GPRSLICE

Version: 1.03.26.01
Last revision date: March 26, 2001
Description: GPRSLICE generates a volume of transformed ground penetrating radar (GPR) data from a series of GPR profiles. The program GPR_PROC should be used before calling GPRSLICE if the data need to be filtered or otherwise processed. GPRSLICE can remove the "background" (the average trace) from each profile and it can expand the dynamic range of the volume data to the maximum allowed by the data storage type (96 decibels).

The purpose of generating a volume of GPR data is to extract information from the radar survey that is not as easily seen in the individual profiles. One way to do this is to enhance features that are correlated or somehow related to each other. This process is commonly called "stacking" in geophysical data processing. Stacking is achieved by summing all the data of interest and dividing by the number of data values. For time series, such as GPR traces, the same sample number from each series is summed and the result divided by the number of series. This creates a new ("average") time series (or GPR trace in this case) in which correlated features are enhanced and random ones are reduced.

The formation of a volume of transformed GPR data is somewhat more complex in that traces are not preserved. The procedure used by GPRSLICE is as follows. Each GPR file is read in individually. A background trace can be removed at this time. The GPR traces that fall within a two-dimensional (the X-Y plane) search window around a volume cell are stacked together. The amplitudes of this stacked trace are then transformed (converted to absolute values or instantaneous amplitudes for example). All sample values that fall within a layer (Z direction) of the volume are averaged and the resulting value is summed with the current value for the volume cell for that layer. After all files have been read, the final volume cell value is calculated by averaging all values assigned to it.

Here is the method in outline form:

```
For each GPR file
   Determine format
   Allocate storage, get MRK and XYZ files, assign spatial coordinates
   Get the GPR data
   Allocate temporary storage
   Remove background if requested
   Place data in the search bins
      for every column in volume (X)
         for every row in volume (Y)
            for every trace in GPR profile
               if within X-Y search bin then add for stacking
            stack traces in the X-Y search bin
            transform amplitudes if requested
            wrap with envelope if requested
            for every layer in volume (Z)
               for every sample in stacked trace
                  if within Z search bin then accumulate sum into volume cell
   Free temporary storage
```

```
Calculate Volume cell values
   for each column (X)
      for each row (Y)
         for each layer (Z)
            divide sum of values by number of values
            find max and min

Expand dynamic range if requested
   for every volume cell: subtract min and multiply by [65535/(max-min)]
```

The volume consists of 16-bit, unsigned integers (range 0 to 65535). The volume can be sliced perpendicular to one of the three coordinate axes. The most common slices would be horizontal layers perpendicular to the Z-axis. Slices from the volume are then written to disk in any, or all, of three formats described in the Output Files section or as a single binary file for direct input into the Fortner T3D program.

The input to GPRSLICE.EXE is a "CMD" file, an ASCII text file containing keywords (or commands) which are discussed below.

The GPR data can be read from disk using the following formats only:
- GSSI SIR-10A "DZT" files,
  NOTE: DZT files should have only ONE channel. Multiple-channel files WILL NOT be sliced correctly!
- Sensors and Software pulseEKKO "DT1" and "HD" files, or
- Society of Exploration Geophysicists SEG-Y files that follow the Sensors and Software style for formatting the reel (file) header.

A message file called GPRSLICE.LOG is opened when the program starts. It is located either in the directory where the program was called from or in the root directory of drive C. This open file may prevent more than one session of the program from executing in the same directory if using multiple DOS windows in MS Windows. The log file may contain more information regarding the failure or success of GPRSLICE as it executes. Sessions are appended at the end of the log file.

GENERATING THE VOLUME

To generate the volume you specify start and stop locations and the number of sections in all three dimensions, X, Y, and Z. These define the volume cells, or compartments, that the data will be assigned to. Sections along the X-axis are called columns. Along the Y-axis, sections are called rows. Layers are sections along the Z-axis. If, for example, you want to define 3 layers between 0 and 30 ns, Z_first would be 0, Z_last would be 30, and Z_layers would be 3. Each layer would be 10 ns "thick" (0-10, 10-20, 20-30 ns). Cell dimensions in the X and Y directions are defined similarly.

The location of the middle of the cell is used for its coordinate location (such as in the text file used for input to Golden Software's SURFER™). For example, if 2 columns are defined for the X direction between X_first = 0 and X_last = 2, then each cell is 1 meter wide and the cell locations are recorded as 0.5 and 1.5 meters.

There must be at least one slice (one section in the axis direction that the slice is perpendicular to) and two sections in each of the other two directions. For example, for horizontal slices (perpendicular to the

Z-axis) there must be one section in the Z direction and at least two sections in each of the X and Y directions. This program generates planes of data, not points or lines.

A search box, which is centered on the middle of each volume cell, is used to determine which data will be assigned to a volume cell. By default, the search box is the same size and location as the cell. The user has the option to change the size of the search box, either smaller or larger, to accommodate dense or sparse data sets respectively. The search box size essentially modifies the size of the volume cell to make them overlap or to separate them by space/time. For example, for a horizontal slice between 10 and 20 ns, the default search box size is 10 ns wide in the Z direction and the box is centered on the middle of the layer at 15 ns. GPR trace values between 10 and 20 ns are assigned to the volume cell. If the search box size in the Z direction is increased to 20 ns, then GPR trace values between 5 and 25 ns are used (15 plus and minus 10 ns). This feature is particularly useful when the length of the volume cells along the X or Y directions are less than the distance between GPR profiles. Likewise, the search box size can be restricted to include only one profile or to include adjacent profiles when the cell size is close to or larger than the profile spacing.

The location of the GPR data can have any orientation with respect to the volume. Parallel profiles are not required. Storage formats can be different. The time window, samples per trace, sample rate, and antenna frequency can vary from file to file. Other programs such as GPR_REV, GPR_STAK, GPR_CNDS, GPR_JOIN, GPR_XFRM, and GPR_SAMP can be used but are not required to provide the corrected positioning of traces or to change the number of samples or traces. GPR data outside of the search boxes are ignored.

X, Y, and Z values for each trace in a profile are determined from the (required) “MRK” and "XYZ" files. MRK files contain the location of “marked” traces in each file, that is the traces for which the spatial locations are known. XYZ files contain the spatial locations that correspond to each marked trace. The Z values are not used as the sample “locations” are travel-time offsets (in ns) from the start_time (see this keyword below).

Each GPR file is read in sequence, and the amplitudes of the trace samples are assigned to search boxes. The traces within each search box are stacked. The stacked trace values are converted to absolute values, squared values, instantaneous amplitude, or instantaneous power. In addition, an envelope can be draped across the positive peaks of the transformed trace. Empty search boxes are set to zero or a median value (32768), depending on the transformation. Please note that empty search boxes can be avoided by enlarging the cell size and/or the search box size. The fastest way to generate slices will be using absolute values. Using instantaneous amplitude or power (calculated from the analytic signal determined from the Hilbert transform of the trace) will take the longest time to generate slices.

OUTPUT FILES

The volume can be written to disk as separate files for each slice or as a single file representing the entire volume. An information file is written, with the “INF” filename extension, which summarizes the manipulations of GPRSLICE and the output filenames and locations.

Each slice from the volume is written to disk as a separate file with numerically increasing filenames. The user supplies a template (up to six characters long) and the program appends numbers (01, 02, 03, ..., 11, 12, etc.). There are two storage formats to choose from. Either or both can be chosen. The user can also select the output directory. The third option is to write the entire volume out as a binary file for input to the Fortner program T3D™.

"TXT" files are text files that list the two spatial coordinates and the amplitude value for each station in the slice. In text files only, the amplitudes of the volume are reduced by a factor of 8 to range from 0 to 8191 ($2^{13} - 1$). Text files can be used in spreadsheet programs or as input data for Golden Software's program SURFER™. SURFER™ re-grids individual slices and displays them separately.

"PCX" files are graphics files containing the amplitudes stored in the compressed PCX format. The PCX graphics files can be input into any graphics application (word processors, draw programs such as Corel Draw™, and paint programs such as Corel PhotoPaint™).

The volume slice in the two output file types above is organized depending on the direction the slice is taken and the storage format. The first slice is written to the first file. Slices are written "upside down" for PCX files, which have the origin at the upper left of the CRT screen. Layers correspond to increments in the Z direction; rows to increments in the Y direction; and columns to increments in the X direction. For horizontal slices (perpendicular to the Z-axis) information is written row by row starting near the origin. For vertical slices information is written layer by layer. Each slice has a constant value for its distance from the coordinate origin that increments uniformly along the axis that the slice is perpendicular to. For example, vertical slices perpendicular to the Y-axis will have a constant Y-value for all the information in a file.

"T3D" files are designed to be input data for the visualization program T3D™. The data type is unsigned byte, with a range of 0 to 255. The number of layers (Z), rows (Y), and columns (X) can be determined from the INF file that was written along with the T3D file. The data are written to the T3D file so that they have the proper X-Y-Z orientation when read in by T3D™.
Values are written as follows:

```
  for (layer=0; layer<num_layers; layer++)
    for (row=num_rows-1; row>=0; row--)
      for (col=num_cols-1; col>=0; col--)
```

"SLD" files are designed to be input data for the visualization program SlicerDicer™. The data type is unsigned 2-byte, with a range of 0 to 65535. The number of layers (Z), rows (Y), and columns (X) can be determined from the INF file that was written along with the SLD file. The data are written to the SLD file so that they have the proper X-Y-Z orientation when read in by SlicerDicer™.
Values are written as follows:

```
  for (layer=num_layers-1; layer>=0; layer--)
    for (row=0; row<num_rows; row++)
      for (col=0; col<num_cols; col++)
```

THE KEYWORDS

Following is the list of keywords and their default values. The documentation format is:
"KEYWORD: **keyword** = default value".
Look at GPRSLICE.CMD as an example command file with correct usage and default keyword values. The file GPRSLICE.CMD has most comments stripped out, and GPRSLICE.CM_ has all comments removed.

********** PROGRAM CONTROL **********

KEYWORD: **batch** = "FALSE"
Place the program in batch mode (no pauses) if "TRUE". The program will normally pause at times before ending.

KEYWORD: **debug** = "FALSE"
Place the program in debug mode if "TRUE" (for developers)

********** SPECIFICATION OF INPUT DATA **********
One or more data files must be read. The data storage format is determined by inspecting the file. If the program cannot recognize a flavor of the three formats below then an error message will be issued. All GPR data are converted to 16-bit unsigned data (range 0 to 65535) for internal storage.

Recognized storage formats are:
DZT - GSSI SIR-10A file with embedded (512- or 1024-byte) info header
DT1 - Sensors & Software pulseEKKO file with a matching HD info file
SGY - SEG SEG-Y format

DT1 and HD files are assumed paired, i.e. both have the same filename with different extensions. So, if a data file with a ".DT1" extension is specified, the ".HD" filename will be assumed. Only DT1/HD files must have those filename extensions.

A MRK file is required that specifies the marked trace numbers. Each MRK file must have the same filename as the data file but with the extension ".MRK".

Example MRK file (comments are ignored and not required):
```
3               ; number of marked traces
104
256
897
```

An XYZ file is required that specifies the X-, Y-, and Z-locations for every trace in the MRK file. Each XYZ file must have the same filename as the data file but with the extension ".XYZ".

Example XYZ file (comments are ignored and not required):
```
3               ; number of records
10.0  10.0  293.456   ; XYZ for trace 104
20.0  10.0  294.567   ; XYZ for trace 256
30.0  10.0  295.678   ; XYZ for trace 897
```

########## NOTE ##########
A cubic spline is used to interpolate between XYZ values for each marked trace, except when there are only 2 marked traces (and linear interpolation is used), presumably (HINT) at the beginning and end of evenly spaced data on a flat surface.
##############################

KEYWORD: **num_input_files** = 0
Replace 0 with the number of actual input files; this parameter MUST be entered before "input_filelist[]".

KEYWORD: **input_filelist[]**
Add an equal sign after this keyword and then the GPR filenames separated by a space. Names can continue onto the next line. For example:
num_input_files = 10
input_filelist[] = file1.dzt file2.dzt file3.dzt file4.dzt file5.dzt
file6.dzt file7.dzt file8.dzt file9.dzt file10.dzt
file10.dzt

############ NOTE #############
The complete pathname must be given if the program is not executed from the directory where the input files are stored, or if files are in several directories. Consider the following examples where "data_dir" is the name of the directory that the data files are stored in, "gprslice_in" is the directory where the program GPRSLICE.EXE is stored, "gprslice_called_from" is the current directory that GPRSLICE.EXE is invoked from, and "input_filelist" are examples of how much of the path that must be specified after input_filelist[] above. These examples assume that c:\exe is in the DOS path.

```
 data_dir data_dir gprslice_in gprslice_called_from input_filelist
|        |        |           |                   |                    |
 c:\gpr1           c:\gpr1     c:\gpr1             file1.dzt, etc.
 c:\gpr1           c:\exe      c:\gpr1             file1.dzt, etc.
 c:\gpr1           c:\exe      c:\                 c:\gpr1\file1.dzt,
                                                   c:\gpr1\file2.dzt,
 c:\gpr1  c:\gpr2  c:\gpr1     c:\gpr1             c:\gpr1\file1.dzt, etc.
                                                   c:\gpr2\file1.dzt, etc.
 c:\gpr1  c:\gpr2  c:\exe      c:\gpr1             c:\gpr1\file1.dzt, etc.
                                                   c:\gpr2\file1.dzt, etc.
 c:\gpr1  c:\gpr2  c:\exe      c:\                 c:\gpr1\file1.dzt, etc.
                                                   c:\gpr2\file1.dzt, etc.
```

################################

********** SPECIFYING OUTPUT FILES **********
The volume generated by this program consists of unsigned 16-bit integers (2-bytes) that range in value from 0 to 65535. Vertical slices are perpendicular to the X- or Y-axis. Horizontal slices are perpendicular to the Z-axis. Each slice is stored in its own file in one or all of the following formats. At least one storage format must be selected.

TEXT (*.txt): an ASCII text file where each line contains the two coordinates plus the amplitude for each station in the slice. Amplitudes for this type of file only are reduced by a factor of 8 (range will be o to 8191).

PCX (*.pcx): a graphics file containing the amplitudes for each slice.

The file names, one for each layer, each have the same first characters (up to 6) in the filename followed by a number for each layer. The extension depends on the storage format. For example, the filenames for 12 slices stored in PCX format might look like:
slice01.pcx, slice02.pcx, etc. up to slice12.pcx.

T3D (*.t3d): a binary file designed to be easily read in by the T3D program.

"SLD" (*.sld) files are designed to be input data for the visualization program SlicerDicer™.

KEYWORD: **out_directory** = ""
This keyword defines the pathname to the directory where all slice files will be stored. The default is the current directory that this program is run from. A different directory (and path) can be defined by placing the name of the path and directory between the quote marks. If an invalid directory is defined then the directory will default to the current directory.

############ NOTE #############
THIS IS A DOS-STYLE DIRECTORY. NO SPACES IN THE PATH! EACH DIRECTORY NAME IS LIMITED TO 8 CHARACTERS.
###############################

KEYWORD: **inf_outfilename** = ""
This keyword defines name of the text file that describes the slice files. If the string is left empty, then the name will default to the first input GPR file name. The filename extension will be forced to "INF".

KEYWORD: **txt_outfilename** = ""
This keyword defines the template for the series of text files. Only the first six characters are used. Extension will be forced to "TXT".

KEYWORD: **pcx_outfilename** = ""
This keyword defines the template for the series of graphics files. Only the first six characters are used. Extension will be forced to "PCX".

KEYWORD: **t3d_outfilename** = ""
This keyword defines the filename for the binary file for input to the T3D program. Only the first eight characters are used. Extension will be forced to "T3D".

KEYWORD: **sld_outfilename** = ""
This keyword defines the filename for the binary file for input to the SlicerDicer program. Only the first eight characters are used. Extension will be forced to "SLD".

KEYWORD: **overwrite_protect** = "TRUE"
Existing files on disk are protected from being overwritten unless this keyword is set to "FALSE"

**************************** DEFINING THE VOLUME ****************************
The volume consists of a collection of three-dimensional cells. The cell locations are determined from the keywords in this section. Each slice through the volume consists of all the cells that reside on a single plane.

X and Y units are in distance (assumed to be meters). Z units are in time (assumed to be nanoseconds). Z units increase “down” toward the last layer in the volume.

############ NOTE #############
In ALL cases, "..._last" must be greater than "..._first", as these define two opposite sides of the volume. The X_columns, Y_rows, and Z_layers keywords must be greater than zero. At least two must be equal to 2 or more. Only one can be equal to one to create a single slice in the slice direction chosen.
###############################

In the text files (used for input to SURFER), the location of the middle of the cell is used for its coordinate location. For example, if 2 columns are defined for the X direction between X_first = 0 and X_last = 2, then the cell locations are recorded at 0.5 and 1.5 meters.

KEYWORD: **slice_direction** = "Z"
This keyword defines which axis the slices are perpendicular to. Either a numeric value or a string can be assigned to the keyword. Choices are:
0 or “X” = X axis; vertical slices parallel to the Y-Z plane
1 or “Y” = Y axis; vertical slices parallel to X-Z plane
2 or “Z” = Z axis (default); horizontal slices parallel to the X-Y plane
NOTE: only one slice direction may be chosen at this time.

KEYWORD: **X_first** = 0.0
This keyword defines where the left side of the volume is in the X direction (in meters).

KEYWORD: **X_last** = 0.0
This keyword defines where the right side of the volume is in the X direction (in meters).

KEYWORD: **X_columns** = 0
This keyword defines the number of columns (sections) in the X direction. It must be a whole number greater than zero.

KEYWORD: **Y_first** = 0.0
This keyword defines where the left side of the volume is in the Y direction (in meters).

KEYWORD: **Y_last** = 0.0
This keyword defines where the right side of the volume is in the Y direction (in meters).

KEYWORD: **Y_rows** = 0
This keyword defines the number of rows (sections) in the Y direction. It must be a whole number greater than zero.

KEYWORD: **Z_first** = 0.0
This keyword defines where the top of the volume is in the Z direction (in ns).

KEYWORD: **Z_last** = 0.0
This keyword defines where the bottom of the volume is in the Z direction (in ns).

KEYWORD: **Z_layers** = 0
This keyword defines the number of layers (sections) in the Z direction. It must be a whole number greater than zero.

KEYWORD: **start_time** = 0.0
This keyword allows you to define the offset from time zero to the first sample time, in nanoseconds. The default is 0.0, that is time zero is at the first sample. The limit is plus/minus the total time for the GPR file. Negative times indicate that time zero occurs after the first sample. Positive times indicate that time zero occurs before the first sample. If you only know the sample number for time zero, you'll have to calculate the offset to the first sample. For example, if there are 0.5 ns between samples and time zero is at sample 15, then start_time would be -7.5;

NOTE: This start time adjustment is applied the same to all files that are read in, regardless of each file’s total time.

********** DEFINING THE SEARCH BOX **********
The size of the three-dimensional search box defaults to the same size as the volume cell. You have the option of changing the box size to a number greater or less than the cell size. The cell size can be determined from the first and last sides of the volume along an axis and the number of sections along that direction.

For example, with X_first = 0, X_last = 10, and X_columns = 10, each cell is 1 meter long in the X direction. The value for box_Xsize will default to 1 meter. The search distance will be 0.5 meter on either side of the center of the cell in the X direction. If a larger search distance is required (because for instance lines are spaced 2.5 meters in the X direction), then the search distance can be increased to 1.3 (box_Xsize would be 2.6) to be sure every cell intersects at least one radar line.

Only values greater than or equal to zero are accepted.

KEYWORD: **box_Xsize** = 0.0
X-direction width of search box (column) in meters.

KEYWORD: **box_Ysize** = 0.0
Y-direction width of search box (row) in meters.

KEYWORD: **box_Zsize** = 0.0
Z-direction width of search box (layer) in ns.

********** MANIPULATING THE AMPLITUDES **********
These keywords determine how the trace amplitudes are manipulated or transformed and how they are assigned to the volume stations. A GPR record is a sequential collect of GPR traces. A trace consists of a certain number of samples, say 512. For each sample a number is associate with it that is the amplitude of the receiving-antenna response over a short period of time to an electromagnetic field.

To assign values to the volume cells, the values of all the samples that fall within the search box are averaged together. The sample values can remain as recorded and read by this program, or they can be transformed according to the options below. In addition, with any transform method selected, the sample values can be replaced by the interpolated values of a line that connects the positive peaks of the transformed trace.

KEYWORD: **xfrm_method** = "ABS"
This keyword determines the method of transforming the sample amplitudes. After transforming the values are normalized to fit within the range 0 to 65535. Either a numeric value or a string can be assigned to the keyword. Choices are:
0 or "NONE" = use values as read in from the GPR data file.
1 or "ABS" = use the absolute values of the amplitudes (default).
2 or "SQR" = use the square of the amplitudes.
3 or "INST" = use the instantaneous amplitudes calculated using the Hilbert transform
4 or "POW" = use the instantaneous power calculated using the Hilbert transform of the input trace and the analytic signal.

Options "ABS" and "INST" produce similar results. Options "SQR" and "POW" produce similar results.

For options 3 and 4, an analytic function is constructed using the original trace as the real component and its Hilbert transform as the imaginary component.

The modulus of the complex function (the square root of the sum of the squares of the real and imaginary components) is called the instantaneous amplitude of the function (option 3 here). For GPR data it measures the reflectivity strength, reducing the appearance of random signal in the data.

The square of the modulus of the complex function can be called the instantaneous power (option 4 here). For GPR data it measures the total energy of the GPR signal at an instant in time. The effect on the appearance of the data is similar to converting to instantaneous amplitude, but noise is reduced even further.

############ NOTE #############
Empty cells are set to a value of 0 for options 1 through 4 and to 32768 for option 0.
################################

KEYWORD: **envelope** = "FALSE"
If this keyword is set to “TRUE”, then a cubic spline will be used to interpolate a curved line that connects the positive high amplitudes from any of the xfrm_method choices. Because a cubic spline is used, overshoots may occur and consequent clipping. Empty cells will be set to a value of zero.

KEYWORD: **expand** = "FALSE"
If set to "TRUE", this keyword increases the dynamic range of the data after the volume has been calculated. For example, if after taking the absolute values of the traces and the volume station amplitudes range from 56 to 32000, then 56 will be subtracted from all values, and the values multiplied by 65535/31944 to increase the maximum value to 65535 (the maximum 16-bit unsigned integer). Amplitudes stored in TXT files only will then be divided by 8.

KEYWORD: **multiply** = ""
This keyword increases or decreases the amplitude of the data after the volume has been calculated and after the dynamic range has been expanded (if requested). Only values greater than 0.0 are accepted. A value of 1 has no effect.

KEYWORD: **background** = "FALSE"
If set to "TRUE" then a "background trace" (the average of all traces in a profile) is removed as soon is the file is read in. The stacking process used to calculate the background trace enhances coherent signal and reduces randomly varying signal (noise). In this case, the coherent signal is the horizontal banding often seen in GPR data (system noise) and the randomly varying signal is the received radar signal from the subsurface. The appearance of the data is often improved by removing the horizontal banding. Caution must be used, however, with small data sets (less than a few hundred traces) or data that has strong natural horizontal reflectors.

Usage: `GPRSLICE cmd_filename`
Required command line arguments:
cmd_filename - The name of the keyword file.
Optional command line arguments (do not include brackets): none
Examples:

```
gprslice xfile1.cmd
```