SEISRISK III: A Computer Program for Seismic Hazard Estimation

U.S. GEOLOGICAL SURVEY BULLETIN 1772

SEISRISK III: A Computer Program for Seismic Hazard Estimation

By BERNICE BENDER and DAVID M. PERKINS

U.S. GEOLOGICAL SURVEY BULLETIN 1772

DEPARTMENT OF THE INTERIOR
DONALD PAUL HODEL, Secretary

U.S. GEOLOGICAL SURVEY
Dallas L. Peck, Director

UNITED STATES GOVERNMENT PRINTING OFFICE, WASHINGTON: 1987

For sale by the
Books and Open-File Reports Section
U.S. Geological Survey
Federal Center
Box 25425
Denver, CO 80225

CONTENTS

FIGURES

SEISRISK III:
A Computer Program for Seismic Hazard Estimation

By Bernice Bender *and* David M. Perkins

SUMMARY

SEISRISK III is a revision of SEISRISK II, a computer program for seismic hazard analysis, described in Open File Report 82-293. Specifically, SEISRISK II and SEISRISK III were designed to compute maximum ground-motion levels that have a specified probability of not being exceeded during fixed time periods at each of a set of sites uniformly spaced on a two-dimensional grid. The principal differences between SEISRISK II and SEISRISK III are:

1. SEISRISK II assumes that seismicity within a seismic source zone is uniform; that is, each point within a source zone has the same probability of being the epicenter of a future earthquake. This assumption means that the projected rate of earthquakes changes abruptly at a source zone boundary; an effect of such a change in seismicity is that calculated probabilistic ground-motion levels may differ substantially at sites a few kilometers apart near a boundary. SEISRISK III retains the concept of seismic source zones but allows earthquakes within a zone to be normally rather than uniformly distributed. This allows some of the earthquakes that previously would have occurred within the zone to occur outside the zone, permitting seismicity to vary smoothly across the boundaries of the zone. The result is that calculated acceleration levels also vary more smoothly at sites near a boundary.
2. SEISRISK III does a partial "magnitude smoothing" that treats the "closest distance" ruptures as if they occurred over a range of magnitudes. In the fault-rupture model employed, ground motion (for example, acceleration) at a site resulting from a rupture along a fault is regarded as a function of closest site-to-rupture distance. This means that for the longer ruptures associated with higher magnitude earthquakes, a substantial fraction of the possible ruptures of a given magnitude along the fault may be at the same (closest) distance from the site, and the same ground-motion level will be calculated for all of these ruptures. Because the computations are performed for a discrete set of magnitudes, the ground-motion levels produced by the ruptures at closest distance for each of these magnitudes will be calculated as occurring more frequently than other ground-motion levels. This has the effect that the calculated probability of exceeding a specified ground-motion level does not vary smoothly as a function of ground-motion level. Increasing the number of magnitudes at which the calculations are performed reduces these irregularities but increases the computation time. By treating the closest distance ruptures for each magnitude as if they occurred over a range of magnitudes, SEISRISK III smooths the acceleration densities resulting from these ruptures.
3. The fault pattern may be quite complex within an active fault zone, and faults may be spread over a wide area. In both SEISRISK II and SEISRISK III, "artificial" parallel faults may be modeled when the presence of faulting is inferred from geologic considerations, and the orientation of the faults is assumed to be known. If a series of equidistant parallel faults are input into the program, SEISRISK III can do a partial "distance smoothing" to simulate a finer spacing between faults in order to better approximate a uniform distribution. (SEISRISK II did not permit this smoothing.)
4. Both SEISRISK II and SEISRISK III calculate probabilistic ground motions at all sites resulting from earthquakes in a single source, before proceeding to the next source. This requires retaining intermediate calculations for each site to accumulate ground-motions from successive sources. SEISRISK II wrote these intermediate results onto a disk; SEISRISK III saves the results in two-dimensional arrays in memory. Currently, the program requires two arrays dimensioned $(56,N)$, where N=number of sites on grid of sites for which calculations are performed.

INTRODUCTION

In most seismic hazard analyses, earthquakes are modeled as

1. points located randomly within seismically homogeneous areal source zones; and (or)
2. finite length ruptures that occur randomly along linear fault segments.

Probabilistic models are used because sizes and locations of future earthquakes cannot be predicted precisely. Source zones are typically regarded as seismically homogeneous, that is, each point within a source zone is assumed to have the same probability of being the epicenter of a future earthquake, because the analyst has

no reason to assume that an earthquake is more likely to occur at one point than at another within the zone.

The analyst uses catalogs of recorded earthquakes, and whatever geologic, geophysical or other information is available to help him to determine seismic source zones. However, as Thenhaus (1983, p.31) noted, "The procedures used in delineating seismic source zones are ill defined. No single standard exists by which source zones across the nation can be drawn, primarily because of the nonuniform level of pertinent seismological, geological and geophysical information available for areas of vastly differing tectonic and geologic settings. The equivocal association of seismicity with geologic structure throughout most of the United States compounds the problem." Because the evaluation and interpretation of the various types of information available depends strongly on individual judgment or opinion, different experts may define source zones differently.

Assuming that seismicity is uniform within a source zone means that seismicity changes abruptly at each source-zone boundary. An effect of these abrupt changes in seismicity is that probabilistic ground-motion levels calculated at sites a few kilometers apart near a boundary may differ significantly. Bender (1986) noted changes of 50-80 percent or more in acceleration levels calculated for longer exposure times at sites 20 km apart near a boundary of a source zone. Such changes seem unreasonable if, in fact, source zones are not well determined.

To avoid sudden changes in seismicity at source-zone boundaries, SEISRISK III, unlike SEISRISK II, permits an option of "earthquake location uncertainty". Source-zone boundaries are specified in SEISRISK III as in SEISRISK II, but now each point within the zone is regarded as the mean or most likely location of a future earthquake; the locations of actual earthquakes are normally distributed with standard deviation σ about their mean locations (σ is an input parameter). This permits earthquake rates to vary smoothly near a boundary, without being significantly different in the center of the region from the rates assumed in the uniform-seismicity model (fig.1). As σ increases, a higher percentage of the earthquakes that were expected to occur within the source zone when no location uncertainty (σ=0) is assumed, occur instead beyond the boundaries.

In the following sections, we discuss in more detail earthquake location uncertainty and how this uncertainty is implemented in SEISRISK III. We also discuss "magnitude smoothing" and "distance smoothing" for ground-motions resulting from ruptures along a fault, and the parameters that control this smoothing in the computer program. A listing of SEISRISK III (Appendix C) and information regarding data input and formats (Appendix B) are also included.

TERMINOLOGY

In this report, we use acceleration as a generic measure of ground shaking; we could equally well have used velocity or any other measure. By acceleration occurrences in an interval $a_{j-1} < a \leq a_j$ we mean the average annual occurrence rate of accelerations in the range $a_{j-1} < a \leq a_j$ experienced at a specific site. Acceleration levels or probabilistic acceleration levels refer to the accelerations that are calculated as having a specified probability of not being exceeded during a given time period at particular sites.

MODELING EARTHQUAKE LOCATION UNCERTAINTY

In the model implemented in SEISRISK III, the seismicity associated with each point in a source zone is regarded as normally distributed about that point. The possible earthquakes, rather than being uniformly distributed throughout a source region, have mean locations that are uniformly distributed, with scatter about each mean location given by a circular-normal distribution. Stated more precisely, in this model, if σ is the standard deviation in location, and if the expected (mean) location of an earthquake is at (X_e, Y_e), the probability that the earthquake will occur instead in a small area A surrounding $(X_e+\Delta x, Y_e+\Delta y)$ is given by

$$p_A(\Delta x, \Delta y) = \frac{A}{2\pi\sigma^2} \exp\left(-\frac{\Delta x^2+\Delta y^2}{2\sigma^2}\right). \qquad (1)$$

Figure 2 (*A–C*) shows acceleration levels calculated as having a 90-percent probability of not being exceeded during exposure times of 10, 50, 250, and 1000 years when no variability in earthquake location ($\sigma = 0$) is assumed, and when $\sigma = 10$ and 20 km (the attenuation function of Joyner and Boore (1981) with $\sigma_a = 0.5$ in $\log_e$ acceleration was used). The sites are located 5 km apart on a line through the centers of two contiguous source zones. Each zone is a square 60 km wide; the second zone has an earthquake rate 10 times higher than the first. Unless σ is a substantial fraction of the length or width of the source zone (for example, $\sigma = 20$ km in fig.2), the acceleration levels calculated at sites near the center and at sites on the boundaries of the source zone are nearly the same for various values of σ. For a site outside the source, if $\sigma \neq 0$ is assumed, some of the earthquakes will occur closer to the site than when $\sigma = 0$ is assumed, and these closer earthquakes will produce higher accelerations at the site than will any earthquakes of the same magnitude occurring within the fixed source zone.

Figure 2*D* shows the difference in acceleration level calculated for the 1000-year exposure time at sites 5

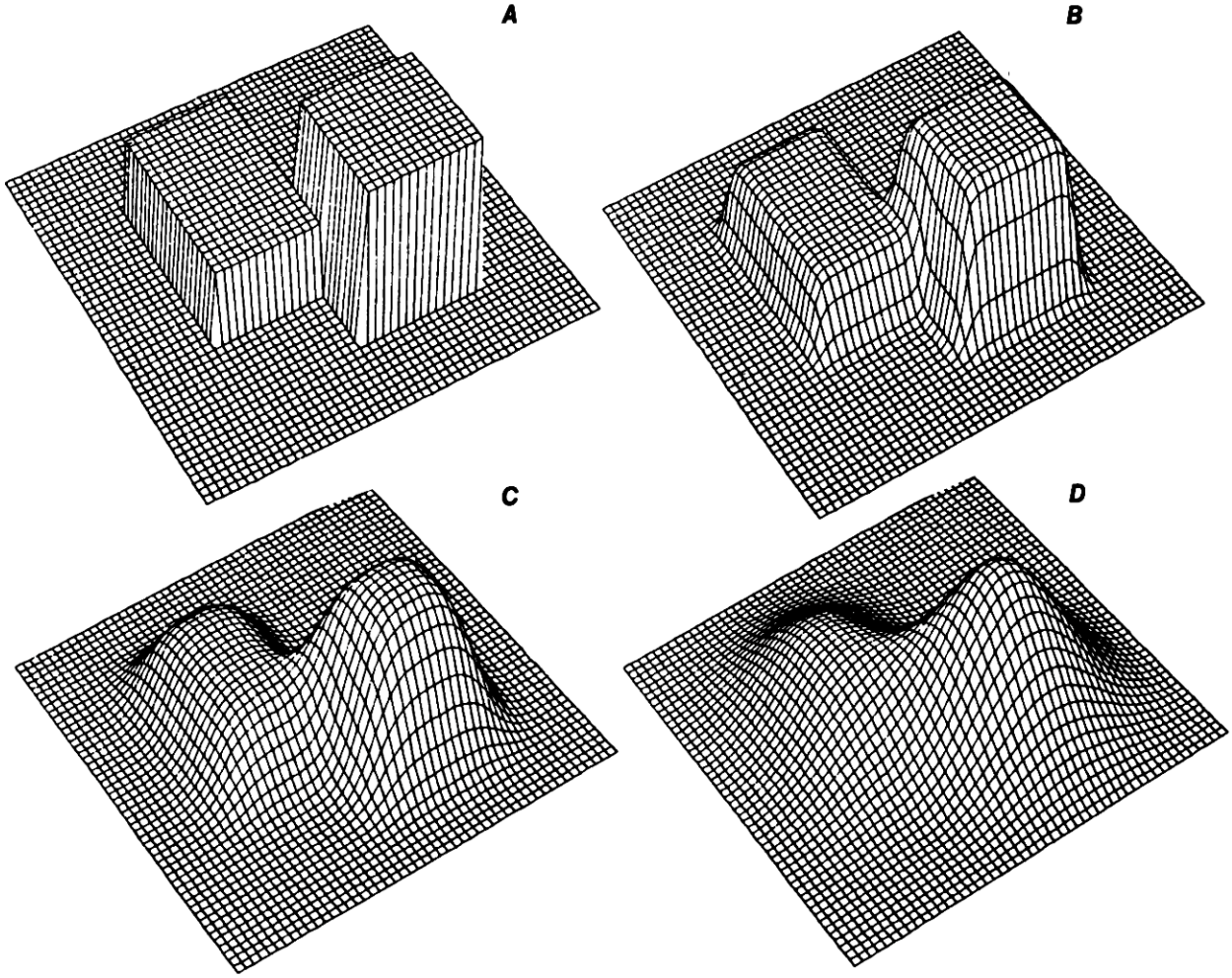


Figure 1. Earthquake rates within gridded areas (10 km x 10 km). In *A* earthquakes occur only within two well-defined seismically homogeneous source zones. In *B*, earthquake locations are normally distributed with standard deviation $\sigma = 10$ km; in *C*, $\sigma = 25$ km; in *D*, $\sigma = 50$ km.

km apart on the line when $\sigma = 0$, 10, and 20 km; for example, a change in acceleration of +0.05 g at the site at X km means that the acceleration level calculated at the site at X+5 km is 0.05 g higher than the level calculated at X.

To explain how acceleration levels are calculated in SEISRISK III when earthquake-location variability is taken into account, we first observe that, if earthquakes are normally distributed, those that occur at distance $(\Delta x, \Delta y)$ from their mean locations, in effect, create a new source zone translated by $(\Delta x, \Delta y)$ from the original zone. This means that the possible locations of the source zone (rather than of the earthquakes) may be regarded as having a normal distribution. More explicitly, the source zone is said to be located at $(\Delta x, \Delta y)$ if an arbitrary point in the zone that would have been at (X_b, Y_b) if the zone were at its mean or most likely location, lies instead at $(X_b + \Delta x, Y_b + \Delta y)$. The probability that the zone will be located at $(\Delta x, \Delta y)$, is proportional to $\exp(-\frac{\Delta x^2 + \Delta y^2}{2\sigma^2})$, where σ is now the standard deviation in source location. Furthermore, the acceleration occurrences in an interval $a_{j-1} < a \leq a_j$ that are calculated for a site at (X, Y) when the source location is moved to $(\Delta x, \Delta y)$ are the same as those calculated for a site at $(X - \Delta x, Y - \Delta y)$ when the source is at its original location. This means that the effect of source-zone location uncertainty at a site at (X, Y) can be produced by computing a weighted average of the acceleration occurrences at various sites, calculated using the original seismically homogeneous source zone. This last fact was used to implement earthquake location (or, equivalently, source-zone location) uncertainty in SEISRISK III as follows.

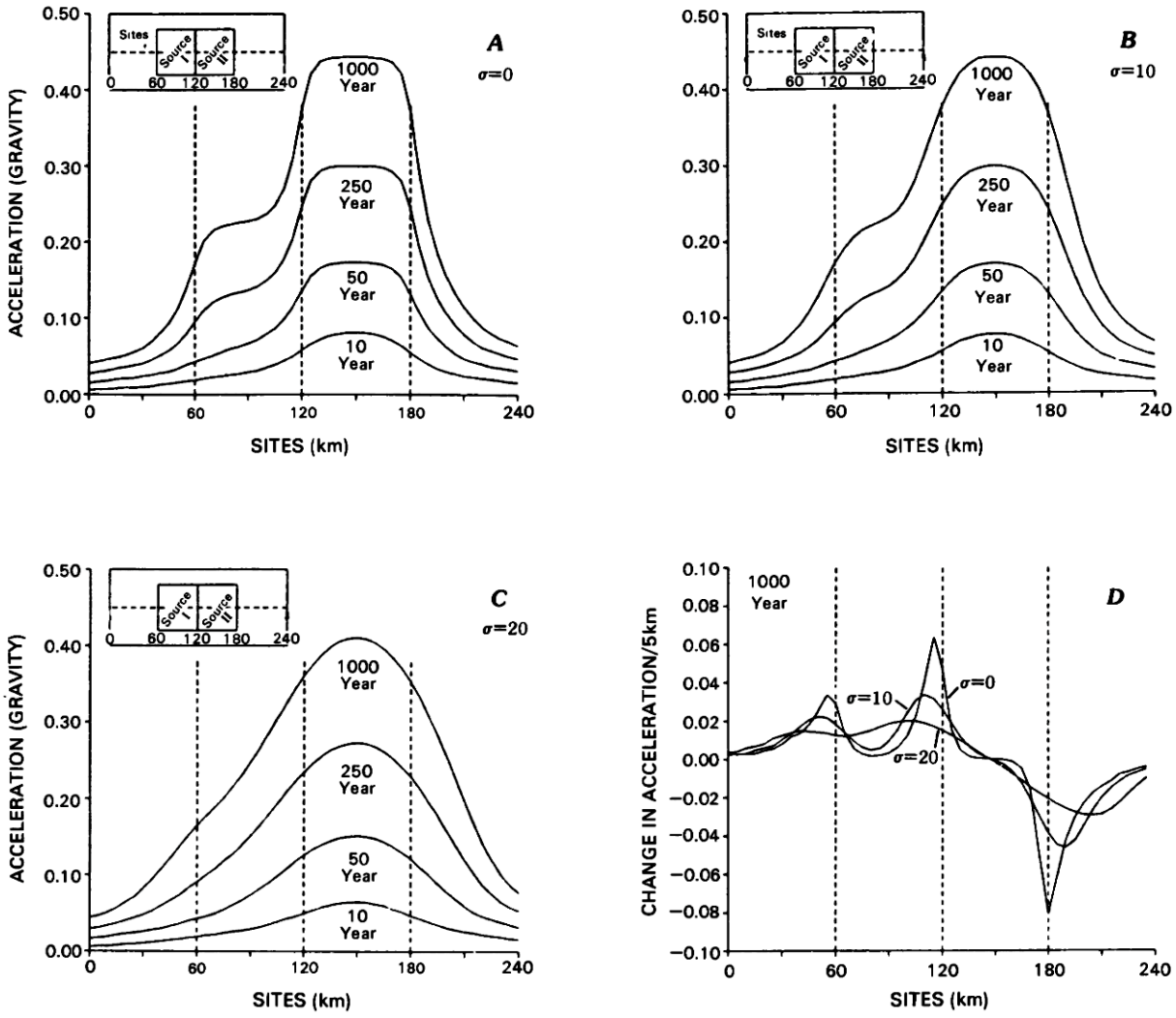


Figure 2. (A)-(C): Accelerations calculated to have a 90 percent probability of not being exceeded during various time periods, at sites on a line through the centers of two seismic sources. Source II has an earthquake rate 10 times higher than Source I. Earthquake locations are normally distributed with A, σ=0 km; B, σ=10 km; C, σ=20 km. D, Differences in acceleration levels calculated above for 1000-yr period at sites 5 km apart on the line of sites.

SEISRISK III begins, as does SEISRISK II, by calculating expected annual accelerations at each site on a grid of sites, resulting from earthquakes in well-defined, seismically homogeneous zones. Sites are spaced at increments $dx, dy = dx$. The acceleration occurrences at a site at (X_1, Y_1) that result when earthquake location variability is taken into account is a weighted sum of the acceleration occurrences that were calculated at each point on the grid, assuming homogeneous source zones. The weight assigned to acceleration occurrences in the interval $a_{j-1} < a \leq a_j$ at the point at $(X_1 + k_1 dx, Y_1 + k_2 dy)$ is proportional to

$$w(k_1, k_2) = \frac{dx dy}{2\pi\sigma^2} \exp\left(-\frac{k_1^2 dx^2 + k_2^2 dy^2}{2\sigma^2}\right). \qquad (2)$$

The occcurrences in the interval $a_{j-1} < a \leq a_j$ that result when earthquake location variability is taken into account is given by

$$D = \frac{\sum_{k_1} \sum_{k_2} w(k_1, k_2) \rho[(X_1 + k_1 dx, Y_1 + k_2 dy), j]}{\sum_{k_1} \sum_{k_2} w(k_1, k_2)} \qquad (3)$$

where $\rho[(X_1 + k_1 dx, Y_1 + k_2 dy), j]$=the acceleration occurrences calculated in the interval $a_{j-1} < a \leq a_j$ at the site at $X_1 + k_1 dx, Y_1 + k_2 dy$ assuming no earthquake location variability (i.e., homogeneous source zones).

SEISMIC HAZARD MODEL FOR HOMOGENEOUS SOURCES

To determine acceleration rates in SEISRISK III that result when earthquake location variability is taken into account, we first compute acceleration rates, as in SEISRISK II, assuming that earthquakes occur randomly within seismically homogeneous source zones or along linear fault segments. In this section, we briefly describe the model before incorporating earthquake location uncertainty.

Earthquake occurrences are assumed to have a Poisson distribution, and rates that remain constant during the time periods of interest. Mean or median ground-motion from an earthquake is an increasing function of magnitude and decreasing function of site-to-source distance. Earthquakes occur as points within quadrilateral source zones or as finite-length ruptures along linear fault segments. Ruptures of a given length may, with equal likelihood, be centered at any point on the fault for which neither end of the rupture would extend beyond the end of the fault.

Earthquakes in a given zone or fault are restricted to occur within a specified magnitude range; the range may be different for different zones and faults. More specifically, for each zone (and fault) a maximum magnitude m_{max}, a minimum magnitude m_0, and a magnitude interval Δm are assumed. All magnitudes occurring within a magnitude interval are grouped at the center of the interval, that is, at a set of n distinct magnitudes m_j:

$$m_j = m_0 + (j+1/2)\Delta m, \quad 0 \le j \le n-1,$$
$$\Delta m = \frac{m_{max} - m_0}{n}. \tag{4}$$

The rate of earthquakes per unit time in the zone (or on the fault) in each magnitude interval is specified by the program user; it frequently follows (but is not required to follow) a relationship of the form

$$\log N_m = a - bm \tag{5}$$

where a and b are constants. If $N_{m(j)}$ is the total number of earthquakes in the j^{th} magnitude interval, the fractional occurrences expected in any small subarea ΔA is

$$\frac{\Delta A}{A} N_{m(j)} \tag{6}$$

where A =total area of zone.

Seismicity is assumed to remain constant during the time periods being considered; that is, the average rate of earthquakes per unit time for each magnitude interval does not change with time.

A table of median (or mean) ground-motion values (for instance, acceleration) for the desired attenuation function is entered into the program for a set of earthquake magnitudes and distances. The program assumes that ground-motion increases with magnitude and decreases with site-to-source distance; ground motions for various magnitudes and distances are obtained as needed by interpolating values in the table.

Given the source zones, earthquake rates and attenuation function, an average yearly rate of accelerations within each of 55 acceleration intervals is constructed for each site. (In SEISRISK II 100 intervals were used; the number was decreased to save time and space, but can be adjusted within the program.) Acceleration occurrences in the range $a_{i-1} < a \le a_i$ are accumulated into $reg(i)$, the accumulator for the i^{th} interval, as described below.

For the given attenuation function,, let $d_m(i)$ be the distance at which a magnitude m earthquake produces median acceleration a_i at a site. The probability that an acceleration in the range $a_{i-1} < a \le a_i$ occurs at the site is the probability that a magnitude m earthquake occurs at a distance $d_m(i) \le d < d_m(i-1)$ integrated (summed) over all permissible magnitudes. (Note $d_m(i) \le d_m(i-1)$ because shorter distances produce higher accelerations.) The probability that an earthquake of magnitude m occurring within the zone (or on the fault) occurs within the distance interval $d_m(i) \le d < d_m(i-1)$ is equivalent to the fraction of the area of the zone (or length of the fault) that lies within this interval. The area (or length) within the interval

$$A[m(i)] = f[\bar{d}_m(i), \Delta d_m(i)], \tag{7}$$

is a function of interval width

$$\Delta d(i) = d_m(i-1) - d_m(i) \tag{8}$$

and interval midpoint

$$\bar{d}_m(i) = \frac{d_m(i-1) + d_m(i)}{2}. \tag{9}$$

The rate of accelerations in the range $a_{i-1} < a \le a_i$ at the site resulting from magnitude m earthquakes within the zone then is equal to the rate of magnitude m earthquakes within $A[m(i)]$:

$$f_m(i) = \frac{A[m(i)]}{A} N_m \tag{10}$$

where N_m=rate of magnitude m earthquakes for the entire zone or fault
A=total area of zone (or total fault length).

The program determines $A[m(i)]$ for each magnitude and adds the corresponding rate $\rho_m(i)$ into the i^{th} accumulator, $reg(i)$. The total rate accumulated for accelerations in the range $a_{i-1} < a \leq a_i$ from one source zone then is the sum over magnitude of

$$\rho(i) = \sum_{j=1}^{n} \rho_{m_j}(i) \qquad (11)$$

where n=total number of magnitude intervals.
The process is repeated for all $1 \leq i \leq 55$, for all sources.

These calculations assume a single value of ground motion (acceleration) for earthquakes of a given magnitude and distance. Calculations that take into account ground motion variability will be discussed later. First, we shall consider the calculation of (median) accelerations resulting from ruptures along faults and some ways of smoothing the calculated acceleration densities.

THE FAULT-RUPTURE MODEL

As previously noted, both SEISRISK II and SEISRISK III allow earthquakes to be modeled as finite-length ruptures along linear fault segments. In both programs, acceleration occurrences resulting from fault ruptures are added to those resulting from earthquakes in source areas. However, unlike SEISRISK II, SEISRISK III permits a partial "smoothing" of acceleration occurrences at a site caused by ruptures along a fault. We shall discuss the fault-rupture model and the changes that have been made to smooth those occurrences that result from fault ruptures.

In the fault-rupture model, sets of one or more faults are defined as fault zones; if a fault zone contains more than one fault, earthquakes in the zone are distributed among the faults in proportion to the lengths of the faults. An individual fault of length L may consist of as many as 24 connected straight-line segments. (The maximum number of segments may be altered within the program.) A rupture may span several segments (fig.3). Ruptures occur with equal likelihood anywhere along the fault, so long as they are wholly contained within the fault. Rupture lengths are a function of magnitude and are lognormally distributed with median length $l(m)$ and standard deviation σ_l.

As with source areas, the ground-motion at a site is a function of earthquake magnitude and distance. However, in the case of faults, distance from the site to a rupture is always taken to be the distance to the point on the rupture closest to the site. (The fault-rupture model is described in detail in Bender, 1984b.) All magnitude m ruptures along the fault whose closest distance

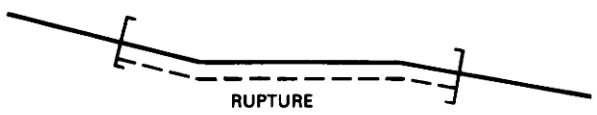


Figure 3. Fault consisting of three connected straight-line segments. A rupture may span several segments. Closest distance from rupture to site is used to determine ground motion at the site.

d to the site is in the range $d_m(j) \leq d < d_m(j-1)$ will produce acceleration a, $a_{j-1} < a \leq a_j$, at the site. The calculations to determine site-to-rupture distance and the fraction of ruptures of each magnitude within each distance range are described in Appendix A.

Smoothing of Accelerations Calculated Using Fault Ruptures

In the fault-rupture model, a substantial fraction of the ruptures of earthquakes of a given magnitude may occur at the same closest distance from the site and may produce the same median acceleration at the site. For example, assume the fault extends on the x-axis from $0 \leq x \leq L$. A rupture of length $L/4$ may be centered at any point x along the fault such that $1/8L < x \leq 7/8L$. If the site is located on a line perpendicular to the center of the fault at $(L/2, P)$, all ruptures that have their centers in the range $1/4L \leq x \leq 3/4L$ along the fault will be at the same (closest) distance P from the site. This means that two-thirds of the possible ruptures of length $L/4$ will be at distance P from the site; for rupture length $L/2$, all possible ruptures will be at distance P from the site. Because longer ruptures are associated with higher magnitude earthquakes, the fraction of the ruptures that will be at closest distance P from the site increases with magnitude.

Because calculations are normally done only at a set of magnitudes m_i spaced Δm apart, ruptures of magnitude m_i earthquakes at distance P from the site. will cause a "spike" of acceleration occurrences to be accumulated in some $reg(j)$. (Recall that occurrences of accelerations in an interval $a_{j-1} < a \leq a_j$ are accumulated in a corresponding accumulator $reg(j)$.) The number of spikes in the acceleration occurrences at a site depends on the number of magnitudes used; the locations and sizes of the spikes depend upon the actual magnitude values. If no acceleration variability is assumed (for example, mean or median acceleration values only are used), or if σ_a, the standard deviation in log acceleration, is small, spikes in acceleration occurrences will result in sharp changes in the accelera-

tion exceedance curves at the accelerations at which the spikes are present. Furthermore, these spikes may occur at different accelerations at different sites, causing the probabilistic accelerations calculated at different sites to vary irregularly. SEISRISK III does a partial smoothing of acceleration occurrences resulting from the ruptures at distance P, as described below.

Magnitude Smoothing for Closest Distance Ruptures

Earthquakes at a fixed distance P from the site, with magnitudes in the range $m(j-1) \leq m < m(j)$ will produce median accelerations in the corresponding interval $a_{j-1} \leq a < a_j$. If earthquakes within the magnitude interval $m_i - \Delta m/2 \leq m_i \leq m_i + \Delta m/2$ are assumed to follow a Gutenberg-Richter magnitude-frequency relationship, the number $h(j)$ of those earthquakes with magnitudes that lie between $m(j-1) \leq m \leq m(j)$ can be determined ($h(j)$ contains magnitudes in the range $\max[m_i - \Delta m/2, m(j-1)] \leq m \leq \min[m(j), m_i + \Delta m/2]$). If $f(i)$ is the fraction of ruptures of length $l_i(m)$ of magnitude m_i earthquakes that occur at distance P from the site, approximately the same fraction of earthquakes will be at distance P for all magnitudes in the range $m_i - \Delta m/2 \leq m_i \leq m_i + \Delta m/2$, for small values of Δm. This means that a rate of accelerations $r(i,j) = f(i)h(j)$ resulting from ruptures at distance P for magnitudes $m_i - \Delta m/2 \leq m_i \leq m_i + \Delta m/2$ can be assigned to the acceleration interval $a_{j-1} \leq a < a_j$.

The solid lines in figure 4 show the yearly rate of exceedances of various accelerations calculated at a site at a perpendicular distance of 1 km from the center of a 300 km long fault, when the magnitude range is $5.0 \leq m \leq 8.0$ and the magnitude increments are $\Delta m = 0.6$, 0.3, and 0.15, respectively. The dashed lines in figure 4 were obtained in each case by magnitude smoothing of the spikes of acceleration occurrences. Very little difference appears in the smoothed results for $\Delta m = 0.6$, 0.3 and 0.15, suggesting that for sites near the center of the fault a magnitude increment of $\Delta m = 0.6$ might suffice if the spikes are smoothed in this manner.

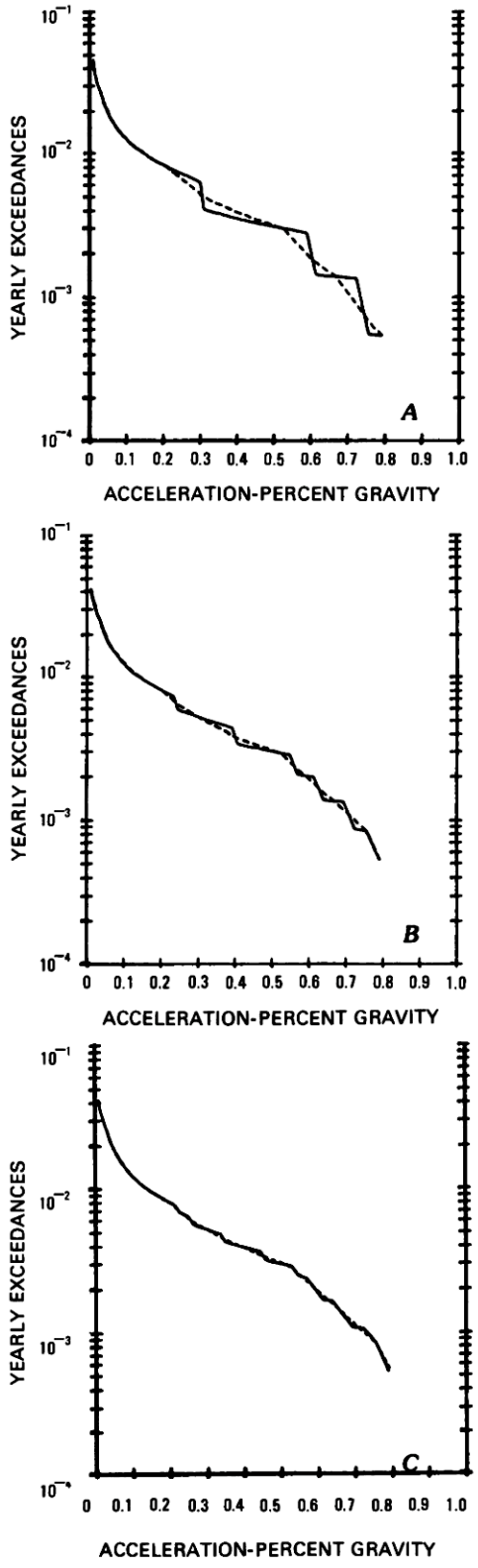


Figure 4. Yearly acceleration exceedance rates calculated at a site at a perpendicular distance of 1 km from the center of a 300-km-long fault, using magnitude increments *A*, Δm=0.6; *B*, Δm =0.3; *C*, Δm=0.15. The magnitude range is $5.0 \leq m \leq 8.0$. Attenuation curves are those of Schnabel and Seed, (1973). The dashed line in each case show the exceedances calculated when the acceleration density spikes resulting from ruptures that intersect the point on the fault closest to the site are smoothed for each magnitude.

In smoothing acceleration occurrences, occurrences normally assigned to magnitude $m_{max} - \Delta m/2$ are distributed throughout the interval $m_{max} - \Delta m/2 \leq m \leq m_{max} + \Delta m/2$; in particular, some of the occurrences are assigned to the interval containing the acceleration produced at distance P by the maximum magnitude, m_{max}. However, when magnitude smoothing is not done, the highest magnitude used in the calculations continues to be $m_{max} - \Delta m/2$, (equation 4). For a site beyond the end of the fault at (X, P) (either $X < 0$ or $X > L$), there are no ruptures at distance P, and, hence, no acceleration occurrence spikes from ruptures at closest distance; therefore, no magnitude smoothing is done for acceleration occurrences at such a site. This implies, in effect, a lower maximum magnitude for sites beyond the end of the fault.

Distance Smoothing for Ruptures Along Artificial Parallel Faults

The fault pattern may be quite complex within an active fault zone, and faults may be spread over a wide area. SEISRISK III allows one to model artificial parallel faults when the presence of faulting is inferred from geologic considerations and the orientation of the faults is assumed to be known. In this model, faults with a specified orientation are assumed to be equally likely anywhere within a given area; for example, faults parallel to the x-axis are equally likely to have their y-coordinate anywhere in the range $0 \leq y \leq w$. (The y-coordinate of the fault is uniformly distributed in this model.) To approximate a uniform distribution of faults, coordinates of evenly spaced parallel faults must be entered into the computer program. However, unless the spacing between faults is quite small (for example, 1 or 2 km), or unless some sort of smoothing is done, the acceleration levels calculated at sites a short distance apart can be quite different. Figure 5 illustrates how calculated acceleration levels may change as a function

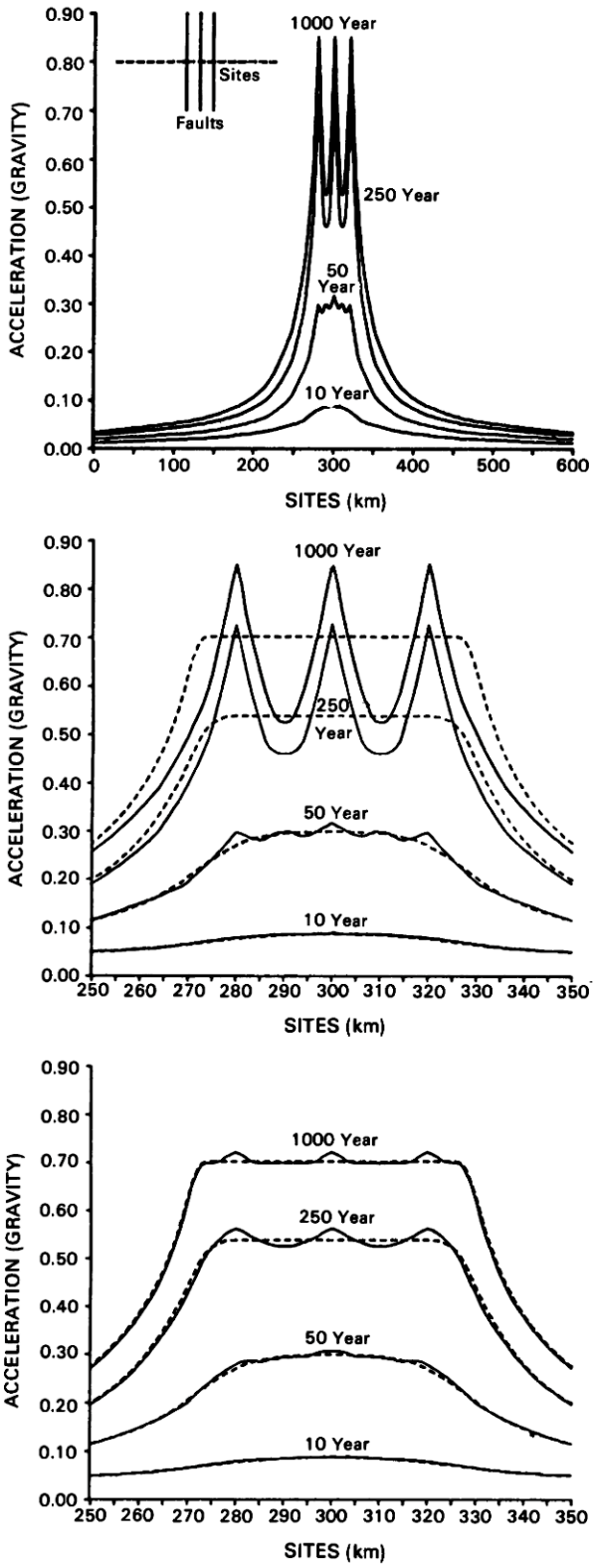


Figure 5. Accelerations calculated to have a 90 percent probability of not being exceeded during various time periods at sites on a line perpendicular to the centers of three parallel faults. A, 20 km apart. B, the scale is expanded and the dashed lines show values obtained when faults are placed instead at increments of 1 km over a distance of 60 km and the original seismicity is divided evenly among the faults. C, the solid lines show the results when 20-km fault spacing is used in the calculations, but ruptures on each fault that occur "as close as possible" to the site are placed instead on a line perpendicular to the original faults; the dashed lines are the same as in B. The Schnabel and Seed (1973) attenuation curves are used.

of distance from a fault. Figure 5*A* shows levels calculated at sites on a line perpendicular to the centers of three parallel faults, 20 km apart. In figure 5*B* the scale was expanded; the solid lines again show acceleration levels for the three fixed fault locations; the dashed lines give results obtained when faults were placed 1 km apart and the seismicity adjusted accordingly.

A distance smoothing is desirable when a single fault is used to represent a uniform distribution of faults. For example, assume a fault that extends from $(0, 0)$ to $(L, 0)$ was entered into the program to represent faults with y coordinate anywhere in the range $-y_0 \leq y \leq y_0$. Accelerations are calculated only for ruptures along the fault at y=0. If the site is located "above" the fault at (X, P) $(0 \leq X \leq L)$, an acceleration density spike will result from ruptures at distance P. To smooth the accelerations resulting from these ruptures, assume that any rupture with end points $(x_1, 0)$, $(x_2, 0)$ along the original fault, at distance P from the site, may instead have end points (x_1, y), (x_2, y) on any fault such that $-y_0 \leq y \leq y_0$. The seismicity associated with such a rupture along a fault at y concentrated at the point on the fault closest to the site at (X, y). This means that if faults [with end points at $(0, y)$, (L, y), $-y_0 \leq y \leq y_0$], are uniformly distributed parallel to the original fault, ruptures at closest distance on each fault will produce the same accelerations as points uniformly distributed on a line through the site perpendicular to, and extending $\pm y_0$ km from, the original fault. Acceleration occurrences in each interval $a_{i-1} \leq a \leq a_i$ may be determined from the frequency of the "point-source" earthquakes of each magnitude at the appropriate distances along this perpendicular line. In figure 6*C*, the ruptures at closest distance to each site were treated as point-source earthquakes along a line through the site and perpendicular to the faults; the calculation was done using $\Delta m = 0.6$.

MODELING ACCELERATION VARIABILITY

The ground-motion computations that have been described assume a single value of acceleration results from earthquakes of each magnitude and distance. However, in fact, a range of accelerations results; this variability in acceleration is commonly modeled by assuming that accelerations from earthquakes of a given magnitude and distance are lognormally distributed with standard deviation σ_a in $\log_e$ acceleration. Assuming a lognormal distribution of accelerations tends to cause the calculated probabilistic ground-motion levels at all sites to be higher than the levels calculated when only median accelerations are used (for example, Bender, 1984a).

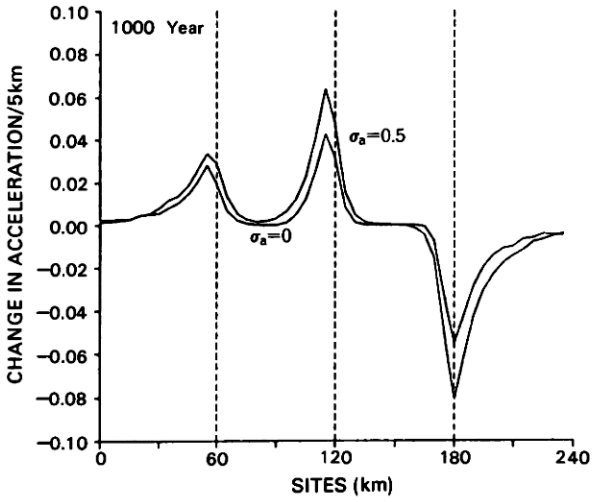


Figure 6. Differences in acceleration levels calculated for 1000-yr period at sites 5 km. apart, assuming σ_a=0 and assuming σ_a=0.5 in $\log_e$-acceleration in the attenuation function of Joyner and Boore (1981); no uncertainty (σ=0) in earthquake location.

Figure 6 shows the differences in probabilistic acceleration levels calculated for a 1000-year exposure time at sites 5 km apart, when the sites and sources are the same as in figure 2. The calculation was done assuming no uncertainty in earthquake location using median values in the Joyner and Boore attenuation and using $\sigma_a = 0.5$ in $\log_e$ acceleration. (The calculation for $\sigma_a = 0.5$ is the same as that shown in figure 2D for the 1000-year exposure period for $\sigma = 0$.) Figure 3 was included to illustrate that taking variability in the attenuation function into account, rather than reducing the differences in levels at sites near a boundary, may increase the site-to-site acceleration differences above those calculated when only median values are used in the calculations. This is discussed in detail in Bender (1986).

Assuming that σ_a, the standard deviation in log acceleration, is independent of magnitude and distance permits SEISRISK III to take acceleration variability into account only after doing calculations using median acceleration values for all earthquakes. Median rather than mean acceleration values are used initially because log acceleration (rather than acceleration) is assumed to be normally distributed; the median value of acceleration corresponds to the mean value of log-acceleration.

Let us assume occurrences of median accelerations in each interval $a_{j-1} < a \leq a_j$ have been accumulated in the corresponding $reg(j)$. Let $regs(j)$ be a second

array in which to accumulate occurrences of accelerations $a_{j-1} < a \leq a_j$ after acceleration variability has been taken into account. SEISRISK III redistributes a fraction of the acceleration occurrences accumulated in $reg(j)$ to $regs(k)$ (for each j and k) as follows.

The entry in $reg(j)$ is assumed to be concentrated at acceleration $\bar{a}$ where

$$\log_e(\bar{a}) = \frac{\log_e[a_{j-1}] + \log_e[a_j]}{2}. \tag{12}$$

In a lognormal distribution, given $\log_e \bar{a}$, the probability of an acceleration in the range $a_1 \leq a \leq a_2$ is the area under the normal probability curve:

$$p(a_1, a_2) = \frac{1}{\sqrt{2\pi}\sigma_a} \int_{x=\log_e a_1}^{\log_e a_2} \exp\left[-\frac{(x-\log_e \bar{a})^2}{2\sigma_a^2}\right] dx. \tag{13}$$

From equation 13, a fraction $f(k)$ of the accelerations in $reg(j)$ is placed in $regs(k)$,

$$f(k) = p(a_{k-1}, a_k)\rho(j). \tag{14}$$

where $\rho(j)$=occurrences of (median) accelerations in the range $a_{j-1} < a \leq a_j$ accumulated in $reg(j)$. Accounting for acceleration variability in this manner may be considerably faster computationally than methods which include acceleration variability for earthquakes of each magnitude and distance. Accounting for acceleration variability after calculations for median accelerations have been completed is possible because of the assumption that σ_a, the standard deviation in log acceleration is independent of distance and magnitude.

CALCULATING GROUND-MOTION EXCEEDANCE PROBABILITIES

Given that average annual acceleration occurrence rates in the interval $a_{j-1} << a \leq a_j$ have been accumulated in $regs(j)$, $1 \leq j \leq 55$, $Ex[a_j]$, the yearly occurrence rate of accelerations $a > a_j$, is simply

$$Ex[a_j] = \sum_{i=j+1}^{55} \rho^*(i) \tag{15}$$

where $\rho^*(i)$ = accelerations in the range $a_{i-1} < a \leq a_i$ accumulated in $regs(i)$. The yearly exceedance rate $Ex(a)$ of acceleration a can be obtained by interpolating the $Ex[a_j]$ in equation 15.

We are now in a position to determine the probability associated with the exceedance of any acceleration. Earthquake occurrences are assumed to have a Poisson distribution; that is, the probability that an earthquake will occur is the same at all times regardless of when the last earthquake occurred. For a Poisson distribution with mean rate μ, the probability of exactly k events during time t is

$$P(k,t) = \frac{(\mu t)^k \exp(-\mu t)}{k!}. \tag{16}$$

The probability of no event during the same time interval is, therefore,

$$P(0,t) = \exp(-\mu t). \tag{17}$$

Setting $\mu = Ex(a)$, the probability of not exceeding a in t years is

$$P(0,t) = \exp[-Ex(a)t]. \tag{18}$$

Taking the natural logarithm ($\log_e$) of both sides of equation 18 gives

$$Ex(a) = -\frac{\log_e[P(0,t)]}{t}. \tag{19}$$

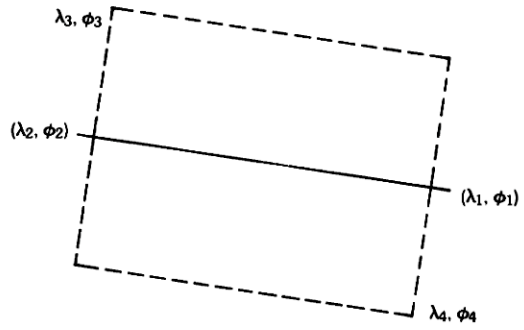
That is, the value of a for which equation 19 is satisfied is the acceleration which has probability $p = P(0,t)$ of not being exceeded during the time interval t. (Equivalently, a has probability $q = 1 - P(0,t)$ of being exceeded one or more times during time t.) By specifying any two of the three parameters (p, a, t), one can determine the remaining parameter. However, in SEISRISK III a single probability level p and several values (up to 20) of time are specified and the corresponding values of a are obtained.

MODEL IMPLEMENTATION–DETAILS AND REQUIREMENTS

In this section, we further describe the model and how it is implemented in SEISRISK III. Input parameters required to run SEISRISK III are identified and described briefly at the beginning of the program listing in Appendix A; here we give a more detailed explanation of the meaning and roles of some of these parameters.

Geometry: Transformation to Local Equator

In both SEISRISK II and SEISRISK III, a pair of latitude and longitude values are selected (input) to be two points on a great circle that becomes the equator in a transformed coordinate system. The great circle is chosen to pass through the region of interest, and all subsequent input coordinates (for example, end points of source areas and fault segments) are transformed relative to the new equator. Furthermore, the subsequent calculations are done assuming a rectangular, rather than a spherical coordinate system; in the new system the (new) equator becomes the x-axis.



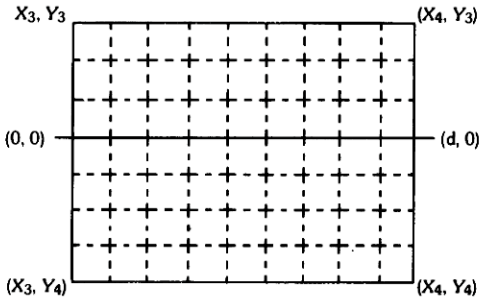


Figure 7. Four points define transformation and seismic-felt area. Great circle through (λ_1, ϕ_1), (λ_2, ϕ_2) becomes new equator. $(\lambda_1, \phi_1) \rightarrow (d, 0)$. $(\lambda_2, \phi_2) \rightarrow (0, 0)$. (λ_3, ϕ_3) and (λ_4, ϕ_4) become end points of rectangular area containing seismic felt points. $(\lambda_3, \phi_3) \rightarrow (x_3, y_3)$. $(\lambda_4, \phi_4) \rightarrow (x_4, y_4)$.

For points in the vicinity of the equator, calculations done using a rectangular coordinate system give a good approximation to those obtained using spherical coordinates. Because of convergence of parallels of latitude, the error could be substantial if the original $(long, lat)$ coordinates are treated as being on a flat surface. However, this error is minimized when the coordinates are transformed to lie in the vicinity of a new equator, so that, for example, within a square region 12 degrees wide and centered on this equator, the maximum error introduced by using the straight line distance between two points (x_i, y_i), (x_j, y_j) instead of the great circle distance is less than 0.6 percent.

Sites at which accelerations are calculated lie on a rectangular m by n grid in which the site (x_i, y_j) is

$$x_i = \text{ck(longitude of row } i), \quad 1 \le i \le m$$
$$y_j = \text{ck (latitude of col } j), \quad 1 \le j \le n$$

where ck=conversion factor for degrees to kilometers at the equator.

Coordinates $(long, lat)$ of two opposite corners (upper left, lower right) of the seismic felt area in the initial system and grid spacing are input to the program (fig.7). Sites in the new coordinate system are located at even increments in $\Delta x, \Delta y$ ($\Delta y = \Delta x$) at intersections of grid lines. (In SEISRISK II, $\Delta y = \Delta x$ was not required.) At the end of the computation, sites are transformed to the coordinates they would have in the original system. Sites may also be points evenly spaced on a number of lines; in this case, $(long, lat)$ of two end points of each line, number of sites per line, and number of lines are specified as input. However, calculations for sites on a line are done assuming only seismically homogenous source zones; earthquake location variability cannot be implemented for sites that are not on a grid of sites.

Acceleration Table Levels

A table of 55 acceleration levels a_k, such that $a_k < a_{k+1}$ for $1 \le k \le 55$, is constructed within the program. A scale factor (input) allows the original range of acceleration values in the table to be expanded or contracted. For a scale factor of one, 50 of the acceleration levels are in the range 0.02-1.0 g at increments of 0.02 g. Four acceleration levels are less than 0.02 g; accelerations at these levels are saved for calculations that are done when acceleration variability is taken into account; these accelerations are not printed in the output and, hence, are invisible to the analyst. A scale factor of 0.5 would permit values from 0.01g-0.5g. (The table is created in the subroutine "box" and other values may be substituted by replacing this subroutine.)

Given the table of acceleration levels a_k, a corresponding array $reg(k)$ is set aside to accumulate fractional acceleration occurrences, with occurrences in the range $a_{j-1} < a \le a_j$ being placed in $reg(j)$. Accelerations less than $a(1)$ are placed in $reg(1)$ and those exceeding the highest entry $a(max)$ (where max =54) are placed into $reg(max+1)$. Earthquakes as point events within quadrilateral source areas and as finite length ruptures along linear fault segments may contribute to accelerations at a site. We shall discuss these two cases separately.

Source Zone Computations

Area Computations

A seismic source zone is defined as a seismically homogeneous area enclosed by one or more arbitrary quadrilaterals, connected or disjoint (fig.8).

Recall that an acceleration "box" represents occurrences of accelerations in some range $a_{j-1} < a \le a_j$. For a specified magnitude, acceleration box boundaries $[a_{j-1}, a_j]$ correspond to earthquakes at distances in the range $r(j) \le r < r(j-1)$ from the site. The contribution to accelerations $a_{j-1} < a \le a_j$ produced by magnitude m earthquakes in a seismically homogeneous quadrilateral is equal to the rate per unit area within the

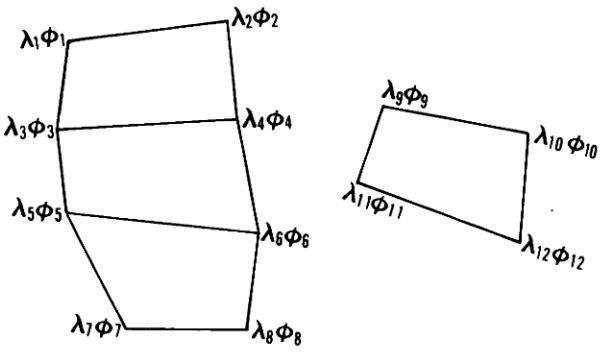


Figure 8. Seismic source zone consisting of two sets of quadrilaterals

Input: For each quadrilateral set

(1) jseg ifr tot

jseg=number+1 of quadrilaterals in this set
ifr identifies current set, $1 \leq$ ifr $\leq$ tot
tot=total number of sets of quadrilaterals

(2) pairs of quadrilateral corner points.

In this example, input is

4 1 2
λ_1 ϕ_1 λ_2 ϕ_2
λ_3 ϕ_3 λ_4 ϕ_4
λ_5 ϕ_5 λ_6 ϕ_6
λ_7 ϕ_7 λ_8 ϕ_8
2 2 2
λ_9 ϕ_9 λ_{10} ϕ_{10}
λ_{11} ϕ_{11} λ_{12} ϕ_{12}

quadrilateral, multiplied by the area within the quadrilateral at distances $r(j) \leq r < r(j-1)$ from the site. The area of the quadrilateral within the distance range $r(j) \leq r < r(j-1)$ is approximately the area contained within a ring of radius r and width $dr = |r(j-1) - r(j)|$ centered at the site (fig.9).

The distance range over which earthquakes produce accelerations in the interval $a_{j-1} < a \leq a_j$ depends on magnitude. The accelerations from earthquakes of each magnitude in the appropriate distance range within a source area are summed to obtain the total contribution from these earthquakes added into $reg(j)$. The procedure is repeated for all acceleration intervals, and subsequently for all sources.

To save computation time, for a particular site and source, both SEISRISK II and SEISRISK III create a table of distance versus arc-length (for a predetermined set of radii) and then later interpolate in this table to obtain areas at desired distances. In this case

$$Area = \phi r dr \qquad (20)$$

where ϕ=arc length in radians of that portion of the circle of radius $r = r_j$ that lies wholly within the quadrilateral
dr=interval width.

The quantity ϕr is tabulated as a function of r for seven evenly spaced radii r per quadrilateral and for two to eight additional radii to vertices and edges, depending upon the location of the site relative to the source. The accuracy of the calculated areas depends on the geometry of the source, location of the site, and on the number of radii used in the calculations.

To test the accuracy of the arc-area computations for a fixed value of *itst*, areas for a set of 5856 quadrilateral-site pairs (used in a California study) were evaluated using *itst*=5, 7, 9, 11, and 13. For *itst*=7, the

("arc estimate" area - "true" area)/ "true" area

exceeded 0.03 only five times and never exceeded 0.04; for *itst*=5, the error exceeded 0.04 seven times; for *itst*=9, 11 and 13, the largest error was 0.02. We concluded that *itst* = 7 gave sufficient accuracy. The number of evenly spaced radii is set by parameter *itst* = 7 in subroutine RRISK; *itst* may be increased be recompiling the program, but this should probably be done only under unusual circumstances (for example, for testing purposes), because using larger values of *itst* increases the computation time.

Grid Spacing Requirements For Modeling Uncertainty

SEISRISK III initially calculates expected acceleration occurrences at each site on a grid assuming that seismicity is uniform throughout a source zone. Then to determine acceleration occurrences at a site at (X_1, Y_1),

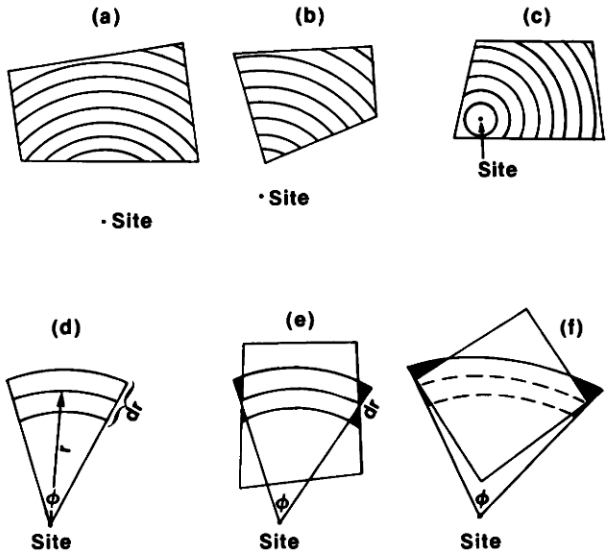


Figure 9. Arcs of circles with centers at the site are used to approximate the area of a quadrilateral. $Area = \phi r dr$ in D, E, and F. Shaded area in E and F is the difference between the quadrilateral area and arc area. In E errors nearly cancel; in F the arc area is an overestimate.

with earthquake location uncertainty included, SEISRISK III computes a weighted average of acceleration occurrences previously calculated at various sites assuming uniform seismicity. The fractional contribution or weight assigned to the accelerations calculated at a site at (X_i, Y_k) depends on σ and on the distance between (X_1, Y_1) and (X_i, Y_k) (equation 2).

The grid spacing or distance between sites is of major importance in these calculations. Accelerations at sites at distances greater than 2.0σ - 2.5σ from (X_1, Y_1) have low weights and little effect on the weighted accelerations at (X_1, Y_1). This means that if the spacing between sites is large relative to σ, the only accelerations that will have much weight at (X_1, Y_1) are those originally calculated at (X_1, Y_1) (for uniform seismicity), and, in this case, the weighted results will give a poor approximation to the true effects of earthquake location uncertainty. In tests to determine the maximum acceptable grid spacing, we found that we could use a spacing dx as large as σ and still obtain reasonable results; in these tests, for $dx = \sigma$ results were quite consistent with those obtained using $dx = \sigma/2$.

In order to take into account accelerations at sites at distances 2σ from every site of interest, SEISRISK III extends the gridded region to include additional sites (fig.10). If the sites of interest are at row and column intersections on an $(n)(n)$ grid, and the distance between rows and columns is dx, then the grid must be extended in each direction by at least $n_{rc} = 2.0\sigma/dx$ rows and columns; that is, the calculations must be done at $(n + 2n_{rc})^2$ sites in order to model effects of earthquake location variability at each of the original n sites. The number of additional sites can be quite large if σ is small relative to dx. For example, if $dx = 10$ and $\sigma = 40$, the number of sites increases from n^2 to $(n+16)^2$.

Increasing the number of sites at which the calculations are performed increases the storage requirements as well as the computation time. Because acceleration occurrences calculated for uniform seismicity at various sites are used in calculating the weighted averages, we retain these occurrences in a two-dimensional array. This requires an array of dimension of at least (II, JJ), where II=number of acceleration levels in the *reg* array, and JJ=the number of sites (including the sites added for earthquake location variability) at which the calculation is performed ($JJ \geq (n+2n_{rc})^2$). Furthermore, a second array of the same size is set aside in which to accumulate the weighted acceleration occurrences. In SEISRISK III, the two arrays RAWBIN and SMOBIN for saving the accelerations are dimensioned (55,1600). A total of 1600 sites (including the sites added for earthquake location variability) can be processed. (Dimensions of arrays SMOBIN and RAWBIN can be adjusted to change the number of sites permitted; the parameter *maxar* must be set equal to the site dimension.) Saving results in two-dimensional arrays contrasts with the procedure in SEISRISK II, in which acceleration densities calculated at a single site were written onto disc (logical unit 3 or logical unit 4) rather than being saved in a large array.

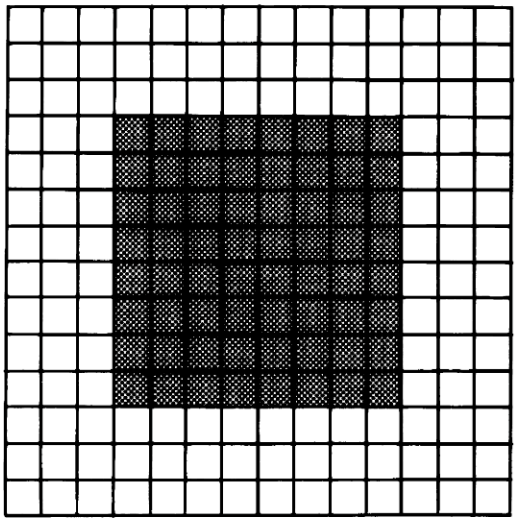


Figure 10. Sites of interest located at intersections of grid lines within darkened area. If the spacing between grid lines is dx and the standard deviation in earthquake location is σ, the gridded region is extended in each direction by $n_{rc} = 2.0\sigma/dx$ (rounded to the next highest integer) rows and columns.

Using Sources with Different σ Values

SEISRISK III permits earthquakes associated with different source-zones to have different values of σ. If σ is the same for earthquakes in a number of zones, weighted averages of accelerations can be calculated after the calculations assuming uniform seismicity are completed for earthquakes in all of the zones. On the other hand, if a different σ is associated with earthquakes in each source zone, the weighting (equation 2) of accelerations resulting from earthquakes in a zone must be performed after the calculations for uniform seismicity have been done for that zone.

To instruct the program when to calculate weighted averages, a value of an input parameter *num* is entered for each source zone along with other data for that zone; *num*=98 indicates averaging should be performed for all accelerations accumulated since the averaging was last performed. The current value of σ is used in this averaging, and one need enter a value of σ only for source zones for which $num = 98$. Figure 11*A* gives an abbreviated flowchart for the source zone computations.

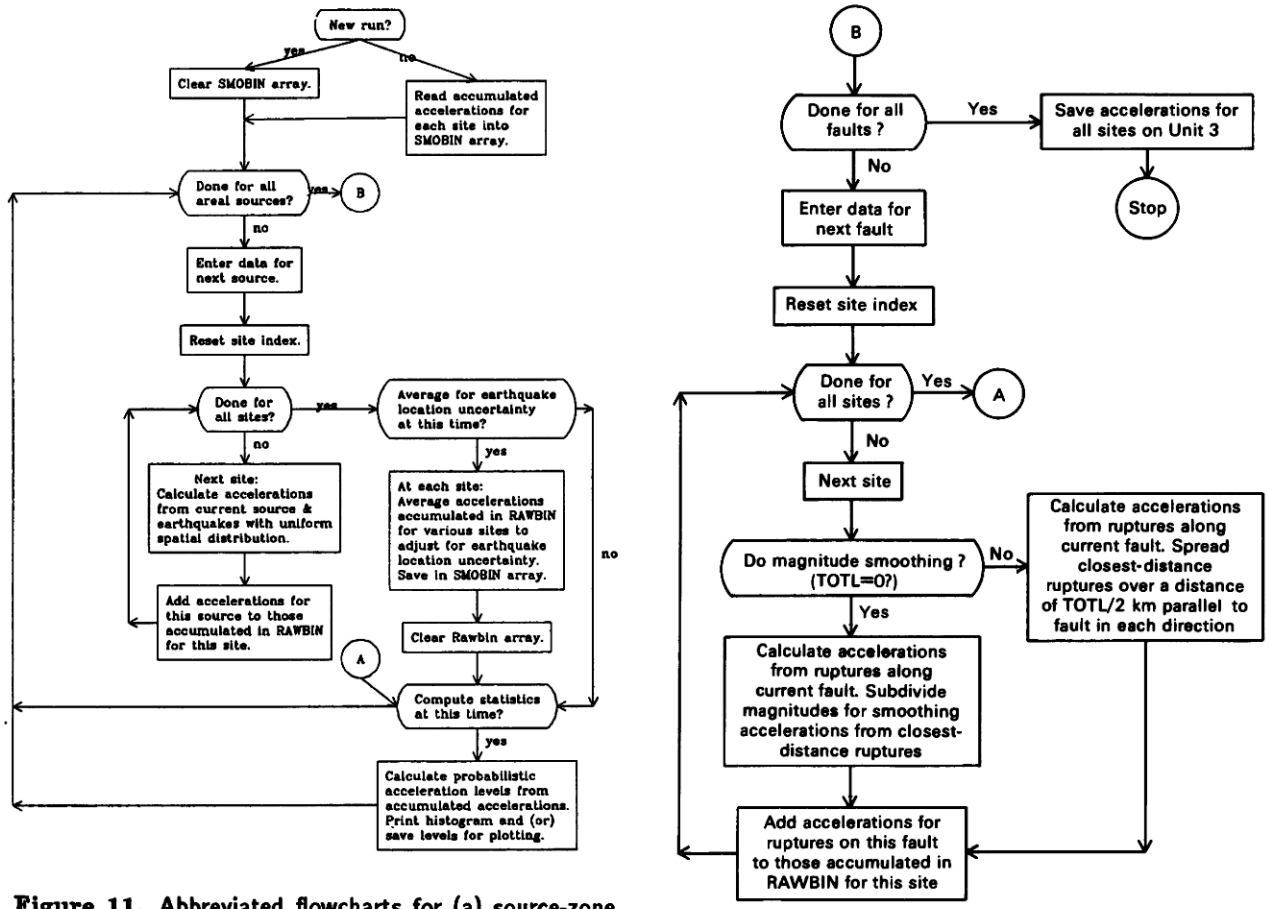


Figure 11. Abbreviated flowcharts for (a) source-zone calculations and (b) fault-rupture calculations.

When considering using different values of σ for earthquakes in different zones, we should keep the following considerations in mind.

1. The number of rows and columns and the spacing between sites remains the same throughtout the analysis. The value of n_{rc}, the number of additional rows and columns by which the initial gridded area must be extended in each direction to account for earthquake location uncertainty, is determined by the largest value of σ to be used for any source-zone. However, n_{rc} is calculated before the source zone inputs are processed, and therefore, the largest value of σ must be identified in the program initially.
2. If considerably different values of σ are assigned to earthquakes in different areas, the grid spacing must be sufficiently small to adequately represent the intended variability for those earthquakes with the smallest σ. At the same time, a sufficient number of marginal sites (determined by n_{rc}) must be provided to permit accurate calculations when σ is large. Accomodating both these requirements can lead to evaluating accelerations at considerably more sites than necessary, both when σ is small (because of an unnecessarily large value of n_{rc} in this case) and when σ is large (because of the small increment dx relative to σ).
3. Allowing different values of σ can lead to an anomalous ridge of seismicity along one side of a joint boundary of contiguous source zones. In general, we suggest that two source zones with different σ values be separated by at least two to three times the difference in the σ values.

Fault Computations

Rupture Length Computations

A rupture of a magnitude m earthquake is defined to have length $l_i(m)$ with probability $p(i)$, where

$$\log_{10}[l_i(m)] = a + bm + \eta(i)\sigma_l \qquad (21)$$

where a and b are constants.

$p(i)$ =probability of rupture of length $l_i(m)$

$$p(i) = \frac{1}{\sqrt{2\pi}} \int_{f(i)}^{f(i+1)} \exp(-\frac{x^2}{2}) dx \qquad (22)$$

where $f(i)$ is selected so that $\eta(i)$ is

$$f(i) < \eta(i) \leq f(i+1); \quad f(1) = -\infty; \quad f(n_r+1) = \infty;$$

$$\sum_{i=1}^{n_r} p(i) = 1.$$

n_r=number of rupture lengths per magnitude.

SEISRISK III allows either one rupture length or or five rupture lengths per magnitude depending on whether median rupture lengths only are assumed or rupture length variability is to be modeled in the length versus magnitude relationship. Thus

$$\eta(i) = \begin{cases} 0, \text{ if only median rupture length is used} \\ \quad \text{(one rupture length per magnitude),} \\ \text{five values in the range } -2 \leq \eta(i) \leq 2 \\ \quad \text{(five lengths per magnitude).} \end{cases} \tag{23}$$

The values of a, b and σ_l may be provided as input for the rupture-length magnitude relationship (equation 21). If a, b and σ_l are all zero or blank, previous values of these quantities are used. If no values are provided, default values of Bonilla and Buchanan (1970) are used:

$$a = -1.085 \qquad b = 0.389 \qquad \sigma_l = 0.52$$

If $a \neq 0$, $b \neq 0$, and $\sigma_l = 0$ are input, a single rupture-length value is used.

Magnitude Smoothing: Calculation of Gutenberg-Richter b-Value

The Gutenberg-Richter b-value used to determine the number of earthquakes in an arbitrary magnitude interval is calculated from the number $N(m_1)$ of earthquakes at magnitude m_1 and $N(m_2)$, the number at magnitude m_2; that is,

$$b = -\frac{1}{\Delta m} \log\left[\frac{N(m_2)}{N(m_1)}\right]. \tag{24}$$

where m_1 is the lowest magnitude used in the calculations and $m_2 = m_1 + \Delta m$ (equation 4). However, if only a single magnitude is specified, neither the magnitude interval nor the b-value can be calculated, and hence default values of b and Δm are provided. (The default values are $b = 1.0$ and $\Delta m = 0.6$.)

Distance Smoothing Option

To instruct the program to do distance (rather than magnitude) smoothing, the distance d between the parallel faults must be input.

$$totl = \begin{cases} d & \text{distance between faults if this is a set of} \\ & \text{"dummy" faults used to approximate} \\ & \text{a uniform field of faults.} \\ 0 & \text{if individual fault.} \end{cases}$$

If $totl = 0$ (or is blank), the program assumes that the faults are individual faults, and the previously discussed magnitude (rather than distance) smoothing is performed. Distance smoothing over a range of $\pm d/2$ km for a single fault can be forced by setting $totl = d$ for a single fault. The program cannot do both distance and magnitude smoothing. Figure 11*B* shows an abbreviated flowchart for the fault rupture calculations.

Attenuation Function: Input Table, Interpolation, Scale Factor

The program computes ground-motion (acceleration) by interpolating in a table of acceleration as a function of magnitude and distance. Interpolations are linear in distance and linear in log acceleration and magnitude; in SEISRISK II, interpolations were linear in log magnitude rather than in magnitude.

The table of ground-motions is read in for up to eight magnitudes and 20 distances. Velocity or some other ground-motion parameter may be substituted for acceleration. If the ground-motion values read in are in a range different from 0.02 to 1.0, the appropriate scale factor must be input to make the "box" levels compatible with the desired values.

Halving the Magnitude Intervals

The program requires that earthquake rates be entered for each source at equally spaced magnitude intervals. The magnitudes that are entered are assumed to be values at the centers of the intervals. If input paramater $inos = 1$, the program will divide each magnitude interval into two parts and perform the calculations for magnitudes at the midpoints of the new intervals. Earthquake rates are calculated for each interval from the total earthquakes assigned to the original intervals using a Gutenberg-Richter b-value based on the number of earthquakes in the first two (original) magnitude intervals, as in equation 24.

Output Options

As stated earlier, we seek the acceleration that, with probability q, will be exceeded (or probability $p = 1 - q$ will not be exceeded) during t years. We have seen that this is the value a of acceleration for which $Ex(a)$ the yearly exceedance rate of a at the site is

$$Ex(a) = \frac{\log_e(p)}{t}. \tag{25}$$

Assuming yearly exceedance rates (as in equation 15) have been calculated for each of 55 acceleration levels, the solution a that corresponds to the specified (p, t) may be found by interpolating between the calculated values.

Solutions are computed for up to 20 time periods t using rates derived for median accelerations only and also for the same set of times using rates that include acceleration variability.

The results, including the acceleration histogram, may optionally be written to file 16 for later printing. Summary results may be saved on file 2 for future plotting.

Accelerations from the earthquakes within a source area or along a set of faults are calculated for all sites before proceeding to the next source area or fault, with acceleration occurrences being added to those obtained previously for each site. Therefore, partial results may be examined at any stage. Output options are given for each source set by the input parameter *iprint*.

iprint=

-1: continue to next set; no print, no statistics

+1: compute solutions a; write histogram and solutions to file 16, then continue to next set

+2: same as +1 but in addition save the solutions on file 2

+3: no write to file 16; save solutions on file 2.

Accumulating Rates In Sucessive Runs

Rates that have been accumulated in each acceleration interval for each site are saved in file 3 at the end of a run. These values may be read back into the 2-dimensional array SMOBIN for a subsequent run. (The saved rates are rates that have been adjusted for earthquake location uncertainty.) Doing successive runs makes it possible, for example, to accumulate accelerations from sources having different attenuation functions. The equator, maximum σ, gridded region, and rows and columns containing sites of interest should not be changed from their original values in a run continuation.

$$isw = \begin{cases} 1 & \text{if this is a run continuation} \\ 0 & \text{if this is a new run.} \end{cases}$$

REFERENCES CITED

Bender, B., and Perkins, D. M., 1982, SEISRISK II, A computer program for seismic hazard estimation: U.S. Geological Survey Open-File Report 82-293, 103 p.

Bender, B., 1984a, Incorporating acceleration variability into seismic hazard analysis: Bulletin of Seismological Society of America v. 74, p. 1451-1462.

——1984b. Seismic hazard analysis using a finite fault rupture model: Bulletin of Seismological Society of America v. 74, p. 1899-1923.

Bender, B., 1986. Modeling source zone boundary uncertainty in Seismic hazard analysis: Bulletin of Seismological Society of America v. 76, p. 329-341.

Bonilla, M. G., and Buchanan, J. M., 1970, Interim report on world wide historic and surface faulting: U. S. Geological Survey Open-File Report, 32 p.

Joyner, W. B., and Boore, D. M., 1981. Peak horizontal acceleration and velocity from strong motion records including records from the 1979 Imperial Valley, California earthquake. Bulletin of Seismological Society of America v. 71, p. 2011-2038.

Schnabel, P. B., and Seed, H. B., 1973, Acceleration in rock for earthquakes in the Western United States: Bulletin of Seismological Society of America v. 63, p. 501-516.

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

APPENDIX A

Details of Fault-Rupture Computations

A fault consists of one or more connected line segments. A rupture may be wholly contained within one segment or overlap any number of fault segments. It may be as long as the fault, but cannot extend beyond the end of the fault. The distance from a rupture to a site is the distance from the site to the point on the rupture closest to the site.

A conceptually simple approach for calculating accelerations at a site resulting from ruptures along the fault pretends that ruptures of length $l_i(m)$ of magnitude m earthquakes occur only at a set of n distinct locations, with rupture centers evenly spaced at increments $\Delta l_i(m)$ on the fault. In this case, the k^{th} rupture center is at a distance $d_i(k)$ from one end of the fault where

$$d_i(k) = \frac{l_i(m)}{2} + (k-1)\Delta l_i(m) \qquad (A1)$$

and

$$\Delta l_i(m) = \frac{L - l_i(m)}{n-1}, \quad 1 \le k \le n. \qquad (A2)$$

The fractional seismicity $C(i,m)$ associated with one such rupture is

$$C(i,m) = \frac{\rho(m)p(i)}{n} f, \qquad (A3)$$

$\rho(m)$=rate of magnitude m earthquakes for the set of faults,

f=length of this fault/total length of faults in set,

$p(i)$=probability of rupture of length $l_i(m)$, for either 1 or 5 rupture lengths per magnitude.

For each rupture, the closest distance to the site is determined (fig.A1). For one such rupture of magnitude m and length $l_i(m)$, the acceleration interval is found that contains the acceleration produced by a rupture of this magnitude at this distance. The fractional seismicity $C(i,m)$ is added to the accumulator $reg(j)$, for the appropriate j.

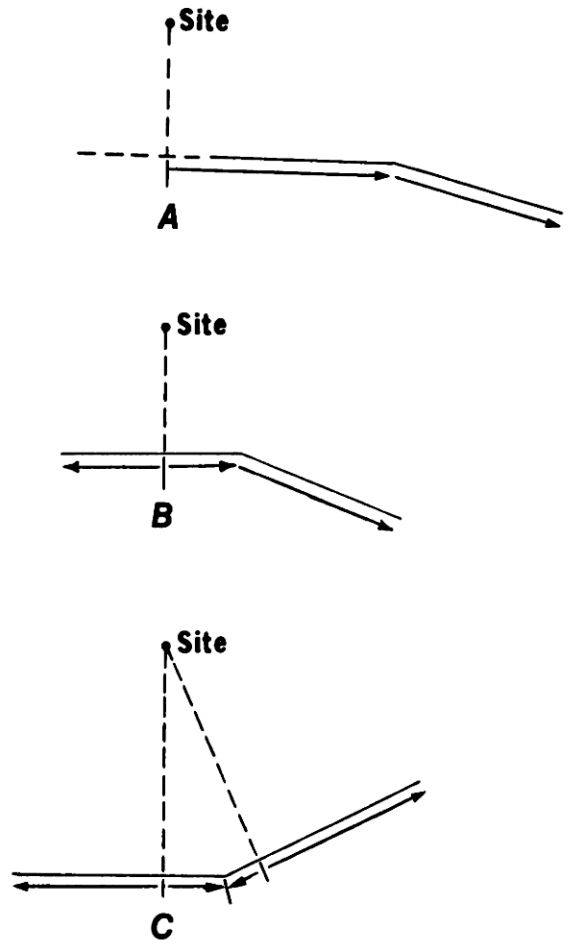


Figure A1. Arrows indicate direction of increasing distance along the fault from the site.
A Closest distance from site to fault is at left end of fault. Distance increases monotonically along the fault as one moves in the direction of the arrows.
B Distance to site decreases as one moves along the fault from left end of fault to point A, then increases monotonically as one continues to other end.
C Distance to site alternately decreases and increases as one moves along the fault.

The computer time required may be substantial if the computation of nearest distance from the site to a rupture is done repeatedly for different rupture locations along the fault. Some transformations are made in SEISRISK III that simplify the site-to-rupture computations. The following describes these transformations.

For an arbitrary fault segment $1 \le i \le n$, let

$$L(i) = \sqrt{[x_1(i)-x_2(i)]^2+[y_1(i)-y_2(i)]^2} \qquad (A4)$$
$$= \text{length of segment } i,$$

where $[x_1(i), y_1(i)], [x_2(i), y_2(i)]$ =coordinates of endpoints of segment i.

The coordinates are transformed so that fault end points

$$[x_1(i), y_1(i)] \rightarrow (0,0)$$
$$[x_2(i), y_2(i)] \rightarrow [L(i), 0] \qquad (A5)$$

and the site at (F_x, F_y)

$$(Fx, Fy) \rightarrow [X(i), P(i)]$$

where

$$X(i) = [Fx-x_1(i)]\cos\alpha+[Fy-y_1(i)]\sin\alpha,$$
$$P(i) = -[Fx-x_1(i)]\sin\alpha+[Fy-y_1(i)]\cos\alpha. \qquad (A6)$$
$$= \text{perpendicular distance from site to line } i.$$

$$\cos\alpha = \frac{x_2(i)-x_1(i)}{L(i)}, \qquad \sin\alpha = \frac{y_2(i)-y_1(i)}{L(i).}$$

After $L(i), X(i)$ and $P(i)$ have been determined, the fault segment may be regarded as located on the x-axis extending from (0,0) to $[L(i), 0]$ and the site located at $[X(i), P(i)]$ (fig.A2).

If the fault consists of N connected line segments, SEISRISK III computes and saves $L(i), X(i), P(i)$ for each segment, $1 \le i \le N$. Assume an arbitrary rupture has one end point on segment j and the second on segment k, where $1 \le j \le k \le N$. The coordinates of the end points of the part of the rupture that lies on segment i $(j \le i \le k)$ are defined as $[R_1(i), 0], [R_2(i), 0]$ (fig.A3).

$$0 \le R_1(i) \le R_2(i) \le L(i) \quad \text{for } i=j \text{ or } i=k, \qquad (A7)$$

$$R_1(i)=0, R_2(i)=L(i) \quad \text{for } j<i<k.$$

If a rupture extends over three or more fault segments, the rupture overlap with an interior segment is the entire segment.

In the new system, the relative end and overlap points $R_1(i), R_2(i)$ are determined from the total distance δ along the fault of the initial end point (start) of the given rupture and from the length of the rupture and fault segments, by calculating $Rem(j)$ for each segment j, where $Rem(j)$ is defined as

$$Rem(j) = \delta - \sum_{i=0}^{j-1} L(i), \text{ where } L(0)=0.$$

The first segment containing the rupture beginning at δ is the segment j for which $Rem(j)>0$ and $Rem(j+1) \le 0$. (For this j, $R_1(j) = Rem(j)$.) The segment containing the final end point of a rupture of length l is determined similarly by substituting $(\delta+l)$ for (δ) above. If the rupture overlaps segment i, let $D^2(i)$ =square of the closest distance from the site to the rupture on segment i. If the rupture extends over several segments, the value of $D^2(i)$, that is the minimum for all such segments, is required. $D^2(i)$ is defined as follows:

$$D^2(i) = \begin{cases} [R_1(i)-X(i)]^2+P(i)^2 & \text{if } X(i)<R_1(i), \\ [X(i)-R_2(i)]^2+P(i)^2 & \text{if } X(i)>L(i), \\ P(i)^2 & \text{if } R_1(i) \le X(i) \le R_2(i). \end{cases} \qquad (A8)$$

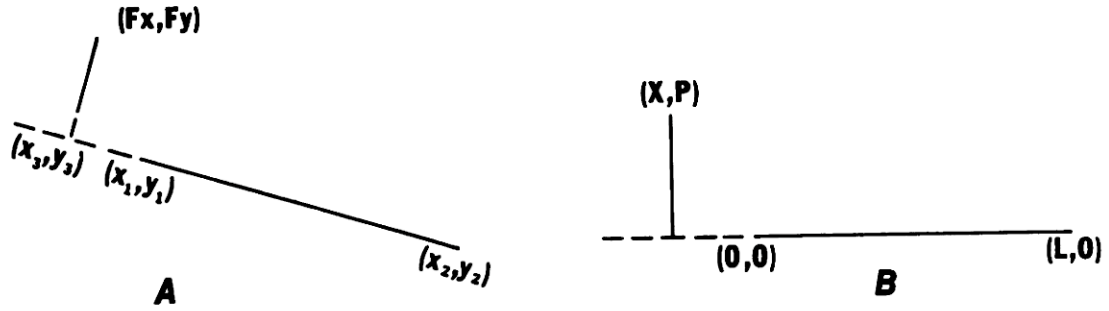


Figure A2. Fault segment and site. *A* original coordinates, *B* relocated so that segment lies on X-axis. *A* Original coordinates. Segment of fault extends from (x_1, y_1) to (x_2, y_2); site is at (F_x, F_y). Perpendicular from site intersects extended fault line at (x_3, y_3). *B* Relocated. Fault and site moved so that fault extends from (0,0) to $(L, 0)$. Site moved to (X, P).

Thus, having calculated and saved $L(i), P(i), X(i)$, determining the (square of the) closest distance requires only a simple computation for each fault segment that the rupture overlaps.

Assigning the fractional seismicity (equation A3) associated with a rupture to a single acceleration interval results in chunks of seismicity in a set of intervals. This means that the calculated acceleration densities do not vary smoothly as a function of acceleration level. In some cases, SEISRISK III can avoid doing calculations for a distinct set of ruptures and instead directly calculate the fraction of ruptures of magnitude m earthquakes that produce accelerations in each interval, $a_{j-1} < a \leq a_j$. When the more direct approach cannot be employed, SEISRISK III calculates site-to-rupture distance for n ruptures of magnitude m earthquakes, evenly spaced along the fault (equation A1). If rupture i is at distance d_i from the site, and rupture $i+1$ is at distance d_{i+1} from the site, SEISRISK III assumes a fraction $1/n$ of the possible ruptures are in the distance range $d_i \leq d \leq d_{i+1}$ (assuming $d_i \leq d_{i+1}$). It then finds the accelerations $a(i)$ and $a(i+1)$ produced by magnitude m ruptures at distances d_i and d_{i+1}. Assuming $a(i+1) \leq a_j \leq a_{j+1} \leq a(i)$, the fractional seismicity assigned to the acceleration interval $a_j \leq a \leq a_{j+1}$ is given by

$$C(i,m)\frac{a_{j+1}-a_j}{a(i)-a(i+1)}, \qquad (A9)$$

where $C(i,m)$ is defined as in equation A3.

We next consider two cases in which a more direct calculation of fractional accelerations can be done for each acceleration interval. We use the notation and definitions already developed.

Case 1. $X(i) < 0$ for all $1 \leq i \leq n$ (or $L(i) < X(i)$ for all $1 \leq i \leq n$). The point on the rupture nearest the site is at one end of the rupture with the distance to the site increasing (or decreasing) monotonically along the rupture.

If $X(i) < 0$ for all i, for any rupture overlapping segments j through k, where $j \leq k$, then $[R_1(j), 0]$ is the point closest to the site. Similarly if $X(i) > L(i)$ for all i, then $[R_2(k), 0]$ is the point of the rupture closest to the site.

Assume $X(i) < 0$, for all i. ($L(i) < X(i)$ could be considered in a symmetrical argument.) In this case the rupture point nearest to the site is $R_1(i)$ for the lowest i containing the rupture. Recall that, for a rupture of length l, rupture centers may be located at any distance d_c along the fault for which

$$\frac{l}{2} \leq d_c \leq L - \frac{l}{2} \qquad (A10)$$

or, equivalently, the lowest end point of a rupture may be at any distance δ for which

$$0 \leq \delta \leq L - l.$$

The acceleration produced by a rupture of magnitude m and length $l_i(m)$ decreases as the rupture moves along the fault from a maximum acceleration $a_{max}[l_i(m)]$ for a rupture with closest end point at $\delta = 0$ to a minimum acceleration $a_{min}[l_i(m)]$ for a rupture with closest end point at $\delta = L - l(m)$. Therefore, for an acceleration interval such that

$$a_{min}[l_i(m)] \leq a_{j-1} < a \leq a_j \leq a_{max}[l_i(m)]$$

a corresponding interval exists along the fault

$$h_m(j) \leq \delta \leq h_m(j-1)$$

for magnitude m earthquakes in which (closest) end points of ruptures producing these accelerations may be located. Because these ruptures are at a distance $d_m(j) \leq d \leq d_m(j-1)$ from the site,

$$h_m(j) = \sqrt{d_m(j)^2 - P(i)^2} + X(i); \quad [X(i) < 0] \qquad (A11)$$

$$h_m(j-1) = \sqrt{d_m(j-1)^2 - P(i)^2} + X(i). \qquad (A12)$$

Ruptures at distance along the fault $h_m(j) \le \delta < h_m(j-1)$ therefore contribute to accelerations in the range $a_{j-1} < a \le a_j$ at the rate

$$rat[a_j, l_i(m)] = \rho(m) \frac{h_m(j-1) - h_m(j)}{L - l_i(m)} f\ p(i) \quad (A13)$$

where f, $p(i)$ and $\rho(m)$ are defined as in equation A3. For n rupture lengths, given that

$$l_1(m) > l_2(m) ... > l_n(m), \quad (A14)$$

a corresponding section of fault $0 \le \delta \le L - l_i(m)$ may contain closest end points of ruptures and

$$L - l_n(m) > L - l_{n-1}(m) > ... > L - l_1(m). \quad (A15)$$

Now define

$$h_{m,i}(j) = \begin{cases} h_m(j) & \text{if } h_m(j) < L - l_i(m) \\ L - l_i(m) & \text{otherwise,} \quad 1 \le i \le n. \end{cases}$$

The total contribution to accelerations $a_{j-1} < a \le a_j$ is the sum over rupture lengths for magnitude m ruptures:

$$rat(a) = \rho(m) \sum_{i=1}^{n} \frac{h_{m,i}(j-1) - h_{m,i}(j)}{L - l_i(m)} f\ p(i) \quad (A16)$$

where $h_{m,i}(0) = 0$.

Thus, one need only compute one set of distances $h_m(j)$ per magnitude along the fault rather than repeating the process for each rupture length, so that including rupture length variability in the calculation does not substantially increase the computation time.

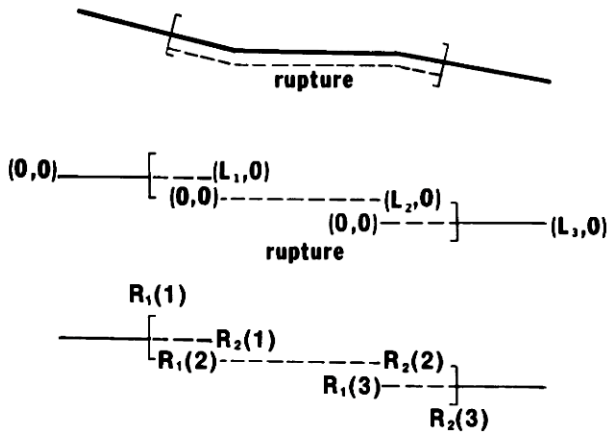


Figure A3. Rupture overlaps three fault segments. Each segment is relocated to lie on the X-axis. End points of the part of the rupture on segment i are $[R_1(i), 0]$ and $[R_2(i), 0]$.

Case 2. The distance from the fault to the site decreases monotonically as one moves a distance d_c along the fault to some point on segment k [$(I_k, 0)$ in the new system] and then increases steadily as one continues to the other end. This corresponds to:

$$X(i) < 0 \text{ for } 1 \le i \le k-1,$$
$$X(k) = I_k = d_c - \sum_{i+1}^{k-1} L(i),$$
$$X(i) > L(i) \text{ for } k+1 \le i \le n.$$

In this case, all ruptures of length $l_i(m)$ with midpoints at a distance δ along the fault

$$d_c - \frac{l_i(m)}{2} < \delta < d_c + \frac{l_i(m)}{2} \quad (A17)$$

overlap the point d_c and hence are regarded as being at the same closest distance from the site. The acceleration at the site produced by these magnitude m ruptures is the acceleration at distance $P(k)$, the perpendicular distance to the k^{th} segment. Because ruptures cannot extend past the end of the fault, the location of possible rupture centers is restricted to a part of the fault and the contribution at $P(k)$ becomes

$$rat[a(j), l_i(m)] = \rho(m) f \sum_{i=1}^{n} \frac{D_{u_i}(m) - D_{l_i}(m)}{L - l_i(m)} p(i), \quad (A18)$$

where

$$D_{u_i}(m) = \min[d_c + \frac{l_i(m)}{2}, L - \frac{l_i(m)}{2}], \quad (A19)$$
$$D_{l_i}(m) = \max[d_c - \frac{l_i(m)}{2}, \frac{l_i(m)}{2}].$$

Now the bookkeeping becomes more complicated, but we can look at all ruptures of length $l_i(m)$ with centers at distance δ on the fault

$$\frac{l_i(m)}{2} \le \delta \le L - \frac{l_i(m)}{2} \quad (A20)$$

and remove those already accounted for, those with centers

$$d_c - \frac{l_i(m)}{2} \le \delta < d_c + \frac{l_i(m)}{2}. \quad (A21)$$

The remaining ruptures may be separated into two sets: those (if any) with centers

$$\frac{l_i(m)}{2} \le \delta < d_c - \frac{l_i(m)}{2} \quad (A23)$$

and those for which

$$d_c + \frac{l_i(m)}{2} < \delta \le L - \frac{l_i(m)}{2}.$$

The last two intervals correspond to ruptures whose closest end points to the site are at δ, where

$$\min[l_i(m), d_c] \leq \delta < d_c \qquad (A24)$$

for one set of ruptures, and at

$$d_c < \delta \leq \max[d_c, L - l_i(m)] \qquad (A25)$$

for the other set. These two sets of ruptures do not overlap and the fault may be regarded as consisting of two distinct segments in which the distance from the rupture to the site decreases (or increases) monotonically as one moves along the fault. Hence, each of these segments may be treated separately as in Case 1, and the accelerations resulting from ruptures along these segments may be added to those already accounted for.

APPENDIX B

Sample Inputs and Outputs

Sample inputs and outputs are provided in this appendix. An explanation of the input variables (file 15) and formats is provided at the beginning of the program listing in Appendix C.

The output file (file 16) gives ground-motion statistics for each site. The site coordinates (longitude, latitude) in decimal degrees appear first, followed by the shortest distance from the site to the closest fault. Then two arrays are given, one for zero attenuation variability and one for attenuation variability σ_a. The columns in each array give for the ground-motion (g.m.) in the i^{th} row:

occ/yr: occurrence rate per year of earthquakes producing ground motions at the site in the range g.m.(i−1) < g.m. ≤ g.m.(i);

exc/yr: exceedance rate per year of g.m.(i);

r(events): the average number of events required to produce an exceedance of g.m.(i),

$$\text{r(events)} = \frac{\text{total yearly events}}{\text{exc/yr of g.m.(i)}};$$

r(yrs): return period in years,

$$\text{r(yrs)} = \frac{1}{\text{exc/yr of g.m.(i)}}.$$

Following the arrays is "total yearly events," the total number of earthquakes from all sources in this run. This total should be nearly the same at all sites; slight differences between sites are due to inaccuracies in calculating fractional areas and assigning proportional seismicity to these areas. Large differences probably indicate an error in the computation.

The concluding lines for each site read

"xxx ext prob=yyy for zzz years",

that is, xxx is the extreme probability that ground-motion yyy will not be exceeded in zzz years.

Included also in this appendix are a listing of a program to read (binary) output file 02 and the output of that program for the sample problem. File 02 contains only the summary information needed for mapping purposes: site coordinates (long, lat) and ground-motions calculated at the specified probability level for the times of interest (for both zero-attenuation variability and one value of σ_a).

File 15--Inputs

```
SEISRISK III Sample Problem
0 20.
.90 3 10 50 250
1.  0 .5 0
123.0 38.0 120.0 38.0
123.00     39.00      120.00     37.00       .100       .100
5 6 5 6
1
3
123.  39.  123.  38
6 12
'watashh79'          8.5        7.6        6.6        5.6        5.2        4.2
3.22        .74        .73        .67        .45        .195       .072
6.43        .64        .62        .53        .36        .135       .047
16.09       .49        .43        .32        .19        .052       .02
32.18       .36        .28        .17        .09        .02        .0052
64.3        .21        .14        .06        .035       .0051      .0013
96.5        .12        .07        .03        .0138      .0023      .00042
160.9       .045       .025       .015       .005       .00083     .0001
321.8       .013       .0076      .0026      .0012      .00021     .00003
643.0       .0034      .0019      .00065     .0003      .00005     .00001
1288.0      .00085     .00047     .00016     .00007     .00001     .00001
2570.0      .00021     .00012     .00004     .00002     .00001     .00001
5140.0      .0001      .00006     .00002     .00001     .00001     .00001
00 100.       -1            zn01
 3   1   1
122.20 37.22 122.00 37.32
123.85 38.97 123.68 39.05
124.12 39.32 123.89 39.40
.106   .281   .74    1.97
6.100 5.500 4.900 4.300
00 65.2       -1            zn02
  2   1   1
120.05 34.09 119.26 34.20
120.40 33.50 119.77 33.39
.031   .073   .172   .405   .955   2.248
7.300 6.700 6.100 5.500 4.900 4.300
98 1.0        -1            zn03 20.
  7   1   2
120.40 34.68 120.00 34.78
120.56 34.93 120.28 35.00
120.78 35.20 120.38 35.39
121.00 35.48 121.00 36.00
121.51 36.00 121.47 36.50
121.82 36.45 121.80 36.80
122.00 36.73 122.21 37.11
  2   2   2
121.80 36.80 121.77 36.88
122.21 37.11 122.20 37.22
.0198 .0641 .2074 .6712
6.100 5.500 4.900 4.300
99 1.0        -1            ft01-1.085 .389 .50
  12   1   1
117.37      34.17      118.09      34.59      118.67      34.79      119.03      34.85
119.26      34.90      119.54      35.06      121.27      36.66      121.69      36.94
122.10      37.27      123.77      39.01      124.01      39.36      124.16      39.93
.0021 .0057 .015   .040
8.500 7.900 7.300 6.700
00 10.        +2            ft02-1.085 .389 .50
  4   1   4
122.40      38.10      122.90      38.66      123.38      39.10      123.94      39.79
  3   2   4
122.60      38.68      123.10      39.20      123.74      39.95
  5   3   4
122.16      38.16      122.46      38.80      122.75      38.98      123.18      39.56
123.40      39.98
  3   4   4
122.50      39.10      122.80      39.25      122.95      39.43
.0072 .0200 .057
7.900 7.300 6.700
99
99
```

File 16--Outputs

```
SEISRISK III Sample Problem
isw=0: new run--no previous results included
extreme probability  0.900
  for exposure times (years)    10   50  250
scale factor for ground motion "box" levels=   1.00
coordinates input in decimal degrees
   coordinates are printed in decimal degrees
variability in attenuation, sigma=  0.50
grid oriented parallel to great circle thru ( 123.00,  38.00),( 120.00,  38.00)
corners of gridded area-upper left= 123.00,   39.00
                       lower right= 120.00,   37.00
longitude increment= 0.1000 (decimal degrees)
latitude increment = 0.1000 (decimal degrees)
gridded region contains   27 rows,    30 cols' including border    3 rows and cols
for this run begin at row   5 end row   6, begin col   5 end col   6
new coordinates (km) gridded area
      upper left=   294.73      0.00; lower right=   144.70   -111.30
sites are also located on   1 line(s)
  3 sites per line
line  1 end points at 123.000,  39.000 and 123.000,  38.000
attenuation function watashh79
                                magnitude
  dist(km)     8.50       7.60       6.60       5.60       5.20       4.20
      3.22   0.74000    0.73000    0.67000    0.45000    0.19500    0.07200
      6.43   0.64000    0.62000    0.53000    0.36000    0.13500    0.04700
     16.09   0.49000    0.43000    0.32000    0.19000    0.05200    0.02000
     32.18   0.36000    0.28000    0.17000    0.09000    0.02000    0.00520
     64.30   0.21000    0.14000    0.06000    0.03500    0.00510    0.00130
     96.50   0.12000    0.07000    0.03000    0.01380    0.00230    0.00042
    160.90   0.04500    0.02500    0.01500    0.00500    0.00083    0.00010
    321.80   0.01300    0.00760    0.00260    0.00120    0.00021    0.00003
    643.00   0.00340    0.00190    0.00065    0.00030    0.00005    0.00001
   1288.00   0.00085    0.00047    0.00016    0.00007    0.00001    0.00001
   2570.00   0.00021    0.00012    0.00004    0.00002    0.00001    0.00001
   5140.00   0.00010    0.00006    0.00002    0.00001    0.00001    0.00001

yrnoc=   100.  iprint=-1 for area zn01
   122.200    37.220    122.000    37.320
   123.850    38.970    123.680    39.050
   124.120    39.320    123.890    39.400
nr of levels of seismicity =  4
zn01 beta= -1.6319
before normalizing to rate/year
occurrences=  1.970000  0.740000  0.281000  0.106000
earthquake rate / year
occurrences=  0.019700  0.007400  0.002810  0.001060
 magnitudes=      4.30      4.90      5.50      6.10
zn01 area=     5457.  sq km, rate/sq km= 0.36101E-05 for mags  4.00- 4.60

yrnoc=    65.  iprint=-1 for area zn02
   120.050    34.090    119.260    34.200
   120.400    33.500    119.770    33.390
nr of levels of seismicity =  6
zn02 beta= -1.4268
before normalizing to rate/year
occurrences=  2.248000  0.955000  0.405000  0.172000  0.073000  0.031000
earthquake rate / year
occurrences=  0.034479  0.014647  0.006212  0.002638  0.001120  0.000475
 magnitudes=      4.30      4.90      5.50      6.10      6.70      7.30
zn02 area=     5212.  sq km, rate/sq km= 0.66157E-05 for mags  4.00- 4.60

yrnoc=     1.  iprint=-1 for area zn03
smooth for earthquake location uncertainty sigma= 20.0
   120.400    34.680    120.000    34.780
   120.560    34.930    120.280    35.000
   120.780    35.200    120.380    35.390
   121.000    35.480    121.000    36.000
   121.510    36.000    121.470    36.500
   121.820    36.450    121.800    36.800
   122.000    36.730    122.210    37.110
   121.800    36.800    121.770    36.880
   122.210    37.110    122.200    37.220
nr of levels of seismicity =  4
zn03 beta= -1.9574
earthquake rate / year
occurrences=  0.671200  0.207400  0.064100  0.019800
 magnitudes=      4.30      4.90      5.50      6.10
zn03 area=    10075.  sq km, rate/sq km= 0.66621E-04 for mags  4.00- 4.60

yrnoc=     1.  iprint=-1 for area ft01
Distance between dummy faults=  0.0
fault  1 of  1
   117.37    34.17,    118.09    34.59,    118.67    34.79,    119.03    34.85,
   119.26    34.90,    119.54    35.06,    121.27    36.66,    121.69    36.94,
   122.10    37.27,    123.77    39.01,    124.01    39.36,    124.16    39.93,
nr of levels of seismicity =  4
ft01 beta= -1.6347
earthquake rate / year
occurrences=  0.040000  0.015000  0.005700  0.002100
 magnitudes=      6.70      7.30      7.90      8.50
fault rupture length parameters a1= -1.085 b1=  0.389  sig1=  0.50

yrnoc=    10.  iprint= 2 for area ft02
Distance between dummy faults=  0.0
fault  1 of  4
   122.40    38.10,    122.90    38.66,    123.38    39.10,    123.94    39.79,
fault  2 of  4
   122.60    38.68,    123.10    39.20,    123.74    39.95,
fault  3 of  4
   122.16    38.16,    122.46    38.80,    122.75    38.98,    123.18    39.56,
   123.40    39.98,
fault  4 of  4
   122.50    39.10,    122.80    39.25,    122.95    39.43,
nr of levels of seismicity =  3
ft02 beta= -1.7455
before normalizing to rate/year
occurrences=  0.057000  0.020000  0.007200
earthquake rate / year
occurrences=  0.005700  0.002000  0.000720
 magnitudes=      6.70      7.30      7.90
fault rupture length parameters a1= -1.085 b1=  0.389  sig1=  0.50
```

```
SEISRISK III Sample Problem
site at long  122.480, lat   38.605
shortest dist to fault=    9.128 km
        zero attenuation variability                    variability in atten, sigma= 0.50
  g.m.    occ/yr     exc/yr  r(events) r(yrs)    g.m.    occ/yr     exc/yr  r(events) r(yrs)
  0.02  1.08967  0.03459    32.5    28.9        0.02  1.08964  0.03462    32.5    28.9
  0.04  0.01039  0.02421    46.4    41.3        0.04  0.01057  0.02405    46.7    41.6
  0.06  0.00559  0.01862    60.4    53.7        0.06  0.00611  0.01794    62.7    55.8
  0.08  0.00627  0.01235    91.0    81.0        0.08  0.00403  0.01390    80.9    71.9
  0.10  0.00241  0.00994   113.2   100.7        0.10  0.00281  0.01109   101.4    90.2
  0.12  0.00220  0.00774   145.3   129.3        0.12  0.00203  0.00906   124.1   110.4
  0.14  0.00092  0.00682   164.8   146.6        0.14  0.00152  0.00754   149.0   132.6
  0.16  0.00112  0.00570   197.2   175.4        0.16  0.00116  0.00638   176.2   156.7
  0.18  0.00101  0.00469   239.9   213.4        0.18  0.00092  0.00546   205.8   183.0
  0.20  0.00077  0.00392   286.8   255.1        0.20  0.00074  0.00473   237.9   211.6
  0.22  0.00083  0.00309   363.4   323.2        0.22  0.00060  0.00412   272.6   242.5
  0.24  0.00048  0.00261   430.7   383.1        0.24  0.00050  0.00362   310.3   276.0
  0.26  0.00029  0.00232   484.3   430.7        0.26  0.00042  0.00320   351.2   312.4
  0.28  0.00021  0.00211   533.3   474.3        0.28  0.00036  0.00284   395.4   351.7
  0.30  0.00018  0.00193   583.9   519.3        0.30  0.00031  0.00254   443.3   394.3
  0.32  0.00013  0.00180   625.8   556.6        0.32  0.00027  0.00227   495.2   440.5
  0.34  0.00015  0.00164   684.4   608.7        0.34  0.00023  0.00204   551.5   490.5
  0.36  0.00018  0.00146   769.8   684.7        0.36  0.00020  0.00184   612.4   544.7
  0.38  0.00013  0.00134   842.0   748.9        0.38  0.00018  0.00166   678.6   603.6
  0.40  0.00008  0.00126   893.0   794.3        0.40  0.00016  0.00150   750.4   667.4
  0.42  0.00007  0.00119   943.4   839.1        0.42  0.00014  0.00136   828.3   736.7
  0.44  0.00024  0.00095  1180.6  1050.1        0.44  0.00013  0.00123   912.9   812.0
  0.46  0.00021  0.00074  1518.3  1350.4        0.46  0.00011  0.00112  1004.7   893.7
  0.48  0.00018  0.00056  2002.5  1781.2        0.48  0.00010  0.00102  1104.5   982.4
  0.50  0.00014  0.00042  2672.0  2376.7        0.50  0.00009  0.00093  1212.8  1078.8
  0.52  0.00014  0.00028  4015.7  3571.9        0.52  0.00008  0.00085  1330.5  1183.4
  0.54  0.00009  0.00019  5936.6  5280.4        0.54  0.00007  0.00077  1458.2  1297.1
  0.56  0.00006  0.00013  8550.0  7604.9        0.56  0.00007  0.00070  1597.0  1420.5
  0.58  0.00012  0.00001 99999.9 99999.9        0.58  0.00006  0.00064  1747.6  1554.4
  0.60  0.00001  0.00000 99999.9 99999.9        0.60  0.00006  0.00059  1911.0  1699.8
  0.62  0.00000  0.00000 99999.9 99999.9        0.62  0.00005  0.00054  2088.4  1857.6
  0.64  0.00000  0.00000 99999.9 99999.9        0.64  0.00005  0.00049  2280.8  2028.7
  0.66  0.00000  0.00000 99999.9 99999.9        0.66  0.00004  0.00045  2489.4  2214.2
  0.68  0.00000  0.00000 99999.9 99999.9        0.68  0.00004  0.00041  2715.5  2415.4
  0.70  0.00000  0.00000 99999.9 99999.9        0.70  0.00003  0.00038  2960.5  2633.3
  0.72  0.00000  0.00000 99999.9 99999.9        0.72  0.00003  0.00035  3225.9  2869.4
  0.74  0.00000  0.00000 99999.9 99999.9        0.74  0.00003  0.00032  3513.3  3125.0
  0.76  0.00000  0.00000 99999.9 99999.9        0.76  0.00003  0.00029  3824.4  3401.7
  0.78  0.00000  0.00000 99999.9 99999.9        0.78  0.00002  0.00027  4160.9  3701.0
  0.80  0.00000  0.00000 99999.9 99999.9        0.80  0.00002  0.00025  4524.8  4024.7
  0.82  0.00000  0.00000 99999.9 99999.9        0.82  0.00002  0.00023  4918.3  4374.7
  0.84  0.00000  0.00000 99999.9 99999.9        0.84  0.00002  0.00021  5343.4  4752.8
  0.86  0.00000  0.00000 99999.9 99999.9        0.86  0.00002  0.00019  5802.5  5161.2
  0.88  0.00000  0.00000 99999.9 99999.9        0.88  0.00002  0.00018  6298.2  5602.1
  0.90  0.00000  0.00000 99999.9 99999.9        0.90  0.00001  0.00016  6833.2  6077.9
  0.92  0.00000  0.00000 99999.9 99999.9        0.92  0.00001  0.00015  7410.3  6591.3
  0.94  0.00000  0.00000 99999.9 99999.9        0.94  0.00001  0.00014  8032.6  7144.8
  0.96  0.00000  0.00000 99999.9 99999.9        0.96  0.00001  0.00013  8703.4  7741.4
  0.98  0.00000  0.00000 99999.9 99999.9        0.98  0.00001  0.00012  9426.1  8384.3
  1.00  0.00000  0.00000 99999.9 99999.9        1.00  0.00001  0.00011 10204.4  9076.5
  1.07  0.00000  0.00000 99999.9 99999.9        1.07  0.00003  0.00008 13359.0 11882.4
  1.14  0.00000  0.00000 99999.9 99999.9        1.14  0.00002  0.00006 17403.6 15480.0
  1.21  0.00000  0.00000 99999.9 99999.9        1.21  0.00001  0.00005 22565.4 20071.3
  1.27  0.00000  0.00000 99999.9 99999.9        1.27  0.00001  0.00004 29122.6 25903.7
  1.34  0.00000  0.00000 99999.9 99999.9        1.34  0.00001  0.00003 37417.7 33282.0
  1.41  0.00000  0.00000 99999.9 99999.9        1.41  0.00001  0.00002 47868.3 42577.4
  1.48  0.00000  0.00000 99999.9 99999.9        1.48  0.00001  0.00002 60977.2 54237.5
  1.55  0.00000  0.00000 99999.9 99999.9        1.55  0.00000  0.00001 77362.9 68812.0
  1.62  0.00000  0.00000 99999.9 99999.9        1.62  0.00000  0.00001 97755.8 86951.0
  1.69  0.00000  0.00000 99999.9 99999.9        1.69  0.00000  0.00001 99999.9 99999.9
total yearly events     1.12427                 total yearly events     1.12426

      zero attenuation variability                    variability in atten, sigma=0.50
  0.900 ext prob =  0.095 for    10 years                  0.105 for   10 years
  0.900 ext prob =  0.280 for    50 years                  0.334 for   50 years
  0.900 ext prob =  0.500 for   250 years                  0.676 for  250 years
ratio  250 yr 0.900 extreme value to   10 yr val=   5.28          6.43
```

```
c   program to read summary file02 created by seisrisk iii
c
c  file 02 contains for each site the level of ground motion
c  that has probability "perct" of not being exceeded during
c  "jtim" years for each of "ntim" values of "jtim".
c  ground motion values are given first for "zero variability in
c  attenuation" for all times "jtim", then for "variability
c  in attenuation  sigma = sd" for all values of "jtim".
c
c  first record contains:
c   irow1,irow2,icol1,icol2 = first and last rows,  first and last
c           columns in gridded area containing sites for which
c           ground motions were calculated in seisrisk iii.
c   indvpt = total number of sites on lines
c   ntims = number of exposure times
c   jtims = exposure times in years
c   sd = standard deviation in attenuation variability.
c   perct: ground motion calculated has probability "perct"
c       of not being exceeded during "jtim" years
c
c   all records after first:
c   xlong,ylat = longitude,  latitude of site in decimal degrees
c   sol(i) = ground motion at (xlat,ylong) corresponding to
c  "perct" probability and jtim(k) years where
c      i=1,2,.....ntim,    k=i  for zero attenuation variability
c      i = ntim+1, ntim+2.....2*ntim,   k=i-ntim
c                    for variability in attenuation sigma = sd
c
      dimension sol(48),jtim(24)
      read (2) irow1,irow2,icol1,icol2,indvpt,ntim, (jtim(jv),jv=1,
     &ntim),perct,sd
      ntim2=2*ntim
      write(44,10),(jtim(jv),jv=1,ntim),(jtim(jv),jv=1,ntim)
   10 format(30x'exposure times'//  long     lat '6i9)
c    format for printout will depend on scale used for solution
   20 read (2,end=40) xlong,ylat,(sol(jt),jt=1,ntim2)
      write (44,30),xlong,ylat,(sol(jt),jt=1,ntim2)
   30 format(1h 2f7.3,6f9.3)
      go to 20
   40 call exit
      end
```

The following are the summarized results that were written to file 02 for the sample problem

		no acceleration variability			sigma= 5 in log acceleration		
long	lat	10yr	50yr	250yr	10yr	50yr	250yr
122.48	38.61	0.095	0.280	0.500	0.105	0.334	0.676
122.35	38.61	0.079	0.224	0.772	0.087	0.295	0.871
122.48	38.51	0.104	0.277	0.418	0.115	0.333	0.614
122.35	38.51	0.086	0.224	0.714	0.096	0.301	0.802
123.00	39.00	0.129	0.351	0.558	0.138	0.422	0.784
123.00	38.50	0.239	0.398	0.454	0.210	0.520	0.834
123.00	38.00	0.219	0.443	0.499	0.203	0.570	0.925

APPENDIX C

Listing of Computer Program SEISRISK III

```
c                                                                       + (f11,ph1)
c                                                                       1
c                                                                     (x1,y1) 1
c                                                                             1
c                                                                               1
c                                                                                 1
c                                                                                   1
c                                                                                     1
c                                                                                       1
c                                                                                 (f12,ph2)+
c                                                                                         (x2,y2)
c
c                                                                     area within dots represents felt region
c                                                                     '1' s represents line joining (x1,y1) (x2,y2)
c                                                                     phinc (long,lat) increment in degrees (in new coordinate
c                                                                       system) for which risk is to be computed
c
c                                                                   7. irow1, irow2, icol1, icol2   (free field)
c                                                                       starting and ending rows and columns for this run
c                                                                       accelerations computed for sites in these rows
c                                                                       and columns--may be subset of seismic felt region defined
c                                                                       in input 6.   Also irow1=irow2=icol1=icol2=0 permitted
c                                                                       if only individual sites on lines (input 8) are selected.
c                                                                   8. indv (free field)
c                                                                       number of line segments containing individual sites at
c                                                                       which acceleration is to be computed; zero if
c                                                                       only fixed grid is used.
c                                                                     If indv > 0 read next inputs, otherwise skip to input 9.
c                                                                     nvs (free field)
c                                                                       number of sites per line segment
c                                                                     xe1,ye1,xe2,ye2 (free field--indv lines, one pair per line)
c                                                                       end points (long, lat) of line segment:  sites will be
c                                                                       evenly spaced on line in new coordinate system
c                                                                     NOTE: NO SMOOTHING OF ACCELERATIONS AT SITES ON LINE FOR
c                                                                     ACCELERATIONS FROM EARTHQUAKES WITHIN AREAL SOURCE ZONES.
c                                                                     CALCULATIONS FOR UNIFORM SEISMICITY ONLY AT THESE SITES.
c                                                                   9. jant,mdis (free field)
c                                                                       jant=no of magnitudes for which acceleration is
c                                                                         tabulated as a function of distance   (maximum 8)
c                                                                       mdis=number of distances in attenuation table (maximum 20)
c                                                                  10. nam, tm (jant values of tm)
c                                                                       nam=identifier up to 10 characters-for example 'schn-seed'
c                                                                         identifies attenuation curves used
c                                                                       tm=magnitude for which table of distance versus acceleration
c                                                                         values follows  (free field)
c                                                                     ********magnitudes must be in descending order***
c                                                                  11.  rtab(i), (atab(i,j),j=1,jant)   (mdis lines, i=1,mdis)
c                                                                       rtab(i)=ith tabular distance in kilometers from earthquake source
c                                                                       atab(i,j)=mean peak acceleration (or other ground motion parameter
c                                                                                 at i th distance for j th magnitude.
c
c                                                                     source area inputs:
c
c                                                                     each source deck -- 
c                                                                  15. num, yrnoc, iprint, totl, dumid, als, bls, sigls
c                                                                     format(i2, f10.0, i2, f10.0, a4, 2f6.2, f5.2)
c                                                                     num=0 for seismic source areas if no averaging for earthquake
c                                                                           location uncertainty is to be done at end of ground
c                                                                           motion computations for this source zone
c                                                                     num=98 for seismic source zones if averaging for earthquake
c                                                                           location uncertainty is to be done at end of computations
c
c                    SEISRISK III
c
c     computer program for determining ground motions
c          for use in seismic hazard mapping
c
c     ******Allows locations of future earthquakes in seismic
c       source zones to have a normal distribution******
c
c     SEISRISK III is a revision of SEISRISK II, documented in Open
c        File Report 82-293.
c
c       See model description of SEISRISK III and additional
c         documentation in USGS Bulletin 1772
c
c     files:
c        file 15:  inputs
c        file 16:  outputs
c        file 02:  summary for plots--see text
c        file 03:  accelerations saved on file 03 for use in next
c                  run if next run is a continuation
c
c                        inputs
c
c  1. title-up to 80 characters   (free field)
c  2. isw, sigmax   (free field)
c       isw=0 new run (usual case)  isw=1 restart or continue
c             from previous run
c       sigmax=maximum standard deviation (km) for earthquake
c              location variability in seismic source zones for
c              this run.
c  3. prob,ntims, jtim(1),jtim(2),....jtim(ntims)
c     acceleration is sought for which there is probability prob of
c     not being exceeded in jtim(1), jtim(2), ....jtim(ntim)
c     years
c     prob = extreme probability in decimal
c     ntims = number of times for which calculation is done
c     jtim(i)  =  durations  (years) for which extreme motions are to be
c           calculated at the prob probability level. 1 <= i <= ntims
c  4. scale, dsw, sd, inos  (free field)
c        scale=scaling factor for ground motion boxes
c        scale=1 for motions .02 to 1.0--scale accordingly
c        dsw:  =1 if inputs are degrees and minutes
c              =0 if inputs are decimal degrees
c        sd=standard deviation in log acceleration
c          for acceleration variability around mean value
c        inos=1: divide magnitude interval in half and do calculations
c             at twice as many magnitudes (Assumes Gutenberg-Richter
c             b-value is same for all intervals.)
c        inos= any number other than 1: do calculations for original
c             magnitude intervals
c  5.  x1,y1, x2,y2 (long,lat) in decimal degrees (free field)
c     transform great circle thru (x1,y1), (x2,y2), (0,0)
c     to equator for new coordiate system
c  6.  f11,ph1,f12,ph2,phinc
c     (f11,ph1) upper left   (f12,ph2) lower right  (long, lat)
c     corners of seismic felt region for risk computation
c     rectangular region with sides parallel to arc
c     of great circle thru (x1,y1), (x2,y2) (0,0) --defined by 5.
c     other two sides perpendicular and thru given end points
c
```

```
c           for this source zone
c         =99 for 1st fault set
c         =0 for all fault sets after 1st
c         =99 at end of computation
c     yrnoc=number of years over which the earthquake occurrences
c               take place
c     iprint = -1  no statistics for this (intermediate) set of
c                  occurrences
c            =1 statistical calculations and printout (on unit 16)
c            =2 same as=1 plus summary file for plot, etc (unit 2)
c            =3 omit printout; do summary file for plot (unit 2)
c    totl ignored here but used for faults
c    dumid = four character identifier for source zone or fault
c    als=earthquake location uncertainty standard deviation in km;
c        current value of als is ignored when num=0
c    bls, sigls ignored here but used for faults
c
c  2s. jseg,ifr,itot
c    jseg=num of pairs of quadrilaters1 corner points in this source
c        set (jseg-1 quads).
c    seismicity to be apportioned by fractional area among itot sources.
c     ifr  = set number (ifr =1,2...itot in sequence)
c     itot=10 max;     total quad pairs for itot sources maximum 50
c  3s. jseg quadrilateral endpoint lines (two points per line):
c    quad corner points (left long, left lat), (right long, right lat),
c    in decimal degrees (if dsw=0)  degrees, minutes (if dsw=1)
c
c      x1,y1 _____________ x2,y2
c           1             1
c           1             1
c           1             1
c           1             1
c      x3,y3 _____________ x4,y4
c            1           1
c             1         1
c              1       1
c               1     1
c                1   1
c                 1_____1
c           x5,y5         x6,y6
c
c    jseg=3 in this example:  quad endpoint pairs are
c                   (x1,y1)---(x2,y2)
c                   (x3,y3)---(x4,y4)
c                   (x5,y5)---(x6,y6)
c     subregions of a set are defined as shown
c     subregions of one set are separate from those of other sets
c     repeat 2s, 3s for itot sources
c   4s. noc(l)  l=1,lev  (lev=12f6.2)   (lev=12 maximum)
c     numbers of events expected in yrnoc years in each
c           magnitude interval for earthquake occurrences
c   5s. fm(l), l=1,lev  (12f6.2)
c     fm(l)=center of magnitude interval for which noc(l) occur
c
c      repeat 1s. thru 5s. for remaining sources
c
c         fault inputs:
c
c   1f.  num, yrnoc, iprint, totl, dumid, als, bls, sigls
c     parameters defined as in 1s. above
c   **note num=99 for first fault, =0 thereafter for successive faults
c     als,bls,sigls rupture length parameters
c     if als, bls non zero:
c         length=10**(als +bls*m +fr*sigl)
c         where m=magnitude;    sigl=standard deviation (in log length)
c     fr=(normally) 5 values in range (-2,+2)  if sigl non zero
c     fr=0 (1 value)--mean rupture length only if sigl=0
c     if als,bls,sigls=0 (or blank) previous values are used
c     if no previous values input, default values are used:
c         als=-1.085     bls=.389    sigls=.52
c     totl=distance between faults if this set of faults is a set
c          of "dummy" faults  used to approximate a uniform field
c          of faults.
c      totl=0 if individual fault or a number of well defined faults
c          are used.
c      totl determines whether ground motions for a site are to
c          be smoothed in distance or smoothed in magnitude
c          (smoothed in magnitude if totl=0)
c          (smoothed in distance (totl nonzero).
c  2f. jseg, ifr, itot  (free field)
c     jseg=num of fault segment end points to be connected
c         into single fault    (jseg-1 segments)
c     ifr, itot as defined in 2s
c  3f. (xl(i,i(fr),yl(i,i(fr), i=1,jseg)   (free field)
c        xl=long  yl=lat (degrees)  (ifr=fault number)
c      jseg=24 max          itot=26 max
c  4f. noc(l)   same as in 4s.
c  5f. fm(l) same as 5s.
c     repeat 2f, 3f for itot faults
c
c     repeat 1f. thru 5f. for remaining faults
c
c     end with num=99 (omit other inputs on line)
c * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      character  idsw*16, nam*10,tytle*80,dumid*4
       data imaxsq/2500/
      common/cc/dumfac,fac(2500)
      dimension idsw(2),ibr(10),lsub(26),dtot(26)
      dimension xpt2(500),ypt2(500),xpt1(500),ypt1(500)
      dimension prls(10),rlls(10,25)
      dimension nrlsv(25),prlrl(10,25),rll(10,25)
      dimension disl(50),angl(50)
      common/ata/ata(20,25)
      common/inout/mdum(4),stsize
      common/slinds/sld1(4),perpa(4),sld2(4),sllq(4),
     & mina(4),perpaq(4),sldum2(8)
      common/extra/isid,shdis,distl(24),inear
      common /xlyl/xll(4),yl(4),xr(4),yr(4),xc(4),yc(4)
      common/itabs/dtab(105,25),maxa(25)
      common/magdis/jent,mdis
      dimension tmdif(10),atab(20,25),tm(10),atadif(20,25)
      common/rtb/rtab(20)
      common/xyarea/xsav(2,50),ysav(2,50),sarea(50)
c*******need maxdim=dimension of reg & of rawbin(maxdim,1600)
      data maxdim,maxar/55,1600/
      dimension regg(1600)
      dimension reg(55),rawbin(55,1600),  smobin(55,1600)
      common/rdis/rdis
      common/wds/max,maxp,maxsp
      common/ext/prob,ex,ex1
      common/tims/ftim(20),jtim(20),ntims,nwt,iprint
```

```
      data iscom/500/
c*** iscom = dimension of following sin,cos arrays
      common/sincos/sinx(500),cosx(500),siny(500),cosy(500)
      common/quad/maxq,ac(105),dumlog,aclog(105)
      dimension x(4),y(4),noc(24),fm(25)
      common/linps/dist(24,26),disq(24,26),jseg,jsegm,
     & x1(24,26),yp(24,26),coa(24,26),sia(24,26)
      common/lindis/dls(24),perp(24),min(24),persq(24)
      dimension jm(26), disrem(24)
      common/accel/gm(25),exdif(101),atah(20,25),aclim(2)
c arrays on previous line inserted for spreading out accelerations
      real noc
      integer dsw
      data linset,line,nrls/0,0,0/
      data idsw/'decimal degrees ','degrees, minutes'/
      data r,pi2,pi,pih/6378.,6.2831854,3.1415927,1.5707963/
      rad=pi/180.
c     input on unit 15
      rewind 15
c     saved on unit 3 for later use or continuation
      rewind 3
      allerr=0.
      dlevsv=0.
      flevst=0.
      levno=0.
      tot5=0.
      read(15,20) tytle
   20 format(a80)
      write(16,30) tytle
   30 format('1'a80)
c  isw tells whether to include data from a previous run, zero if no.
c  sigmax=maximum value of sigma (km, in earthquake location)
      read(15,*)isw,sigmax
      if(isw.eq.0) write(16,40)
   40 format(' isw=0: new run--no previous results included')
      if(isw.eq.1) write (16,50)
   50 format(' isw=1: run continuation; add to previous results')
      if(isw.eq.0) go to 70
      idrc=1
   60 read(3,end=70) (smobin(iqrp,idrc),iqrp=1,maxdim)
      idrc=idrc+1
      go to 60
   70 rewind 3
c    prob=extreme probability in decimal
c    ntims are number of 'jtims' for which calculation is done
c    jtims are the durations for which the extreme motions are
c    to be calculated at the prob extreme probability level
      read (15,*) prob, ntims, (jtim(jv),jv=1,ntims)
      if (prob.lt.1.) go to 90
      write(16,80) prob
   80 format(' prob ='f8.3' must be decimal less than one')
      call exit
   90 continue
      write (16,100) prob,(jtim(jv),jv=1,ntims)
  100 format(' extreme probability 'f6.3
     &/'   for exposure times (years) '2(10i5/))
      ex=-alog(prob)
      ex1=alog(ex)
      do 110 jv=1,ntims
  110 ftim(jv)=jtim(jv)
c
      read(15,*) scale,dsw,sd,inos
      write(16,120) scale,idsw(dsw+1),idsw(dsw+1),sd
  120 format(' scale factor for ground motion "box" levels='
     &f7.2/' coordinates input in ' a16,
     &/'    coordinates are printed in 'a16,
     &/' variability in attenuation, sigma='f6.2)
c       scale=multiplicative factor for acceleration levels
c       dsw=0 if input is decimal degrees; =1 for degrees,minutes
c       sd=standard deviation for log acceleration
c   read (x1,y1), (x2,y2) =long,lat in dec. degrees of 2 points
c   to transform to equator
c       (x1,y1) to (0.,0)(x2,y2) to (dist. between points,0)
  130 read(15,*) x1,y1,x2,y2
      write (16,140) x1,y1,x2,y2
  140 format(' grid oriented parallel to great circle thru ('
     & f7.2','f6.2'),('f7.2','f6.2')')
c  convert to decimal degrees unless dsw=0.
      if(dsw.ne.0) call cond(x1,x2,y1,y2)
      x1=x1*rad
      x2=x2*rad
      y1=y1*rad
      y2=y2*rad
c   transform to equator
      call inieqr(x2,y2,x1,y1)
c
      call toeqr(x1,y1,x1t,y1t)
      call toeqr(x2,y2,x2t,y2t)
c  read in upper left and lower right corners of affected rectangle
c  -- degrees, minutes ie 20.30=20 degrees,30 minutes (dsw=1)
c       (decimal degrees 20.50 = 20 1/2 degrees (dsw=0) )
c  opposite corners of gridded area (in latitude and longitude)
c  gridded area becomes rectangle surrounding new equator.
c  sites are at uniform increments in latitude and longitude
c  within this gridded area in new coordinate system.
c  in order to limit affected area you can specify
c   beginning and ending rows and columns as irow1 and 2
c   and icol1 and 2 (next input line)
c       ph1=phi=latitude
c       fl1=lambda=longitude
c
      read(15,*) fl1,ph1,fl2,ph2,phinc
      write (16,160) fl1,ph1,fl2,ph2
  160 format(' corners of gridded area-upper left='f7.2','f7.2,
     &/23x' lower right='f7.2','f7.2)
c
c  convert to decimal degrees (unless dsw=0)
      if(dsw.eq.0) go to 170
      call cond(ph1,fl1,ph2,fl2)
      call condec(phinc)
  170 continue
c
c     determine number of subregions in map area and set up subscripting
c     irow=no of rows
c     icol=no of cols
      write(16,180) phinc,phinc
  180 format(' longitude increment='f7.4' (decimal degrees)'
     &/' latitude increment ='f7.4' (decimal degrees)')
      phinc=phinc*rad
      flinc=phinc
```

```
      dphinc=phinc*r
      dphinch=dphinc/2.
      dflinc=flinc*r
      fl1=fl1*rad
      ph1=ph1*rad
      fl2=fl2*rad
      ph2=ph2*rad
c     transform to equator and x-y rectangular coordinate system
c     convert to kilometers
      call toeqr(fl1,ph1,flam1,phi1)
      call to eqr(fl2,ph2,flam2,phi2)
      if(phi1.gt.0) go to 210
      if(phi2.gt.0) go to 220
c     apparent error in inputs or xformataion
  190 write(16,200)
  200 format(' probable input error--felt region upper left,'
     &' lower right'/' must be on opposite sides of new equator')
      call exit
  210  if(phi2.le.0) go to 230
      go to 190
  220 phs=phi1
      phi1=phi2
      phi2=phs
      fls=flam1
      flam1=flam2
      flam2=fls
c     have proper orientation--continue
  230 ph1=phi1
      if(flam1.lt.0) go to 240
      fl1=flam1
c     find total number of cols
      go to 260
  240 write(16,250)flam1,flam2
  250 format(' apparent error in xforming lat and long limits'
     & 'to eqr--'/'  flam1='f9.3'  fl2='f9.3)
      call exit
  260 continue
      ph1r=ph1*r
      phi2r=phi2*r
      fl1r=fl1*r
      flam2r=flam2*r
c   adjust to include area beyond edges for later smoothing
c     determine nbord=number of additional rows and columns at
c      which ground motions must be calculated to account for
c      earthquake location variability.
      nbord=2.*sigmax/dphinc+.01
      if(sigmax.eq.0.) go to 280
      if(nbord.ge.2) go to 280
      write (16,270) dphinc,sigmax,nbord
  270 format(' increments between sites ='e12.5' too large for'
     & 'sigmax='e12.5' nbord='i5)
      call exit
  280 ph1=ph1+nbord*phinc
      if (flam2.gt.fl1) flinc=-flinc
      fl1=fl1+nbord*flinc
      irow=(ph1-phi2)/phinc+1.01+nbord
      icol=(fl1-flam2)/flinc+1.01+nbord
      write(16,290)irow,icol,nbord
  290 format(' gridded region contains'i4,' rows,  'i4' cols'
     &' including border 'i4' rows and cols')
      read(15,*)irow1,irow2,icol1,icol2
c
c     limits rows and columns of felt points for which calculation is
c     to be made.
c
      write(16,300)irow1,irow2,icol1,icol2
  300 format(' for this run begin at row 'i3' end row 'i3
     & ', begin col 'i3' end col 'i3)
c    for earthquake location variability, compute at more sites
c    calculate at additional rows and cols
      irow2=irow2+2*nbord
      icol2=icol2+2*nbord
      icols=icol2-icol1+1
      irows=irow2-irow1+1
      if(irow1.eq.0) irows=0
      if(icol1.eq.0) icols=0
      if(irow1.le.irow2) go to 320
      write(16,310) irow1
  310 format(' irow1 ='i4'  too large--error stop')
      call exit
  320 if(icol1.le.icol2) go to 340
      write(16,330) icol1
  330 format(' icol1 ='i4'   too large--error stop')
      call exit
  340 if(icols*irows.le.maxar) go to 370
  350 write(16,360)irows,icols,indvpt,maxdim
  360 format(' irows='i4' icols='i4' pts on line='i4
     &' exceeds maxdim of 'i4)
      call exit
  370 continue
c   read in number of line segments containing sites at which
c   calculation is to be done  (in addition to or instead of
c   computing for sites on a fixed grid)
c   zero if no line segments
      uply=ph1*r
      uplx=fl1*r
      boty=phi2*r
      botx=flam2*r
      write(16,380) uplx,botx,uply,boty
  380 format(' new coordinates (km) gridded area'
     &/'       upper left='2f9.2'; lower right='2f9.2)
      read(15,*)indv
      write(16,390) indv
  390 format(' sites are also located on 'i3' line(s)')
      if (indv.eq.0) go to 460
c     calculate accelerations at nvs sites on each of indv lines
c     read in end point pairs (long,lat) xe1,ye1,xe2,ye2
c     for each of indv lines
c     nvs sites are evenly spaced on line*
c     nvs=2 gives accelerations at end points of line;
c     nvs=3 acceleration at 2 end points plus center of line,etc.
      read(15,*) nvs
      write(16,400) nvs
  400 format(i4' sites per line')
      indvpt=0
      do 450 i=1,indv
      read(15,*) xe1,ye1,xe2,ye2
      write(16,410) i,xe1,ye1,xe2,ye2
  410 format(' line 'i2' end points at 'f7.3','f7.3' and 'f7.3
     &','f7.3)
```

```
      if(dsw.ne.0) call cond(xe1,xe2,ye1,ye2)
      xe1=xe1*rad
      xe2=xe2*rad
      ye1=ye1*rad
      ye2=ye2*rad
      call toeqr(xe1,ye1,xo1,yo1)
      call toeqr(xe2,ye2,xo2,yo2)
      fr=0.
      if (nvs.gt.1) frinc=1./(nvs-1)
      do 440 iq=1,nvs
      indvpt=indvpt+1
      if(indvpt.le.500) go to 430
      write(16,420)
  420 format(' more than 500 single points requested as sites'
     & 1x,'do 500')
      go to 460
  430 xl1=xo1+fr*(xo2-xo1)
      yp1=yo1+fr*(yo2-yo1)
      cosy(1)=cos(yp1)
      siny(1)=sin(yp1)
      sinx(1)=sin(xl1)
      cosx(1)=cos(xl1)
      call backw(1,1,xpt1(indvpt),ypt1(indvpt))
      xpt2(indvpt)=xl1*r
      ypt2(indvpt)=yp1*r
      fr=fr+frinc
  440 continue
  450 continue
  460 if(icols*irows+indvpt.gt.maxar) go to 350
      nrc=icols*irows
c
c     jent=number of magnitude entries in table of acceleration as a
c        function of magnitude and distance
c     mdis=number of distance entries
      read(15,*) jent,mdis
      if (jent.le.10) go to 480
      write (16,470) jent
  470 format(' jent='i3' too large--max 10 allowed')
      call exit
  480 if (mdis.le.20) go to 500
      write(16,490) mdis
  490 format(' mdis='i4'  too large--max 20 distances allowed')
      call exit
  500 continue
c     read identifier and magnitudes for which acceleration versus
c     distance table follows.  magnitudes must be in decreasing order
c
      read(15,*)nam,(tm(i),i=1,jent)
c    print magnitude headings for attenuation function.
      write (16,510) nam
  510 format(' attenuation function 'a10/30x'magnitude')
      write(16,520) (tm(i),i=1,jent)
  520 format('   dist(km) 'f7.2,10f10.2)
c  read in table of acceleration vs distance values for set
c  of magnitudes (distance increasing in table)
      do 580 m=1,mdis
      read(15,*) rtab(m),(atab(m,j),j=1,jent)
      if(atab(1,1).lt.10) go to 540
      write (16,530) rtab(m),(atab(m,j),j=1,jent)
  530 format(1h f10.2,10f10.2)
      go to 560
  540 write(16,550) rtab(m),(atab(m,j),j=1,jent)
  550 format(1h f10.2,8f10.5)
c  take logs of tabular values since interp. is in log acc. etc.
  560 do 570 j=1,jent
  570 atab(m,j)=alog(atab(m,j))
  580 continue
c
c      define limits for acceleration boxes:    accelerations in the
c      range ac(j-1) < ac < ac(j) will be collected the jth box.
c      max=number of boxes;  acceleration values too large for
c      any of the box levels correspond to maxp=max+1
c      as here constituted boxes have limits ranging from
c     .02 to 1.0
c     For motions exceeding 1(g)., provide
c     appropriate distance vs ground table and select 'scale'
c     for multiplying boxes to get consistent range
c     or rewrite and recompile subroutine BOX
  600 call box(scale,sd)
c
c     basic step size for distances in source areas
      do 610 i=1,maxsp
  610 aclog(i)=alog(ac(i))
c     number of digits after decimal for print of
c     accelerations (subroutine out) determined by scale
c
c     identify rows and cols for file02
  620 ir2=irow2-2*nbord
      icl2=icol2-2*nbord
      write(2) irow1,ir2,icol1,icl2,indvpt,ntims,(jtim(jv),jv=1,ntims),
     &prob,sd
c     create table of sines and cosines for later back transformation
c     of felt points
      if(icol1.eq.0) go to 680
      yp1=(ph1-(irow1-1.)*phinc)
      xl1=(fl1-(icol1-1.)*flinc)
      irow2p=irow2+1
      icol2p=icol2+1
      if(irow2p.le.iscom) go to 650
  630 write(16,640) irow2p,icol2p,iscom
  640 format (i5' rows,' i5' cols needed in sincos arrays,'
     &' dimension='i4/' need to increase dimension')
      call exit
  650 if(icol2p.gt.iscom) go to 630
      do 660 ira=irow1,irow2p
      cosy(ira)=cos(yp1)
      siny(ira)=sin(yp1)
  660 yp1=yp1-phinc
      do 670 ii=icol1,icol2p
      cosx(ii)=cos(xl1)
      sinx(ii)=sin(xl1)
  670 xl1=xl1-flinc
  680 read(15,700)num,yrnoc,iprint,totl,dumid,als,bls,sigls
      numsav=num
  690 format('0yrnoc= 'f6.0,' iprint='i2,' for area 'a4)
  700 format(i2,f10.0,i2 ,f10.2,a4,2f6.2,f5.2)
c
      if(num.ne.99) go to 710
      if(line.eq.1) go to 2950
c  yrnoc=0 here if only source areas, no faults included
```

```
      if(yrnoc.eq.0.) go to 2950
      line=1
  710 write(16,690)yrnoc,iprint,dumid
  720 if(line.eq.1) go to 830
      if(num.eq.98) write (16,705) als
  705 format(' smooth for earthquake location uncertainty sigma='
     1 f5.1)
      nbr=0
      ist=0
      iend=0
c  read identifiers for source area inputs
  730 read(15,*)jseg,ifr,itot
c     write(16,740) ifr,itot
  740 format(' source 'i2' of 'i2)
c     jseg=number of pairs of quadrilateral end points in this
c        single source
c     itot=number of sources in this source area
c     ifr=identifies which source,  ifr=1,2,....itot
      if (itot.le.10) go to 760
      write(16,750) itot,jseg,ifr
  750 format('  itot ='i4'  to large--max 10  jseg='i4
     & ' ifr='i4)
      call exit
  760 ist=iend+1
      iend=ist+jseg-1
      if (iend.le.50) go to 780
      write(16,770)
  770 format('  too many quadrilaterals in source--max 50')
      call exit
c  read in boundaries for quadrilaterals for all subareas
c  in this source
c      read limits of seismic area x(i)=lambda(i),   y(i)=phi(i)
c      upper left(x,y) then  upper right(x,y) --1st quadrilateral
c      2nd line   lower left,lower right 1st quadrilateral=
c      2nd card   upper left,upper right   2nd quadrilateral,  etc.
c
  780 do 810 ii=ist,iend
      read(15,*)  (xsav(i,ii),ysav(i,ii),  i=1,2)
  790 format(4f6.0)
  800 format(1h 4f10.3)
      write(16,800)(xsav(i,ii),ysav(i,ii),i=1,2)
  810 continue
      if(ifr.eq.itot) go to 820
      nbr=nbr+1
      ibr(nbr)=iend
      go to 730
  820 num=iend
      go to 960
c     fault line read
  830 j=1
      write(16,840) totl
  840 format(' Distance between dummy faults='f5.1)
      dsum=0.
c    fault inputs
  850 read(15,*) jseg ,ifr,itot
      write(16,860) ifr,itot
  860 format(' fault 'i2' of 'i2)
c    jseg=number of end points of connected segments
c       for the current fault
c       jseg=one plus number of segments
c     itot=number of distinct faults in this zone
c     ifr identifies current fault,  ifr=1,2,...itot
      if (jseg.le.24) go to 890
  870 write (16,880) jseg,itot
  880 format('  jseg='i4' (max 24) itot='i4' (max 26)-stop')
  890 if(itot.gt.26) go to 870
c     read in end points long,lat of each segment
      read(15,*)(xl(i,j),yp(i,j),i=1,jseg)
      write(16,900)(xl(i,j),yp(i,j),i=1,jseg)
  900 format(1h 4(f10.2,f8.2,',',))
      jm(j)=jseg
      do 920 i=1,jseg
c     convert to decimal degrees unless dsw=0
      if(dsw.eq.0) go to 910
      call condec(yp(i,j))
      call condec(xl(i,j))
  910 continue
c    transform to equator--rectangular coordinates
      yin=yp(i,j)*rad
      xin=xl(i,j)*rad
      call to eqr(xin,yin,xout,yout)
      xl(i,j)=xout*r
      yp(i,j)=yout*r
  920 continue
c    compute line parameters, lengths for fault segments
      jsegm=jseg-1
      dtot(j)=0.
      do 930 i=1,jsegm
      xdelta=xl(i+1,j)-xl(i,j)
      ydelta=yp(i+1,j)-yp(i,j)
      disq(i,j)=xdelta*xdelta+ydelta*ydelta
      dist(i,j)=sqrt(disq(i,j))
      coa(i,j)=(xl(i+1,j)-xl(i,j))/dist(i,j)
      sia(i,j)=(yp(i+1,j)-yp(i,j))/dist(i,j)
  930 dtot(j)=dtot(j)+dist(i,j)
      dsum=dsum+dtot(j)
      if(ifr.eq.itot) go to 950
  940 j=j+1
      go to 850
c     read in num of occurrences for each magnitude (12f6.2)
c     both area and fault input
  950 jtot=j
  960 read (15,1170) (noc(i),i=1,12)
      lev=12
  970 if(noc(lev).ne.0.0) go to 990
      lev=lev-1
      if(lev.gt.0) go to 970
      write(16,980)
  980 format(' apparent input err for no of occurrences at each level')
      go to 2990
  990 continue
      write(16,1000)lev
 1000 format(' nr of levels of seismicity = 'i2)
c
c  read in corresponding mag. interval center points. (lev of them)
c
 1010 format(' before normalizing to rate/year')
 1020 format( ' earthquake rate / year')
      read(15,1170)(fm(i),i=1,lev )
c  magnitudes must be evenly spaced and in increasing order.
```

```
c  if not entered in increasing order, program reverses order
c  of magnitudes and corresponding earthquake occurrences
c  at each magnitude.
c  Don't confuse this with the original table of ground
c   motions as a function of magnitude and distance.
c   The magnitudes in that table must be read in decreasing
c   (rather than increasing) order.
      if(fm(1).lt.fm(2)) go to 1040
      if(lev.eq.1) go to 1040
      levh=lev/2
      ll=lev
      do 1030 l=1,levh
      sav=fm(l)
      fm(l)=fm(ll)
      fm(ll)=sav
      sav=noc(l)
      noc(l)=noc(ll)
      noc(ll)=sav
 1030 ll=ll-1
 1040 continue
      if(lev.gt.1) go to 1050
c     we cannot calculate beta from numbers of earthquakes
c     if only one level--so give a default beta so we
c     can later spread out
      beta=-2.0
      fm(2)=fm(1)+.6
      delm=.6
      go to 1060
 1050 continue
      dlev=abs(fm(1)-fm(2))
      beta=alog( noc(2)/noc(1) )/dlev
 1060 write(16,1070)dumid, beta
 1070 format(1x,a4,' beta='f8.4)
      if(yrnoc. eq. 1.) go to 1090
      write(16,1010)
      write(16,1100) (noc(l),l=1,lev)
c     this is a normalization to rate per year
      do 1080 lm=1,lev
      noc(lm)=noc(lm)/yrnoc
 1080 continue
 1090 write(16,1020)
      write(16,1100)(noc(l),l=1,lev)
 1100 format(' occurrences='12f10.6)
      write(16,1110)(fm(l),l=1,lev)
 1110 format('  magnitudes='12f10.2)
      summoc=0.
      do 1120 lm=1,lev
 1120 summoc=summoc+noc(lm)
c     evaluate at twice as many magnitudes as initially read
c     in if inos=1
      if(inos.ne.1)  go to 1140
      dlev=(fm(2)-fm(1))/2.
      zac=exp(abs(beta)*dlev)
      noc(1)=noc(1)*zac/(1.+zac)
      summoc2=noc(1)
      lev=2*lev
      fm(1)=fm(1)-dlev/2.
      do 1130 ll=2,lev
      fm(ll)=fm(ll-1)+dlev
      noc(ll)=noc(ll-1)/zac
 1130 summoc2=summoc2+noc(ll)
      write(16,1150) summoc,summoc2
      write(16,1100) (noc(l),l=1,lev)
      write(16,1110) (fm(l),l=1,lev)
 1140 continue
 1150 format(' total eqs='2e12.5)
 1160 continue
 1170 format(12f6.2)
c
      if (line.eq.0) go to 1250
c    determine which rupture length magnitude relationship to use
      if(als.eq.0)   go to 1240
      al=als
      bl=bls
      sigl=sigls
c     compute break length
 1180 write(16,1190) al,bl,sigl
 1190 format(' fault rupture length parameters al='f7.3' bl='f7.3
     &'  sigl='f6.2)
      linset=1
      if (sigl.gt.0) go to 1230
c     which single break default do we want
c     if bl=0, al non zero use old Algermissen-Perkins default values
c     use parameter values supplied for one break only
 1200 do 1210 i=1,lev
 1210 rlls(1,i)=10.**(al+bl*fm(i))
 1220 prls(1)=1.
      nrls=1
      go to 1350
c    set up rupture lengths and probabilities as fct of magnitude
c    nrls lengths per magnitude
c    al, bl, sigl non zero
 1230 call pbreak(fm,al,bl,sigl,rlls,prls,lev,nrls)
      go to 1350
 1240 if (nrls.eq.1) go to 1200
      if(linset.eq.1) go to 1230
c     compute default values
      al=-1.085
      bl=.389
      sigl=.52
      linset=1
      go to 1180
c  convert quad corner points to decimal degrees  (if dsw =1)
c
 1250 do 1280 ii=1,num
      do 1280 i=1,2
      if(dsw.eq.0) go to 1260
      call condec(xsav(i,ii))
      call condec(ysav(i,ii))
 1260 continue
      xin=xsav(i,ii)*rad
      yin=ysav(i,ii)*rad
      call to eqr(xin,yin,xout,yout)
      xsav(i,ii)=xout*r
      ysav(i,ii)=yout*r
c     write(16,1270) xsav(i,ii),ysav(i,ii),ii
 1270 format(' xsav,ysav='2f9.3' ii='i3)
 1280 continue
c
 1290 nm=num-1
```

```
      stot=0.
      do 1330 ii=1,nm
      if(nbr.eq.0) go to 1310
      do 1300 iq=1,nbr
      if(ii.eq.ibr(iq)) go to 1330
 1300 continue
c     set line parameters for subsource
 1310 call setreg(ii)
      do 1320 i=1,2
      x(i)=xsav(i,ii)
      x(i+2)=xsav(i,ii+1)
      y(i)=ysav(i,ii)
 1320 y(i+2)=ysav(i,ii+1)
c     find area of subregion ii
      s1=abs(x(1)*(y(2)-y(3))+x(2)*(y(3)-y(1))+x(3)*(y(1)-y(2)))
      s2=abs(x(4)*(y(2)-y(3))+x(2)*(y(3)-y(4))+x(3)*(y(4)-y(2)))
      sarea(ii)= (s1+s2)/2.
      stot=stot+sarea(ii)
 1330 continue
      eua=noc(1)
c    find earthquake rate per unit area for lowest magnitude
c    rate will be printed but not used in program
      delm=(fm(2)-fm(1))/2.
      fm1=fm(1)-delm
      fm2=fm(1)+delm
      eua=eua/stot
      write(16,1340) dumid,stot,eua,fm1,fm2
 1340 format(1h a4' area='f9.0' sq km, rate/sq km='e12.5,
     & ' for mags 'f5.2'-'f5.2)
 1350 dlev=fm(1)-fm(2)
c     create table of log accelerations for set of magnitudes
c     in (tm) for the original set of mdis distances as a function
c     of distance
c     next three lines test whether table has already
c      been computed--if so skip over
      if(lev.gt.levno) go to 1360
      if(fm(1).ne.flevst) go to 1360
      if(dlev.eq.dlevsv) go to 1520
 1360 do 1370 ill=2,jent
 1370 tmdif(ill)=tm(ill)-tm(ill-1)
      do 1420 ll=1,lev
      do 1380 il=2,jent
      if(fm(ll).gt.tm(il)) go to 1390
 1380 continue
      il=jent
 1390 ilm=il-1
      do 1400 j=1,mdis
 1400 ata(j,ll)=atab(j,ilm)+(fm(ll)-tm(ilm))/tmdif(il)
     & *(atab(j,il)-atab(j,ilm))
      do 1410 j=2,mdis
 1410 atadif(j,ll)=ata(j,ll)-ata(j-1,ll)
 1420 lsub(ll)=il
c     find distance corresponding to each acceleration in
c     table ac (set up in subroutine box) for each magnitude
      do 1510 l=1,lev
      do 1500 j=1,maxp
      do 1430 i=2,mdis
      if(aclog(j).gt.ata(i,l)) go to 1450
 1430 continue
c     beyond table
 1440 dtab(j,l)=rtab(mdis)
      go to 1500
 1450 if(atadif(i,l).ne.0) go to 1490
c     no change in acceleration at this distance
 1460 if(i.eq.2) go to 1480
      write(16,1470) i,l,j
 1470 format(' apparent error indist-table i,l,j='3i4)
      call exit
 1480 dtab(j,l)=0.
      go to 1500
 1490 dtab(j,l)=rtab(i)-(rtab(i)-rtab(i-1))*(ata(i,l)-aclog(j))
     &/atadif(i,l)
      if(dtab(j,l).lt.0) dtab(j,l)=0.
 1500 continue
 1510 continue
      dlevsv=dlev
      levno=lev
      flevst=fm(1)
 1520 continue
c     add half values of mag for later spreading out
      dmag=fm(2)-fm(1)
      bmag=fm(1)-dmag/2.
      levp=lev+1
      do 1530 i=1,levp
      gm(i)=bmag
 1530 bmag=bmag+dmag
      do 1570 ll=1,levp
      do 1540 il=2,jent
      if(gm(ll).gt.tm(il)) go to 1550
 1540 continue
      il=jent
 1550 do 1560 j=1,mdis
      atah(j,ll)=atab(j,il-1)+(gm(ll)-tm(il-1))/tmdif(il)
     & *(atab(j,il)-atab(j,il-1))
 1560 continue
 1570 continue
c     take exp differences for later accel spread out
c     the "exdif" values are used in subroutine "spread"
      if (lev.eq.1) betad=2.0
      if(lev  .ne.1) betad=alog(noc(1)/noc(2))
      beta=betad/dmag
      fmul=1./(1.-exp(-beta*dmag))
      frak=exp(-beta*dmag/100.)
      frm=frak
      exdif(1)=0.
      do 1580 i=2,101
      exdif(i)=(1.-frm)*fmul
 1580 frm=frm*frak
      dtmax=dtab(1,1)
      if(dlev.lt.0) dtmax=dtab(1,lev)
c     find entry for each magnitude in distance-acceleration table
c     for zero distance (highest acceleration for that magnitude)
      do 1600 l=1,lev
      j=maxp
 1590 if(dtab(j,l).ne.0.) go to 1600
      j=j-1
      go to 1590
 1600 maxa(l)=j
c
      irow1s=irow1
```

```
      irow2s=irow2
      icol1s=icol1
      icol2s=icol2
      idrc=0
      do 2770 ispt=1,2
      if(ispt.eq.1) go to 1610
c     case of calculation done for single points on line
      if(indvpt.eq.0) go to 2770
      irow1s=1
      irow2s=1
      icol1s=1
      icol2s=indvpt
      go to 1620
 1610 if(irow1.eq.0) go to 2770
 1620 do 2760 ira=irow1s,irow2s
c
      ymid=(ph1-(ira-1.)*phinc)*r
      do 2750 ii=icol1s,icol2s
c
      idrc=idrc+1
      if(ispt.eq.2) go to 1630
      fii=ii
      xmid=(fl1-(fii-1.)*flinc)*r
      go to 1640
 1630 xmid=xpt2(ii)
      ymid=ypt2(ii)
 1640 if(isw.eq.0) go to 1660
c
c     after first time thru, read accelerations calculated
c     for previous sources at this site (saved in 2 dimensional
c     array rawbin) into reg (1 dimensional) array.
c
      do 1650 iqrp=1,maxdim
 1650 reg(iqrp)=rawbin(iqrp,idrc)
      go to 1680
c
c    first time thru clear reg array if not continuation run
c
 1660 do 1670 ll=1,maxp
 1670 reg(ll)=0.
      regg(idrc)=9999.999
c     set regg to min dist from fault
c
c     for fixed  affected area in row ira, col ii
c
c       source area computation
 1680 if(line.eq.1) go to 2120
      do 2110 k=1,nm
      if(nbr.eq.0) go to 1700
      do 1690 iq=1,nbr
      if(k.eq.ibr(iq)) go to 2110
 1690 continue
 1700 approx=0.
      do 1710 i=1,2
      xll(i)=xsav(i,k)
      xll(i+2)=xsav(i,k+1)
      yl(i)=ysav(i,k)
      yl(i+2)=ysav(i,k+1)
 1710 continue
      xr(3)=xll(1)
      yr(3)=yl(1)
      xr(1)=xll(2)
      yr(1)=yl(2)
      xr(2)=xll(4)
      yr(2)=yl(4)
      xr(4)=xll(3)
      yr(4)=yl(3)
      aread=sarea(k)/stot
c     determine whether site is inside or outside k th quadrilateral
c     subregion.  find distance rc to closest endpoint
c     and rf to farthest end point.  Set up table of distance versus
c     arc length for distances in the range rc to rf using
c     subroutines outsid or inside.
c      itst=no of radii at fixed increments between rc and rf
c           at which arc lengths are to be evaluated in table
c      itst=7   set in subroutine rrisk--recompile to change
      call rrisk(xmid,ymid,in,k,dis,rc,rf,itst)
      if(in.eq.1) go to 1740
      if(rc.lt.dtmax) go to 1730
      do 1720 ll=1,lev
 1720 reg(1)=reg(1)+aread*noc(ll)
      go to 2110
 1730 rcs=rc
      disl(1)=rc
      angl(1)=0.
      itst=itst+1
      ibeg=2
      go to 1750
 1740 disl(1)=0.
      angl(1)=0
      disl(2)=rc
      angl(2)=pi2*rc
      ibeg=3
      if(rc.eq.0) ibeg=2
      rcs=0.
      itst=itst+2
c    compute distance versus arc length table for even increments
c    in distance
 1750 do 1810 iii=ibeg,itst
      if(in.eq.0.) go to 1760
      call inside(xmid,ymid,k,dis,angle)
      go to 1780
 1760 call outsid(xmid,ymid,k,dis,angle,kerr)
 1770 format('err outside for ira='i4' ii='i4)
 1780 continue
      disl(iii)=dis
      angl(iii)=angle*dis
      if(kerr.eq.0) go to 1810
      write (16,1770) ira,ii
      do 1790 imkl=1,iii
 1790 write(16,1800) disl(imkl),angl(imkl)
 1800 format(' dist, ang='2e12.5)
 1810 dis=dis+stsize
      itst=itst+1
      disl(itst)=rf
      angl(itst)=0.
c    compute distance versus arc length table for distances
c    to selected vertices and perpendiculars to edges
      rcsq=rc*rc
      rfsq=rf*rf
```

```
      do 1940 i=1,4
      ire=0
      if(in.eq.0) ire=2
      if(sllq(i).le.rcsq+1.) go to 1930
      if(sllq(i).ge.rfsq-1.) go to 1930
      dis=sqrt(sllq(i))-.001
      if(dis.le.rc) goto 1930
c     add angle, distance corresponding to vertex
      if (in.eq.0) go to 1830
      call inside(xmid,ymid,k,dis,angle)
      go to 1860
 1820 if(mina(i).ne.3) go to 1940
c    closest side next
      dis=abs(perpa(i))
      if(dis.le.rc+.5) go to 1940
      if(dis.ge.rf-.5) go to 1940
      dis=dis-.001
      call inside(xmid,ymid,k,dis,angle)
      ire=1
      go to 1860
 1830 continue
      call outsid(xmid,ymid,k,dis,angle,kerr)
      go to1860
 1840 if(mina(i).ne.3) go to 1940
      dis=abs(perpa(i))
      ire=1
      do 1850 iql=1,4
      if(abs(sllq(iql)-perpaq(i)).le. 2.) go to 1940
 1850 continue
      if(dis.le.rc+1.) go to 1940
      if(dis.ge.rf-1.) go to 1940
      dis=dis-.001
      call outsid(xmid,ymid,k,dis,angle,kerr)
 1860 continue
      if(kerr.eq.0) go to 1880
      write (16,1770) ira,ii
      do 1870 imkl=1,iii
 1870 write(16,1800) disl(imkl),angl(imkl)
 1880 continue
      do 1890 ll=1,itst
      if(dis-disl(ll)) 1900,1920,1890
 1890 continue
c    insert new distance, angle
 1900 jk=itst
      do 1910 mm=ll,itst
      angl(jk+1)=angl(jk)
      disl(jk+1)=disl(jk)
 1910 jk=jk-1
      disl(ll)=dis
      angl(ll)=angle*dis
      itst=itst+1
 1920 if(ire.eq.0) go to 1820
      if(ire.eq.2) go to 1840
      go to1940
 1930 if(in)1840,1840,1820
 1940 continue
      itst=itst+1
      disl(itst)=rf
      angl(itst)=0.
c    for each magnitude, source, determine fractional source area giving
c    acceleration a:  ac(i-1) < a < ac(i)  for each entry in acceleration
c    table ac.  equivalent to finding fractional source area at distance
c    from site between successive entries in acceleration-distance
c    table:  dtab(i,l) < d < dtab(i-1,l) for magnitude l.
c    interpolates for distance is disl vs angl table
c    area=integral(angle*dist*delta(dist))  from dtab(i-1,l) to dtab(i,l)
      do 2090 ll=1,lev
      stnoc=noc(ll)/stot
      iswd=0
      maxm=maxa(ll)
      if(maxm.le.0.) go to 1960
      approx=0.
      if(in.eq.0) go to 1950
      jlo=maxm
      go to 1980
 1950 if(rcs.lt.dtab(i,ll)) go to 1970
 1960 reg(1)=reg(1)+aread*noc(ll)
      go to 2080
 1970 jlo=indpt(dtab(1,ll),rcs,maxm)
 1980 j=jlo
      iqs=2
      do 2060 jk=1,jlo
      if(jk.gt.1) go to 1990
      dbot=rcs
      ddis=dtab(j,ll)
      if(dtab(j,ll).lt.rf) goto 2010
      reg(j+1)=reg(j+1)+sarea(k)*stnoc
      go to 2080
 1990 if(dtab(j,ll).lt.rf) go to 2000
      dbot=dtab(j+1,ll)
      ddis=rf
      iswd=1
      go to 2010
 2000 ddis=dtab(j,ll)
      dbot=dtab(j+1,ll)
 2010 anarea=0.
      do 2020 iq=iqs,itst
      if(dbot.le.disl(iq)) go to 2030
 2020 continue
      iq=itst
c    distance exceeds last value-error since table set up to handle this
 2030 dtop=disl(iq)
      if(disl(iq).gt.ddis) dtop=ddis
      dmid=(dtop+dbot)/2.
      delr=dtop-dbot
      fr=(dmid-disl(iq-1))/(disl(iq)-disl(iq-1))
      ang=angl(iq-1)+(angl(iq)-angl(iq-1))*fr
      anarea=anarea+ang*delr
      if(dtop.eq.ddis) go to 2050
      dbot=dtop
      iq=iq+1
      if(iq.le.itst) go to 2030
      write(16,2040)
 2040 format(' err in arc area computation')
      go to 2030
c    interpolate in angle , distance table
 2050 approx=approx+anarea
      reg(j+1)=reg(j+1)+anarea*stnoc
      j=j-1
      if(iswd.eq.0) go to 2060
```

```
      arerr=(approx-sarea(k))/sarea(k)
      allerr=allerr+1.
      if (abs(arerr).lt. .05) go to 2090
      tot5=tot5+1.
      if(abs(arerr).le. .10) go to 2090
      write(16,2100) arerr,ira,ii,rc,rf,k,ll,in,dumid
      go to 2090
 2060 iqs=iq
c     need to add remaining area if any to lowest acceleration
      dif=sarea(k)-approx
      if(dif.lt.0) go to 2080
 2070 reg(1)=reg(1)+dif*stnoc
 2080 approx=sarea(k)
 2090 continue
 2100 format(' int err='f6.3' row 'i2' col 'i2' rc='
     & f7.2' rf='f7.2' k='i2' ll='i2' in='i1,1x,a4)
 2110 continue
      go to 2730
c     fault computation
 2120 do 2720 j=1,jtot
      jseg=jm(j)
      jsegm=jseg-1
      call cldis(xmid,ymid,j)
      do 2160 ll=1,lev
c     fill in rupture lengths--make  certain that rupture length
c     does not exceed total fault length
      nrl=nrls
      do 2130 irl=1,nrl
      rll(irl,ll)=rlls(irl,ll)
 2130 prlrl(irl,ll)=prls(irl)
      if(rll(1,ll).lt.dtot(j)) go to 2160
      rll(1,ll)=dtot(j)
      if(nrl.eq.1) go to 2160
 2140 if(rll(2,ll).lt.dtot(j)) go to 2160
      prlrl(1,ll)=prlrl(1,ll)+prlrl(2,ll)
      nrl=nrl-1
      if(nrl.eq.1) go to 2160
      do 2150 i=2,nrl
      prlrl(i,ll)=prlrl(i+1,ll)
 2150 rll(i,ll)=rll(i+1,ll)
      go to 2140
 2160 nrlsv(ll)=nrl
      pt1=sqrt(shdis)
      if(pt1.lt.regg(idrc)) regg(idrc)=pt1
      if (isid.eq.0) go to 2490
c    distance is monotonically increasing for all segments
c    (the decreasing case has been converted to increasing)
c    or single shortest distance is interior to fault
      do 2480 ll=1,lev
      nrl=nrlsv(ll)
      frcon=dtot(j)/dsum*noc(ll)
      maxm=maxa(ll)
      jlo=indpt(dtab(1,ll),pt1,maxm)
      if (jlo.gt.0) go to 2170
c   distance beyond table--lump acceleration in first box
      reg(1)=reg(1)+frcon
      go to 2480
 2170 continue
      if (isid.eq.1) go to 2190
 2180 ibeg=1
      ifin=jsegm
      ixt=1
      dnear=dtot(j)
      inear=jsegm
      go to 2310
 2190 ibeg=1
      ifin=inear
      ixt=2
c   dnear=distance along fault from one end to point on fault
c   closest to the site
c   dnear=total length of segments 1,2,...inear
c   inear defined sub cldis--in common/extra/
      dnear=0.
      do 2200 iq=1,inear
 2200 dnear=dnear+distl(iq)
      dfar=dtot(j)-dnear
      totac=0.
      do 2220 irl=1,nrl
      smd=rll(irl,ll)
      if(smd.lt.dtot(j)) go to 2210
c     break is entire segment
      totac=totac+frcon*prlrl(irl,ll)
      sumr=totac
      go to 2220
 2210 smd2=smd/2.
c    find fraction of fault at closest distance to site
c    for interior site
      arcn=smd2
      arcf=smd2
      if(dnear.lt.smd) arcn=dnear-smd2
      if(dfar.lt.smd) arcf=dfar-smd2
      arct=arcn+arcf
      if(arct.lt. .0001) go to 2220
      totac=totac+frcon/(dtot(j)-smd)*arct*prlrl(irl,ll)
 2220 continue
      sumr=totac
      totacs=totacs+totac
      jlop=jlo+1
      if(totl.gt.0) go to 2230
c     spread dummy faults over "totl" km distance if "totl" > 0
c      if "totl" = 0, spread over magnitude instead
      call spread(reg,jlop,ll,totac,pt1)
      go to 2310
 2230 width=totl/2.
      sumr=sumr/totl
      add=0
      if(abs(pt1).gt.width) go to 2280
      qtot=width+abs(pt1)
      dsvs=0.
      jlo2=maxm
      do 2270 iqk=1,2
      jrp=jlo2
      do 2250 jk=1,jlo2
      dl3 =dtab(jrp,ll)
      if(dl3.gt.qtot) go to 2260
      cary=dl3-dsvs
      reg(jrp+1)=reg(jrp+1)+sumr*cary
      add=add+sumr*cary
      jrp=jrp-1
 2240 format(' add='e12.5)
```

```
 2250 dsvs=d13
      jrp=0
 2260 cary=qtot-dsvs
      add=add+sumr*cary
      jrpp=jrp+1
      reg(jrp+1)=reg(jrp+1)+sumr*cary
      dsvs=0.
      qtot=width-abs(pt1)
 2270 jrp=jlo2
      jrpp=jrp+1
      go to 2310
 2280  dstart=abs(pt1)-width
      qtot=totl+dstart
      jlo2=indpt(dtab(1,ll),dstart,maxm)
      dsvs=dstart
      jrp=jlo2
      do 2290 jk=1,jlo2
      d13=dtab(jrp,ll)
      if(d13.gt.qtot) go to 2300
      cary=d13-dsvs
      reg(jrp+1)=reg(jrp+1)+sumr*cary
      add=add+sumr*cary
      jrp=jrp-1
 2290 dsvs=d13
      jrp=0
 2300 cary=qtot-dsvs
      reg(jrp+1)=reg(jrp+1)+cary*sumr
c    find frac of fault in distance range dtab(jr,ll) < d <dtab(jr-1,ll)
c    for each magnitude ll
 2310 do 2470 ithru=1,ixt
      jr=jlo
      ilo=1
      iii=ibeg
      casum=0.
      irl=1
      dsav=-dls(iii)
 2320 frac=0.
      jk=1
      smd=rll(irl,ll)
      if(smd.lt.dtot(j)) go to 2330
      if(ixt.eq.2)go to 2450
c     add in entire fault length
      totac=frcon*prlrl(irl,ll)
      jlop=jlo+1
      call spread(reg,jlop,ll,totac,pt1)
      go to 2450
 2330 cuml=dnear-smd
      if(cuml.le.0) go to 2450
      do 2340 iq=irl,nrl
 2340 frac=frac+prlrl(iq,ll)/(dtot(j)-rll(iq,ll))
      frac=frac*frcon
      dlen=0.
      do 2350 iq=ibeg,ifin
      disrem(iq)=distl(iq)
      dlen=dlen+disrem(iq)
      if (dlen.ge.cuml) go to 2370
 2350 continue
      write(16,2360)dlen,ibeg,ifin,cuml
 2360 format(' error for fault line dlen='e12.5' ibeg='i3
     &' ifin='i3' cuml='e12.5)
      call exit
 2370 disrem(iq)=cuml-dlen+disrem(iq)
 2380 ifins=iq
 2390 dend=disrem(iii)-dls(iii)
c     dsav=starting distance along segment iii
 2400 continue
      do 2410 jk=ilo,jlo
      d13=sqrt(dtab(jr,ll)**2-persq(iii))
      if(d13.gt.dend) go to 2440
      carea=d13-dsav
      casum=casum+carea
      reg(jr+1)=reg(jr+1)+carea*frac
      jr=jr-1
 2410 dsav=d13
c    at top of table--remaining dist beyond table
 2420 do 2430 iq=irl,nrl
      cuml=dnear-rll(iq,ll)
 2430 reg(1)=reg(1)+(cuml-casum)*prlrl(iq,ll)*frcon
     & /(dtot(j)-rll(iq,ll))
      go to 2460
 2440 carea=dend-dsav
      casum=casum+carea
      reg(jr+1)=reg(jr+1)+carea*frac
      dsav=dend
c      at end of segment iii
      if(iii.eq.ifins)  go to 2450
      iii=iii+1
      dsav=-dls(iii)
      ilo=jk
      go to 2390
 2450 irl=irl+1
      ilo=jk
      if(irl.le.nrl) go to 2320
      if(ithru.eq.ixt) go to2480
 2460 irl=1
      ibeg=inear+1
      ifin=jsegm
      dnear=dtot(j)-dnear
 2470 continue
 2480 continue
      go to 2720
c     more than one turning point
c     brute force.  find contribution for single break
c     increment break centers by distance deltal along fault from
c     one end to other.
 2490 continue
      deltal=10.
      if(pt1.gt.100) deltal=20.
      if(pt1.lt.15.) deltal=5.
      do 2710 ll=1,lev
      nrl=nrlsv(ll)
      do 2710 irl=1,nrl
      smd=rll(irl,ll)
      prnoc=noc(ll)*prlrl(irl,ll)
      dj=dtot(j)-smd
      nd=dj/deltal+1.5
      if(nd.gt.1) go to 2500
      nd=2
      if(dj.gt.1.) go to 2500
c     break is entire fault length
```

```
      maxm=maxa(ll)
      jlo=indpt(dtab(1,ll),pt1,maxm)
      add=prnoc*dtot(j)/dsum
      jlop=jlo+1
      call spread(reg,jlop,ll,add,pt1)
      go to 2710
 2500 continue
      dmvp=prnoc*dtot(j)/(dsum*(nd-1))
      del=dj/(nd-1)
      do 2700 m=1,nd
      dst=(m-1)*del
      do 2510 i=1,jsegm
      if(dst.lt.dist(i,j)) go to 2530
      dst=dst-dist(i,j)
 2510 continue
      write(16,2520)dst,j
 2520 format(' apparent error fault dist--dst='e12.5'  j='i4)
      go to 2990
 2530 ddsav=1.e8
      dfin=dst+smd
 2540 dover=0.
      if(dfin.le.dist(i,j)) go to 2550
      dover=dfin-dist(i,j)
      dfin=dist(i,j)
 2550 if(dls(i).lt.dst) go to 2560
      if(dls(i).gt.dfin) go to 2570
c      shortest distance to segment at point within segment
      dd=persq(i)
      go to 2580
c      shortest dist at start of segment
 2560 dd=(dls(i)-dst)**2+persq(i)
      go to 2580
c      shortest distance at end of segment
 2570 dd=(dls(i)-dfin)**2+persq(i)
 2580 continue
      if(dd.lt.ddsav) ddsav=dd
      if(abs(dover).lt.0.001) go to 2600
      dst=0
      i=i+1
      dfin=dover
      if(i.le.jsegm) go to 2540
 2590 format(' i too large i='i2)
      write(16,2590) i
      call exit
 2600 dis=sqrt(ddsav)
c  have distance to subsegment of arc of fault
c
      if(m.gt.1) go to 2610
      jen2=indpt(dtab(1,ll),dis,maxa(ll))
      disav=dis
      go to 2700
 2610 disdif=dis-disav
      if(disdif.eq.0) go to 2620
      dfac=dmvp/disdif
c     spread out contributions into accel boxes
      if(disdif) 2630,2620,2660
 2620 jlop=jen2+1
      call spread(reg,jlop,ll,dmvp,dis)
      go to 2700
c      distance decreasing
 2630 jen2=jen2+1
 2640 distab=dis-dtab(jen2,ll)
      if (distab.lt.0) go to 2650
      reg(jen2)=reg(jen2)+(dis-disav)*dfac
      add=(dis-disav)*dfac
      if(add.lt.0) print *,2740
      jen2=jen2-1
      disav=dis
      go to 2700
 2650 reg(jen2)=reg(jen2)+(dtab(jen2,ll)-disav)*dfac
      add=(dtab(jen2,ll)-disav)*dfac
      if(add.lt.0) print*,2750
      disav=dtab(jen2,ll)
      jen2=jen2+1
      go to 2640
 2660 if(jen2.eq.0) go to 2680
 2670 distab=dis-dtab(jen2,ll)
      if(distab.gt.0) goto 2690
 2680 reg(jen2+1)=reg(jen2+1)+(dis-disav)*dfac
      add=(dis-disav)*dfac
      if(add.lt.0) print *, 2780
      disav=dis
      go to 2700
 2690 reg(jen2+1)=reg(jen2+1)+(dtab(jen2,ll)-disav)*dfac
      add=(dtab(jen2,ll)-disav)*dfac
      if(add.lt.0) print*, 2790
      disav=dtab(jen2,ll)
      jen2=jen2-1
      if (jen2.ge.1) go to 2670
      go to 2680
 2700 continue
 2710 continue
c     above ends loop for more than one turning point
 2720 continue
 2730 continue
      do 2740 iqr=1,maxdim
 2740 rawbin(iqr,idrc)=reg(iqr)
c      save accelerations for this source & site in rawbin array
c
c     ends loop for one site, one area or fault
c     repeat for all sites for this source before going on to next
c     source
 2750 continue
 2760 continue
c     one source completed for all sites
 2770 continue
c----smooth here if sigma for earthquake locations has changed
c----from previous value
      if(line.eq.1) go to 2850
c-----no smoothing for faults
      if(numsav.ne.98) go to 2850
c    if sigma=0, no smoothing
      if(als.eq.0) go to 2850
       if(nrc.eq.0) go to 2850
c-----
      ismsw=1
       sigd=(dphinc/als)**2
      do 2780 im=1,imaxsq
 2780 fac(im)=exp((-im*sigd)/2.)
      kmax=10
```

```
      eqsum=0.
      dumfac=1.
c     compute icon based on sigma and dphinc
c     probably enough to go out 2 sigma on either side of point
c     find how many rows or columns that corresponds to
      icon=2*als/dphinc+1.01
      if(icon.lt.5) icon=5
c    if icon<5, set icon=5
      irm=irow2-nbord
      icm=icol2-nbord
      isub1=irow1+nbord
      isub2=icol1+nbord
      do 2830 iia=isub2,icm
      j1=iia-icon
      j2=iia+icon
      if(j2.gt.icol2) j2=icol2
      if(j1.lt.icol1) j1=icol1
      do 2830 iraa=isub1,irm
      i1=iraa-icon
      i2=iraa+icon
      if(i1.lt.irow1) i1=irow1
      i2=iraa+icon
      if(i2.gt.irow2) i2=irow2
      do 2800 iq=1,maxp
      add=0.
      wtt=0.
      do 2790 i=i1,i2
      do 2790 j=j1,j2
      ksq=(iraa-i)**2+(iia-j)**2
      if(ksq.gt.imaxsq) go to 2790
c     we need to convert to proper subscript in rawbin
c      in rawbin, first all cols for one row
c      (i-1)*icol(s)+j=row and col with single subscript in rawbin
      isub=(i-irow1)*icols+j-icol1+1
      add=add+fac(ksq)*rawbin(iq,isub)
      wtt=wtt+fac(ksq)
 2790 continue
      reg(iq)=0
      if(wtt.ne.0) reg(iq)=add/wtt
      eqsum=eqsum+reg(iq)
 2800 continue
 2810 continue
      isub=(iraa-irow1)*icols+iia-icol1+1
      do 2820 iq=1,maxdim
 2820 smobin(iq,isub)=reg(iq)+smobin(iq,isub)
 2830 continue
c    now delete entries from rawbin array
      do 2840 isub=1,nrc
      do 2840 iq=1,maxdim
 2840 rawbin(iq,isub)=0.
 2850 continue
      if(iprint.lt.0) go to 2940
c  output all results to this point
      do 2930 ispt=1,2
      if (ispt.eq.1) go to 2860
      if(indvpt.eq.0) go to 2930
      irow1s=1
      irow2s=1
      icol1s=1
      icol2s=indvpt
      go to 2870
 2860 if(irow1.eq.0) go to 2930
      irow1s=irow1+nbord
      irow2s=irow2-nbord
      icol1s=icol1+nbord
      icol2s=icol2-nbord
 2870 do 2920 ira=irow1s,irow2s
      do 2920 ii=icol1s,icol2s
      if(ispt.eq.1) go to 2890
 2880 xouts=xpt1(ii)
      youts=ypt1(ii)
      isub=irows*icols+ii
      go to 2900
c     back transform point from equator to dec degrees
 2890 call backw(ii,ira,xouts,youts)
c     results calculated and printed in subroutine out
      isub=(ira-irow1)*icols+ii-icol1+1
 2900 do 2910 iqr=1,maxp
 2910 reg(iqr)=rawbin(iqr,isub)+smobin(iqr,isub)
      rdis=regg(isub)
      if(iprint.le.2) write(16,30)tytle
      call out(reg,naf,xouts,youts,sd)
 2920 continue
 2930 continue
 2940 continue
      isw=1
      go to 680
c     finished
 2950 nvals=icols*irows+indvpt
c     write to unit 3 for possible run continuation
c     may delete to save time and space if no run
c     continuations are desired.
      do 2970 i=1,nvals
      do 2960 iqr=1,maxdim
 2960 reg(iqr)=rawbin(iqr,i)+smobin(iqr,i)
      write(3) reg
 2970 continue
c    transfer to unit 3 for possible continuation or restart
 2980  continue
 2990 continue
c     write(16,2530) tot5,allerr
 3000 format(/' sources with error in area > .05='f7.0
     & ' out of 'f7.0' sources ')
 3010 format(' number of brute force points='i4)
      stop
      end
      subroutine cond(x1,x2,y1,y2)
      call condec(x1)
      call condec(x2)
      call condec(y1)
      call condec(y2)
      return
      end
      subroutine condec(phi)
      i=phi
c     fr=phi-i
      phi=float(i)+(phi-float(i))/.60
      return
      end
c ************************************************************
```

```
      subroutine lin(x1,y1,x2,y2,a,b,c,x3,y3)
c   coefficients a, b, c are determined so that points which are in the
c   same half plane as x3,y3 will be a positive distance from the line
c   joining x1,y1 and x2,y2
c       a x + b y + c=0
      if(x1.eq.x2) go to 10
      a=(y1-y2)/(x1-x2)
      b=-1
      c=y1-a*x1
      go to 20
   10 a=1.
      b=0
      c=-x1
   20 d=a*x3+b*y3+c
      if(d.gt.0.) return
      a=-a
      b=-b
      c=-c
      return
      end
c ************************************************************
      subroutine linint(reg,t,sol)
c     calculates acceleration "sol" that has probability
c     "prob" of not being exceeded during "t" years at
c     site.
c     expected yearly accelerations in the range
c     ac(j-1) < a < ac(j) are in reg(j).
c     linint is called from subroutine out.
      common/tims/dummy(42),iprint
      common/quad/maxp,ac(105),dumlog,aclog(105)
      common/ext/prob,ex,ex1
      dimension reg(106)
      ii=maxp-1
      sum=0.
   10 if(ii.le.2) go to 20
      if(reg(ii).gt.0.) go to 40
      ii=ii-1
      go to 10
   20 sol=0.
      if(iprint.le.2) write(16,30)
   30 format(' sol not obtained for time='f7.0)
      return
   40 sum=sum+reg(ii)
      if(t*sum.gt.ex) go to 50
      ii=ii-1
      if(ii.ge.2) go to 40
      go to 20
   50 if(sum.eq.reg(ii)) go to 70
      y2=alog(t*(sum-reg(ii)))
      y1=alog(t*sum)
      ex1t=ex1
   60 a=(y1-y2)/(ac(ii-1)-ac(ii))
      b=y1-a*ac(ii-1)
      sol=(ex1t-b)/a
      return
   70 y1=t*sum
      y2=0.
      ex1t=ex
      if(iprint.le.2) write(16,80)
   80 format(' log interpolation fails--use linear')
      go to 60
      end
c ************************************************************
      subroutine box(scale,sd)
c     determine boundaries ac(i) for acceleration levels
c     accelerations in the range ac(i-1) < a <ac(i) will be
c     accumulated in reg(i)
c     scale multiplies the basic levels computed
c     scale=1 for accelerations in the range .02 to 1.
      common /quad/maxq,ac(105),dumlog,aclog(105)
      common/wds/max,maxp,maxsp
      maxsp=104
      max=54
      maxp=55
      ac(5)=.02*scale
      dif=ac(5)
      do 10 i=6,55
      ac(i)=ac(i-1)+dif
   10 continue
c ----------------note important-------------
c     the following inserted to allow acceleration boxes for small
c     values below range of interest for sd computation
c     this should be changed later as desired
      j=4
      efac=exp(-sd/2.)
      do 20 i=1,4
      ac(j)=ac(j+1)*efac
   20 j=j-1
c    for later attenuation variability, expand acceleration range
c    used only in subroutine "out"
      y=alog(ac(max))
      w=exp(y+4.4*sd)
      w1=exp(y+2.0*sd)
c     dimension of ac, aclog is 105; we cannot exceed this dimension
      n=maxsp-max
      n2=n/2
      i2s=max+n2
      dif=(w1-ac(max))/n2
      do 30 i=maxp,i2s
      ac(i)=ac(i-1)+dif
   30 continue
      i2sp=i2s+1
      dif=(w-ac(i2s))/n2
      do 40 i=i2sp,maxsp
      ac(i)=ac(i-1)+dif
   40 continue
      return
      end
c ************************************************************
      subroutine inieqr(x1,y1,x2,y2)
c     read xlong, ylat in radians for 1st point
c     read xlong, ylat in radians for 2nd point
c     points 1 and 2 to be transformed to equator
c     point 1 will be transformed to (0,0)
c     point 2 will be transformed to (d,0)
c    where d=great circle distance between (x1,y1) and (x2,y2)
c     output is transformation matrices f (forward) fi (backward)
c       generated and stored in common
      common/eqr/ r1(3,3),r2(3,3),r3(3,3),d(3,3),f(3,3),vec(3)
      common/inv/fi(3,3),res(3)
```

```
      cp1=cos(y1)
      cp2=cos(y2)
      cl1=cos(x1)
      sp1=sin(y1)
      sp2=sin(y2)
      sl1=sin(x1)
c  initialize matrices
      do 10 i=1,3
      do 10 j=1,3
      r1(i,j)=0.
      r2(i,j)=0.
   10 r3(i,j)=0.
      r1(1,1)=1.
      r2(2,2)=1.
      r3(3,3)=1.
      r2(1,1)=cp1
      r2(3,3)=cp1
      r2(1,3)=sp1
      r2(3,1)=-sp1
      r3(1,1)=cl1
      r3(2,2)=cl1
      r3(1,2)=sl1
      r3(2,1)=-sl1
      s1=cp1*sp2-sp1*cp2*cos(x2-x1)
      c1=cp2*sin(x2-x1)
      den=sqrt(c1*c1+s1*s1)
      sinth=s1/den
      costh=c1/den
      r1(2,2)=costh
      r1(3,3)=costh
      r1(2,3)=sinth
      r1(3,2)=-sinth
      call matmpy(r1,r2,d)
      call matmpy(d,r3,f)
c   transformation matrix in f! inverse matrix in fi
c   compute inverse
      do 20 i=1,3
      do 20 j=1,3
      r1(i,j)=0.
      r2(i,j)=0.
      r3(i,j)=0.
   20 continue
      r1(3,3)=1.
      r2(2,2)=1.
      r3(1,1)=1.
      r1(1,1)=cl1
      r1(2,2)=cl1
      r1(2,1)=sl1
      r1(1,2)=-sl1
      r2(1,1)=cp1
      r2(3,3)=cp1
      r2(1,3)=-sp1
      r2(3,1)=sp1
      r3(2,2)=costh
      r3(3,3)=costh
      r3(3,2)=sinth
      r3(2,3)=-sinth
      call matmpy(r1,r2,d)
      call matmpy(d,r3,fi)
      return
      end
c ************************************************************
      subroutine toeqr(x3,y3,flam6,phi6)
c     transform point at (x3,y3)  (long,lat in radians) to new
c     equator using transformation defined in subroutine ineqr
      common/eqr/  r1(3,3),r2(3,3),r3(3,3),d(3,3),f(3,3),vec(3)
      common/inv/fi(3,3),res(3)
      cxmid=cos(x3)
      cymid=cos(y3)
      sxmid=sin(x3)
      symid=sin(y3)
      vec(1)=cxmid*cymid
      vec(2)=sxmid*cymid
      vec(3)=symid
      call vecmpy(f,vec,res)
      phi6=asin(res(3))
      cp6=cos(phi6)
      cl6=res(1)/cp6
      sl6=res(2)/cp6
      if(cl6.gt.1.) cl6=1.
      flam6=acos(cl6)
      if(sl6.lt.0.) flam6=-flam6
      return
      end
c ************************************************************
      subroutine backw(ix,iy,flii,phii)
c     for printout and identification: point defined by
c     row ix and col iy in new coordinate system is transformed
c     to location (long, lat. in dec degrees) in original
c     coordinate system.
      common/inv/fi(3,3),res(3)
      dimension vec(3)
c     need cos, sins in common array
      common/sincos/sinx(500),cosx(500),siny(500),cosy(500)
      data rad/.0174533/
      sl6=sinx(ix)
      cl6=cosx(ix)
      cp6=cosy(iy)
      sp6=siny(iy)
      vec(1)=cl6*cp6
      vec(2)=sl6*cp6
      vec(3)=sp6
      call vecmpy(fi,vec,res)
      if(abs(res(3)).lt.1.0) go to 30
      if(res(3).ge.1.) go to 10
      phii=-90.
      go to 20
   10 phii=90.
   20 flam=0.
      return
   30 phii=asin(res(3))
      cpi=cos(phii)
      phii=phii/rad
      cli=res(1)/cpi
      sli=res(2)/cpi
      flii=acos(cli)
      if(sli.lt.0.) flii=-flii
      flii=flii/rad
      return
      end
```

```
c ***************************************************************
      subroutine matmpy(a,b,c)
c     multiply two 3 x 3 matrices A x B = C
      dimension a(3,3),b(3,3),c(3,3)
      do 10 i=1,3
      do 10 j=1,3
      c(i,j)=0.
      do 10 k=1,3
   10 c(i,j)=c(i,j)+a(i,k)*b(k,j)
      return
      end
c ***************************************************************
      subroutine vecmpy(f,vec,r)
c     vector multiplication used by subroutines backw, toeqr
      dimension f(3,3),vec(3),r(3)
      do 10 i=1,3
      r(i)=0.
      do 10 k=1,3
   10 r(i)=r(i)+f(i,k)*vec(k)
      return
      end
c ***************************************************************
      subroutine csqdis(x,y,xl,yl,xr,yr,isub)
c     for a quadrilateral source area, calculates
c     nearest distance from site at (x,y) to each side
c     and distance to each vertex.
c     called from subroutine rrisk
      dimension xl(4),yl(4),xr(4),yr(4)
      common/slinp/fsa(4,50),fsb(4,50),fsc(4,50),rta(4,50),dist(4,50)
     & ,disq(4,50)
      common /slinds/dls(4),perp(4),sl1(4),slsq(4),min(4),persq(4)
     &,srsq(4),dlp(4)
      do 10 i=1,4
   10 perp(i)=(fsa(i,isub)*x+fsb(i,isub)*y+fsc(i,isub))/rta(i,isub)
      do 20 i=1,4
      slsq(i)=(x-xl(i))**2+(y-yl(i))**2
      srsq(i)=(x-xr(i))**2+(y-yr(i))**2
   20 continue
      do 50 i=1,4
      persq(i)=perp(i)*perp(i)
      dl1=slsq(i)-persq(i)
      dl2=srsq(i)-persq(i)
      if(dl1.lt.0.) dl1=0.
      if(dl2.lt.0.) dl2=0.
      if(dl1+dl2.le.disq(i,isub)) go to 40
c      shortest distance external to segment
      if(dl1.lt.dl2) go to 30
      min(i)=1
      dlp(i)=sqrt(dl2)
      dls(i)=dist(i,isub)+dlp(i)
      go to 50
   30 min(i)=2
      dls(i)=-sqrt(dl1)
      go to 50
   40 min(i)=3
      dls(i)=sqrt(dl1)
      dlp(i)=0.
   50 continue
      return
      end
```

```
c ************************************************************
      subroutine rrisk(xnot,ynot,in,isub,r,rc,rf,itst)
       common/inout/ic,ics,ifar,azimf,stsize
       common/xyarea/xsav(2,50),ysav(2,50),sarea(50)
       common/slinds/dls(4),perp(4),sli(4),slsq(4),min(4),persq(4)
     &,srsq(4),dlp(4)
       common/slinp/aa(4,50),bb(4,50),cc(4,50),rta(4,50),dist(4,50),
     & disq(4,50)
       common/xlyl/xl(4),yl(4),xr(4),yr(4),xc(4),yc(4)
c          subroutine which loads temporary arrays with this
c            subsource's corners and determines if this site is
c            inside or outside the subrource.
      data pi2/6.2831853/
c    fixed number of radii at which distance-vs arc length
c    table is computed is set here   itst=7 ***************
      itst=7
      rc2=10000000.
      ics=0
      call csqdis(xnot,ynot,xl,yl,xr,yr,isub)
      rf2=0.
      do 20 ii=1,4
      if(slsq(ii).gt.rc2) go to 10
      rc2=slsq(ii)
   10 if(slsq(ii).lt.rf2) go to 20
      rf2=slsq(ii)
      ifar=ii
   20 continue
      do 30 ii=1,4
      if(min(ii).ne.3) go to  30
      if(persq(ii).gt.rc2) go to 30
      rc2=persq(ii)
      ics=ii
   30 continue
      rc=sqrt(rc2)
      rf=sqrt(rf2)
      stsize=(rf-rc)/itst
c        step thru source area.
      r=rc+(0.5)*stsize
c        determine if site is inside source area.
   40 do 50 ii=1,4
      d=aa(ii,isub)*xnot+bb(ii,isub)*ynot+cc(ii,isub)
      if(d.lt.0)go to 90
   50 continue
      in=1

c           determine azimuth of farthest point with respect to site
      azimf=acos((xl(ifar)-xnot)/rf)
      if (yl(ifar)-ynot) 60,70,70
   60 azimf=pi2 - azimf
   70 continue
c       rc is now distance from site to closest side.
c       rf is now distance from site to farthest corner.
c         pick step size based on fraction of area left
   80 return
c          if do loop finished, point lies within area.
   90 in=0
  100 return
      end
c ************************************************************
      subroutine setreg(isub)
c    find equation of line thru each side of quadrilateral
c    find length of each side, etc.
      common/outs/xouts,youts
      dimension xl(4),yl(4),xr(4),yr(4)
      common/xyarea/xsav(2,50),ysav(2,50),sarea(50)
      common/slinp/aa(4,50),bb(4,50),cc(4,50),rta(4,50),dist(4,50),
     & disq(4,50)
      data r/6378./
      do 10 i=1,2
      xl(i)=xsav(i,isub)
      xl(i+2)=xsav(i,isub+1)
      yl(i)=ysav(i,isub)
      yl(i+2)=ysav(i,isub+1)
   10 continue
      xr(3)=xl(1)
      yr(3)=yl(1)
      xr(1)=xl(2)
      yr(1)=yl(2)
      xr(2)=xl(4)
      yr(2)=yl(4)
      xr(4)=xl(3)
      yr(4)=yl(3)
c         determine if any sides are vertical lines
      do 70 ii=1,4
      if(xl(ii).eq.xr(ii)) go tc 20
      aa(ii,isub)=(yl(ii)-yr(ii))/(xl(ii)-xr(ii))
      bb(ii,isub)=-1.
      cc(ii,isub)=yl(ii)-aa(ii,isub)*xl(ii)
      go to 30
   20 aa(ii,isub)=1.
      bb(ii,isub)=0.
      cc(ii,isub)=-xl(ii)
   30 if(ii.gt.2) go to 40
      d=aa(ii,isub)*xl(3)+bb(ii,isub)*yl(3)+cc(ii,isub)
      go to 50
   40 d=aa(ii,isub)*xl(2)+bb(ii,isub)*yl(2)+cc(ii,isub)
   50 if (d.gt.0.) go to 60
      aa(ii,isub)=-aa(ii,isub)
      bb(ii,isub)=-bb(ii,isub)
      cc(ii,isub)=-cc(ii,isub)
   60 disq(ii,isub)=(xr(ii)-xl(ii))**2+(yr(ii)-yl(ii))**2
      dist(ii,isub)=sqrt(disq(ii,isub))
   70 rta(ii,isub)=sqrt(aa(ii,isub)*aa(ii,isub)+bb(ii,isub)*
     &bb(ii,isub))
      return
      end
      function indpt(y,yval,n)
c      points in descending order--non zero values from points
c      1 to n:  find first value in table exceeding yval
      dimension y(102)
      if(yval.lt.y(1)) go to 10
      indpt=0
      return
   10 if(yval.gt.y(n)) go to 20
      indpt=n
      return
   20 num=n
      idiv=2
   30 nh=num/idiv
      if(nh.le.1) go to 50
      if(yval.lt.y(nh)) go to 40
```

```
      idiv=2*idiv
      go to 30
c     yval lies approx between nh and 2*nh or 2*nh
   40 idiv=2*idiv
      nh2=nh+num/idiv
      if(nh.eq.nh2) goto 80
      if(yval.gt.y(nh2)) go to 40
c     yval further down on table than y(nh2)
      nh=nh2
      go to 40
   50 do 60 i=1,n
      if(y(i).lt.yval) go to 70
   60 continue
   70 indpt=i-1
      return
   80 continue
      do 90 i=nh,n
      if(y(i).lt.yval) go to70
   90 continue
      end
c ************************************************************
      subroutine cldis(x,y,jl)
c     subroutine to compute shortest distance, etc to line
c     containing fault segment i of jl th fault (for each i)
c     determines whether closest point to fault is at
c     low end (min(i)=2)
c     interior  (min(i)=3)
c     high end  (min(i)=1)
c     set isid=0 if multiple turning points
c         isid=1 if single closest distance from site to fault
c         interior to fault
c         isid=2 if closest distance from site to fault at one end
c         of the fault
      dimension dlp(20)
      common/linps/distl(24,26),disq(24,26),jseg,jsegm,
     & xl(24,26),yp(24,26),coa(24,26),sia(24,26)
      common/extra/isid,shdis,dist(24),inear
      common /lindis/dls(24),perp(24),min(24),persq(24)
      shdis=1.e9
      do 10 ii=1,jsegm
   10 dist(ii)=distl(ii,jl)
      i1=0
      i2=0
      i3=0
      isid=0
      do 20 i=1,jseg
      slsq=(x-xl(i,jl))**2+(y-yp(i,jl))**2
      if(slsq.lt.shdis) shdis=slsq
   20 continue
      do 50 i=1,jsegm
      xd=x-xl(i,jl)
      yd=y-yp(i,jl)
      perp(i)=-xd*sia(i,jl)+yd*coa(i,jl)
      dls(i)=xd*coa(i,jl)+yd*sia(i,jl)
      persq(i)=perp(i)*perp(i)
      if(dls(i).lt.0) go to 30
      if(dls(i).le.dist(i)) go to 40
       shortest distance external to segment
      min(i)=1
      i1=i1+1
      dlp(i)=dls(i)-dist(i)
      go to 50
   30 min(i)=2
      i2=i2+1
      go to 50
   40 min(i)=3
      i3=i3+1
      if(persq(i).lt.shdis) shdis=persq(i)
      dlp(i)=0.
   50 continue
      if(i1.ne.jsegm)  go to 90
      iup=jsegm
      isid=2
   60 iuh=iup/2
      if(2*iuh.ne.iup) dls(iuh+1)=-dlp(iuh+1)
      if(iup.eq.1)  go to 80
      j=iup
      do 70 i=1,iuh
      t=persq(i)
      persq(i)=persq(j)
      persq(j)=t
      dls(i)=-dlp(j)
      dls(j)=-dlp(i)
      t=dist(i)
      dist(i)=dist(j)
      min(i)=2
      min(j)=2
      dist(j)=t
   70 j=j-1
   80 return
   90 if(i2.ne.jsegm)  go to 100
      isid=2
      return
  100 if(i3.gt.1) return
      if(min(1).eq.2) return
      iturn=0
      iq=1
      do 110 k=1,jsegm
      if(min(k).ne.1) go to 120
      iturn=iturn+1
  110 continue
  120 if(iturn.ne.i1) return
      if(i3.eq.1 .and. min(k).eq.2) return
c   we have single turning point
      isid=1
      inear=k-1
      if(min(k).ne.3) go to 140
      inear=k
c      adjust to 2 sep segments
      j=jseg
      do 130 i=k,jsegm
      dls(j)=dls(j-1)
      persq(j)=persq(j-1)
      dist(j)=dist(j-1)
  130 j=j-1
      jsegm=jseg
      jseg=jseg+1
      dist(k+1)=dist(k+1)-dls(k)
      dist(k)=dls(k)
      dls(k+1)=0
```

```
  140 iup=inear
      go to 60
      end
c ************************************************************
      subroutine pbreak(fm,al,bl,sigl,rlls,prls,lev,nrls)
      dimension fm(12),prls(12),rlls(10,12)
      dimension pmid(5),prang(5)
      data pmid/1.575,.74,0.,-.74,-1.575/
      data prang/.1151,.2295,.3108,.2295,.1151/
      data nrl/5/
c      evaluate al+bl*fm(l)+pmid(i)*sigl
c     where  1 <l <lev      and 1 < i < nrl
c     prang probability associated with jth break
c     for each magnitude; compute nrl rupture lengths
c      :::::::: must be ordered in descending order--largest break
c      ::::::::: length first for each magnitude
      do 10 j=1,lev
      con=al + bl*fm(j)
      do 10 i=1,nrl
      c=con+pmid(i)*sigl
   10 rlls(i,j)=10.**c
      do 20 i=1,nrl
   20 prls(i)=prang(i)
      nrls=nrl
      return
      end
c ************************************************************
      subroutine bbgaus(x,gg)
c   computes normal probability integral gg from -(infinity) to xx
c    sets gg=1 if xx < -6;   gg=0  if xx > +6
c   uses approximation Handbook of Mathematical Functions-
c   NBS Applied Math Series 55--p299 7.1.26 for error function
      ax=abs(x)/1.4142136
      if(ax-4.2426408) 20,20,10
   10 gg=0.
      if(x.lt.0) gg=+1.
      return
   20 continue
      d=1.+(((((4.30638e-5*ax+2.765672e-4)*ax+1.520143e-4
     & )*ax+9.2705272e-3)*ax+4.2282012e-2)*ax+7.0523078e-2)*ax
      d=d*d
      d=d*d
      d=d*d
      gg=.5/(d*d)
      if(x) 30,30,40
   30 gg=1.-gg
   40 return
      end
c ************************************************************
      subroutine outsid(xnot,ynot,isub,r,angle)
      common/inout/ic,ics,ifar,azimf,stsize
      common/xysav/xsav(2,50),ysav(2,50),sarea(50)
       common/xlyl/xl(4),yl(4),xr(4),yr(4),  xc(4),yc(4)
       common/slinp / aa(4,50),bb(4,50),cc(4,50),rta(4,50),dist(4,50)
     &,disq(4,50)
       common/slinds/dls(4),perp(4),sli(4),slsq(4),min(4),persq(4)
     &,srsq(4),dlp(4)
      dimension jsub(3)
c     subroutine for calculating angle when site is outside
c          (quadrilateral) source area.
c
c      r is the radial distance for which angle is sought
c
c     (xnot,ynot)=point outside quadrilateral source region isub
c     r=radius of circle (or radius of annular ring of width dr)
c     centered at (xnot,ynot)
c     isub = index identifying quadrilateral source region
c
c    the area contained in the intersection of the region and annular
c     ring is determined by the fraction of the ring within the region.
c              area = angle x r x dr
c     where angle = the angle (in radians) subtended by the
c     part of the annular ring contained within the region.
c
c          (rc is closest distance between site and source.)
c          (rf is furthest distance between site and source)
c       subroutine csqdis has computed distance to vertices of subregion
c       and shortest distance to each side
c       adapted from subroutine by mcguire
c
      data jsub/2,3,1/
      npt=0
      angle=0.
      signal=1.
c        loop on each side
      rsq=r*r
      do 140 ii=1,4
      if(min(ii)-2) 10,40,70
c     closest point is xr(ii),yr(ii)
c       at most one intersection
   10 if(rsq.gt.slsq(ii)) go to 140
      if(rsq.lt.srsq(ii)) go to 140
      d=(sqrt(rsq-persq(ii))-dlp(ii))/dist(ii,isub)
   20 yl=(yl(ii)-yr(ii))*d+yr(ii)
      xl=(xl(ii)-xr(ii))*d+xr(ii)
c  intersects in range
c          does circle (radius r) intersect it?
c        store first point
   30 npt=npt+1
      xc(npt)=xl
      yc(npt)=yl
      go to 140
c   closest point is at xl(ii),yl(ii)
   40 if(rsq.lt.slsq(ii)) go to 140
      if(rsq.gt.srsq(ii)) go to 140
   50 d=(sqrt(rsq-persq(ii))+dls(ii))/dist(ii,isub)
   60 xl=xl(ii)+(xr(ii)-xl(ii))*d
      yl=yl(ii)+(yr(ii)-yl(ii))*d
      go to 30
   70 if(rsq.lt.persq(ii)) go to 140
      if(rsq.le.slsq(ii)) go to 80
      if(rsq.gt.srsq(ii)) go to 140
c    single intersection
c     on side nearer xr,yr
      go to 50
   80 if(rsq.gt.srsq(ii)) go to 90
c     two intersections on this side
c    can compute angle directly without determining coordinates
      arg=abs(perp(ii))/r
      as=acos(arg)
      go to 100
```

```
   90 d=(dls(ii)-sqrt(rsq-persq(ii)) )/dist(ii,isub)
      go to 60
c    single intersection
c       see if this side is closest to point, if so, treat specially.
  100 if (ii-ics) 110,120,110
c          both points are on boundary, calculate angle betwen them.
  110 sign=-1.
      go to 130
  120 sign=1.
      signal=-1.
  130 angle=sign*2.*as + angle
c       see if second point only is on boundary
  140 continue
      if(npt.gt.0) go to 150
      if(signal) 310,320,320
  150 go to (320,240,160,250),npt
c     this is an error unless radius is on a vertex and
c      was counted twice.
  160 do 170 i=1,3
      j=jsub(i)
      if(abs(xc(i)-xc(j)) .ge. .01) go to 170
      if(abs(yc(i)-yc(j)) .le. .01) go to 220
  170 continue
c   check for one of three points being a vertex
      do 210 i=1,4
      do 180 j=1,3
      if(abs(xc(j)-x1(i)).gt. .001) go to 180
      if(abs(yc(j)-y1(i)) .le. .001) go to 190
  180 continue
      go to 210
  190 if(j-2) 200,200,230
  200 xc(j)=xc(3)
      yc(j)=yc(3)
      go to 230
  210 continue
      go to 320
  220 if(i.ne.1) go to 230
      xc(2)=xc(3)
      yc(2)=yc(3)
  230 npt=2
      go to 240
  240 ad=sqrt((xc(1)-xc(2))*(xc(1)-xc(2))+(yc(1)-yc(2))*(yc(1)-yc(2)))
      angle=angle + signal*2.*asin(ad/(2.*r))
      go to 310
c          four intersection points (each on a different side).
c          determine angle by finding closest 2 intersections to
c          farthest corner, calculate angle between, and add angle
c          between other two intersections.
  250 dist1=10000000.
      ii=0
      i2=0
      i3=0
      i4=0
      do 300 jj=1,4
      distn=(x1(ifar)-xc(jj))*(x1(ifar)-xc(jj))
     &  +(y1(ifar)-yc(jj))*(y1(ifar)-yc(jj))
      if (distn-dist1) 260,260,270
  260 dist2=dist1
      dist1=distn
      i4=i3
      i3=i2
      i2=i1
      i1=jj
      go to 300
  270 if (distn-dist2) 280,280,290
  280 dist2=distn
      i4=i3
      i3=i2
      i2=jj
      go to 300
  290 i4=i3
      i3=jj
  300 continue
c           calculate angle between 2 closest points to farthest corner.
      ad=sqrt((xc(i1)-xc(i2))*(xc(i1)-xc(i2))
     &  + (yc(i1)-yc(i2))*(yc(i1)-yc(i2)))
      angle=2.*asin(ad/(2.*r))
c          calculate angle between 2 points farthest from
c          farthest corner and add to previous angle.
      ad=sqrt((xc(i3)-xc(i4))*(xc(i3)-xc(i4))
     &  +(yc(i3)-yc(i4))*(yc(i3)-yc(i4)))
      angle=2.*asin(ad/(2.*r)) + angle
  310 continue
c         compute rate of earthquakes in this annular source
      return
c         error printout
  320 write(16,330)isub,ic,npt,xnot,ynot,(x1(i),y1(i),  i=1,4),r,angle,
     &(xc(i),yc(i),i=1,4)
  330 format(' ***** error in subroutine outsid.  source no. ',i3,
     &' debug values follow..... ',/10x,2i5,5(/10x,4f14.6))
  340 call exit
      end
c ************************************************************
      subroutine inside(xnot,ynot,isub,r,pangle)
      common/xyarea/xsav(2,50),ysav(2,50),sarea(50)
      common/inout/ic,ics,ifar,azimf,stsize,stepo
      common/slinds/dls(4),perp(4),sll(4),slsq(4),min(4),persq(4)
     &,srsq(4),dlp(4)
      common/slinp/aa(4,50),bb(4,50),cc(4,50),rta(4,50),dist(4,50),
     & disq(4,50)
      common/xlyl/xl(4),yl(4),xr(4),yr(4),xc(4),yc(4)
      dimension jsub(3)
c
c     (xmid,ymid)=point inside quadrilateral source region ii
c     r=radius of circle (or radius of annular ring of width dr)
c     centered at (xmid,ymid)
c     ii = index identifying quadrilateral source region
c
c    the area contained in the intersection of the region and annular
c     ring is determined by the fraction of the ring within the region.
c           area = pangle x r x dr
c     where pangle = the angle (in radians) subtended by the
c     part of the annular ring contained within the region.
c     subroutine insid determines pangle.
c
c       adapted from subroutine by mcguire
      data pi,pi2/3.1415927,6.2831853/
      data jsub/2,3,1/
      rsq=r*r
      npt=0
```

```
      angle=0.
c          loop on each side
      do 110 ii=1,4
      if(min(ii)-2) 10,40,70
c     closest point is xr(ii),yr(ii)
c       at most one intersection
   10 if(rsq.gt.slsq(ii)) go to 110
      if(rsq.lt.srsq(ii)) go to 110
      d=(sqrt(rsq-persq(ii))-dlp(ii))/dist(ii,isub)
   20 y1=(yl(ii)-yr(ii))*d+yr(ii)
      x1=(xl(ii)-xr(ii))*d+xr(ii)
c  intersects in range
c          does circle (radius r) intersect it?
c        store first point
   30 npt=npt+1
      xc(npt)=x1
      yc(npt)=y1
      go to 110
c   closest point is at xl(ii),yl(ii)
   40 if(rsq.lt.slsq(ii)) go to 110
      if(rsq.gt.srsq(ii)) go to 110
   50 d=(sqrt(rsq-persq(ii))+dls(ii))/dist(ii,isub)
   60 x1=xl(ii)+(xr(ii)-xl(ii))*d
      y1=yl(ii)+(yr(ii)-yl(ii))*d
      go to 30
   70 if(rsq.lt.persq(ii)) go to 110
      if(rsq.le.slsq(ii)) go to 80
      if(rsq.gt.srsq(ii)) go to 110
c    single intersection
c     on side nearer xr,yr
      go to 50
   80 if(rsq.gt.srsq(ii)) go to 90
c     two intersections on this side
c    can compute angle directly without determining coordinates
      arg=abs(perp(ii))/r
      as=acos(arg)
      go to 100
   90 d=(dls(ii)-sqrt(rsq-persq(ii)))/dist(ii,isub)
      go to 60
c    single intersection
c          both points are on boundary, calculate angle betwen them.
  100 angle=2.*as + angle
c        see if second point only is on boundary
  110 continue
      if (npt) 570,120,140
  120 if (angle-0.001) 570,130,130
c        following is for case of no single intersection points;
c        angle is 2 * pi - angle calculated so far.
  130 pangle=pi2-angle
      go to 560
  140 go to (570,230,150,500),npt
c     this is an error unless radius is on a vertex and
c      was counted twice.
  150 do 160 i=1,3
      j=jsub(i)
      if(abs(xc(i)-xc(j)) .ge. .01) go to 160
      if(abs(yc(i)-yc(j)) .le. .01) go to 210
  160 continue
c   check for one of three points being a vertex
      do 200 i=1,4
      do 170 j=1,3
      if(abs(xc(j)-xl(i)).le. .001) go to 170
      if(abs(yc(j)-yl(i)) .gt. .001) go to 180
  170 continue
      go to 200
  180 if(j-2) 190,190,220
  190 xc(j)=xc(3)
      yc(j)=yc(3)
      go to 220
  200 continue
      go to 570
  210 if(i.ne.1) go to 220
      xc(2)=xc(3)
      yc(2)=yc(3)
  220 npt=2
      go to 230
c          2 intersection points; determine azimuths.
  230 if (xc(1)-xnot-r) 260,250,240
  240 if (xc(1)-xnot-r-0.001) 250,250,570
  250 azim1=0.0
      go to 300
  260 if (xc(1)-xnot+r) 270,280,290
  270 if (xc(1)-xnot+r+0.001) 570,280,280
  280 azim1=pi
      go to 320
  290 azim1=acos((xc(1)-xnot)/r)
  300 if (yc(1)-ynot) 310,320,320
  310 azim1=pi2  - azim1
  320 if (xc(2)-xnot-r) 350,340,330
  330 if (xc(2)-xnot-r-0.001) 340,340,570
  340 azim2=0.0
      go to 390
  350 if (xc(2)-xnot+r) 360,370,380
  360 if (xc(2)-xnot+r+0.001) 570,370,370
  370 azim2=pi
      go to 410
  380 azim2=acos((xc(2)-xnot)/r)
  390 if (yc(2)-ynot) 400,410,410
  400 azim2=pi2 -azim2
  410 pangle=azim2-azim1
      if (pangle) 420,570,460
  420 if (azim1-azimf) 430,570,440
  430 pangle=pi2 +pangle -angle
      go to 560
  440 if (azimf-azim2) 430,570,450
  450 pangle=-pangle-angle
      go to 560
  460 if (azim2-azimf) 470,570,480
  470 pangle=pi2 -pangle -angle
      go to 560
  480 if (azimf-azim1) 470,570,490
  490 pangle=pangle-angle
      go to 560
c        four intersection points (each on a different side).
c        determine angle by finding closest 2 intersections to
c        farthest corner, calculate angle betwen, and add angle
c        between other two intersections.
  500 dist1=10000000.
      i1=0
      i2=0
```

```
      i3=0
      i4=0
      do 550 jj=1,4
      distn=(xl(ifar)-xc(jj))*(xl(ifar)-xc(jj))
     &   +(yl(ifar)-yc(jj))*(yl(ifar)-yc(jj))
      if (distn-dist1) 510,510,520
  510 dist2=dist1
      dist1=distn
      i4=i3
      i3=i2
      i2=i1
      i1=jj
      go to 550
  520 if(distn-dist2) 530,530,540
  530 dist2=distn
      i4=i3
      i3=i2
      i2=jj
      go to 550
  540 i4=i3
      i3=jj
  550 continue
c         calculate angle between 2 closest points to farthest corner
      ad=sqrt((xc(i1)-xc(i2))*(xc(i1)-xc(i2))
     &  + (yc(i1)-yc(i2))*(yc(i1)-yc(i2)))
      pangle=2.*asin(ad/(2.*r))
c          calculate angle between 2 farthest points from
c          farthest corner and add to previous angle.
      ad=sqrt((xc(i3)-xc(i4))*(xc(i3)-xc(i4))
     & + (yc(i3)-yc(i4))*(yc(i3)-yc(i4)))
      pangle=2.*asin(ad/(2.*r))+pangle
c        angle for this radius is now known, calculate risk
  560 return
c     anarea=pangle*r*stsize
c          error printout
  570 write (16,580) isub,ifar,npt,xnot,ynot,(xl(i),yl(i), i=1,4),r,
     &pangle,(xc(i),yc(i),i=1,4)
  580 format (' ***** error in subroutine inside. source no. ',
     & i3,' debug values follow..... ',/10x,2i10,10(/10x,2f12.6))
  590 call exit
      end
c*************************************************************
      subroutine spread(reg,jlo,ll,totac,dmid)
c     magnitude smoother for fault ruptures:
c     totac acceleration occurrences calculated for ruptures of
c     magnitude fm(ll) at distance dmid from site
c     are "spread" to acceleration levels corresponding
c     to magnitudes in the range
c     fm(ll)-delta/2 < fm(ll) < fm(ll)+delta/2
c     [where fm(ll-1)=fm(ll)-delta].
c     earthquake occurrences follow a Gutenberg-Richter
c     magnitude-frequency relationship, calculated from number
c     of earthquakes input at 2 successive magnitudes.
      common/magdis/jent,mdis
      common/rtb/rtab(20)
      dimension reg(106)
      common/accel/gm(25),exdif(101),atah(20,25),aclim(2)
      common/wds/max,maxp
      common/quad/maxq,ac(105),dumlog,aclog(105)
   10 continue
      if(dmid.eq.-1.) go to 60
      do 20 i=2,mdis
      if(dmid.le.rtab(i)) go to 40
   20 continue
c  distance exceeds table--we should not even be here
      write(16,30) dmid
   30 format(' in spread, dmid='e12.5' too large-should not be here')
      call exit
   40 fr=(rtab(i)-dmid)/(rtab(i)-rtab(i-1))
      ilim=1
      lp=ll+1
      do 50 lm=ll,lp
      aclim(ilim)=atah(i,lm)-fr*(atah(i,lm)-atah(i-1,lm))
   50 ilim=2
   60 acdif=aclim(2)-aclim(1)
      if(acdif.ne.0.) go to 90
      jbot=jlo+1
   70 if(aclog(jbot).le.aclim(1)) go to 80
      jbot=jbot-1
      if(jbot.ge.1) go to 70
   80 jup=jbot+1
      reg(jup)=reg(jup)+totac
      return
   90 add=0.
      jbot=jlo+1
  100 if(aclog(jbot).le.aclim(1)) go to 110
      jbot=jbot-1
      if(jbot.ge.1) go to 100
c    we are alower accel level--for lowest mag of range
  110 aclo=exdif(1)
      jup=jbot+1
  120 upl=aclog(jup)
      if(upl.gt.aclim(2)) upl=aclim(2)
      fr=(upl-aclim(1))/acdif
      iachi=100.*fr+1.00001
      quan=(+exdif(iachi)-aclo)*totac
      add=add+quan
      reg(jup)=reg(jup)+quan
      if(aclim(2).eq.upl) go to 130
      aclo=exdif(iachi)
      jup=jup+1
      if(jup.le.maxp) go to 120
      write(16,125)
  125 format(' spread routine needs higher acceleration level;'
     1/' rewrite BOX or increase scale factor')
      call exit
  130 continue
      return
      end
c*************************************************************
      subroutine out(reg,naf,xouts,youts,sd)
      common/tims/ftim(20),jtim(20),ntims,nwt,iprint
      common/rdis/rdis
      dimension qp(140),sol(20),sols(20)
      common/quad/maxq,ac(105),dumlog,aclog(105)
      dimension reg(106)
      common/ext/prob,ex,ex1
      common/wds/max,maxp,maxsp
      dimension regs(106),cdfs(105),cdf(105)
      if(iprint.le.2) write(16,10) xouts,youts,rdis,sd
```

```
   10 format(' site at long 'f8.3', lat 'f8.3,
     &/' shortest dist to fault='f9.3' km'/10x
     &'zero attenuation variability'18x'variability in atten, sigma='
     & f5.2/2( '    g.m.    occ/yr    exc/yr  r(events) r(yrs) '))
      do 20 l=1,maxsp
   20 regs(l)=0
   30 sd6=4.4*sd
      kls=1
      do 70 l=1,max
      if(reg(l+1).eq.0) go to 70
      asav=(aclog(l)+aclog(l+1))/2.
      kl=kls
   40 if(aclog(kl)-asav.ge.-sd6) go to 50
      kl=kl+1
      if(kl.le.max) go to 40
      go to 70
   50 kls=kl
      w=(aclog(kl)-asav)/sd
      call bbgaus(w,g1)
      if(kl.eq.1) regs(1)=regs(1)+(1.-g1)*reg(l+1)
   60 kl=kl+1
      wh=(aclog(kl)-asav)/sd
      call bbgaus(wh,g2)
      regs(kl)=regs(kl)+(g1-g2)*reg(l+1)
      if(wh.ge.4.4) go to 70
      if(kl.ge.maxsp) go to 70
      g1=g2
      go to 60
   70 continue
c  we are really not interested in 1st 4 acceleration boxes but used
c  them for computation of variability in acceleration--correct so
c  box 5 accumulates all accelerations below ac(5)
c  ac(5) is the first value printed
   80 continue
      regs(1)=regs(1)+reg(1)
      regsav=reg(5)
      regssv=regs(5)
      do 90 i=1,4
      regs(5)=regs(5)+regs(i)
   90 reg(5)=reg(5)+reg(i)
      cdfs(maxsp)=0.
      cdf(maxp)=0.
      maxq=maxp
      k=max
      do 100 i=5,max
      cdf(k)=cdf(k+1)+reg(k+1)
  100 k=k-1
      mm=maxsp-1
      k=mm
      do 110 i=5,mm
      cdfs(k)=cdfs(k+1)+regs(k+1)
  110 k=k-1
      if((cdfs(5).ne.0.0).and.(cdf(5).ne.0.0)) go to 140
      if(iprint.ge.2) write(16,120)
  120 format(' no acceleration events accumulated.  statistics calc.
     &bypassed')
      do 130 ier=1,ntims
      sol(ier)=0.0
  130 sols(ier)=0.
      go to 380

  140 continue
c
c  if accelerations do not exceed ac(5), the first acceleration
c  of interest, skip statistics calculation to avoid dividing by
c  zero or upsetting interpolation routines.
c
      cnums=cdfs(5)+regs(5)
      cnum=cdf(5)+reg(5)
      yrncnum=1./cnum
      nrep=0
      idone=0
      if(iprint.eq.3) go to 330
      do 300 i=5,maxsp
      if(i.gt.max) go to 150
      if(cdf(i).eq.0.0) go to 170
      if(reg(i).ne.0) go to 160
  150 if(regs(i).eq.0.) go to 190
c
c   compute return period in number of events and number of years
c   return period in events = average number of earthquakes needed
c   to produce an acceleration exceeding a
c      r(events)=(yearly earthquakes)/(yearly exceedances of a)
c   return period in years = average time between earthquakes
c   which cause acceleration a to be exceeded
c      r(yrs)=1/(yearly exceedances of a)
c
  160 if(i.le.max) retev=cnum/cdf(i)
      retyr=retev*yrn cnum
c
c   compute acceleration which has probability prob of not being
c   exceeded during t years.  this is given by the value of a
c   for which
c      prob = exp(-t * number of exceedances of a per year)
c
      go to 180
  170 retev=99999.9
      retyr=99999.9
  180 if(cdfs(i).eq.0.) go to 190
      retevs=cnums/cdfs(i)
      retyrs=retevs/cnums
      go to 200
  190 retevs=99999.9
      retyrs=99999.9
      if(cdf(i).ne.0.) go to 200
      idone=1
  200 continue
      qp(nrep+1)=ac(i)
      if(i.le.max) go to 210
      qp(nrep+2)=0
      qp(nrep+3)=0
      retev=99999.9
      retyr=99999.9
      go to 220
  210 qp(nrep+2)=reg(i)
      qp(nrep+3)=cdf(i)
  220 if(retev.ge.100000.) retev=99999.9
      if(retyr.ge.100000.) retyr=99999.9
      qp(nrep+4)=retev
      qp(nrep+5)=retyr
      qp(nrep+6)=ac(i)
```

```
      qp(nrep+7)=regs(i)
      qp(nrep+8)=cdfs(i)
      if(retevs.ge.100000.) retevs=99999.9
      if(retyrs.ge.100000.) retyrs=99999.9
      qp(nrep+9)=retevs
      qp(nrep+10)=retyrs
      if(nrep.ne.130) go to 260
      if(naf.le.3) go to 230
      write(16,280) qp
      go to 240
  230 write (16,270) qp
  240 continue
      nrep=0
  250 format(ih1)
      if(idone.eq.1) go to 330
      go to 300
  260 nrep=nrep+10
  270 format(1h ,8(2x,5,2f8.1,f12.2,2f8.5,2f8.1))
  280 format(1h ,8(4x,2f9.5,2f8.1,f12.4,2f9.5,2f8.1))
      if(retev.eq.99999.9 .and. retevs.eq.99999.9) go to 310
  300 continue
  310 if(nrep.eq.0) go to  330
      if(naf.le.3) go to 320
      write(16,280) (qp(ll),ll=1,nrep)
      go to 330
  320 write(16,270) (qp(ll),ll=1,nrep)
  330 if (iprint.gt.2) go to 350
      write(16,340) cnum,cnums,sd
  340 format(2(/'   total yearly events ' f10.5,12x)
     &//10x'zero attenuation variability'18x,
     & 'variability in atten. sigma='f4.2)
c
c  compute acceleration which has probability prob of not being
c  exceeded during t years.  this is given by the value of a
c  for which
c     prob = exp(-t * number of exceedances of a per year)
c
  350 reg(5)=regsav
      regs(5)=regssv
      do 360 jv=1,ntims
      maxq=max
      call iinint(reg,ftim(jv),sol(jv))
      maxq=maxsp
      call iinint(regs,ftim(jv),sols(jv))
      if(iprint.le.2) write(16,370)prob,sol(jv),jtim(jv),sols(jv),
     &jtim(jv)
  360 continue
  370 format(f10.3' ext prob ='f7.3' for '15' years'19x,
     & f7.3' for '14' years')
  380 rat=0.0
      if(sol(1).ne.0.0) rat=sol(ntims)/sol(1)
      ratsd=0
      if(sols(1).ne.0) ratsd=sols(ntims)/sols(1)
      if(iprint.le.2)write(16,390)jtim(ntims),prob,jtim(1),rat, ratsd
      if(iprint.ge.2) write(2) xouts,youts,(sol(jt),jt=1,ntims),
     &(sols(jt),jt=1,ntims)
  390 format('   ratio'15' yr'f6.3' extreme value to'15' yr val
     &='f7.2,7x,f7.2)
      return
      end
```


☆ U.S. GOVERNMENT PRINTING OFFICE: 1987—773-047/46,069 REGION NO. 8