVY2 = Y(2)
      SVY3 = Y(3)
      ADJSQ=XADJSQ
      IF(NDEC .EQ. 0) PRMSSQ=RMSSQ
      IF(NDEC.GE.1) GO TO 110
  400 NI = NI + 1
      IF(NI .LE. NIMAX) GO TO 111
C------- RESET ORIGIN TIME ----------------------------------------------
  500 ORG=ORG+XMEAN(4)
      GO TO 502
  501 XMEAN(4)=0.0
  502 DO 505 I=1,5
  505 SUM(I)=0.0
      SUMM = 0.0
      DO 510 I=1,NR
      IF (KSMP(I) .EQ. 0) X(4,I)=X(4,I)-XMEAN(4)
      IF (WT(I) .EQ. 0.) GO TO 510
      IF(INST .NE. 9) GO TO 509
      XWTS=WT(I)*(X(4,I)**2)
      IF(KSMP(I) .EQ. 0) XWTS=WT(I)*((X(4,I)-AVRPS)**2)
      SUMM = SUMM + XWTS
  509 XWT=X(4,I)*WT(I)
      SUM(1)=SUM(1)+XWT
      SUM(2)=SUM(2)+ABS(XWT)
      SUM(3)=SUM(3)+X(4,I)*XWT
      SUM(5)=SUM(5)+XWT*(1-KSMP(I))
  510 CONTINUE
      RM9SV = SUMM/FNO
      AVR=SUM(1)/FNO
      AVRPS = 0.0
      IF(FNO .GT. FMO) AVRPS=SUM(5)/ONF
      AAR=SUM(2)/FNO
      RMSSQ=SUM(3)/FNO
C------- COMPUTE ERROR ESTIMATES BY SOLVING FULL NORMAL EQUATION -------
  520 KF=2
      KP=1
      KZ=0
      CALL SWMREG
      DO 521 I =1,3
  521 Y(I)=0.0
      IF(INST.EQ.1) KZ = 1
      CALL OUTPUT
      QNO(JAV)=QNO(JAV)+1.
      IF (JAV .GT. IQ) GO TO 523
C------- COMPUTE SUMMARY OF TRAVEL TIME RESIDUALS ----------------------
      DO 522 I=1,NRP
      IF ((WT(I).EQ.0.) .OR. (KSMP(I).EQ.1))  GO TO 522
      JI=KDX(I)
      NRES(KNO,JI)=NRES(KNO,JI)+1
      SR(KNO,JI)=SR(KNO,JI)+X(4,I)*WT(I)
      SRSQ(KNO,JI)=SRSQ(KNO,JI)+X(4,I)**2*WT(I)
      SRWT(KNO,JI)=SRWT(KNO,JI)+WT(I)
  522 CONTINUE
  523 IF (KTEST .NE. 1) GO TO 550
C------- COMPUTE RMS AT AUXILIARY POINTS -------------------------------
```

```
      RMSSV = SQRT(RMSSQ)
      IF(INST.EQ.9) RMSSV = SQRT(RM9SV)
      ERLMT = 1.
      LATSV = LATEP
      LONSV = LONEP
      ZSV = Z
      AVL = LATEP/60.
      M1 = AVL + 1.5
      M2 = AVL*10. + 1.5
      DELAT = TEST(13)/CB(M1)
      DELON = TEST(13)/(CA(M1)*COS((M2-1)*.0017453))
      DEZ = TEST(13)
      WRITE(9,525)
  525 FORMAT (/"        LAT        LON          Z      AVRPS        RMS
     1                 DRMS"/)
      NA=1
      GO TO 111
  550 TIME1=TIME2
  575 CONTINUE
C-------CHECK FOR MULTIPLE SOLUTIONS OF THE SAME EARTHQUAKE -----------
      IF(IPRO.NE.ISTTT) RETURN
      NR=NRP
      NRP1=NR +1
      READ(8,600)  CHECK,IPRO,KNST,INST,ZRES,LAT1,LAT2,LON1,LON2,
     1 AZRES(NRP1)
      WRITE(9,601) CHECK,IPRO,KNST,INST,ZRES,LAT1,LAT2,LON1,LON2
  601 FORMAT(//2A4,9X,2I1,F5.2,1X,2(I4,F6.2),"--- RUN AGAIN ---")
  600 FORMAT(2A4,9X,2I1,F5.2,1X,2(I4,F6.2),T21,A4)
      LATRT=60.*LAT1+LAT2
      LONRT=60.*LON1+LON2
      IF(CHECK.EQ.BLANK) GO TO 30
      WRITE(9,610) CHECK
  610 FORMAT(/" ERROR ",A4," SKIPPED.   INST. CARD DID NOT FOLLOW ***")
      RETURN
      END
```

swmreg

102

```
      SUBROUTINE SWMREG                                                 1653.
C------- COMPUTE GEIGER ADJUSTMENTS BY STEP-WISE MULTIPLE REGRESSION OF 1654.
C          TRAVEL TIME RESIDUALS ---------------------------------------1655.
      CHARACTER*8 ENT,ELM                                               1656. 0
      CHARACTER*4 ISW,IPRO,ISTTT,AHEAD
      CHARACTER*1 IW
      COMMON /A6/ NMAX,LMAX,NS,NL,MMAX,NR,FNO,Z,X(4,101),ZSQ,NRP,DF(101)1657.
      COMMON /A7/ KP,KZ,KOUT,W(101),Y(4),BSE(4),XMEAN(4)                1658. 1
      COMMON /A16/ KLSS(102),CALS(102),IPRN,ISW
      COMMON /A19/ KNO,IELV(102),TEST(15),FLT(2,102),MNO(102),IW(102)   1660.
      COMMON /A21/ KSMP(102),FMO,ONF,B(4),IPH,KF,AVRPS,IEXIT            1661.
      COMMON /A24/ FLTEP,IPRO,ISTTT,ISKP(4),AHEAD(12),FLIM,AF(3),NDEC   1662.
      DIMENSION XSUM(4),SIGMA(4),IDX(4),V(3),PF(3),A(7,7),T(7,7),S(4,4) 1663.
      DATA L,M,MM,M1/3,4,7,5/,ENT,ELM/"ENTERING","LEAVING."/            1664.
C-----------------------------------------------------------------------1665.
      KFLAG=0                                                           1666.
      SVTEST = TEST(3)                                                  1667.
      ONF=0.0                                                           1668.
      FLIM = TEST(3)                                                    1669.
      DO 2 I=1,3                                                        1670.
      AF(I)=-1.00                                                       1671.
    2 CONTINUE                                                          1672.
      DO 5 I=1,NR                                                       1673.
      ONF=ONF + W(I)*(1-KSMP(I))                                        1674.
    5 CONTINUE                                                          1675.
      DO 10 I=1,MM                                                      1676.
      DO 10 J=1,MM                                                      1677.
   10 A(I,J)=0.                                                         1678.
C-----COMPUTE MEANS,STANDARD DEVIATIONS,AND CORRECTED SUMS OF SQUARE    1679.
      DO 40 I=1,M                                                       1680.
      XSUM(I)=0.                                                        1681.
      XMEAN(I)=0.                                                       1682.
      DO 40 J=1,M                                                       1683.
   40 S(I,J)=0.                                                         1684.
      DO 50 K=1,NR                                                      1685.
      DO 50 I=1,M                                                       1686.
      TEMP=X(I,K)*W(K)                                                  1687.
      ETMP=TEMP*(1-KSMP(K))                                             1688.
      XSUM(I)=XSUM(I)+ETMP                                              1689.
      DO 50 J=I,M                                                       1690.
   50 S(I,J)=S(I,J)+TEMP*X(J,K)                                         1691.
      DO 70 I=1,M                                                       1692.
      IF (ONF .EQ. 0.) GO TO 65                                         1693.
      XMEAN(I)=XSUM(I)/ONF                                              1694.
      DO 60 J=I,M                                                       1695.
   60 S(I,J)=S(I,J)-XSUM(I)*XSUM(J)/ONF                                 1696.
   65 A(I,I)=1.                                                         1697.
      IF (S(I,I) .LT. 0.000001) S(I,I)=0.000001                         1698.
      SIGMA(I)=SQRT(S(I,I))                                             1699.
   70 CONTINUE                                                          1700.
C-----COMPUTE AND AUGMENT CORRELATION MATRIX A                          1701.
      DO 80 I=1,L                                                       1702.
      I1=I+1                                                            1703.
      DO 80 J=I1,M                                                      1704.
      QZ=SIGMA(I)*SIGMA(J)
      IF(QZ.EQ.0.) QZ=.1E-10
      A(I,J)=S(I,J)/QZ
   80 A(J,I)=A(I,J)                                                     1706.
      PHI=FNO-1.                                                        1707.
      DO 120 I=M1,MM                                                    1708.
      A(I-M,I)=1.                                                       1709.
  120 A(I,I-M)=-1.                                                      1710.
  130 DO 140 I=1,M                                                      1711.
      B(I)=0.                                                           1712.
      Y(I)=0.
      BSE(I)=0.
  140 IDX(I)=0
      IF (IPRN .LT. 3) GO TO 150
      WRITE(9,45)
   45 FORMAT(///, "***** DATA *****",//,4X,"K",8X,"W"
     1,14X,"X1",14X,"X2",14X,"X3",14X,"X4",/)
      DO 47 K=1,NR
      WRITE(9,46) K,W(K),(X(I,K),I=1,M)
   46 FORMAT(I5,8E16.8)
   47 CONTINUE
      WRITE(9,75) (XMEAN(I),I=1,M)
   75 FORMAT(/," MEAN",16X,8E16.8)
      WRITE(9,76) (SIGMA(I),I=1,M)
   76 FORMAT(/," SIGMA",15X,7E16.8)
      WRITE(9,77)
   77 FORMAT(///," ***** CORRECTED SUMS OF SQUARES MATRIX *****",/)
      DO 78 I=1,M
   78 WRITE(9,95) (S(I,J),J=1,M)
      WRITE(9,85)
   85 FORMAT(///," ***** CORRELATION MATRIX R *****",/)
      DO 90 I=1,M
   90 WRITE(9,95) (A(I,J),J=1,M)
   95 FORMAT(7E18.8)
C-----STEPWISE MULTIPLE REGRESSION
      WRITE(9,125) NR,L,TEST(3)
  125 FORMAT(///, "********** STEPWISE MULTIPLE REGRESSION ANALYSIS"
     1," **********",//" NUMBER OF DATA....................",I5
     2,               /," NUMBER OF INDEPENDENT VARIABLES...",I5
     3,               /," CRITICAL F-VALUE..................",F8.2)
  150 DO 300 NSTEP=1,L
      NU=0
      MU=0
      IF (IPRN .LT. 3) GO TO 155
      WRITE(9,154) NSTEP,KZ,KF
  154 FORMAT(//," ***** STEP NO.",I2," *****",5X,"KZ =",I2,5X,"KF =",I2)
C-----FIND VARIABLE TO ENTER REGRESSION
  155 VMAX=0.
      MAX=NSTEP
      DO 160 I=1,L
      IF(ISKP(I).EQ.1) GO TO 160
      IF (IDX(I) .EQ. 1) GO TO 160
      IF ((I.EQ.3).AND.(KZ.EQ.1)) GO TO 160
      IF(ABS(A(I,I)).LT.1.E-10) A(I,I)=1.E-10
      V(I)=A(I,M)*A(M,I)/A(I,I)
      IF (V(I) .LE. VMAX) GO TO 160
      VMAX=V(I)
      MAX=I
  160 CONTINUE
      F=0.0
      IF(VMAX EQ.0.0) GO TO 163
      IF(VMAX.EQ.A(M,M)) VMAX=A(M,M)-1.E-12
      F=(PHI-1.)*VMAX/(A(M,M)-VMAX)
      IF(F .GE. 1000.) F=999.99
  163 AF(MAX)=F
      IF(KF .GE. 2) GO TO 165
      IF (F .LT. TEST(3)) GO TO 400
  165 IF ((MAX.EQ.3).AND.(KZ.EQ.1)) GO TO 300
  166 NU=MAX
      IDX(NU)=1
      PHI=PHI-1.
C-----COMPUTE MATRIX T FOR THE ENTRANCE OF VARIABLE X(NU)
      DO 170 J=1,MM
      IF(A(NU,NU).EQ.0.0) A(NU,NU)=.1E-10
  170 T(NU,J)=A(NU,J)/A(NU,NU)
      DO 180 I=1,MM
      IF (I .EQ. NU) GO TO 180
      DO 175 J=1,MM
  175 T(I,J)=A(I,J)-A(I,NU)*A(NU,J)/A(NU,NU)
  180 CONTINUE
      DO 190 I=1,MM
      DO 190 J=1,MM
  190 A(I,J)=T(I,J)
      DO 200 I=1,L
      IF (IDX(I) .EQ. 0) GO TO 200
      IF (ABS(A(M,M)*A(I+M,I+M)) .LT. .000001 ) GO TO 195
      PF(I)=PHI*A(I,M)**2/(A(M,M)*A(I+M,I+M))
      IF(PF(I) .GE. 1000.0) PF(I)=999.99
      AF(I) = PF(I)
      GO TO 200
  195 PF(I) = 999.99
  200 CONTINUE
      IF (IPRN .LT. 3) GO TO 210
C=     CALL ANSERR(A,S,XMEAN,SIGMA,IDX,PHI,L,M,MM,PF,NU,ENT)
       WRITE(9,555)
555    FORMAT(/," S.ANSERR ELIMINATED",//)
  210 IF (KF .EQ. 2) GO TO 300
      IF(KF .GE. 3) GO TO 450
C-----FIND VARIABLE TO LEAVE REGRESSION
      DO 250 K=1,L
      IF (IDX(K) .EQ. 0) GO TO 250
      IF (PF(K) .GE. TEST(3)) GO TO 250
      MU=K
      F=PF(MU)
      IDX(MU)=0
      PHI=PHI+1.
      DO 220 J=1,MM
      IF(A(MU+M,MU+M).EQ.0.0) A(MU+M,MU+M)=.1E-10
  220 T(MU,J)=A(MU,J)/A(MU+M,MU+M)
      DO 230 I=1,MM
      IF (I .EQ. MU) GO TO 230
      DO 225 J=1,MM
      IF (J .EQ. MU) GO TO 225
      T(I,J)=A(I,J)-A(I,MU+M)*A(MU+M,J)/A(MU+M,MU+M)
  225 CONTINUE
  230 CONTINUE
      DO 240 I=1,MM
      IF (I .EQ. MU) GO TO 240
      T(I,MU)=A(I,MU)-A(I,MU+M)/A(MU+M,MU+M)
  240 CONTINUE
      DO 245 I=1,MM
      DO 245 J=1,MM
  245 A(I,J)=T(I,J)
      IF (IPRN .LT. 3) GO TO 250
C=    CALL ANSERR(A,S,XMEAN,SIGMA,IDX,PHI,L,M,MM,PF,MU,ELM)
      WRITE(9,555)
  250 CONTINUE
  300 CONTINUE
C-----CHECK TERMINATION CONDITION
  400 KOUT=0
      DO 410 I=1,L
  410 KOUT=KOUT+IDX(I)
      B(4)=XMEAN(M)
      IF (KOUT .NE. 0) GO TO 450
      IF(KF .NE. 1) GO TO 420
      KF = 3
      GO TO 150
  420 TEST(3)= TEST(3)/TEST(6)
```

```
      FLIM=TEST(3)
      KF=1
      KFLAG = 0
      IF(TEST(6) .GT. 1.) GO TO 150
      KFLAG = 1
      KF = 4
      GO TO 150
C-----COMPUTE REGRESSION CONSTANT,COEFFICIENTS,AND STANDARD ERRORS
  450 YSE=77.7
      IF (PHI .GE. 1) YSE=SIGMA(M)*SQRT(ABS(A(M,M)/PHI))
      DO 500 I=1,L
      IF (IDX(I) .EQ. 0) GO TO 500
      IF(S(I,I).EQ.0.0) S(I,I)=.1E-10
      B(I)=A(I,M)*SQRT(S(M,M)/S(I,I))
      BSE(I)=YSE*SQRT(ABS(A(I+M,I+M)/S(I,I)))
      IF(KF .NE. 3) Y(I)=B(I)
      IF(KFLAG .EQ. 0) GO TO 480
      IF(ABS(B(I)) .LE. TEST(6)*BSE(I)) Y(I)=0.
  480 IF(PHI .LT. 1.) BSE(I) = 0.
      B(4)=B(4)-Y(I)*XMEAN(I)
  500 CONTINUE
      IF(KF .NE. 3) Y(4)=B(4)
      TEST(3)=SVTEST
      RETURN
      END
```

trvdrv

```
      SUBROUTINE TRVDRV
C------- COMPUTE TRAVEL TIME AND DERIVATIVES FROM CRUSTAL MODEL --------
      REAL*8 TIME1,TIME2
      REAL LAT,LON,LATR,LONR,MAG
      CHARACTER*1 IW
      CHARACTER*4 ISW,IONE,IPRO,ISTTT,AHEAD
      COMMON /A2/ LAT(102),LON(102),DELTA(101),DX(101),DY(101),T(101)
      COMMON /A6/ NMAX,LMAX,NS,NL,MMAX,NR,FNO,Z,X(4,101),ZSQ,NRP,DF(101)
      COMMON /A8/ CAL(101),XMAG(101),FMAG(101),NM,AVXM,SDXM,NF,AVFM,
     1       SDFM,MAG,KDX(101),AMX(101),PRX(101),CALX(101),FMP(101)
      COMMON /A10/ ANIN(101),AZ(101),TEMP(101),CA(71),CB(71)
      COMMON /A16/ KLSS(102),CALS(102),IPRN,ISW
      COMMON /A17/ TIME1,TIME2,LATR,LONR,KTEST,KAZ,KSORT,KSEL,XFN
      COMMON /A19/ KNO,IELV(102),TEST(15),FLT(2,102),MNO(102),IW(102)
      COMMON /A20/ V( 9),D( 9),VSQ( 9),THK( 9),TID( 9, 9),DID( 9, 9)
      COMMON /A22/ F( 9, 9),G(4, 9),H( 9),DEPTH( 9),IONE
      COMMON /A24/ FLTEP,IPRO,ISTTT,ISKP(4),AHEAD(12),FLIM,AF(3),NDEC
      DIMENSION TINJ(21),DIDJ(21),TR(21)
C-----------------------------------------------------------------------
      IF (ISW .EQ. IONE) GO TO 5
C-----INITIALIZATION FOR FIXED LAYER MODEL -----------------------------
      DO  1 L=1,NL
      IF (D(L) .GT. Z) GO TO  2
    1 CONTINUE
      JL=NL
      GO TO  3
    2 JJ=L
      JL=L-1
    3 TKJ=Z-D(JL)
      TKJSQ=TKJ**2+0.000001
      IF (JL .EQ. NL) GO TO  5
      DO  4 L=JJ,NL
      SQT=SQRT(VSQ(L)-VSQ(JL))
      TINJ(L)=TID(JL,L)-TKJ*SQT/(V(L)*V(JL))
    4 DIDJ(L)=DID(JL,L)-TKJ*V(JL)/SQT
      XOVMAX=V(JJ)*V(JL)*(TINJ(JJ)-TID(JL,JL))/(V(JJ)-V(JL))
    5 DO 300 I=1,NR
      IF (ISW .NE. IONE) GO TO 45
C-----INITIALIZATION FOR VARIABLE LAYER MODEL --------------------------
      JI=KDX(I)
      DEPTH(2)=FLT(KNO,JI)
      IF (Z .LT. FLTEP) DEPTH(2)=0.5*(FLT(KNO,JI)+FLTEP)
      THK(1)=DEPTH(2)
      THK(2)=D(3)-DEPTH(2)
      DH1=THK(1)-H(1)
      DH2=THK(2)-H(2)
      DO 10 L=1,NL
      IF (DEPTH(L) .GT. Z) GO TO 20
   10 CONTINUE
      JL=NL
      GO TO 30
   20 JJ=L
      JL=L-1
   30 TKJ=Z-DEPTH(JL)
      TKJSQ=TKJ**2+0.000001
      IF (JL .EQ. NL) GO TO 100
C-----CALCULATION FOR REFRACTED WAVES ----------------------------------
      DO 40 L=JJ,NL
      SQT=SQRT(VSQ(L)-VSQ(JL))
      TIX=F(1,JL)*DH1*G(1,L)+F(2,JL)*DH2*G(2,L)+TID(JL,L)
      DIX=F(1,JL)*DH1*G(3,L)+F(2,JL)*DH2*G(4,L)+DID(JL,L)
      TINJ(L)=TIX-TKJ*SQT/(V(L)*V(JL))
   40 DIDJ(L)=DIX-TKJ*V(JL)/SQT
      TIX=F(1,JL)*DH1*G(1,JL)+F(2,JL)*DH2*G(2,JL)+TID(JL,JL)
      XOVMAX=V(JJ)*V(JL)*(TINJ(JJ)-TIX)/(V(JJ)-V(JL))                         1920.
      GO TO 50                                                                1921.
   45 IF (JL .EQ. NL) GO TO 100                                               1922.
   50 DO 60 M=JJ,NL                                                           1923.
   60 TR(M)=TINJ(M)+DELTA(I)/V(M)                                             1924.
      TMIN=999.99                                                             1925.
      DO 70 M=JJ,NL                                                           1926.
      IF (TR(M) .GT. TMIN) GO TO 70                                           1927.
      IF (DIDJ(M) .GT. DELTA(I)) GO TO 70                                     1928.
      K=M                                                                     1929.
      TMIN=TR(M)                                                              1930.
   70 CONTINUE                                                                1931.
      IF (DELTA(I) .LT. XOVMAX) GO TO 90                                      1932.
C-----TRAVEL TIME & DERIVATIVES FOR REFRACTED WAVE                             1933.
   80 T(I)=TR(K)                                                              1934.
      DTDD=1.0/V(K)                                                           1935.
      DTDH=-SQRT(VSQ(K)-VSQ(JL))/(V(K)*V(JL))                                 1936.
      ANIN(I)=-V(JL)/V(K)                                                     1937.
      GO TO 260                                                               1938.
C-----CALCULATION FOR DIRECT WAVE -----------------------------------------1939.
   90 IF (JL .NE. 1) GO TO 100                                                1940.
      SQT=SQRT(ZSQ+DELTA(I)**2)                                               1941.
      TDJ1=SQT/V(1)                                                           1942.
      IF (TDJ1 .GE. TMIN) GO TO 80                                            1943.
C-----TRAVEL TIME & DERIVATIVES FOR DIRECT WAVE IN FIRST LAYER                 1944.
      T(I)=TDJ1                                                               1945.
      DTDD=DELTA(I)/(V(1)*SQT)                                                1946.
      DTDH=Z/(V(1)*SQT)                                                       1947.
      ANIN(I)=DELTA(I)/SQT                                                    1948.
      GO TO 260                                                               1949.
C-----FIND A DIRECT WAVE THAT WILL EMERGE AT THE STATION                       1950.
  100 XBIG=DELTA(I)                                                           1951.
      XLIT=DELTA(I)*TKJ/Z                                                     1952.
      IF(XLIT.GE. 1.0E16) WRITE(9,997) XLIT,DELTA(I),TKJ,Z
997   FORMAT (' XLIT,DELTA(I),TKJ,Z = ',(2X,4E15.8), ' ST. 1952. ' )
      UB=XBIG/SQRT(XBIG**2+TKJSQ)                                             1953.00
      UL=XLIT/SQRT(XLIT**2+TKJSQ)                                             1954.00
      IF(XLIT.GE.1.0E16)  WRITE(9,996)  XLIT
996   FORMAT ('  STMNT. 1954, XLIT =  ', E15.8)
      UBSQ=UB**2                                                              1955.
      ULSQ=UL**2                                                              1956.
      DELBIG=TKJ*UB/SQRT((1.-UB)*(1.+UB)+0.000001)                            1957.00
      DELLIT=TKJ*UL/SQRT((1.-UL)*(1.+UL)+0.000001)                            1958.00
      J1=JL-1                                                                 1959.
      IF(JL.LE.1) WRITE(9,995) JL
995   FORMAT (   '  STMNT 1959, JL =   ', I8)
      IF (J1.LT.1) GO TO 115
      DO 110 L=1,J1                                                           1960.
      VR=V(JL)/V(L)
      DELBIG=DELBIG+(THK(L)*UB)/SQRT((VR-UB)*(VR+UB)+.000001)                 1961.00
  110 DELLIT=DELLIT+(THK(L)*UL)/SQRT((VR-UL)*(VR+UL)+.000001)                 1962.00
  115 CONTINUE
      DO 170 LL=1,25                                                          1963.
c----
c----following abs function added to prevent numerical roundoff making
c----  dellit slightly greater than delbig
      IF (ABS(DELBIG-DELLIT) .LT. 0.02) GO TO 180                             1964.00
      IF(XLIT.GE. 1.0E16) WRITE(9,994) XLIT
994   FORMAT ('  STMNT 1964, XLIT= ', E15.8)
      XTR=XLIT+(DELTA(I)-DELLIT)*(XBIG-XLIT)/(DELBIG-DELLIT)                  1965.
      IF(XTR.GE.1.0E18) GOTO 99
      U=XTR/SQRT(XTR**2+TKJ*TKJ)                                              1966.
      USQ=U**2                                                                1967.
      DELXTR=TKJ*U/SQRT(1.000001-USQ)                                         1968.
```

```
C----following statement is an identity, from the preceding 3 statements
      DELXTR=XTR
      DO 120 L=1,J1
  120 DELXTR=DELXTR+(THK(L)*U)/SQRT(VSQ(JL)/VSQ(L)-USQ)
      XTEST=DELTA(I)-DELXTR
      IF (ABS(XTEST) .LE. 0.02) GO TO 190
      IF (XTEST) 140,190,150
  140 XBIG=XTR
      DELBIG=DELXTR
      GO TO 160
  150 XLIT=XTR
      DELLIT=DELXTR
  160 IF (LL .LT. 10) GO TO 170
      IF (1.0-U .LT. 0.0002) GO TO 190
  170 CONTINUE
  180 XTR=0.5*(XBIG+XLIT)
      U=XTR/SQRT(XTR**2+TKJSQ)
      USQ=U**2
  190 IF (1.0-U .GT. 0.0002) GO TO 220
C-----IF U IS TOO NEAR 1, COMPUTE TDIR AS WAVE ALONG THE TOP OF LAYER JL
      IF (ISW .EQ. IONE) GO TO 195
      TDC=TID(JL,JL)+DELTA(I)/V(JL)
      GO TO 200
  195 TIX=F(1,JL)*DH1*G(1,JL)+F(2,JL)*DH2*G(2,JL)+TID(JL,JL)
      TDC=TIX+DELTA(I)/V(JL)
  200 IF (JL .EQ. NL) GO TO 210
      IF (TDC .GE. TMIN) GO TO 80
  210 T(I)=TDC
      DTDD=1.0/V(JL)
      DTDH=0.0
      ANIN(I)=0.9999999
      GO TO 260
C-----TRAVEL TIME & DERIVATIVES FOR DIRECT WAVE BELOW FIRST LAYER
  220 TDIR=TKJ/(V(JL)*SQRT(1.0-USQ))
      DO 240 L=1,J1
  240 TDIR=TDIR+(THK(L)*V(JL))/(VSQ(L)*SQRT(VSQ(JL)/VSQ(L)-USQ))
      IF (JL .EQ. NL) GO TO 245
      IF (TDIR .GE. TMIN) GO TO 80
  245 T(I)=TDIR
      SRR=SQRT(1.-USQ)
      SRT=SRR**3
      ALFA=TKJ/SRT
      BETA=TKJ*U/(V(JL)*SRT)
      DO 250 L=1,J1
      STK=(SQRT(VSQ(JL)/VSQ(L)-USQ))**3
      VTK=THK(L)/(VSQ(L)*STK)
      ALFA=ALFA+VTK*VSQ(JL)
  250 BETA=BETA+VTK*V(JL)*U
      DTDD=BETA/ALFA
      DTDH=(1.0-V(JL)*U*DTDD)/(V(JL)*SRR)
      ANIN(I)=U
C-----SET UP PARTIAL DERIVATIVES FOR REGRESSION ANALYSIS ---------------
  260 X(1,I)=-DTDD*DX(I)/DELTA(I)
      X(2,I)=-DTDD*DY(I)/DELTA(I)
      X(3,I)=DTDH
  300 CONTINUE
      RETURN
99    CONTINUE
      WRITE(9,998)  TKJ,Z
998   FORMAT ( '  TKJ= ',E15.8, '  Z= ',E15.8)
      WRITE(9,999) I,LL,XTR,XLIT,XBIG,DELLIT,DELBIG,DELTA(I)
999   FORMAT (1X,2I5, 6(2X,E15.8))
      END
```

xfmags

```
      SUBROUTINE XFMAGS
C------- COMPUTE X-MAGNITUDE AND F-MAGNITUDE ---------------------------
      INTEGER*4 IBLANK
      CHARACTER*4 NSTA,MBK,MDOL,BLANK,MSTAR,DOT,STAR4,MCENT,ISW,CBLANK
      CHARACTER*4 LAZT
      CHARACTER*3 CRMK
      CHARACTER*1 QUES,ISTAR,IW
      REAL LAT,LON,MAG,RBLANK
      COMMON /A1/ NSTA(102),DLY(2,102),FMGC(102),XMGC(102),KLAS(102),
     1       PRR(102),CALR(102),ICAL(102),IS(102)
      COMMON /A2/ LAT(102),LON(102),DELTA(101),DX(101),DY(101),T(101)
      COMMON /A5/ ZTR,XNEAR,XFAR,POS,IQ,KMS,KFM,IPUN,IMAG,IR,QSPA(9,40)
      COMMON /A6/ NMAX,LMAX,NS,NL,MMAX,NR,FNO,Z,X(4,101),ZSQ,NRP,DF(101)
      COMMON /A8/ CAL(101),XMAG(101),FMAG(101),NM,AVXM,SDXM,NF,AVFM,
     1       SDFM,MAG,KDX(101),AMX(101),PRX(101),CALX(101),FMP(101)
      COMMON /A14/ MBK,MDOL,BLANK,MSTAR,DOT,STAR4,QUES,CRMK,MCENT,ISTAR
      COMMON /A16/ KLSS(102),CALS(102),IPRN,ISW
      COMMON /A19/ KNO,IELV(102),TEST(15),FLT(2,102),MNO(102),IW(102)
      COMMON /S25/ ZDOT,CBLANK,IBLANK,RBLANK,LAZT
      DIMENSION RSPA(8,20)
      DATA ZMC1,ZMC2,PWC1,PWC2/0.15,3.38,0.80,1.50/
      DATA RSPA/-0.02, 1.05,-0.15,-0.13, 0.66, 0.55, 0.17, 0.42,
     2          0.14, 1.18,-0.01, 0.01, 0.79, 0.66, 0.27, 0.64,
     3          0.30, 1.29, 0.12, 0.14, 0.90, 0.76, 0.35, 0.84,
     4          0.43, 1.40, 0.25, 0.27, 1.00, 0.86, 0.43, 0.95,
     5          0.55, 1.49, 0.38, 0.41, 1.08, 0.93, 0.49, 1.04,
     6          0.65, 1.57, 0.53, 0.57, 1.16, 1.00, 0.55, 1.13,
     7          0.74, 1.63, 0.71, 0.75, 1.23, 1.07, 0.63, 1.24,
     8          0.83, 1.70, 0.90, 0.95, 1.30, 1.15, 0.72, 1.40,
     9          0.92, 1.77, 1.07, 1.14, 1.38, 1.25, 0.83, 1.50,
     A          1.01, 1.86, 1.23, 1.28, 1.47, 1.35, 0.95, 1.62,
     B          1.11, 1.96, 1.35, 1.40, 1.57, 1.46, 1.08, 1.73,
     C          1.20, 2.05, 1.45, 1.49, 1.67, 1.56, 1.19, 1.84,
     D          1.30, 2.14, 1.55, 1.58, 1.77, 1.66, 1.30, 1.94,
     E          1.39, 2.24, 1.65, 1.67, 1.86, 1.76, 1.40, 2.04,
     F          1.47, 2.33, 1.74, 1.76, 1.95, 1.85, 1.50, 2.14,
     G          1.53, 2.41, 1.81, 1.83, 2.03, 1.93, 1.58, 2.24,
     H          1.56, 2.45, 1.85, 1.87, 2.07, 1.97, 1.62, 2.31,
     I          1.53, 2.44, 1.84, 1.86, 2.06, 1.96, 1.61, 2.31,
     J          1.43, 2.36, 1.76, 1.78, 1.98, 1.88, 1.53, 1.92,
     K          1.25, 2.18, 1.59, 1.61, 1.82, 1.72, 1.37, 1.49/
C-----------------------------------------------------------------------
      NM=0
      AVXM=0.
      SDXM=0.
      NF=0
      AVFM=0.
      SDFM=0.
      DO 40 I=1,NRP
      XMAG(I)=RBLANK
      RAD2=DELTA(I)**2+ZSQ
      IF ((RAD2.LT.1.).OR.(RAD2.GT.360000.)) GO TO 30
      JI=KDX(I)
      K=KLAS(JI)
      AMXI=ABS(AMX(I))
      CAL(I)=CALX(I)
      IF ((CAL(I).LT.0.01).OR.(ICAL(JI).EQ.1)) CAL(I)=CALR(JI)
      IF ((AMXI.LT.0.01).OR.(CAL(I).LT.0.01)) GO TO 30
      IF ((K.LT.0).OR.(K.GT.8)) GO TO 30
      XLMR=0.
      IF (K  EQ.  0) GO TO 20
      PRXI=PRX(I)
      IF (PRXI .LT.  0.01) PRXI=PRR(JI)
      IF (IR .EQ.  0) GO TO 10
      IF ((PRXI.GT.20.).OR.(PRXI.LT.0.033)) GO TO 30
      FQ=10.*ALOG10(1./PRXI)+20.
      IFQ=FQ
      XLMR=QSPA(K,IFQ)+(FQ-IFQ)*(QSPA(K,IFQ+1)-QSPA(K,IFQ))
      GO TO 20
   10 IF ((PRXI.GT.3.).OR.(PRXI.LT.0.05)) GO TO 30
      FQ=10.*ALOG10(1./PRXI)+6.
      IFQ=FQ
      XLMR=RSPA(K,IFQ)+(FQ-IFQ)*(RSPA(K,IFQ+1)-RSPA(K,IFQ))
   20 BLAC=ALOG10(AMXI/(2.*CAL(I)))-XLMR
      RLD2=ALOG10(RAD2)
      BLNT=ZMC1-PWC1*RLD2
      IF (RAD2 .GE. 40000.) BLNT=ZMC2-PWC2*RLD2
      XMAG(I)=BLAC-BLNT+XMGC(JI)
      NM=NM+1
      AVXM=AVXM+XMAG(I)
      SDXM=SDXM+XMAG(I)**2
   30 FMAG(I)=RBLANK
      IF (FMP(I) .EQ.RBLANK) GO TO 40
      FMAG(I)=TEST(7)+TEST(8)*ALOG10(FMP(I))+TEST(9)*DELTA(I)+FMGC(JI)
      NF=NF+1
      AVFM=AVFM+FMAG(I)
      SDFM=SDFM+FMAG(I)**2
   40 CONTINUE
      IF (NM .EQ. 0) GO TO 50
      AVXM=AVXM/NM
      SDXM=SQRT(SDXM/NM-AVXM**2 + .0001)
   50 IF (NF .EQ. 0) GO TO 60
      AVFM=AVFM/NF
      SDFM=SQRT(SDFM/NF-AVFM**2 + .0001)
   60 IF (NM .EQ. 0) AVXM=RBLANK
      IF (NF .EQ. 0) AVFM=RBLANK
      IF (IMAG-1) 70,80,90
   70 MAG=AVXM
      RETURN
   80 MAG=AVFM
      RETURN
   90 MAG=0.5*(AVXM+AVFM)
      IF (AVXM .EQ.RBLANK) GO TO 80
      IF (AVFM .EQ.RBLANK) GO TO 70
      RETURN
      END
```

sort

```
      SUBROUTINE SORT(X,KEY,NO)
      DIMENSION X(NO),KEY(NO)
C-----------------------------------------------------------------------
      DO 1 I=1,NO
 1    KEY(I)=I
      MO=NO
 2    IF (MO-15) 21,21,23
 21   IF (MO-1) 29,29,22
 22   MO=2*(MO/4)+1
      GO TO 24
 23   MO=2*(MO/8)+1
 24   KO=NO-MO
      JO=1
 25   I=JO
 26   IF (X(I)-X(I+MO)) 28,28,27
 27   TEMP=X(I)
      X(I)=X(I+MO)
      X(I+MO)=TEMP
      KEMP=KEY(I)
      KEY(I)=KEY(I+MO)
      KEY(I+MO)=KEMP
      I=I-MO
      IF (I-1) 28,26,26
 28   JO=JO+1
      IF (JO-KO) 25,25,2
 29   RETURN
      END
      program mizing
      dimension nsta(400),slat(400),slon(400),idx(400),msta(200),
     1azm(200),key(200),ltap(400),ldev(400)
      character*1 ibk,ifmt,nst4
      character*2 ldev
      character*3 ltap
      character*131 a,card
      character*4 nsta,msta,zzzz,kblk,ktst,dtst,itst,mdol
      character*5 blk5,mag
      data blk5/'     '/,ibk/' '/,zzzz/'ZZZZ'/
      data kblk/'    '/,dtst/'DATE'/,itst/'STN '/,mdol/' $$$'/
      data gptst/15.0/,tdz/25.0/,tde/1.70/
      open(unit=8,file='calstn',blank='zero')
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
      rewind 8
      i=0
    2 i=i+1
      read(8,690) nsta(i),lat,xlat,lon,xlon,ltap(i),ldev(i),nst4
      if(nsta(i) .eq. kblk .or. nsta(i) .eq. zzzz) go to 12
      if(nst4 .ne. ibk) go to 3
      slat(i)=60.*lat+xlat
      slon(i)=60.*lon+xlon
      go to 2
    3 i=i-1
      go to 2
   12 nbst=i-1
  690 format(2x,a4,i2,f5.2,1x,i3,f5.2,t72,a3,1x,a2,t6,a1)
   15 read(5,700,end=150) ifmt,a
  700 format(a1,a131)
      if(ifmt .ne. ibk) go to 120
      read(a,701) card,ktst
  701 format(a131,t2,a4)
      write(6,707) card
  707 format(1x,a131)
      if(ktst .ne. dtst) go to 15
   16 read(5,700,end=150) ifmt,a
      if(ifmt .ne. ibk) go to 121
      read(a,702) card,kyr
  702 format(a131,t1,i2)
      write(6,707) card
      if(kyr .gt. 90 .or. kyr .lt. 1) go to 150
      read(a,703)late,xlate,lone,xlone,xmag,dmin,mag
  703 format(18x,i2,1x,f5.2,1x,i3,1x,f5.2,8x,f5.2,3x,f3.0,t45,a5)
      ihd=0
      elat=60.*late+xlate
      elon=60.*lone+xlone
      do 17 i=1,nbst
      idx(i)=999
   17 continue
   18 read(5,700,end=150) ifmt,a
      if(ifmt .ne. ibk) go to 122
      read(a,701) card,ktst
      write(6,707) card
      if(ktst .ne. itst) go to 18
      j=0
   19 read(5,700,end=28) ifmt,a
      if(ifmt .ne. ibk) go to 123
      read(a,704) card,ktst
  704 format(a131,t1,a4)
      write(6,707) card
      if(ktst .eq. kblk .or. ktst .eq. mdol) go to 32
      j=j+1
      read(a,705)msta(j),azm(j)
  705 format(a4,6X,f4.0)
      do 25 i=1,nbst
      if(nsta(i) .eq. msta(j)) go to 27
   25 continue
      write(6,725) msta(j)
  725 format(5x,a4,' not on station list')
      go to 19
   27 idx(i)=j
      go to 19
   28 if(j .lt. 1) go to 150
   32 nobs=j
      call sort(azm,key,nobs)
      nj=nobs+1
      azm(nj)=azm(1)+360.
      tdel=tdz*xmag**tde
      if(mag .eq. blk5) tdel=100.
      do 100 i=1,nbst
      if(idx(i) .ne. 999) go to 100
      call dstaz(slat(i),slon(i),elat,elon,dist,azim)
      if(dist .gt. tdel) go to 100
      if(azim .le. azm(1)) azim=azim+360.
      do 70 j=2,nj
      if(azim .lt. azm(j)) go to 80
   70 continue
   80 exgap=azm(j)-azm(j-1)
      rdgap =azm(j)-azim
      tgap=azim-azm(j-1)
      if(tgap .lt. rdgap) rdgap=tgap
      if((dist .gt. dmin) .and. (rdgap .lt. gptst)) go to 100
      if (azim .gt. 360. ) azim=azim-360.
      if(ihd .eq. 1) go to 82
      write(6,755)
  755 format(/,10x,'missing station  delta   azim  ex-gap  rd-gap',
     1'   tape   dev')
      ihd=1
   82 write(6,760) nsta(i),dist,azim,exgap,rdgap,ltap(i),ldev(i)
  760 format(21x,a4,2f7.1,2f8.1,4x,a3,4x,a2)
  100 continue
      go to 15
  120 write(6,770)
  770 format(/)
      go to 15
  121 write(6,770)
      go to 16
  122 write(6,770)
      go to 18
  123 write(6,770)
      go to 19
  150 stop
      end
      SUBROUTINE SORT(X,KEY,NO)
      DIMENSION X(NO),KEY(NO)
      DO 1 I=1,NO
    1 KEY(I)=I
      MO=NO
    2 IF(MO-15) 21,21,23
   21 IF(MO-1) 29,29,22
   22 MO=2*(MO/4)+1
      GO TO 24
   23 MO=2*(MO/8)+1
   24 KO=NO-MO
      JO=1
   25 I=JO
   26 IF(X(I)-X(I+MO)) 28,28,27
```

```
 27 TEMP=X(I)
    X(I)=X(I+MO)
    X(I+MO)=TEMP
    KEMP=KEY(I)
    KEY(I)=KEY(I+MO)
    KEY(I+MO)=KEMP
    I=I-MO
    IF(I-1) 28,26,26
 28 JO=JO+1
    IF(JO-KO) 25,25,2
 29 RETURN
    END
    SUBROUTINE DSTAZ(SLAT,SLON,ELAT,ELON,DIST,AZIM)
    ALAT=0. 5*(SLAT+ELAT)
    ALAT1=ALAT*3. 14159/10800.
    ALAT2=2. *ALAT1
    ALAT3=3. *ALAT1
    ALAT4=4. *ALAT1
    AA=(111. 4151-0. 0946*COS(ALAT3)/COS(ALAT1))/60.
    BB=(111. 1321-0. 5661*COS(ALAT2)+0. 0021*COS(ALAT4))/60.
    XF=AA*COS(ALAT1)*(SLON-ELON)
    YF=BB*(SLAT-ELAT)
    DIST=SQRT(XF*XF+YF*YF)
    AYF=ABS(YF)
    IF(AYF .LT. 0. 000001) YF=0. 000001
353 IF(YF .LE. 0. 0) GO TO 356
354 AZ=-ATAN(XF/YF)
    GO TO 357
356 AZ=3. 14159-ATAN(XF/YF)
357 IF(AZ) 358,358,359
358 AZ=AZ+6. 28318
359 AZIM=57. 29578*AZ
    RETURN
    END
```

mising

```
      program mising
      dimension nsta(400),slat(400),slon(400),idx(400),msta(200),
     1azm(200),key(200),ltap(400),ldev(400),ptt(200),kard(80)
      character*1 ibk,iprt,kard,idol,nst4,istr
      character*2 ldev
      character*3 ltap
      character*80  card
      character*80 a
      character*4 nsta,msta,zzzz,kblk
      character*5 blk5,mag
      data blk5/'     '/,ibk/' '/,zzzz/'ZZZZ'/,istr/'*'/
      data kblk/'    '/,idol/'$'/
      data gptst/15.0/,tdz/25.0/,tde/1.70/
      open(unit=8,file='calstn',blank='zero')
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
      rewind 8
      i=0
    2 i=i+1
      read(8,705) nsta(i),lat,xlat,lon,xlon,ltap(i),ldev(i),nst4
      if(nsta(i) .eq. kblk .or. nsta(i) .eq. zzzz) go to 12
      if(nst4 .ne. ibk) go to 3
      slat(i)=60.*lat+xlat
      slon(i)=60.*lon+xlon
      go to 2
    3 i=i-1
      go to 2
   12 nbst=i-1
  705 format(2x,a4,i2,f5.2,1x,i3,f5.2,t72,a3,1x,a2,t6,a1)
      read(5,707) iprt
  707 format(a1)
   15 read(5,710,end=150) card,kyr,sec,late,xlate,lone,xlone,xmag,
     1dmin,mag
  710 format(a80,t1,i2,t13,f5.2,1x,i2,1x,f5.2,1x,i3,1x,f5.2,9x,f5.2,
     17x,f5.2,t46,a5)
      if(kyr .eq. 99 .or. kyr .lt. 1) go to 150
      ihd=0
      elat=60.*late+xlate
      elon=60.*lone+xlone
      write(6,711)
  711 format(///,' DATE    ORIGIN     LAT      LONG      DEPTH     MAG',
     1' NO GAP DMIN  RMS  ERH  ERZ QM')
      write(6,722) card
  722 format(a80)
      do 20 i=1,nbst
      idx(i)=999
   20 continue
      if(iprt .ne. ibk) go to 21
      write(6,713)
  713 format(/,'STN    DIST   AZIM   ANIN PHSE    POBS  PTTM  PRES  PWT  ',
     1'XMAG  FMAG')
   21 do 30 j=1,300
      read(5,701) a
  701 format(a80)
      read(a,702) kard
  702 format(80a1)
      if(kard(2) .eq. idol) go to 32
      if(kard(30) .eq. istr) go to 203
      read(a,720) msta(j),azm(j),ptt(j)
  720 format(a4,6x,f6.1,11x,f6.2 )
      if(iprt .ne. ibk) go to 22
      pobs=ptt(j)+sec
      write(6,723) msta(j),(kard(l),l=5,27),pobs,(kard(l),l=28,50),
     1(kard(l),l=55,60)
```

```
723 format(a4,23a1,f7.2,29a1)
    go to 22
203 read(a,721) msta(j),azm(j)
721 format(a4,6x,f6.1)
    if(iprt .ne. ibk) go to 22
    write(6,724)msta(j),(kard(l),l=5,27),(kard(l),l=28,50),
   1(kard(l),l=55,60)
724 format(a4,23a1,' ***** ',29a1)
 22 do 25 i=1,nbst
    if(nsta(i) .eq. msta(j)) go to 27
 25 continue
    write(6,725) msta(j)
725 format(5x,a4,' not on station list')
    go to 30
 27 idx(i)=j
 30 continue
 32 write(6,726)
726 format(' $$$')
    nobs=j-1
    call sort(azm,key,nobs)
    nj=nobs+1
    azm(nj)=azm(1)+360.
    tdel=tdz*xmag**tde
    if(mag .eq. blk5) tdel=100.
    do 100 i=1,nbst
    if(idx(i) .ne. 999) go to 100
    call dstaz(slat(i),slon(i),elat,elon,dist,azim)
    if(dist .gt. tdel) go to 100
    if(azim .le. azm(1)) azim=azim+360.
    do 70 j=2,nj
    if(azim .lt. azm(j)) go to 80
 70 continue
 80 exgap=azm(j)-azm(j-1)
    rdgap =azm(j)-azim
    tgap=azim-azm(j-1)
    if(tgap .lt. rdgap) rdgap=tgap
    if((dist .gt. dmin) .and. (rdgap .lt. gptst)) go to 100
    if (azim .gt. 360. ) azim=azim-360.
    if(ihd .eq. 1) go to 82
    write(6,755)
755 format(/,10x,'missing station  delta   azim  ex-gap  rd-gap',
   1'   tape   dev')
    ihd=1
 82 write(6,760) nsta(i),dist,azim,exgap,rdgap,ltap(i),ldev(i)
760 format(21x,a4,2f7.1,2f8.1,4x,a3,4x,a2)
100 continue
    go to 15
150 stop
    end
    SUBROUTINE SORT(X,KEY,NO)
    DIMENSION X(NO),KEY(NO)
    DO 1 I=1,NO
  1 KEY(I)=I
    MO=NO
  2 IF(MO-15) 21,21,23
 21 IF(MO-1) 29,29,22
 22 MO=2*(MO/4)+1
    GO TO 24
 23 MO=2*(MO/8)+1
 24 KO=NO-MO
    JO=1
 25 I=JO
 26 IF(X(I)-X(I+MO)) 28,28,27
 27 TEMP=X(I)
```

```
      X(I)=X(I+MO)
      X(I+MO)=TEMP
      KEMP=KEY(I)
      KEY(I)=KEY(I+MO)
      KEY(I+MO)=KEMP
      I=I-MO
      IF(I-1) 28,26,26
   28 JO=JO+1
      IF(JO-KO) 25,25,2
   29 RETURN
      END
      SUBROUTINE DSTAZ(SLAT,SLON,ELAT,ELON,DIST,AZIM)
      ALAT=0. 5*(SLAT+ELAT)
      ALAT1=ALAT*3. 14159/10800.
      ALAT2=2. *ALAT1
      ALAT3=3. *ALAT1
      ALAT4=4. *ALAT1
      AA=(111. 4151-0. 0946*COS(ALAT3)/COS(ALAT1))/60.
      BB=(111. 1321-0. 5661*COS(ALAT2)+0. 0021*COS(ALAT4))/60.
      XF=AA*COS(ALAT1)*(SLON-ELON)
      YF=BB*(SLAT-ELAT)
      DIST=SQRT(XF*XF+YF*YF)
      AYF=ABS(YF)
      IF(AYF .LT. 0.000001) YF=0.000001
  353 IF(YF .LE. 0.0) GO TO 356
  354 AZ=-ATAN(XF/YF)
      GO TO 357
  356 AZ=3. 14159-ATAN(XF/YF)
  357 IF(AZ) 358,358,359
  358 AZ=AZ+6. 28318
  359 AZIM=57. 29578*AZ
      RETURN
      END
```

```
C     PROGRAM PLTFM
C     TO SCREEN FIRST MOTIONS FOR RELIABILITY AND TO PLOT THEM, AS C OR
C     D FOR CERTAIN AND AS + OR - FOR UNCERTAIN ONSETS, ON AN EQUAL
C     AREA PROJECTION FROM HYPO71 OUTPUT PUNCHED CARDS
C
C     INPUT: TEST VALUES FOR RELIABILITY PARAMETERS,LIST OF REVERSED
C     STATIONS, HYPO71 OUTPUT CARDS, AND ONE " $$$" CARD TO SIGNAL
C     END OF DATA
C
C     OUTPUT: FOR EACH EARTHQUAKE, A LISTING OF INPUT DATA AND A
C     PRINTER PLOT OF FIRST MOTION DATA ON AN EQUAL AREA PROJECTION
C
C          CODE FOR SCREENING PARAMETERS
C     IF XMAG <TMAG   DELETE EVENT
C     IF NO<KTTA   DELETE EVENT
C     IF PRES>XFD   DOWNGRADE SYMBOL
C     IF PRES>XFE   DELETE STATION
C     IF DIST>XDD   DOWNGRADE SYMBOL
C     IF DIST>XDE   DELETE STN
C     IF WT>KWD   DOWNGRADE SYMBOL
C     IF WT>KWE   DELETE STATION
C     IF KQD=1  DELETE STATIONS WITHOUT U OR D OR + OR -
C     IF KQE=1   DELETE STATIONS WITHOUT U OR D
C     IF KED=1   DOWNGRADE SYMBOLS FOR EMERGENT P PHASES
C     IF KEE=1   DELETE STATIONS WITH EMERGENT P PHASES
      DIMENSION CARD(20), KARD(20), LSTA(100), MSTA(300),DELTA(300),
     1AZ(300),AIN(300),PRK1(300),PRK2(300),PRK3(300),PRK4(300),
     2PRES(300),RMK(300),SYM(300)
      CHARACTER*3 RMK
      CHARACTER*4 LSTA,CARD,MSTA,KARD
      CHARACTER*4 BLANK,NEND,MEND,NLST
      CHARACTER*1 BLK,LP,SU,SD,SC,SP,SM,SI
      CHARACTER*1 PRK1,PRK2,PRK3,SYM
      INTEGER PRK4
      DATA BLANK, NEND, MEND, NLST/"    ","$$$$"," $$$","NLST"/
      DATA BLK,LP,SU,SD,SC,SP,SM,SI/" ","P","U","D","C","+","-","I"/
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
C     READ AND PRINT TEST VALUES AND LIST OF REVERSED STATIONS
      NREV=0
      LCYC=0
  102 LCYC=LCYC+1
  104 READ(5,702) TMAG,KTTA,XFD,XFE,XDD,XDE,KWD,KWE,KQD,KQE,KED,KEE
  702 FORMAT(F5.1,I5,2F5.2,2F5.1,6X,2I2,6X,2I2,6X,2I2)
      IF(LCYC .LT. 2 .OR. NREV .EQ. 0) GO TO 110
      DO 106 IR=1,NREV
      LSTA(IR)=BLANK
  106 CONTINUE
  110 DO 4 IR=1,100
      READ(5,704) LSTA(IR)
      IF(LSTA(IR) .EQ. NEND) GO TO 6
    4 CONTINUE
  704 FORMAT(2X,A4)
    6 NREV=IR-1
      WRITE(6,706) TMAG,KTTA,XFD,XFE,XDD,XDE,KWD,KWE,KQD,KQE,KED,KEE
  706 FORMAT('\f',20X,"TEST PARAMETERS",/,11X,"TMAG",3X,"KTTA",4X,
     1"XFD",4X,"XFE",4X,"XDD",4X,"XDE",4X,"KWD",4X,"KWE",4X,"KQD",4X,
     2"KQE",4X,"KED",4X,"KEE",/,10X,F5.1,2X,I5,2(2X,F5.2),
     32(2X,F5.1),6(5X,I2))
      WRITE(6,708) NREV
  708 FORMAT(///,5X,I3,2X,"REVERSED STATIONS",/)
      IF(NREV .LT. 1) GO TO 10
      DO 8 IR=1,NREV
      WRITE(6,710) LSTA(IR)
  710 FORMAT(10X,A4)
    8 CONTINUE
C     READ SUMMARY CARDS
      READ(5,714) CARD
  714 FORMAT(20A4)
   10 READ(5,715,END=550) CARD,KSTA,XMAG
  715 FORMAT(20A4,T51,I3,T45,F6.2)
      KTST =1
      IF(CARD(1) .EQ. BLANK) GO TO 550
      IF(CARD(1) .EQ. NLST) GO TO 102
      IF(XMAG .LT. TMAG .OR. KSTA .LT. KTTA) KTST =0
      IF(KTST .LT. 1) GO TO 18
      WRITE(6,716) CARD
  716 FORMAT('\f',5X,20A4)
   18 I=1
   20 READ(5,725,END=540) KARD,MSTA(I),DELTA(I),AZ(I),AIN(I),PRK1(I),
     1PRK2(I),PRK3(I),PRK4(I),PRES(I),RMK(I)
  725 FORMAT(20A4,T1,A4,3F6 1,1X,3A1,I1,6X,F6.1,12X,A3)
      IF(MSTA(I) .EQ.MEND) GO TO 540
      IF(KTST .LT. 1) GO TO 538
C     ELIMINATION OF BAD READINGS
      IF(PRK2(I) .NE. LP) GO TO 20
      IF(PRK3(I) .EQ. SU .OR. PRK3(I) .EQ. SD) GO TO 22
      IF(PRK3(I) .EQ. BLK .OR. KQE .EQ. 1) GO TO 20
   22 IF(PRK1(I).EQ.BLK .OR. PRK1(I).NE.SI.AND.KEE.EQ.1) GO TO 20
      IF(PRK4(I) .GT. KWE) GO TO 20
      IF(DELTA(I) .GT. XDE) GO TO 20
      FABS=ABS(PRES(I))
      IF(FABS .GT. XFE) GO TO 20
C     CONVERT U TO C
      IF(PRK3(I) .EQ. SU) PRK3(I)=SC
C     DOWNGRADING UNCERTAIN READINGS
      IF(PRK1(I) .NE. SI .AND. KQD .EQ. 1) GO TO 25
      IF(PRK4(I) .GT. KWD) GO TO 25
      IF(FABS .GT. XFD) GO TO 25
      IF(DELTA(I) .GT. XDD) GO TO 25
      GO TO 30
   25 IF(PRK3(I) .EQ. SC) GO TO 27
      IF(PRK3(I) .EQ. SD) GO TO 28
      GO TO 30
   27 PRK3(I)=SP
      GO TO 30
   28 PRK3(I)=SM
   30 CONTINUE
C     CORRECT REVERSED STATIONS
      IF(NREV .LT. 1) GO TO 60
      DO 40 IR=1,NREV
      IF(LSTA(IR) .EQ. MSTA(I)) GO TO 42
   40 CONTINUE
      GO TO 60
   42 IF(PRK3(I) .EQ. SC) GO TO 43
      IF(PRK3(I) .EQ. SD) GO TO 44
      IF(PRK3(I) .EQ. SP) GO TO 45
      IF(PRK3(I) .EQ. SM) GO TO 46
      GO TO 60
   43 PRK3(I)=SD
      GO TO 60
   44 PRK3(I)=SC
      GO TO 60
   45 PRK3(I)=SM
      GO TO 60
   46 PRK3(I)=SP
   60 SYM(I)=PRK3(I)
      WRITE(6,750)KARD,SYM(I),I
  750 FORMAT(5X,20A4,4X,A1,2X,I3)
  538 I=I+1
      IF(I .GT. 300) GO TO 550
      GO TO 20
  540 LAST=I-1
      IF(KTST .LT. 1) GO TO 10
      IF(LAST .LT. 1) GO TO 10
      CALL QPROJ(AZ,AIN,SYM,LAST,CARD)
      GO TO 10
  550 STOP
      END
      SUBROUTINE QPROJ(AZ,IN,SYM,LAST,CARD)
C     PLOTS AN EQUAL AREA PROJECTION OF A SET OF POINTS EACH DEFINED
C     BY AN AZIMUTH AND AN ANGLE OF INCIDENCE
C     ARGUMENTS
C           AZ     THE ARRAY OF AZIMUTHS
C           IN     THE ARRAY OF ANGLES OF INCIDENCE
C           SYM    AN ARRAY OF SYMBOLS, EACH ELEMENT OF WHICH IS
C                        =C     FOR COMPRESSION
C                  OR    =D     FOR DILATATION
C           LAST   THE NUMBER OF OBSERVATIONS
C           CARD   HEADING FOR PLOT
C
      CHARACTER*1 BORD,BLANK,PL,CR,DOT,SI,A,B,C,D,E,F,CD,SYM,GRAPH,TEMP
      CHARACTER*4 CARD
      REAL IN,INN
      DIMENSION AZ(LAST),IN(LAST),SYM(LAST),GRAPH(95,59),CARD(20)
      DATA BORD,BLANK,PL,CR,DOT,SI/"*"," ","+","-",".","I"/
      DATA A,B,C,D,E,F,CD/"A","B","C","D","E","F","X"/
C
      NOX=95
      NOY=59
      XN=NOX-1.
      YN=NOY-3.
      RADIUS=10.
      RMAX=RADIUS/2.54
      ADD=4.75
      XSCALE=9.5/XN
      YSCALE=9.5/YN
      IX=RMAX*10.+.5
      IY=RMAX*6.+.5
      NOX2=NOX/2+1
      NOY2=NOY/2+1
      DO 10 I=1,NOX
      DO 10 J=1,NOY
   10 GRAPH(I,J)=BLANK
      DO 20 I=1,180
      PHI=I*2.*.0174533
      X=RMAX*COS(PHI)+ADD
      Y=RMAX*SIN(PHI)+ADD
      JX=X/XSCALE+1.5
      JY=Y/YSCALE+.5
      JY=NOY-JY-1
      GRAPH(JX,JY)=BORD
   20 CONTINUE
      IT=NOX2-IX-1
      GRAPH(IT,NOY2)=CR
      IT=NOX2+IX+1
      GRAPH(IT,NOY2)=CR
      IT=NOY2-IY-1
      GRAPH(NOX2,IT)=SI
      IT=NOY2+IY+1
      GRAPH(NOX2,IT)=SI
      DO 50 I=1,LAST
      IF(IN(I).GT.90.) GO TO 31
      INN=IN(I)
      AZZ=AZ(I)
      GO TO 32
   31 INN=180.-IN(I)
      AZZ=180.+AZ(I)
   32 R=RMAX*1.414214*SIN(INN  *.0087266)
      X=R*SIN(AZZ  *.0174533)+ADD
      Y=R*COS(AZZ  *.0174533)+ADD
      JX=X/XSCALE+1.5
      JY=Y/YSCALE+.5
      JY=NOY-JY-1
      TEMP=GRAPH(JX,JY)
C     OVER-WRITE TEMP IF IT IS EQUAL TO BLANK,DOT,*,+,-
      IF(TEMP.EQ.BLANK) GO TO 47
      IF(TEMP.EQ.DOT)GO TO 47
      IF(TEMP.EQ.BORD) GO TO 47
      IF(TEMP.EQ.PL) GO TO 47
      IF(TEMP.EQ.CR) GO TO 47
C     TEMP IS OCCUPIED; SO IF SYM(I)=+ OR - SKIP THIS STATION
      IF (SYM(I).EQ PL) GO TO 50
      IF (SYM(I).EQ.CR) GO TO 50
      IF(SYM(I).EQ.C) GO TO 40
      IF(GRAPH(JX,JY).NE.D) GO TO 35
      GRAPH(JX,JY)=E
      GO TO 50
   35 IF(GRAPH(JX,JY).NE.E) GO TO 37
      GRAPH(JX,JY)=F
      GO TO 50
   37 IF(GRAPH(JX,JY).EQ.F) GO TO 50
      GRAPH(JX,JY)=CD
      GO TO 50
   40 IF(GRAPH(JX,JY).NE.C) GO TO 43
      GRAPH(JX,JY)=B
      GO TO 50
   43 IF(GRAPH(JX,JY).NE.B) GO TO 45
      GRAPH(JX,JY)=A
      GO TO 50
   45 IF(GRAPH(JX,JY).EQ.A) GO TO 50
      GRAPH(JX,JY)=CD
      GO TO 50
   47 GRAPH(JX,JY)=SYM(I)
   50 CONTINUE
      GRAPH(NOX2,NOY2)=BORD
      WRITE(6,60) CARD
   60 FORMAT('\f',2X,20A4)
      WRITE(6,61)
   61 FORMAT(1H0,67X,"0")
      NOY1=NOY-2
      DO 80 I=3,NOY1
      IF(I.EQ.NOY2) GO TO 70
      WRITE(6,65) (GRAPH(J,I),J=1,NOX)
   65 FORMAT(1H ,20X,95A1)
      GO TO 80
   70 WRITE(6,75) (GRAPH(J,I),J=1,NOX)
   75 FORMAT(1H,16X,"270",1X,95A1," 90")
   80 CONTINUE
      WRITE(6,85)
   85 FORMAT(67X,"180")
      RETURN
      END
      program lerck
      real*8 time1,time2
      character*1 kq,irq
      character*4 icard,ir,ktime,kn,ks,ke,kw,kdepth,kmag,knum,kgap,
     1kdmin,krms,kerh,kerz,kb
      dimension icard(20),ir(11)
      data ktime,kn,ks,ke,kw/'TIME','  NN','  SS','  EE','  WW'/
      data kdepth,kmag,knum,kgap/' DTH',' MAG',' NUM',' GAP'/
      data kdmin,krms,kerh,kerz,kb/' DMN',' RMS',' ERH',' ERZ','    '/
      data kq/'Q'/
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
  100 time1=0.d+00
      no=1
    1 do 5 i=1,11
    5 ir(i)=kb
   10 read(5,15,end=40) icard,idate,ihrmn,sec,lat,xlat,lon,xlon,depth
     x,fmag,nsta,igap,dmin,rms,erh,erz,irq
   15 format(20a4,t1,i6,1x,i4,1x,f5.2,1x,i2,1x,f5.2,1x,i3,1x,f5.2
     x,2(2x,f5.2),i3,1x,i3,4f5.1,a1)
      if(icard(1) .eq. kb) go to 50
      time2=1.d+06*idate+1.d+02*ihrmn+sec
      if((time2-time1) .le. 1.d+00) ir(1)=ktime
      elat=lat+xlat/60.
      elon=lon+xlon/60.
      if(elat .gt  39.25) ir(2)=kn
      if(elat .lt. 35.75) ir(2)=ks
      if(elon .lt. 120.00) ir(3)=ke
      if(elon .gt. 123.00) ir(3)=kw
      if((depth .ge. 20.) .or. (depth .eq. 5)) ir(4)=kdepth
      if((fmag .le. 0.) .or. (fmag .gt. 3.45)) ir(5)=kmag
      if(nsta .le. 5) ir(6)=knum
      if(igap .ge. 300) ir(7)=kgap
      if((dmin .le. 0.2) .or. (dmin .ge. 50.)) ir(8)=kdmin
      if(rms .ge. 0.5) ir(9)=krms
      if(erh .ge. 10.) ir(10)=kerh
      if(erz .ge. 10) ir(11)=kerz
      if(irq .eq. kq) go to 20
      write(6,16) no,icard,ir
   16 format(i5,1x,20a4,'*',11a4)
      go to 30
   20 write(6,26) no,icard,ir
   26 format(i5,1x,20a4,1x,11a4)
   30 time1=time2
      no=no+1
      go to 1
   50 write(6,55)
   55 format('\f')
      go to 100
   40 stop
      end
```

qrychk

#
123

```
c      lp039: extract summary cards for quarry regions (11/21/71)
c      revised 5/9/72
c      revised for unix 8/2/80
      common /a/ ip,ir,iq,im,idx,idata,lat,lon
      real lat,lon,latmin,latmax,lonmin,lonmax
      character*1 ip,ir,iq,im,ib1,is,it
      character*4 idata(19),icard(30,8),ib
      dimension x(5),y(4),latmin(30),latmax(30),lonmin(30),lonmax(30),
     1klat1(30),klat2(30),klon1(30),klon2(30),xlat1(30),xlat2(30),
     1xlon1(30),xlon2(30)
      data ib,ib1,is,it/'    ',' ','*','q'/
      open(unit=4,file='qrylst',status='old',access='sequential',
     1form='formatted',blank='zero')
      open(unit=8,file='misq',status='scratch',access='sequential',
     1form='formatted',blank='zero')
      open(unit=7,file='qrycds',status='new',access='sequential',
     1form='formatted')
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
      rewind 4
      rewind 7
      rewind 8
c     input quarry list
      j=0
   10 j=j+1
      read(4,715,end=18)(icard(j,l),l=1,8),(x(l),l=1,5)
  715 format(8a4,t6,f2.0,1x,f5.2,1x,f3.0,1x,f5.2,5x,f4.0)
      latmin(j)=60.*x(1)+x(2)-x(5)
      latmax(j)=60.*x(1)+x(2)+x(5)
      lonmin(j)=60.*x(3)+x(4)-x(5)
      lonmax(j)=60.*x(3)+x(4)+x(5)
      klat1(j)=latmin(j)/60.
      klat2(j)=latmax(j)/60.
      klon1(j)=lonmin(j)/60.
      klon2(j)=lonmax(j)/60.
      xlat1(j)=latmin(j)-klat1(j)*60.
      xlat2(j)=latmax(j)-klat2(j)*60.
      xlon1(j)=lonmin(j)-klon1(j)*60.
      xlon2(j)=lonmax(j)-klon2(j)*60.
      write(6,716)(icard(j,l),l=1,8),klat1(j),xlat1(j),klon1(j),
     1xlon1(j),klat2(j),xlat2(j),klon2(j),xlon2(j)
  716 format('\f',10x,8a4,10x,'region bounded by: ',4(i5,'-',f5.2))
      go to 10
   18 nq=j-1
c     input quake list (skip night data: 3<hour<13)
      n=0
   20 n=n+1
   25 read(5,725,end=300)(idata(i),i=1,19),ip,ir,iq,im,y,ihr
  725 format(19a4,4a1,t19,f2.0,1x,f5.2,1x,f3.0,1x,f5.2,t8,i2)
      if(idata(1) .eq. ib) go to 300
      if((ihr .gt. 3) .and. (ihr .lt. 13)) go to 25
      lat=60.*y(1)+y(2)
      lon=60.*y(3)+y(4)
      idx=0
      do 80 j=1,nq
      if((lat .lt. latmin(j)) .or. (lat .gt. latmax(j))) go to 80
      if((lon .lt. lonmin(j)) .or. (lon .gt. lonmax(j))) go to 80
      write(6,730) (idata(l),l=1,19),ip,ir,iq,im
  730 format(10x,19a4,4a1)
c     punch cards for replacements
      if((ir .eq. is) .or. (ir .eq. it)) go to 60
c     add 'q' to ir
      ir=it
      go to 70
```

```
c     'q' or '*' already exists,change im to blank
   60 im=ib1
   70 write(7,735)(idata(l),l=1,19),ip,ir,iq,im
  735 format(19a4,4a1)
      idx=1
   80 continue
   85 continue
      if((ir .ne. it) .or. (idx .eq. 1)) go to 20
      write(8,730)(idata(l),l=1,19),ip,ir,iq,im
c     change ir to blank
      ir = ib1
      write(7,735)(idata(l),l=1,19),ip,ir,iq,im
      go to 20
  300 rewind 8
      write(6,775)
  775 format('\f',' ***** mis-identified quarries*****')
  310 read(8,730,end=400)(idata(l),l=1,19),ip,ir,iq,im
      write(6,730)(idata(l),l=1,19),ip,ir,iq,im
      go to 310
  400 stop
      end
```

catprog

#2

```
C     PROGRAM CATPROG TO PREPARE CATALOG OUTPUT FROM SUMMARY CARD INPUT
      REAL MAG
      REAL*8 TIME, TIMEX
      CHARACTER*1 RMK1, RMK2, Q, IC, SYM, QU, STAR, B1, QUEUE
      CHARACTER*3 MONTH, MON, MBLANK
      CHARACTER*4 TITLE1, TITLE2, ERH, IRZ, BLANK, FINI, ALPHA
      CHARACTER*8 BLANKS, ERR, ERROR
      CHARACTER*32 QUAD, QBLANK
      CHARACTER*80 A
      DIMENSION TITLE1(20), TITLE2(20), MONTH(12)
      DATA ERR, BLANKS, BLANK/'***ERROR', '        ', '    '/
      DATA QBLANK/'                                '/
      DATA MONTH/'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG'
     X, 'SEP', 'OCT', 'NOV', 'DEC'/
      DATA STAR/'*'/, QUEUE/'Q'/, MBLANK/'   '/, FINI/'****'/
      DATA B1/' '/
      open(unit=2, status='scratch', access='direct', form='unformatted',
     1recl=50)
      open(unit=4, file='quadlst', status='old', access='sequential',
     1form='formatted', blank='zero')
      open(unit=5, access='sequential', form='formatted', blank='zero')
      rewind 5
      rewind 4
      READ(5, 110)LAT, XLAT, LON, XLON, DELLAT, DELLON, IMAX, JMAX
      XLATO=60. *FLOAT(LAT)+XLAT
      XLONO=60. *FLOAT(LON)+XLON
 110  FORMAT(I2, F6. 2, 1X, I3, F6. 2, 2X, 2F5. 1, 2I2)
 203  FORMAT(' ', 'LATO= ', I3, '-', F5. 2, '  LONO= ', I4, '-', F5. 2,
     1'DLAT, DLON= ', 2F5. 2, ' IMAX, JMAX= ', 2I3)
C     READ IN TITLE CARDS
    1 READ(5, 3) TITLE1
      READ(5, 3) TITLE2
    3 FORMAT(20A4)
      READ(5, 4)IPRT, KPAGE
    4 FORMAT(I1, 1X, I3)
      WRITE(6, 203) LAT, XLAT, LON, XLON, DELLAT, DELLON, IMAX, JMAX
      WRITE(6, 3)TITLE1
      WRITE(6, 3) TITLE2
      WRITE(6, 5)IPRT, KPAGE
    5 FORMAT(5X, 'IPRT=', I3, 5X, 'KPAGE=', I4)
  101 IPAGE=KPAGE
      KSTP=0
      GO TO 200
    6 CONTINUE
      TIME=0.
      L=-1
      MM=0
C     INPUT NEW SUMMARY CARD
   10 READ(5, 15, END=400) IY, MO, ID, KHR, KMIN, SEC, LAT, XLAT, LON, XLON, RMK1
     X, DEPTH, RMK2, MAG, NO, IGAP, DMIN, RMS, ERH, ERZ, Q, IRZ, IC, SYM, QU
   15 FORMAT(3I2, 1X, 2I2, F6. 2, I3, 1X, F5. 2, I4, 1X, F5. 2, A1, F6. 2, A1, 1X, F5. 2
     X, I3, I4, F5. 1, F5. 2, 1X, A4, F5. 1, 1X, A1, T74, A4, T45, A1, T80, A1, T78, A1)
      IF(IY. LT. 0) GO TO 100
      IF (QU . EQ.  QUEUE) GO TO 10
      IF (SYM . NE.  STAR) SYM=B1
      IF (ERZ . GE.  99. 9) IRZ=BLANK
      IF (NO . LE.  3) RMK2=STAR
C     CHECK CHRONOLOGICAL SEQUENCE
      ERROR=BLANKS
      TIMEX=SEC+1. D02*KMIN+1. D04*KHR+1. D06*ID+1. D08*MO+1. D10*IY
      IF (TIMEX. GT. TIME) GO TO 18
      ERROR=ERR
   18 TIME=TIMEX
C     45 LINES PER PAGE
```

```
      L=L+1
      LL=MOD(L,45)
      IF (LL.NE.0) GO TO 44
      GO TO 405
  400 KSTP=1
  405 WRITE(6,19) IPAGE
   19 FORMAT(//,50X,I3)
      IF(KSTP .EQ. 1) GO TO 600
      IPAGE=IPAGE+1
      IF (L.NE.0) GO TO 424
      WRITE(6,422) TITLE1
  422 FORMAT('\f',     10X,20A4)
      GO TO 428
  424 WRITE(6,426) TITLE2
  426 FORMAT('\f',     10X,20A4)
  428 WRITE(6,43) IY
   43 FORMAT(/, 8X,'19',I2,'  HR MN   SEC   LAT N    LONG W   DEPTH',
     X'  MAG  NO GAP  DMIN   RMS  ERH  ERZ Q  QUADRANGLE')
      GO TO 45
   44 MON=MBLANK
      IF (MO.EQ.MM) GO TO 50
   45 MM=MO
      MON=MONTH(MO)
C     EXTRA LINE FOR EVERY FIVE LINES
   50 LL=MOD(L,5)
      IF (LL.NE.0) GO TO 54
      WRITE(6,51)
   51 FORMAT(/)
 54   CONTINUE
      GO TO 500
   52 CONTINUE
      WRITE(6,55) IC,IC,MON,ID,KHR,KMIN,SEC,LAT,XLAT,LON,XLON,RMK1,DEPTH
     X,RMK2,MAG,
     Y          NO,IGAP,DMIN,RMS,ERH,IRZ,Q,SYM,QUAD,ERROR
   55 FORMAT(5X,2A1,1X,A3,3I3,F6.1,I4,'-',F4.1,I5,'-',F4.1,A1,F5.1,A1
     X,F5.1,1X,I3,I4,
     Y          F6.1,F6.2,1X,A4,1X,A4,1X,2A1,1X,A32,1X,A8)
      GO TO 10
  100 WRITE(6,105)
  105 FORMAT('\f')
      IF(IY.LT.0) GO TO 1
  200 IF(IPRT .EQ. 0) GO TO 210
      WRITE(6,272)
  210 READ(4,268,END=6)A,ALPHA
      IF(ALPHA .EQ. FINI) GO TO 6
      READ(A,269)QUAD,LAT,XLAT,LON,XLON
  268 FORMAT(A80,T1,A4)
  269 FORMAT(A32,29X,I2,F4.1,2X,I3,F4.1)
      I=(60.*FLOAT(LAT)+XLAT-XLATO)/DELLAT+1.00001
      J=(60.*FLOAT(LON)+XLON-XLONO)/DELLON+1.00001
      IF(I.LE.0.OR.I.GT.IMAX) GO TO 230
      IF(J.LE.0.OR.J.GT.JMAX) GO TO 230
      IJX=(I-1)*JMAX+J
      WRITE(2,REC=IJX)QUAD,LAT,XLAT,LON,XLON
      IF(IPRT .EQ. 0) GO TO 210
      WRITE(6,270) I,J,QUAD,LAT,XLAT,LON,XLON
      GO TO 210
C
  230 IF(IPRT .EQ. 0) GO TO 210
      WRITE(6,271) I,J,QUAD,LAT,XLAT,LON,XLON
      GO TO 210
  270 FORMAT(' ',2I5,5X,A32,33X,I2,'-',F5.2,2X,I3,'-',F5.2)
 271  FORMAT(' ',2I5,5X,A32,33X,I2,'-',F5.2,2X,I3,'-',F5.2,
     1 '   OUTSIDE DEFINED AREA')
 272  FORMAT('\f',' INDEX    ',T43,'QUAD',T82,'LAT',T93,'LON')
  500 IQ=(60.*FLOAT(LAT)+XLAT-XLAT0)/DELLAT+1.00001
      JQ=(60.*FLOAT(LON)+XLON-XLON0)/DELLON+1.00001
      IF(IQ.LE.0.OR.IQ.GT.IMAX) GO TO 510
      IF(JQ.LE.0.OR.JQ.GT.JMAX) GO TO 510
C
      IJX=(IQ-1)*JMAX+JQ
      READ(2,REC=IJX)QUAD,KAT,XKAT,KON,XKON
      GO TO 52
C
 510  IQ=0
      JQ=0
      QUAD=QBLANK
      GO TO 52
  600 STOP
      END
      program srthyp
      double precision tm
      character*80 a,ihead
      character*1 ld,lcd,itst
      dimension ida(12),tm(1000),idx(1000)
      data ld/'d'/,lcd/'D'/
      data ida/0,31,59,90,120,151,181,212,243,273,304,334/
      open(8,status='scratch',access='direct',
     1 form='formatted',recl=80,blank='zero')
      open(unit=5,access='sequential',form='formatted',blank='zero')
      rewind 5
      iph=0
      i=0
    1 continue
      read(5,90,end=300) a,itst
      if(itst .eq. ld .or. itst .eq. lcd) go to 10
   90 format(a80,t2,a1)
  100 format(a80)
      i=i+1
      read(a,120)kyr,kmo,kdy,khr,kmn,sec
  120 format(3i2,1x,2i2,1x,f5.2)
      jdy=365*kyr+ifix((kyr-1)/4. )-29219
      jdy=jdy+ida(kmo)+kdy
      mod=kyr-ifix(kyr/4. )*4
      if((kmo .gt. 2) .and. (mod .eq. 0)) jdy=jdy+1
      ktm=86400*jdy+3600*khr
      if(i .gt. 1) go to 200
      ktm1=ktm
  200 tm(i)=float(ktm-ktm1)+float(60*kmn)+sec
      ind=i
      write(8,100,rec=ind) a
      go to 1
   10 iph=1
      ihead=a
      go to 1
  300 nrec=i
      call sort(tm,idx,nrec)
      if(iph .eq. 0) go to 400
      write(6,100) ihead
  400 do 450 m=1,nrec
      ind=idx(m)
  130 format(2i5,f10.1)
      read(8,100,rec=ind) a
      write(6,100) a
  450 continue
      stop
      end
      subroutine sort(x,key,no)
      double precision x
      dimension x(no),key(no)
      do 1 i=1,no
1     key(i)=i
      mo=no
2     if (mo-15) 21,21,23
21    if (mo-1) 29,29,22
22    mo=2*(mo/4)+1
      go to 24
23    mo=2*(mo/8)+1
24    ko=no-mo
      jo=1
25    i=jo
26    if (x(i)-x(i+mo)) 28,28,27
27    temp=x(i)
      x(i)=x(i+mo)
```

#6
129

```
      x(i+mo)=temp
      kemp=key(i)
      key(i)=key(i+mo)
      key(i+mo)=kemp
      i=i-mo
      if (i-1) 28,26,26
28    jo=jo+1
      if (jo-ko) 25,25,2
29    return
      end
```