DEPARTMENT OF THE INTERIOR
U. S. GEOLOGICAL SURVEY

Auxiliary Programs For Support
of Seismic Hazard Analysis

by

Stanley L. Hanson[1] and David M. Perkins[1]

Open-File Report 85-615

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

1 USGS
Denver, Colorado

1985

CONTENTS

ILLUSTRATIONS

Tables

AUXILIARY PROGRAMS FOR SUPPORT OF SEISMIC HAZARD ANALYSIS

by Stanley L. Hanson and David M. Perkins

Introduction

Bender and Perkins (1982) presented documentation of a computer program (SEISRISK II) used within the U.S. Geological Survey to calculate probablistic ground-motion values for seismic-hazard mapping (Algermissen and others, 1982). However, considerable data manipulation and analysis is required prior to executing the SEISRISK II program.

This report presents documentation for a system of auxiliary programs used in earthquake catalog manipulation, determination of seismic-rate parameters, and preparation of input data for SEISRISK II, as well as post-processing of the program output in support of seismic-hazard analysis and mapping. (See fig. A for brief flowchart of these auxiliary programs.)

Data pre-processing generally consists of digitizing a set of seismic source zones and faults that are to be modeled, sorting of earthquake catalog data into the seismic source zones, investigation of earthquake catalog completeness, and statistical analysis of earthquake recurrences.

While programs for digitizing and sorting are generally straightforward, requiring little or no interpretive intervention, programs for estimating earthquake catalog completeness and analyzing of earthquake recurrences are not. The programs to achieve these later tasks are designed to present, at these interpretive intervention points in the system, tables and graphic data needed to accommodate the necessary analytical judgments. At these intervention points in the system, optional data presentations or solutions are provided and are based on either standard statistical techniques or different approaches to the problem being addressed at that step in the system. The options are not intended to be exhaustive, but rather, techniques we have found useful in a number of applications. Responsibility lies with the analyst to: (1) select the appropriate solution within the limitations of his data set for continuation within the system, (2) reiterate preceding steps within the system if necessary using, perhaps, different subsets of the original data set, and (3) providing some of the optional solutions are acceptable, terminate the analysis.

Post-processing of data from SEISRISK II are plotting routines to display the ground-motion values.

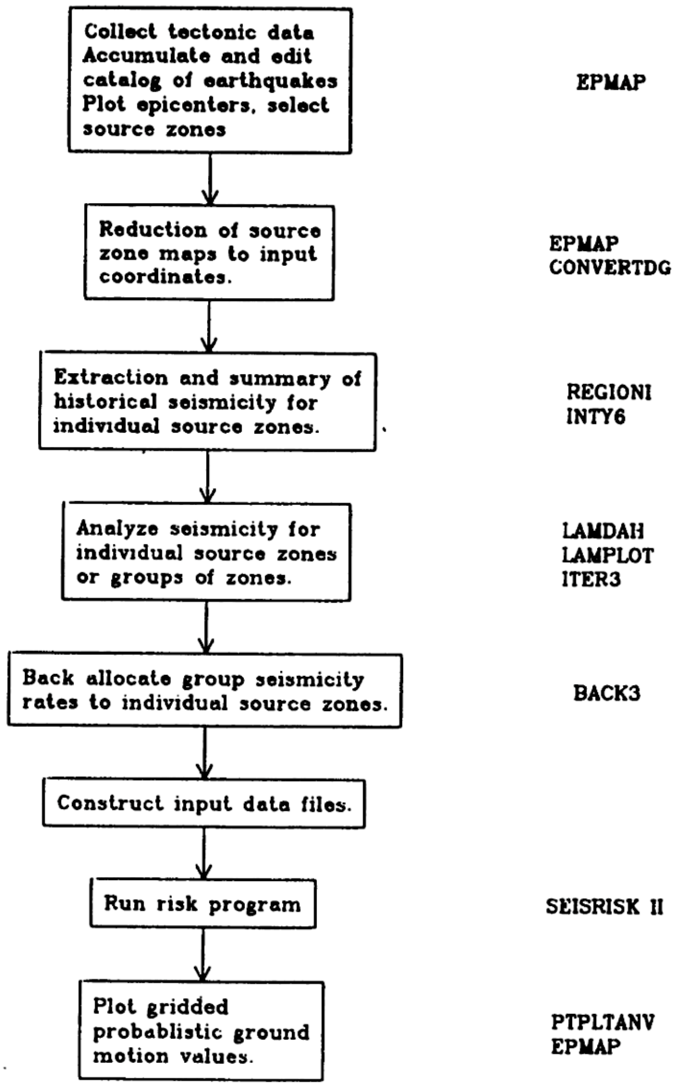

Figure A.--System flowchart diagram showing briefly the order of the steps in the system and the programs used at each step.

System Description

Seismic Source Zones

The pre-processing system requires a hard-copy map of seismic source zones. The source zones provide limits to the geological distribution of earthquakes in a two-dimensional space and are based on a number of geological and seismological considerations (Thenhaus, 1983). The initial step in the system is an operator intervention. (Throughout this report an operator intervention is defined to be any hand manipulation of the data, selection, or a decision, either analytical or empirical, that must be performed according to human, rather than computer judgments.) This first operator intervention subdivides the given source zones into the quadrilaterals, which define the source zone to SEISRISK II. The quadrilaterals are also needed in the auxiliary programs to extract source zone earthquakes from the data catalog. The task is accomplished with a pencil and a straightedge. The quadrilaterals within a source zone should be planned and laid out in such a way as to minimize their total number, so as to best approximate the source zones (see fig. B). The quadrilaterals, through any given source zone, must form a continuous chain from one end of the zone to the other, much like the rungs on a ladder (see fig. C). If a continuous chain is not possible, then the zone is divided into two or more sets of quadrilaterals, and a new starting point is necessary (see fig. D). This break is signaled in the input file. (See the documentation for the program CONVERTDG.FOR, for an example of this zone, named i003.)

Upon completion of the drawing of the quadrilaterals within source zones, they should be checked for any obvious defects, such as quadrilaterals inadvertently drawn with 3 or 5 sides, or quadrilaterals that form a continuous chain linking each quadrilateral to the next by adjacent sides rather than by opposite sides as required. Interior angles cannot be greater than, or equal to, 180 degrees.

At this time, faults that will be modeled in SEISRISK II should be approximated as jointed line segments. If it is desired to collect seismicity that has occurred in the vicinity of these faults, narrow source zones should be designed for this purpose and quadrilateralized.

The quadrilateralized source zones and jointed fault lines are now ready for digitizing. A computer system utility program exists on the VAX that can be used to digitize the quadrilateral corner points and convert the points into latitude and longitude pairs. This program will create a data file of points that will be processed by the program CONVERTDG.FOR. There is a certain order in which the data is to be digitized (see program documentation for CONVERTDG.FOR and the digitizing programs to obtain the proper file format).

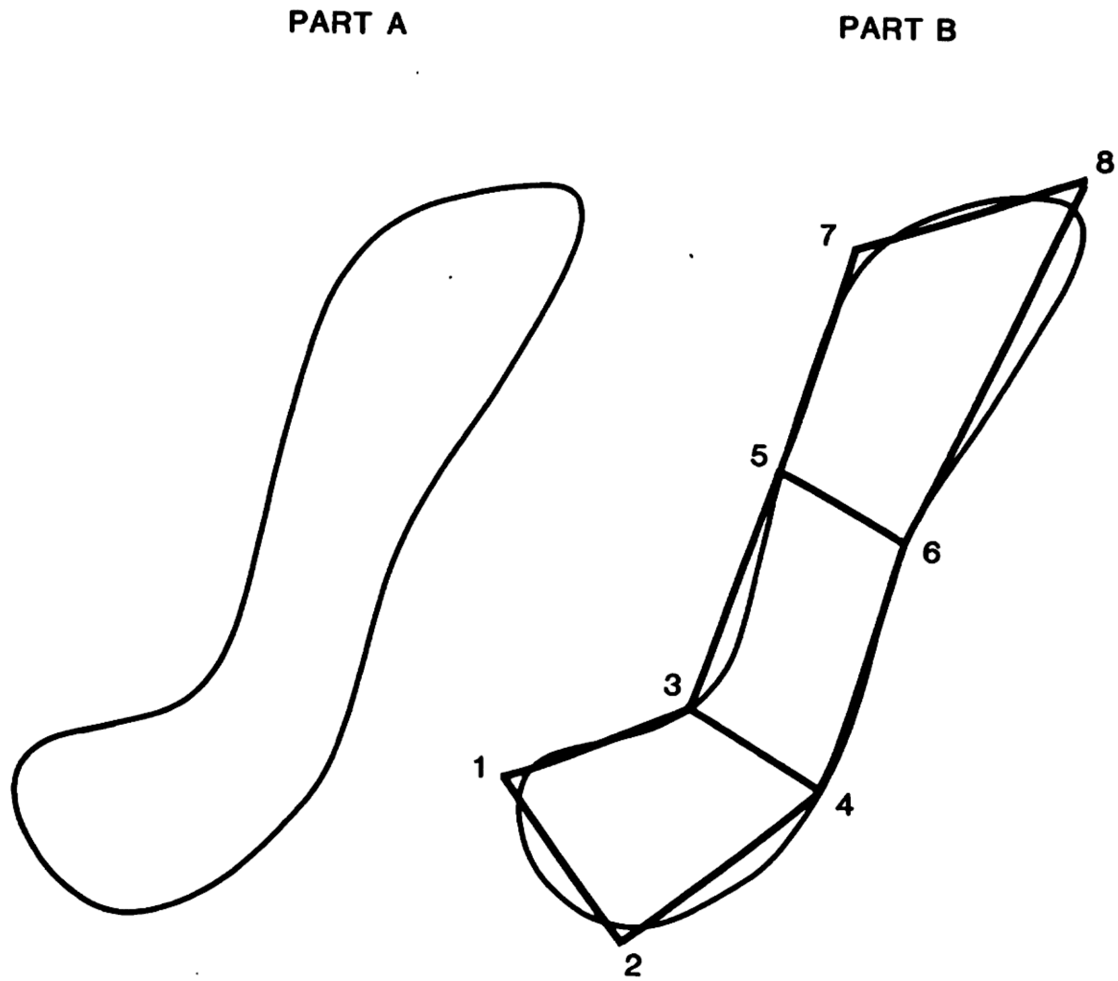

Figure B.--Part A, sample source zone. Part B, sample source approximated by quadrilaterals.

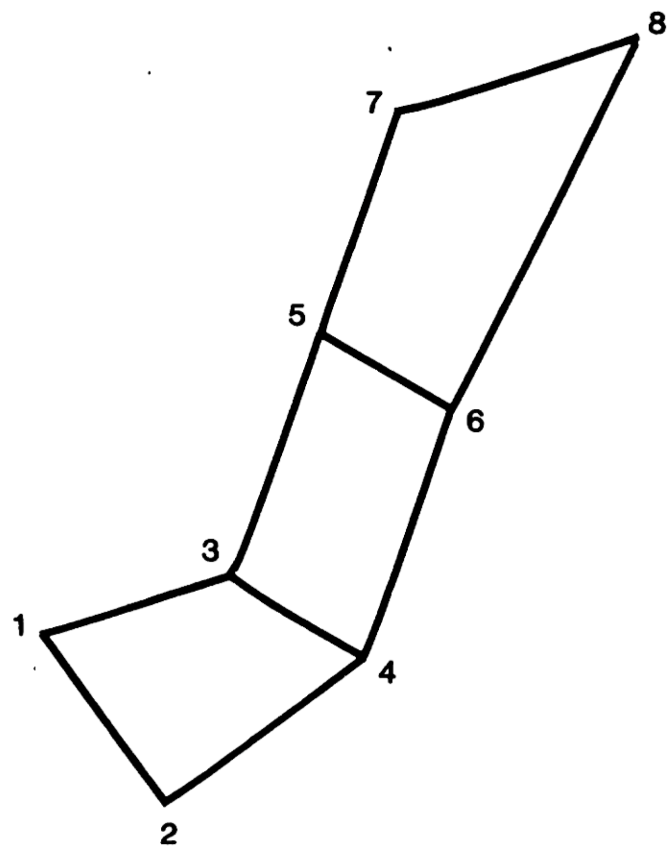

Figure C.--The corner points of each of the quadrilaterals are ordered in such a way as to form the rungs of a ladder. This zone is defined by the four pairs of points: (1,2),(3,4),(5,6),(7,8).

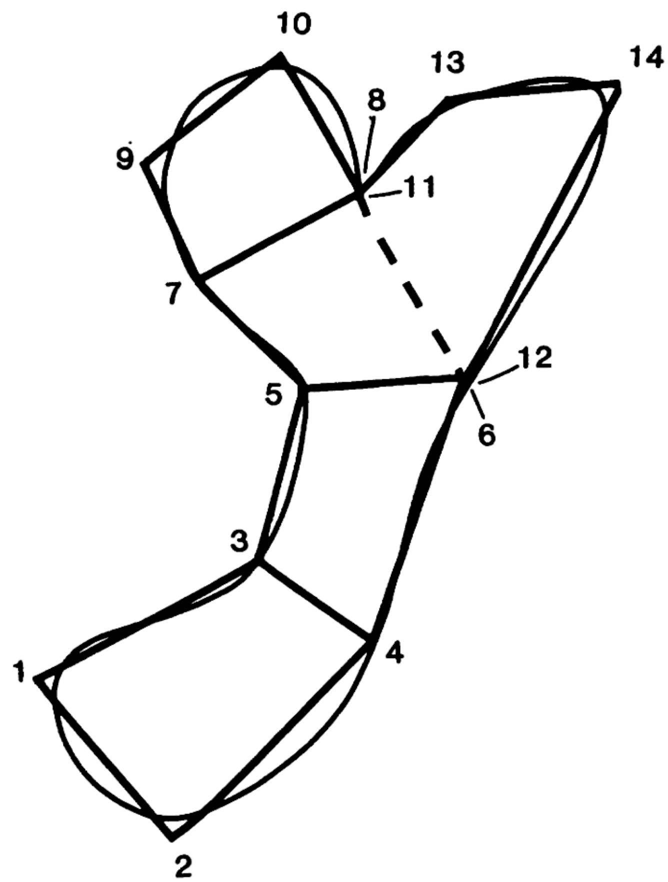

Figure D.--This source zone is best represented by two sets of quadrilaterals, the set on the left consisting of the points (1,2),(3,4),(5,6),(7,8),(9,10), and the set on the right consisting of the points (11,11),(13,14).

The program CONVERTDG.FOR reads the digitized data file and produces one file in the SEISRISK II program input format (Bender and Perkins, 1982) that contains all the zones and faults. At this point the file can be plotted using program EPMAP.FOR (to be described later) to compare the digitized seismic source zone boundaries, quadrilaterals, and faults with the originals.

Earthquake Data Collection

Following verification of seismic source zone and fault-source input, the program REGIONI.FOR reads the earthquake catalog (or catalogs) and the output file of CONVERTDG.FOR. REGIONI.FOR produces a catalog of earthquakes for each zone processed. These smaller catalogs are sorted into chronologic order using a computer system utility sort program.

The catalogs can easily be examined for foreshock-aftershock sequences, earthquake swarm activity, or other characteristic traits of seismicity within the source zones. Foreshock-aftershock sequences and events less than some minimum magnitude of interest can be edited out. Another operator intervention in the system is achieved through a local computer system editor. The zone catalogs are then processed by the program INTY6.FOR. Output from INTY6.FOR is a summary table of earthquake activity within each source zone. It shows, decade-by-decade, the number of earthquakes as a function of intensity categories (table 1). We will subsequently refer to these tables as "decade tables." These decade tables are used to estimate completeness and seismic rates by the program described below. We have found the decade table summaries (table 2) useful in assessing the presence of foreshock or aftershock sequences to be searched for in the source zone catalogs.

Table 1
Decade Table

0	1	0	0	0	0	0	0
21	2	0	0	0	0	0	0
24	5	0	1	0	0	0	0
2	2	0	0	0	0	0	0
26	7	5	1	0	0	0	0
3	1	0	0	0	0	0	0
0	3	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 2
Summary Decade Table

```
-----------------------------------------------------------------
 Enter the entire input file name (XXXXXXNNN.dat) or quit to stop the run.
>ecnewi106.dat
 decade       V     VI     VII   VIII     IX    X      XI   XII  TOTAL


 1870-1879    1      0      2      0      0      0      0      0    3

 1900-1909    1      0      0      0      0      0      0      0    1

 1910-1919    0      3      2      1      0      0      0      0    6

 1920-1929    3      1      0      0      0      0      0      0    4

 1930-1939   26      7      5      1      0      0      0      0   39

 1940-1949    2      2      0      0      0      0      0      0    4

 1950-1959   24      5      0      1      0      0      0      0   30

 1960-1969   21      2      0      0      0      0      0      0   23

 1970-1973    0      1      0      0      0      0      0      0    1

 total number of earthquakes with intensity INT > or = V = 111
 Enter the entire input file name (XXXXXXNNN.dat) or quit to stop the run.
>quit
 FORTRAN STOP
-----------------------------------------------------------------
```

Earthquake Catalog Recurrence Rates and Completeness

As can be seen from table 1, not all of the intensity categories are completely reported for the entire length of the catalog. Moderate intensities are more frequently reported for more recent decades and reflect more complete reportings at these levels due to perhaps increased settlement of the region, or improved seismic monitoring, or both. A significant bias would be introduced if a rate of moderate intensity earthquake activity were to be calculated by averaging the higher activity of recent decades with previous decades of no reported activity of this size. We therefore need to identify the period of time for which each intensity category is completely reported and calculate average rates of occurrences on this complete set of data. A procedure for doing this is given by (Stepp, 1973).

The following two programs were developed on his approach to (1) calculate the mean rate of occurrence and the standard deviation of occurrence for each intensity level, and (2) to plot these values so the occurrence rates and completeness times can more easily be estimated (fig. E). The program LAMDAH.FOR uses the decade tables as input and calculates the mean and standard deviation of the rates in table form. These tables become the input to LAMPLOT.FOR, which produces plots that represent rates per decade for each intensity level for each zone or set of zones that have been combined together. Another operator intervention is needed at this time to decide what are the appropriate rates and completeness times per intensity level from the plots for each zone or groups of zones.

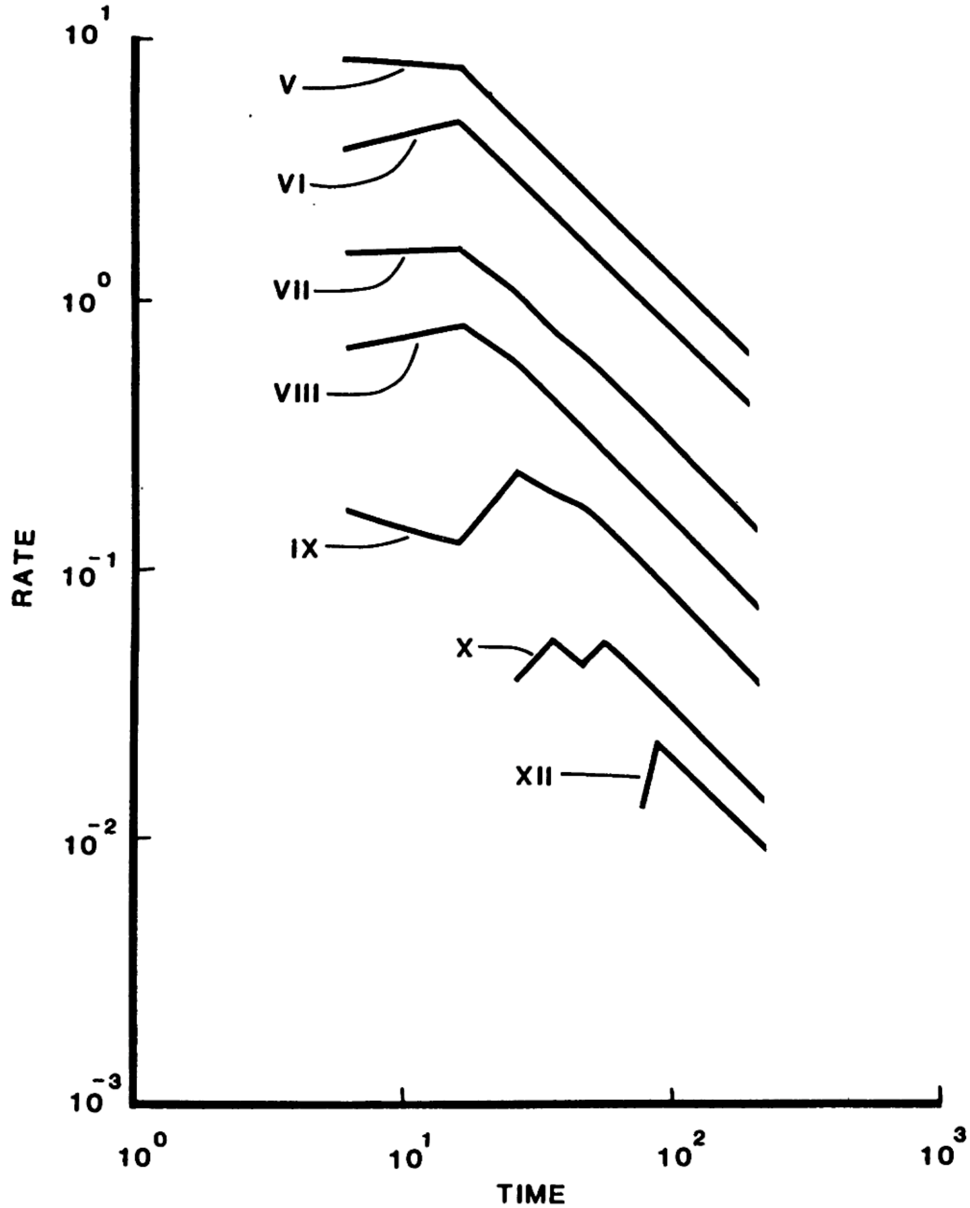

Figure E.--Sample computer plot of mean occurrence rate per decade plotted on a log-log graph, redrafted here for clarity.

Earthquake Frequency Calculation

ITER3.FOR is a statistical program that will calculate a b-value, an a-value for the relationship LogN=a-bI, and the predicted rates from the input expected values selected from the plots produced by LAMPLOT.FOR. (There is no documentation for ITER3.FOR as yet.) In ITER3.FOR, the predicted rates are calculated a variety of ways, and the operator must select the rates that are desired the most suitable.

If rates are determined for an ensemble of source zones, the ensemble rate must be redistributed amongst the constituent zones. For this purpose, BACK3.FOR accepts as input the ensemble rate and the decade table summary for each constituent zone.

This program accepts the input from the terminal and produces six different tables of rates. These tables represent the redistribution, "back allocation," of the input rate into each individual source zones using six different methods (see Redistribution of Rates). The operator must again intervene and select the method that in his opinion best represents the data collected.

At this point the risk input data file created earlier, containing source zones and faults, is edited adding the appropiate attenuation function table (acceleration or velocity) and the rates for each zone selected from the output of BACK3.FOR above.

The input is now ready for SEISRISK II (Bender and Perkins, 1982) which produces probabilistic ground motion values as provided by the input attenuation function table.

Post-Processing

Ground motion values for selected probability levels are output in a binary file from SEISRISK II which is the input to the program PTPLTANV.FOR. This post-processing program produces a line printer plot of the data and an ASCII data file which is then plotted geographicaly by EPMAP.FOR.

EPMAP.FOR will produce a geographical map plot of the acceleration or velocity values that is suitable for contouring.

This program can also be used to plot an epicenter map to assist the geologist in producing a seismic source zone map in step one of this system. It will also produce a plotted map of the seismic source zones after they have been put into the risk program input format to check against the original map.

Before proceeding on to the individual program documentation sections, the user may wish to look at figure F for a more detailed look at the entire system, and see how and where all of the programs fit into the entire system.

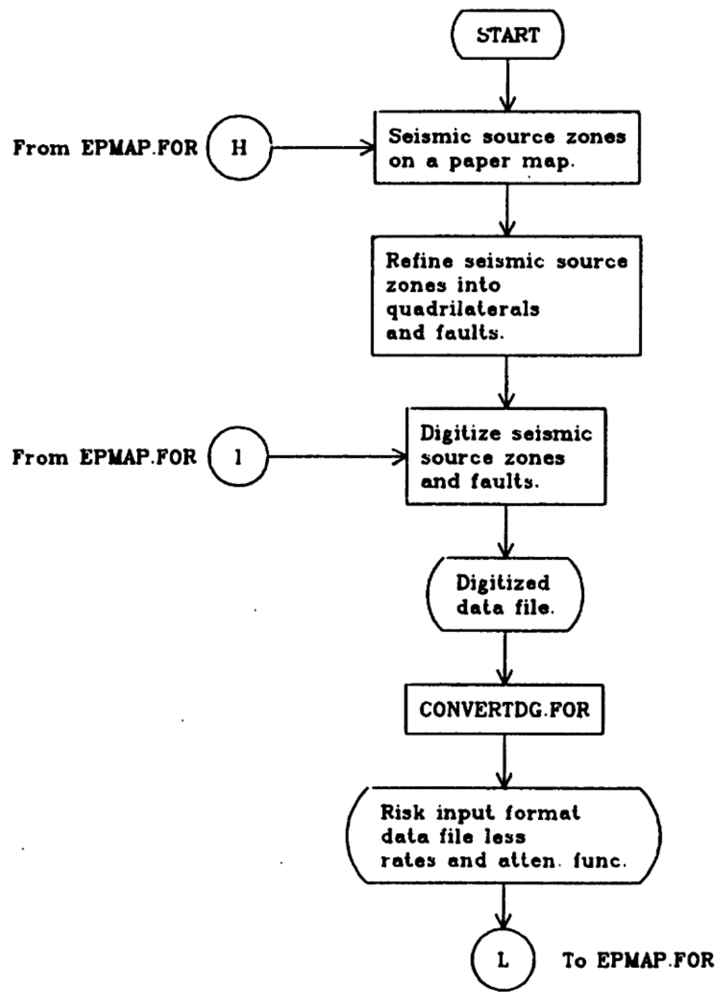

Figure F.--Part A, an expanded view of the first and second processes of Fig. A and step 1 of the system description.

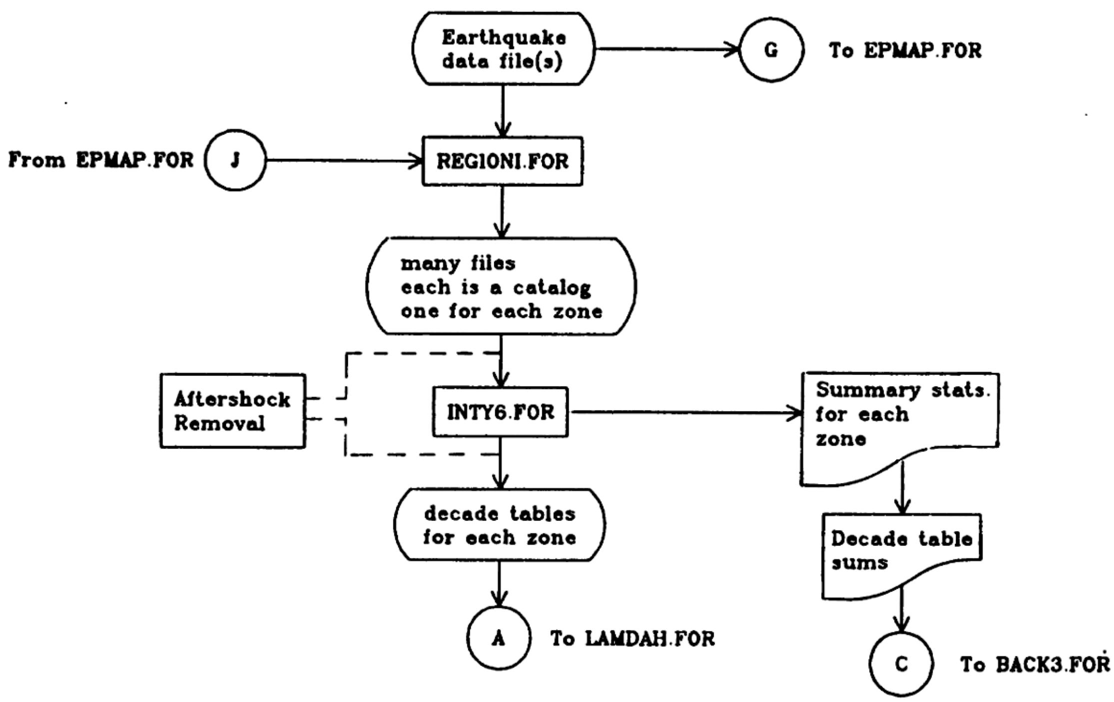

Figure F.--Part B, an expanded view of the third process in Fig. A and step 2 of the system description.

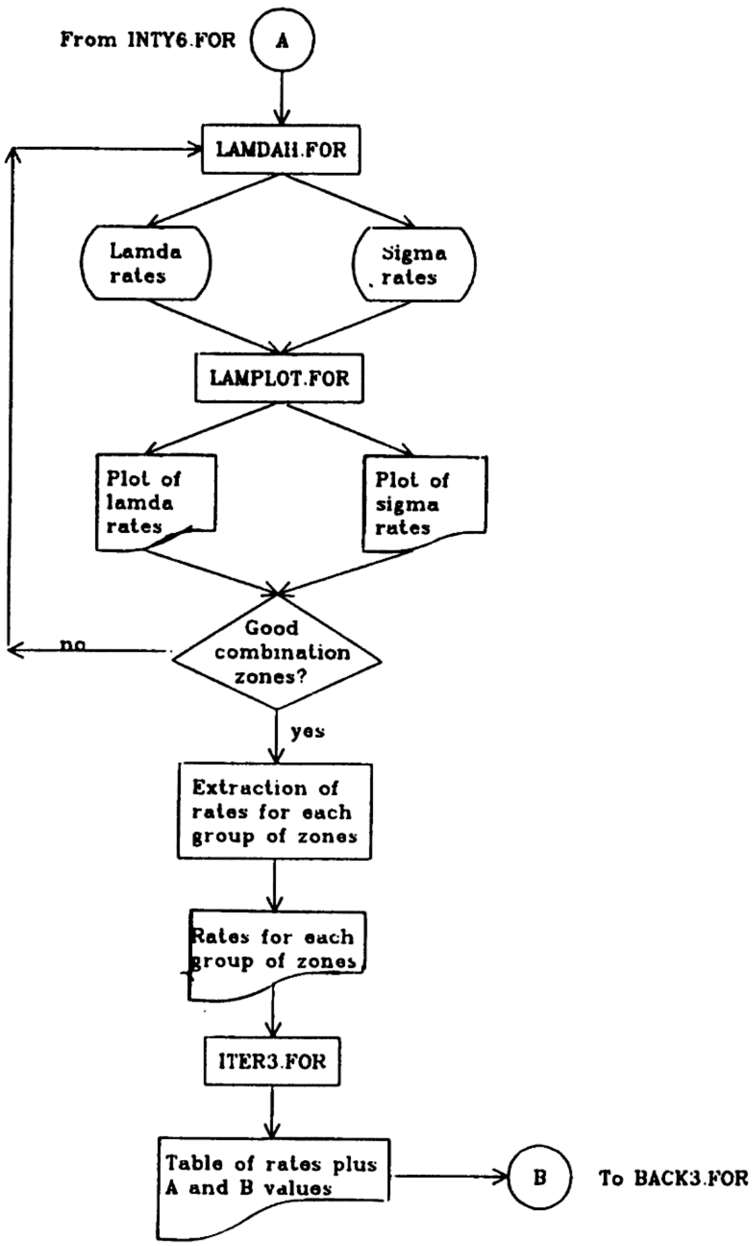

Figure F.--Part C, an expanded view of the fourth process of Fig. A and steps 3 and part of step 4 of the system description.

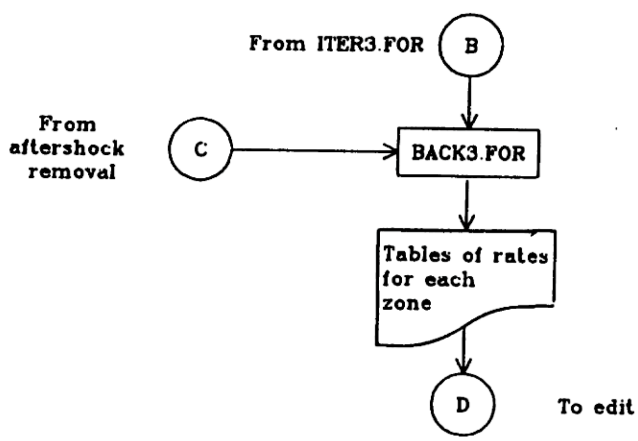

Figure F.--Part D, an expanded view of the fourth process of Fig. A and the last of step 4 of the system description.

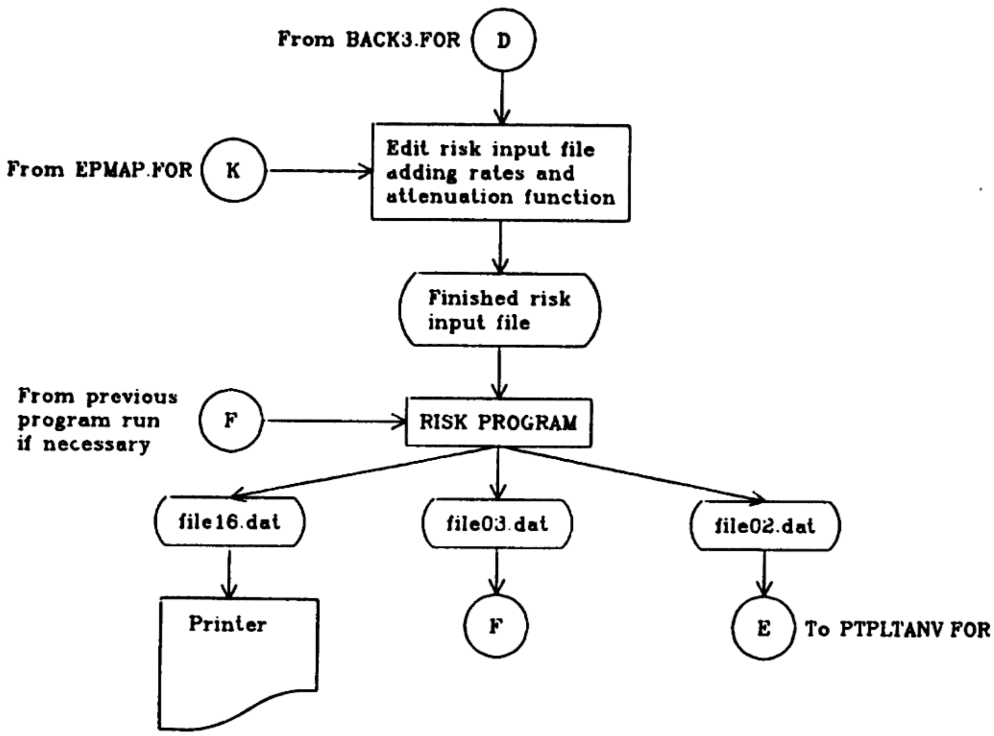

Figure F.--Part E, an expanded view of the sixth and seventh processes of Fig. A. Editing the risk program input file and executing the risk program.

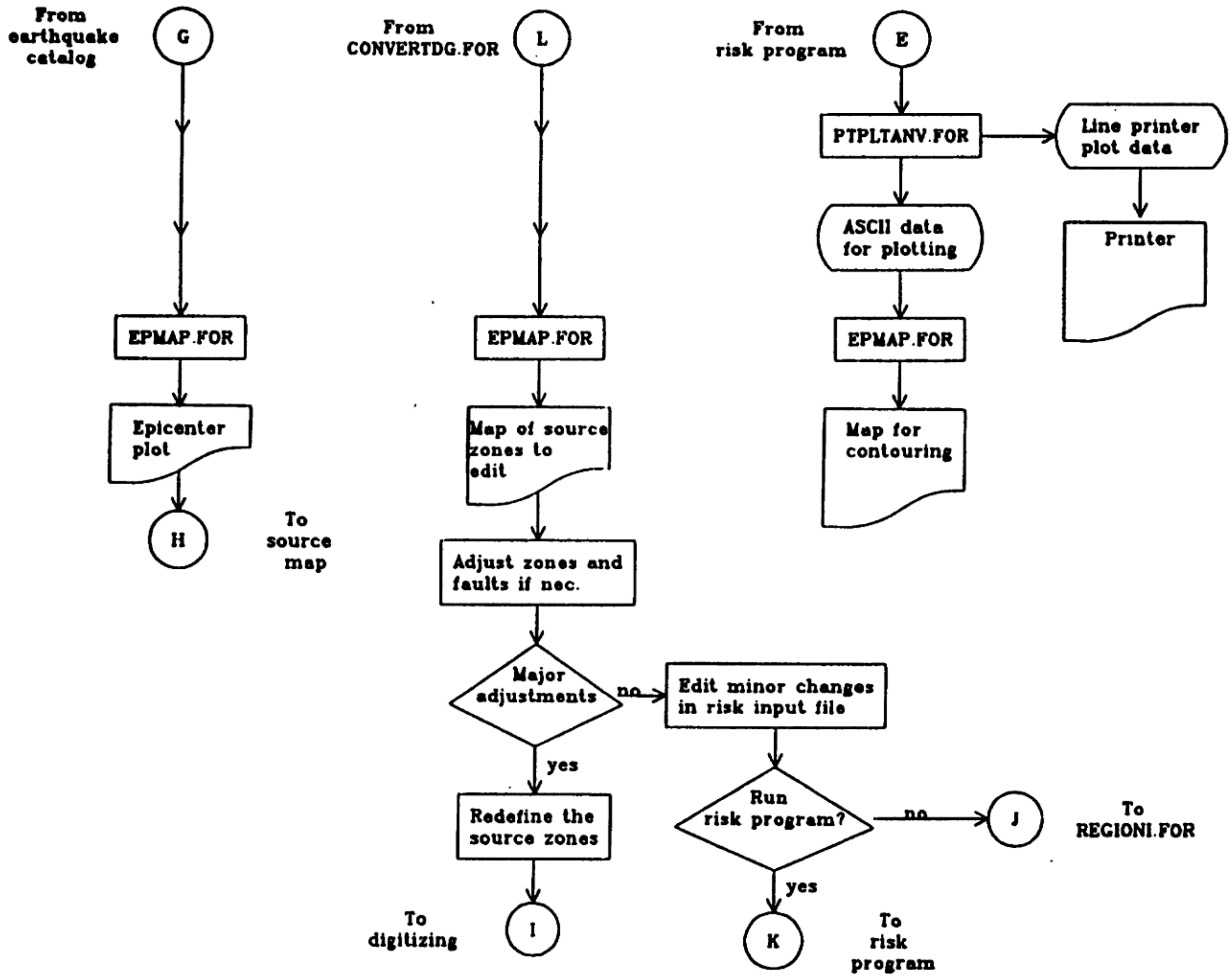

Figure F.--Part F, an expanded view of the eighth process of Fig. A and step 5 of the system description.

Plotting Routine

Description

EPMAP.FOR is an interactive, geographic-map plotting program that will produce one of three different "working plots." It is used first in producing epicenter maps for assistance in drawing source zones (fig. F, see part F connector E). It is used later to plot digitized source zones and faults as a check on the accuracy of the quadrilateralizing, segmenting, and digitizing (fig. F, see part F connector L). Finally it is used to plot gridded probablistic ground motion values for contouring ground motion hazard maps (fig. F, see part F connector E). The maps can be produced in one of three different projections, Albers, Lambert, and Mercator, to suit the user's needs. These plots are accomplished through the use of the BUPLOT subroutines on the VAX/VMS computer.

Input Files Description

There are four input files required for EPMAP.FOR, one of which is optionally input from the terminal. The remaining three files must already exist in the working directory.

These four files are (1) a parameter file, (2) a point-value file, (3) a geographical boundary data file, and (4) a source zone and faults file. The geographical boundary data file and the source zone data files have fixed names NA.DAT and REGION.INC, respectively, and particular formats. The parameter file and the point-value files have variable names with the point-value file having a variable format input by the user.

The parameter file, containing the geographic plotting window information, is the optionally input file. That is, it can be entered from the terminal at execution time, through answers to a seies of questions, and then stored in a file name specified by the user and used again at a later time, or read in from a previously created file. (See appendix A at the end of this section for an example.)

The point-value data file contains a series of one or more data which are to be plotted at particular geographic locations. Each data record contains the latitudes and longitudes of a point and a "third" value to be plotted at that point. This third value can be plotted as a symbol for magnitude or intensity values for an epicenter map, or as a literal string such as ground motion values or as elevation values for a contour map. (See appendix B at the end of this section for an example.)

The geographical boundary data file (see appendix C at the end of this section for an example) is a file containing digitized geographical data and political boundaries.

The file containing the latitudes and longitudes that define source zones and faults, REGION.INC (see appendix D at the end of this section for an example), is the fourth and final input file.

Output File Description

The output is a metacode plot file that can be plotted on a graphics terminal or on a hard copy plotting device. The output file that is created is called pltfl.dat by default and is overwritten with each execution of the program.

Program Operation

The program is interactive in order to add the flexibility of being able to use the same program to accept and plot different data pertinent to the seismic hazard system described previously. The program first queries the user for the plot window and scaling information. This may be input from the terminal or from an existing file whose file name maybe arbitrary. The next query is for the file name containing the point-value data. It now asks for the FORTRAN format of the point-value data, which must be of the form: Latitude, Longitude, Third-parameter.

The program now expects to find the two remaining data files in the working directory, one containing the geographical boundaries, NA.DAT, and the other, REGION.INC, containing the source zones and faults.

It should be made clear at this point that although all these files are required, two of the four input files, REGION.INC and the point-value data file (call it PTS.DAT) can be altered in such a way that their contents are not produced on the plot. For example creating a file with no source zones and faults, a "dummy" REGION.INC file, will result in a plot with no source zones (see appendix E at the end of this section for an example). Likewise a data file with no epicenters, a "dummy" data set (say PTS.DAT), will result in a plot with no symbols because the location of the point is outside the plot window (see appendix F at the end of this section for an example). Combining this information with the parameter file information the program will produce a working plot of either epicenter data, source zones, or ground motion values, or a combination of epicenters and source zones. or ground motion values and source zones.

The plotting of epicenter data also requires a switch to be set in the parameter file to get the correct subroutine to read the data (see the subroutine TTYIN.FOR in the program listing for the switch number, appendix G). The three subroutines that read the data are similar, but one is adapted to plotting literal strings, ground motion data, another plots symbols from integer valued data (intensity data), and the other plots symbols for numbers with decimal fractions (magnitude data).

Appendix A. Parameter Input File

```
    2    2
 44.0000   50.0000 -118.0000   29.5000   45.5000
    2    2
  4.0000   18.0000
    1    1    0    2
  0.1200    0.2000
```

```
Plot window input file description.
     The file name is arbitrary and the format is as follows:
     The first line is the projection type 1 Mercator 2 Alberts 3
        Lambert and the plot type 0 geography only
                                  1 geography and literals
                                  2 geography and integer symbols
                                  3 geography and real symbols
        with a 2i5 format.
     The second line is bottom,top latitudes,central
        longitude and the two standard parallels
        for conic projections or the left
        and right longitudes of the window
        for a mercator plot with a 5f10.4 format.
     The third line is the degree spacing between latitude
        and longitude lines with a 2i5 format.
     The fourth line is the number of degrees from central
        longitude to the right side and the height in inches.
     The fifth line contains line quality information
        first position 0 or 1 labels only or labels and lat,long lines
        second position 0 or 1 lat,long line solid or dash
        third position 0 or 1 geography lines solid or dash
        fourth position index of symbol to be plotted
        this line is a 4i5 format
     The sixth line contains the latitude and longitude
        label character height and symbol heights.
        2f10.4 format.
```

Appendix B. Point Value Input Data

The file name is arbitrary and the format of the data is input by the user at execution time.

				LAT	LONG			INT	MAG
1872	03	14	00:00:00.0	39.5	-117.0	G	54	3ALG	3.0
1872	03	26	00:00:00.0	39.5	-117.0	G	54	3ALG	3.0
1873	03	24	19:00:00.0	39.5	-117.0	G	54	2ALG	2.3
1881	12	08	01:50:00.0	39.5	-116.0	G	54	4ALG	3.7
1892	11	06	00:00:00.0	39.5	-117.0	G	54	3ALG	3.0
1892	11	25	00:00:00.0	39.5	-117.0	G	54	3ALG	3.0
19 1	07	26	22:20:00.0	40.8	-115.7	F	54	7ALG	5.0
1914	03	06	10:00:00.0	41.0	-116.0	G	54	3ALG	3.0

Appendix C. Geographical Input Data

File name NA.DAT format 2f11.5,i5

```
49.00000-122.750001    10
48.95000-122.733331    10
48.96667-122.800001    10
48.93333-122.816671    10
48.93333-122.750001    10
```

Appendix D. Zone Input Data File

Source zone data file
(embedded here in the input to the risk program)
File name REGION.INC format fixed

```
east coast zones acc
7 0
1
 84.0 39.0  76.0 32.0
       4.0     1000.      20.00        1.0         .50
  0  0  0  0
0
ecatten79          8.5         7.6         6.6         5.6         5.2         4.2
3.22        .74         .73         .67         .45         .195        .072
6.43        .64         .62         .53         .36         .135        .047
16.09       .49         .43         .34         .19         .055        .0228
32.18       .36         .285        .18         .097        .0275       .0113
64.3        .21         .148        .091        .049        .0138       .00565
96.5        .139        .097        .060        .0358       .0093       .00375
160.9       .083        .059        .036        .0196       .00535      .0018
321.8       .042        .0295       .0182       .0095       .00205      .00058
643.        .021        .0145       .0082       .00375      .00066      .0001
960.        .0132       .0084       .00455      .00195      .0001       .0001
1600.       .00645      .00365      .0019       .00085      .0001       .0001
3200.       .00204      .00117      .00061      .0001       .0001       .0001
 78.0      42.00       80.00       29.00        .250        .250
000 1.         1.0        -1.90               100   500 2501099
  8  1  5
 81.01 27.62 79.76 27.13
 83.00 29.00 79.91 27.96
 86.38 32.01 80.08 28.86
 85.90 32.58 79.90 29.78
 85.58 32.98 79.85 30.12
 84.89 33.47 79.82 30.36
 83.93 34.11 79.64 30.86
 83.28 34.70 79.46 31.17
  4  2  5
 82.32 35.45 78.50 31.81
```

Appendix E. Dummy Point Value Data

The point can be selected to be out of plot window
so it will not be plotted.
The file name is arbitrary and the format is
input by the user at execution time.

```
36.5-89.5 25
```

Appendix F. Dummy Zone Input Data

File name REGION.INC (format fixed).
Assembled to have the same input switches as the risk input file, i.e., the 99's in the first positions of the last 2 lines which indicate a switch from source zones to faults and finally an end-of-file.

```
new madrid zones 29 oct 80
7 0
0
      4.0      1000.      20.00        1.0         .50
  0  0  0  0
0
eavsh79       8.6        7.6        6.6        5.6        5.1        4.2
        2. 130.        89.        57.        27.5       13.        1.55
        5. 120.        76.        45.        20.5        9.1        .77
       10. 101.        58.        33.        14.2        6.2        .42
       20.  83.        42.        21.         8.6        3.6        .19
       50.  53.        22.         9.1        3.5       1.45        .065
      100.  30.5       11.         4.3        1.6         .65        .027
      200.  15.5        5.3        1.9         .7         .29        .0098
      500.   5.5        1.7         .6         .21        .085       .0026
     1000.   2.1         .64        .22        .074       .03        .00094
     2000.    .76        .23         .08       .027       .011       .00034
     5000.    .20        .061        .021      .0071      .0029      .000090
    10000.    .0668      .0203       .007      .00237     .00097     .00003
    20000.    .02227     .00677      .00233    .00079     .00032     .00001
125.0     50.0      70.0      25.0        1.0       1.0
990 1.      1.0      -1.90          100   500 2501087
 99         1.0      +2.90          100   500 250ft87
 99
```

Appendix G. Program Listing

Listing of the complete program EPMAP.FOR.

```
c     program   epmap   --epicenter map
c.......initialize plot and set origin
c
c  Program to produce maps of epicenters and the geography
c  for three types of projections:#Mercater, Albers, and Lambert.
c  the user has the option to input the map parameters from the
c  terminal or from a file.  The data to be plotted must be in a
c  file and contain the latitude, longitude of the
c  point to be plotted.  The user inputs the format of the data.
c  There are two possible types of plots: on the Calcomp or on
c  the Tektronics.  The program will ask the user for the type
c  of devise and then ask how you want to input the data:terminal or
c  data file.  If you choose the terminal, the program will save the
c  map parameters and will produce a file of these map parameters
c  for your use later on (you supply the file name).
c
c  This program is a many-times modified derivation
c  of a program originally programmed by by Carl Stover
c  and William Dillinger (NOAA Tech. Memo. ERL ESL-13 Aug. 1971)
c
c  It has been modified by Glen Reagor, Michael Mc Grath,
c  and Stanley L. Hanson.
c
c
c  beware all ye who enter here------
      common/lab/xyzl,xyzr,ela,elo,ia
      common/v2/map,nlabel,lin,ngeog,siz,ipchar,pcsiz
      common/po/igeog,ipro
      common/edat2/ iii,cx(3000),cy(3000),am(3000),de(3000)
      common/mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/ boty,sn,cro,croh,rzero
      common/mapco3/ nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
      common/store1/ax(512),ay(512)
      common/beta/ sbe(902)
      common/v1/xx(512),yy(512)
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      dimension iend(1)
        character name1*20

        character perid*10,projid*10
      character fmt*80
       character name2*20
      character title*80
c     this plotting program contains the following projections
c     Mercator - (ipro=1)
```

```
c     Lambert conformal conic, 2 std parallels - (ipro=3)
c     Albers equal area, 2 std parallels - (ipro=2)
c**********************************************************************
c     eflo (phi(2)) - left longitude or south std parallel(30)
c     ritlo - right longitude or north std parallel(45)
c     botla - (phi1) - bottom latitude
c     topla - top latitude
c     centlo - central longitude (conic)
c     sca - map scale factor
c     lati - latitude coordinate interval
c     longi - longitude coordinate interval
c     scl - desired decimal rounding of data
c     nbx - number in degrees from centlo at mid-latitude
c     longitude desired at right edge of map (conic only)
c     nby - length of map in y-direction(sca bases on nby) in inches
c     from botla.
c     ipro - map projection wanted
c     igeog - 0= geography only, 1= geography and epicenters,
c     2= epicenters only
c     nlabel=0 for labels only
c     nlabel=1 for lat and lon lines and labels
c     map for alcon projection only
c     map=0 for full map
c     lin determines the pen type for lat and long lines
c     ngeog determines the pen tye for geography
c     siz determines size of character'height in inches'(about .15)
c     of lat and long lines
c**********************************************************************
      real nbx,nby
       scl=.01
      sca=200000
       irr=0.
      inp=10
1        write(6,111)
111     format(' enter name of file containing map parameters '
     1 ,'(tty if from terminal)')
        read(5,112)name1
112     format(a20)
      if (name1.eq.'tty        ' )call ttyin
       if(name1.eq.'tty        ')go to 1127
       open(unit=10,file=name1,status='old')
      read(inp,87)ipro,igeog
87     format(5i5)
      read(inp,88)botla,topla,centlo,eflo,ritlo
88     format(5f10.4)
      read(inp,87)lati,longi
      read(inp,88)nbx,nby
       read(inp,87)nlabel,lin,ngeog,ipchar
       read(inp,88)siz,pcsiz
1127     write(6,113)
113     format(' enter name of data file - (s if same)')
```

```
1135   read(5,112)name2
       if(name2.ne.'s                         ')inp=11
      if(name2.ne.'s')open(unit=11,file=name2,status='old')
c
c  get format of lat and long data in file
c
      write(6,690)
690    format(' enter format of data including parens ')
       read(5,691)fmt
691    format(a80)
c
c  ask for title and title size
c
      write(6,692)
692    format(' enter title of map(80 char or less)')
       read(5,693)title
693    format(a80)
       write(6,694)
694    format(' enter size of title characters (.15)')
       read(5,695)titsiz
695    format(f)

c     ****************************************************************
c     north latitudes are positive
c     west longitudes are negative
c     ****************************************************************
115     go to (11,12,13),ipro
11        call mercat
       go to 100
12      call alcon
       go to 100

13      call lamcon
100   continue
      call geog(sca,ty,tx,ipro)
      if(igeog.le.0) go to 1004
      if(igeog.eq.2) go to 151
       if(igeog.eq.3) go to 152
c
c     routine to read data cards
c
150    call lesen(inp,ierr,ipchar,pcsiz,kk,fmt)
       go to 1004
151    call lesen2(inp,ierr,ipchar,pcsiz,kk,fmt)
       go to 1004
152    call lesen3(inp,ierr,ipchar,pcsiz,kk,fmt)
1004    continue
c
c  plot the title above the data area 80 char max
c
      xti = 1.0
```

```
        yti = ty+1.0
        call symbol(xti,yti,titsiz,title,0.0,80)
c
c draw a box around the map
c
                 call box(tx,ty,siz)
      call endplt
777     write(6,776)kk
776     format(' there were',i6' points plotted.')
         if(ierr.ne.0)write(6,778)ierr
778      format(' there were ',i6,' points that could not be  plotted.'
     1 /' please check error.dat for those locations.')
        if(name1.ne.'tty')close(10)
        if (inp.eq.11)close(11)
        stop
      end


      subroutine alcon
      dimension ep(5),inch(2)
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr,ela,elo,ia
      common/mapco1/ eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/boty,sn,cro,croh,rzero
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
      common/store1/ax(512),ay(512)
      common/beta/sbe(902)
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      real nbx,nby
       data inch /1,0/
c     ****************************************************************
c     This subroutine computes the constants necessary for the Albers
c     equal area map projection.
c     The mathematics for this program have been taken from the
c     USCGS special publication number 68
c     ecc=the eccentricity of the spheriod in use
c     sa=equatorial radius of the spheriod
c     sc=lowercase c of page 98 cgs sp-68
c     sbe(i)=array of values for the sin(beta) page 99
c     the following routine evaluates sc and sbe(i)
c     ****************************************************************
c     arithmetic stmt funtion for computing the radius of a parallel e

      roh(ela)=sqrt(croh-cro*sbeta(ela))
      sa=6378206.40
      ecc=.08227185422
```

```
      sc=10.0**6.980420742
      dtr=.0174532925
      ep2=ecc*ecc
      epp=ep2
      phi(1)=eflo
      phi(2)=ritlo
      a=2.0
      b=3.0
      d=1.0
      do 10 i=1,4
      ep(i)=a*epp/b
      a=a+1.0
      b=b+2.0
      epp=epp*ep2
      d=d+ep(i)
10    continue
      d11=d
      ep(5)=a*epp/b
      d=d+ep(5)
      sc=sqrt(sa*sa*(1.0-ep2)*d)
      i=0
      ela=0.
      top=90.
      itop=top
      if(top.gt.90.) top=90.
14    if(top-ela)30,15,15
15    s=sin(ela*dtr)
      i=i+1
      ss=s*s
      st=ss
      a=1.0
      do 20 j=1,5
      a=a+ep(j)*st
      st=st*ss
20    continue
      sbe(i)=s*a/d
      ela=ela+.1
      go to 14
30    phi1=phi(1)*dtr
      phi2=phi(2)*dtr
      au1=sqrt(1.-ep2*sin(phi1)**2)/sa
      au2=sqrt(1.-ep2*sin(phi2)**2)/sa
c
c
c     sn=lowercase n on page 100 USCGS SP-68, 5th edition
c
      sn=(cos(phi1)**2/(au1**2)-(cos(phi2)**2/(au2**2)))
     1/(2.*sc*sc*(sbeta(phi(2))-sbeta(phi(1))))
c
c     ro1=radius of parallel of true scale phi(1)
c
```

```
      rol=cos(phil)/(sn*aul)
      cro=2.*sc*sc/sn
      croh=rol*rol+cro*sbeta(phi(1))
c
c     rzero=radius of parrallel for botla
c
      rzero=roh(botla)

c     ****************************************************************
c     constants for the Albers projection have been computed. Constants
c     for plotting follow
c     necessary common phi(1),phi(2),centlo,rzero,cro,croh,sn,botla,
c     ecc,rx,tx,ty,nbx,nby
c     the table of sin(beta) will span 90 degrees beginning
c     at zero.  but will not exceed 90 degrees north or south lat.
c     ****************************************************************
c     next 8 statements computes sca to fit geography and
c     coordinates to specific map size (nbx by nby)
c
      ela=topla
      elo=centlo
      call axymap(x,y,ela,elo)
      sca=y/nby
      elo=centlo+nbx
      ela=botla+(topla-botla)/2.0
      ro=sqrt(croh-cro*sbeta(ela))
      t=sn*(elo-centlo)*dtr
      x=ro*sin(t)
      tx=2*x/sca
      ty=nby
      write(6,1000)tx,ty
1000  format(5x,7hwidth =,f7.2,5x,8hheight =,f7.2)
c
c     maximum width of map is 180 degrees at botla + 10.
c
      ispr=nbx+10
      spread=ispr
      min=centlo-spread
      max=centlo+spread
c
c........initialize the plot and set origin
c
      width = tx+1.0
      call plots(0,42.0)
      call plot(1.0,1.0,-3)
c
      ic=botla-10.
      ia=0
      do 211 i=ic,itop,lati
      ela=i
      if(ela.ge.botla) go to 4000
```

```
      istart=4*ispr
      elo=min
      do 4001 j=1,istart
      ia=ia+1
      call axymap(x,y,ela,elo)
      ax(ia)=x/sca
      ay(ia)=y/sca
      if(ax(ia).lt.-2.0)go to 187
      if(ax(ia).gt.(tx+2.0)) go to 187
      if(ay(ia).lt.-2.0) go to 187
      if(ay(ia).le.(ty+2.0)) go to 4001
187   ia=ia-1
4001  elo=elo+.5
      go to 4002
4000  do 210 j=min,max
      elo=j
      ia=ia+1
      call axymap(x,y,ela,elo)
      ax(ia)=x/sca
      ay(ia)=y/sca
      if(ax(ia).lt.-2.0) go to 186
      if(ax(ia).gt.(tx+2.0)) go to 186
      if(ay(ia).lt.-2.0) go to 186
      if(ay(ia).le.(ty+2.0)) go to 210
186   ia=ia-1
210   continue
4002  if(ia.le.1) go to 90
      xyzl=0.0
      xyzr=tx
      izy=0
      call label(izy)
90    ia=0
211   continue
      icc=botla-10.
      ia=0
305   do 311 j=min,max,longi
      elo=j
      do 310 i=icc,itop
      ela=i
      ia=ia+1
      call axymap(x,y,ela,elo)
      ax(ia)=x/sca
      ay(ia)=y/sca
      if(ay(ia).lt.-2.0) go to 9
      if(ay(ia).gt.(ty+2.0)) go to 9
      if(ax(ia).lt.-2.0) go to 9
      if(ax(ia).le.(tx+2.0)) go to 310
9     ia=ia-1
310   continue
      if(ia.le.1) go to 212
      izy=1
```

```
      call label(izy)
212   ia=0
311   continue
      return
      end



      subroutine axymap(x,y,ela,elo)
      common/mapco1/ eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/ boty,sn,cro,croh,rzero
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
c
c     *****************************************************************
c     This subroutine is for the Albers projection
c     *****************************************************************
c
      dtr=.017453295
      one=-1.
      if(ela.gt.90.) ela=90.
      botl=botla-10.
      if(ela.lt.botl) ela=botl
      edge=(abs(centlo)+180.)-360.
      if(centlo.lt.0.) go to 30
      if(elo.lt.centlo.and.elo.ge.edge) go to 47
      if(elo.ge.centlo) go to 47
      add=elo+180.
      elo=180.+add
      go to 47
30    edge=edge*one
      if(elo.ge.centlo.and.elo.le.edge) go to 47
      if(elo.lt.centlo) go to 47
      add=elo-180.
      elo=-180.+add
47    continue
      ro=sqrt(croh-cro*sbeta(ela))
      t=sn*(elo-centlo)*dtr
      x=ro*sin(t)+tx*sca*.5
      y=rzero-ro*cos(t)
      return
      end



      subroutine lamcon
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr,ela,elo,ia
      common/mapco1/ eflo,ritlo,botla,topla,centlo,phi(2)
```

```
      common/mapco2/ boty,sn,cro,croh,rzero
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
      common/store1/ax(512),ay(512)
      common/beta/sbe(902)
      common/plo/ xd(2),yd(2),xp(4),yp(4),ipltr(1)
       dimension inch(2)
      real nbx,nby
      data inch /1,0/
c
c     *****************************************************************
c     This subroutine computes the constants necessary for
c     the Lambert conic map projection.
c     this subroutine uses math taken from the US CGS SP251
c     For sl and ck see page 117 formulas 384 and 385for etau(i)
c     see page 86 and 68
c     For an1,an2 see n page 59
c     sbe(i) is a table of radius values see top of page 117 first
c     sentence for the formula for the radius.
c     *****************************************************************
c     arithmetic stmt function
c
      etau(ela)=((1.-ecc*sin(ela))/(1.+ecc*sin(ela)))**eco2
     1*sin((pio2+ela)/2.)/cos((pio2+ela)/2.)
      pi=3.1415927
      dir=.0174532925
      sa=6378206.400
      ecc=.08227185422
      ecc2=ecc*ecc
      pio2=pi/2.
      phi(1)=eflo
      phi(2)=ritlo
      phi1=phi(1)*dir
      phi2=phi(2)*dir
      an1=sa/sqrt(1.0-ecc2*sin(phi1)**2)
      an2=sa/sqrt(1.0-ecc2*sin(phi2)**2)
      eco2=ecc/2.
      sl=(alog(cos(phi1))-alog(cos(phi2))+alog(an1)-alog(an2))
     1/(alog(etau(phi2))-alog(etau(phi1)))
      sn=sl
      ck=an1*cos(phi1)/(sl*(1.0/etau(phi1))**sl)
      i=0
      ela=0.
      top=89.0
      itop=top
24    if(top-ela)50,25,25
25    i=i+1
      elar=ela*dir
      sbe(i)=ck*(1.0/etau(elar))**sl
      ela=ela+.1
      go to 24
```

```
c      ******************************************************************
c      computes sca to fit geography and coordinates to map nbx by nby
c      ******************************************************************

50     rzero=sbeta(botla)
       ela=topla
       elo=centlo
       call lxymap(x,y,ela,elo)
       sca=y/nby
       elo=centlo+nbx
       ela=botla+(topla-botla)/2.0
       r=sbeta(ela)
       t=sn*(elo-centlo)*dir
       x=r*sin(t)
       tx=2.*x/sca
       ty=nby
       write(6,1000)tx,ty

c      ******************************************************************
c      maximum width of map is 180 degrees at botla +10.
c      ******************************************************************

       ispr=nbx+10
       spread=ispr
       min=centlo-spread
       max=centlo+spread
c
c.........initialize the plottting
c
       width =tx+1.0
       call plots(0,42.0)
       call plot(1.0,1.0,-3)
c
233    ia=0
       ic=botla-0.
       do 111 i=ic,itop,lati
       ela=i
       if(ela.ge.botla) go to 4000
       istart=4*ispr
       elo=min
       do 4001 j=1,istart
       ia=ia+1
       call lxymap(x,y,ela,elo)
       ax(ia)=x/sca
       ay(ia)=y/sca
       if(ax(ia).lt.-2.0) go to 187
       if(ax(ia).gt.(tx+2.0)) go to 187
       if(ay(ia).lt.-2.0) go to 187
       if(ay(ia).le.(ty+2.0)) go to 4001
187    ia=ia-1
4001   elo=elo+.5
```

```
      go to 4002
4000  do 110 j=min,max
      elo=j
      ia=ia+1
      call lxymap(x,y,ela,elo)
      ax(ia)=x/sca
      ay(ia)=y/sca
      if(ax(ia).lt.-2.0) go to 6
      if(ax(ia).gt.(tx+2.0)) go to 6
      if(ay(ia).lt.-2.0) go to 6
      if(ay(ia).le.(ty+2.0)) go to 110
6     ia=ia-1
110   continue
4002  if(ia.le.1) go to 90
      xyzl=0.0
      xyzr=tx
      izy=0
      call label(izy)
90    ia=0
111   continue
      icc=botla-10.
      ia=0
205   do 211 j=min,max,longi
      elo=j
      do 210 i=icc,itop
      ela=i
      ia=ia+1
      call lxymap(x,y,ela,elo)
      ax(ia)=x/sca
      ay(ia)=y/sca
      if(ay(ia).lt.-2.0) go to 9
      if(ay(ia).gt.(ty+2.0)) go to 9
      if(ax(ia).lt.-2.0) go to 9
      if(ax(ia).le.(tx+2.0)) go to 210
9     ia=ia-1
210   continue
      if(ia.le.1) go to 212
      xyzl=0.0
      xyzr=tx
      izy=1
      call label(izy)
212   ia=0
211   continue
1000  format(5x,8hwidth  =,f7.2,5x,8hheight =,f7.2)
      return
      end


      subroutine lxymap (x,y,ela,elo)
```

```
c
      common/mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/boty,sn,cro,croh,rzero
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
c
c       this subroutine is for the lambert conformal conic projection
c
       dir=.017453295
      one=-1.
      if (ela.gt.89.) ela=89.
      botl=botla-0.
      if (ela.lt.botl) ela=botl
      edge=(abs(centlo)+180.)-360.
      if (centlo.lt.0.) go to 30
      if (elo.lt.centlo.and.elo.ge.edge) go to 47
      if (elo.ge.centlo) go to 47
      add=elo+180.
      elo=180.+add
      go to 47
   30 edge=edge*one
      if (elo.ge.centlo.and.elo.le.edge) go to 47
      if (elo.lt.centlo) go to 47
      add=elo-180.
      elo=-180.+add
   47 continue
      r = sbeta(ela)
      t = sn* (elo - centlo) * dir
      x = r * sin(t) + tx * sca * .5
      y = rzero -  r * cos(t)
      return
      end



      subroutine mercat

c    ***************************************************************
c    This subroutine computes the constants for the Mercator
c    map projection and plots the map.
c    ***************************************************************

      common /lab/ xyzl,xyzr
      common/mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/boty,sn,cro,croh,rzero
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
      common/store1/ax(512),ay(512)
      common/beta/sbe(902)
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
```

```
      dimension beta(902),inch(2)
      character rrlon * 5
       character bbla*5
      real nbx,nby
      data inch/1,0/
      data ecc/.08227185422/
      data rad/.017453295/
      data a/.8/
      data b/1.0/
      data sa/6378206.4/
      data one/1.0/
      jj=0
      kz=0
      eco2=ecc*.5
      pio4=45.0*rad
      blat=botla-float(lati)
      rlon= ritlo
       xlongi=float(longi)
      do 2000 i=1,900
      phii=i-1
      phii=phii*.1*rad
      es=ecc*sin(phii)
      beta(i)=alog(((1-es)/(1+es))**eco2*sin(pio4+phii*.5)/
     1cos(pio4+phii*.5))
      sbe(i)=beta(i)
2000  continue
      i=abs(botla)*10.+1.5
      elo=ritlo
      ela=topla
      boty=sa*beta(i)*(botla/abs(botla))
      call mxymap(x,y,ela,elo)
      sca=y/nby
      ty=y/sca
      tx=x/sca
      xyzl=0.
       xyzr=tx
      write(6,1000)tx,ty
1000  format(1x,'width=',f7.2,5x,'height=',f7.2)
c......initialize plot and set origin
c
      width =tx+1.0
      call plots(0,42.0)
      call plot(1.0,1.0,-3)
c
100   l=botla*10.+1.5*sign(one,botla)
      latii=lati*10
      ntop=topla*10.+1.5*sign(one,topla)
      l=l-latii
      ke=1
      lc=0
3     ii=1
```

```
2       l=l+latii
        if(l.eq.-1) l=l+2
        lo=-longi
        ii=ii-1
        if(l.gt.ntop) go to 60
        if(l)7,6,6
6       y=(sa*beta(l)-boty)/sca
        go to 9
7       ll=-l
        y=(-sa*beta(ll)-boty)/sca
9       lo=lo+longi
        ii=ii+1
        elo=lo
        x=elo*sa*rad/sca
        if(x.gt.tx+1.0) go to 55
        ax(ii)=x
        ay(ii)=y
        go to 9
55      ii=ii-1
        blat=blat+float(lati)
        encode(5,4000,bbla)blat
4000    format(f5.1)
        call symbol(ax(1),ay(1),0.1,bbla,0.0,5)
         do 4200 ij=1,ii
        call plot(ax(ij),ay(ij),2)
4200     continue
        call symbol(ax(ii),ay(ii),0.1,bbla,0.0,5)
        if(ay(1).le.0.001) go to 38
        go to 3
38       prlon=abs(rlon)
        encode(5,4001,rrlon)prlon
4001    format(f5.1)
        call symbol(ax(ii),0.0,0.1,rrlon,0.0,5)
         call plot(ax(ii),0.0,3)
        call plot(ax(ii),ty,2)
        call symbol(ax(ii),ty,0.1,rrlon,0.0,5)
        ii=ii-1
        if(ii.eq.0) go to 3
        rlon=rlon-xlongi
        prlon=abs(rlon)
        call plot(ax(ii),ty,3)
        encode(5,4001,rrlon)prlon
        call symbol(ax(ii),ty,0.1,rrlon,0.0,5)
         call plot(ax(ii),ty,3)
        call plot(ax(ii),0.0,2)
        call symbol(ax(ii),0.0,0.1,rrlon,0.0,5)
        ii=ii-1
        if(ii.eq.0) go to 3
        rlon=rlon-xlongi
        prlon=abs(rlon)
        call plot(ax(ii),0.0,3)
```

```
      go to 38
60    return
      end



      subroutine mxymap(x,y,ela,elo)

c    ***************************************************************
c  This subroutine is used to compute some of the constants
c  for the Mercator map projection.
c    ***************************************************************

      common/mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/boty,sn,cro,croh,rzero
      common/beta/sbe(902)
      data rad/.017453295/
      data sa/6378206.4/
1     if(elo.gt.eflo) go to 4
3     elo=(360.0-eflo+elo)*rad
      go to 5
4     elo=(elo-eflo)*rad
5     a=ela*10.
      if(a)7,6,6
6     i=a+1.0
      c=i-1
      y=sa*(sbe(i)+(c-a)*(sbe(i)-sbe(i+1)))-boty
      go to 8
7     i=-a+1.0
      c=i-1
      y=-sa*(sbe(i)+(c+a)*(sbe(i)-sbe(i+1)))-boty
8     x=elo*sa
c     all values computed in coord will be + with respect to eflo
      return
      end



      function sbeta(ela)

c
c   A subfunction used in computing constants for the different
c   map projections
c

      common/beta/sbe(902)
      common/mapco1/ eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco2/ boty,sn,cro,croh,rzero
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
```

```
      common/mapco4/lat1,longi
      i=ela*10.0+1.
      b=(i-1)*0.1
      sbeta=sbe(i)+(sbe(i+1)-sbe(i))*(ela-b)/0.1
      return
      end
c    ****** subroutine  lesen2   *******

c  Subroutine to read in the data to be plotted in the program
c  epmap. Presently the program reads in the latitude and longitude
c  of the epicenters to be plotted, and the intensity.
c  The subroutine then selects a symbol for each intensity
c  4 through 10 and plots that symbol.
c
      subroutine lesen2(inp,ierr,ipchar,pcsiz,kk,fmt)
       character ipn*4
       character astx*4
       character icount*5
      character fmt*40
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr
      common/po/igeog,ipro
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      dimension isym(9)
       real nbx,nby
       data isym/49,50,51,52,53,54,55,56,57/
       astx='   *'
      ierr=0
      txlat=0.
       txlon=0.
       charht=.1
3002  kk=0
1     read(inp,fmt,end=11) txlat,txlon,idpt
      ela=txlat
      elo=txlon
      elat=txlat
      elon=txlon
      go to (3,29,4),ipro
3     call mxymap(x,y,ela,elo)
      go to 5
29    call axymap(x,y,ela,elo)
      go to 5
4     call lxymap(x,y,ela,elo)
5     ab=(x/sca+scl/2.)/(scl*100.)
      k=ab*100.
      txlon=(k*scl)-xyzl
      if(txlon.gt.xyzr-xyzl) go to 10
      if(txlon.lt.0.0) go to 10
      ab=(y/sca+scl/2.)/(scl*100.)
      k=ab*100.
```

```
      txlat=k*scl
      if(txlat.lt.0.0)go to 10
      if(txlat.gt.ty) go to 10
       call ctype(3)
       ipchar=4
      if(idpt.eq.4) ipchar=6
      if(idpt.eq.5) ipchar=5
      if(idpt.eq.6) ipchar=1
      if(idpt.eq.7) ipchar=0
       if(idpt.eq.8) ipchar=2
       if(idpt.eq.9) ipchar=3
       if(idpt.eq.10) ipchar=9
       call ptplot(txlon,txlat,charht,ipchar,0,3)
      kk=kk+1
      go to 1
10        ierr=ierr+1
      if(ierr.eq.1) open( unit=14, file='error.dat',status='new')
        write(14,14)elat,elon
14      format(' xlat=',f8.3,'   xlon=',f8.3)
        go to 1
11     if(ierr.eq.0)return
       close (14)
      return
      end
c    ****** subroutine  lesen3   *******

c  Subroutine to read in the data to be plotted in the program
c  epmap. Presently the program reads in the latitude and longitude
c  of the epicenters to be plotted, and the magnitude.
c  The subroutine then selects a symbol for a particular
c  magnitude and plots that symbol.
c
      subroutine lesen3(inp,ierr,ipchar,pcsiz,kk,fmt)
       character ipn*4
       character astx*4
       character icount*5
      character fmt*40
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr
      common/po/igeog,ipro
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      dimension isym(9)
       real nbx,nby,idpt
       data isym/49,50,51,52,53,54,55,56,57/
       astx='   *'
      ierr=0
      txlat=0.
       txlon=0.
       charht=.1
3002  kk=0
```

```
1     read(inp,fmt,end=11) txlat,txlon,idpt
      ela=txlat
      elo=txlon
      elat=txlat
      elon=txlon
      go to (3,29,4),ipro
3     call mxymap(x,y,ela,elo)
      go to 5
29    call axymap(x,y,ela,elo)
      go to 5
4     call lxymap(x,y,ela,elo)
5     ab=(x/sca+scl/2.)/(scl*100.)
      k=ab*100.
      txlon=(k*scl)-xyzl
      if(txlon.gt.xyzr-xyzl) go to 10
      if(txlon.lt.0.0) go to 10
      ab=(y/sca+scl/2.)/(scl*100.)
      k=ab*100.
      txlat=k*scl
      if(txlat.lt.0.0)go to 10
      if(txlat.gt.ty) go to 10
       call ctype(3)
       ipchar=4
       if(idpt.lt.2.0) ipchar=4
      if(idpt.ge.2.0.and.idpt.lt.3.0) ipchar=6
      if(idpt.ge.3.0.and.idpt.lt.4.0) ipchar=5
      if(idpt.ge.4.0.and.idpt.lt.5.0) ipchar=1
      if(idpt.ge.5.0.and.idpt.lt.6.0) ipchar=0
       if(idpt.ge.6.0.and.idpt.lt.7.0) ipchar=2
       if(idpt.ge.7.0.and.idpt.lt.8.0) ipchar=3
       if(idpt.ge.8.0) ipchar=9
       call ptplot(txlon,txlat,charht,ipchar,0,3)
      kk=kk+1
      go to 1
10       ierr=ierr+1
      if(ierr.eq.1) open( unit=14, file='error.dat',status='new')
        write(14,14)elat,elon
14      format(' xlat=',f8.3,'   xlon=',f8.3)
        go to 1
11     if(ierr.eq.0)return
       close (14)
      return
      end
c    ****** subroutine  lesen   *******

c  Subroutine to read in the data to be plotted in the program
c  epmap. Presently the program reads in the latitude and longitude
c  and a literal string to be plotted.
c  The subroutine is designed to plot up to only four (4)
c  characters. If the value is negative the values will be
c  plotted with a different pen color.(by convention red pen
```

```
c  position 2) An asterisk is plotted to denote the value zero.
c  A capital X is plotted when the value exceeds 4 characters.
c  The values are plotted centered over the lat, long point.
c
      subroutine lesen(inp,ierr,ipchar,pcsiz,kk,fmt)
       character ipn*4
       character astx*4,astx2*4
       character icount*5
      character fmt*40
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr
      common/po/igeog,ipro
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
      dimension isym(9)
       real nbx,nby
       data isym/49,50,51,52,53,54,55,56,57/
       astx='   *'
       astx2='   X'
      ierr=0
      txlat=0.
       txlon=0.
       charht=.05
3002  kk=0
1     read(inp,fmt,end=11) txlat,txlon,idpt
       if(idpt.ge.10000) go to 71
      if(idpt.eq.0) go to 70
      if(idpt.lt.0) go to 72
      encode(4,60,ipn)idpt
60    format(i4)
      go to 80
70    encode(4,90,ipn) astx
90    format(a4)
       go to 80
71     continue
       encode(4,90,ipn)astx2
       go to 80
72     continue
       iidpt=jiabs(idpt)
       encode(4,60,ipn)iidpt
       idpt=iidpt
       call ptype(2)
80    continue
      ela=txlat
      elo=txlon
      elat=txlat
      elon=txlon
      go to (3,29,4),ipro
3     call mxymap(x,y,ela,elo)
      go to 5
29    call axymap(x,y,ela,elo)
```

```
      go to 5
4     call lxymap(x,y,ela,elo)
5     ab=(x/sca+scl/2.)/(scl*100.)
      k=ab*100.
      txlon=(k*scl)-xyzl
      if(txlon.gt.xyzr-xyzl) go to 10
      if(txlon.lt.0.0) go to 10
      ab=(y/sca+scl/2.)/(scl*100.)
      k=ab*100.
      txlat=k*scl
      if(txlat.lt.0.0)go to 10
      if(txlat.gt.ty) go to 10
      if(idpt.lt.10000.and.idpt.ge.1000) txlon=txlon+(-2.5*(.55*charht))
      if(idpt.lt.1000.and.idpt.ge.100) txlon=txlon+(-3.0*(.55*charht))
      if(idpt.lt.100.and.idpt.ge.10) txlon=txlon+(-3.5*(.55*charht))
      if(idpt.lt.10) txlon=txlon+(-4.0*(.55*charht))
      if(idpt.ge.10000) txlon=txlon+(-4.0*(.55*charht))
       txlat=txlat-(.5*charht)
       call symbol(txlon,txlat,.1,ipn,0,4)
       call ptype(1)
      kk=kk+1
      go to 1
10       ierr=ierr+1
      if(ierr.eq.1) open( unit=14, file='error.dat',status='new')
        write(14,14)elat,elon
14      format(' xlat=',f8.3,'   xlon=',f8.3)
       call ptype(1)
        go to 1
11     if(ierr.eq.0)return
       close (14)
       call ptype(1)
      return
      end
      subroutine geog(sca,ty,tx,ipro)
c
c This subroutine is used to plot source zones and faults from
c the input data file for the risk programs.
c This program requires as input the input data file to the risk program
c with the name region.inc.
c
c The data is read in in groups and plotted in groups.
c
c  The source zone,s are plotted with the default pen. (position one
c  black pen) The faults are plotted with pen position two a
c  red pen. A final subroutine call is made to coast which is the
c  subroutine that will plot the political and aquatic boundaries
c  of the U.S.

       common /mapcol/eflo,ritlo,botla,topla,centlo,phi(2)
       common /mapco3/nbx,nby,dum(6)
       common/mapco4/lati,longi
```

```
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr,ela,elo,ia
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      dimension eela(512),eelo(512),lsx(512)
      isw=0
      if(map.eq.0) goto 3001
      xd(1)=0.0
      xd(2)=tx/2.
      yd(1)=0.0
      yd(2)=ty
      xp(1)=tx/2.
      xp(2)=1.0
      yp(1)=ty
      yp(2)=1.0
      go to 3002
3001  xd(1)=0.0
      xd(2)=tx
      yd(1)=0.0
      yd(2)=ty
      xp(1)=tx
      xp(2)=1.0
      yp(1)=ty
      yp(2)=1.0
      flag=0.0
3002  lsn=0
      xlon=longi
      xlat=lati
      do 4 i=1,512
      eelo(i)=9999.99999
      eela(i)=99999.99999
      lsx(i)=999
4     continue
       open(12,file='region.inc',status='old')

        read(12,9)isw
9       format(2(/),50x,i1)
c
c The zone quadrilateral data is read in and
c sorted into a sequential data file (group) suitable for plotting
c

        read(12,5,end=102)num
5       format(18(/),i2)
        if(num.eq.99) go to 80
14      continue
        read(12,6,end=102)jsegs,ifr,itot
6       format(3i3)
12      continue
        ibox=1
        k=1
       read(12,7,end=102)eelo(k),eela(k),eelo(k+1),eela(k+1)
```

```
        lsx(k)=ibox
        lsx(k+1)=ibox

        ifrst=4
        do 10 l=1,(jsegs-1)
        k=k+2
        read(12,7,end=102)eelo(k+1),eela(k+1),eelo(k),eela(k)
 7      format(4f6.2)
        lsx(k)=ibox
        lsx(k+1)=ibox
        k=k+2
         ifrst=5
         if(l.eq.1) ifrst=4
        eelo(k)=eelo(k-ifrst)
        eela(k)=eela(k-ifrst)
        lsx(k)=ibox
        eelo(k+1)=eelo(k-2)
        eela(k+1)=eela(k-2)
        ibox=ibox+1
        lsx(k+1)=ibox
10      continue
        npts=jsegs*4
600   continue
      do 222 i=1,npts
      ela=eela(i)
      elo=-eelo(i)

c
c The input longitudes and latitudes are in degree minute form.
c The subroutine condec converts the input into degree decimal form.
c

       if(isw.eq.0) go to 50
       call condec(ela)
       call condec(elo)
50     continue
      if(lsx(i).eq.999) go to 202
      if(lsn.eq.0) go to 103
      if(lsn.ne.lsx(i)) go to 107
      go to (444,442,443) ipro
443   call lxymap(x,y,ela,elo)
      if(x/sca.lt.(-2.0*xlon)) go to 222
      if(x/sca.gt.(tx+(2.0*xlon))) go to 222
      if(y/sca.lt.(-2.0*xlat)) go to 222
      if(y/sca.gt.(ty+(2.0*xlat))) go to 222
      xx=x/sca
      yy=y/sca
      go to 1
442   call axymap(x,y,ela,elo)
      if(x/sca.lt.xyzl) go to 222
      if(x/sca.gt.xyzr) go to 222
```

```
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca-xyzl
      yy=y/sca
      go to 1
444   call mxymap(x,y,ela,elo)
      if(x/sca.lt.0.0) go to 222
      if(x/sca.gt.tx) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca
      yy=y/sca
c
c     plot continuation line
c
1     call plot(xx,yy,2)
      lsn=lsx(i)
      go to 222
103   go to (445,446,447) ipro
447   call lxymap(x,y,ela,elo)
      if(x/sca.lt.(-2.0*xlon)) go to 222
      if(x/sca.gt.(tx+(2.0*xlon))) go to 222
      if(y/sca.lt.(-2.0*xlat)) go to 222
      if(y/sca.gt.(ty+(2.0*xlat))) go to 222
      xx=x/sca
      yy=y/sca
      go to 88
446   call axymap(x,y,ela,elo)
      if(x/sca.lt.xyzl) go to 222
      if(x/sca.gt.xyzr) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca-xyzl
      yy=y/sca
      go to 88
445   call mxymap(x,y,ela,elo)
      if(x/sca.lt.0.0) go to 222
      if(x/sca.gt.tx) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca
      yy=y/sca
c
c     plot no continuation line for first entry
c
88    call plot(xx,yy,3)
      lsn=lsx(i)
      go to 222
107   go to (448,449,450) ipro
450   call lxymap(x,y,ela,elo)
      if(x/sca.lt.(-2.0*xlon)) go to 222
```

```
      if(x/sca.gt.(tx+(2.0*xlon))) go to 222
      if(y/sca.lt.(-2.0*xlat)) go to 222
      if(y/sca.gt.(ty+(2.0*xlat))) go to 222
      xx=x/sca
      yy=y/sca
      go to 89
449   call axymap(x,y,ela,elo)
      if(x/sca.lt.xyzl) go to 222
      if(x/sca.gt.xyzr) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca-xyzl
      yy=y/sca
      go to 89
448   call mxymap(x,y,ela,elo)
      if(x/sca.lt.0.0) go to 222
      if(x/sca.gt.tx) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca
      yy=y/sca
c
c     plot no continuation line due to region change(lsn.ne.lsx(i))
c
89    call plot(xx,yy,3)
      lsn=lsx(i)
222   continue
202    continue
      do 11 i=1,512
      eelo(i)=9999.99999
      eela(i)=99999.99999
      lsx(i)=999
11    continue

      if(flag.ne.1) go to 45
      if(ifr.ne.itot) go to 80
      go to 81

45    continue
      if(itot.ne.ifr) go to 14
      read(12,8,end=102)num
8     format(/,/,i2)
      if(num.eq.99) go to 80
      go to 14


81    read(12,71,end=102)num
71    format(/,/,i2)
      if(num.eq.99) go to 102

c
```

```
c     The fault segment data is read in and plotted one section at a time.
c

80    continue
       call ptype(2)
      flag=1
      read(12,6,end=102)jsegs,ifr,itot
      if(jsegs.eq.99) go to 102

       read(12,21,end=102)(eelo(i),eela(i),i=1,jsegs)
21     format(8f10.2)

500   continue
      lsx(1)=1

      do 501 kl=2,jsegs
      lsx(kl)=1
501   continue
      npts=jsegs
      lsn=0
      go to 600
102    close (12)
       call ptype(1)

       call coast(sca,ty,tx,ipro)

        return
      end
      subroutine condec(phi)
      i=phi
      fr=phi-i
      phi=i+fr/.60
      return
      end
      subroutine coast(sca,ty,tx,ipro)
       common /mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
       common /mapco3/nbx,nby,dum(6)
       common/mapco4/lati,longi
      common/v2/map,nlabel,lin,ngeog,siz
      common/lab/xyzl,xyzr,ela,elo,ia
      common/plo/xd(2),yd(2),xp(4),yp(4),ipltr(1)
      dimension eela(512),eelo(512),lsx(512)
      bot95=.95*botla
      cent1=centlo-(1.05*nbx)
      cent2=centlo+(1.05*nbx)
      if(map.eq.0) goto 3001
      xd(1)=0.0
      xd(2)=tx/2.
      yd(1)=0.0
      yd(2)=ty
      xp(1)=tx/2.
```

```
      xp(2)=1.0
      yp(1)=ty
      yp(2)=1.0
      go to 3002
3001  xd(1)=0.0
      xd(2)=tx
      yd(1)=0.0
      yd(2)=ty
      xp(1)=tx
      xp(2)=1.0
      yp(1)=ty
      yp(2)=1.0
3002  lsn=0
      xlon=longi
      xlat=lati
       open(12,file='na.dat',status='old')
42    read(12,169,end=102) (eela(i),eelo(i),lsx(i),i=1,100)
169    format(f10.5,f11.6,i5)
      do 222 i=1,100
      ela=eela(i)
      elo=eelo(i)
      if(lsx(i).eq.999) go to 102
      if((ela.lt.bot95).or.(ela.gt.topla)) go to 222
      if((elo.lt.cent1).or.(elo.gt.cent2)) go to 222
      if(lsn.eq.0) go to 103
      if(lsn.ne.lsx(i)) go to 107
      go to (444,442,443) ipro
443   call lxymap(x,y,ela,elo)
      if(x/sca.lt.(-2.0*xlon)) go to 222
      if(x/sca.gt.(tx+(2.0*xlon))) go to 222
      if(y/sca.lt.(-2.0*xlat)) go to 222
      if(y/sca.gt.(ty+(2.0*xlat))) go to 222
      xx=x/sca
      yy=y/sca
      go to 1
442   call axymap(x,y,ela,elo)
      if(x/sca.lt.xyzl) go to 222
      if(x/sca.gt.xyzr) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca-xyzl
      yy=y/sca
      go to 1
444   call mxymap(x,y,ela,elo)
      if(x/sca.lt.0.0) go to 222
      if(x/sca.gt.tx) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca
      yy=y/sca
c
```

```
c     plot continuation line
c
1     call plot(xx,yy,2)
      lsn=lsx(i)
      go to 222
103   go to (445,446,447) ipro
447   call lxymap(x,y,ela,elo)
      if(x/sca.lt.(-2.0*xlon)) go to 222
      if(x/sca.gt.(tx+(2.0*xlon))) go to 222
      if(y/sca.lt.(-2.0*xlat)) go to 222
      if(y/sca.gt.(ty+(2.0*xlat))) go to 222
      xx=x/sca
      yy=y/sca
      go to 88
446   call axymap(x,y,ela,elo)
      if(x/sca.lt.xyzl) go to 222
      if(x/sca.gt.xyzr) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca-xyzl
      yy=y/sca
      go to 88
445   call mxymap(x,y,ela,elo)
      if(x/sca.lt.0.0) go to 222
      if(x/sca.gt.tx) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca
      yy=y/sca
c
c     plot no continuation line for first entry
c
88    call plot(xx,yy,3)
      lsn=lsx(i)
      go to 222
107   go to (448,449,450) ipro
450   call lxymap(x,y,ela,elo)
      if(x/sca.lt.(-2.0*xlon)) go to 222
      if(x/sca.gt.(tx+(2.0*xlon))) go to 222
      if(y/sca.lt.(-2.0*xlat)) go to 222
      if(y/sca.gt.(ty+(2.0*xlat))) go to 222
      xx=x/sca
      yy=y/sca
      go to 89
449   call axymap(x,y,ela,elo)
      if(x/sca.lt.xyzl) go to 222
      if(x/sca.gt.xyzr) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca-xyzl
      yy=y/sca
```

```
      go to 89
448   call mxymap(x,y,ela,elo)
      if(x/sca.lt.0.0) go to 222
      if(x/sca.gt.tx) go to 222
      if(y/sca.lt.0.0) go to 222
      if(y/sca.gt.ty) go to 222
      xx=x/sca
      yy=y/sca
c
c     plot no continuation line due to region change(lsn.ne.lsx(i))
c
89    call plot(xx,yy,3)
      lsn=lsx(i)
222   continue
      go to 42
102    close(12)
        return
      end

c  Subroutine to read in the map parameters for the program
c  epmap from the terminal and to create a file of these
c  parameters on request.  The file is in the proper
c  format to be read by the program epmap.

      subroutine ttyin
      common/v2/map,nlabel,lin,ngeog,siz,ipchar,pcsiz
      common/po/igeog,ipro
      common/mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
      common/mapco3/ nbx,nby,tx,ty,sca,scl,xbd,ybd
      common/mapco4/lati,longi
      character ans1*3
       character mappar*12
      real nbx,nby
       write(6,400)
400     format(' enter map projection 1=mercator 2=albers 3=lambert')
       read(5,401)ipro
401    format(i)
c
c  Select the proper subroutine to read the input file for plotting.
c  That is to plot intensity,magnitude or literal string data.
c
       write(6,402)
402    format(' geography only=0, geog+literals=1, geog+int symbols=2',
     * /,1x,'geog + real symbols=3')
      read(5,401)igeog
        write(6,403)
403     format(' enter bottom latitude in degrees (r)')
       read(5,*)botla
        write(6,404)
404    format(' enter top latitude in degrees (r)')
       read(5,*)topla
```

```
      write(6,405)
405    format(' enter central longitude (conic) in degrees (r)')
       read(5,*)centlo
      write(6,415)
      write(6,4155)
415     format(' enter left long (mercat) or south std parallel')
4155   format(' (lambert,albers)(usually 30 for U.S.) in degrees (r)')
       read(5,*)eflo
       write(6,416)
416    format(' right long or north std parallel (degrees) (r)')
       read(5,*)ritlo
       write(6,406)
      write(6,4065)
406    format(' interval between latitude grid lines in degrees - ')
4065   format('(not less than 1 degree) (i)')
       read(5,*)lati
       write(6,407)
407    format(' interval between longitude grid lines in degrees - '
     1,'(not less than 1 degree) (i)')
      read(5,*)longi
       write(6,408)
408     format(' degrees from central longitude to right edge of'
     1 ,' map at mid latitude (degrees) (r)')
       read(5,*)nbx
       write(6,409)
409    format(' height of map (y-direction) in inches (r)')
       read(5,*)nby
       write(6,410)
410    format(' enter label code. 0=labels only  1=labels +'
     1 ,' lat and long lines')
       read(5,*)nlabel
       write(6,412)
412    format(' type of lat and long lines: 0=solid 1=dash')
       read(5,*)lin
       write(6,413)
413    format(' type of geography lines: 0=solid 1=dash ')
       read(5,*)ngeog
      write(6,420)
420    format('index of symbol to be used for epicenter (2)?')
       read(5,*)ipchar
       write(6,414)
414    format(' height of label characters in inches (about .12) ')
       read(5,*)siz
      write(6,618)

618    format(' height of symbol in inches (.12.) ')

       read(5,*) pcsiz
      write(6,610)
610    format(' do you want to sav map parameters in a file?')
       read(5,611) ans1
```

```
611     format(a3)
        if(ans1.eq.'no')return

c
c  write out data file of map parameters
c
        write(6,612)
612     format(' name of file to contain parameters?')
        read(5,613) mappar
613     format(a12)
       open(unit=15,file=mappar,status='unknown')

        write(15,614) ipro,igeog
614     format(5i5)
        write(15,615) botla,topla,centlo,eflo,ritlo
615     format(5f10.4)
        write(15,614) lati,longi
        write(15,615) nbx,nby
        write(15,614) nlabel,lin,ngeog,ipchar
        write(15,615) siz, pcsiz
       close(unit=15,status='keep')
       return
        end
       subroutine label(izy)
       common/v2/map,nlabel,lin,ngeog,siz
       common/mapco1/eflo,ritlo,botla,topla,centlo,phi(2)
       common/mapco3/nbx,nby,tx,ty,sca,scl,xbd,ybd
       common/store1/ax(512),ay(512)
       common/lab/xyzl,xyzr,ela,elo,ia
       character eela*2
        character eelo*3
       if(izy.eq.1) go to 667
       do 3010 jk=1,ia
       if(ax(jk).ge.xyzl.and.ay(jk).ge.0.0) go to 3011
3010   continue
       return
3011   if(ax(1).gt.xyzl) go to 91
       if(ay(1).lt.0.0) go to 91
       do 642 jj=1,ia
       if(ax(jj).ge.xyzl.and.ay(jj).ge.0.0) go to 643
       go to 642
643    ii=jj
       il=jj
       go to 992
642    continue
992    if(ax(ii).eq.xyzl) go to 644
       slope=(ay(ii)-ay(ii-1))/(ax(ii)-ax(ii-1))
       bb=ay(ii)-(slope*ax(ii))
       bl=bb
       yyy=(slope*xyzl)+bb
       if(yyy.lt.0.0) go to 91
```

```
      if(yyy.gt.ty) go to 91
      iela=ela
      encode(2,100,eela)iela
100   format(i2)
      xx=-2.2*siz
      yyy=yyy-.5*siz
        call plot(xx ,yyy,3)
      call symbol(xx,yyy,siz,eela,0.0,2)
      go to 91
644   iela=ela
      encode(2,100,eela)iela
      xx=-2.2*siz
      yy=ay(ii)-.5*siz
        call plot(xx ,ay(ii),3)
      call symbol(xx ,ay(ii),siz,eela,0.0,2)
91    do 842 jj=ia,1,-1
      if(map.ne.1) go to 996
      if(ax(jj).le.xyzr) go to 843
       go to 842
996   if(ax(jj).le.xyzr.and.ay(jj).ge.0.0) go to 843
       go to 842
843   ii=jj
      ir=jj
      go to 993
842   continue
      return
993   if(ax(ii).eq.xyzr) go to 844
      slope=(ay(ii)-ay(ii+1))/(ax(ii)-ax(ii+1))
      bb=ay(ii)-(slope*ax(ii))
      yyy=(slope*xyzr)+bb
      br=yyy
      if(yyy.lt.0.0) return
      if(yyy.gt.ty) return
      iela=ela
      encode(2,100,eela)iela
      abc=xyzr+.3*siz
      yyy=yyy-.5*siz
        call plot(abc,yyy,3)
      call symbol(abc,yyy,siz,eela,0.0,2)
      go to 90
844   iela=ela
      encode(2,100,eela)iela
      abc=xyzr+.3*siz
      yy=ay(ii)-.5*siz
        call plot(abc,yy,3)
      call symbol(abc,yy,siz,eela,0.0,2)
90    do 3002 kk=1,ia
      ax(kk)=ax(kk)-xyzl
3002  continue
      if(nlabel.eq.0) return
c
```

```
      call plot(tx,br,3)
      ip=2
      do 110 im=ir,il,-1
         call plot(ax(im),ay(im),ip)
110   continue
c
      call plot(0.0,bl,2)
      return
c*****
c The code below this point is for plotting the longitude
c lines. it starts at the top of the box and draws the long
c value first. then draws the line.
c the code at label 648 to 212 is for the case when the elo
c variable is equal to the central longitude(centlo).
c*******
667   if(elo.eq.centlo) go to 648
      do 845 jj=ia,1,-1
      if(ay(jj).gt.ty) go to 845
846   ii=jj
      it=jj
      ib=ia
      go to 994
845   continue
c
c if at this point then none of the points on the line fall
c below the top of the box ie ty=y.
c
      return
994   if(ay(ii).eq.ty) go to 847
      slope=(ay(ii)-ay(ii+1))/(ax(ii)-ax(ii+1))
      bb=ay(ii)-(slope*ax(ii))
      xxx=(ty-bb)/slope
      xt=xxx
      if(xxx.lt.xyzl) return
      if(xxx.gt.xyzr) return
      ielo=abs(elo)
c
c***convert the longitude >180 to <180 for labels
c
      if(ielo.gt.180)ielo=360-ielo
      encode(3,101,eelo)ielo
101   format(i3)
      xxx=xxx-1.5*siz
        yy=ty+.3*siz
        call plot(xxx,yy,3)
      call symbol(xxx,yy,siz,eelo,0.0,3)
      go to 944
847   ielo=abs(elo)
      encode(3,101,eelo)ielo
        xx=ax(ii)-1.5*siz
        yy=ty+.3*siz
```

```
          call plot(xx,yy,3)
        call symbol(xx,yy,siz,eelo,0.0,3)
944     do 645 jj=1,ia
        if(ay(jj).lt.0.0) go to 645
646     ii=jj
        ib=jj
        go to 995
645     continue
        stop 7777
995     if(ay(ii).eq.0.0) go to 647
        xxx=-bb/slope
        xb=xxx
        if(xxx.lt.xyzl) go to 211
        if(xxx.gt.xyzr) go to 211
        xxx=xxx-1.5*siz
          yy=-1.3*siz
          call plot(xxx,yy,3)
647     call symbol(xxx,yy,siz,eelo,0.0,3)
        ycpt=0.0
        go to 212
648     ielo=abs(elo)
        if(ielo.gt.180)ielo=360-ielo
        encode(3,101,eelo)ielo
        xxx=tx/2.-1.5*siz
        xt=xxx
        xb=xxx
          yy=0.0-1.3*siz
          call plot(xxx,yy,3)
765     call symbol(xxx,yy,siz,eelo,0.0,3)
c
        x2=tx/2.0
        if(nlabel.eq.0)go to 769
        call plot(x2,0.0,3)
        call plot(x2,ty,2)
769        continue
          yy=ty+.2*siz
          call plot(xxx,yy,3)
        call symbol(xxx,yy,siz,eelo,0.0,3)
        return
c
c* this code is to set y intercept for case when x<0 >tx
c
211     continue
        ycpt=-slope*xxx
        if(xxx.lt.0.0)xb=0.0
        if(xxx.gt.tx)xb=tx
c
c**redo the y intercept for case x=tx
c
        if(xb.eq.tx)ycpt=slope*tx+ycpt
212     continue
```

```
      if(nlabel.eq.0) return
c
        call plot(xb,ycpt,3)
       ip=2
      do 111 im=ib,it
        if(ax(im).lt.0.0 .or. ax(im).gt.tx)go to 111
        if(ay(im).gt.ty)go to 119
        call plot(ax(im),ay(im),ip)
111   continue
119         call plot(xt,ty,2)
c
          return
      end
```

Digitizing

Description

Once the seismic source zones have been defined and the quadrilaterals constructed, their corner points are digitized using the digitizer on the VAX/VMS and then converted to latitudes and longitudes using the program MAPTRAN.FOR on the VAX/VMS. It is at this point the program CONVERTDG.FOR is used to convert the digitized quadrilateral corner points and fault segments, if any, into the risk program input format. (fig. F, see connector "start".)

Input File Description

The input file to CONVERTDG.FOR is created by using the local system program [USCAT]DIGTIZ to digitize the points and [USCAT]MAPTRAN to translate the points into the proper latitude and longitude values. The file that is created by DIGTIZ must have in it the necessary zone header information needed later for the risk program input format, the switch digits either 00 or 99, the four character zone name on one line, then the number of data pairs and set numbers (see Figs. C and D) on the next line. (Example):

```
 $    2 = # of text lines
00i001
  3  1  1
 %    6  = # of points        0.00000 = area        0.00000 = perimeter
    29.269  -104.346   3
    28.620  -102.862   2
    30.514  -104.328   2
    29.187  -102.512   2
    31.535  -104.762   2
    29.929  -102.103   2
 $    2 = # of text lines
00i002
  3  1  1
 %    6  = # of points        0.00000 = area        0.00000 = perimeter
    30.580  -107.214   3
```

On the second line the switch value, either 00 or 99, followed by four characters to identify the source zone, on the next line the number of data pairs or ladder rungs and set sequence numbers, in this case three pairs of points in the first of one set, (format 3i3). (See Bender and Perkins, 1982 for further explanation of these numbers.) The lines begining with a $ sign or a % sign are provided by the program [USCAT]DIGTIZ, all other lines are user input or digitized quadrilateral corner points and fault segments, if any.

(See appendix A at the end of this section for a complete input file.)

By placing 99 in the first two positions of the zone title line of the first fault to be processed, (00 in the remaining fault title lines) the program will switch to processing only faults. The next occurrence of 99 in the first position will cause the program to terminate.

Output File Description

The output file will be formatted the same as the risk program input file. However there will be blank lines where the header, parameters, and attenuation function table will later be placed, and blank lines at the end of each zone and fault where the rates and magnitude levels belong. (See appendix B at the end of this section for a complete output file.)

Program Operation

The program is an interactive program asking for an output file name and the digitized-input file name.

It reads the input file and creates the master output file containing all the seismic source zones and faults structured in the SEISRISK II input format. The zone name is read from the second line in the input, (see appendix A at the end of this section). The leading two zeros indicate the processing of zones, the last four characters are the zone name. By placing 99 in the first two positions of the first fault and 00 in the first position of the succeeding faults the program will switch modes to process faults instead of the zone quadrilaterals.

The program reads the line containing the number of points and set numbers and writes them to the output file. The program reads a pair of latitude, longitude points from the digitized file and writes out the longitude, latitude pairs to the output files for source zones. For any digitized faults the program reads in a string of latitude, longitude points that form the fault segment and outputs that string in a longitude, latitude format.

When the end-of-file is reached the program inserts a line with 99 in the first two positions to indicate the end-of-data for the risk program. (See appendix C at the end of this section for a program listing.)

Appendix A. Sample Input File

The area and perimeter values are outputs from the digitizing program and can be ignored.
Zone i003 is a two set zone.
F001 is a set of ten parallel faults, while f002 is a single fault.

```
 $    2 = # of text lines
00i001
  3  1  1
 %    6  = # of points          0.00000 = area          0.00000 = perimeter
    29.269  -104.346   3
    28.620  -102.862   2
    30.514  -104.328   2
    29.187  -102.512   2
    31.535  -104.762   2
    29.929  -102.103   2
 $    2 = # of text lines
00i002
  3  1  1
 %    6  = # of points          0.00000 = area          0.00000 = perimeter
    30.580  -107.214   3
    30.693  -106.360   2
    31.858  -107.356   2
    31.825  -106.417   2
    32.075  -107.348   2
    32.705  -106.481   2
 $    2 = # of text lines
00i003
  6  1  2
 %   12  = # of points          0.00000 = area          0.00000 = perimeter
    30.570  -107.336   3
    30.580  -107.214   2
    31.804  -107.728   2
    31.858  -107.359   2
    32.523  -107.619   2
    32.078  -107.345   2
    33.827  -107.343   2
    32.316  -107.012   2
    33.743  -107.018   2
    32.469  -106.814   2
    33.852  -106.718   2
    32.705  -106.474   2
 $    1 = # of text lines
  6  2  2
 %   12  = # of points          0.00000 = area          0.00000 = perimeter
    29.478  -104.835   3
```

```
   29.266  -104.343   2
   30.693  -106.363   2
   30.512  -104.328   2
   32.705  -106.474   2
   32.733  -105.275   2
   34.212  -106.793   2
   34.093  -105.530   2
   34.529  -106.592   2
   34.803  -105.551   2
   35.425  -106.389   2
   35.688  -105.441   2
 $    2 = # of text lines
99f001
  2  1 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   29.489  -104.341   3
   28.789  -102.762   2
 $    1 = # of text lines
  2  2 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   29.702  -104.338   3
   28.957  -102.665   2
 $    1 = # of text lines
  2  3 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   29.905  -104.333   3
   29.125  -102.552   2
 $    1 = # of text lines
  2  4 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   30.111  -104.333   3
   29.287  -102.461   2
 $    1 = # of text lines
  2  5 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   30.326  -104.334   3
   29.460  -102.359   2
 $    1 = # of text lines
  2  6 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   30.520  -104.322   3
   29.628  -102.264   2
 $    1 = # of text lines
  2  7 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   30.766  -104.436   3
   29.787  -102.175   2
 $    1 = # of text lines
  2  8 10
 %    2  = # of points        0.00000 = area        0.00000 = perimeter
   31.017  -104.534   3
```

```
   30.079  -102.334   2
$    1 = # of text lines
 2   9 10
%    2  = # of points         0.00000 = area          0.00000 = perimeter
   31.253  -104.639   3
   30.765  -103.458   2
$    1 = # of text lines
 2 10 10
%    2  = # of points         0.00000 = area          0.00000 = perimeter
   31.489  -104.742   3
   31.416  -104.560   2
$    2 = # of text lines
00f002
  3  1  1
%    3  = # of points         0.00000 = area          0.00000 = perimeter
   30.575  -107.275   3
   31.031  -107.543   2
   32.300  -107.482   2
```

Appendix B. Sample Output File

```
            .
            .
            .
            .
            .
            .
            .
            .
            .
      21 blank lines for
       attenuation function table
       and other risk program
       input parameters
       to be inserted later.
            .
            .
            .
            .
            .
            .
            .
 0                                                  i001
  3  1  1
104.35 29.27102.86 28.62
104.33 30.51102.51 29.19
104.76 31.53102.10 29.93
     blank line for rates
     blank line for magnitudes
 0                                                  i002
  3  1  1
107.21 30.58106.36 30.69
107.36 31.86106.42 31.83
107.35 32.08106.48 32.71
     blank line
     blank line
 0                                                  i003
  6  1  2
107.34 30.57107.21 30.58
107.73 31.80107.36 31.86
107.62 32.52107.35 32.08
107.34 33.83107.01 32.32
107.02 33.74106.81 32.47
106.72 33.85106.47 32.71
  6  2  2
104.83 29.48104.34 29.27
106.36 30.69104.33 30.51
106.47 32.71105.28 32.73
```

```
106.79 34.21105.53 34.09
106.59 34.53105.55 34.80
106.39 35.42105.44 35.69
     blank line
     blank line
99                                              f001
  2  1 10
    104.34     29.49     102.76     28.79
  2  2 10
    104.34     29.70     102.67     28.96
  2  3 10
    104.33     29.91     102.55     29.13
  2  4 10
    104.33     30.11     102.46     29.29
  2  5 10
    104.33     30.33     102.36     29.46
  2  6 10
    104.32     30.52     102.26     29.63
  2  7 10
    104.44     30.77     102.18     29.79
  2  8 10
    104.53     31.02     102.33     30.08
  2  9 10
    104.64     31.25     103.46     30.76
  2 10 10
    104.74     31.49     104.56     31.42
      blank line
      blank line
00                                              f002
  3  1  1
    107.28     30.58     107.54     31.03     107.48     32.30
      blank line
      blank line
99
```

Appendix C. Program Listing

```
c Program to convert digitized quads and faults
c from digitizer output format into
c the risk input format. The program will read the
c output file from the digitizer and produce a file
c that contains the zone name the proper numbers
c for the variables jseg,ifr,itot and the blank lines
c necessary for the attenuation, rates and magnitude levels.
c Upon encountering a switch the program processes
c coordinates as fault segment data for sets of
c fault data.
c Written 12 April 1982 by Stanley L. Hanson
        character inam*13,title*4,znam*8
        character onam*13,f*3,fnam*4,junk*1
        real lat(100),long(100)
        iend=99
        write(6,5)
5       format(1x,'Enter the name of the output file.')
        read(5,17)onam
17      format(a13)
        open(14,file=onam,status='new',carriagecontrol='list')
        write(14,7)
7       format(21(/))
        write(6,10)
10      format(1x,'Enter the input file name')
        read(5,17)inam
        open(12,file=inam,status='old')
20      read(12,15,end=99)inum,title
15      format(/,i2,a4)
c
c Firat 99 terminates processing source zone
c coordinates and begins processing fault segment
c coordinates.
c
        if(inum.eq.99) go to 100
        write(14,16)inum,title
16      format(i2,47x,a4)
30      read(12,35) jseg,ifr,itot
35      format(3i3)
         read(12,37)
        do 40 i=1,jseg*2
        read(12,36)lat(i),long(i)
36      format(2f10.3)
        long(i)=-long(i)
40      continue
        write(14,35)jseg,ifr,itot
        jseg=jseg*2
        do 42 i=1,jseg,2
```

```
      write(14,45)long(i),lat(i),long(i+1),lat(i+1)
45    format(4f6.2)
42    continue
      if(ifr.ne.itot) go to 31
      write(14,43)
43    format(/)
      go to 20
31    continue
      read(12,37)
37    format(1x,a1)
      go to 30
105   read(12,15,end=99)inum,title
100   continue
      write(14,16)inum,title
106   read(12,35)jseg,ifr,itot
      read(12,37)
      do 111 j=1,jseg
     read(12,36)lat(j),long(j)
      long(j)=-long(j)
111   continue
      write(14,35)jseg,ifr,itot
      write(14,112)(long(k),lat(k),k=1,jseg)
112   format(8f10.2)
      if(ifr.ne.itot) go to 107
      write(14,43)
      go to 105
107   continue
      read(12,37)
      go to 106
99    close(12)
      write(14,125)iend
125   format(i2)
      close(14)
      stop
      end
```

Earthquake Collection

Description

After verification of the source zone and fault coordinates, the next phase in the system is earthquake data collection. (fig. F, see part B, connector J.)

The program REGIONI.FOR is designed to extract earthquakes (see Appendix C at the end of this section for the program listing) from one or more catalogs and create subcatalogs for each seismic source zone that is input. This task is accomplished by reading in each source zone from the risk program input file and checking the earthquake catalog for those events that fall within each zone. This is the first step towards the statistical analysis of the earthquake data.

Input File Description

There are two input files, the file containing the source zone quadrilateral coordinates, and the file containing the earthquake catalog. (See appendix A and appendix B at the end of this section for examples.)

The source zone file is the SEISRISK II (Bender and Perkins, 1982) program input file created by the program CONVERTDG.FOR. The catalog file is read in only once, although each run of REGIONI.FOR accesses the catalog as many times as there are source zones. Additional catalogs may be read before the program is terminated.

Output File Description

An output file is created for each of the source zones. A unique name is assigned to each of these output files by encoding the source zone name from the input file with a 5-character name supplied by the user during an interactive phase of the program. These files are appended to when multiple earthquake catalog input files are used. The output files are subsets of the main catalog with the same format as the input catalog, containing only those events whose epicenter lie within a particular zone.

Program Operation

The program is an interactive program, asking for the source zone description data file name as input. An earthquake epicenter catalog file name is then asked for. This file is read in and stored internally for the duration of the program run. A maximum of 6500 earthquakes is all the program can handle in any one run. Larger epicenter catalogs will have to be partitioned into subcatalogs.

Individual subcatalogs can be input in succession within the same program run. Subcatalog earthquakes found in any zone are added to those previously found in preceeding subcatalogs. The word quit entered in place of a catalog file name will cause the program to terminate.

The program asks for a 5-character name with which it will create an output file name for each source zone. The output file name combines the 5-character name with the 4 character source zone name read from the source zone description input file, creating a unique name.

Appendix A. Sample Input File

A typical source zone quadrilateral file.
In partial risk program input format.

```
              .
              .
              .
              .
              .
              .
              .
              .
              .
       21 blank lines
        attenuation function table
        is inserted here later.
              .
              .
              .
              .
              .
              .
              .
              .
              .
 0                                         i001
   3  1  1
104.35 29.27102.86 28.62
104.33 30.51102.51 29.19
104.76 31.53102.10 29.93
      blank line
      blank line
 0                                         i002
   3  1  1
107.21 30.58106.36 30.69
107.36 31.86106.42 31.83
107.35 32.08106.48 32.71
      blank line
      blank line
 0                                         i003
   6  1  2
107.34 30.57107.21 30.58
107.73 31.80107.36 31.86
107.62 32.52107.35 32.08
107.34 33.83107.01 32.32
107.02 33.74106.81 32.47
106.72 33.85106.47 32.71
   6  2  2
```

```
104.83 29.48104.34 29.27
106.36 30.69104.33 30.51
106.47 32.71105.28 32.73
106.79 34.21105.53 34.09
106.59 34.53105.55 34.80
106.39 35.42105.44 35.69
     blank line
     blank line
99                                              f001
  2  1 10
    104.34     29.49     102.76     28.79
  2  2 10
    104.34     29.70     102.67     28.96
  2  3 10
    104.33     29.91     102.55     29.13
  2  4 10
    104.33     30.11     102.46     29.29
  2  5 10
    104.33     30.33     102.36     29.46
  2  6 10
    104.32     30.52     102.26     29.63
  2  7 10
    104.44     30.77     102.18     29.79
  2  8 10
    104.53     31.02     102.33     30.08
  2  9 10
    104.64     31.25     103.46     30.76
  2 10 10
    104.74     31.49     104.56     31.42
      blank line
      blank line
99
```

Appendix B. Sample Input Catalog

An earthquake catalog data file.

```
R  26066 NV 1872 03 14 00:00:00.0 39.5 -117.0   G 54   3 ALG 3.0
R  26068 NV 1872 03 26 00:00:00.0 39.5 -117.0   G 54   3 ALG 3.0
R  26075 NV 1873 03 24 19:00:00.0 39.5 -117.0   G 54   2 ALG 2.3
R  26085 NV 1881 12 08 01:50:00.0 39.5 -116.0   G 54   4 ALG 3.7
```

Appendix C. Program Listing

```
c This program reads an earthquake epicenter from a
c catalog file and determines wheather the epicenter
c lies within a given source zone. If it does, it is
c appended to a subcatalog created for that source zone.
c The program exhausts the earthquake catalog before
c proceeding to the next source zone found on a partial
c risk program input file.
c When all source zones have been exhausted, the program
c asks for any additional catalog files to be considered.
c
c The earliest version of this program was written
c about 1977 by Bernice Bender using algorithms found
c in her 1972 risk map program (Algermissen and
c others, 1976), to determine wheather a given
c geographical point lay inside  or outside a given
c source zone quadrilateral.
      character outnam*5,innam*20
      character a1*40,a2*1,a3*24
      character source*13,segout*13,znam*4
      dimension x(4),y(4),aa(50,4),bb(50,4),c(50,4),xsav(51,2),
     1ysav(51,2),ibr(10)
      common /equake/a1(6500),a2(6500),a3(6500)
       common/equake2/yv(6500),xv(6500)
c
c     nxy=number of xv=long, yv=lat values to test for being in or
c     out of seismic region--assume xv, yv in decimal degrees
c
       iflag=0
      write(6,1)
1     format(x,'Enter the risk program input file name containing ',
     * 'source zones.')
      read(5,200) source
200   format(a13)
       if(iflag.eq.1) go to 600
250   write(6,230)
230   format(1x,'Enter 0 if zones are degree minutes',/,
     *'Enter 1 if zones are degree decimal.')
      read(5,*)degflg
      if(degflg.eq.1.or.degflg.eq.0) go to 260
      go to 250
260   continue
      write(6,30)
30    format(1x,'Enter output file name without extention and exactly',
     *' 5 characters.',/,1x,'Example rocky or newmd etc.')
      read(5,31)outnam
31    format(a5)
600   continue
505    continue
      write(6,32)
```

```
32     format(1x,'Enter the catalog file name or quit to stop run.')
       read(5,33) innam
33     format(a20)
        if(innam.eq.'quit') go to 506
507     continue
        open(59,file=source,status='old')
        open(60,file=innam,status='old')
508     continue
       call datain(nxy)
c
        read(59,51,end=500)
51      format(18(/))
21     nbr=0
       ist=0
       iend=0
        read(59,50,end=500)inum,znam
        write(6,*)inum,znam
50      format(2(/),i3,46x,a4)
        if(inum.eq.99) go to 500
        encode(13,3,segout)outnam,znam
3       format(a5,a4,'.dat')
        open(61,file=segout,status='unknown',access='append')
   20 format(3i3)
c
c    read in jseg=number of cards in this set containing pairs of
c     (long,lat) coordinates
c    ifr=set number
c    itot=total number of sets --for last set,ifr=itot
c
   23 read(59,20,end=500) jseg,ifr,itot
   26 ist=iend+1
      iend=ist+jseg-1
c
c    edges of quadrilateral at
c           xsav(ii,1),ysav(ii,1)--xsav(ii,2),ysav(ii,2)
c           xsav(ii+1,1),ysav(ii+1,1)-xsav(ii+1,2),ysav(ii+1,2)
c    points with subscript (ii,1) and(ii+1,1) form an edge
c    points with subscript (ii,2) and(ii+1,2) form an edge
c      points with subscript(ii,1) and (ii+1,2) are on a diagonal
c      coordinates read in decimal degrees
c
      do 24 ii=ist,iend
      read(59,220)   (xsav(ii,i),ysav(ii,i), i=1,2)
220   format(4f6.2)
   24 continue
      if(degflg.eq.1) go to 240
      do 224 kk=ist,iend
      do 224 k=1,2
      xphi=xsav(kk,k)
      yphi=ysav(kk,k)
      call condec(xphi)
```

```
      call condec(yphi)
      xsav(kk,k)=xphi
      ysav(kk,k)=yphi
224   continue
240   continue
      if(ifr.eq.itot) go to 25
      nbr=nbr+1
      ibr(nbr)=iend
      go to 23
   25 num=iend
      nm=num-1
c
c   set up equations for lines in seismic source region
c
      ij=0
      do 71 ii=1,nm
      if(nbr.eq.0) go to 46
      do 45 iq=1,nbr
      if(ii.eq.ibr(iq)) go to 71
   45 continue
   46 do 44 i=1,2
      x(i)=xsav(ii,i)
      x(i+2)=xsav(ii+1,i)
      y(i)=ysav(ii,i)
   44 y(i+2)=ysav(ii+1,i)
      ij=ij+1
      call lin(x(1),y(1),x(2),y(2),aa(ij,1),bb(ij,1),c(ij,1),x(3),y(3))
      call lin(x(1),y(1),x(3),y(3),aa(ij,2),bb(ij,2),c(ij,2),x(2),y(2))
      call lin(x(2),y(2),x(4),y(4),aa(ij,4),bb(ij,4),c(ij,4),x(1),y(1))
      call lin(x(3),y(3),x(4),y(4),aa(ij,3),bb(ij,3),c(ij,3),x(1),y(1))
   71 continue
      do 80 i=1,nxy
      do 75 jj=1,ij
      do 73 ii=1,4
      d=aa(jj,ii)*xv(i)+bb(jj,ii)*yv(i)+c(jj,ii)
      if(d.le.-1.e-8) go to 75
   73 continue
c
c  the point xv(i), yv(i) is in the given source region
c   here save of process as required
c
      call dataout(i)
      go to 80
   75 continue
   80 continue
       close(61)
      go to 21
500   continue
       iflag=1
       close(59)
       go to 505
```

```
506     continue
        close(60)
       stop
       end
       subroutine lin(x1,y1,x2,y2,a,b,c,x3,y3)
c
c  Coefficients a, b, c are determined so that points which are in the
c    same half plane as x3,y3 will be a positive distance from the line
c   joining x1,y1 and x2,y2
c
       if(x1.eq.x2) go to 10
       a=(y1-y2)/(x1-x2)
       b=-1.
       c=y1-a*x1
       go to 12
   10  a=1.
       b=0.
       c=-x1
   12  d=a*x3+b*y3+c
       if(d.gt.0.) return
       a=-a
       b=-b
       c=-c
       return
       end
       subroutine datain(j)

c The input and output formats are fixed to
c preserve the data record and can be altered
c to accomodate a differently formated catalog.

       character a1*40,a2*1,a3*24
       character*20 zz,zz20
       character z*4,z20*4
       integer pas,brk
       common /equake/a1(6500),a2(6500),a3(6500)
        common/equake2/dd1(6500),dd2(6500)
       j=1
 3     read(60,1,end=200) a1(j),d1,a2(j),d2,a3(j)
 1     format(a40,f5.2,a1,f7.2,a24)
       dd1(j)=d1
       dd2(j)=-d2
c
c              If east longitudes are encountered this program may have
c              to be changed to read E or W off the input file and
c              change signs accordingly
c
       j=j+1
       goto 3
200    j=j-1
       return
```

```
      entry dataout(n)
      if(dd1(n).lt.0) dd1(n)=-dd1(n)
      if(dd2(n).lt.0) dd2(n)=-dd2(n)
      dd2(n)=-dd2(n)
      write(61,6) a1(n),dd1(n),a2(n),dd2(n),a3(n)
6     format(1x,a40,f4.1,a2,f6.1,a24)
      return
      end
       subroutine condec(phi)
       i=phi
       fr=phi-i
       phi=i+fr/.60
       return
       end
```

Statistical Analysis

Description

After the earthquakes have been collected and placed in separate catalogs by zone, these catalogs are examined and the events are counted by intensity level and decade. It is at this point that the foreshocks and aftershocks can be removed if they were not already removed from the master catalog. (fig. F, see part B, INTY6.FOR near connector J.)

Program INTY6.FOR is an interactive program that summarizes historic seismicity in the form of decade tables. For each zone a summary decade table is printed at the user's terminal. The output is a matrix of number count of events arranged by size (intensity) and by decade, with the starting decade being 1970-1979 and going back in history to some pre-determined stating date.

This program operates only on intensity. In many applications, magnitudes of one species or another are more reliable than intensity. This program exists in another version which permits decade counts for magnitude ranges, and in which magnitudes and intensities are converted to whatever magnitude has been chosen to be most reliable.

It should be noted here that the aftershock removal can occur before processing through INTY6.FOR by editing them out of the catalogs altogether, or after processing through INTY6.FOR by simply examining the output catalog and noting the aftershocks, then editing the decade tables to properly reduce the decade number count.

These decade tables are used to analyze the completeness of the historical earthquake data and to calculate b-values.

Input File Description

The input files are the subcatalog file that were produced by the program REGION1.FOR for each source zone. These data files have a unique nine-character name and contain earthquake data by individual zones (see appendix A at the end of this section for an example).

Output File Description

There are two output files produced for each source zone, an earthquake catalog file and a decade table file. The catalog file is a subset of the input catalog with the same format and containing only those events greater than or equal to intensity V. The decade tables consist of a number count of events by intensity (column) and decade (row). The decades start with 1970's and go back in history to a pre-determined starting decade. The most recent decade appears at the top row of the table, the earliest decade at the bottom. The intensity columns are from V to XII from left to right (see appendix B at the end of this section for an example).

A third output is created for each zone and printed at the users terminal. This output is a decade table summary containing only those decades with at least one event of Intensity V or greater for a particular decade (see appendix C at the end of this section for an example).

Program Operation

Program INTY6.FOR (Appendix D) is an interactive program that first asks for a desired earliest decade of interest, then recursively asks for the names of subcatalogs produced by the program REGIONI.FOR. INTY6.FOR steps through the process of producing decade tables, summary decade tables, and purged catalogs, zone by zone. The output file name is created by decoding the name out of the input file name and encoding it into a unique name. The decade table output file name takes the name "dcint" followed by the source zone name. The purged catalog file name takes on the name "int" followed by the source zone name.

The program continues to process until the word "quit" is entered for the next input file.

Appendix A. Sample Input Data File

Earthquake data file for an individual source zone.

s 23085	NH	1800	12	20	00:00:00.0	43.7	-72.3	0	H	126	4	126	3.7
s 22704	VT	1843	03	14	00:00:00.0	44.4	-72.5	0	H	76	4	76	3.7
s 23175	NH	1845	11	00	00:00:00.0	43.6	-72.3	0	H	126	4	126	3.7
s 23291	NH	1852	06	30	00:00:00.0	43.4	-72.3	0	G	126	3	126	3.0
s 23302	NH	1855	05	29	10:00:00.0	44.7	-71.6	0	H	126	4	126	3.7
s 22726	VT	1856	06	10	00:00:00.0	43.1	-72.5	0	H	76	2	76	2.3
R+29956	CN	1858	05	10	00:00:00.0	45.7	-72.1	0	I	76	3	ALG	3.0

Appendix B. Sample Decade Table

An example decade table for an individual source zone.
Format 8i3

18	1	0	1	0	0	0	0
9	2	1	0	0	0	0	0
2	1	0	0	0	0	0	0
3	1	0	0	0	0	0	0
1	2	2	1	0	0	0	0
3	0	0	3	0	0	0	0
2	2	0	0	0	0	0	0
3	0	0	2	0	0	0	0
0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Appendix C. Summary Decade Table

Summary decade table for an individual source zone.

```
 Enter the entire input file name (XXXXXXNNN.dat) or quit to stop the run.
>ecnewi106
```

decade	V	VI	VII	VIII	IX	X	XI	XII	TOTAL
1870-1879	0	1	0	0	0	0	0	0	1
1880-1889	1	0	1	1	1	1	1	1	3
1900-1909	3	0	0	2	0	0	0	0	5
1910-1919	2	2	0	0	0	0	0	0	4
1920-1929	3	0	0	3	0	0	0	0	6
1930-1939	1	2	2	1	0	0	0	0	6
1940-1949	3	1	0	0	0	0	0	0	4
1950-1959	2	1	0	0	0	0	0	0	3
1960-1969	9	2	1	0	0	0	0	0	12
1970-1973	14	1	0	1	0	0	0	0	20

```
 total number of earthquakes with intensity Int > or = V =  64
 Enter the entire input file name (XXXXXXNNN.dat) or quit to stop the run.
>quit
 FortrAN STOP
```

Appendix D. Program Listing

```
c This program is designed to read subcatalogs
c created by REGION1.FOR and produce a decade
c table and a summary decade table for each
c source zone subcatalog file input.
c This program operates only on Intensities
c and will discard any event with intensity
c less  than 5.
c The output decade table and the summary
c decade table show only Intensities 5-12.
c The summary decade table output to the user's
c terminal shows only those decades with valid
c entries, while the decade table output to a
c file contains an entry for each Intensity
c level and each decade back to some starting
c date input by the user.
c       Written by Stanley L. Hanson
c               October 1981
c
      dimension m(9)
       integer dectab(100,8)
      character source*13,segout*11,n6*5
       character nnn*4
       character decout*14
      character pb*3
      character var2*45,var3*9
      real ms,ml,mb,mag,mag1,mag2
      integer icntry,iyr,int
       integer dectyr,comdec,dmax,dplus1

       comdec=900
       write(6,500)
500    format(1x,'Enter the earliest decade of interest i.e., 1530')
       read(5,501)comdec
501    format(i4)
       dmax=(1970-comdec)/10
       dplus1=dmax+1
400   write(6,101)
  101 format(' Enter the entire input file name (XXXXXXNNN.dat) ',
     * 'or quit to stop the run.')
      read(5,102) source
102    format(a13)
      if(source.eq.'quit') go to 999
       iflag=0
      decode(9,103,source) n6,nnn
103   format(a5,a4)
       write(6,103)n6,nnn
      encode(11,104,segout) nnn
104   format('int',a4,'.dat')
       encode(13,250,decout) nnn
```

```
250     format('dcint',a4,'.dat')
        open(10,file=source,status='old')
        open(11,file=segout,status='new')
        open(12,file=decout,status='new')
      do 199 i=1,9
  199 m(i)=0
      write(6,200)
  200 format(1x,'decade        V        VI        VII        VIII      ',
     *'    IX        X         XI        XII   TOTAL',/)
      ii=0
      number=4404
      nn=0
      do 3 i=1,number
      read(10,1,end=100) icntry,iyr,var2,int,var3
1     format(18x,i1,i2,a44,i2,a9)
       dectyr=(icntry*100)+iyr+1000
       if(int.ne.0.0) go to 130
       go to 3

130    continue
      write(11,5)icntry,iyr,var2,int,var3
5     format(16x,'1',i1,i2,a45,i2,a9)
       if(int.lt.5) go to 3
       idec=((dectyr-comdec)/10)+1
      ii=ii+1
  209 yr=iyr/10
      iiyr=yr
       ifyr=iyr-(iiyr*10)
      nyr=icntry*10+iiyr
      if(ii.eq.1) lastyr=nyr
      if(nyr.eq.lastyr) go to 204
  208 jyr=lastyr*10
      if(iflag.eq.0) kyr=(jyr+9)+1000
       if(iflag.eq.1) kyr=(jyr+ifyr)+1000
       jyr=jyr+1000
      write(6,206) jyr,kyr,(m(j),j=1,9)
  206 format(/,1x,i4,'-',i4,7(i6,2x),i5,i6)
      if(nn.gt.0) go to 901
      do 205 k=1,9
  205 m(k)=0
  204 m(9)=m(9)+1
      if(int.eq.5) m(1)=m(1)+1
      if(int.eq.6) m(2)=m(2)+1
      if(int.eq.7) m(3)=m(3)+1
      if(int.eq.8) m(4)=m(4)+1
      if(int.eq.9) m(5)=m(5)+1
      if(int.eq.10) m(6)=m(6)+1
      if(int.eq.11) m(7)=m(7)+1
      if(int.eq.12) m(8)=m(8)+1
      if(int.eq.5)dectab(idec,1)=dectab(idec,1)+1
      if(int.eq.6)dectab(idec,2)=dectab(idec,2)+1
```

```
      if(int.eq.7)dectab(idec,3)=dectab(idec,3)+1
      if(int.eq.8)dectab(idec,4)=dectab(idec,4)+1
      if(int.eq.9)dectab(idec,5)=dectab(idec,5)+1
      if(int.eq.10)dectab(idec,6)=dectab(idec,6)+1
      if(int.eq.11)dectab(idec,7)=dectab(idec,7)+1
      if(int.eq.12)dectab(idec,8)=dectab(idec,8)+1
      lastyr=nyr
    3 continue
  100 nn=i-1
       iflag=1
      go to 208
  901 continue
      write(6,17) ii
   17 format(/,1x,'total number of earthquakes with intensity ',
     *'Int > or = V =',i5)
       do 220 id=1,dmax
       iid=(dplus1+1)-id
       write(12,225)(dectab(iid,im),im=1,8)
225    format(8i3)
220    continue
       close(10)
       close(11)
       close(12)
       do 230 id=1,100
       do 235 im=1,8
       dectab(id,im)=0
235    continue
230    continue
      go to 400
999   stop
      end
```

Raw Rate Calculation

Description

The decade tables produced in the previous step are now used to calculate the raw rates for each intensity level for each decade using the program LAMDAH.FOR. (See fig. F, connector A.)

LAMDAH.FOR (see Appendix C at the end of this section for a program listing) assesses mean rate but also transforms the mean rate data for the purpose of making a plot to enable the anaylst to assess completeness according to the method of Stepp (1973).

If the occurrence data are from a Poisson distribution, the variance of the observed number per interval is equal to the mean observed number.

$$\lambda = \text{MEAN} = \frac{1}{h} \sum_{i=1}^{h} K_i$$

where K = number of earthquakes per unit time interval
and h = number of time intervals, here years of accumulated decades.

In turn, the variance of the mean observed number is the variance of the observed number divided by the number of intervals. Thus the standard deviation of the mean rate per year is given by:

$$\sigma_\lambda = \text{STANDARD DEVIATION} = \frac{\sqrt{\lambda}}{\sqrt{T}}$$

where T = sample length of time.

The program calculates not only lambda, but sigma as well. It is clear that one is just a transformation of the other.

If we plot lambda and sigma as a function of LOG T we can assess lambda by its stability in time about a level value and we assess sigma by its stability around a curve whose slope is LOG $1/\sqrt{T}$.

Input File Description

The decade tables produced by the program INTY6.FOR are the input for this program. The rows are the decades from 1970 back in history from the top down, and the columns are the Intensity levels V through XII from left to right across the top. (See appendix A at the end of this section for an example input file.)

Output File Description

There are two output files produced by this program, one file contains the calculated lambda values and the second contains the calculated sigma values.

NOTE: If the calculated lambda or sigma value is equal to zero, that value can be replaced by the value .0001. This may be necessary to eliminate error messages from plot programs, other than the one that follows, when it tries to plot zero on a log-log plot. (See appendix B at the end of this section for an example file.)

Program Operation

The program (see Appendix C at the end of this section for program listing) is an interactive program that first asks for the number of input decade table files to be combined. Combining decade tables for a number of zones is sometimes necessary in regions of low seismic activity in order to have a large enough data set for reliable estimates of earthquake recurrences. The program recursively asks for the input files and sums the input decade-table values by row and by column.

The output lambda and sigma values are calculated and stored in two internal arrays.

The user is asked to supply unique output file names for the two output files. At this time the program writes the lambda and sigma values to the appropriate files and the program run is terminated.

Appendix A. Sample Input File

An input decade table
Format 8i3

0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Appendix B. Sample Output File

An example output of mean values.
Format i3,8(f7.4,1x,i1)
Column 1 is the decade in years, in this case the decade of the 70's is complete for only 7 years.
Columns 2,4,6,8,10,12,14, and 16 are the mean values for each of the Intensity levels.
Columns 3,5,7,9,11,13,15, and 17 are the column indicators associated with the Intensity column to their imediate left.

```
  7 8.0000 1 3.6667 2 1.5000 3 0.6667 4 0.1667 5 0.0000 6 0.0000 7 0.0000 8
 17 7.5625 1 4.7500 2 1.5625 3 0.8125 4 0.1250 5 0.0000 6 0.0000 7 0.0000 8
 27 4.6538 1 2.9615 2 1.0769 3 0.5769 4 0.2308 5 0.0385 6 0.0000 7 0.0000 8
 37 3.3611 1 2.1389 2 0.7778 3 0.4167 4 0.1944 5 0.0556 6 0.0000 7 0.0000 8
 47 2.6304 1 1.6739 2 0.6304 3 0.3261 4 0.1739 5 0.0435 6 0.0000 7 0.0000 8
 57 2.1607 1 1.3750 2 0.5179 3 0.2679 4 0.1429 5 0.0536 6 0.0000 7 0.0000 8
 67 1.8333 1 1.1667 2 0.4394 3 0.2273 4 0.1212 5 0.0455 6 0.0000 7 0.0000 8
 77 1.5921 1 1.0263 2 0.3816 3 0.1974 4 0.1053 5 0.0395 6 0.0000 7 0.0132 8
 87 1.4070 1 0.9070 2 0.3372 3 0.1744 4 0.0930 5 0.0349 6 0.0000 7 0.0233 8
 97 1.2604 1 0.8125 2 0.3021 3 0.1563 4 0.0833 5 0.0313 6 0.0000 7 0.0208 8
107 1.1415 1 0.7358 2 0.2736 3 0.1415 4 0.0755 5 0.0283 6 0.0000 7 0.0189 8
117 1.0431 1 0.6724 2 0.2500 3 0.1293 4 0.0690 5 0.0259 6 0.0000 7 0.0172 8
127 0.9603 1 0.6190 2 0.2302 3 0.1190 4 0.0635 5 0.0238 6 0.0000 7 0.0159 8
137 0.8897 1 0.5735 2 0.2132 3 0.1103 4 0.0588 5 0.0221 6 0.0000 7 0.0147 8
147 0.8288 1 0.5342 2 0.1986 3 0.1027 4 0.0548 5 0.0205 6 0.0000 7 0.0137 8
157 0.7756 1 0.5000 2 0.1859 3 0.0962 4 0.0513 5 0.0192 6 0.0000 7 0.0128 8
167 0.7289 1 0.4699 2 0.1747 3 0.0904 4 0.0482 5 0.0181 6 0.0000 7 0.0120 8
177 0.6875 1 0.4432 2 0.1648 3 0.0852 4 0.0455 5 0.0170 6 0.0000 7 0.0114 8
187 0.6505 1 0.4194 2 0.1559 3 0.0806 4 0.0430 5 0.0161 6 0.0000 7 0.0108 8
197 0.6173 1 0.3980 2 0.1480 3 0.0765 4 0.0408 5 0.0153 6 0.0000 7 0.0102 8
```

Appendix C. Program Listing

```
c  PROGRAM LAMDA IS DESIGNED TO READ ONE OR SEVERAL
c  INPUT DATA FILES THAT CONTAIN THE DECADE INFORMATION
c  BY INTENSITY LEVEL AND CALCULATE A LAMBDA AND SIGMA
c  FOR EACH DECADE AND EACH INTENSITY LEVEL.
c
c  THE SIGMA VALUE IS THE EXPECTED STANDARD DEVIATION OF
c  THE MEAN RATE, CALCULATED ACCORDING TO THE ASSUMPTION
c  THAT THE OCCURRENCE DATA IS POISSIONIAN, AND HENCE THE
c  SIGMA VALUE IS TAKEN TO BE A TRANSFORMATION OF LAMBDA.
c  THIS CALCULATION IS MADE TO ENABLE THE ESTIMATION OF
c  MEAN RATE OF OCCURRENCE ACCORDING TO A TECHNIQUE
c  BY STEPP (1973) Analysis of completeness of the earthquake
c  sample in the Puget Sound area, Harding, S. T., ed.
c  Contributions to seismic zonings: National Oceanic
c  and Atmospheric Administration Technical Memorandum
c  ERL ESL-13, 38 p.
c
c  INPUT:
c       DECADE TABLE FOR EACH ZONE OR GROUP OF ZONES.
c  OUTPUT:
c        TWO FILES FOR EACH SET OF INPUT.
c  VARIABLES:
c           1)decade(100,8)-INPUT ARRAY.
c           2)sumdec(100,8)-SUMS OF ALL INPUT ARRAYS.
c           3)sigma(100,8)-S. D. VALUES.
c           4)lamda(100,8)-CALCULATED MEAN FROM INPUT.
c           5)date(100)-DATA STATEMENT OF YEARS,COMPLETE TO 1977
c           6)nfile-NUMBER OF INPUT FILES TO BE COMBINED..
c           7)name-NAME OF INPUT FILE.
c           8)name1-NAME OF OUTPUT FILE.
c           9)name2-NAME OF OUTPUUT FILE.
c          10)sum-ACCUMULATOR FOR CALCULATIONS.
c          11)iseed-STARTING SEED FOR COMPLETION DATE -3=1977
c
      dimension decade(100,8),sumdec(100,8),sigma(100,8)
      real lamda(100,8)
      integer date(100),y(8)
      data(y(i),i=1,8)/1,2,3,4,5,6,7,8/
      character name*20,name1*20,name2*20
      write(6,5)
5     format(1x,'Enter the number of files to be combined.')
      read(5,*) nfile
      do 15 i=1,nfile
      write(6,20)
20    format(1x,'Enter the input file name.')
      read(5,25) name
25    format(a20)
```

```
      open(unit=8,file=name,status='old')
c
c  READ IN EACH INPUT DATA FILE AND SUM ALL INPUT IN
c  THE ARRAY SUMDEC.
c
      do 30 k=1,100
      read(8,35,end=99)(decade(k,j),j=1,8)
35    format(8f3.0)
30    continue
99    continue
      ndec=k-1
      close(8)
      iseed=-3
      do 40 k=1,ndec
     iseed=iseed+10
      date(k)=iseed
      do 40 j=1,8
      sumdec(k,j)=sumdec(k,j)+decade(k,j)
40    continue
15    continue
c
c  SUM EACH COLUMN AND CALCULATE A LAMDA FOR EACH ROW OF EACH COLUMN.
c
      do 45 j=1,8
      sum=0.0
      do 45 k=1,ndec
      sum=sum+sumdec(k,j)
      lamda(k,j)=sum/date(k)
45    continue
c
c  CALCULATE THE SIGMA VALUE FOR EACH OF THE LAMDA's
c
      do 55 k=1,ndec
      do 55 j=1,8
      root1=lamda(k,j)**.5
      root2=date(k)**.5
      sigma(k,j)=root1/root2
55    continue
      write(6,60)
60    format(1x,'Enter the output file name for lamda. ')
      read(5,65) name1
      write(6,61)
61    format(1x,'Enter the output file name for sigma sub lamda.')
65    format(a20)
      read(5,65) name2
      open(unit=7,file=name2,status='new')
      open(unit=9,file=name1,status='new')
c
c  WRITE OUT THE TWO DATA FILES FOR PLOTTING.
c
      do 70 k=1,ndec
```

```
      write(9,75) date(k),(lamda(k,j),y(j),j=1,8)
      write(7,75) date(k),(sigma(k,j),y(j),j=1,8)
75    format(1x,i3,8(f7.4,1x,i1))
70    continue
      close(7)
      close(9)
      stop
      end
```

Raw Rate Plotting

Description

The rate tables calculated by the program LAMDAH.FOR in the previous step are now plotted for ease of inspection. They can be plotted by hand or by using LAMPLOT.FOR (fig. F, see part C, near connector A).

The program LAMPLOT.FOR (Appendix B) is designed to plot the output from the program LAMDAH.FOR on a log-log graph. The program uses the NCAR graphics software and produces a metacode plot file that is device independent. The metacode file must be run through a post-processing program, MCPLOT, that will translate the input and direct it to a particular output device.

The program will plot up to eight curves on one graph. The extremes of the graph are calculated internally from the input data to define the plot scaling.

Input File Description

The input to this program is a table of rates per decade for each of the eight Intensity levels produced by the program LAMDAH.FOR (see appendix A at the end of this section for an example).

Output File Description

Output is a device-independent metacode plot file that will need post-processing to a particular plot device. (See the NCAR software documentation for more information on plotting this data.) This metacode data file will be created in the users current working directory and given the name pltfl.dat. With each succesive run of the program this data file will be over-written. (See fig. E for an example plot.)

Program Operation

This program (Appendix B) is an interactive plot program first asking for an input data file. It reads this data file and finds the maximum and minimum values for both x and y to be used to scale the plot. Next the number of whole log cycles are calculated for both the x and the y axes so as to include all the data points. At this point the program generates a metacode plot file containing from one to eight curves, one curve for each intensity level, on a single log-log graph.

At program termination the user must run the post processing program, MCPLOT, selecting the appropriate output device.

Appendix A. Sample Input File

Input data file.
Format i3,8(f7.4,1x,i1)

6	8.0000	3.6667	1.5000	0.6667	0.1667	0.0000	0.0000	0.0000
16	7.5625	4.7500	1.5625	0.8125	0.1250	0.0000	0.0000	0.0000
26	4.6538	2.9615	1.0769	0.5769	0.2308	0.0385	0.0000	0.0000
36	3.3611	2.1389	0.7778	0.4167	0.1944	0.0556	0.0000	0.0000
46	2.6304	1.6739	0.6304	0.3261	0.1739	0.0435	0.0000	0.0000
56	2.1607	1.3750	0.5179	0.2679	0.1429	0.0536	0.0000	0.0000
66	1.8333	1.1667	0.4394	0.2273	0.1212	0.0455	0.0000	0.0000
76	1.5921	1.0263	0.3816	0.1974	0.1053	0.0395	0.0000	0.0132
86	1.4070	0.9070	0.3372	0.1744	0.0930	0.0349	0.0000	0.0233
96	1.2604	0.8125	0.3021	0.1563	0.0833	0.0313	0.0000	0.0208
106	1.1415	0.7358	0.2736	0.1415	0.0755	0.0283	0.0000	0.0189
116	1.0431	0.6724	0.2500	0.1293	0.0690	0.0259	0.0000	0.0172
126	0.9603	0.6190	0.2302	0.1190	0.0635	0.0238	0.0000	0.0159
136	0.8897	0.5735	0.2132	0.1103	0.0588	0.0221	0.0000	0.0147
146	0.8288	0.5342	0.1986	0.1027	0.0548	0.0205	0.0000	0.0137
156	0.7756	0.5000	0.1859	0.0962	0.0513	0.0192	0.0000	0.0128
166	0.7289	0.4699	0.1747	0.0904	0.0482	0.0181	0.0000	0.0120
176	0.6875	0.4432	0.1648	0.0852	0.0455	0.0170	0.0000	0.0114
186	0.6505	0.4194	0.1559	0.0806	0.0430	0.0161	0.0000	0.0108
196	0.6173	0.3980	0.1480	0.0765	0.0408	0.0153	0.0000	0.0102

```
c Program to plot the "lambda" or "sigma"
c values generated by the program lamdah.
c This program uses the NCAR graphics software.
c The program will plot only a log-log graph
c and it scales itself to the nearest whole
c cycle depending on the input data set.
c Written May 82 by Stanley L. Hanson
c
c Output is a metacode data file that can
c be directed to a particular device at
c processing time.
c When compling and linking this program the
c NCAR library must be linked with it.
c  link .............,sys$sysdisk:[NCAR]NCAR.OLB/lib
c
       dimension x(100),y2d(100,8),y(100)
       character nam*13,filbt*2,nam2*16
       ymin=.5
       ymax=.5
       write(6,25)
25     format(1x,'Enter the input filemane.')
       read(5,26)nam
26     format(a13)
       open(8,file=nam,status='old')
       do 10 i=1,100
       read(8,15,end=99)x(i),(y2d(i,j),j=1,8)
15     format(f4.0,8(f7.4,2x))
c
c do loop 20 determines the max and min
c values of the y ordinate.
c
       do 20 k=1,8
       if(y2d(i,k).eq.0.0) go to 20
       if(y2d(i,k).lt.ymin)ymin=y2d(i,k)
       if(y2d(i,k).gt.ymax)ymax=y2d(i,k)
20     continue
10     continue
99     continue
       close(8)
       ipts=i-1
       seed=.001
c
c do loops 35 and 38 determine nearest
c whole upper cycle bound and the nearest
c lower whole cycle bound respectfully.
c
       do 35 i=1,10
```

```
      if(ymax.le.seed) go to 36
      seed=seed*10.0
35    continue
36    ymax=seed
      seed=1.0
      do 38 i=1,10
      if(ymin.ge.seed) go to 39
      seed=seed/10.0
38    continue
39    ymin=seed
c
c NCAR graphic subroutines for
c titling,windowing,scaling
c and plotting the input data.
c
      call anotat(6hYEARS$,6hRATES$,2,1,1,0)
      call agseti(10hWINDOWING.,1)
      call agsetf(6hX/LOG.,-1.)
      call agsetf(6hY/LOG.,-1.)
      call agsetf(10hX/MINIMUM.,1.)
      call agsetf(10hX/MAXIMUM.,1000.)
      call agsetf(10hY/MINIMUM.,ymin)
      call agsetf(10hY/MAXIMUM.,ymax)
      call agsetf(11hLABEL/NAME.,4h   T)
      call agseti(12hLINE/NUMBER.,100)
      call agsetp(10hLINE/TEXT.,11hLAMDA PLOT$,1)
      call agstup(x,ipts,1,ipts,1,y,ipts,1,ipts,1)
      call agback
      do 40 j=1,8
      DO 44 IK=1,ipts
      Y(IK)=Y2D(IK,J)
44    CONTINUE
      CALL AGCURV(x,1,y,1,ipts,-1)
40    continue
      CALL FRAME
      stop
      end
```

Redistribution of Rates

Description

The redistribution or "back allocation" of rates to each of the combined zones is done by the program BACK3.FOR (see Appendix A at the end of this section for program listing). The input for this step in the system are a set of expected rates from the program ITER3.FOR and the decade table sums from the program INTY6.FOR. (See fig. F, connector B.)

BACK3.FOR is designed to "back-allocate" expected future earthquake rates to a maximum of 30 homogeneous zones that were previously combined and processed through LAMDAH.FOR and ITER3.FOR. The program uses six different techniques to back allocate the occurence rates.

They are:

1) Summing the observed number of events then finding the percentage of the total number contributed by each zone. This produces a maximum likelihood "back-allocation".
2) Back allocating according to the distribution of observed intensity VI's in each zone. For many regions in the U. S., intensity VI earthquakes are completely reported for the past 70-100 years and provide sufficiently large numbers of earthquakes to represent relative average recurrence rates between zones.
3) Calculating equivalent VI's and finding the percentage of each contribution by zone. Given the expected rates in the first two intensity categories provided by a fit to a Richter law occurrence, the computer extracts the b-value implied. One intensity V is equivalent to 10^{-b} VI's, one intensity VII is equivalent to 10^{b} VI's etc. These are equivalent in the sense that a Richter law curve of a given b-value through a given number at Intensity VI will yield the equivalent number at the other Intensities.
4) Calculating an "A" value for each of the observed numbers in each category,averaging the a-value for each intensity category for each zone, the percentage of contribution to each zone is proportional to the 10^{a} of each zone.
5) Calculating a weighted "A" value and finding the percentage of each contribution by zone as with 4) except the A for each zone is a weighted average, weighted by observed numbers in each category.
6) Calculating a Chi-sqr value and finding the

percentage of each contribution by zone as with 4) except the A for each zone is the value which minimizes the sum of the Chi-sqr for the given category observations, assuming the given b-value.

The program accepts the input from the users terminal. Output is displayed at the users terminal and consists of a table of rates listed by zone and Intensity level for each of the six methods.

It is up to the user to select the method he desires for the annual rates for each source zone. Because of the characteristics of the completeness of the historical data one or another of the methods used may be judged to best represent the analyst's opinion as to the relative weight desired for each zone. For random data, when there is no completeness role, method 1 is best overall, but somewhat biased since numbers per zone are often small. Minimum variance seems to be achievable by averaging 1 and 3 since they seem to bias in opposite directions.

Techniques (1) and (6) give the best results in sparse samples of data derived from zones having equal rates and the same b-value.

Techniques (4) and (5) give results which most often correspond to an analyst's "feel" for what the relative rate should be for real data.

Input Description

The input to this program is entered from the terminal. The first input is the total number of source zones to be combined. The next is the zone name (four characters long) and the total number of events from the decade table for each Intensity level, (the column totals).

Example:

i106 2 1 0 0 0 0 0 0

An entry like that above is made for each zone.

Finally the expected values for each of the eight Intensity levels is entered. These expected values are usually derived from the picks of the plotted mean and standard deviation values from the program LAMDAH.FOR and calculated in the program ITER3.FOR.

Example:

V	VI	VII	VIII	IX	X	XI	XII
2.8045	.8954	.2859	.0913	.0291	.0093	.003	.0009

Artificial sequences could be used to get allocation weights for different assumed b-values.

Output Description

The output from this program is printed directly at the users terminal. First there is a summary table of the input zone by zone and their corresponding percentages of contribution "weights" to the total for each of the six methods of back allocation.

Then for each of the six back allocation methods a table of annual rates, arranged according to Intensity level, is produced for each zone. An example of this output would be to lengthy to include in this documentation, therefore a sample test should be made by the user ,using at least two zones and the sample expected values above.

Program Operation

The program is an interactive program that asks for the number of zones that will be combined. Then the name and column sums from the decade tables for each zone to be combined, and finally the table of expected values.

The program then performs the six different methods of distribution and produces a table of each. These tables are then displayed on the users terminal and the program is terminated.

```
c THIS PROGRAM IS DESIGNED TO BACK ALLOCATE EARTHQUAKES UP
c TO 30 ZONES THAT ARE TO BE COMBINED TO FORM ONE LARGE ZONE.
c THIS PROGRAM USES SIX DIFFERENT METHODS TO BACK ALLOCATE THE
c EVENTS BY ZONE ACCORDING TO:
c  1)SUMMING THE EVENTS BY ZONE AND THEN FINDING A PERCENTAGE
c  2)LOOKING AT THE INTENSITY VI DISTRIBUTION
c  3)CALCULATING EQIVELENT VI's AND FINDING A PERCENTAGE
c  4)CALCULATING THE 'A' VALUE AND FINDING A PERCENTAGE
c  5)CALCULATING A WEIGHTED "A" VALUE AND FINDING A PERCENTAGE
c  6)CALCULATING A CHI-SQR VALUE AND FINDING A PERCENTAGE
c
c   INPUT:
c        INPUT IS FROM THE TERMINAL
c
c   OUTPUT:
c        OUTPUT IS TO THE TERMINAL
c
c   VARIABLES:
c            exv(8)-EXPECTED VALUES FOR COMBINED ZONES-INPUT
c            tm(8)-TOTALS OF ALL ZONES BY INTENSITY
c            tz(8)-TOTALS OF ALL INTENSITIES BY ZONES
c            tze(8)-TOTALS OF EQUIVALENT VI's BY ZONES
c            pert(8)-P% OF DISTRIBUTION OF TOTAL BY ZONE
c            per6(8)-% DISTRIBUTION OF VI's BY ZONE
c            pere(8)-% DISTRIBUTION OF EQUIVALENT VI's BY ZONE
c            pera(8)-% DISTRIBUTION OF 'A' VALUE BY ZONE
c            aval(8)-CALCULATED 'A' VALUE
c            table(8,9)-ARRAY OF BACK ALLOCATED EVENTS-OUTPUT
c            ar(8,9)-ARRAY OF ZONE NO. AND EVENTS-INPUT
c            mag(8)-DATA STATEMENT ARRAY OF INTENSITY LEVELS
c            nar(8)-ARRAY OF ZONE NAMES-INPUT
c            z(30)-CHI-SQR VALUES
c            perz(30)-% DISTRIBUTION OF CHI-SQR BY ZONE
c            nzone-NUMBER OF ZONES TO BE COMBINED-INPUT
c            cnt-COUNTER
c            bval-CALCULATED 'B' VALUE
c            totz-TOTAL OF ALL EVENTS
c            totze-TOTAL OF ALL EQUIVALENT VI's
c            totza-TOTAL OF ALL 'A' VALUES
c

      dimension exv(8),pert(30),per6(30),tze(30)
      dimension pere(30),pera(30),table(30,8),aval(30)
      dimension perz(30),z(30)

      integer levnum(30),ar(30,9),tz(30),tm(8)

      real mag(8),avalw(30),na(30),antiwa(30),totwa,perwa(30)
```

```
      character nar(30)*6

      data (mag(i),i=1,8)/4.3,4.9,5.5,6.1,6.7,7.3,7.9,8.5/

c
c ENTER THE INPUT INFORMATION FIRST THE NUMBER OF ZONES TO BE
c COMBINED. THEN THE NAME OF EACH ZONE AND ITS NUMBER AND
c THE EVENTS FOR EACH INTENSITY LEVEL. FINALLY THE EXPECTED
c VALUES FOR EACH INTENSITY LEVEL FOR THE COMBINED ZONES.
c

      do 500 i=1,8
      tm(i)=0
      exv(i)=0.0
      do 505 k=1,30
      pert(k)=0.0
      per6(k)=0.0
      tze(k)=0.0
      pere(k)=0.0
      pera(k)=0.0
      table(k,i)=0.0
      aval(k)=0.0
      perz(k)=0.0
      z(k)=0.0
      levnum(k)=8
      ar(k,i)=0
      tz(k)=0
      avalw(k)=0
      na(k)=0
      antiwa(k)=0
      perwa(k)=0
      nar(k)='      '
505   continue
500   continue

      write(6,5)
5     format(1x,'Enter the number of zones to be combined.')
      read(5,*) nzone

      do 15 k=1,nzone
      write(6,21) levnum(k)
21    format(1x,'Enter zone no. and ',i1,'values.')
      read(5,*)(ar(k,i),i=1,levnum(k)+1)
15    continue

      write(6,30)
30    format(1x,'Enter the expected values for the 8 intensity level.')
      read(5,*) (exv(i),i=1,8)
```

```
c
c CALCULATE THE TOTALS BY INTENSITY.
c

      do 35 i=2,9
      l=i-1
      do 35 k=1,nzone
      tm(l)=tm(l)+ar(k,i)
35    continue

c
c CALCULATE THE 'B' VALUE.
c

      val=exv(2)/exv(1)
      bval=-alog10(val)/(mag(2)-mag(1))

c
c CALCULATE THE SUMS BY ZONE FOR EACH OF THE CATEGORIES OF BACK
c ALLOCATION,TOTAL,EQUIVALENT VI's AND THE 'A' VALUE.
c

      do 140 k=1,nzone
      z1=0.0
      z2=0.0
      cnt=0
      do 40 i=2,9
      l=i-1
      zt=-bval*mag(l)*alog(10.0)
      zt=exp(zt)
      z2=z2+zt
      z1=z1+(ar(k,i)*ar(k,i))/zt
c     z(k)=z1/z2
c     avalch=.5*alog(z(k))/alog(10.0)
      tz(k)=tz(k)+ar(k,i)
      tze(k)=tze(k)+(ar(k,i)*10**(bval*(mag(l)-mag(2))))
      if(ar(k,i).eq.0) go to 41
       val=ar(k,i)
      aval(k)=aval(k)+alog10(val)+(bval*mag(l))
      cnt=cnt+1
      go to 40
41    aval(k)=aval(k)
40     continue
     aval(k)=aval(k)/cnt
      z(k)=sqrt(z1/z2)
140    continue




c
```

```
c CALCULATE THE 'A' VALUE.
c

        do 95 k=1,nzone
        aval(k)=10**aval(k)
        do 96 j=1,levnum(k)
        if(ar(k,j+1).eq.0.0) go to 97
        val=ar(k,j+1)
        avalw(k)=alog10(val)+(bval*mag(j))
97      na(k)=na(k)+ar(k,j+1)*avalw(k)
96      continue
95      continue

        do 55 k=1,nzone
c
c CALCULATE THE TOTALS FOR EACH ZONES FOR EACH CATEGORY.
c

        chi2z=chi2z+z(k)
        totz=totz+tz(k)
        totze=totze+tze(k)
        totza=totza+aval(k)
        avalw(k)=na(k)/tz(k)
        antiwa(k)=10**avalw(k)
        totwa=totwa+antiwa(k)
55      continue

c
c CALCULATE THE PERCENTAGES OF DISTRIBUTION FOR EACH ZONE.
c

        do 60 k=1,nzone
        perz(k)=z(k)/chi2z
        pert(k)=tz(k)/totz
        if(tm(2).eq.0.0) go to 63
        per6(k)=float(ar(k,3))/float(tm(2))
63      continue
        pere(k)=tze(k)/totze
        pera(k)=aval(k)/totza
        perwa(k)=antiwa(k)/totwa
60      continue

c
c OUTPUT A SUMMARY OF THE INPUT AND THE PERCENTAGES
c CALCULATED FOR EACH OF THE ZONES FOR EACH CATEGORY.
c

        write(6,100)(mag(i),i=1,8)
100     format(/,1x,'ZONE  no.    ',8(f3.1,4x),'total',5x,' %T ',3x,
     *  ' %6 ',3x,' %E ',3x,' %A ',2x,' %wa ',2x,'%chi2',/,10x,'|')
        do 105 k=1,nzone
```

```
      write(6,110)nar(k),(ar(k,i),i=1,9),tz(k),pert(k),per6(k),pere(k),
     *pera(k),perwa(k),perz(k)
110    format(1x,a6,i3,'|',3x,8(i3,4x),i4,3x,6(2x,f5.3),/,10x,'|')
105    continue

       write(6,125)(tm(i),i=1,8)
125    format(5x,'total',4x,8(i3,4x))

c
c CALCULATE THE BACK ALLOCATION TABLE AND OUTPUT IT TO THE
c TERMINAL. DO THIS FOR EACH OF THE FOUR CATEGORIES
c

       do 65 k=1,nzone
       do 65 i=1,8
       table(k,i)=exv(i)*pert(k)
65     continue
       write(6,70)
70     format(//,3x,'Table I   Back allocated events based on sums.')
       write(6,71)(mag(i),i=1,8)
71     format(1x,'ZONE  no.    ',8(f3.1,4x),/,10x,'|')
       do 72 k=1,nzone
       write(6,73)nar(k),ar(k,1),(table(k,i),i=1,8)
73     format(1x,a6,i3,'|',8(1x,f7.5),/,10x,'|')
72     continue

       do 74 i=1,8
       do 74 k=1,nzone
       table(k,i)=exv(i)*per6(k)
74     continue
       write(6,75)
75     format(//,3x,'Table II   Back allocated events based on the',
     * ' distribution of VI''s.')
       write(6,71)(mag(i),i=1,8)
       do 76 k=1,nzone
       write(6,73)nar(k),ar(k,1),(table(k,i),i=1,8)
76     continue

       do 77 i=1,8
       do 77 k=1,nzone
       table(k,i)=exv(i)*pere(k)
77     continue
       write(6,78)
78     format(//,3x,'Table III   Back allocated events based on the ',
     * 'equivalent number of VI''s.')
       write(6,71)(mag(i),i=1,8)
       do 79 k=1,nzone
       write(6,73)nar(k),ar(k,1),(table(k,i),i=1,8)
79     continue

       do 80 i=1,8
```

```
      do 80 k=1,nzone
      table(k,i)=exv(i)*pera(k)
80    continue
      write(6,81)
81    format(//,3x,'Table IV   Back allocated events based on the ',
     * '''a'' value.')
      write(6,71)(mag(i),i=1,8)
      do 82 k=1,nzone
      write(6,73)nar(k),ar(k,1),(table(k,i),i=1,8)
82    continue

      do 90 i=1,8
      do 90 k=1,nzone
      table(k,i)=exv(i)*perwa(k)
90    continue
      write(6,91)
91    format(//,3x,'Table V  back allocated events based on ',
     * 'the weighted 'a' values.')
      write(6,71)(mag(i),i=1,8)
      do 92 k=1,nzone
      write(6,73)nar(k),ar(k,1),(table(k,i),i=1,8)
92    continue

      do 225 i=1,8
      do 230 k=1,nzone
      table(k,i)=exv(i)*perz(k)
230   continue
225   continue
      write(6,235)
235   format(//,3x,'TABLE VI back allocated events based on ',
     *' the chi-square values.')
      write(6,71)(mag(i),i=1,8)
      do 240 k=1,nzone
      write(6,73)nar(k),ar(k,1),(table(k,i),i=1,8)
240   continue

      stop
      end
```

Post Processing The Risk Data

Description

The program PTPLTANV.FOR (see Appendix B at the end of this section for program listing) is an interactive program that reads the binary data file produced by the risk program (SEISRISK II) and produces two output files. One file is a line-printer plot file and the other is an ASCII file of latitudes and longitudes with the calculated probablistic ground motion values.

The program allows the user to preview the output from the risk program (line-printer plot) before actually plotting the values (to scale) for contouring using EPMAP.FOR. (fig. F, see part F, connector E.)

Input File Description

The input file is a binary data file produced by the SEISRISK II program. This file contains a header line containing, (1) the starting and ending rows and columns selected from the original gridding of site points (see Bender and Perkins, 1982), (2) the total number of points along any fault(s) if any. Following the header are the gridded data points. They follow the form: latitude,longitude,three return period values of ground motion without variability, and three return period values of ground motion with variability.

Output File Description

There are two output files produced by this program. The file names are created by decoding the input file name and encoding it into two unique names. The first output file (with extention .lpp) is a line-printer plot of the gridded data by row and column number. For acceleration values, the values printed are right-justified, the first three significant digits including leading zeros, printed as blanks, with an implied decimal positioned to the left of the first digit. (See appendix A for a description of the output.) For velocity values, these numbers are whole number integers with an implied decimal to the right of the right most digit. The program will produce a grid plot for each of the three return periods, for "with" and "without" variability.

The second file (with extention .tcc) is an ASCII file containing latitudes,longitudes and from one to six values of ground motion. Think of the data in columns. The first two columns are the latitudes and longitudes, the next three values or columns are ground motion values without variability, the last three values or columns are ground motion values with variability. This file is the input file to the plot program EPMAP.FOR to produce a map of gridded data for contouring. (See appendix A at the end of this section for an example of the ASCII data file.)

Program Operation

This program is an interactive program that asks first for the type of data to be processed (either acceleration or velocity). Then a query is made for the input file name, of the form AAAAAAAA.dat. The first eight characters of the input file name will be used to create the first eight characters of the two output file names. The program finally asks for a two digit number that represents the starting and ending columns of data to be processed from the input binary data file. These columns are the calculated ground motion data values for each of the three requested return periods of the SEISRISK II input. Columns 1, 2, and 3 are the ground motion data values without attenuation variability and columns 4, 5, and 6 take into account attenuation variability.

The program reads the input file and writes the starting and ending row and column numbers, (grid size) and the number of sample points along any fault(s), if any at the users terminal. The program creates the two output files and then terminates the run.

Appendix A. Sample Output File

Acceleration values in this table are three digit decimal values. For ease of plotting and contouring, only significant digits are listed. For example the first line of data values listed as 33 65 152 are to be read as 0.033 G 0.065 G and 0.152 G, the numerical representation of the percent of gravity.

-123.900	47.750	33	65	152	40	85	184
-123.887	47.750	33	66	152	41	86	185
-123.873	47.750	34	66	151	41	87	185
-123.860	47.750	34	67	151	42	88	185
-123.846	47.751	34	68	151	42	89	186
-123.833	47.751	35	68	151	43	90	187
-123.819	47.751	35	69	151	43	91	187
-123.806	47.751	36	69	151	44	92	188
-123.792	47.751	36	70	150	44	93	188
-123.779	47.751	37	71	150	45	94	189
-123.765	47.751	37	71	150	45	95	190
-123.752	47.752	38	72	150	46	96	190
-123.738	47.752	38	73	151	47	97	191
-123.725	47.752	39	74	151	47	98	192
-123.711	47.752	39	75	151	48	99	194
-123.698	47.752	40	76	151	49	101	195
-123.684	47.752	40	77	151	49	103	196
-123.671	47.752	41	78	151	50	105	198
-123.657	47.753	42	80	152	51	106	199
-123.644	47.753	42	81	152	51	107	201
-123.630	47.753	43	82	153	52	109	203
-123.617	47.753	44	84	153	53	110	204
-123.604	47.753	44	85	154	53	111	205
-123.590	47.753	45	86	154	54	112	207
-123.577	47.753	46	87	155	55	113	208
-123.563	47.753	47	88	156	56	115	210
-123.550	47.754	48	90	156	57	116	212
-123.536	47.754	48	91	157	58	117	216
-123.523	47.754	49	92	159	59	118	220
-123.509	47.754	50	93	160	60	123	222
-123.496	47.754	51	94	161	61	126	224
-123.482	47.754	52	95	162	62	127	225
-123.469	47.754	53	97	163	63	129	228

Appendix B. Program Listing

```
c THIS PROGRAM IS A SLIGHT VARIATION TO THE PREVIOUS
c PTPLT PROGRAMS. TI IS DESIGNED TO READ THE BINARY DATA FILE
c FROM THE RISK PROGRAM AND PRODUCE A LINE PRINTER PLOT
c PLOT AND A TEKTRONIX PLOTABLE DATA SET.
c
c  INPUT:
c       THE 02FILE FROM THE FLAT EARTH PROGRAM.
c       AAAAAAAA.DAT
c  OUTPUT:
c       LINE PRINTER PLOT.
c       AAAAAAAA.lpp
c
c       TEKTRONIX PLOT FILE.
c       AAAAAAAA.tcc
c
c  VARIABLES:
c               XLONG(N,N) ARRAY OF LONGITUDES
c               YLAT(N,N) ARRAY OF LATITUDES
c               VAL(N,N,N) ARRAY OF REAL VALUES
c               IVAL(N,N,N) ARRAY OF INTEGER VALUES
c               NAME1 INPUT FILE NAME
c               OUT1 OUTPUT FILE NAME TEKTRONIX
c               OUT2-OUTPUT FILE NAME LINEPRINTER
c               N1-       DECODED PORSTION
c               N2-             OF THE
c               N3-           INPUT FILE
c               N4-               NAME
c               IROW1-STARTING ROW NUMBER
c               IROW2-ENDING ROW NUMBER
c               ICOL1-STARTING COLUMN NUMBER
c               ICOL2-ENDING COLUMN NUMBER
c               IROWN-TOTAL NUMBER OF ROWS
c               ICOLN-TOTAL NUMBER OF COLUMNS
c               DIGIT-TWO DIGIT NUMBER FOR COLUMN INDICATORS
c               ISTC-STARTING COLUMN NUMBER
c               LSTC-ENDING COLUMN NUMBER
c

      dimension xlong(450,50),ylat(450,50)
      dimension val(8,450,50),ival(8,450,50)
      dimension ic(50)
      dimension xlo(450),xla(450),pval(8,450),ipval(8,450)

      character name1*12,out1*12,out2*12
      character n1*3,n2*8,n3*4,n4*4,ftype*3
```

```
      write(6,200)
200   format(1x,'Enter acc or vel for the type of data.')
      read(5,201)ftype
201   format(a3)
       write(5,65)
65     format(1x,'Enter the input file name.')
       read(5,70) name1
      iflt=0
70     format(a19)
      write(5,105)
105   format(1x,'Enter a 2 digit number where the tens digit',
     * /,1x,'indicates the starting column and the units',
     * /,1x,'digit is the ending column of a total of 8',
     * /,1x,'possible columns for the line printer plot.')
      read(5,111)digit
111   format(f5.0)
      istc=digit/10.0
      lstc=digit-(istc*10)
      if(istc.eq.9.or.istc.eq.0) istc=1
      if(lstc.eq.9.or.lstc.eq.0) lstc=8

       decode (12,75,name1)n2,n3
75     format(a8,a4)

       encode(12,80,out1)n2
80     format(a8,'.tcc')
       encode(12,85,out2)n2
85     format(a8,'.lpp')

       open (unit=2,file=name1,status='old',form='unformatted')
      open (unit=19,file=out1,status='new')
      open (unit=18,file=out2,status='new')

c READ IN THE DATA FROM THE 02FILE
c  FIRST THE STARTING AND ENDING ROW AND COLUMN NUMBERS
c  THEN THE LONGITUDE LATITUDE AND THE 3 DIFFERENT RETURN
c  PERIOD VALUES.

       read(2) irow1,irow2,icol1,icol2,iflt
      write(5,*)irow1,irow2,icol1,icol2,iflt

      if(irow1.eq.0) go to 12

       irown=irow2-irow1+1
       icoln=icol2-icol1+1

c   NOW FIND OUT HOW MANY BLOCKS OF 30 COLUMNS THERE ARE.

       rloop=float(icoln)/30.0
       do 5 i=1,6
       if(rloop.le.float(i)) go to 6
```

```
5       continue
6       iloop=i

c  NOW WE START READING THE DATA AND PUTTING IT INTO
c  AN ARRAY WITH THE PROPER ROWS AND COLUMNS.

        do 10 i=1,irown
        do 20 j=1,icoln
        read(2)xlong(j,i),ylat(j,i),(val(k,j,i),k=1,6)
       do 35 k=1,6
       if(ftype.eq.'acc') val(k,j,i)=val(k,j,i)*1000
      ival(k,j,i)=val(k,j,i)
35     continue
30      continue
20      continue
10      continue
c read in values on a fault
c if the fault exists
c
12     continue
       if(iflt.eq.0) go to 36
       do 15 i=1,iflt
       read(2)xlo(i),xla(i),(pval(k,i),k=1,6)
       do 16 k=1,6
       if(ftype.eq.'acc') pval(k,i)=pval(k,i)*1000
       ipval(k,i)=pval(k,i)
16     continue
15     continue

36      close (unit=2)

c  WRITE OUT THE TEKTRONIX PLOT FILE.

        do 40 i=1,irown
        do 50 j=1,icoln
        xlong(j,i)=-xlong(j,i)
        write(19,55)ylat(j,i),xlong(j,i),(ival(k,j,i),k=istc,lstc)
55      format(2f10.3,8i4)
50      continue
40      continue
c write out the values on the fault
c if the fault exists
       if(iflt.eq.0) go to 44
       do 43 i=1,iflt
       xlo(i)=-xlo(i)
       write(19,55)xlo(i),xla(i),(ipval(k,i),k=istc,lstc)
43     continue
44     continue

        close (unit=19)
```

```
c  NOW FILL AN ARRAY WITH THE COLUMN NUMBERS

       icl=icol1
       do 60 i=1,icoln
       ic(i)=icl
       icl=icl+1
60     continue

c  NOW COMES THE FUN OF SETTING UP THE LINEPRINTER PLOT.
c  THE ONLY RESTRICTION PLACED ON THE OUTPUT IS THE NUMBER OF
c  OF COLUMNS. I DONT CARE HOW MANY ROWS THERE ARE BUT THE
c  MAXIMUM NUMBER OF COLUMNS WHICH CAN BE PRINTED ACCROSS
c  THE PAGE IS 30.
c

c  THE IDEA HERE IS TO PRINT THE GROUND MOTIONS FOR THE FIRST
c  RETURN PERIOD IN THEIR ENTIRETY, THE SECOND,THEN THE THIRD.
c  WITHIN EACH RETURN PERIOD THE WIDTH MAY GO BEYOND 30
c  COLUMNS. IN THOSE CASSES THE INFORMATION IS PRINTED
c  IN BLOCKS OF 30 COLUMNS WIDE AND 1 THROUGH N ROWS LONG.
c  THEN THE NEXT 30 OR LESS COLUMNS TO THE RIGHT OF THE
c  AND ALL ROWS 1 THROUGH N ARE PRINTED BELOW THE FIRST GROUP.
c  SO, TO CONTOUR THE LINEPRINTER PLOT, SOME CUTTING
c  AND PASTING MAY BE REQUIRED.
c

       do 90 k=istc,lstc

c  INITIALIZE THE COLUMN SETTINGS.

       ist=1
       lst=30
       icn=icoln
      if(icn.lt.30)lst=icoln

       do 100 ig=1,iloop

c  WRITE OUT THE COLUMN HEADINGS.

       write(18,109)
109    format('1')
       write(18,110)(ic(in),in=ist,lst)
110    format(9x,30i4)

c   NOW SET UP THE LOOP FOR DOING THE ROWS.

       ir=irow1
       do 120 i=1,irown
       write(18,125)ir,(ival(k,j,i),j=ist,lst)
125    format(/,4x,i4,2x,30i4)
       ir=ir+1
```

```
120     continue

c  NOE WE NEED TO DO SOME BOOKKEEPING TO INSURE
c  WE GET THE PROPER GROUPS IN THE PROPER ORDER BY COLUMNS.

        nextg=30
        icn=icn-30
        if(icn.lt.30)nextg=icn
c  THE ABOVE IF IS TO GET THE LAST FEW COLUMNS.

c  STARTING AND ENDING COLUMNS MUST BE SHIFTED TO THE RIGHT
c  BY 30 COLUMNS.

        ist=ist+30
        lst=lst+nextg

c  NOW RETURN TO THE TOP ROW AND THE NEXT SET OF 30 COLUMNS
c  TO THE RIGHT.

100     continue

c  NOW RETURN TO THE NEXT RETURN PERIOD GROUP AND REINITIALIZE
c  ALL THE ROW AND COLUMN PARAMETERS.

90      continue
        close (unit=18)
        stop
        end
```

References

Algermissen, S. T., Perkins, D. M., Isherwood, W., Gorden, D., Reagor, G., and Howard, C., 1976, Seismic risk evaluation of the Balkean region: Proceedings of the seminar on seismic zoning maps, Skopje, Yugoslavia, October 27-November 4, 1975, v. 2, p. 172-240.

Algermissen, S. T., Perkins, D. M., Thenhaus, P. C., Hanson, S. L., and Bender, B. L., 1982 Probabilistic estimates of maximum acceleration and velocity in rock in the contiguous United States: U.S. Geological Survey Open-File Report 82-1033 107 p.,6 pls.

Bender, B. L., and Perkins, D. M.,1982, SEISRISK II; computer program for seismic hazard estimation: U.S. Geological Survey Open-File Report 82-293, 104 p.

Stepp, J. C., 1973, Analysis of completeness of the earthquake sample in the Puget Sound area, in Harding, S. T., ed., Contributions to seismic zonings: National Oceanic and Atmospheric Administration Technical Report ERL267-ESL30, p. 16-28.

Stover, C. W., and Dillinger, W. H.,1971, A computer system for the display of geographic data, National Oceanic and Atmospheric Administration Technical Memorandum ERL ESL-13, 38 p.

Thenhaus, P. C., ed., 1983, Summary of workshops concerning regional seismic source zones of parts of the conterminous United States, convened by the U.S. Geological Survey 1979-1980, Golden, Colorado: U.S. Geological Survey Circular 898, 36 p.