RECEIVED

OCT 20 1995

OSTI

**Final Report on Repair Procedure
of Strong Ground Motion Data
from Underground Nuclear Tests**

**Thomas W. Tunnell
EG&G Energy Measurements
Los Alamos Operations**

April 1995

MASTER

**LOS ALAMOS OPERATIONS
Los Alamos, New Mexico 87544**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability of responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe upon privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 567-8401.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

**Final Report on Repair Procedure
of Strong Ground Motion Data
from Underground Nuclear Tests**

**Thomas W. Tunnell
EG&G Energy Measurements
Los Alamos Operations**

April 1995

This document is UNCLASSIFIED

Rodger Liljestrand
Authorized Derivative Classifier

Work performed for the Los Alamos National Laboratory under DOE contract no. DE-AC08-93NV11265

DISTRIBUTION RESTRICTED TO U.S. ONLY ^{4H}

MASTER

Table of Contents

Abstract	1
1. Introduction	1
2. Repair Procedure	3
A. EMP Correction	6
B. dc Correction	6
C. Noise Spike Correction	7
D. Gauge Clipping Correction	8
E. Gauge Tilt Correction	9
F. Procedure for Surface Motion	10
G. Procedure for Tunnel Motion	10
H. Input	12
I. Output	14
3. Flow and Coding	18
A. Flow	18
B. Code Summary	19
C. Detail Code Description	21
4. Examples	45
A. First Example of Easy Repair	45
B. Second Example of Easy Repair	48
C. Example of Tunnel-Repair Motion	50
D. Example of Intermediate Noise Spike Repair	52
E. Example of Setting Earliest Time for Clipping Repair	55
F. Example of Setting LSAI	58
5. Summary	61
References	61

Final Report on Repair Procedure of Strong Ground Motion from Underground Nuclear Tests

Thomas W. Tunnell
EG&G Energy Measurements

Abstract

Certain difficulties arise when recording close-in ground motion from underground nuclear explosions. Data quality can be compromised by a variety of factors, including electromagnetic pulse, noise spikes, direct current effect, and gauge clipping and gauge tilt. From March 1988 through September 1994, EG&G Energy Measurements repaired strong ground-motion data (acceleration data) from underground nuclear tests for the Los Alamos National Laboratory using an automated repair procedure. The automated repair determined and implemented the required repairs based on user input and a consistent set of criteria. A log was kept of each repair so that the repair procedure could be duplicated. This relaxed the requirement to save the repaired data. Developed for the VAX system, the procedure allowed the user to stack up a large number of repairs, plot the repaired data, and obtain hard copies. The plotted data could then be reviewed for a given test to determine the consistency of repair for a given underground test. This feature released the user to perform other tasks while the data were being repaired.

1. Introduction

From March 1988 through September 1994, EG&G Energy Measurements repaired strong ground-motion data (acceleration data) from underground nuclear tests for the Los Alamos National Laboratory (LANL). This report documents the final version of the repair procedure that was developed during this time. The repair procedure is based on the earlier work by Craig Hairston, the SANDIA code.¹ SANDIA was a user-interactive code that allowed the user to repair EMP noise, direct current (dc) imbalances, noise spikes, gauge clipping, and gauge tilt. The code displayed the repaired data on the user's terminal between each repair step. The user could also integrate the acceleration data to obtain a time history of the apparent velocity and displacement. When the user was satisfied with the "look" of the repaired data (acceleration, velocity, or displacement), the repaired data could then be stored and written to disk file. While adequate for a few data sets, such a user-intensive procedure would be inappropriate for the repair of the several hundred to thousands of data sets that were being considered. The degree of user interaction relied too much on specific user interpretation and judgment without a hard definition of what was acceptable. In addition, and more critically, no unambiguous record was kept as to the degree of repair. The work from March 1988 through September 1994 used an automated repair procedure. The automated repair determined and implemented the required repairs based on user input and a consistent set of criteria. A log (repair parameter file) was kept of each repair so that the repair procedure could be duplicated. This relaxed the requirement to save the repaired data. Developed for the VAX system, the procedure allowed the user to stack up a large number of repairs and to plot the repaired data and obtain hard copies. The plotted data could then be reviewed for a given test to determine the consistency of repair for a given test. This feature released the user to perform other tasks while the data were being repaired.

Section 2 reviews the repair procedure from March 1988 through September 1994 and identifies the repairable problems. Differences in the repair procedure between surface ground motion and tunnel motion are described. The input and output data are outlined. Section 3 documents the actual code. Four different modes of operation are identified. The flow for each mode is summarized. A listing of the routines and a brief one-sentence description of their function are presented. A more detailed description of each routine concludes Section 3. Section 4 presents several examples. This report concludes with a summary in Section 5.

The repair effort beginning in March 1988 and continuing to September 1994 evolved through four separate phases. The first phase ran from March 1988 through December 1989. It was during this first phase of effort that the basics of the automated repair procedure were developed. The most important of these was that the gauge clipping and gauge tilt repairs were implemented from a linear least squares fit to the velocity data. The fitting procedure was executed entirely by the code with no interaction required from the user other than supplying the input to run the code. Several hundred data sets were repaired during this phase, which developed versions 4, 5, and 6 of the repair procedure. The code representing version 6, AUTO6², was delivered to LANL, and further development ceased for 1 year.

Late in 1990, the second phase began with an emphasis on the vertical motion. Three modifications were developed during this phase 2 effort. The first modification was that the clipping repair was implemented by fitting changes in the acceleration to a parabola in order to round out the shape of the corrected data. The earlier versions fitted the changes to a triangle. The second modification permitted the user to specify the earliest time at which the gauge clipping and gauge tilt repairs could be implemented. For certain data sets, the first seismic arrival was as large or larger than the slap down spike. In these cases, the repair procedure would identify the first seismic arrival as the best candidate for gauge clipping and/or gauge tilt. Some philosophies regarding ground motion assume such problems occur later in the data set. That is, no problems are expected to occur with the first seismic arrival. Accordingly, the code was modified to allow the user to identify the earliest time that the gauge clipping and gauge tilt repairs would be made. The third modification allowed the user to exercise control in the noise spike repair. The modified repair generates an intermediate noise spike repair file that is discussed in more detail later in this document. This represented the version 7 repair, and a new name, automatic data repair (ADARE7), was given to this new version. The version 7 repaired data were used to predict yields in the ground-motion characterization (GMC) project.³

The third phase of repair began in March 1992. This phase was part of the effort to develop alternate user interface software for the ground motion data that previously was accessible only through a VAX-based "GEODES" data management system in the LANL closed computing environment. George Jordan of EG&G wrote the software, and Fred App of LANL moved and implemented the data base system into the LANL open computing environment on CRAY computers. Several hundred surface ground motion data sets were repaired during this phase. The repair procedure was extended to include the repair of tunnel-motion data as special cases, which is discussed in more detail later on. As part of the final documentation, a self-compensating error was discovered in the clipping repair. The error was introduced during the

modifications to the clipping repair in the second phase of effort. The modified clipping repair actually overcorrects the data. Because the repair iterates, the effect is to calculate a reduced gauge clipping error that is exaggerated the right amount to produce the proper repair. The problem is such that it is only noticeable when the version 7 repair procedure is used to regenerate repaired data using repair parameters other than version 7 files, and vice versa. The error has been corrected in this, the final version, ADARE8. When operating in the regeneration mode, the code scans the repair parameter file to determine the appropriate (i.e., consistent) clipping repair.

2. Repair Procedure

The strong ground-motion data from underground nuclear tests conducted at the Nevada Test Site (NTS) were acquired with accelerometers. Most of these accelerometers measured the acceleration of the ground motion near surface ground zero and on out. A few were implanted in tunnels under the test site and acquired tunnel motion. During recording or digitization a number of events could compromise the quality of the data. Through analysis of the digitized data, the effects of five of these events can be either diminished or eliminated. That is, the data can be repaired or corrected. These five correctable events are noise from electromagnetic pulse (EMP), improperly balanced direct current (dc) effects, noise spikes, gauge clipping, and gauge tilt.

Examples of these problems are illustrated in Figure 1. Figure 1 is generated from the ADARE8 version using a version 4 repair parameter file. The problem data are in the plot in the upper-left hand corner. These are raw data from the JORNADA event in hole U4J, surface-ground motion station GM1, vertical acceleration (ZA) from sensor GMA.

Any EMP present is short and appears as a noise spike at the very beginning of the data. An example is the large spike at the very beginning of the data in the upper-left hand corner of Figure 1. The EMP problems are corrected by zeroing the data over the EMP duration. The dc errors occur when the accelerometer produces a residual signal at zero acceleration. Part of the experimental preparation before a shot includes counterbalancing such residual signals with a bias voltage. If the instrument drifts or if the counterbalancing is done improperly, then a residual signal can persist at the time of acquisition. This residual signal creates the dc error in the data. The problem is solved by estimating the dc residual signal from the data during the "quiet time" before the first seismic arrival. The estimated signal is then subtracted from all non-EMP data. Any dc errors are generally very small and not discernible on the scale used to plot the data in Figure 1.

Noise spikes are characteristically large spikes in the data that often oscillate rapidly between large negative and large positive values. They are usually short in duration (0.05 s or less) and totally destroy any useful information over the same period. Several examples are presented in the upper-left hand corner of Figure 1. There are actually six noise spikes in these data. Five occur between the first seismic arrival and slap-down (spall closure), while the last occurs late at

nearly 4 s. The requirement to distinguish between noise spikes and the slap down spike is the most difficult task of the automated repair procedure.

The noise spike data must be replaced. However, the replacement data can not totally recover useful information. To minimize the loss of useful information, the replacement data are estimated from the data just preceding and just following the noise spike.

Traditionally, gauge clipping occurs when the signal is so large that it exceeds the linear response of the system. The data are modified so that the data recorded are usually smaller in amplitude than the true data. That is, the data are lost. When the acceleration data are integrated to produce velocity data, the missing data will be manifest as a residual non-zero velocity in the late-time velocity data. The gauge-clipping error leaves a signature that betrays the event and size of correction.

The effect of gauge clipping is shown in the lower-left hand corner of Figure 1. The data plotted are velocity data. The solid curve is the repaired data, while the dotted curve is partially repaired (EMP, dc, and noise spike repaired) velocity data. The two curves overlap until about 2.5 s. At this time there is a sharp difference between the two. The difference is the gauge clipping repair and represents the residual nonzero velocity at late time.

The same is true for gauge tilt. Gauge tilt occurs when the accelerometer tilts off axis. Because acceleration is a vector quantity, the signal for the vertical accelerometer will exhibit a bias because its bias voltage no longer counterbalances the 1 g due to the Earth's gravity. The other accelerometers may then experience some bias from Earth's gravity because their new true alignment may contain a vertical component. These new biases will appear as linear ramps in the late-time velocity data. Again, the plot in the lower-left hand corner of Figure 1 illustrates the problem. Here, the downward trend in the velocity data and definite ramp at late times betrays the gauge tilt error.

A linear least squares fit to the velocity data can estimate the gauge clipping and gauge tilt corrections. The slope, which has units of acceleration, is in fact the amount of acceleration that must be subtracted from the late time data. The intercept, which has units of velocity, is the time integral of the lost data. This fit should be performed over late time intervals of the data so that net motion of the ground is effectively zero. This interval is called the least squares averaging interval (LSAI).

It is worthwhile to point out that any error made in the EMP and noise spike corrections will be partially compensated for with the gauge clipping repair. Any error in the dc correction will be partially compensated for with the gauge tilt repair. The individual repairs and how they are parameterized are now discussed in more detail. The discussion will include the input data that are required to implement the repair and the output that are required to regenerate the repair.

Figure 1. Examples of Correctable Problems in Ground Motion Data from Underground Nuclear Tests.

A. EMP Correction

A major problem with the older data sets is the transmission of EMP signals through the recording cables. These signals occur at very early time, well before the arrival of any ground motion associated with the test. These data are recognized by unusually large signals at early time (strong EMP signals) or by an unusual occurrence in the statistics of the noise at early times (weak EMP signals). An unusual occurrence would be if the early time data has a significantly larger root-mean-square (rms) deviation than the remaining data preceding the first seismic arrival. As the EMP signal completely destroys the data, the data values are simply replaced with 0's during the detected EMP duration.

If the repair procedure fails to detect a known EMP spike, the user can force the EMP correction. This method is discussed under input parameters.

Required to implement:

1. early time estimate of EMP duration
2. estimate of first seismic arrival

Required to regenerate:

1. first non-EMP data point

B. dc Correction

A common problem in many data recording situations concerns the definition of the baseline. This is also a problem in the recording of the ground-motion data. A non-zero baseline can be the result of poor setup, drift in the electronics, or a number of other reasons. The problem is easily corrected by rebaselining the data. This is done by computing the average of the data before the first seismic arrival. This average is then subtracted from all the non-EMP data. The EMP correction is applied first.

In certain corrected data sets, it is possible to discern a nonzero displacement before the first seismic arrival. This tends to occur more with data sets that have small displacements. The nature of this signal is unknown. It is possible because the dc correction only averages the acceleration to zero before the first seismic arrival, not necessarily the velocity or displacement. An alternative to eliminate this occurrence would be to simply zero the data before the first seismic arrival. This, however, would deny the user of possibly useful noise statistics before the first seismic arrival, and therefore, it is not done.

Required to implement:

1. first non-EMP data point (from EMP repair)
2. estimate of first seismic arrival

Required to regenerate:

1. first non-EMP data point
2. correction dA to all non-EMP data

C. Noise Spike Correction

Extraneous noise spikes also occur in the recording of the ground motion. Characteristically, noise spike data are extraneously large signals that occur for no good reason. They completely mask the true nature of the ground motion. Once noise spikes are identified, they are eliminated by replacing the data within the noise spike. The data within the noise spike are replaced with a linear interpolation through the interval. The value of the replaced data at the start of the noise spike is simply the average of the data over a small window just preceding the spike. To calculate the slope for the linear interpolation, the average of the data just following the noise spike is computed over the same small window. The difference between the two averages divided by the interval defines the slope. Thus, the ending data point is equal to the average of the data just following the noise spike. Like the EMP problem, the noise spike problem tends to be more of a problem with the older data.

Noise spike identification is based upon their large magnitudes and their oscillatory behavior. Any data that exceed user-supplied thresholds is investigated as being potential noise spikes. In the investigation, the data in question are averaged in two manners. The first is just a simple average of the data. The second is an average of the absolute values. The ratio of the averaged data to the averaged absolute values is tested. If this ratio is below a certain value, the data are identified as noise spike data. The rationale for this procedure is that the oscillatory nature of the noise spike will minimize its average value, while the large oscillations will result in large absolute value averages. The ratio should then be very small.

All questionable data that occur during the least squares averaging interval are deemed to be noise spikes.

Required to implement:

1. lower (negative) acceleration threshold
2. upper acceleration threshold
3. averaging interval for noise spike identification
4. averaging interval for linear interpolation

Required to regenerate:

1. number of noise spikes
and for each noise spike
2. starting point

3. ending point
4. replacement data for starting point
5. increment in data between points (i.e., slope)

There are times when the code will miss some noise spikes and misidentify true features as noise spikes. These problems can be overcome through a two-step process. In the two-step process the code simply identifies potential noise spikes. This information is written to an ASCII file. The user can then edit the file. By editing the file the user can remove true features from the list of potential noise spikes and add other spikes as well. When the code is reexecuted, the noise spikes as defined in the file supersede the default procedure. This method is particularly useful in horizontal surface ground motions when noise spikes may be difficult to identify even for the human user.

D. Gauge Clipping Correction

Gauge clipping commonly occurs when the accelerometer is subjected to accelerations exceeding its rated value, or gauge value. To understand, consider that the ground-motion data were recorded with accelerometers. The accelerometers contain a piezoelectric (or similar) material that produces a voltage when subjected to a stress across the normal area of the material. The stress is from the force associated with acceleration. The voltage is assumed to be proportional (i.e., linear) to the stress, hence force, hence acceleration. This assumption of linearity is valid up to a value of acceleration referred to as the gauge value. For example, an accelerometer with a 20 g gauge value is expected to provide valid data for accelerations between ± 20 g.

When the acceleration exceeds the gauge value, the accelerometer is expected to fail. A common symptom of failure is for the unit to provide a constant voltage (i.e., constant acceleration value) at the level at which it failed until the true ground acceleration reenters the linear regime of the accelerometer.

The presence of gauge clipping is easily revealed as a constant offset in the velocity or a linear ramp in the displacement at late times. The degree of gauge clipping can be determined as the intercept in a linear least squares fit to the late-time velocity, or slope in a quadratic fit to the late-time displacement. The fitting procedure requires that some interval in the late-time data be defined for the fit.

Due to the nature of gauge clipping the lost data are assumed to belong to the strongest peak. Thus, in general, the data are returned to the strongest peak. When the lost data are returned, the acceleration values are simply added to the data at the strongest peak. The data are added in the form of a parabola so that the area under the parabola equals the gauge clipping error. Some minimum duration for a clipping event should be defined as well. Otherwise, if the clipping correction is applied over too small an interval, the repaired data may take on unreal values.

If no one single peak is dominant, then the gauge clipping is added to the strongest peaks. The amount added to each strong peak is determined by its relative strength. This is admittedly an imprecise procedure, but is as reasonable as any other in the absence of other information.

In some circumstances, a phenomenon very similar to gauge clipping (weak gauge clipping) will occur completely in the absence of strong peaks. In these situations, the data are smaller (typically a few g or less) but are rapidly oscillating. What we believe happens is that the entire recording system acts like a low-pass filter that eliminates the highest frequencies. The result is identical to gauge clipping. However, the criteria for identifying where to replace the lost data have changed. Rather than simply replacing data at the strongest peaks, the data are replaced at the strongest peaks where the data are changing most rapidly.

Instruments being what they are, many units begin to deviate from linearity at near 80% of their rated gauge value. To cope with such situations, gauge clipping is always assumed to be present. If the scan of the data shows that none of the peaks exceed 80% of the gauge value, then any gauge-clipping error is assumed to be of the weak gauge clipping variety. The result is that a gauge-clipping error is always calculated and corrected. If there is in fact no gauge clipping at all, then the calculated gauge-clipping error will be very small so that the "correction" will not alter the character of the true data.

Required to implement:

1. gauge value
2. minimum duration of clipping event
3. LSAI defined.

Required to regenerate:

1. number of clipping events
and for each clipping event
2. starting point
3. ending point
4. degree of repair

E. Gauge Tilt Correction

Gauge tilt occurs when the instruments themselves tilt. When this happens, the axis of horizontal accelerometers will be subjected to a new force, namely gravity, and the axis of the vertical accelerometers will be subjected to less force. The result is that the units will have a new baseline, starting at the time of tilt. When present, the degree of gauge tilt can be determined from a linear ramp in the velocity, or from a parabola in the displacement at late times. Gauge tilt is assumed to occur only once and at the strongest peak.

Thus, the gauge-tilt correction must identify where the gauge tilt occurred and how to construct the new baseline. The new baseline is constructed by subtracting a constant from the acceleration data for all data points following the gauge tilt. Thus, the gauge tilt repair can be parameterized in terms of a time when gauge tilt occurs and a correction, Δa , that is to be added or subtracted from the data after gauge tilt.

Required to implement:

1. LSAI defined.

Required to regenerate:

1. time that gauge tilt occurs
2. correction Δa to acceleration data after gauge tilt

F. Procedure for Surface Motion

The surface ground-motion data are repaired in the following procedure.

1. EMP corrections are applied (if needed)
dc correction
2. noise-spike correction
3. gauge-clipping and gauge-tilt corrections (together)

A very important feature is that the gauge-clipping and gauge-tilt corrections are determined simultaneously from a linear fit to the velocity data. The fit is applied to a late time interval after the motion has settled down. There were a few data sets where the very last portion of the data was not included. This was because of some late-time data failure or severe change in the character of the data. These data sets are unrepaired with the procedure described in this report.

Because of numerical limitations, the fit is applied through a series of iterations. Each iteration provides a better estimate of the gauge-clipping and gauge-tilt corrections.

Experience has shown that any change in the acceleration, unless balanced by an opposite change somewhere, will be greatly amplified in the displacement. Accordingly, the interval used to determine the gauge-clipping and gauge-tilt corrections is chosen to minimize the final average displacement.

G. Procedure for Tunnel Motion

The procedure for repairing the tunnel-motion data was modified for three reasons. The first was that one quantity of interest in the tunnel-motion data is the final displacement. The procedure established for repairing the surface ground motion was based on minimizing the final

displacement. Such an approach applied to the tunnel-motion data would improperly bias the result in favor of minimum displacement. Rather, it was decided to exploit the long record length (20 s or more) of the data. A time interval after the motion had subsided was then selected from which to obtain the information to determine gauge-clipping and gauge-tilt errors.

The second reason was that the late-time ground motion included significant low-frequency signals (below 1 Hz). These were suspected to be reverberations, but their true nature is unimportant for repair purposes if it is assumed that the final displacement is determined by the high-frequency energy (above 1 Hz) contained in the earliest part of the shock wave. The problem with the late-time low-frequency signals is that their presence compromised the calculation of the gauge clipping and gauge tilt errors. The periods of the low-frequency signal exceeded that of the averaging interval so that their presence influenced the final average. In short, the gauge clipping and gauge tilt error information was masked by the low-frequency motion. For this reason it was decided to high-pass-filter the data. A simple ($f/1.31 \text{ Hz}$)⁶ filter was applied to the data below $f = 1.31 \text{ Hz}$. No filtering was applied to data above 1.31 Hz. This filter passes 50% of the amplitude near 1.2 Hz and 20% near 1 Hz.

The third reason was one of numerical accuracy. The long times and larger dynamic range in the data associated with the tunnel-motion data presented a special numerical problem. The problem is that the resolution of the repair parameters exceeds their true accuracy. This is somewhat evident in the iterative procedure in which the fit to the data is applied. Each iteration increases the resolution of the repair. In the surface-motion data, no iteration needs to be performed in the regeneration; the errors are so small as not to be noticeable. However, because of the longer time intervals, the errors are noticeable in the regeneration of the tunnel-motion repair. As a result, the tunnel-motion data are repaired and stored as part of the data base.

In the repair procedure, the filtering was performed before any of the other repair steps. The steps were then similar to those for the repair of the surface ground-motion data.

Tunnel-motion data were repaired in this fashion for MINERAL QUARRY (U12N22), MISTY ECHO (U12N23), and HUNTERS TROPHY (U12N24). These data sets were repaired consistently. Data for the Non-Proliferation Experiment (U12N25) presented unique problems.

The U12N25 data were unique because the character of the data appeared to change after the first 2-3 s of data acquisition. The reason for this is unknown. However, the effect was that there was no easy way to correlate the gauge-clipping and gauge-tilt corrections derived from the late part of the data to the actual errors. In general, the repair was so bad that the unrepaired velocities generally looked more acceptable than the repaired velocities.

The decision was made to derive the repair parameters from the 0.50-2 s time interval. This choice reflects the best compromise of selecting good data while performing the fitting procedure over a relatively quiet interval.

H. Input

The input required to run the code is now described. The input can be separated into procedural data and parameters. The procedural data involves the following:

1. describing the type of data , e.g., data from a mini-data base (GMGET) file, ASCII file, or seismic analysis code (SAC) file
2. repair mode
3. requesting the code to plot the data
4. requesting the code to write the repaired data to an ASCII file

To describe the type of data, the user answers with G, A, or S when prompted by the computer. The different modes of repair are described under the flow in the next chapter. The responses to plot and write the repaired data are simple Y (yes) or N (no) answers that are self-explanatory.

The input parameters that really affect the repair of the data are discussed next. These are all the input parameters that were just previously identified as being "required to implement" repair. Satisfactory input parameters can normally be estimated from examining plots of the data.

These input parameters are as follows:

1. filename to be repaired
2. early time estimated duration of EMP, in seconds
3. estimated time for the first seismic arrival, in seconds
4. lower threshold for detecting noise spikes, in g's
5. upper threshold for detecting noise spikes, in g's
6. start time for the least squares averaging procedure, in seconds
7. end time for the least squares averaging procedure, in seconds
8. estimated gauge value, in g's
9. duration of averaging time interval used for noise spike identification
10. duration of averaging time interval used for noise spike repair
11. duration of averaging time interval used for gauge clipping identification
12. minimum duration of gauge clipping event
13. earliest time for clipping and tilt repairs

Filename

If the data set to be repaired is to be read from a GMGET mini-data base, the user must encode the filename in the following manner. The filename must be encoded as VVVV_XXXX_YYYY_ZZZZ where VVVV is the holename (e.g., U4J), XXXX is the station name (e.g., GM1), and YYYY_ZZZZ defines the measurement (e.g., GMA_XA, GMB_YA, GMA_ZA, etc.). The nomenclature XA, YA, and ZA define the data sets as horizontal x-, horizontal y-, and vertical components respectively. The notation GMB_ZA indicates that there were at least two vertical (ZA) acceleration data sets at the specified ground-motion station. The

GMB qualifier identifies the data set as the second vertical ground-motion data set. Along this line, GMC_ZA would be the third vertical ground-motion data set.

If the input data set is an ASCII file, there is no restriction to the file name other than it must be identified as an ASCII file when the user is prompted. .asc specifier. The ASCII file must be formatted so that the data set name is on the first line. The second line should contain the number of data points and sample interval. The time series then follows.

The user may also specify a SAC file for input.

Early time EMP duration

Typically, a good estimate of the early time EMP duration is 0.05 s. The user can force an EMP correction out to a specified time by entering the negative of the desired time. For example, a value of -0.03 would force the code to implement the EMP correction between times $t=0$ and $t=0.03$ s.

Estimated first seismic arrival

Precise accuracy of the first seismic arrival is not critical. Visual observation of plotted data may be good enough if the first seismic arrival is after 0.30 s. However, if an error is to be made it should be in underestimating the first seismic arrival. If too large a value is used, the dc correction will include some of the motion data, which would be incorrect.

Lower and upper noise spike thresholds

If the data are clear and free of noise spikes, the noise spike repair can be turned off. This is accomplished by entering a lower threshold that is below any of the data values and an upper threshold that is above any of the data values.

Gauge value

If the data are being read from a GMGET mini-data base, the header information will generally contain the gauge value for the accelerometer. If the code fails to detect gauge tilt by the default procedure, the alternative procedure may induce some artifacts. If this happens, some control over gauge clipping repair can be exercised by the user. If the user specifies a negative value for the gauge clipping, then the code will accept the absolute value as the gauge value. In such cases, values at roughly 80% of the slap-down spike generally produce good results.

Averaging interval for noise spike identification

This is the length of the interval used in the absolute value and data averaging to implement the noise spike recognition. This time interval is eventually parameterized as $NW \, dt$ in the code where dt is the sample interval.

Averaging interval for noise spike repair

This is the length of the interval used in the data averaging just preceding and following identified noise spikes. The replacement data are derived from these data averages. This time interval is parameterized as NA dt.

Time interval for alternative gauge clipping identification

This is the length of the interval used to average the acceleration data to identify gauge tilt and when the alternative method to identify clipping events is used. The clipping repair procedure follows the alternative method when the default procedure fails. The alternative method scans the average data to identify the largest peaks.

Minimum duration of gauge clipping

This time interval specifies the minimum duration of the clipping events. Unless a minimum duration is specified, the clipping repair may be applied to a too-narrow time interval, in which case the repair causes unrealistic values.

Earliest time for clipping and tilt repairs

This time defines the earliest time that the code will attempt to locate gauge clipping and gauge tilt events. If not set after the first seismic arrival, the code may identify the first seismic arrival as a clipping or tilt event and "repair" it. Depending on the philosophy of the user, a repair of the first seismic arrival may or may not be acceptable. This parameter allows the user to repair the data according to a particular philosophy.

I. Output

This section describes the data format for the repair parameter files, intermediate spike repair files, and ASCII format for the repaired data. These output files contain the data set name and repair version (currently 08) with 012, 013, and 016 extensions for repair parameter, intermediate noise spike repair, and repaired data set, respectively. The repair parameter file format is discussed first.

Repair Parameter File

The repair parameter file for the repaired data plotted in Figure 1 is used as an example to follow this description of the repair parameter file. The repair parameter file is shown in Figure 2.

```

AUTO-4
U4J_GM1_GMA_ZA
175 620 3.844E-02 3.854E-02 1.336E-01
6
2123 2246 -1.001E+00 -2.979E-04
3260 3513 -1.042E+00 9.535E-05
3870 4023 -1.030E+00 -4.078E-05
4447 4702 -9.783E-01 -4.018E-05
4826 4979 -8.667E-01 2.750E-03
7895 8033 -3.689E-01 1.189E-03
1 5062 10307 13996 4.353E-01 1.226E-01
5027 5131 1.000E+00
-2.841E-03 7.416E-07 -1.623E+00
1.00000E-01 3.10000E-01 4.00000E+00 7.00000E+00
-2.00000E+00 1.00000E+01 1.50000E+01
2.50000E-02 2.50000E-02 2.50000E-02 2.50000E-02
0.00000E+00 0.00000E+00 0.00000E+00

```

Figure 2. Example of Repair Parameter File.

The repair parameter files contain six basic parts, as follow.

Part 1 - The first part of the repair parameter file contains quality control information. This information identifies the following:

1. the version of the repair code (i.e., AUTO-4) on line 1.
2. the data set that was repaired (i.e., U4J_GM1_GMA_ZA) on line 2.

Part 2 - Part 2 begins the tabulation of the actual repair parameters on line 3. Line 3 contains

LAST	N0	AVE	DC	RMS
------	----	-----	----	-----

where the integer LAST (175) is the last EMP data point (LAST = 0 if no EMP), the integer N0 (620) is an estimate of the first seismic arrival, the real variable AVE (3.844E-02) is the average acceleration data between LAST and N0, the real variable DC (3.854E-02) is the dc correction (slightly different from AVE), and the real variable RMS (1.336E-01) is the root-mean-square deviation in the signal about AVE between LAST and N0. The EMP correction is contained entirely by the integer LAST, while the dc correction information is contained by LAST and DC.

Part 3 - Part 3 begins the noise-spike corrections on line 4 as

N_SPIKE

where the integer N_SPIKE (6) is simply the number of noise spikes. If N_SPIKE is greater than 0, then for i equal 1 to N_SPIKE, the following information is presented

ISI(i) ISF(i) SP1(i) SP2(i)

The integers ISI(i) and ISF(i) are, respectively, the beginning and ending data points of the i^{th} noise spike. The real variables SP1(i) and SP2(i) are the corresponding value of the data at ISI(i) and slope of the line that is used to interpolate through the noise spike. The corresponding values for the first spike are 2123, 2246, -1.001E+00, and -2.979E-04 respectively.

Part 4 - Part 4 begins the listing of the gauge clipping and gauge tilt corrections. On the first line following the noise spike correction information, the line

NC IL2 N1 N2 VRC ADC

is given. The integer NC (1) is the number of clipping events, and integer IL2 (5062) identifies the data point where the gauge tilt correction should begin. The integers N1 (10307) and N2 (13996) identify the beginning and ending points of the interval over which the linear fit was performed in extracting the gauge-clipping and gauge-tilt correction. The integers N1 and N2 are used in the regeneration but are listed as part of the documentation since the algorithm is allowed to choose the averaging interval. The real variables VRC (4.353E-01) and ADC (1.226E-01) are, respectively, the gauge-clipping and gauge-tilt errors. Units are g-s and g.

The gauge-tilt correction is implemented simply by adding ADC to data points after IL2. The gauge-clipping correction is more involved.

Completion of the gauge-clipping correction requires additional input. This input is provided in the following NC lines. Each line contains

IT1(i) IT2(i) WT(i)

where the integers IT1(i) and IT2(i) identify, respectively, the beginning and ending data points of the i^{th} clipping event. WK(i) is the corresponding weight. The total correction for clipping event i is $VRC \cdot WT(i)$. This amount of velocity is then distributed over the interval IT1(i) to IT2(i). It is distributed so that the integrated acceleration over this interval is larger (or smaller, whichever is the case) by an additional amount of $VRC \cdot WT(i)$. These values for the one clipping event for this data set are 5027, 5131, and 1.000E+00 respectively. The degree of repair for this event is 4.353E-00 times 1.000E+00.

Part 5 - Part 5 presents the final results of the repair. The average acceleration (g), average velocity (m/s), and average displacement (in m) obtained over the final averaging interval are presented next. These are, respectively,

AVE_A AVE_V AVE_D

These final averages are presented for comparison purposes only. For these data sets, they are -2.841E-03, 7.416E-07, and -1.623E+00 respectively.

Part 6 - Part 6 completes the repair parameter file. The information presented here is the information that was input to the repair code that generated the repair results just described. This information is included as part of the quality control of the repair. The repair for this data set was generated with the following:

1. early time estimated duration of EMP = 0.10 s
2. estimated first seismic arrival = 0.31 s
3. lower spike threshold = -2 g
4. upper spike threshold = 10 g
5. start time for the LSAI as 4 s
6. end time for the LSAI as 7 s
7. estimated gauge value = 15 g
8. averaging time interval for noise spike identification = 0.025 s
9. averaging time interval for noise spike repair = 0.025 s
10. averaging time interval for gauge tilt identification = 0.025 s
11. minimum duration of gauge clipping event = 0.025 s
12. earliest time for clipping and tilt repairs = 0 s

Intermediate Noise Spike Repair File

When the user chooses to exercise more direct control of the noise spike repair procedure, the code scans the data to generate an intermediate spike repair file. This file lists all suspicious data as defined by the user. The user may then edit the file and specify the nature (i.e., spike or not a spike) of the data. The format for the intermediate noise spike repair file is the following. For each potential noise spike that is identified, the code generates one line of data:

I1	I2	T1	T2	SPIKE
----	----	----	----	-------

I1 and I2 are the starting and ending indices of the noise spike respectively. The times T1 and T2 are the corresponding times. The parameter SPIKE is always written as -1. The last line of the file is written with I1 equal to -1. The -1 value identifies this line as the end of the data set.

The intermediate noise spike repair file is recognized by its 013 extension. The user may then print out the intermediate noise spike repair file and review it with a plot of the data. Potential noise spikes can then be removed by either deleting the line or changing the SPIKE value for the line to 0.

Repaired data ASCII file

When specified by the user, the code will generate an ASCII file for the repaired data. The data set name that was repaired is written on the first line. The number of data points and sample interval are written on the second line. The data are written in F12.5 format from the third line to the end of the data set.

3. Flow and Coding

This chapter describes the flow and coding of the repair code. The flow is discussed first.

A. Flow

The flow depends on the execution mode. The following four modes are possible:

1. a complete repair
2. regeneration repair
3. incomplete repair
4. resumption of repair

Selection of the repair mode is the first input that the user provides.

Mode 1 is the complete repair mode. This is the most critical repair mode as it is used to repair almost all of the data. The flow for this mode is illustrated in Figure 3. The first step in this mode is to read the input parameters. The second step is to read the data that are to be repaired. The repair procedure then begins with the EMP, dc, filtering (if tunnel-motion), and noise spike repairs. The details of the noise spike repair involves two phases. The first phase identifies the noise spikes. The second phase implements the repair of the identified noise spikes. The intervals are then determined for the minimum least squares averaging. The minimum least squares averaging is then performed through a series of iterations to complete the gauge clipping and gauge tilt repairs. Repaired data are then written or plotted as requested by the user.

Read Data

Read Repair Parameters
Read data

Repair Data

STEP 1 Repairs
EMP repair
dc imbalance repair
filter if tunnel motion
STEP 2 Repairs
Identify noise spikes
Repair noise spikes
STEP 3 Repairs
Gauge clipping and tilt repairs

Figure 3. Plot and Write Repaired Data

Mode 2 is the regeneration repair mode. In this mode, the code regenerates the repaired data from the repair parameter file. Figure 4 maps out the flow for this mode. The flow is fairly simple. The repair parameter file and acceleration data are first read. The repair parameter specifies all the repairs that are to be performed so that they are easily implemented. Repaired data are then written or plotted as requested by the user.

Read Data

- Read repair parameters
- Read data

Repair Data Regeneration

- EMP repair
- dc imbalance repair
- filter if tunnel motion
- Repair noise spikes
- Gauge clipping and tilt repairs

Figure 4. Plot and Write Repaired Data

Noise spike removal can be troublesome in some cases. In these cases, the user may want to exercise greater control over identification of the noise spikes. This is accomplished through the combination of the repair modes 3 and 4. In mode 3, the flow begins as in mode 1. However, as the noise spike repair procedure begins, the code generates the intermediate noise spike repair file and then terminates. The user then edits the intermediate noise spike repair file, which is identified by its 013 extension.

In mode 4, the flow again begins as in mode 1. However, when the code should be beginning the noise spike repair, it reads the intermediate noise spike repair file. It then uses this information to determine which data are noise spike data. The flow according to mode resumes with implementation of the noise spike repair. Repaired data may then be written or plotted as requested by the user.

B. Code Summary

The repair procedure consists of four main tasks. These are overhead and input/output operations, EMP and dc repair, noise spike repair, and gauge clipping and gauge tilt repair. The individual routines can be separated according to the task that they belong to. In the summaries that follow, the routines are separated according to the overall task that they represent and listed. The listing includes a brief, one-line description of the routines task. More detailed descriptions follow later in the Code Detail section.

Overhead and I/O:

The overhead and input/output operations describe the first task. This includes actions not necessarily specific to individual repair steps, such as integration of data. Data are integrated to implement the gauge clipping and tilt repairs as well as computing error bars.

AUTOP	: main code
CODER	: determines hole name, station name, etc. for GMGET data sets
DV4	: estimates error bars in repaired data
N_GEN	: reduces time parameters into indexed window widths
P401	: plots raw and repaired data
READ_ASCII	: reads data from specified ASCII file
READ_PARAM	: reads input parameters
READ_SAC	: reads data from specified SAC file
READ_SUB	: reads data from GMGET file
STEP1	: integrates data as function of time

STEP 1 **Repair: EMP, dc, and Filtering**

The actual repair process begins with the Step 1 repairs, namely the EMP and dc corrections. This includes the filtering of the tunnel-motion data.

DC_C	: determines and implements dc repair
EMP_C	: determines and implements EMP repair
FILT	: high-pass filters tunnel motion
FOUR1	: discrete Fourier transform for filtering
NDFT	: properly normalized discrete Fourier transform

STEP 2 **Repair: Noise Spike**

The Step 2 repair is the noise spike repair. During this repair step, the data are scanned to identify possible noise spikes. The questionable data are separated into intervals of a minimum width specified by the user. The questionable data are then processed with the data averages and absolute value averages being computed over each interval.

SP_C	: determines and implements repair for identified noise spikes
SPT_C1	: scans Step 1 repaired data to identify possible noise spikes
SPT_C2	: determines intervals for possible noise spike data
SPT_C3	: identifies possible noise spikes as either valid data or noise spikes

STEP 3 **Repair: Gauge Clipping and Gauge Tilt**

ARB NX	: finds absolute minimum and maximum of a data set
CLIP_1	: default procedure for identifying gauge clipping

CLIP_11	: alternative method for identifying gauge clipping
CLIP_2	: implements gauge clipping repair
FIX_VEL	: control code for minimum least squares fit
TILT_1	: computes data averages
TILT_2	: finds largest peaks of data averages; largest defines when tilt occurs
TILT_C	: implements tilt correction as determined by TILT_2 and the minimum least squares averaging procedure
SQD_L	: linear least squares fit to velocity
SQD_Q	: quadratic least squares fit to displacement
SQD_XY	: estimates minimum displacement interval for horizontal data set
SQD_Z	: estimates minimum displacement interval for vertical data sets
WT_NEW	: recomputes new clipping weights for default method
WT_N11	: recomputes new clipping weights for alternative method

C. Detail Code Descriptions

PROGRAM ADARE8

Purpose:

ADARE8 is an acronym for automatic data repair. It is the main code for the updated version (version 8) of the automatic data repair code, AUTO6. AUTO6 was the first version of automatic data repair of ground-motion data from underground nuclear tests that was delivered to LANL. The current version repairs the following:

1. EMP noise
2. direct current (dc) imbalances
3. noise spikes
4. gauge clipping
5. gauge tilt

Enhanced capabilities over earlier versions include the following:

1. forcing EMP repair
2. forcing selection of least squares averaging interval
3. allows user to specify gauge value
4. allows greater user interaction in noise spike repair
5. allows user to specify earliest time for clipping and tilt repairs
6. repairs tunnel-motion data
7. can read ASCII and SAC files

Parameters:

The parameters for the repair procedure are read in with the call to READ_PARAM. The parameters are returned via common blocks.

1. T_EMP	= early time estimated duration of EMP
2. T0	= estimated first seismic arrival
3. S_LOW	= lower spike threshold
4. S_HIGH	= upper spike threshold
5. T1	= start for the LSAI
6. T2	= end time for the LSAI
7. PEG	= estimated gauge value
8. TSPIKE	= averaging time interval for noise spike identification
9. TAVE	= averaging time interval for noise spike repair
10. T_TILT	= averaging time interval for gauge tilt identification
11. T_WIDTH	= minimum duration for clipping event
12. T0_CLIP	= earliest time for clipping and tilt repairs

Other important parameters and variables include the following:

RAW(i)	- raw input data
DT	- sample interval
TIME(i)	- time values for each data point = (i-1)DT
R2(i)	- partially repaired data
R3(i)	- completely repaired data
NPTS	- number of data points
N0	- sample point for estimated first seismic arrival
N1	- sample point for start of LSAI
N2	- sample point for end of LSAI
NA	- averaging interval for noise spike repair
NE	- estimated interval for EMP
NS	- minimum interval for noise spike duration
NT	- averaging interval for tilt determination
NW	- minimum interval for clipping event

Common Blocks:

```
CHARACTER*80 FILE,FNM1,FILE12,FILE13
CHARACTER ANSWER*3,ANSWER1*3,HOLE*10,STATION*12,SENSOR*10
COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR
COMMON/PROC_PARAM/T_EMP,T0,T1,T2,S_LOW,S_HIGH,PEG,
1    TSPIKE,TAVE,T_TILT,T_WIDTH
COMMON/PROC_NEW/T0_CLIP
COMMON/BLCK1/RAW(108000),R2(108000),R3(108000),
```

```
1      TIME(108000),NPTS  
COMMON/BLCKN/N0,N1,N2,NA,NE,NS,NT,NW  
COMMON/BLCKP/GAUGE  
DIMENSION DATA(108000,2)  
INTEGER*2 CONSOL/8/  
DATA I_VER/8/
```

Dependencies:

AUTOP4, CODER, N_GEN, READ_ASCII, READ_PARAM, READ_SAC, READ_SUB,
REPAIR, RERUN06,

SUBROUTINE ARBNX(VI,IMIN,IMAX,NPTS)

Purpose:

ARBNX scans the real array to identify the data points (IMIN and IMAX) with the minimum and maximum absolute value. NPTS is the number of points in the array.

Parameters:

VI(I) - real array
IMIN - index of data point with minimum magnitude
IMAX -- index of data point with maximum magnitude
NPTS -- number of data points in array VI

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE AUTOP4

Purpose:

AUTOP4 prepares the raw and repaired ground motion data for plotting. This includes integrating the acceleration data to produce the velocity and displacement data. Axis labels and header information are included.

As of August 1994, three pages of plots were generated with four plots per page. On a given page, the lower-left hand plot is prepared first, then the upper-left hand plot, then the lower-right hand plot, and then the upper-right hand plot.

On the first page, the information corresponds to the velocity data, raw acceleration data, the displacement data, and the repaired acceleration data, respectively. In the velocity and

displacement plots, the solid curves correspond to repaired data, the dotted curves are the partially repaired data (EMP and dc corrections only), while the dashed curve is an estimate of the error bars. After all the data to be plotted have been calculated, they are then processed with the call to SIFTER. SIFTER resamples the data at a coarser sample interval so that fewer data points are plotted. The data are then plotted as in Figure 1. The same information is plotted on the second page, but at smaller time scales to show detail near peak of pulse. The third page shows the four major corrections (in terms of clipping) to the acceleration data. The first major correction (upper-left hand corner) is repeated when there are fewer than four clipping events identified.

All information is passed via common block. Note that the statement:

CHARACTER*1 ICN

is current DISSPLA authorization code requirement. This may change in time and with location license.

Dependencies:

DV4, P401, SIFTER, STEP1, and DISSPLA codes GDONE, GPLOT, LIBDISP

Parameters:

None. Information is passed via common blocks.

Common Blocks:

```
CHARACTER HOLE*10,STATION*12,SENSOR*10,FNM1*80,FILE*0
COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR
COMMON/BLCK1/RAW(108000),R2(108000),R3(108000),
      TIME(108000),NPTS
COMMON/BLCK2/VI(108000),DI(108000)
COMMON/BLCKE/EV1(108000),EV2(108000)
COMMON/B1993/IWT_1,IWT_2,IWT_3,IWT_4
```

SUBROUTINE CLIP_1(TIME,R3,GAUGE,IT1,IT2,IT0_CLIP,WT,NC,NW,N1)

Purpose:

CLIP_1 identifies the clipping events based on comparison of absolute magnitude of data (real array R3) with the supplied gauge value (GAUGE). The search for gauge clipping events is restricted to the interval between IT0_CLIP and N1 (starting index for the least squares averaging interval). The IT0_CLIP value is usually set to either 0 or some time after the first seismic arrival to gauge and tilt corrections after the first seismic arrival. Because the least squares averaging is performed over late-time intervals after the ground is assumed to be quietest, gauge clipping is assumed to have occurred before this interval. This is why gauge clipping is restricted to points between IT0_CLIP and N0.

The magnitude comparison is the default means of clip identification. For each clipping event that is identified, CLIP_1 determines the beginning (integer array IT1) and ending (integer array IT2) data points. The gauge clipping is assumed to begin when the magnitude of the data exceeds the gauge value. After it begins, clipping is assumed to occur until the magnitude of the data relaxes below the gauge value divided by 2.72.

However, there is a minimum duration that clipping is assumed to last, which is NW (in terms of sample intervals). Clipping is not allowed to relax until it after it lasts at least NW samples. The reason for this is that if the gauge clipping event is allowed sooner in some cases, the repaired acceleration may exceed believable values.

The code can identify more than one clipping event. In such cases, a relative weight is assigned to each clipping event. The weight is based on the sum of the absolute value of the data during the clipping event.

Parameters:

TIME(I)	- time array
R3(I)	- input/spike corrected acceleration data
GAUGE	- gauge, acceleration value at which clipping occurs
IT1(I)	- index identifying start of i^{th} clipping event
IT2(I)	- index identifying end of i^{th} clipping event
WT(I)	- weight of i^{th} clipping event
NC	- number of clipping events
NPTS	- number of data points
N1	- starting index for the averaging interval

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE CLIP_2(X,Y,WT,IT1,IT2,VR,NC,NPTS)

Purpose:

CLIP_2 implements the clipping corrections for a total velocity deficiency (i.e., clipping correction) of VR. The clipping correction for an individual clipping event is $WT(I)*VR$ where $WT(I)$ is the relative weight for the event.

The clipping correction was originally made by adding the correction to the data. The added data formed the shape of a triangle. The shape of the modified data has since been modified as a parabola to show a more rounded data peak.

The correction is made in two steps. The first step simply mimics the repair with a unit correction. The true effect of the correction is then scaled to $WT(I)*VR$. The two-step procedure is followed to minimize numerical error.

In preparation of this document, an error was discovered in the repair code. The clipping correction was being overcompensated. However, since the repair was performed over many iterations, the code compensated for it. The result is the following. All version 07 regenerations have the gauge clipping correction increased by 50%. All other versions are made without modification.

Parameters:

X(I)	- in time array
Y(I)	- input/clipping repaired acceleration data
WT(K)	- weight of K^{th} clipping event
IT1(K)	- index identifying starting point of i^{th} clipping event
IT2(K)	- index identifying ending point of i^{th} clipping event
VR	- total clipping correction: DOUBLE PRECISION
NC	- number of clipping events
NPTS	- number of data points.

Common Blocks:

None.

Dependencies:

None.

**SUBROUTINE CLIP_11(R3, S, WT, IT1, IT2, IT0_CLIP,
NT, NW, NC, N1)**

Purpose:

When the default procedure for determination of clipping events finds no clipping events, CLIP_11 identifies clipping events on a combined amplitude and slope basis. The idea assumes that the data are changing faster than the band limitation of the system can respond. In this case, the recording system acts like a low-pass filter and eliminates high-frequency information. Combining amplitude and slope is an attempt to identify segments in the data where high-frequency information may have been lost.

CLIP_11 works in the following manner. First, the partially repaired data (EMP, dc, and spike) are summed over windows of width NT (done in call to TILT_1). This is equivalent to a low-pass filter. This processed data will still contain high-frequency information, particularly in the peaks. The largest peak of the summed data is identified first (in call to TILT_2). Its relative weight is determined by the point-to-point squared differences

(roughly the local slope of the data). So this summation is computed over the interval NT centered about this first peak. The next largest peak that is separated by at least NW data points is then identified. The steps are iterated until ten peaks have been processed.

CLIP_11 can be unstable and produce acceleration peaks that are obviously wrong. In such cases, the user can force the default procedure to find clipping events by entering a negative value for the gauge parameter. The absolute value is forced as the gauge parameter and overrides the entry from the data base.

Parameters:

R3(I)	- input acceleration data
S(I)	- data sum values
WT(I)	- weight of i^{th} clipping event
IT1(I)	- index of starting point of i^{th} clipping event
IT2(I)	- index of ending point of i^{th} clipping event
NT	- length of averaging window for clip determination
NW	- minimum width of clipping events
NC	- returned number of clipping events
NPTS	- total number of data points

Common Blocks:

None.

Dependencies:

TILT_1, TILT_2

SUBROUTINE CLIP_T(TIME,WT,TW,IT1,IT2,NC)

Purpose:

CLIP_T computes an effective time, TW, at which all clipping events are assumed to occur. This time TW is used to estimate the final displacement without performing detailed calculations. These estimates of the final displacement are the means by which the code selects the least squares averaging interval that minimizes displacement. This effective time is computed as the weighted average of all clipping events.

Parameters:

TIME(I)	- time array
WT(I)	- weight of I^{th} clipping event
TW	- effective clipping time
IT1(K)	- index of starting point for k^{th} clipping event
IT2(K)	- index for ending point for k^{th} clipping event
NC	- number of clipping events

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE CODER(IFO)**Purpose:**

For data read from a mini data base, CODER determines the hole name, station name, and sensor type from the input filename, FNM1.

Parameters:

Most of the parameters are returned via common blocks.

IFO = 1, identifies as surface ground from GMGET mini-data base
= 4, identifies as tunnel motion from GMGET mini-data base

Common Blocks:

COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR

Dependencies:

None.

SUBROUTINE DC_C(R1,AVE,DC,RMS,LAST,NE,N0,NPTS)**Purpose:**

DC_C computes and implements the dc correction to the data set. The correction is determined so that the velocity is zero at the end of the correction interval. This dc correction differs from the average acceleration in that the first and last acceleration data points in the interval are weighted as 1/2 rather than 1. The average acceleration and the root-mean-square standard deviation are also computed over the same interval. Differences between the dc correction and the average are minimal.

The interval chosen for determining the dc correction is the interval following the last EMP-detected point (LAST) and the first seismic arrival, estimated as N0.

Parameters:

R1(I)	- input/returned dc corrected acceleration data
AVE	- data average from LAST+1 to N0
DC	- dc correction
RMS	- rms of data from LAST+1 to N0
LAST	- last detected EMP data point (0, if no EMP detected)

N0 - index for first seismic arrival
NPTS - number of data points

Common Blocks:

None.

Dependencies:

None.

**SUBROUTINE DV4(TIME,R3,EV1,EV2,IT1,IT2,WT,NC,TILT,RMS,
NE,N0,N1,N2,NPTS)**

Purpose:

DV_4 estimates the error bars for the repaired data. All acceleration errors are assumed to be uncorrelated. However, the velocity and displacement errors will be correlated because

of the changes being made to the data. Because of this, the velocity and displacement errors can become quite large by the end of the data set.

Parameters:

TIME(I)	- time array
R3(I)	- input acceleration data
EV1(I)	- error bar for repaired velocity
EV2(I)	- error bar for repaired displacement
IT1(I)	- index for starting point of i th clipping event
IT2(I)	- index for ending point of i th clipping event
WT(I)	- weight of i th clipping event
NC	- number of clipping events
TILT	- tilt correction
RMS	- rms standard deviation in acceleration from DC_C
NE	- EMP test limit
N0	- first seismic arrival
N1	- start of averaging window
N2	- end of averaging window
NPTS	- number of data points

Common Blocks:

None.

Dependencies:

CLIP_2,STEP1

SUBROUTINE EMP_C(RAW,R1,NE,N0,NPTS,T_EMP,LAST)

Purpose:

EMP_C determines if there is an EMP spike at the beginning of the data (between index 0 and NE) and implements the correction if necessary. If any EMP spike is detected, the datum between 0 and the determined end of the EMP is set to zero.

Two tests are made to detect any EMP. The first is a simple threshold that checks for values below $-g/2$ or above $g/2$. Data exceeding these values are assumed to be EMP generated.

The other test is based on the statistics of the data. If the beginning of the data is more noisy than later times (though still before the first seismic arrival at N0), the noise is assumed to be EMP.

Should either of these tests fail to detect a known EMP spike, the user can force the algorithm to implement an EMP correction. This is accomplished by entering the estimated EMP duration as a negative number. The code knows to interpret the negative time as forcing.

Parameters:

RAW(I)	- input acceleration data
R1(I)	- EMP corrected acceleration data
NE	- interval at which to test data for EMP
N0	- estimated first seismic arrival
NPTS	- number of data points
T_EMP	- estimated early time duration of EMP
LAST	- first non-EMP data point, if EMP spike is detected

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE FILT(R1)

Purpose:

FILT filters the EMP-repaired tunnel-motion data prior to applying remaining corrections. The data are filtered because the large low-frequency content of the tunnel-motion data severely impacted the determinations of the gauge clipping and tilt corrections. The values selected were too dependent upon the averaging interval selected. The data were filtered to eliminate these frequencies.

Parameters:

R1(I) - input/filtered acceleration data

Common Blocks:

CHARACTER*80 FILE13

CHARACTER HOLE*10,STATION*12,SENSOR*10,FNM1*80,FILE*80

COMMON/BLCK1/RAW(108000),R2(108000),R3(108000),TIME(108000),
NPTS

COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR

Dependencies:

FOUR1, NDFT

**SUBROUTINE FIX_VEL(R3,TIME,VI,WT,IT1,IT2,NC,TILT,FIX,VRC,ADC,
 NC1,N1,N2,NPTS)**

Purpose:

FIX_VEL determines the gauge clipping and gauge tilt corrections simultaneously based on a least squares fit to the repaired velocity or displacement data. The corrections are applied in an iterative fashion, with new corrections being computed after each iteration. Accumulative gauge clipping and tilt corrections, VRC and ADC, are updated at the end of each iteration, as are the relative weights for the clipping events.

For all cases other than tunnel motion, the final displacement is estimated as a function of averaging interval at the beginning of each iteration. The interval that minimizes the final displacement is selected. The algorithm was designed to accept a predetermined value of displacement, FIX, to converge towards. This was never implemented, but the structure for this option remains.

Parameters:

R3(I)	- dc, EMP and spike corrected acceleration data
TIME(I)	- time array
VI(I)	- velocity work array
WT(I)	- weight of i th clipping event
IT1(I)	- index for starting point of i th clipping event
IT2(I)	- index for ending point of i th clipping event
NC	- number of clipping events
TILT	- time of gauge tilt
FIX	- not used
VRC	- total gauge clipping correction: DOUBLE PRECISION
ADC	- total gauge tilt correction: DOUBLE PRECISION
N1	- start of averaging window

N2 - end of averaging window
NPTS - number of data points

Common Blocks:

None.

Dependencies:

ARBNX, CLIP_2, CLIP_T, SQD_L, SQD_Q, STEP1, TILT_C, WT_NEW, WT_N11

SUBROUTINE FOUR1(DATA,NN,ISIGN)

Purpose:

FOUR1 is a standard discrete Fourier transform from Numerical Recipes. It computes forward or inverse Fourier transform and is used in ADARE to high-pass filter the tunnel-motion data.

Parameters:

DATA - input complex array, real part is acceleration data
NN - number of data points
ISIGN = 1 identifies forward transform
 = -1 identifies inverse transform

Dependencies:

None.

SUBROUTINE NDFT(C1,DT,ID,N)

Purpose:

NDFT computes a properly normalized Fourier transform of the input complex data C1. The Numerical Recipes algorithm, FOUR1, is used to compute the discrete Fourier transform (DFT). For proper normalization, the DFT returned by FOUR1 is multiplied by the sample interval DT for forward transforms and by 1/N DT for inverse transforms. Normalization factors should be reinvestigated if other DFT algorithms are substituted.

Parameters:

C1(I) = input/Fourier transformed array
DT = sample interval
ID = 0 identifies inverse transform
 = not zero, identifies forward transform
N = number of data points

Dependencies:
FOURI

SUBROUTINE READ_ASCII(XYZ,DATA,NPTS,GAUGE,IFO)

Purpose:

READ_ASCII reads ground motion from an ASCII file. The format must be with the data set name on first line. The number of data points, NPTS, and sample interval, DT, are entered on second line. The data values are entered from the third line to the end of the data set.

Parameters:

XYZ	- identifies as vertical or horizontal motion
DATA(I)	- time array
RAW(I)	- raw data
NPTS	- returned number of data points
GAUGE	- returned gauge value form GEODES
IFO	- identifies data set as ASCII data set when = 2

Dependencies:

CHARACTER*80 FILE,FNM1,FILE12,FILE13
CHARACTER HOLE*10,STATION*12,SENSOR*10
COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR

SUBROUTINE READ_PARAM

Purpose:

READ_PARAM reads the repair parameters that are required to perform the data repair procedure. The parameters are returned to the main program via common block.

Parameters:

R3(I)	- dc, EMP, and spike-corrected acceleration data
TIME(I)	- time array
VI(I)	- velocity work array
WT(I)	- weight of i th clipping event
IT1(I)	- index for starting point of i th clipping event
IT2(I)	- index for ending point of i th clipping event
NC	- number of clipping events
TILT	- time of gauge tilt
FIX	- not used
VRC	- total gauge clipping correction: DOUBLE PRECISION
ADC	- total gauge tilt correction: DOUBLE PRECISION

N1 - start of averaging window
 N2 - end of averaging window
 NPTS - number of data points

Common Blocks:

```
COMMON/PROC_PARAM/T_EMP,T0,T1,T2,S_LO,S_HIGH,PEG,
1  TSPIKE,TAVE,T_TILT,T_WIDTH
COMMON/PROC_NEW/T0_CLIP
COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR

CHARACTER*80 PARAMFILE,FNM1,FILE
CHARACTER HOLE*10,STATION*12,SENSOR*10
```

Dependencies:

None.

SUBROUTINE READ_SUB(XYZ,DATA,RAW,NPTS,GAUGE,IFO)

Purpose:

READ_SUB was originally designed to read a GEODES ground-motion binary output file created by GMGET (written by Fred App). It has been modified to read ASCII data files and SAC data files.

Parameters:

XYZ - identifies as vertical or horizontal motion
 DATA - time array
 RAW(I) - raw data array
 NPTS - returned number of data points
 GAUGE - returned gauge value form GMGET mini-data base
 IFO = 1, identifies as surface motion from GMET mini-data base
 = 4, identifies as tunnel motion from GMGET mini-data base

Common Blocks:

```
PARAMETER (LREC=256,LSTRAT=72,LSTATS=73)
REAL*4 H_NNEV,H_ENEV,H_ELEV,H_DIAM,H_TD,H_WP,H_AVRAD,
      H_SWL,H_ATCD,
1  H_TPCD,S_NNEV,S_ENEV,S_ELEV,S_SEISFA,TSTART,DELTAT,M_FSA,
2  M_PCTFSC,DEPTH(LSTRAT),VBUFF(LREC)
INTEGER*4 H_AREA,MCNT,NSTRAT,NVALS,NSTA
INTEGER*2 CONSOL/8/,DATA9/9/
INTEGER*2 IPNT(LSTATS)
CHARACTER H_COMMENT*240,S_TITLE*30,S_COMMENT*240,M_FILTER*20,
M_TITLE*30,M_DSYS*50,M_TRANS*50,M_COMMENT*60,M_DDATE*9
```

```

CHARACTER*12 HOLNAM,STATNS(LSTATS),STATION
CHARACTER*10 H_WPMED,H_FLTP,S_STALOC,S_STAMED,S_DATAACQ,MTYPE,
HOLE,SENSOR,STRAT(LSTRAT)
CHARACTER*80 FILE,FNM1
BYTE HOLHD(316),STAHD(320),MEAHD(249), LABEL(3,80)
EQUIVALENCE (HOLNAM,HOLHD),(MCNT,STAHD),(MTYPE,MEAHD)
COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR
COMMON/HOLCOM/HOLNAM,H_NNEV,H_ENEV,H_ELEV,H_DIAM,H_TD,
      H_WP,H_AVRAD,H_SWL,H_ATCD,H_TPCD,H_AREA,H_COMMENT,
H_WPMED,H_FLTP
COMMON /STACOM/ MCNT,S_NNEV,S_ENEV,S_ELEV,S_SEISFA,S_TITLE,
S_COMMENT,S_STALOC,S_STAMED,S_DATAACQ
COMMON /MEACOM/MTYPE,TSTART,DELTAT,NVALS,M_FSA,M_PCTFSC,
      M_FILTER,M_TITLE,M_DSYS,M_TRANS,M_COMMENT,M_DDATE

```

Dependencies:

None.

SUBROUTINE REPAIR(XYZ,GAUGE,MODE,IFO,NF13,FILE13)

Purpose:

REPAIR controls the execution of the repair process. It makes the calls that implement the EMP, dc, noise spike, gauge clipping, and gauge tilt corrections. It also writes the repair parameter file out to disk.

The repairs are implemented in the following order. First, the EMP correction is performed with call to EMP_C. If the data are tunnel-motion data, then the data are high-pass filtered with call to FILT. The dc correction is then performed by DC_C. The noise spike repair is performed next.

How the noise spike repair is done next depends on the parameter MODE. In most cases, MODE is passed as 0 (normal run). In this case, the code makes its own determination as to which data are noise spike data and which data are not.

If MODE is 3, then the code proceeds to identify the possible noise spikes and write this information out to file. The user may then edit the file to specify which data are noise spikes and remove real data from the list. When the program is rerun with MODE = 4, the code will automatically use the information in the intermediate noise spike file to identify noise spikes. Repair then proceeded as if MODE = 0.

Parameters:

XYZ	- flag set if surface ground motion and if vertical
GAUGE	- gauge factor for accelerometer

MODE - parameter that identifies MODE of repair
 IFO = 1, identifies as surface ground from GMGET data base
 = 4, identifies as tunnel motion from GMGET data base
 NF13 - number of characters in filename
 FILE13 - filename for the noise spike identifier file

Common Blocks:

```

    CHARACTER HOLE*10,STATION*12,SENSOR*10,FNM1*80,FILE*80
    COMMON/BLCK1/RAW(108000),R2(108000),R3(108000),TIME(108000),
           NPTS
    COMMON/BLCK2/VI(108000),DI(108000)
    COMMON/BLCKN/N0,N1,N2,NA,NE,NS,NT,NW
    COMMON/BLCKE/EV1(108000),EV2(108000)
    COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR
    COMMON/PROC_PARAM/T_EMP,T0,T1,T2,S_LOW,S_HIGH,PEG,
    TSPIKE,TAVE,T_TILT,T_WIDTH
    COMMON/PROC_NEW/T0_CLIP
    COMMON/B1993/IWT_1,IWT_2,IWT_3,IWT_4
    *
```

Dependencies:

DC_C, DV4, EMP_C, CLIP_1, CLIP_11, FIX_VEL, SP_C, SPT_C1, SPT_C2, SPT_C3, STEP1, TILT_1, TILT_2

SUBROUTINE RERUN06

Purpose:

RERUN06 regenerates the repaired data from the repair parameter files. It is not used to regenerate repaired tunnel-motion data.

RERUN06 regenerates the repaired data in the same sequence in which they were originally repaired. That is, the EMP, dc, and noise spike removal corrections are applied first. Since these repairs are simple to implement once they have been determined, they are done within the RERUN06 routine. The gauge clipping and tilt repairs are next executed with the calls to CLIP_2 and TILT_C. The error bars are then estimated in the call to DV4.

Parameters:

None. Information is passed via common blocks.

Common Blocks:

```

    CHARACTER*12 STATION
    CHARACTER*10 HOLE,SENSOR
    CHARACTER*80 FNM1,FILE
```

```

DOUBLE PRECISION VRC,ADC
COMMON/NAME/FNM1,NF1,FILE,HOLE,STATION,SENSOR
COMMON/BLCK1/RAW(108000),R2(108000),R3(108000),TIME(108000),
      NPTS
COMMON/BLCK2/VI(108000),DI(108000)
COMMON/BLCKE/EV1(108000),EV2(108000)
COMMON/BLCKN/N0,N1,N2,NA,NE,NS,NT,NW
COMMON/B1993/IWT_1,IWT_2,IWT_3,IWT_4

```

Dependencies:

CLIP_2, CODER, DV4, FILT, READ_SUB, STEP1, TILT_C, STEP1

SUBROUTINE SIFTER(X1,Y1,FACT,X2,Y2,N1,N2,N0)

Purpose:

SIFTER resamples the input X1 and Y1 data arrays at a lower sample frequency (lower by factor of N0) and stores results in output X2 and Y2 data arrays. The data are renormalized by a factor of FACT as they are resampled. It is used primarily for plotting so that not all data points are plotted.

Parameters:

X1(I)	- input X array
Y1(I)	- input Y array
FACT	- renormalization factor
X2	- returned X array
Y2	- returned Y array
N1	- original number of points
N2	- returned number of points in thinned arrays
N0	- resampling factor (every N0 th point is sampled)

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE SPT_C1(R1,Z,S_LOW,S_HIGH,NPTS)

Purpose:

SPT_C1 identifies possible noise spikes in input acceleration data by comparing data to lower and upper threshold values. If the data exceed the ranges of these limits (i.e., fall

below or more negative than lower threshold or rise above upper threshold), data are identified as a possible spike.

Parameters:

R1(I)	- input acceleration data
Z(I)	- parameter array, -1 if $R1(I) \leq S_LOW$ or $\geq S_HIGH$
S_LOW	- lower spike threshold
S_HIGH	- negative spike threshold
NPTS	- number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE SPT_C2(Z,NA,NS,NPTS)

Purpose: SPT_C2 searches through the possible spike data identified by SPT_C1. SPT_C2 then constructs contiguous noise spike data intervals so that they are at least NS data points in length with a minimum of NA data points between each possible interval of noise spike data.

Parameters:

Z(I)	- parameter array returned from SPT_C.
NA	- averaging window to correct spikes
NS	- minimum width of spike zones
NPTS	- total number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE SPT_C3(R1,Z,Z1,XYZ,N1,NS,NPTS)

Purpose:

SPT_C3 tests large acceleration data points to determine if the data are real data or noise spike data. The test is made by assuming that noise spikes are characterized by rapid oscillations between large positive and large negative values. Thus, the average over a

sample will be small due to the cancellations. However, the average magnitude will be large. The ratio of the average to the magnitude average should be smaller still.

Parameters:

R1(I) - input acceleration data
Z(I) - parameter array
Z1(I) - parameter array
XYZ - SENSOR CONTAINS 'Z'
NS - averaging width
NPTS - number of points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE SP_C(R1,R2,Z,SP1,SP2,ISI,ISF,N_SPIKE,NA,NPTS)

Purpose:

SP_C repairs data that are identified as noise spike data. The repair is implemented as a simple linear interpolation across the noise spike data. The linear interpolation is determined from the average of data values preceding and following the noise spike. The averages are computed over the last NA data points just preceding the noise spike and the first NA points just following the noise spike.

Parameters:

R1(I) - input acceleration data
R2(I) - returned spike repaired data
Z(I) - spike identification array
= 0 if not a spike, = -1 otherwise
SP1(J) - value of first repaired data point of jth noise spike
SP2(J) - change in repair per data point of jth noise spike
ISI(J) - index for starting point of jth noise spike
ISF(J) - index for ending point of jth noise spike
N_SPIKE - number of noise spikes
NA - number of points to include in averaging window
NPTS - number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE SQD_L(TIME,VI,FIX,TW,TILT,A0,A1,N1,N2,NPTS,IER)**Purpose:**

SQD_L performs a linear least squares fit to the velocity data to determine the gauge clipping and gauge tilt corrections. The gauge clipping correction is determined as the intercept, while the gauge tilt correction is determined as the slope.

Parameters:

TIME(*)	- time array
VI(*)	- velocity array
FIX	- not used
TW	- weighted clipping time
TILT	- time that gauge tilt occurs
A0	- clipping correction determined by least squares fit
A1	- tilt correction determined by least squares fit
N1	- start of averaging window
N2	- end of averaging window
NPTS	- total number of data points

Common Blocks:

None.

SUBROUTINE SQD_Q(TIME,DI,FIX,TW,TILT,A0,A1,N1,N2,NPTS,IER)**Purpose:**

SQD_Q performs a quadratic least squares fit to the displacement data to determine the gauge clipping and gauge tilt corrections. The gauge clipping correction is determined as the slope (first-order coefficient), while the gauge tilt correction is determined from the second-order coefficient.

Parameters:

TIME(*)	- time array
DI(*)	- displacement array
FIX	- not used
TW	- weighted clipping time
TILT	- time that gauge tilt occurs
A0	- clipping correction determined by least squares fit
A1	- tilt correction determined by least squares fit
N1	- start of averaging window

N2 - end of averaging window
NPTS - total number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE SQD_XY(TIME,VI,DI,FIX,TW,TILT,N1,N2,M1,NPTS)

Purpose:

SQD_XY estimated the final displacement for surface ground motion data as a function of averaging interval used in the least squares fitting procedure.

Parameters:

TIME(*) - time array
VI(*) - velocity array
DI(*) - displacement array
FIX - not used
TW - effective clipping time
TILT - gauge tilt correction
N1 - start of averaging window
N2 - end of averaging window
M1 - start of averaging interval that minimizes displacement
NPTS - number of data points

Common Blocks:

None.

SUBROUTINE SQD_Z(TIME,VI,DI,FIX,TW,TILT,N1,N2,M1,NPTS)

Purpose:

SQD_Z estimates the final displacement for surface ground motion data as a function of averaging interval used in the least squares fitting procedure.

Parameters:

TIME(*) - time array
VI(*) - velocity array
DI(*) - displacement array
FIX - not used
TW - effective clipping time

TILT	- gauge tilt correction
N1	- start of averaging window
N2	- end of averaging window
M1	- returned start of averaging window
NPTS	- number of data points

Common Blocks:

COMMON/PROC_PARAM/T_EMP,T0,T1,T2,S_LOW,S_HIGH,PEG,
TSPIKE,TAVE,T_TILT,T_WIDTH

Dependencies:

None.

SUBROUTINE STEP1(X,Y,Z,NPTS)

Purpose:

STEP1 performs a simple integration of Y as a function of X and returns Z. Integration is by trapezoidal rule.

Parameters:

X(I)	- independent variable
Y(I)	- input dependent variable
Z(I)	- returned integration of Y
NPTS	- number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE TILT_1(R3,S,NT,NPTS)

Purpose:

TILT_1 calculates data value sums over fixed window sizes in preparation for call to TILT_2.

Parameters:

R3(I)	- input noise spike repaired acceleration data
S(I)	- data value sums
NT	- window size for data value sums
NPTS	number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE TILT_2(S,IL1,IL2,IT0_CLIP,NT,NW,N1)**Purpose:**

TILT_2 scans the data value sums computed by TILT_1 to determine the next maximum data value sum. The next maximum must be separated from the previous maximum by at least NW data points.

Parameters:

S(I) - data value sums.
IL1 - last maximum
IL2 - new maximum
NT - averaging window
NW - minimum separation
NPTS - number of data points

SUBROUTINE TILT_C(R3, DA, I_TILT, NPTS)**Purpose:**

TILT_C performs the gauge tilt correction by subtracting the gauge tilt correction, DA, from the acceleration from all points after I_TILT.

Parameters:

R3(I) - input spike corrected acceleration data
DA - gauge tilt correction
I_TILT - index for starting the tilt correction
NPTS - number of data points

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE WT_N11(R3,WT,IT1,IT2,NC)

Purpose:

WT_N11 updates the new relative clipping weights. The update is required because of modifications to the data, and the clipping weights are determined from the data.

Parameters:

R3(*) - input spike repaired acceleration data
WT(I) - weight for i^{th} clipping event
IT1(I) - index for starting point of the i^{th} clipping event
IT2(I) - index for ending point of the i^{th} clipping event
NC - number of clipping events

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE WT_NEW(R3,WT,IT1,IT2,NC)

Purpose:

WT_NEW updates the new relative clipping weights. The update is required because of modifications to the data and the clipping weights are determined from the data.

Parameters:

R3(*) - input spike repaired acceleration data
WT(I) - weight for i^{th} clipping event
IT1(I) - index for starting point of the i^{th} clipping event
IT2(I) - index for ending point of the i^{th} clipping event
NC - number of clipping events

Common Blocks:

None.

Dependencies:

None.

SUBROUTINE YL(Y,Y1,Y2,NL,N)

Purpose:

YL searches through the N data points of the three data sets (Y, Y1, and Y2) to be plotted. The purpose of the search is to terminate data sets Y1 and Y2 when their absolute values exceed the largest absolute value of Y. The purpose is to ensure that the display does not compromise dynamic range of Y.

Parameters:

Y(i) - repaired data
Y1(i) - raw data
Y2(i) - error bars
NL - point at which to terminate Y1 and Y2
N - total number of data points

Common Blocks:

None.

Dependencies:

None.

4. Examples

Examples of the repair procedure are now presented. The examples begin with easy problems and progress towards more difficult problems. Most repair falls into the easy category.

A. First Example of Easy Repair

An easy repair example is presented first. The example is U6E GM1 GMA ZA. The raw and repaired data are plotted in Figure 5a, and the repair parameter file used in the repair is listed in Figure 5b. The raw data are plotted in the upper-left hand corner of Figure 5a. A quick examination of the data shows that they are relatively clean (i.e., no EMP and no noise spikes). Easy repair is anticipated. In addition, there are only a few strong peaks. The repaired data are plotted in the upper-right hand corner of Figure 5a. No changes to the data are perceivable.

Figure 5a. First Example of Easy Repair

```

AUTO-7
U6E GM1 GMA ZA
1 134 -6.866E-05 -8.545E-05 4.138E-03
0
7 175 1530 4482 4.715E-04 -4.498E-04
159 191 1.725E-01
401 433 1.341E-01
535 567 4.144E-01
626 658 2.025E-01
459 491 1.376E-02
318 350 1.386E-02
593 625 4.888E-02
5.874E-05 5.358E-05 1.753E-04
5.00000E-02 2.10000E-01 2.00000E+00 8.00000E+00
-2.00000E+00 2.50000E+01 1.00000E+01
2.50000E-02 2.50000E-02 5.00000E-02 5.00000E-02
0.00000E+00

```

Figure 5b. Repair Parameter File for First Example of Easy Repair

The repaired velocity data are plotted in the lower-left hand corner as the solid curve. The partially repaired velocity data are plotted as the dotted curve, which is not visible since it overlaps with the repaired velocity. This indicates that gauge tilt and gauge tilt repairs are very small. The dashed curve at late times is the estimated error bars, which are very small.

The repaired displacement data are plotted in the lower right-hand corner as the solid curve. The dotted curve is the partially repaired displacement data. Some differences are now discernible. The primary difference is that the partially repaired displacement exhibits a slight ramp at late times. This is evidence of very slight gauge clipping. Again the dashed curve is the error bars.

The first line of the repair parameter file in Figure 5b shows that the data were repaired with the version 7 code. The second line documented that the U6E GM1 GMA ZA data set were repaired. The number 1 beginning the third line shows that no EMP spikes were found. The second number, 134, is the index for the first seismic arrival. The interval between 1 and 134 was used to determine the dc correction, -6.866E-05, which is very small. The implied precision is unrealistic. The average and rms over the interval are -8.545E-05 and 4.138E-03 respectively.

The zero entered on the fourth line records that no noise spikes were found. The next line shows that seven clipping events were identified. From the same line, the entries 175, 1530, and 4482 declare that gauge tilt was determined to occur at index 175 and that the LSAI was 1530 to 4482. The gauge clipping and gauge tilt repairs were 4.715E-04 and -4.498E-04 respectively.

The remaining information to implement the gauge clipping repair are entered on the next seven lines. These lines identify the start, end, and relative size of each clipping event.

Remember that the input that produced this repair is entered at the end of the data. This input begins with the last four lines. For example, the early time estimate of the EMP is 0.05 s.

B. Second Example of Easy Repair

Another example of easy repair is presented. The data are plotted in Figure 6a, and the repair parameter file is shown in Figure 6b. Again, the repair is expected to be easy. These data were read from an ASCII file and not a GMGET data base. The plots of the raw and repaired data in the upper portions of the figure show little or no difference and are not discussed. The interest turns immediately to the repaired velocity data in the lower-left hand corner. The solid curve is the repaired velocity, and the dotted curve is the partially repaired velocity data. The major difference between the two is that the ramp displayed by the partially repaired velocity data is at late times. This is evidence of gauge tilt.

Figure 6a: Second Example of Easy Repair

```

AUTO-7
NTS NPE SF IPR
1 1000 -4.94957E-05 -5.01407E-05 1.20635E-03
0
15 1663 5585 8798 -6.15753E-04 9.51904E-04
1317 1334 -1.09419E-03
1329 1344 -5.68968E-04
1341 1466 -7.26727E-03
1465 1494 -8.44639E-02
1492 1829 -8.62829E-02
1828 1949 -4.33182E-01
1948 2071 -8.10170E-01
2070 2160 -3.85014E-01
2160 2186 -2.26798E-02
2186 2273 -7.48856E-02
2272 2324 -2.26393E-01
2323 2352 -4.27545E-02
2351 2397 -9.68573E-02
2406 2447 -4.19712E-02
2446 2500 -1.38324E-01
3.93907E-04 7.65760E-04 3.79253E-02
5.00000E-02 4.00000E+00 1.00000E+01 4.00000E+01
-2.00000E+00 2.00000E+00 -2.50000E-02
2.50000E-02 2.50000E-02 5.00000E-02 5.00000E-02
0.00000E+00

```

Figure 6b. Repair Parameter File for Second Example of Easy Repair

The repaired displacement is shown in the lower-right hand corner. The final displacement is on the order of 5 cm or less. The partially repaired displacement (dotted curve) runs off the plot around 12 s.

The primary interest in the repair parameter file is the input, particularly the estimated gauge value, -0.025 g. The negative entry forces the code to accept the absolute value of 0.025 g as the gauge value. The small value is used so that the gauge clipping repair is spread out over many peaks. This tends to be a good strategy when there are no dominant peaks.

C. Example of Tunnel-motion Repair

An example of tunnel-motion repair is presented next. The data set is U12N24 TM1 GMA ZA and was part of a GMGET mini-data base. The results are plotted in Figure 7a, and the repair parameter file is listed in Figure 7b. The raw data plotted in the upper-left hand corner shows an EMP spike at early time. If the spike cannot be clearly resolved from the vertical axis, the fact that the scale is much too large for the visible data indicates that EMP is present. The plot of the repaired data in the upper-right hand corner shows that the EMP has been removed.

Figure 7a. Example of Tunnel Motion Repair

```

AUTO- 8
U12N24 TM1 GMA ZA
 34   268   1.045E-03   8.891E-04   1.645E-02
 0
 4   410  53570  81922  -7.984E-04   8.889E-04
299   342   6.338E-01
317   366   1.803E-02
390   435   1.783E+00
517   560   1.306E+00
-1.760E-05  -1.782E-04   1.487E-03
2.50000E-02  1.00000E-01  2.00000E+01  4.00000E+01
-2.00000E+01  2.00000E+01 -2.00000E+00
2.50000E-02  2.50000E-02  1.50000E-02  1.50000E-02
0.00000E+00

```

Figure 7b. Repair Parameter File for Tunnel Motion Repair

The plot of the repaired and partially repaired velocity in the lower-left hand corner shows that there was slight gauge tilt. The final displacement plotted in the lower-right hand corner is very small.

The repair parameter file in Figure 7b shows that the EMP was determined to last until sample 33 (34-1). No noise spikes were found, and four clipping events were identified using an estimated gauge of 2 g.

D. Example of Intermediate Noise Spike Repair

Return now to the data plotted in Figure 1. Assume that the code was not able to discriminate between the noise spikes and slap down spike. In such cases, the MODE 3 option is used to generate an intermediate noise spike repair file. The file is shown in Figure 8a. There should be seven spikes; the six true spikes and the slap down spike. However, there are only six. What has happened is that because the slap down spike is now determined to be a noise spike, it has blended into the spike just preceding it. This is because the code must set up the averaging windows for the spike repair.

```

2105  2263  1.052  1.131  -1.000
3257  3519  1.628  1.759  -1.000
3874  4079  1.937  2.040  -1.000
4444  4712  2.222  2.356  -1.000
4823  5191  2.412  2.596  -1.000
7875  8066  3.938  4.034  -1.000
-1  13998  3.938  4.034  -1.000
U4J GM1 GMA ZA 08.013

```

Figure 8a. First Example of Intermediate Noise Spike Repair File
The file erroneously identifies only six possible spikes.

The problem is elevated by decreasing the averaging intervals for noise spike identification and repair from 0.025 to 0.01, and the intermediate noise spike repair file is regenerated. The file is edited. The edited file is shown in Figure 8b. The editing removes the sixth noise spike (suspected to be the slap down spike) by changing the SPIKE parameter from -1 to 0. The line could have been deleted, but changing the SPIKE parameter to 0 is preferred. The wrong "spike" might be removed.

2135	2203	1.067	1.101	-1.000
3287	3456	1.643	1.728	-1.000
3904	4031	1.952	2.016	-1.000
4474	4550	2.237	2.275	-1.000
4584	4663	2.292	2.332	-1.000
4853	4927	2.427	2.464	-1.000
5016	5139	2.508	2.570	0.000
7905	7990	3.953	3.996	-1.000
-1	13998	3.953	3.996	-1.000
U4J GM1 GMA ZA 08.013				

Figure 8b. Second Example of Intermediate Noise Spike Repair File.

This file was generated by rerunning the repair procedure with a value of 0.01 s for the minimum width for noise spike duration. The intermediate noise spike repair file was then edited to remove the slap down spike (spike at index i=5016 to 5139) from the list. This can be done by deleting the line or changing the -1.000 to 0.000.

The code is next run in MODE 4, continuation repair. The results are plotted in Figure 8c. The repair is nearly identical to that in Figure 1. The repair parameter file is shown in Figure 8d.

Figure 8c. Repair Generated with Edited Intermediate Noise Spike Repair File

```

AUTO- 8
U4J GM1 GMA ZA
  175   620  3.835E-02  3.836E-02  1.334E-01
    6
  2125  2230 -1.015E+00 -7.510E-04
  3277  3488 -1.054E+00  2.538E-04
  3894  4063 -1.030E+00  8.866E-05
  4464  4679 -1.029E+00  1.485E-04
  4843  4948 -8.928E-01  1.744E-03
  7895  8022 -4.049E-01  1.256E-03
    1  5057 10307 13996  4.272E-01  1.282E-01
  4975  5183  2.504E+00
  2.926E-03 -3.955E-02 -2.122E+00
  1.00000E-01 3.10000E-01 4.00000E+00 7.00000E+00
 -2.00000E+00 1.00000E+01 1.50000E+01
  1.00000E-02 1.00000E-02 2.00000E-02 1.00000E-01
  0.00000E+00

```

Figure 8d. Repair Parameter File Generated with Edited Intermediate Noise Spike Repair File.

E. Example of Setting Earliest Time for Clipping Repair

An example of setting the earliest time for clipping repair is now presented. Two results are presented. The first result is with the earliest time for clipping repair parameter defaulted to 0. The second is presented with the parameter set to implement the repair after the first seismic arrival. The first results are plotted in Figure 9a.

Figure 9a. Example of Setting Earliest Time for Clipping Repair
Earliest time for clipping repair is first defaulted to 0.

Figure 9b. Repair with Earliest Time for Clipping Repair Increased to 1.75 s.

The acceleration data plotted in the upper portion of the figure show that the largest peak is indeed the first seismic arrival. The code will attempt to implement clipping and tilt repairs here. The result of the repair becomes evident when the velocity data are examined in the lower-left hand corner. The differences between the repaired data (solid curve) and partially repaired data (dotted curve) show that the clipping and tilt repairs modified the first seismic arrival. The differences are even evident in the plot of the displacement data in the lower-right hand corner.

The code was rerun using the same parameters, but with the earliest clipping time parameter set to 1.75. The results are presented in Figure 9b. A comparison of Figures 9a and 9b fails to show any changes in the acceleration. This is because the changes that are made are small. However, the velocity data in Figure 9b show that the gauge clipping and gauge tilt repairs are implemented at the peak near 1.75 s, well after the first seismic arrival. The effect is to also decrease the final displacement.

F. Example of Setting LSAI

In some cases, "something" happens well after all the interesting data. These events can be noise spikes, data failures, or unexplained changes in the data. One solution to these problems is ending the averaging interval before the problems occur. One such example is shown in Figure 10a. The problem obviously is the structure at the very end of the data set. The problem is solved (from a repair point of view) by terminating the averaging interval just before the structure. The plot shown in Figure 10b shows that the LSAI was terminated at 6.25 s. The structure at the end has been excluded.

Figure 10a. Example of Setting the LSAI

The problem is to prevent the structure on the end of the data set from affecting the least squares fitting procedure.

Figure 10b. Repair Produced by Terminating Least Squares Fitting Procedure at 6.25 s. Termination is forced by entering -6.25 for this parameter. The negative forces the code to accept the absolute value.

5. Summary

This report has documented the final version of the repair procedures of strong ground motion from underground nuclear tests. Section 1 reviewed the history of this effort from March 1988 through September 1994. This effort evolved through three phases during this time, and each phase was briefly described.

The problems that can occur with the recording and digitizing of the data were identified. These included EMP, dc errors, noise spikes, gauge clipping, and gauge tilt. The characteristics of each problem were summarized. The procedure to repair each problem was described. The description included what type of input is required by the user to influence the repair and how the repairs are documented so that the repaired data can be easily regenerated.

The input and output data sets were described. Four different repair modes were identified and summarized. The various routines used in the repair procedures were separated according to their function or type of repair. The routines were then listed and discussed in more detail. The discussion included their relationships with the others.

Examples of the repair procedures were then presented. The examples included the easy and also more difficult problems such as using the intermediate noise spike repair file.

Finally, the author wishes to acknowledge the effort of Albert Martinez, who executed most of the data repairs.

References

1. Craig Hairston, "SANDIA, A CDC-6600 and PDP-11 Computer Program," EG&G 1183-5102 (1983).
2. Thomas W. Tunnell, "User's Guide for the AUTO06 Repair Code," EG&G Technical Memorandum, LAO-2732-1872, EG&G Energy Measurements, Los Alamos Operations, Los Alamos, New Mexico (1989).
3. Chris A. Ciarcia, unpublished work.
4. Frederick N. App and Thomas W. Tunnell, "User Manual for the NTS Ground-Motion Data Base Retrieval Program ntsgm," Los Alamos National Laboratory, LAUR94-1538 (1994).

Distribution

EG&G/EM Los Alamos Operations

K. Jones

J. Manning

D. Martinson

T. Rupp

T. Tunnell (5)

Los Alamos National Laboratory

F. App, EES-3, F659 (20)

OSTI

Technical Information Officer

US DOE Nevada Operations