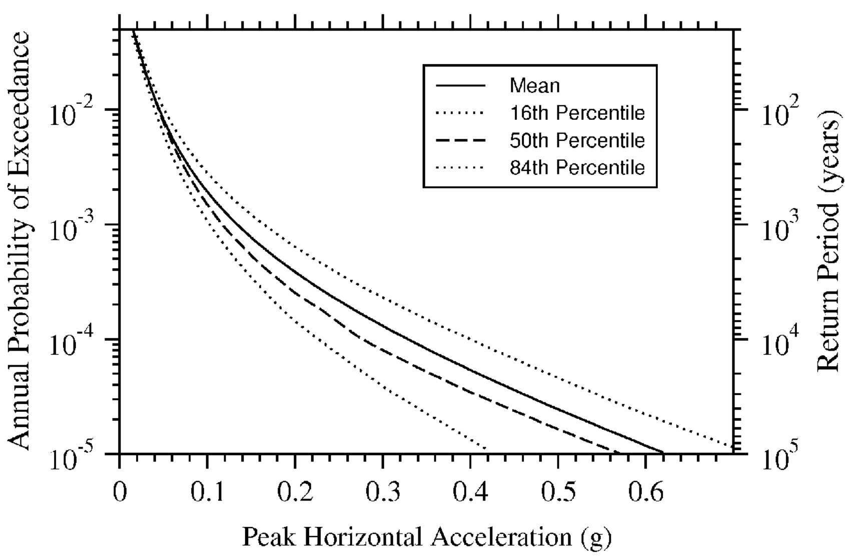
RECLAMATION

Managing Water in the West

Report DSO-07-01

Manual for Reclamation Fault-Based Probabilistic Seismic Hazard Analysis Software



Dam Safety Technology Development Program

U.S. Department of the Interior
Bureau of Reclamation
Technical Service Center
Denver, Colorado

April 2007

REPORT DOCUMENTATION PAGE						*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
04-2007	Technical	

4. TITLE AND SUBTITLE	
Manual for Reclamation Fault-Based Probabilistic Seismic Hazard Analysis Software	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Roland LaForge	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Bureau of Reclamation Technical Service Center Seismotectonics and Geophysics Group Denver Colorado	**8. PERFORMING ORGANIZATION REPORT NUMBER** DSO-07-01
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Bureau of Reclamation Denver, Colorado	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) DSO-07-01

12. DISTRIBUTION/AVAILABILITY STATEMENT
National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Probabilistic Seismic Hazard Analysis (PSHA) is the preferred technique for translating seismic hazard into ground shaking characterizations that engineers can use for testing structures and incorporating into risk assessments. The results are presented in terms of the annual probability of exceedance (APE) of a ground motion parameter such as peak horizontal acceleration, damped peak spectral acceleration at some oscillator frequency, or peak horizontal velocity. A uniform hazard spectrum for a specified annual probability of exceedance or return period is often used to estimate the probability of damage determined from a dynamic analysis of the structure. Software developed at Reclamation for performing PSHA is described in this manual. The main categories are programs for generating APE data for fault and areal seismic sources, postprocessing these results in a logic tree format, and programs for other, more general seismic hazard analysis.

15. SUBJECT TERMS
probabilistic seismic hazard analysis, PSHA, risk assessment, ground motion attenuation function, annual frequency, Earthquake Magnitude Rate Distribution Characterizations, fault, fault characterizations, logic tree, seismic sources, spectral response period, hazard curve

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**a. THIS PAGE**	SAR	59	**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U			

BUREAU OF RECLAMATION
Technical Service Center, Denver, Colorado
Seismotectonics and Geophysics Group, 86-68330

Report DSO-07-01

Manual for Reclamation Fault-Based Probabilistic Seismic Hazard Analysis Software

Dam Safety Technology Development Program
Denver, Colorado

Prepared: Jerry Wright (for Roland LaForge)
Manager, Seismotectonics and Geophysics Group, 86-68330

Peer Review: Jerry Wright (for John Ake)
Manager, Seismotectonics and Geophysics Group, 86-68330

11-9-07
Date

REVISIONS					
Date	Description	Prepared	Checked	Technical Approval	Peer Review

Mission Statements

The mission of the Department of the Interior is to protect and provide access to our Nation's natural and cultural heritage and honor our trust responsibilities to Indian Tribes and our commitments to island communities.

The mission of the Bureau of Reclamation is to manage, develop, and protect water and related resources in an environmentally and economically sound manner in the interest of the American public.

Acknowledgments

The Dam Safety Technology Development Program of the Bureau of Reclamation sponsored this research.

Contents

Page

Figures

No. Page

faultsource Technical Guide

faultsource is the generic name of the program, developed by Reclamation, that generates hazard curve data for probabilisitic seismic hazard analysis (PSHA). The author began development of the program in the late 1990s with Chris Wood of the Seismotectonics and Geophysics Group contributing later additions. In 2005, software development was placed under the version control system, *subversion*; since that time, reports using faultsource include the program version number. The current version of the code is called *faultsource_20*. When using the code's results, it is highly advisable to mention the version number used.

faultsource_20 is one of the codes that has participated in the PEER (Pacific Earthquake Engineering Research Center) Lifelines "Working Group for Validation of Probabilistic Seismic Hazard Computer Programs" project (see http://peer.berkeley.edu/year7/yr7_projects/ta3/607.html). Although that project has not been completed, *faultsource_20* results have been shown to be consistent with those of other codes currently in use.

This guide describes the basic PSHA equation, the technical details of how *faultsource_20* models earthquake ruptures, and the computations involved in generating mean and fractile hazard curves.

PSHA Theory

Cornell (1968) first described the general theory and computational procedure of PSHA. PSHA provided an alternative to the deterministic "maximum credible earthquake" approach, which was to list all nearby seismic sources and associated maximum magnitudes and use the ground motion from the distance-magnitude combination that gave the highest amplitude. The main advantage of PSHA is that it introduces the time element; that is, it allows one to calculate *how often* ground motions of all levels are exceeded. This provides a natural connection to risk assessment studies, where safety standards are tied directly to annual probabilities of structural damage and/or failure. Another advantage is that it can account for uncertainties in the various measurements and estimates that go into the calculations. By integrating formal parameter distributions, and/or estimated weights of various scenarios, mean and fractile results can be generated in a general logic tree format. While these results are valuable in themselves, the logic tree can be manipulated to determine the sensitivity of the results to the input parameter distributions. This in turn can provide guidance in deciding what research or data collection activities would be most effective in reducing the overall uncertainty of the PSHA results.

The PSHA formula is a variation of the Total Probability Theorem:

$$P(v \geq V) = \sum_{S} \int_{m} \int_{r} P(v \geq V \mid (m,r)) P(m) P(r) dr dm \tag{1}$$

where,

- $P(v \geq V)$ is the probability that ground motion v exceeds some value V. S is all sources
- $P(v \geq V \mid (m,r))$ is the probability that ground motion v exceeds value V given a magnitude and distance
- $P(m)$ is the probability of a given magnitude
- $P(r)$ is the probability of a given distance

The left side of the equation is given in exceedances per year. The equation states that the probability of the ground motion exceeding some value is equal to the sum of the probabilities that all possible magnitudes occurring at all possible locations will exceed that value for a given source zone. The total probability for a given level is the sum of the annual frequency contributions from all sources.

Ground Motion Distribution

$P(v \geq V \mid (m,r))$ is provided by a ground motion attenuation function. In regions where strong motion records are abundant, such as western California, these can be derived empirically. However, where data are sparse and wave propagation effects not well determined, such as in eastern North America, ground motions are often generated synthetically and compared to the few existing records. Because ground motions depend on specific source, path, and site conditions, they can vary dramatically for a given magnitude-distance pair. Attempts are made to isolate the effects of parameters such as “style of faulting” (usually strike-slip, normal, and reverse), site conditions (“rock” or “soil,” average shear wave velocity in the top 30 m, or depth to bedrock), and rupture directivity effects for near-source motions. Separate attenuation functions have also been derived for subduction zone events.

The ground motion for a magnitude-distance pair is fit to a lognormal distribution, described by a median and standard deviation. Since physical limitations must prevent the upper tail of the distribution from going to infinity, an upper bound must be imposed. The truncation point is an area of controversy, since the relatively small number of strong motion records doesn’t allow for its reliable estimation. Because very low annual frequencies of exceedance sample more of the upper tail area, this becomes an important issue where very long return periods are required, such as for critical structures. Another problem arises when ground motions for long return periods are calculated for a high-activity rate fault: the ground motion associated with the long return period may lie beyond the truncation point. In this case, the truncation point must be increased; but then possible physical ground motion limits are ignored.

The truncation point is set in *faultsource_20.h*, as *GM_ZMAX*, the number of standard deviations above the median. The default value is 3.0, which encompasses 99.87% of the total probability. *faultsource_20* discretizes the ground motion distribution for each magnitude-distance pair produced by the geometric model into 0.01-g intervals from 0.01 g to the upper limit defined by *GM_ZMAX*. This method produces smooth hazard curve results out to the largest ground motions.

Magnitude Rate Distribution

P(m) is the magnitude distribution for the fault. Three models are currently available in *faultsource_20*: the maximum moment, exponential, and characteristic. The maximum moment model assumes that a fault ruptures most or all of its potential rupture area, producing magnitudes in accordance with empirical observations (e.g., Wells and Coppersmith, 1994). This can be cast as a single magnitude, or a distribution. Available distributions are Gaussian, uniform, symmetric triangle, and discrete. The maximum moment distribution is described by Wesnousky (1986). Activity rates for the maximum moment model can be based on slip rate, annual rate, or return period.

The exponential distribution assumes that magnitudes are distributed exponentially according to the Gutenberg-Richter formula, Log(n) = *a* - *b*(M), where Log(n) is the number of events above magnitude M, and *a* and *b* are constants. The distribution is truncated at lower and upper magnitude bounds.

The characteristic model (Youngs and Coppersmith, 1985) is a hybrid between the maximum moment and exponential models. The recurrence curve consists of an exponential portion for the smaller magnitudes, and a boxcar-shaped portion for the "characteristic" magnitude (actually a range of magnitudes). The "characteristic" magnitudes occur at a higher rate than an extrapolation of the exponential part would imply, in accordance with observations of paleoseismic data compared to recent seismicity data in California and Utah (Schwartz and Coppersmith, 1984). Parameters for the exponential and characteristic models are the same, consisting of a lower magnitude bound, an upper or "characteristic" magnitude, and the *b* value in the Gutenberg-Richter equation.

Both the exponential and characteristic models use slip rate as the rate parameter, and distribute magnitude rates through the moment rate-fault area-magnitude rate conversion (Anderson, 1979; Hanks and Kanamori, 1979). Therefore, care must be taken when applying a slip rate to these models. If the the applied slip rate is based on surface fault offset observations, the offsets, to be observable at the surface, are most likely produced by the larger magnitudes that the fault is capable of producing. In this case, the maximum moment model is most appropriate. On the other hand, if the slip rate is a regional rate based on geodetic observations, the regional slip (apportioned onto the fault) can produce a large range of magnitudes, and the characteristic and/or exponential models can be used.

Since the characteristic and exponential models "predict" the activity rates of smaller events, if the seismicity rates are high enough (such as in plate boundary environments) the historic record can be examined to see if such events have actually occurred. This assumes that earthquakes are occurring in a Poissonian (time-independent) manner.

Distance Distribution

The distance distribution is determined by the distances of individual ruptures to the site. Attenuation functions define the distance parameter in different ways. Some (e.g., Sadigh et al., 1997; Abramson and Silva, 1997) define the distance parameter as the nearest distance from the site to the rupture area on the fault. Some, notably Boore, Joyner, and Fumal (1997), and Spudich et al. (1999), use the horizontal distance to the map projection of the fault plane, and some for subduction zones (e.g., Youngs et al., 1997) use both horizontal distance and depth to the closest point on the rupture plane. *faultsource_20* accounts for these differences and uses the appropriate distance measure for the attenuation function used. The next section describes details of how the distances are measured.

The Geometric Fault Model

faultsource_20 creates a 3-D model of a fault, generates individual rupture patches on the fault, and computes the necessary distance measures to the site. Because of the probabilistic nature of the analysis, many rupture patches are created which account for the input distributions on activity rate and magnitude distributions. Each discrete realization carries a weight with it, and ultimately the aggregation of realizations are integrated to obtain the posterior ground motion distribution.

A fault is parameterized by a set of surface points, a dip (or dip distribution), and minimum and maximum depths. The edges of the fault are perpendicular to their corresponding surface traces. Figure 1 shows some examples. In the figure, the updip trace is shown as the bold line. In most cases, this will be the surface trace, but blind thrusts can be modeled with a nonzero minimum depth.

Dip Distribution

A dip distribution can be invoked. Practically speaking, this should be used only for faults within about 10 kilometers of the site; at greater distances, the effect on the results is negligible. Even at close distances, the effect can be negligible, due to the tradeoff between dip and fault area. Available dip distributions are Gaussian, triangular, uniform, and discrete (see *Fault Dip Distributions*, p. 37). If a dip distribution is used, *faultsource_20*, by convention, fixes the maximum depth of the fault and varies the fault area. Since the magnitude distribution is independently set, care must be taken that the variations in fault area due to varying dip do not grossly disagree with the magnitude distribution. An option is

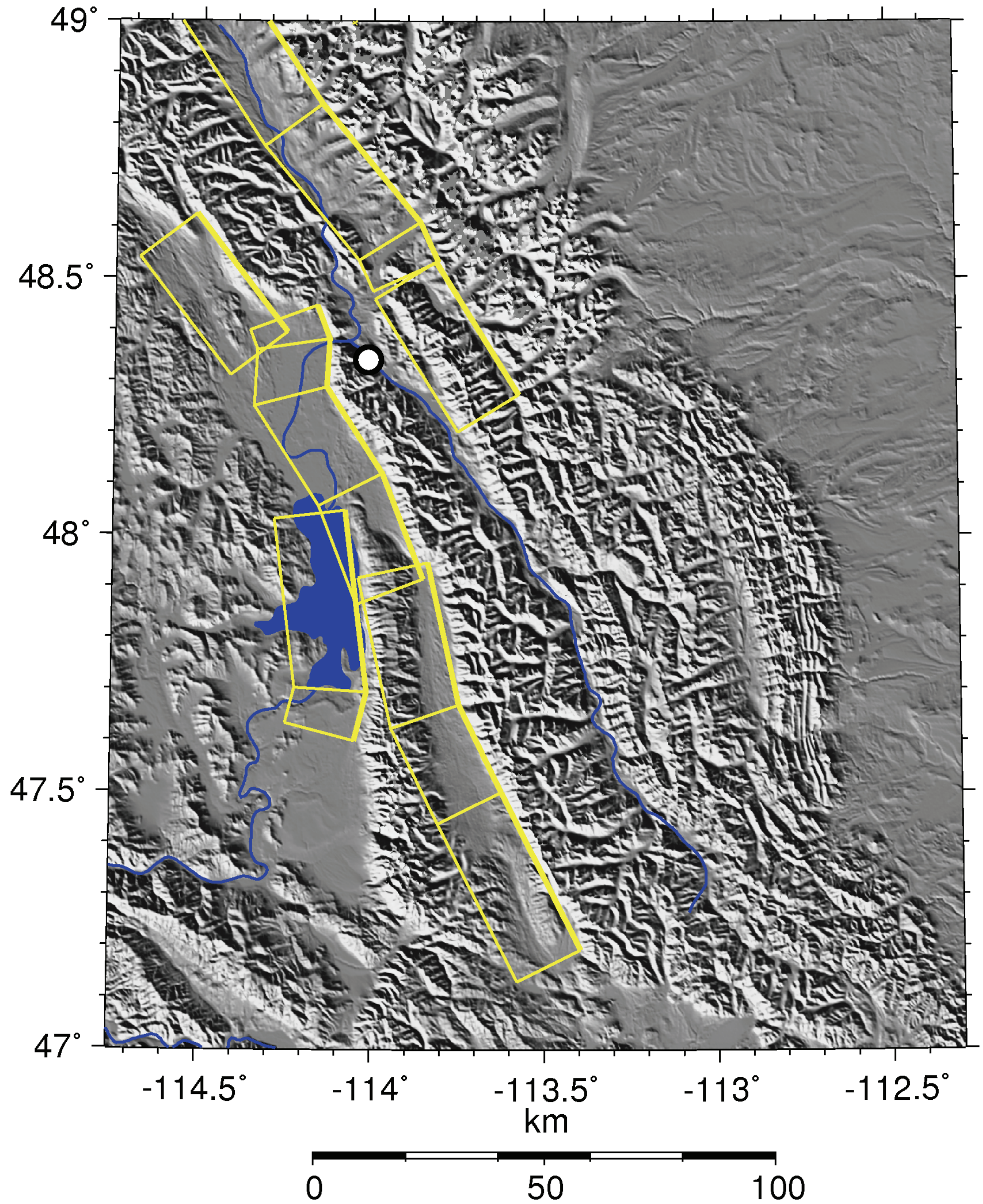


Figure 1.—Example fault characterizations. Bold lines are updip traces; thinner lines show downdip boundaries.

planned that will allow the depth to vary with dip while keeping the fault area constant. The current scheme assumes that knowledge of the maximum seismogenic depth is better known than the rupture area as a function of magnitude; the other option would assume the reverse.

Use of *fault_params*

The geometric scheme uses fault edges perpendicular to the first and last surface segments. Therefore, in a single-segment fault (two coordinates specified), the plane consists simply of a rectangular plane. In a two-segment fault (three coordinates specified), the fault consists of two rectangular sections (on the ends) and two triangular sections in the middle. For faults with more than four coordinates, the interior segments consist of a rectangular plus two triangular sections, and the end segments consist of a rectangular plus one triangular section. This scheme was agreed upon during the PEER Working Group sessions as the most appropriate.

This characterization is subject to some geometric limitations in its application. Looking at figure 1, a fault characterization can be described by updip lines, downdip lines (parallel to the updip lines), and segment boundaries. If the azimuthal angle between two adjacent updip lines is too great, this can result in downdip lines that intersect each other or segment boundary lines. *faultsource_20* will give unpredictable results or error messages, or crash under these circumstances. Therefore, a program, *fault_params* (section 4.3.6), was developed to check for these conditions. It is invoked by typing *fault_params* < *fault_coordinate_file*. The standard output will tell whether any lines have crossed or intersected.

fault_params gives other useful information as well. *faultsource_20* will not allow two coordinates to be less than 2 kilometers apart (this can happen due to digitization errors); *fault_params* will check for that. If any of the above conditions exist, the fault coordinates have to be modified and retested.

fault_params also:

- Computes the fault area, surface length, and fault width and calculates magnitude based on these parameters

- Computes the aspect ratio (length/width) and gives a warning if it is less than 1

- Computes the nearest, farthest, and mean distance to the fault plane

- Computes the nearest and farthest horizontal distance to the fault

- Computes the latitude, longitude, and depth of the closest point on the fault surface to the site

- Computes the coordinates of the updip lines, downdip lines, and segment boundaries (in file *map_coords.txt*). These are the lines plotted in figure 1.

fault_params also produces a tab-delimited file (*faultparam_data.txt*) that contains the geometric information for each fault. This file can be imported into Word or FrameMaker as a table. It is advisable to create this table and review it with the geologist before proceeding with the PSHA. It is also advisable to make a map such as shown in figure 1 to make sure the fault characterization is correct.

There are other general considerations when characterizing faults: (1) PSHA is based on computing accurate distances to the fault plane; it is therefore important to model faults or parts of faults near the site as accurately as possible; (2) that said, it is not necessary to match every wiggle of a fault trace, especially for more distant faults. It is best to keep the characterization as simple as possible.

Intersecting Faults

In certain complicated groupings of faults, the fault characterizations can lead to faults that intersect at depth. This situation is physically unrealistic, and difficult to diagnose by plotting the fault outlines as in figure 1. The program *fault_intersects* reads a fault coordinate file, and checks the first fault in the list with subsequent ones for intersecting planes. Diagnostic information is produced, and files containing the fault outlines and the lines (actually points) of intersection. These fault outlines and intersection points can then be plotted and analyzed. Type *fault_intersects -q* for usage.

Generating Fault Ruptures and Calculating Distances

The purpose of the geometric fault model is to obtain distances from rupture surfaces. If the fault area is equal to or less than that obtained from the fault area-magnitude relation, the entire fault area is simply used as the rupture area. However, if the rupture area is less than the fault area, the rupture area must be distributed on the fault surface.

A number of schemes can be imagined for accomplishing this task. One commonly used scheme is to start in one corner of the fault plane and "march" the rupture patch along strike, and then updip, at some fixed interval (a version of faultsource under development uses this scheme). Another would be to randomly distribute the centers of the patch on the fault plane.

The scheme used by faultsource creates a block of rupture patches that fit the maximum number of integral times within the fault plane (see fig. 2). Because there is usually space left over on the fault plane, the block of patches must be shifted at least once along strike and updip to ensure that the patches cover the entire fault plane. In the input file (see *Example 2—faultsource_20 input file—maximum moment model*, p. 21), the number of shifts is controlled by the "Strike Direction Offsets=" and "Dip Direction Offsets=" lines. These are nominally set to 2, meaning one shift in each direction. This has been found to give stable results in the great majority of cases. For the second "shift" in the along-strike direction, the right side of the block of ruptures would coincide with the right side of the fault in Figure 2; the third would be centered in the center of the along-strike dimension. The same holds true for the along-dip shifts. Each rupture patch realization is given equal weight in the final integration.

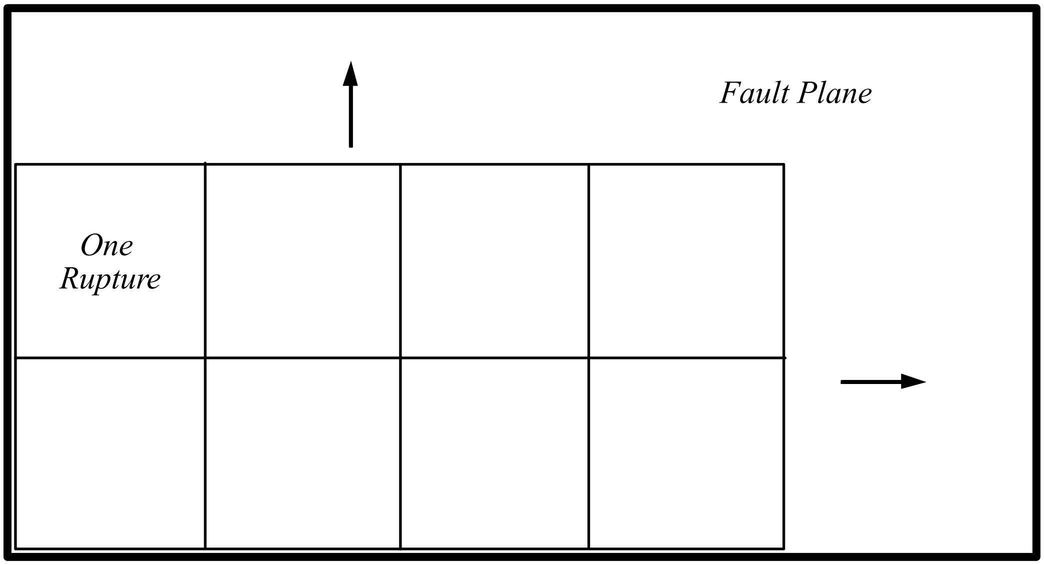


Figure 2.—Scheme for distributing ruptures on the fault plane used by *faultsource_20*.

The aspect ratio of the rupture patches (length/width) is nominally set to 1. This can be changed in *faultsource_20.c* where the variable aspect_ratio[0] is set. The option to discretely set and weight rupture aspect ratios exists in the program, but has never been exercised.

Distances from the patch to the site are obtained by filling the patch with uniformly spaced points and finding the point that yields the minimum distance. Nearest horizontal distance and depth of nearest point are also computed, which are needed by some attenuation functions. The grid spacing is nominally set to 1.0 kilometer as the DGRID variable in *faultsource_20.h*, to be changed at the user's discretion.

Calculation of the Hazard Curve

As shown in equation 1, the final result consists of an integration over all magnitudes, distances, and sources. *faultsource_20* is also designed to account for variations in maximum magnitude, location of rupture patch on the fault plane, fault dip, ground motion as a function of magnitude and distance, and activity rate. However, the program can deal with only one attenuation function and recurrence model at at time. These variations (and others, such as alternate faulting scenarios) must therefore be handled in a logic tree format. The variations that *faultsource_20* does account for internally result in many "realizations" of the hazard, each with its own weight. For each ground motion level (at 0.01-g intervals), the annual frequency results are sorted and integrated to obtain the mean and fractile results. The results for each fault are summed, giving the cumulative curves.

It is not well documented in the literature how uncertainties from PSHAs should be treated. The implied consensus, however, appears to be that posterior distributions should consist of variations of the *means* of all possible realizations. For example, if mean results from many different attenuation functions are obtained, the overall mean result would be the weighted mean of this collection of means. The *PSHA Practical Guide*, page 10, describes how to obtain these results.

Program Control

A number of controls and limitations are provided for in the C header file *faultsource_20.h.* Limits on such parameters as the total number of faults, number of geographic points per fault, maximum number of probability distribution discretization points, and rupture patch grid density are set here. The program stops and issues a diagnostic if any of these limits are exceeded. If any of the limits in the header file are changed, the program must be recompiled before use.

Some command-line arguments are also recognized; type *faultsource_20 -q* to see these. The **-P** and **-N** options have not been implemented. The **-R** option computes the rate information (contained in file *Magrates.txt*) and stops. The **-C** option allows one to modify the attenuation function-derived ground motions by applying a file with correction factors for each spectral response period. The number of periods in the file must match the number of periods used by the attenuation function in the input file. The **-l** option gives a list of attenuation codes and their references. The **-a** and **-b** options allow one to limit the magnitude range of the analysis.

The **-d** option sets a ground motion level for deaggregation purposes. In this case, the program only creates a file with the deaggregation data (deaggregation is discussed in LaForge, 2001; the same deaggregation programs can be used for *faultsource_20*-generated data). The file containing the deaggregation data is *fs2.0.deagg*. The **-t** option causes the deaggregation information to be written in text rather than binary format.

The **-m** option overrides the magnitude information in the input file, and instead calculates fault area directly from the specified fault area-magnitude formula. The **-v sitevelocity** option is required by the jb97 and jb97x attenuation codes; this is the shear wave velocity in the top 30 meters at the site.

-X produces hazard curve files that are tab delimited; **-T** produces space-delimited files. **-q** is also equivalent to **-h**, **-H**, and **-?**.

PSHA Practical Guide

This guide shows how to conduct a PSHA using *faultsource_20*, the multiple random sources (*mrs*) programs, and a number of auxiliary programs used for logic tree analysis. *faultsource_20* computes the probabilistic ground motion hazard from fault sources, while the *mrs* series computes the hazard from areal zones of randomly occurring "background" seismicity. The *faultsource Technical Guide*, page 1, describes the theoretical and implementation aspects of *faultsource_20*, and LaForge (2001) described the theory and operation of the *mrs* programs.

The PSHA procedure consists of two main steps: generating the hazard curves using *faultsource_20* and/or the *mrs* programs, then combining, integrating, and weighting the results in a logic tree procedure. This description utilizes a Unix/Linux operating system using *bash* shell scripts; adequate results have also been obtained using Excel in the Windows environment.

As discussed in *faultsource Technical Guide*, integration of the hazard curves takes a "mean and fractile of means" approach.

Generate Hazard Curves

Steps 1 and 2 assume you are working with a geologist who has access to the Seismotectonics and Geophysics Group's ARC-GIS suite of mapping software. If these tools are not available, the input files will have to be generated with a text editor.

STEP 1: Create and Verify Input Files

1. Get ARC-Map with PSHA faults from geologist (usually put in shared drive).

2. Run the BOR-TOOLS program to get the *faultsource_20* input file and the fault coordinate file. The input file contains slip rate and magnitude information for each fault. The fault coordinate file contains fault geometry information. *faultsource_20* requires these two files.

3. Check the fault coordinate file visually for obvious errors. Then run *fault_params* on the fault coordinate file. The standard output will give (1) geometric information regarding the fault, such as surface length and area; (2) magnitudes from surface length and area, based on commonly used empirical relationships; (3) distance measures to the site; (4) aspect ratio (length/width) of the fault; if significantly less than 1, in addition to not being physically realistic, the aspect ratio can cause problems for *faultsource_20*; (5) whether any of the lines joining the fault segments intersect; this can happen if fault segments take radical changes in azimuth;

if intersections occur, the fault configuration must be changed; (6) whether two successive coordinates are less than 2 kilometers apart; this can happen due to digitization errors. *faultsource_20* will not accept points less than 2 kilometers apart.

4. In addition to checking for errors and geometric viability, *fault_params* also generates *map_coords.txt*, which is used to produce a three-dimensional representation of the faults in GMT or ARC, and *faultparam_data.txt*, which is a tab-delimited file that can be imported as a table into FrameMaker or MS Word. Make this table, as it provides an easy-to-read record of the fault geometry parameters that can be reviewed for accuracy and included in the final report.

5. Make a map in either GMT or ARC of the three-dimensional fault representations, with the updip trace marked in some way, for example by a thicker line than the downdip segment boundaries. Compare this map to the one the geologist made and see if they are consistent, and check for correct dip direction.

6. Give the map, tables, and *fault_params* output to the geologist, and have him/her check it over to make sure the results are what was intended. In the *fault_params* table, have the geologist write the maximum magnitude. Often, geologic considerations for this parameter supersede the empirical relations.

STEP 2: Create Directory Structure and Runscript

1. Create a directory structure for the project. A top-level directory with the name *projectname_month-year* works best. Below this level, make the directories infiles/, runpsha/, psha/, gmt/ (if you are making gmt maps), figures/, and others such as random/ and/or recur/ if you are doing random seismicity analysis. The goal should be for all files related to a project to be contained in one directory.

2. All input files should be placed into infiles/, and analyses done on them (such as described in STEP 1 above) should be performed here.

   The next step is to break an input file up into one file for each fault or fault scenario. This makes it easier during the logic tree process to combine and weight different fault combinations and scenarios. The fault-specific input file is modified further. Initially, the file typically contains a slip rate distribution. For the purposes of eventually computing mean and fractile hazard curves, you need to discretize the distribution and assign weights to each discrete result. Because the hazard curve scales linearly with slip rate, you only need to generate a hazard curve for the initial discrete slip rate, and then multiply that curve by the appropriate factor for the distribution.

The appropriate scale factor and corresponding weight is obtained by executing *faultsource_20* with the *-R* option. The factors and weights are contained in the diagnostic file *Magrates.txt*. The factors are in a line enclosed by parentheses; these are directly pasted into the runscript. The weights are also listed, and should be saved for later use in the logic tree script. While most of the distributions are discretized into equal-probability units, some are not. Nine discretization points usually provide enough resolution when integrated to produce smooth fractile curves. It is important that all slip rate distributions are discretized into the same number of points.

This procedure can be automated with the program *makefsfiles* (*makefsfiles—generate a runscript from a* faultsource_20 *Input File*, p. 52). This program generates a template runscript, which then must be modified to include the desired attenuation functions and correct results file. The execution portion of the script is generalized, and should be checked over and modified as necessary.

3. The runscript is a bash script that runs *faultsource_20* on the input files, generates separate hazard curves for each discrete rate distribution point, and deposits them into the psha subdirectory. See *Example 5—faultsource_20 results file*, page 24. The script contains a "fault lineup," which is a translation table between the fault sequence number and the fault name, variables for the number of faults, the recurrence models to be run, attenuation functions to be used. The next part of the script consists of a number of functions that run *faultsource_20*, looping through the different recurrence models and attenuation functions.

   An example runscript is shown in *Example 5—faultsource_20 results file*, page 24. A good reference for understanding bash syntax is Newham and Rosenblatt (1998).

STEP 3: Execute Runscript, Generate Hazard Curves

The runscript is executed by typing **./runscript.sh**. If *faultsource_20* fails, some useful diagnostics are produced. Hazard curve files are put into the psha directory. It's a good idea to check some of the files to make sure they look correct and actually contain data. The file names have the form *hazdat.fault#.$recurtype.r#.attstr*, where # is the fault sequence number, *recurtype* is the recurrence model code, *r#* is the rate discretization number, and *attstr* is the attenuation function code.

The script generally works by the "Fault linuep" section first translating between the fault name and its assigned number. It is in the form of comments and is for informational use only. This is important for debugging and keeping track of fault names when the logic tree is executed. The rate multiplication factors are listed, then the list of attenuation functions to be used, and the list of recurrence models to be run.

Next is a set of functions that perform certain tasks. *switch_attfunc* changes the attenuation function code to the first on the list for the first iteration, and switches it to the next one for subsequent iterations. *revert_to_original_attfunc* switches the code back to the first one after the last iteration. *run_faultsource_and_save_output* executes *faultsource_20*, renames the output file according to the scheme described above, and moves it to the output directory. Note that if the jb97 or jb97x attenuation function is used, a shear wave velocity in m/s is required on the *faultsource_20* command line, and so must be run as a separate command. The appropriate shear wave velocity should be entered as the *-v* option in *run_faultsource_with_jb97_and_save_output*. If *faultsource_20* fails for any reason, the script stops, displaying the name of the input file that caused the failure.

The fault coordinate and results configuration files are then copied from the *infiles* directory. The following loop structure executes the program, iterating over all faults, recurrence models, and attenuation functions, scaling the hazard curves by the rate factors, and moving the output files to the output directory. The "cleanup" section makes a tar file of the hazard curve files in the output directory, runs a script that removes the *faultsource_20* diagnostic files, and deletes the fault coordinate and results configuration files.

Input File Descriptions

Three input files are required to run *faultsource_20*: the fault input file, which contains information needed to run a specific fault or set of faults; the fault coordinate file, which contains the geometric information for the complete set of faults; and the results file, which contains the type of output information desired. The fault coordinate and results files must exist in the directory the program is run from. Example input files are shown in *Example 2—faultsource_20 input file—maximum moment model*, page 21, and *Example 3—faultsource_20 input file—characteristic model*, page 22. Example 2 is for the maximum moment model; Example 3 is for the characteristic model. An input file for the exponential model would be equivalent to that for the characteristic model, with the word "exponential" substituted for "characteristic."

faultsource Input File

Each file contains the name of the dam, its coordinates, and the names of the fault coordinate and results files. Then comes the number of faults contained in the file. The name of the fault must match one in the fault coordinate file. The attenuation function line invokes one of the available attenuation functions. A list of attenuation function codes can be obtained by typing *faultsource_20 -l* (Note that in the runscript in *Example 1—Runscript*, page 18, the code is automatically set to the first in the code list before execution). The magnitude-area code invokes an equation relating magnitude to fault area. Possible choices are

wc94ss, **wc94t**, **wc94n**, and **wc94all** for Wells and Coppersmith (1994) strike-slip, thrust, normal, and "all types," respectively; **wg99** for WCGEP (1999) for California strike-slip events; **ge93sz** for subduction zone events (Geomatrix, 1993); **som99** for Somerville et al. (1999); and **peer** for the Peer PSHA Code Working Group formula. The following "Number of points" line allows one to distribute the computed fault area normally according to the standard deviation provided by the regression. Because of uncertainties regarding the source of uncertainties in the regression process, this option has never been exercised and tested. The Dip Direction and Strike Direction offsets lines are described in *Generating Fault Ruptures and Calculating Distances*, page 7.

The next line describes the recurrence model, and here the formats diverge. For the maximum moment model, the magnitude can be a single value, or a distribution. Possible distributions and their formats are shown in *faultsource Probability Distributions*, page 37. For the characteristic and exponential models a variable characteristic or upper magnitude is allowed, but has not been tested comprehensively. Therefore, use caution in its use. "The Number of Points" line gives the number of points the magnitude distribution is discretized into. At present, the integration over magnitude distribution is performed within the program; separate hazard curves are not generated for each magnitude scenario.

The next section is the Rate Mode. The three choices are Slip Rate, Annual Rate, and Return Period. For the exponential and characteristic models, only the Slip Rate mode can be used. A number of distributions are available for the rate characterizations; these and their formats are shown in *Rate Distribution Characterizations*, page 40.

Fault Coordinate File

The fault coordinate file contains the geometric information for the set of faults analyzed. An example file is shown in *Example 4—faultsource_20 fault coordinate file*, page 22. Each fault entry consists of the fault name, and a list of geographical coordinates defining the updip trace of the fault. The third number is the fault dip; these should be the same for all points except the last, which is zero. Fault type choices are strike (for strike-slip), normal, thrust (or reverse), or oblique.

The dip direction must be either *east* or *west*. In the case where the latitudes of the first two points are exactly the same, *north* or *south* can be used. For a multisegment fault, the dip of the entire fault is defined by the dip direction of the segment specified by the first two points listed. Possible dip distributions and their formats are shown in *Fault Dip Distributions*, page 37. The "Number of Points" line gives the number of discretization points; three points have been found to be adequate in most cases. See *Dip Distribution*, page 4, for guidance on using dip distributions.

Results File

The results file, seen in *Example 5—faultsource_20 results file*, page 24, sets some universal parameters needed for the PSHA. The first set of lines sets the hazard curve fractile levels (in addition to the mean) desired. Usually, the 16th, 50th, and 84th percentiles are called for. The "Periods Processed" line should be set to "all," to get hazard curves for all spectral response periods, or "1" for one period. The next line, "Hazard curve period:," should be set to "all," or the single period needed; 0.0 for PHA, 0.5 for 0.5 second, etc. If the Hazmatrix switch is "on," a file is created that contains a matrix of the full posterior annual frequency distribution for fractile levels spaced 0.01 apart, not just the few fractile levels specified above. This can be used to create a three-dimensional plot of the posterior annual frequency distribution. The following line specifies the period desired and an upper ground motion limit to the matrix. The values in the matrix are log10 return period; this parameter allows for linear x-y (ground motion vs. fractile) axes and contouring of return periods.

Execute the Logic Tree

If all the input parameters for a PSHA consisted of single values, a logic tree would not be necessary. This is rarely the case, however, since our knowledge of these parameters is usually incomplete. There are usually multiple choices of attenuation function, faulting scenarios (e.g., whether a group of faults will rupture independently or simultaneously), fault dip, fault type, correct recurrence model, magnitude distribution, rate distribution, etc. Each choice is represented as a node of the logic tree, along with its assigned weight (McGuire, 2004). The weight can be assigned on the basis of "professional judgement" or can come from a discretization of a probability density function. Given the number of choices, the logic tree can be quite large and complex.

Example 6—Logic Tree Script, page 24, shows a sample logic tree. This one is fairly complicated, but illustrates many of the situations the analyst will encounter. The creation and use of this script is important because it (1) creates a record of how the analysis was done; (2) can easily be modified later for changed or added features and characteristics; and (3) can be easily executed later to reproduce the PSHA results, or operate on changed or modified hazard curve files.

A number of auxiliary programs are used in the logic tree script. These are described in detail in *Logic Tree Programs*, page 43.

When first using a logic tree script it is advisable to run the individual parts of it sequentially and separately; for debugging purposes, and to check to see if the desired results are being achieved. For example, create a file called *LogTree_part.sh* with all the information down to the "#----------" line in it, and below that insert the individual logic tree sections and run them sequentially.

The first section of the logic tree looks much like the runscript, with the fault lineup and project name. However the logic tree may create "new" faults that must be added. These account for intermediate results and various scenarios. In the example, fault 8 is the Cascadia interface assuming magnitude 9 events will occur, and fault 9 is the same assuming magnitude 8.5 events will occur. These scenarios should be weighted and should result in the final Cascadia scenario, which can be called fault 12. The "faults=" line contains an enumeration of the total number of original faults for which hazard curves were initially generated, and the "faults_final=" line gives the seismic sources to portray (in order) in the final hazard curve plot. The "depths=" line is used for the random seismicity treatment: each random seismicity hazard curve contains information about a discrete hypocentral depth; there are five discrete, equally weighted depth scenarios.

This example contains four types of seismic sources: a subduction zone interface, subduction zone intraslab, crustal faults, and random seismicity. Each source type uses its own set of recurrence models and attenuation functions (the functions for crustal faults and random seismicity are the same). The next set of lines contains the spectral response period set to be interpolated to; this is the Abrahamson and Silva (1997) period set. The "select_period=" line lists the periods to create hazard curves for; this will depend on the engineering requirements of the project. The "returnperiods=" line lists the return periods to generate uniform hazard spectra for. The "#--------" line separates this "setup" part of the script from the execution part.

Because each attenuation function uses its own unique period set, the first task is to interpolate each hazard curve to a standard period set. This is done for each seismic source group, since each has its own set of attenuation functions. The next step is to break up each hazard curve file, which contains all periods, into separate ones, one for each period.

The next step is to perform the integration for the crustal faults. There are two equally weighted attenuation functions, and nine activity rates, also equally weighted. Therefore for each hazard curve, the weight is the same. Looping over the crustal faults and periods produces mean and fractile hazard curves for each fault and period.

Next do a similar integration for fault 8, the magnitude 9 Cascadia interface scenario. In this case, there are three attenuation functions with different weights, and nine rates with different weights. The weight in front of each file is obtained by the appropriate multiplication. A similar calculation is performed for the magnitude 9 scenario. Next, integrate for the combined Cascadia scenario, with the magnitude 9 scenario weighted 0.7 and the magnitude 8 scenario 0.3. Use this result in the final hazard curve plot, but compute the two scenarios separately in case you want to examine them individually.

Next, weight the two Cascadia Instraslab source zones. Because there are no rate variations for these sources, the two curves are simply weighted.

This concludes the fault sources. The next task is to deal with the random seismicity. In this case, there is one random source zone, with two attenuation functions, nine rates, and five hypocentral depths. Since all of these discretizations are equally weighted, all weights are the same.

Next, sum the sources to obtain the final results. For each period the final source group is concatenated into a file, and summed (each file contains a program version line at the top—the *delete_other_versionlines* command deletes all except the first one). The results from the final source group are then summed. You now have hazard curves for all sources and the sum, for all periods. At this point, create the cumulative result for all periods; from this you can make the uniform hazard spectra (UHS) for the return periods specified. At this point, you can also sum specific groups of faults for UHS or portrayal on the final hazard curves if, for example, you want UHS or a hazard curve for the category of "nearby crustal faults." This is useful if, for example, you want to select representative time histories for a group of faults with similar characteristics.

The above steps complete the final file generation. The "exit 0" line stops execution; at this point it is advisable to run the remaining commands individually, since the plot appearances will probably have to be iteratively modified.

This workflow method uses the program *xmgrace* for making two-dimensional plots of hazard curves, and relative contribution plots (which provide a source deaggregation). *grace_hazcrv* and *grace_relcon* read a hazard curve file and produce a *xmgrace* plot file that can be opened, manipulated if necessary, and saved in a variety of graphics formats. See *Execute the Logic Tree*, page 15, for usage. After the "PROJNAME=" from the top of the script is executed in the shell window, the "grace_hazcrv -i ...." line can be copied and pasted into the window to generate the plot file. The axis bounds usually have to be modified until an acceptable looking plot is obtained. It's a good idea to make a postscript (.eps) file at this point. This format can be imported directly into a document, or converted into a more appropriate format for presentation purposes.

The "# make relcon files" commands generate the relative contribution data; plot files are created with the *grace_relcon* command, similar to the hazard curve plots. When the figures look acceptable, the figure files and important hazard curve files are moved into the *figures/* directory.

When the analysis is complete and final results and figures are obtained, the intermediate files from the logic tree process (there will be many of them) can be deleted, since the original files generated by *faultsource_20* have been archived as

.tar files. The command *delhaz.sh* does this, by deleting all files starting with *hazdat*. We can always "untar" the original files and start from scratch if necessary. As a final step, the .tar files can be compressed with the *bzip2* command; this reduces the .tar file to about 20 percent of its original size.

Example Scripts and Runfiles

Example 1—Runscript

```
#!/bin/bash

# Scoggins Dam, Oregon
# crustal faults
# May 2006
# generate exponential and maximum moment fault data

# set -n    # uncomment to check for syntax errors

# Fault lineup

# 1  Gales Creek - Northern Section
# 2  Gales Creek - Carpenter Creek Section
# 3  Gales Creek - Gales Valley Section
# 4  Gales Creek - Scoggins Section
# 5  Mount Angel
# 6  Portland Hills
# 7  Tillamook Bay fault zone

# Project name

PROJNAME=scoggins

let nfaults=7

nmults=( 0 1 2 3 4 5 6 7 )
multfact1=( 1.528 1.891 2.187 2.442 2.683 2.959 3.299 3.792 )
multfact2=( 1.634 2.070 2.425 2.732 3.046 3.410 3.858 4.508 )
multfact3=( 1.406 1.686 1.914 2.110 2.288 2.485 2.727 3.079 )
multfact4=( 1.068 1.115 1.153 1.186 1.219 1.257 1.304 1.372 )
multfact5=( 1.212 1.359 1.481 1.611 1.756 1.924 2.130 2.430 )
multfact6=( 1.380 1.642 1.855 2.048 2.260 2.506 2.808 3.247 )
multfact7=( 1.697 2.177 2.567 2.905 3.245 3.638 4.122 4.825 )

attfuncs=( ge97r as97hrF )    # attenuation function codes
recurmodels=( mm )    # recurrence models (any combination of mm, exp, and char)

# set input file and target directories

USERHOME=/home/roland

INFILEDIR=../infiles
TARGETDIR=../psha
```

```
# define functions here

        # if attenuation function not the original, switch to next
        # if original, set to first function in list

switch_attfunc ()
{
        if [ $attseq != 0 ]; then
                let attprev=attseq-1
                sed s/${attfuncs[$attprev]}/$attstr/ $INFILEDIR/$INFILE > $INFILEDIR/xx
                mv $INFILEDIR/xx $INFILEDIR/$INFILE
        else
                initialize_attfunc ${attfuncs[0]} < $INFILEDIR/$INFILE > xx
                mv xx $INFILEDIR/$INFILE
        fi
}

        # run faultsource and save output

run_faultsource_and_save_output ()
{
        if [ $attstr = jb97x ] ; then
                run_faultsource_with_jb97_and_save_output
        else
                run_faultsource_straight_and_save_output
        fi
}

run_faultsource_straight_and_save_output ()
{
        if faultsource_20 < $INFILEDIR/$INFILE >& ./faultsource.out
        then
                mv fs2.0.hazdat.txt $TARGETDIR/$INITOUTFILE
        else
                echo "faultsource_20 crashed with input file $INFILE; faultsource.out"
                exit 1
        fi
}

        # run faultsource with jb97 and save output

run_faultsource_with_jb97_and_save_output ()
{
        if faultsource_20 -v 750 < $INFILEDIR/$INFILE >& ./faultsource.out
        then
                mv fs2.0.hazdat.txt $TARGETDIR/$INITOUTFILE
        else
                echo "faultsource_20 crashed with input file $INFILE; faultsource.out"
                exit 1
        fi
}
```

```
	# check for last attenuation function; if true, switch back to original

revert_to_original_attfunc ()
{
	let nattfuncs=attseq+1
	if [ $nattfuncs -eq ${#attfuncs[@]} ]; then
		sed s/$attstr/${attfuncs[0]}/ $INFILEDIR/$INFILE > $INFILEDIR/xx
		mv $INFILEDIR/xx $INFILEDIR/$INFILE
	fi
}


# get fault coordinate and results file

FAULTFILE=Scoggins_5-06.coords
RESULTS=results.in.all

cp $INFILEDIR/$FAULTFILE .
cp $INFILEDIR/$RESULTS .

scaleroot=multfact

let faultnum=1

while [ $faultnum -le $nfaults ]; do

	for recurstr in ${recurmodels[@]}; do

		let attseq=0

		for attstr in ${attfuncs[@]}; do

			echo " Begin fault $faultnum - attfunc= $attstr"

			INFILE=$PROJNAME.fault$faultnum.$recurstr.in
			echo INFILE = $PROJNAME.fault$faultnum.$recurstr.in

			# switch attenuation function if not first

			switch_attfunc

			INITOUTFILE=hazdat.fault$faultnum.$recurstr.r1.$attstr
			echo INITOUTFILE = $INITOUTFILE

			run_faultsource_and_save_output

			# scale each hazard curve by multfact

			let ratenum=2

			name1=$scaleroot$faultnum

			for multnum in ${nmults[@]}; do

				OUTFILE=hazdat.fault$faultnum.$recurstr.r$ratenum.$attstr
```

```
                    name2="\${$name1[\$multnum]}"

                    scalefactor=`eval echo $name2`

                    scalehaz $scalefactor < $TARGETDIR/$INITOUTFILE \
                          > $TARGETDIR/$OUTFILE

                    echo Finished with $OUTFILE

                    let ratenum++

               done

               # switch back to first attfunc if last attfunc

               revert_to_original_attfunc

               let attseq++

               echo "  End atten loop- attfunc= $attstr attseq= $attseq"

          done

     done

     let faultnum++

done

# clean up

cd $TARGETDIR
tar cf fault_files.$PROJNAME_5-06.tar hazdat*fault*
cd ../runpsha

rm $FAULTFILE
rm $RESULTS

fsclean.sh
```

Example 2—*faultsource_20* input file—maximum moment model

```
Scoggins Dam, Oregon
 45.4728 -123.2011
 Fault Catalog: Scoggins_5-06.coords
 Results File: results.in.all
 1 Fault Processed

Gales Creek - Northern Section
  Attenuation Function= ge97r
  Magnitude-Area formula= wc94all
    Number of points= 1
  Dip Direction offsets= 2
```

Strike Direction offsets= 2
Maximum_Moment Model
Variable Magnitude
Distribution Type= gaussian
Mean= 6.9 magunits
Standard Deviation= .20
Number of points= 5
Rate Mode: Slip Rate
Distribution Type= none
Rate= 0.1791 mm/yr

Example 3—*faultsource_20* input file—characteristic model

Hungry Horse Dam, Montana
48.342 -114.013
Fault Catalog: HungHorse_7-05.coords
Results File: results.in.all
1 Fault Processed
Swan Fault-Lake Blaine
Attenuation Function= ge97r
Magnitude-Area formula= wc94n
Number of points= 1
Dip Direction offsets= 2
Strike Direction offsets= 2
Characteristic Model
b-value= 0.7
Minimum Magnitude= 5.5
Fixed Maximum Magnitude= 7.2
Rate Mode: Slip Rate
Distribution Type= none
Rate= 0.053 mm/yr

Example 4—*faultsource_20* fault coordinate file

Gales Creek - Northern Section
45.67113 -123.28106 80
45.87544 -123.66774 0
Fault type= strike
Dip Direction= west
Depth range= 0.000 - 30.000 km
Distribution type= triangle,sym
±10.0 degrees
Number of points= 3

Gales Creek - Carpenter Creek Section
45.48317 -123.14687 80
45.53834 -123.24225 80
45.66551 -123.30958 0
Fault type= strike
Dip Direction= west
Depth range= 0.000 - 30.000 km
Distribution type= triangle,sym
±10.0 degrees
Number of points= 3

Gales Creek - Gales Valley Section
45.47335 -123.12863 80
45.58276 -123.20999 80
45.66785 -123.27732 0
Fault type= strike
Dip Direction= west
Depth range= 0.000 - 30.000 km
Distribution type= triangle,sym
±10.0 degrees
Number of points= 3

Gales Creek - Scoggins Section
45.50435 -123.25645 80
45.47178 -123.20400 80
45.45053 -123.17725 80
45.40459 -123.15531 80
45.36756 -123.08846 0
Fault type= strike
Dip Direction= west
Depth range= 0.000 - 30.000 km
Distribution type= triangle,sym
±10.0 degrees
Number of points= 3

Mount Angel
45.15542 -122.87059 80
45.07683 -122.80954 80
44.96562 -122.61137 0
Fault type= strike
Dip Direction= east
Depth range= 0.000 - 30.000 km
Distribution type= none

Portland Hills
45.72149 -122.93414 80
45.52834 -122.70169 80
45.37115 -122.54281 0
Fault type= strike
Dip Direction= west
Depth range= 0.000 - 30.000 km
Distribution type= none

Tillamook Bay fault zone
45.55896 -123.93634 60
45.51615 -123.84176 60
45.41188 -123.62719 0
Fault type= reverse
Dip Direction= east
Depth range= 5.000 - 25.000 km
Distribution type= triangle,sym
±20.0 degrees
Number of points= 3

Cascadia-Intermediate-Puget
  46.7174 -123.5600 16.7
  47.9907 -123.4597 0
  Fault type= thrust
  Dip Direction= east
  Depth range= 25.000 - 55.000 km
  Distribution type= none

Cascadia-Intermediate-Portland
  44.8900 -123.6982 16.7
  46.7174 -123.5600 0
  Fault type= thrust
  Dip Direction= east
  Depth range= 25.000 - 55.000 km
  Distribution type= none

Example 5—*faultsource_20* results file

3 percentile levels
  .16
  .50
  .84
Periods processed= all
Hazard curve period: all
Hazmatrix switch: off
Hazmatrix period= 0.0 limit= 1.0 g

Example 6—Logic Tree Script

```
#!/bin/sh

# logic tree for Scoggins Dam, Oregon
# May 2006

# set -n    # uncomment to check for syntax errors

# Fault lineup

# 1  Gales Creek - Northern Section
# 2  Gales Creek - Carpenter Creek Section
# 3  Gales Creek - Gales Valley Section
# 4  Gales Creek - Scoggins Section
# 5  Mount Angel
# 6  Portland Hills
# 7  Tillamook Bay fault zone
# 8   Cascadia-Interface Mag 9
# 9   Cascadia-Interface Mag 8.5
# 10  Cascadia-Intermediate-Puget
# 11  Cascadia-Intermediate-Portland

# 12  Cascade-Interface scenarios

PROJNAME=scoggins

faults=( 1 2 3 4 5 6 7 8 9 10 11 )
```

```
let nfaults=11
let n_crustal_faults=7
let ndepths=5

# final fault grouping

faults_final=( 12 10 11 1 2 3 4 5 6 7 )

depths=(1 2 3 4 5)

rates=( 1 2 3 4 5 6 7 8 9 )

crustal_attfuncs=( ge97r as97hrF )
intraslab_attfuncs=( ab03Ib yc97rI )
interface_attfuncs=( ab03Tb yc97rT gea02r )

# as97hrF period set

allperiods=( 0.0 0.02 0.03 0.04 0.05 0.06 0.075 0.09 0.1 0.12 \
0.15 0.17 0.2 0.24 0.3 0.36 0.4 0.46 0.5 0.6 0.75 0.85 1.0 1.5 2.0 )

# periods for hazard curves

select_period=( 0.0 0.5 0.75 1.0 )

# return periods for UHS

returnperiods=( 2500 5000 10000 50000)

#------------------------------------------------------------


# interpolate to as97hrF period set

let faultnum=1

while [ $faultnum -le $n_crustal_faults ]; do

        for ratenum in ${rates[@]}; do

                interp_allhazdat -t as97hrF < hazdat.fault$faultnum.mm.r$ratenum.ge97r \
                        > hazdat.fault$faultnum.mm.r$ratenum.ge97r.int

        done

        let faultnum++

done
```

```
# Interface

for attstr in ${interface_attfuncs[@]}; do

        for ratenum in ${rates[@]}; do

                interp_allhazdat -t as97hrF < hazdat.fault8.mm.r$ratenum.$attstr \
                        > hazdat.fault8.mm.r$ratenum.$attstr.int

                interp_allhazdat -t as97hrF < hazdat.fault9.mm.r$ratenum.$attstr \
                        > hazdat.fault9.mm.r$ratenum.$attstr.int

        done
done

# Intraslab

for attstr in ${intraslab_attfuncs[@]}; do

        interp_allhazdat -t as97hrF < hazdat.fault10.exp.$attstr \
                > hazdat.fault10.exp.$attstr.int

        interp_allhazdat -t as97hrF < hazdat.fault11.exp.$attstr \
                > hazdat.fault11.exp.$attstr.int

done


echo "Done with period interpolation"


# separate all periods, 7 crustal faults

let faultnum=1

while [ $faultnum -le $n_crustal_faults ]; do

        for period in ${allperiods[@]}; do

                for ratenum in ${rates[@]}; do

                        strip_period $period < hazdat.fault$faultnum.mm.r$ratenum.ge97r.int > \
                                hazdat.fault$faultnum.mm.r$ratenum.ge97r.$period
                        strip_period $period < hazdat.fault$faultnum.mm.r$ratenum.as97hrF > \
                                hazdat.fault$faultnum.mm.r$ratenum.as97hrF.$period

                done
        done

        let faultnum++

done
```

```
# separate all periods, Cascadia Interface

let faultnum=8

while [ $faultnum -le 9 ]; do

	for period in ${allperiods[@]}; do

		for ratenum in ${rates[@]}; do

			strip_period $period < hazdat.fault$faultnum.mm.r$ratenum.ab03Tb.int > \
				hazdat.fault$faultnum.mm.r$ratenum.ab03Tb.$period
			strip_period $period < hazdat.fault$faultnum.mm.r$ratenum.yc97rT.int > \
				hazdat.fault$faultnum.mm.r$ratenum.yc97rT.$period
			strip_period $period < hazdat.fault$faultnum.mm.r$ratenum.gea02r.int > \
				hazdat.fault$faultnum.mm.r$ratenum.gea02r.$period

		done
	done

	let faultnum++

done

# separate all periods, Cascadia Intraslab

let faultnum=10

while [ $faultnum -le 11 ]; do

	for period in ${allperiods[@]}; do

			strip_period $period < hazdat.fault$faultnum.exp.ab03Ib.int > \
				hazdat.fault$faultnum.exp.ab03Ib.$period
			strip_period $period < hazdat.fault$faultnum.exp.yc97rI.int > \
				hazdat.fault$faultnum.exp.yc97rI.$period

	done

	let faultnum++

done

echo "Done with period separation"

# weight attfunc given rate

	# ge97r= .5, as97hrF= .5

let faultnum=1
```

```
while [ $faultnum -le $n_crustal_faults ]; do

    for period in ${allperiods[@]}; do

    hazdat_fracs << ! > hazdat.fault$faultnum.fracs.$period

        0.055556 hazdat.fault$faultnum.mm.r1.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r2.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r3.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r4.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r5.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r6.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r7.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r8.ge97r.$period
        0.055556 hazdat.fault$faultnum.mm.r9.ge97r.$period

        0.055556 hazdat.fault$faultnum.mm.r1.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r2.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r3.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r4.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r5.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r6.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r7.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r8.as97hrF.$period
        0.055556 hazdat.fault$faultnum.mm.r9.as97hrF.$period
!

    done

    let faultnum++

done

echo "Done with integration for fractiles, crustal faults"


# isolate, weight Cascadia Interface 8.5-9 scenarios

    # ab03Tb : .7
    # gea02r : .15
    # yc97rT : .15

    # rate1 : .00849
    # rate2 : .03808
    # rate3 : .11117
    # rate4 : .21136
    # rate5 : .26182
    # rate6 : .21136
    # rate7 : .11117
    # rate8 : .03808
    # rate9 : .00849
```

```
# Cascade Interface - Mag 9

for period in ${allperiods[@]}; do

        hazdat_fracs << ! > hazdat.fault8.fracs.$period

                0.005936 hazdat.fault8.mm.r1.ab03Tb.$period
                0.026656 hazdat.fault8.mm.r2.ab03Tb.$period
                0.077819 hazdat.fault8.mm.r3.ab03Tb.$period
                0.147952 hazdat.fault8.mm.r4.ab03Tb.$period
                0.183274 hazdat.fault8.mm.r5.ab03Tb.$period
                0.147952 hazdat.fault8.mm.r6.ab03Tb.$period
                0.077819 hazdat.fault8.mm.r7.ab03Tb.$period
                0.026656 hazdat.fault8.mm.r8.ab03Tb.$period
                0.005936 hazdat.fault8.mm.r9.ab03Tb.$period

                0.001274 hazdat.fault8.mm.r1.gea02r.$period
                0.005712 hazdat.fault8.mm.r2.gea02r.$period
                0.016676 hazdat.fault8.mm.r3.gea02r.$period
                0.031704 hazdat.fault8.mm.r4.gea02r.$period
                0.039273 hazdat.fault8.mm.r5.gea02r.$period
                0.031704 hazdat.fault8.mm.r6.gea02r.$period
                0.016676 hazdat.fault8.mm.r7.gea02r.$period
                0.005712 hazdat.fault8.mm.r8.gea02r.$period
                0.001274 hazdat.fault8.mm.r9.gea02r.$period

                0.001274 hazdat.fault8.mm.r1.yc97rT.$period
                0.005712 hazdat.fault8.mm.r2.yc97rT.$period
                0.016676 hazdat.fault8.mm.r3.yc97rT.$period
                0.031704 hazdat.fault8.mm.r4.yc97rT.$period
                0.039273 hazdat.fault8.mm.r5.yc97rT.$period
                0.031704 hazdat.fault8.mm.r6.yc97rT.$period
                0.016676 hazdat.fault8.mm.r7.yc97rT.$period
                0.005712 hazdat.fault8.mm.r8.yc97rT.$period
                0.001274 hazdat.fault8.mm.r9.yc97rT.$period

!

        done

# Cascade Interface - Mag 8.5

for period in ${allperiods[@]}; do

        hazdat_fracs << ! > hazdat.fault9.fracs.$period

                0.005936 hazdat.fault9.mm.r1.ab03Tb.$period
                0.026656 hazdat.fault9.mm.r2.ab03Tb.$period
                0.077819 hazdat.fault9.mm.r3.ab03Tb.$period
                0.147952 hazdat.fault9.mm.r4.ab03Tb.$period
                0.183274 hazdat.fault9.mm.r5.ab03Tb.$period
                0.147952 hazdat.fault9.mm.r6.ab03Tb.$period
                0.077819 hazdat.fault9.mm.r7.ab03Tb.$period
                0.026656 hazdat.fault9.mm.r8.ab03Tb.$period
                0.005936 hazdat.fault9.mm.r9.ab03Tb.$period
```

```
          0.001274 hazdat.fault9.mm.r1.gea02r.$period
          0.005712 hazdat.fault9.mm.r2.gea02r.$period
          0.016676 hazdat.fault9.mm.r3.gea02r.$period
          0.031704 hazdat.fault9.mm.r4.gea02r.$period
          0.039273 hazdat.fault9.mm.r5.gea02r.$period
          0.031704 hazdat.fault9.mm.r6.gea02r.$period
          0.016676 hazdat.fault9.mm.r7.gea02r.$period
          0.005712 hazdat.fault9.mm.r8.gea02r.$period
          0.001274 hazdat.fault9.mm.r9.gea02r.$period

          0.001274 hazdat.fault9.mm.r1.yc97rT.$period
          0.005712 hazdat.fault9.mm.r2.yc97rT.$period
          0.016676 hazdat.fault9.mm.r3.yc97rT.$period
          0.031704 hazdat.fault9.mm.r4.yc97rT.$period
          0.039273 hazdat.fault9.mm.r5.yc97rT.$period
          0.031704 hazdat.fault9.mm.r6.yc97rT.$period
          0.016676 hazdat.fault9.mm.r7.yc97rT.$period
          0.005712 hazdat.fault9.mm.r8.yc97rT.$period
          0.001274 hazdat.fault9.mm.r9.yc97rT.$period

!

     done

 # Cascade Interface - Scenarios

     # Mag 9   : .7
     # Mag 8.5 : .3

for period in ${allperiods[@]}; do

     hazdat_fracs << ! > hazdat.fault12.fracs.$period

          0.004155 hazdat.fault8.mm.r1.ab03Tb.$period
          0.018659 hazdat.fault8.mm.r2.ab03Tb.$period
          0.054473 hazdat.fault8.mm.r3.ab03Tb.$period
          0.103566 hazdat.fault8.mm.r4.ab03Tb.$period
          0.128292 hazdat.fault8.mm.r5.ab03Tb.$period
          0.103566 hazdat.fault8.mm.r6.ab03Tb.$period
          0.054473 hazdat.fault8.mm.r7.ab03Tb.$period
          0.018659 hazdat.fault8.mm.r8.ab03Tb.$period
          0.004155 hazdat.fault8.mm.r9.ab03Tb.$period

          0.000892 hazdat.fault8.mm.r1.gea02r.$period
          0.003998 hazdat.fault8.mm.r2.gea02r.$period
          0.011673 hazdat.fault8.mm.r3.gea02r.$period
          0.022193 hazdat.fault8.mm.r4.gea02r.$period
          0.027491 hazdat.fault8.mm.r5.gea02r.$period
          0.022193 hazdat.fault8.mm.r6.gea02r.$period
          0.011673 hazdat.fault8.mm.r7.gea02r.$period
          0.003998 hazdat.fault8.mm.r8.gea02r.$period
          0.000892 hazdat.fault8.mm.r9.gea02r.$period

          0.000892 hazdat.fault8.mm.r1.yc97rT.$period
```

```
        0.003998 hazdat.fault8.mm.r2.yc97rT.$period
        0.011673 hazdat.fault8.mm.r3.yc97rT.$period
        0.022193 hazdat.fault8.mm.r4.yc97rT.$period
        0.027491 hazdat.fault8.mm.r5.yc97rT.$period
        0.022193 hazdat.fault8.mm.r6.yc97rT.$period
        0.011673 hazdat.fault8.mm.r7.yc97rT.$period
        0.003998 hazdat.fault8.mm.r8.yc97rT.$period
        0.000892 hazdat.fault8.mm.r9.yc97rT.$period

        0.001781 hazdat.fault9.mm.r1.ab03Tb.$period
        0.007997 hazdat.fault9.mm.r2.ab03Tb.$period
        0.023346 hazdat.fault9.mm.r3.ab03Tb.$period
        0.044386 hazdat.fault9.mm.r4.ab03Tb.$period
        0.054982 hazdat.fault9.mm.r5.ab03Tb.$period
        0.044386 hazdat.fault9.mm.r6.ab03Tb.$period
        0.023346 hazdat.fault9.mm.r7.ab03Tb.$period
        0.007997 hazdat.fault9.mm.r8.ab03Tb.$period
        0.001781 hazdat.fault9.mm.r9.ab03Tb.$period

        0.000382 hazdat.fault9.mm.r1.gea02r.$period
        0.001714 hazdat.fault9.mm.r2.gea02r.$period
        0.005003 hazdat.fault9.mm.r3.gea02r.$period
        0.009511 hazdat.fault9.mm.r4.gea02r.$period
        0.011782 hazdat.fault9.mm.r5.gea02r.$period
        0.009511 hazdat.fault9.mm.r6.gea02r.$period
        0.005003 hazdat.fault9.mm.r7.gea02r.$period
        0.001714 hazdat.fault9.mm.r8.gea02r.$period
        0.000382 hazdat.fault9.mm.r9.gea02r.$period

        0.000382 hazdat.fault9.mm.r1.yc97rT.$period
        0.001714 hazdat.fault9.mm.r2.yc97rT.$period
        0.005003 hazdat.fault9.mm.r3.yc97rT.$period
        0.009511 hazdat.fault9.mm.r4.yc97rT.$period
        0.011782 hazdat.fault9.mm.r5.yc97rT.$period
        0.009511 hazdat.fault9.mm.r6.yc97rT.$period
        0.005003 hazdat.fault9.mm.r7.yc97rT.$period
        0.001714 hazdat.fault9.mm.r8.yc97rT.$period
        0.000382 hazdat.fault9.mm.r9.yc97rT.$period
!

    done


# isolate, weight Intraslab

    # ab03TI : .7
    # yc97rI : .3

 # Cascade Intraslab - Puget
```

```
for period in ${allperiods[@]}; do

        wtavg_ya << ! > hazdat.fault10.fracs.$period

                0.70 hazdat.fault10.exp.ab03Ib.$period
                0.30 hazdat.fault10.exp.yc97rI.$period
!

        done

 # Cascade Intraslab - Portland

for period in ${allperiods[@]}; do

        wtavg_ya << ! > hazdat.fault11.fracs.$period

                0.70 hazdat.fault11.exp.ab03Ib.$period
                0.30 hazdat.fault11.exp.yc97rI.$period
!

        done


# weight random seismicity results, break into periods

# interpolate to as97hrF period set

for ratenum in ${rates[@]}; do

        for dep in ${depths[@]}; do

                 interp_allhazdat -t as97hrF < hazdat.randomsource1.r$ratenum.d$dep.ge97r \
                         > hazdat.randomsource1.r$ratenum.d$dep.ge97r.int

        done
done

# separate all periods

for period in ${allperiods[@]}; do

        for ratenum in ${rates[@]}; do

                for dep in ${depths[@]}; do

                        strip_period $period < hazdat.randomsource1.r$ratenum.d$dep.ge97r.int > \
                                hazdat.randomsource1.r$ratenum.d$dep.ge97r.$period
                        strip_period $period < hazdat.randomsource1.r$ratenum.d$dep.as97hrF > \
                                hazdat.randomsource1.r$ratenum.d$dep.as97hrF.$period

                done
        done
done

echo "Done with random seismicity period separation"
```

```
# get fractiles for random seismicity

for period in ${allperiods[@]}; do

        hazdat_fracs << ! > hazdat.randomsource1.fracs.$period

                0.011111 hazdat.randomsource1.r1.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r1.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r1.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r1.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r1.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r2.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r2.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r2.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r2.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r2.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r3.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r3.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r3.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r3.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r3.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r4.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r4.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r4.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r4.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r4.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r5.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r5.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r5.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r5.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r5.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r6.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r6.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r6.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r6.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r6.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r7.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r7.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r7.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r7.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r7.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r8.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r8.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r8.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r8.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r8.d5.ge97r.$period
                0.011111 hazdat.randomsource1.r9.d1.ge97r.$period
                0.011111 hazdat.randomsource1.r9.d2.ge97r.$period
                0.011111 hazdat.randomsource1.r9.d3.ge97r.$period
                0.011111 hazdat.randomsource1.r9.d4.ge97r.$period
                0.011111 hazdat.randomsource1.r9.d5.ge97r.$period

                0.011111 hazdat.randomsource1.r1.d1.as97hrF.$period
                0.011111 hazdat.randomsource1.r1.d2.as97hrF.$period
```

```
        0.011111 hazdat.randomsource1.r1.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r1.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r1.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r2.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r2.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r2.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r2.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r2.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r3.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r3.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r3.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r3.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r3.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r4.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r4.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r4.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r4.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r4.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r5.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r5.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r5.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r5.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r5.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r6.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r6.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r6.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r6.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r6.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r7.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r7.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r7.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r7.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r7.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r8.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r8.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r8.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r8.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r8.d5.as97hrF.$period
        0.011111 hazdat.randomsource1.r9.d1.as97hrF.$period
        0.011111 hazdat.randomsource1.r9.d2.as97hrF.$period
        0.011111 hazdat.randomsource1.r9.d3.as97hrF.$period
        0.011111 hazdat.randomsource1.r9.d4.as97hrF.$period
        0.011111 hazdat.randomsource1.r9.d5.as97hrF.$period
!

done

echo “Done with random seismicity integration for fractiles”

echo “Done with random seismicity”
```

```
# Sum sources to be used for hazard curves

for period in ${allperiods[@]}; do

    # remove file if it already exists, otherwise create it

    if [ -f hazdat.fracs.pre_final.$period ] ; then
        rm hazdat.fracs.pre_final.$period
    else
        touch hazdat.fracs.pre_final.$period
    fi


    for faultnum in ${faults_final[@]}; do

        cat hazdat.fault$faultnum.fracs.$period >> hazdat.fracs.pre_final.$period

    done

    cat hazdat.randomsource1.fracs.$period >> hazdat.fracs.pre_final.$period

    # delete extra version lines

    delete_other_versionlines <  hazdat.fracs.pre_final.$period > xx
    mv xx hazdat.fracs.pre_final.$period

    sumhazdat -a < hazdat.fracs.pre_final.$period > hazdat.fracs.final.$period

done

# create all-period cumulative file for UHS

filename=hazdat.fracs.cum.final

if [ -f $filename ] ; then
    rm $filename
else
    touch $filename
fi

let periodseq=0

for period in ${allperiods[@]}; do

    if [ $periodseq -eq 0 ]; then

        copycum < hazdat.fracs.final.$period >> $filename

    else

        copycum < hazdat.fracs.final.$period | sed 1,2d  >> $filename
    fi
```

```
        let periodseq++

done

# create cumulative files for selected periods

for period in ${select_period[@]}; do

        copycum < hazdat.fracs.final.$period > hazdat.fracs.cum.$period

done



# make UHS files

for retpd in ${returnperiods[@]}; do

        post_uhs -r $retpd < hazdat.fracs.cum.final > uhs.$PROJNAME.$retpd.dat

done

echo "Done with final file creation"

exit 0

# make mean hazard curve files

grace_hazcrv -i hazdat.fracs.final.0.0 -x 1.6 -a 1e-5 -b .1 > $PROJNAME.hazdat.0.0.agr
grace_hazcrv -i hazdat.fracs.final.0.5 -x 2.6 -a 1e-5 -b .1 -p 0.5 > $PROJNAME.hazdat.0.5.agr
grace_hazcrv -i hazdat.fracs.final.0.75 -x 2.0 -a 1e-5 -b .1 -p 0.75 > $PROJNAME.hazdat.0.75.agr
grace_hazcrv -i hazdat.fracs.final.1.0 -x 1.5 -a 1e-5 -b .1 -p 1.0 > $PROJNAME.hazdat.1.0.agr

# make fractile hazard curve files

grace_hazcrv -i hazdat.fracs.cum.0.0 -f -x 1.6 -a 1e-5 -b .1 > $PROJNAME.hazdat.fracs.0.0.agr
grace_hazcrv -i hazdat.fracs.cum.0.5 -f -x 2.6 -a 1e-5 -b .1 -p 0.5 >
$PROJNAME.hazdat.fracs.0.5.agr
grace_hazcrv -i hazdat.fracs.cum.0.75 -f -x 2.0 -a 1e-5 -b .1 -p 0.75 >
$PROJNAME.hazdat.fracs.0.75.agr
grace_hazcrv -i hazdat.fracs.cum.1.0 -f -x 1.5 -a 1e-5 -b .1 -p 1.0 >
$PROJNAME.hazdat.fracs.1.0.agr


# make relcon files

for period in ${select_period[@]}; do

        post_relcon < hazdat.fracs.final.$period > relcon.$PROJNAME.$period

done
```

```
    # relative contributions

grace_relcon -i relcon.$PROJNAME.0.0 -x 1.6 -t .1 > $PROJNAME.relcon.0.0.agr
grace_relcon -i relcon.$PROJNAME.0.5 -x 2.6 -p 0.5 -t .1 > $PROJNAME.relcon.0.5.agr
grace_relcon -i relcon.$PROJNAME.0.75 -x 2.0 -p 0.75 -t .1 > $PROJNAME.relcon.0.75.agr
grace_relcon -i relcon.$PROJNAME.1.0 -x 1.5 -p 1.0 -t .1 > $PROJNAME.relcon.1.0.agr

# move results to ../figures

mv *.agr ../figures
mv uhs.* ../figures
mv *.eps ../figures
mv relcon.* ../figures
mv hazdat.fracs.cum.final ../figures
mv hazdat.fracs.final.0.0 ../figures
mv hazdat.fracs.final.0.5 ../figures
mv hazdat.fracs.final.0.75 ../figures
mv hazdat.fracs.final.1.0 ../figures
```

faultsource Probability Distributions

This section describes the various probability distributions and their parameterizations, that are currently available for use in *faultsource_20*. The distributions are for three specific applications: fault dip distributions that are used in the fault coordinate file; magnitude distributions that are used in input files for the maximum moment recurrence distribution; and rate distributions (slip rate, annual rate, or return period) that are used in all input files. When using these, keep in mind that leading white space is not important, but subsequent white space is. The input file parser counts the number of continuous character strings, delimited by whitespace or tabs, to arrive at and read the correct rate parameter.

Fault Dip Distributions

Fault dip distributions follow the depth range line in the fault coordinate file (see *Example 4—faultsource_20 fault coordinate file*, p. 22). Note that there are two characterizations for the gaussian distribution. Notes are provided in parentheses. Gaussian distributions are truncated at ±3 standard deviations. Gaussian Format II specifies the endpoints.

None

Distribution Type= none

(Single dip, specified in coordinate lines)

Gaussian—Format I

Distribution Type= gaussian
Standard Deviation= 10 degrees
Number of points= 3

Gaussian—Format II

Distribution Type= gaussian
Range= 60 to 80 degrees
Number of points= 3

(Gaussian distributions)

Uniform

Distribution Type= uniform
±10 degrees
Number of points= 3

Symmetric Triangle

Distribution Type= triangle,sym
±10 degrees
Number of points= 3

Asymmetric Triangle

Distribution Type= triangle,asym
Range= 45 to 80 degrees
Number of points= 3

The "Apex" of the distribution is that given in the coordinate lines.

Discrete

Distribution Type= discrete
Number of points= 3
Dip= 30 degrees Weight= 0.3
Dip= 45 degrees Weight= 0.4
Dip= 60 degrees Weight= 0.3

Magnitude Distributions for Maximum Moment Model

None

Fixed Magnitude
Magnitude= 7.5

Gaussian—Format I

Variable Magnitude
Distribution Type= gaussian

Mean= 7.5 magunits
Standard Deviation= .20
Number of points= 3

Gaussian—Format II

Variable Magnitude
Distribution Type= gaussian
Range= 6.6 to 7.0 magunits
Number of points= 3

Uniform

Variable Magnitude
Distribution Type= uniform
Range= 6.6 to 7.0 magunits
Number of points= 3

Symmetric Triangle

Variable Magnitude
Distribution Type= triangle,sym
Range= 7.0 to 7.3 magunits
Number of points= 3

Asymmetric Triangle

Variable Magnitude
Distribution Type= triangle,asym
Range= 7.0 to 7.3 magunits
Apex= 7.1 magunits
Number of points= 5

Trapezoidal

Distribution Type= trapezoid
A= 6.5 magunits
B= 7.0
C= 7.5
D= 8.0
Number of points= 5

The trapezoidal distribution looks like figure 3.

Distribution Type= discrete
Number of points= 3
magnitude= 1.0 magunits Weight= 0.2
magnitude= 2.0 magunits Weight= 0.6
magnitude= 3.0 magunits Weight= 0.2

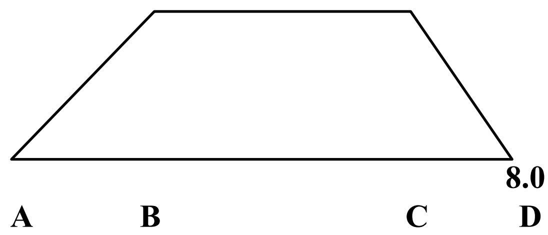


Figure 3.—Example of trapezoidal probability density function.

Generic

Distribution Type= generic
Number of discretization points= 7
Number of distribution points= 5
Range= 6.0 to 8.5 magunits
x1= 6.0 y1= 0
x2= 6.5 y2= 0.25
x3= 7.0 y3= 0.5
x4= 7.5 y4= 1.0
x5= 8.0 y5= 0

The "generic" distribution allows a great deal of flexibility. The x points give the x coordinates of the distribution; the y points give the relative probability density (from 0 to 1) on the y-axis. Note that the first and last y points must be zero.

Magnitude Characterization for Exponential and Characteristic Models

None

Fixed Maximum Magnitude= 7.5

Fixed Characteristic Magnitude= 7.5

Rate Distribution Characterizations

The units for the rate distributions depend on which Rate Mode is used: for Annual Rate mode, *events/yr* should be used; for Return Mode, *yrs/event* should be used; and for Slip Rate mode, either *mm/yr* or *cm/yr* should be used.

None

Distribution Type= none
Rate= .001 (yrs/event,events/yr,mm/yr,cm/yr)

Gaussian—Format I

Distribution Type= gaussian
Range= .01 to .02 (events/yr,mm/yr,cm/yr)
Number of points= 5

Gaussian—Format II

Distribution Type= gaussian
Mean= 7.5 units (events/yr,mm/yr,cm/yr)
Standard Deviation= .25
Number of points= 5

Uniform

Distribution Type= uniform
Range= .01 to .02 units (events/yr,mm/yr,cm/yr)
Number of points= 5

Symmetric Triangle

Distribution Type= triangle,sym
Range= .01 to .02 units (events/yr,mm/yr,cm/yr)
Number of points= 5

Asymmetric Triangle

Distribution Type= triangle,asym
Range= .01 to .09 units (events/yr,mm/yr,cm/yr)
Apex= .03
Number of points= 7

Trapezoidal

Distribution Type= trapezoid
A= .01 units (events/yr,mm/yr,cm/yr)
B= .02
C= .10
D= .15
Number of points= 5

Lognormal

Distribution Type= lognormal
median= .002273 (events/yr, mm/yr, cm/yr, yrs/event)
log_stdev= 0.58
Number of points= 5

For the lognormal distribution, the median is non-log value; standard deviation is in (natural) log units.

Discrete

Distribution Type= discrete
Number of points= 3
rate= 1.0 (mm/yr, cm/yr, events/yr, yrs/event) Weight= 0.2
rate= 2.0 (mm/yr, cm/yr, events/yr, yrs/event) Weight= 0.6
rate= 3.0 (mm/yr, cm/yr, events/yr, yrs/event) Weight= 0.2

Generic

Distribution Type= generic
Number of discretization points= 3
Number of distribution points= 5
Range= .12 to 0.41 mm/yr
x1= 0.12 y1= 0
x2= 0.12 y2= 0.5
x3= 0.26 y3= 1.00
x4= 0.41 y4= 0.5
x5= 0.41 y5= 0

Blockprobs

Distribution Type= blockprobs
Number of discretization points= 9
Number of blocks= 4
Block1: triangle   Prob= 0.1   0.0 to 0.012 mm/yr
Block2: rectangle   Prob= 0.6 0.012 to 0.08 mm/yr
Block3: l_trapezoid   Prob= 0.2 0.08 to 0.1 mm/yr
height-ratio= 0.5
Block4: triangle   Prob= 0.1 0.1 to 0.5 mm/yr

An example of a "blockprobs" distribution is shown in figure 4. A "block" can be a triangle, rectangle, or trapezoid. Each block is parameterized by a probability, and x-axis endpoints. Trapezoids are either "left" or "right" (l_trapezoid or r_trapezoid), depending on whether the left or right side of the trapezoid is higher. The "height-ratio," always less than 1, gives the ratio of the lower height to the higher height, of the two vertical sides of the trapezoid. The sum of the block probabilities must equal 1. This distribution allows the analyst to cast the distribution in terms such as "I think there's a 20-percent probability that the slip rate is greater than 0.75 mm/yr, tapering off linearly and going to zero at 1.5 mm/yr."

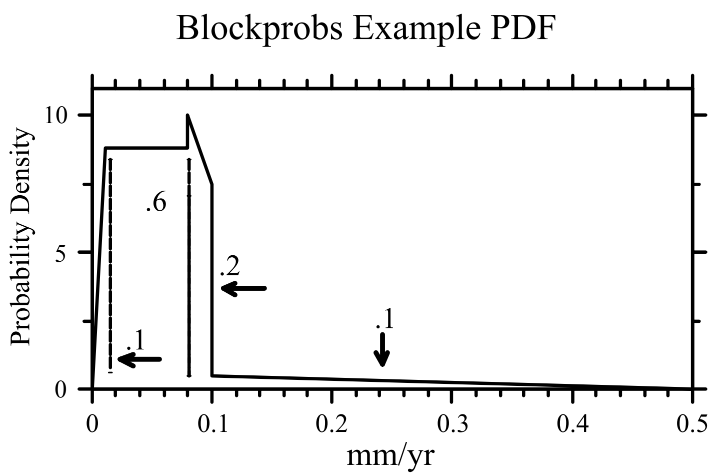


Figure 4.—Example blockprobs probability density function. Weights for each "block" are shown.

PSHA Auxiliary Programs

This section describes auxiliary software used for PSHA and general seismic hazard analysis. There are two main categories: programs used in logic tree procedures that usually operate on hazard curve files generated by the faultsource and *mrs* programs, and utility programs for general seismic hazard use, and for diagnosing faultsource input files.

Logic Tree Programs

addhazall—Sum Data from a Number of Files

These programs are used to sum multiple-source hazard curves from separate files. The number of sources must be the same in all files. addhaz expects just one period in each file, while addhazall expects multiple periods in each file. In this case, the period values and their order are checked for consistency. Note that the source name in the output file is the same as that of the first file. This can be changed with changefirstline, described below.

Usage:

```
addhazall << ! > hazdat.summed
        hazdat.file1
        hazdat.file2
!
```

changefirstline—Change the First Line of a File

When summing hazard curve files or sources, by default the seismic source name of the first file is used as the source name of the output file. Or you may want to change the name of a source. Program changefirstline replaces the first line of a file with "Cumulative" as default, or whatever you specify.

Usage:

changefirstline -t 'New Sourcename' < hazdat > hazdat.newtitle

copycum—Extract the Cumulative Hazard Curve(s)

Program copycum copies the cumulative hazard curve data to a new file. It simply looks for Cumulative as the source name and copies the rest of the file.

Usage:

copycum < hazdat > hazdat.cum

deletecum—Delete the Cumulative Hazard Curve(s)

Sometimes it is desirable to delete the cumulative curve, for example, if you want to append random seismicity hazard curves and compute new cumulative data. This is done with deletecum.

Usage:

deletecum < hazdat.all > hazdat.all.noncum

delete_other_versionlines—Deleting Superfluous Version Lines

When concatenating hazard curve files for several sources, the program version line is included. However, other programs expect only one version line (the first line) to exist. **delete_other_versionlines** removes the superfluous version lines.

Usage:

delete_other_versionlines < hazdat > hazdat.newfile

grace_hazcrv—Create a Grace Hazard Curve Plot

Xmgrace is a two-dimensional plotting program available for Unix/Linux operating systems. **grace_hazcrv** takes a single-period, multisource hazard curve file and generates a plot file that can opened in xmgrace. The plot can be saved in a number of formats, such as .ps, .eps, .jpg, and .png. A maximum of 50 sources can be plotted. For eight or fewer, a half-page plot is generated; for more than eight sources, the plot takes up about two-thirds of a page. The .agr suffix denotes an xmgrace file.

Usage:

grace_hazcrv -i infile [-t][-p period] [-x upper_xlimit] [-a y_lowlimit] [-b y_highlimit] [-f] [-t] > outfile.agr

Options:

-i infile, input file, a hazdat. file, one period, one or more sources.
-t means last source is not the Total, or cumulative hazard. Without, -t, the last source is assumed to be the "Total", and is given a heavier black line.
-p period, if frequency is not peak horizontal acceleration, type the period value, and the x -axis will be labeled correctly.
-x upper_xlimit, upper (right side) bound on x-axis.
-a y_lowlimit, lower bound on y-axis.
-b y_highlimit, upper bound on y-axis.
-f means input file is single period, single source, with fractiles. Mean and fractile curves are plotted.

grace_hazcrv -q gives options and defaults.

grace_relcon—Create a Grace Relative Contribution Plot

Xmgrace is a plotting program available on Unix/Linux operating systems. **grace_relcon** takes a single-period, multisource hazard curve file and generates a plot file that can opened in xmgrace. The plot can be saved in a number of formats, such as .ps, .eps, .jpg, and .png. The plot is two-dimensional with ground motion level on the x-axis and relative contribution (as percent) on the y-axis. A typical modification is to draw marks specifying the ground motion associated with specific return periods (see **gm_forretpd**). For most applications, this provides a useful source deaggregation for the spectral period. The .agr suffix denotes an xmgrace file.

Usage:
grace_relcon -i infile [-t threshold percentage][-p period] [-x upper_xlimit] > outfile.agr

Options:
-i infile, a file generated by **post_relcon**.
-t threshold percentage sources with contributions less than the threshold percentage are not plotted. Used for limiting the number of sources plotted.
-p period, if frequency is not peak horizontal acceleration, type the period value, and the x -axis will be labeled correctly.
-x upper_xlimit, upper (right side) bound on x-axis.

grace_relcon -q gives options and defaults.

hazdat_fracs—Integrate a Set of Hazard Curves

hazdat_fracs integrates a set of single-period, multisource hazard curves, to obtain mean and fractile curves. The source names and their order must be the same in all files. Default fractiles are 16th, 50th, and 84th. These can be changed in the program source code. Weights are placed in front of the file names, and must add up to 1. The total number of files and columns is set to 1,000 and 5, respectively; these can be changed in program source code.

Usage:

```
hazdat_fracs << ! > hazdat.fracs
        .33 hazdat.file1
        .33 hazdat.file2
        .34 hazdat.file2
!
```

hazmovavg—Smooth Hazard Curves

By the nature of the discrete integration procedure used by the generating programs, the hazard curve sometimes has a jagged appearance when plotted. hazmovavg applies a moving-window averaging technique to the data. The number of points used in the window can be supplied on the command line, with the -p option. The default, 11, has been found to give reasonable results. An odd number is required, since the window size tapers to 1 at both ends. In other words, the first point remains the same, the second is the average of points 1-3, the third point the average of points 1-5, etc. The value smoothed is $\log_{10}$ of the annual frequency of exceedance. This program is generally not needed by *faultsource_20+*, since the equal ground motion interval scheme implemented for discretizing ground motion distributions creates smooth curves out to the distribution truncation point; still useful for *mrs* programs.

Usage:

```
hazmovavg -p 9 < hazdat.bbf > hazdats.bbf
```

As a matter of convention, a hazdats prefix designates a smoothed file.

initialize_attfunc—Initialize the Attenuation Code in a Faultsource Input File

initialize_attfunc takes a faultsource input file and changes the attenuation code. For use in faultsource runscripts.

Usage:

```
initialize_attfunc attencode < faultsource_input.file > outfile
```

interp_allhazdat—Interpolate Period Sets

The program interp_allhazdat takes a multisource, multiperiod file and interpolates to the user-specified period set. Each attenuation function used for generating the hazard curves uses a different set of response periods. If we want to average or sum multiple-period files, they must contain the same period set. This program performs a linear interpolation of the annual frequencies of exceedance, to a specified period set. The set to be interpolated to is specified by the attenuation function code. A list of these can be obtained by typing faultsource_20 -l.

Usage:

```
interp_allhazdat -t ge97r < hazdat.as97 > hazdat.as97.int
```

This converts the periods in hazdat.as97 to the period set of the ge97r attenuation function.

post_asi—Generate Acceleration Spectrum Intensity (ASI) Hazard Curves

Acceleration Spectrum Intensity is defined as the area under the acceleration response spectrum curve between two period bounds. The bounding periods must exist in the input data. The output consists of the ASI value (first column) and the annual probability of excedance value (second column), spaced at set intervals. There is one such grouping for each column of input data. Multiple sources are allowed. The default period bounds are 0.1 and 0.5 seconds but can be changed on the command line. Type post_asi -q **to get usage and defaults.**

Usage:

```
post_asi -a 0.5 -b 1.0 < hazdat > asi.bbf
```

post_relcon—Create a Relative Contribution File

Sometimes you are interested in finding the relative contributions (expressed in percent of the total) of a each seismic source, at a particular response period. post_relcon reads a single-period hazard curve file with multiple sources, and generates a table with sources as columns and ground motion level as rows. The entries are the percent contribution of the total of the source at the ground motion level. Use **grace_relcon** to plot the relcon. file.

Usage:

```
post_relcon < hazdat.bbf > relcon.bbf
```

post_uhs—Generate Uniform Hazard Spectra

post_uhs extracts uniform hazard spectra (UHS) from a multiperiod file. The return period is specified on the the command line (default is 10,000 years). The -h option writes the return period as the first line of the output file. One UHS is computed for each column of data. Multiple sources are allowed. Type post_uhs -q **to get usage and defaults. Sometimes the data do not reach the return period. In this case, a default value of** 10.0 **is written in the output file.**

Usage:

```
post_uhs -r 5000 [-h] < hazdat > uhs.5k
```

scalehaz—Scaling a File

scalehaz scales the annual frequencies in a hazard curve file of any type.

Usage:

```
scalehaz 0.4 < hazdat > hazdat.scaled
```

This scales the annual frequency of exceedance in all columns by 40 percent.

strip_period—Extract a Single Period from a Hazard Curve File

Program strip_period extracts hazard curve data for a specified response period from a multisource multiperiod file.

Usage:

```
strip_period 0.075 < hazdat > hazdat.075
```

The period (0.075 s in this example) must exist in the file.

strip_source—Extracting a Single Seismic Source from a Hazard Curve File

Program strip_source will extract hazard curve data for a specified source.

Usage:

```
strip_source “Bigbad Fault” < hazdat > hazdat.bbf
```

If the source consists of more than one word, it must be enclosed in quotes.

sumhazdat—Sum Sources from One File

sumhazdat takes one hazard curve file with multiple sources and sums the annual frequencies. If the file has multiple periods, the number and sequence of periods in each source must be the same. The -a option causes the original data to be copied before the cumulative data are written; otherwise only the cumulative data are written.

Usage:

```
sumhazdat [-a] < hazdat.lotsafaults > hazdat.lotsafaults.cum
```

wtavg_ya —Weight-Averaging

Weight-averaging is used to average two or more hazard curve files. There are two versions; **wtavg_y** expects only one response period per source, while **wtavg_ya** expects more than one response period per source. Both programs check to see that the seismic source names in each file to be averaged are exactly the same, are in the same order, and have the same order of response periods. The **-n** option negates the name checking.

Usage:

```
wtavg_ya [-n] <<!> hazdat.avg.outfile
    .50 hazdat.file1
    .50 hazdat.file2
!
```

General Seismic Hazard Programs

This category includes diagnostic programs for faultsource input and fault coordinate files, and utility programs for general seismic hazard use.

asiresp—Compute Acceleration Spectum Intensity (ASI) and Velocity Spectrum Intensity (VSI) from a Response Spectrum File

asiresp computes ASI and VSI from a response spectrum (UHS or otherwise) file. Any number of data columns may be present. Default bounds are 0.1 and 0.5 second for ASI, and 1.0 and 2.5 seconds for VSI. These may be changed on the command line. There is also a flag for when the ground motion is in g's instead of cm/s/s. Type asiresp -q for details.

Usage:

```
asiresp [-a lowASIbound -b highASIbound -c lowVSIbound -d
highVSIbound -g (input in g's) ] < uhs.7k
```

avdisp2mag—Compute Magnitude from Average Surface Displacement

avdisp2mag gives magnitude from average surface displacement, using Wells and Coppersmith (1994) relations.

Type **avdisp2mag -q** for usage and options.

diff_uhs—Computing the Difference Between Two UHS Files

diff_uhs computes the difference between two uniform hazard response spectrum files. The number of columns and period set in each file must be the same. The difference is computed as file 2 minus file 1.

Usage:

```
diff_uhs << ! > uhs.diff
    uhs.7k
    uhs.5k
!
```

fault_intersects—Find Intersection Points between Faults

fault_intersects reads a faultsource-formatted fault coordinate file and finds the intersection points between the first fault in the file and subsequent fault. The faults are densely gridded, and intersecting points defined by a proximity threshold. These points can then be plotted in ARC or GMT.

Usage:

fault_intersects [-t target_distance -s grid_spacing] < fault_coordinate_file

`Options:`

`-t` target_distance distance threshold; alter if too many or too few points detected.

-s grid spacing, fault is gridded at 0.15 km spacing for distance measurements by default; can be changed here.

Files produced:

intersect_pts_xy.dat: longitude and latitude of intersection points
intersect_pts_xyz.dat: longitude, latitude, and depth of intersection points
firstfault_pts.dat: grid points of first fault on list
Other diagnostic files as listed for **fault_params**

faultlengths—Compute Fault Lengths from a Fault Coordinate File

faultlengths reads a faultsource-formatted fault coordinate file and computes the surface length of each fault.

Usage:

faultlengths < fault_coordinate_file

fault_params—Diagnose a faultsource Fault Coordinate File

A faultsource fault coordinate can contain errors due to digitization errors or poorly characterized fault geometry. **fault_params** diagnoses each fault in the fault coordinate file, computes some useful measurements, and checks for a number of problem conditions. See *Use of* fault_params, page 5, for details.

Usage:

`fault_params -a sitelat -b sitelon` [-f area-magnitude code] [-s grid spacing] < faultcoord.file > fparams.out

`Options:`

`-a sitelat -b sitelon` required; site latitude and longitude, for measuring distances from fault to site.

`-f` area-magnitude code overrides fault type in fault coordinate file. By default, Wells and Coppersmith (1994) relations are used.

-s grid spacing, fault is gridded at 0.1-km spacing for distance measurements by default. Can be changed here; for example, for large subduction zone faults, greater spacing is appropriate.

Diagnostic files produced:

map_coords.txt: three-dimensional fault outlines, and lines connecting segments
fault_params_Basecoords.txt: diagnostic information for downdip coordinate calculation
fault_params_fillseq.txt: diagnostic information grid point generation
fault_params_patchdiags.txt: diagnostic information grid point generation
faultparam_data.txt: tab-delimited file with various fault measurements; can be opened as a table in Word or Framemaker

`fault_params -q` gives options and defaults.

getgm—Obtain Ground Motions from an Attenuation Function

getgm gives ground motion values for a given magnitude and distance for an attenuation function. Mean, median and a fractile (84th is default) are shown.

Usage:

`getgm` [-t attencode][-r hypdist -e epdist -h depth] [-m magnitude -f Faulttype -s -n period -p -v]

`Options:`

`-t` attencode
`-f Faulttype,` s,n,t,r,a, or o, for strike-slip, normal, thrust (t or r), "all", and oblique
`-s,` sigmas only
`-n period,` give results for this period only
-v SiteVelocity, required for jb97 and jb97x attenuation codes
-p fractile, for fractile ground motion (e.g., .95)

`getgm -q` gives options and defaults.
`getgm -l` gives available attenuation function codes.

gmforretpd—Obtain Ground Motions for Specific Return Periods

gmforretpd reads a single period hazard curve file and finds the ground motion for a range of return periods. If a multisource file, gmforretpd looks for the last, or "Cumulative" source. Number of columns (4; mean and 3 fractiles) and set of return periods can be changed in source code.

Usage:

`gm_forretpd` < hazdat.file > outfile

grace_manymean_hazcrv—Create a Grace Hazard Curve Plot, Plot Each Column

grace_manymean_hazcrv takes a hazdat file with many columns and creates a Grace .agr file, which will produce hazard curves using *xmgrace*. Assumptions:

file is one source for one period, but with NCOLS (set in source code) columns, each representing a mean curve. Each column is plotted as solid line curve. Useful for comparing a set of mean curves.

Usage:

`grace_manymean_hazcrv` -i infile [-t][-p period] [-x upper_xlimit] [-a y_lowlimit] [-b y_highlimit] > outfile.agr

`Options:`

`-i infile,` input file, a hazdat. file, one period, one or more sources.
`-t` means last source is not the Total, or cumulative hazard. Without, -t, the last source is assumed to be the "Total," and is given a heavier black line.
`-p period,` if frequency is not peak horizontal acceleration, type the period value, and the x -axis will be labeled correctly.
-x upper_xlimit, upper (right side) bound on x-axis.
-a y_lowlimit, lower bound on y-axis.
-b y_highlimit, upper bound on y-axis.
-f means input file is single period, single source, with fractiles. Mean and fractile curves are plotted.

mag2ra—Compute Magnitude from Fault Area

mag2ra gives magnitude from rupture area, from empirical regressions.

Usage:

`mag2ra` [-m magnitude -f fault_type]

Type **mag2ra -q** for usage and options.

makefsfiles—generate a runscript from a *faultsource_20* Input File

makefsfiles creates a project-specific runscript template from a *faultsource_20* input file. Generated file names contain project_name.

Usage:

`makefsfiles` [-t project_name –l] < input_file

Type **makefsfiles -q** for usage and options.
Type **makefsfiles –l** for a list of available attenuation functions.

maxdisp2mag—Compute Magnitude from Maximum Surface Displacement

maxdisp2mag gives magnitude from maximum surface displacement, using Wells and Coppersmith (1994) relations.

Type **maxdisp2mag -q** for usage and options.

ra2mag—Compute Fault Area from Magnitude

ra2mag gives rupture area from magnitude, from empirical regressions.

Usage:

ra2mag [-a fault_area (km^2) -f fault_type]

Type **ra2mag -q** for usage and options.

rate_params—Create a Faultsource Activity Rate Table

rate_params takes a faultsource input file and produces a tab-delimited file with fault name and rate distribution parameters. The file can be opened as a table in Word or Framemaker, checked for accuracy, and put into a report. Should be done before breaking up the input file into separate sources.

Usage:

rate_params < faultsource_input.file

The table file name is **faultrate_table_data.txt**.

ratio_uhs—Compute the Ratio of Two UHS Files

ratio_uhs computes the ratio of two uniform hazard response spectrum files. The number of columns and period set in each file must be the same. The ratio is computed as file 2 divided by file 1.

Usage:

```
ratio_uhs << ! > ratio.out
    uhs.7k
    uhs.5k
!
```

sliptomag—Convert from Slip Rate, Magnitude and Fault Area to Annual Rate and Return Period

sliptomag calculates occurrence rate (and return period) for a given magnitude and slip rate. After Molnar (1979), Hanks and Kanamori (1979), and Wells and Coppersmith (1994) for fault areas.

Usage:

sliptomag [-m Mw -u shear modulus -s slip rate (mm/yr) -f fault_type -a faultarea]

For fault type (**-f**), s = strike slip, n = normal, r = reverse, a = unknown [Wells and Coppersmith (1994)]
z = subduction zone (Geomatrix, 1993), p = PEER

Gives rate for the rupture area and the fault area.

sliptomag **-q** gives options and defaults.

srl2mag—Compute Magnitude from Surface Rupture Length

srl2mag gives magnitude from surface rupture length, from empirical regressions.

Usage:

srl2mag -l length (km) [-f fault_type]

Type **srl2mag -q** for usage and options.

szra2mag—Compute Subduction Zone Magnitude from Fault Area

szra2mag gives subduction zone interface rupture area from magnitude, from Geomatrix (1993).

Usage:

szra2mag [-a fault_area (km^2)]

wtuhs—Weight-Average UHS Files

wtuhs weight averages a number of UHS files. The number of columns in each file must be the same, and the weights must add up to 1.

Usage:

```
wtuhs << ! > uhs.avg
        .33 uhs.as97
        .33 uhs.ge93
        .34 uhs.jb88
!
```

References

Abrahamson, N.A., and W.J. Silva. 1997. "Empirical response spectral attenuation relations for shallow crustal earthquakes." *Seismological Research Letters*. 68, 94-128.

Anderson, J.G. 1979. "Estimating the seismicity from geological structure for seismic-risk studies." *Bulletin of the Seismological Society of America*. 69, 135-159.

Boore, D.M., W.B. Joyner, and T.E. Fumal. 1997. "Equations for estimating horizontal response spectra and peak acceleration from western North American earthquakes: A summary of recent work." *Seismological Research Letters*. 68, 128-154.

Cornell, C.A. 1968. "Engineering seismic risk analysis." *Bulletin of the Seismological Society of America*. 58, 1583-1606.

Geomatrix. 1993. *Seismic margin earthquake for the Trojan site*. Final unpublished report prepared for Portland General Electric Trojan Nuclear Plant, Rainier, Oregon. Geomatrix Consultants. San Francisco, California.

Hanks, T.C., and H. Kanamori. 1979. "A moment magnitude scale. *Journal of Geophysical Research*. 84, B5, 2348-2350.

LaForge, R. 2001. *mrs programs for site-specific probabilistic seismic hazard analysis for zones for random seismicity*. Technical Memorandum D8330-2001-13. Bureau of Reclamation. Denver, Colorado. 27 pp.

McGuire, R.K. 2004. *Seismic hazard and risk analysis*. Earthquake Engineering Research Institute MNO-10. 221 pp.

Molnar, P. 1979. "Earthquake recurrence intervals and plate tectonics." *Bulletin of the Seismological Society of America*. 79, 115-133.

Newham, C., and B. Rosenblatt. 1998. *Learning the bash Shell*. 2nd edition. O'Reilly and Associates. 318 pp.

Sadigh, K., D.Y. Chang, J.A. Egan, F. Makdisi, and R.R. Youngs. 1997. "Attenuation relationships for shallow crustal earthquakes based on California strong motion data." *Seismological Research Letters*. 68, 74-86.

Schwartz, D., and K. Coppersmith. 1984. "Fault behavior and characteristic earthquakes: examples from the Wasatch and San Andreas faults." *Journal of Geophysical Research*. 89, 5681-5698.

Somerville, P.G., K. Irikura, R. Graves, S. Sawada, D. Wald, N. Abrahamson, Y. Iwasaki, T. Kagawa, N.Smith, and A. Kowada. 1999. "Characterizing crustal earthquake slip models for the prediction of strong ground motion. *Seismological Research Letters*. 70, 1, 59-80.

Spudich, P., W.B. Joyner, A.G. Lindh, D.M. Boore, B.M. Margaris, and J.B. Fletcher. 1999. "A revised ground motion prediction relation for use in extensional tectonic regimes. *Bulletin of the Seismological Society of America*. 89, 1156-1170.

WGCEP. 1995. "Seismic hazards in southern Californa: Probable earthquakes, 1994-2024." *Bulletin of the Seismological Society of America*. 85, 379-440.

Wells, D.L. and K.J. Coppersmith. 1994. "New empirical relationships among magnitude, rupture length, rupture width, rupture area, and surface displacement. *Bulletin of the Seismological Society of America*. 84, 974-1003.

Wesnousky, S.G. 1986. "Earthquakes, Quaternary faults, and seismic hazard in California. *Journal of Geophysical Research*. 91, No. B-12, 12587-12631.

Youngs, R.R., and K.J. Coppersmith. 1985. "Implications of fault slip rates and earthquake recurrence models to probabilistic seismic hazard estimates. *Bulletin of the Seismological Society of America*. 75, 939-965.

Youngs, R.R., S.J. Chiou, W.J. Silva, and J.R. Humphrey. 1997. "Strong ground motion attenuation relationships for subduction zone earthquakes. *Seismological Research Letters*. 68, 58-74.