CONVERTING HORIZON AND FAULT SURFACES FROM AASCII TEXT FORMAT TO AASPI FORMAT – PROGRAM surface_import

Contents

Overview

Seismic interpreters within a given work environment commonly use multiple interpretation and data analysis software packages including *Seisworks, KingdomSuite, Petrel, VoxelGeo, Paleoscan*, *Geoteric, Seisworks, Transform, IKON*, and others. Any given interpretation package allows the import and export of horizons and faults in multiple formats. Some of the more proprietary formats are written in binary, with little, if any guidance on how to read or write them. The more commonly used formats are written as ASCII (or "text") files, which although larger are relatively easy to read.

The goal of program **surface_import** is to convert the more commonly used ASCI- format horizon and fault files into the AASPI-format. This allows the interpreter to convert a surface file once, rather than in each program that might need it. At the same time, this strategy allows us as algorithm developers to greatly simplify the application programs and their graphical user interfaces (GUIs).

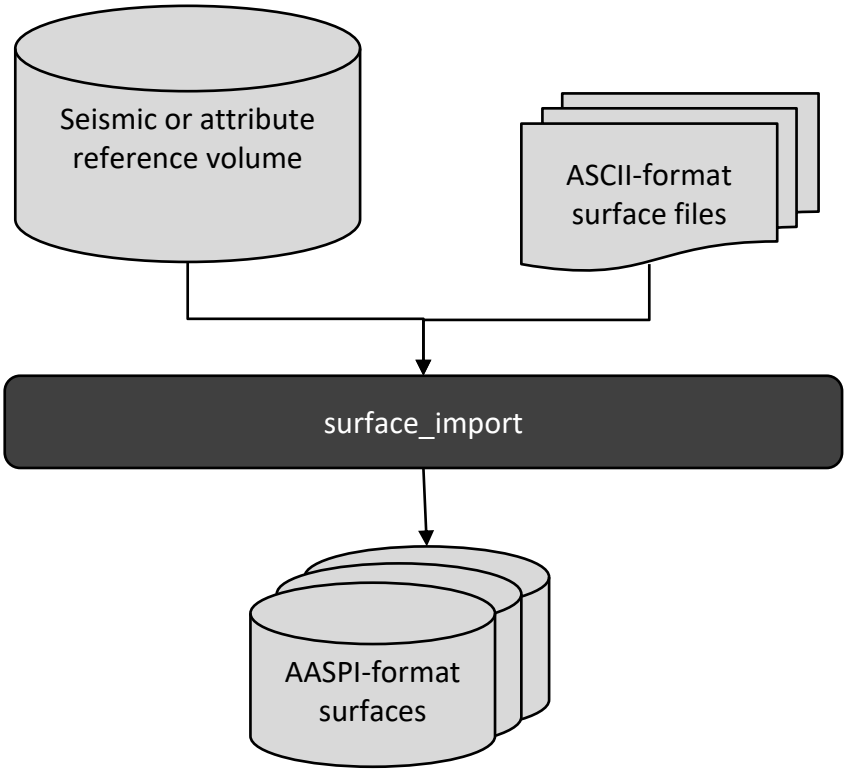
Some commercial software differentiate between a horizon and a surface. For example, in *Petrel*, a horizon consists of the user-defined picks that may be augmented by 2D and 3D autopickers. Such horizons might consist of a grid of picked inlines and crosslines at 10 line and crossline intervals. In contrast, in *Petrel* a "surface" is a computer object that is defined across the entire seismic volume, whether a give trace was picked or not. In several packages, a B-spline defined by a grid of knots is least-squares fit to the picks. In other packages, the picks are interpolated to each valid trace location. In this documentation we will use this definition of a "surface" – a deformed 2D $t(x,y)$ or $z(x,y)$ surface that is defined either at B-spline knots or by picks at each trace in the seismic data volume. Areas where the horizon was not picked, or where the surface does not exist due to a depositional hiatus, onlap, erosion, karstification, or other geologic processes are also defined using the construct of a null value.

Within the AASPI software package, stratigraphic and unconformity surfaces are used to constrain the area of computation for spectral decomposition and for machine learning, and to define the limits for data flattening, stratal slicing, and data registration (e.g., time to depth conversion).

Fault surfaces

Faults can also be represented by surfaces. In our convolutional neural network (CNN) applications fault surfaces will be used as training and validation data "labels." These training data can be *weighted* to provide a level of confidence in the picked faults where some faults are well defined by the seismic data and others in poor data areas are *pushed through* using our understanding of tectonic processes.

Computation flow chart



The flow chart is simple. The input files include an AASPI-format ("*.H") file that represents the extent of the seismic survey as well as one or more ASCII-format ("text") files of the same type exported from an interpretation workstation or geostatistics program. The output consists of a corresponding suite of AASPI-format surface files that are either truncated or extended (using null values) corresponding to the input reference seismic data or attribute volume.

Output file naming convention

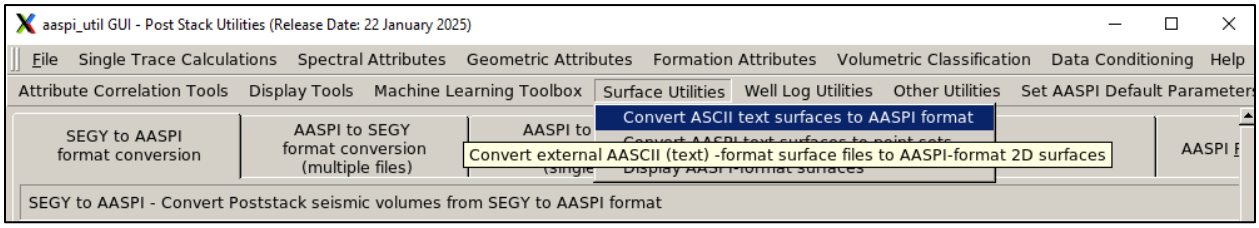
Program **surface_import** will always generate the following output files:

Output file description	File name syntax
Program log information	surface_import_unique_project_name_suffix.log
Program error information	surface_import_unique_project_name_suffix.err
Output surface file name	surface_unique_project_name_suffix/surface_old_input_surface_file_name.H

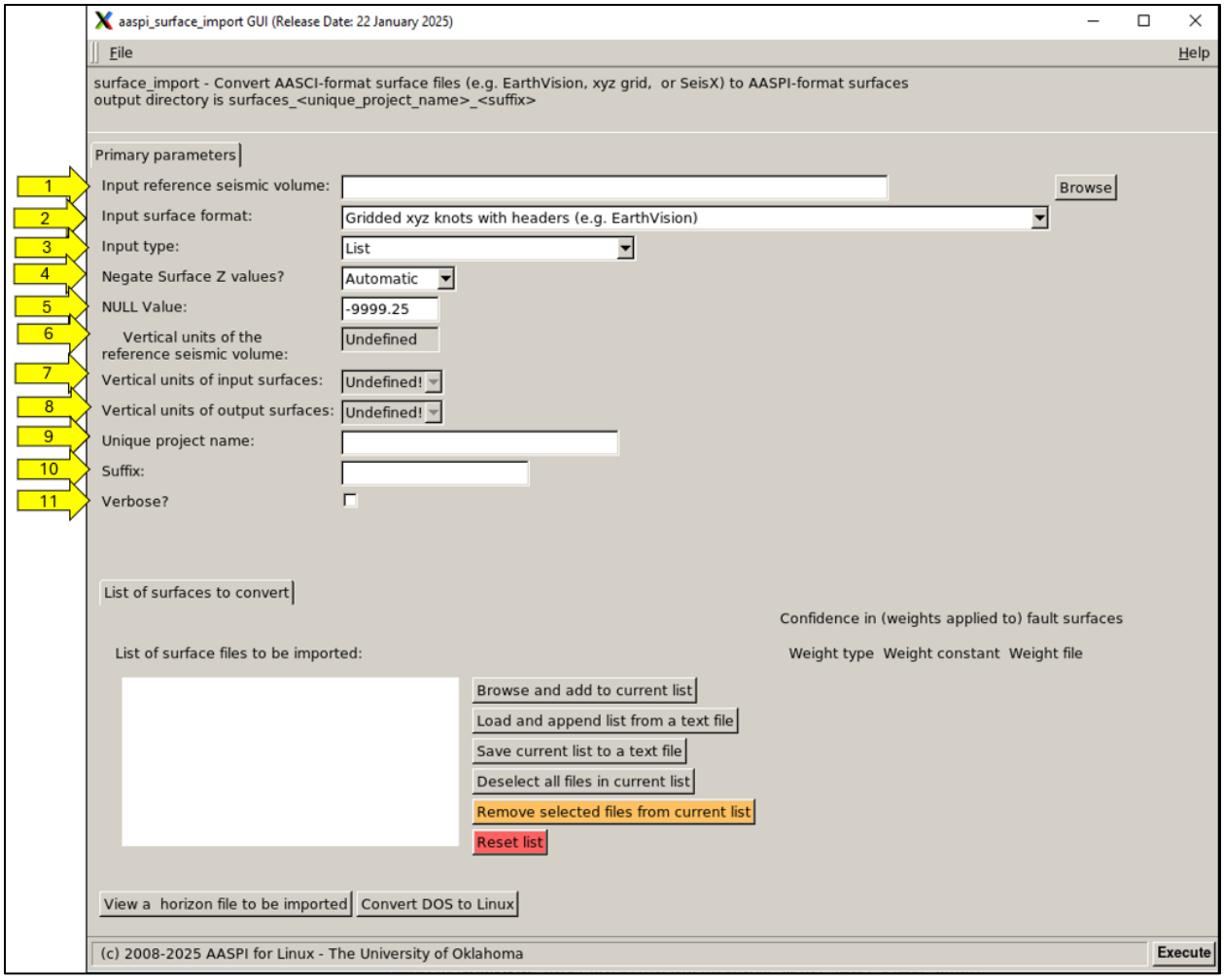
where the values in red are defined by the program GUI and the values in blue are related to the input file names. Errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the **.log* file. More common errors and program completion messages will appear in the **.err* file and appear in a pop-up window upon program completion or termination.

Invoking the surface_import GUI

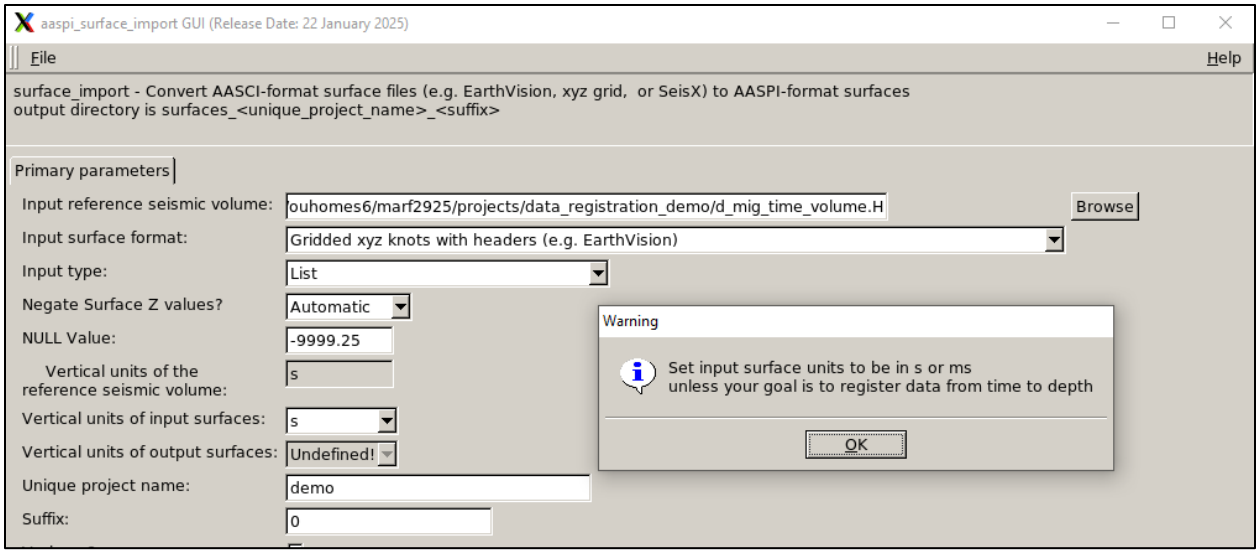
To convert an AASCI-format surface to AASPI-format, click the *Surface Utilities* tab in the **aaspi_util** or **aaspi_main** tab and drop down and select *Import surfaces to AASPI*:



The following GUI appears:



The first step is to *Browse* and select a (1) *Input reference seismic volume* that represents that part of the survey you are using. Surfaces that extend beyond the limits of this data volume will be truncated whereas those that are smaller or have missing pieces will be augmented with *null values* to indicate valid vs. missing fault or horizon picks. At this point, the (6) *Vertical units of the input reference volume*, the (7) *Vertical units of the input surfaces* (in ASCII text format_ surfaces, and the (8) *Vertical units of the output surfaces* in AASPI-format are *Undefined.* Defining these units correctly is the most critical (and error prone) step in the conversion process. For this reason, once the *Input reference seismic volume has been selected,* the following warning appears:



Note that the *Vertical units of the reference seismic volume* is filled in as "s" and cannot be altered. These are the units of your input data volume. In all cases except when converting time to depth or depth to time in AASPI program **surface_driven_data_registration**, these will also be the vertical units of your output AASPI-format surfaces.

Example 1: Importing surfaces with vertical units consistent with the reference seismic volume

The warning message says that you need to enter the units of the input ASCII text format fault or horizon. In this first example, my horizons are in time (either in s or ms) and thus are consistent with the units of the reference seismic volume. Click *OK* and go to the dropdown menu next to (7) *Vertical units of input surfaces*. In this case, the default choices will be determined by the units of the reference seismic data volume and appear as

I know my time horizons are in s so I choose "s." At this point a second warning message appears:

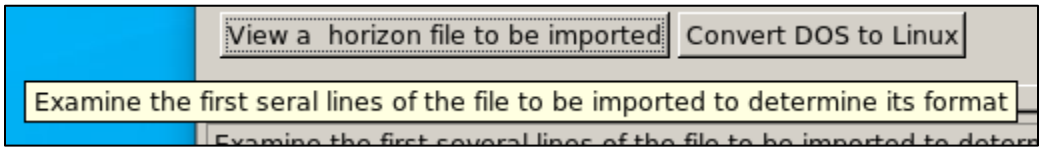
The GUI is smart enough to know that if my input surface is in s or ms, and if the seismic data are in s, the output horizon needs to be in s. I simply click OK. I then go back and change the default *Suffix* read in from the reference seismic volume and type in "in time." In this manner, my output surfaces will go into a directory called *surfaces_demo_in_time*. Later in this exercise I will import some surfaces that have been converted to depth using kriging with external drift using a commercial software package and well control. Those surfaces will be used in **surface_driven_data_registration** to convert the seismic or attribute data from time to depth between the target surfaces.

Returning to the GUI, I note the (4) *Negate surface Z values?* is set to be *Automatic.* Most geoscience and engineering programs define the vertical axis as positive down. However, at least one (*Petrel*) defines the positive vertical axis as positive up. The automatic option always works for time data – the non *null value* picks will be either uniformly positive or negative. Depth horizons take more care on land data where we can see both negative (above sea level) and positive (below sea level) depths. In this case, choose your orientation so that the vertical axis used for your input surface file is in the same direction as your reference seismic data volume.

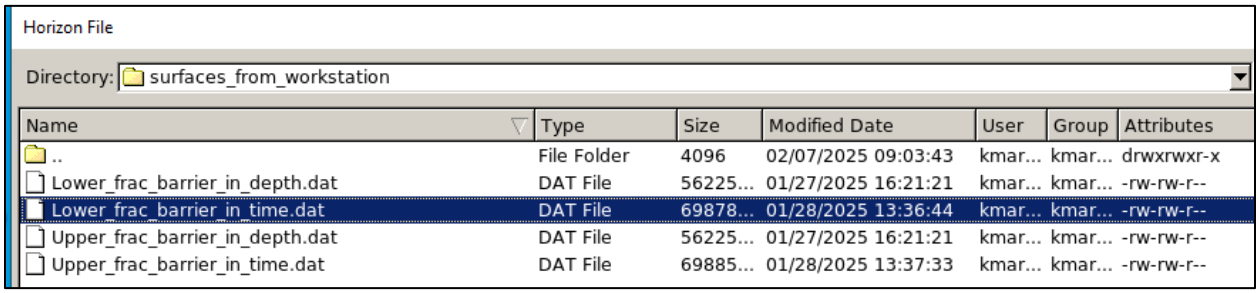
The (5) *null value* defines samples on the surface that are not defined. In many software packages it will be -9999.25, the default in this **surface_import** GUI. If using *EarthVision* format surfaces, the null value will be defined in the headers. In other cases, you need to examine your horizon file to determine what the null value (if any) is for your input surfaces.

Examining an ASCII text format file

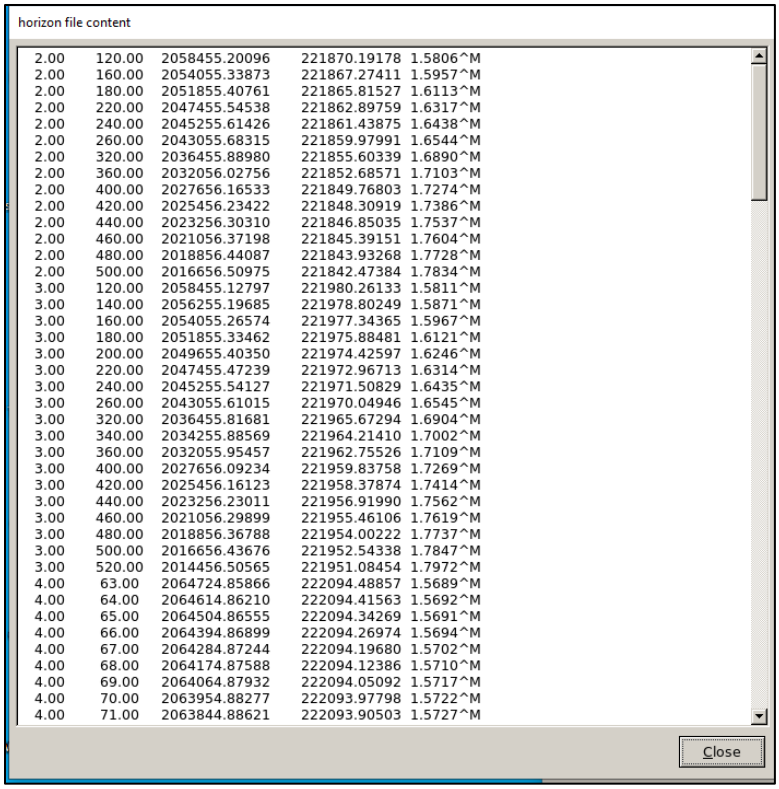
Let's go look at one of the files to be imported. At the bottom of the **surface_import** GUI you will see the following two buttons:



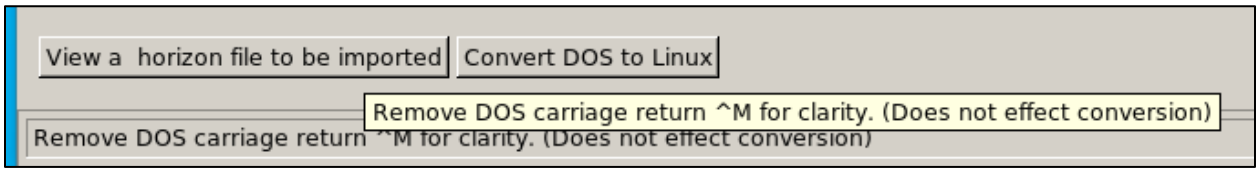
I click *View a horizon file to be imported* and select one



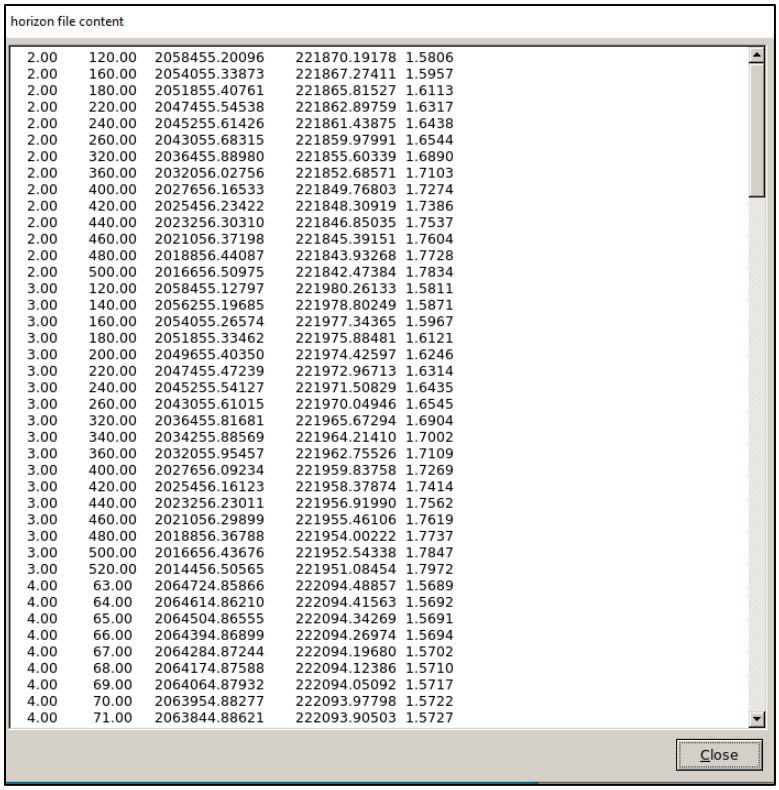
The following window pops up:



I'm writing this documentation on Linux whereas the horizon was picked using *KingdomSuite* which was run under Windows using a *Disk Operating System* or DOS. I see these annoying carriage return symbols *^M* at the end of each line. If you hate them, click *Convert DOS to Linux*:

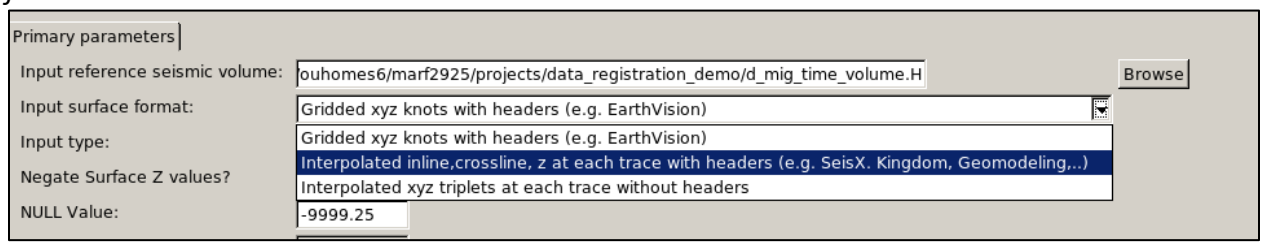


I close my previous window containing the horizon and reopen it by clicking *View a horizon file to be imported* a second time.



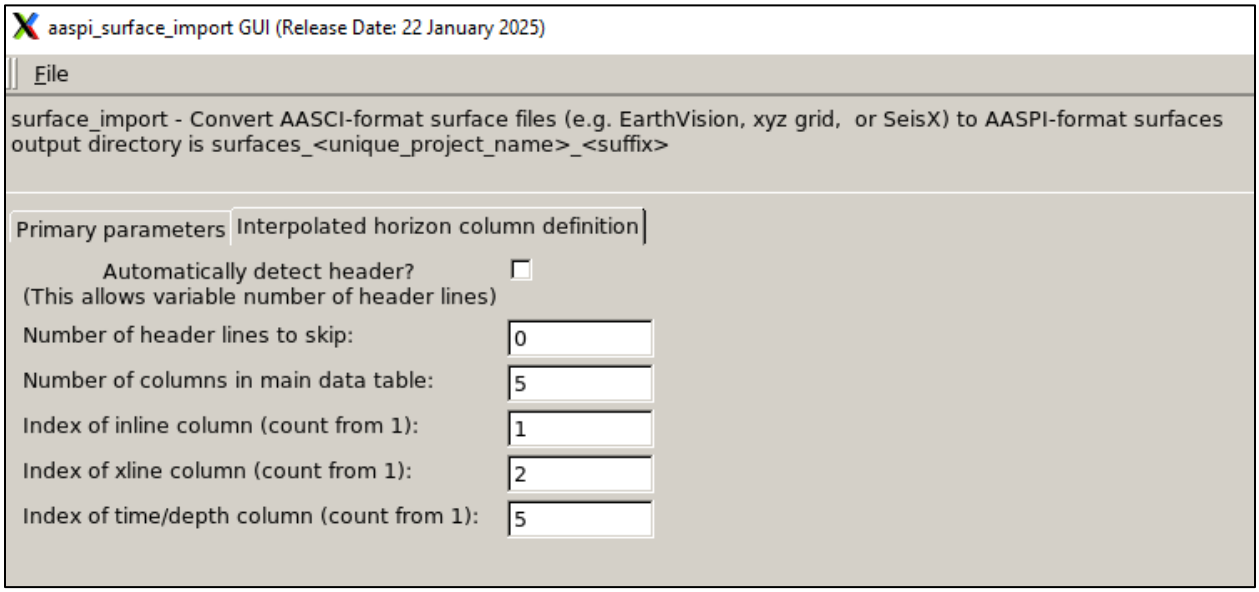
This format is commonly used in interpretation workstations, particularly those that were first built for seismic (versus for well log) interpretation. In this example, I see no headers telling me which column is which and what the units of measurement are. However, familiarity with the data volume indicates that column 1 is the inline number, column 2 is the crossline number, column 3 is the *x* or Northing coordinate, column 4 is the *y* or Easting coordinate, and column 5 is the time of the picked surface which I see is in s (and not in ms).

With this information in mind, I return to the **surface_import** GUI and define the (2) *Input surface format*:



The default format is *EarthVision* which is used by many of the larger interpretation workstations, *GoCad*, most geocellular modeling packages, and was developed and supported by Dynamic Graphics, Inc. with wide use in the geoscience community (including those working in tectonics, earthquake seismology, and volcanology). Interpretation software that originally was developed using only wells (e.g., *Petrel*) or only 2D seismic lines (e.g., Z-map), required the use of knots defining B-splines. These knots are almost always oriented along the North and East axis and usually not aligned with the seismic survey axes. The knot spacing is often coarser than the seismic bin size. These splines will need to be interpolated internal to program **surface_import**.

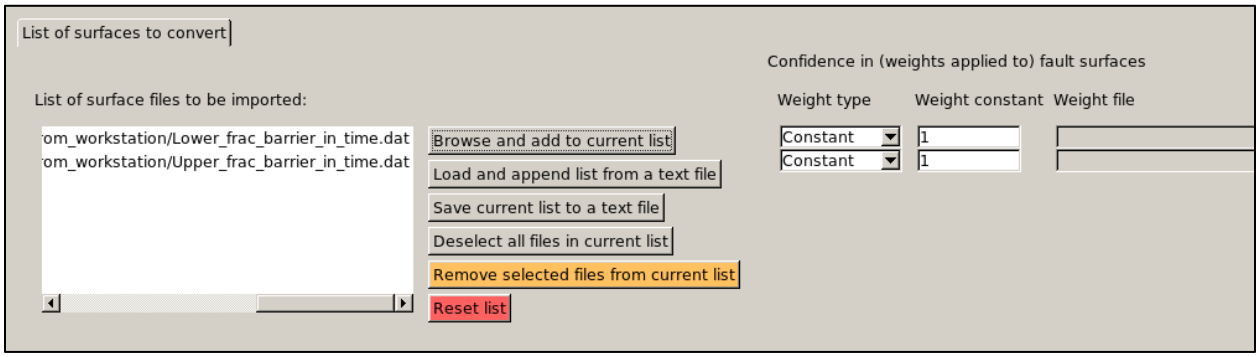
In contrast, surfaces generated from other 3D seismic interpretation software packages will have a time or depth value at each trace. We will refer to these as being *Interpolated inline, crossline, z at each trace*. I select this option, and a new tab appears called *Interpolated horizon column definition* which looks like this:



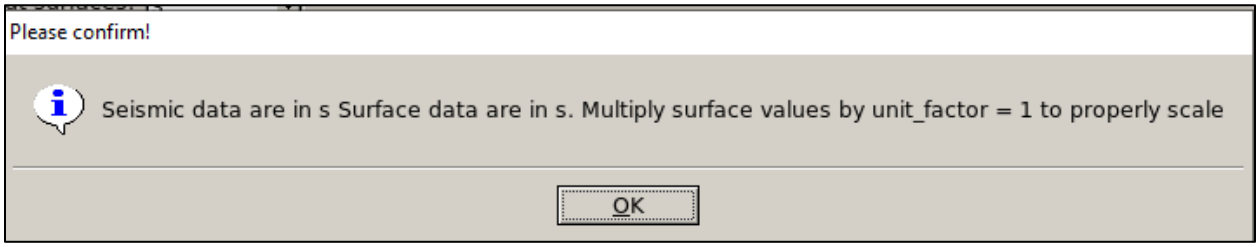
Do *NOT* trust the defaults! Look at your data. Then plot and quality control your results! Yes, I like defaults too but the default of column 3 for the Index of time/depth gave me a map with the Easting (x-value) rather than the time-value for my surface. In this example I counted 5 columns with no headers. I then recognized the time-depth value to be in column 5.

Defining the horizons to be converted to AASPI format

I drop down to my *List of surface files to be imported* and select *Browse and add to current list*. I have four files in my directory – two in time and two in depth. But they have different formats! So I need to do a two-step process. First, I select the two horizons that are in time:



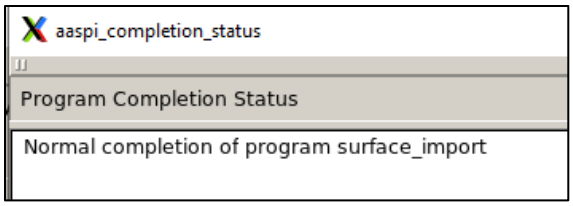
For the current application, the *Confidence in (weights applied to) fault surfaces* can be ignored. This option is only relevant for CNN fault detection. With these two files I am ready to click *Execute*. A warning message comes up:



Yes! My input ASCII text format surfaces are in s and my output surfaces need to be in s to correspond to the reference seismic data volume, so the proper scaling factor is 1.0 , so I click *OK*. Here is a snapshot of some of the information appearing on the terminal, where a message is shown for every 10,000th card in the ASCII file has been read, along with some summary values of the minimum and maximum values of the surface:

```
Processing card 1180000...
Processing card 1190000...
Processing card 1200000...
Processing card 1210000...
Processing card 1220000...
Processing card 1230000...
Processing card 1240000...
Processing card 1250000...
Processing card 1260000...
Processing card 1270000...
Processing card 1280000...
Processing card 1290000...
Processing card 1300000...
Processing card 1310000...
Processing card 1320000...
Processing card 1330000...
Processing card 1340000...
n_valid = 1343949  n_neg = 0
z_max =        1.843  z_min =       1.414
 call scale_unit: unit_factor =    1.000000
 after scaling surface%z_min, surface%z_max =    1.414000       1.843000
 After import: z_max =    1.843000     ; z_min =    1.414000
Writing output file: ./surfaces_demo_in_time/surface_Upper_frac_barrier_in_time.H
 before writing header values: surface%z_min, surface%z_max =    1.414000
   1.843000
 seismic_fn =
 /ouhomes6/marf2925/projects/data_registration_demo/d_mig_time_volume.H
 s_fn = ./surfaces_demo_in_time/surface_Upper_frac_barrier_in_time.H
 surface%n1_seismic,surface%n2_seismic =          589         651
 output j2_seismic,j2_surface:           1          49
 output j2_seismic,j2_surface:           2          50
 output j2_seismic,j2_surface:           3          51
 output j2_seismic,j2_surface:           4          52
 output j2_seismic,j2_surface:           5          53
 output j2_seismic,j2_surface:           6          54
 output j2_seismic,j2_surface:           7          55
 output j2_seismic,j2_surface:           8          56
 output j2_seismic,j2_surface:           9          57
```

After completion, a pop-up window appears:

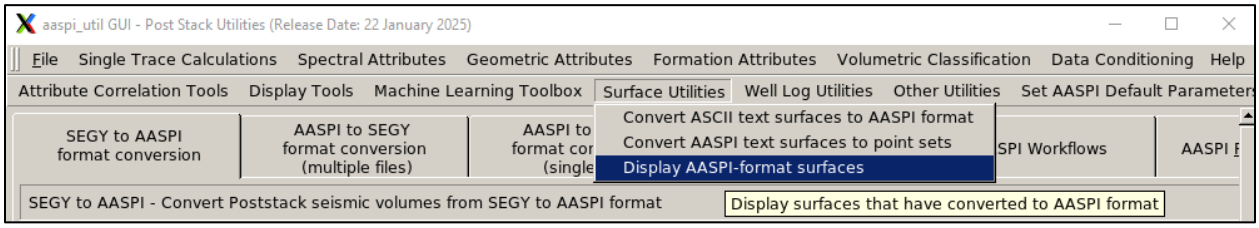


I can list the contents of the directory containing the AASPI-format surfaces. Using the construct in defining the directory described above I list the files in *surfaces_demo_in_time time*:

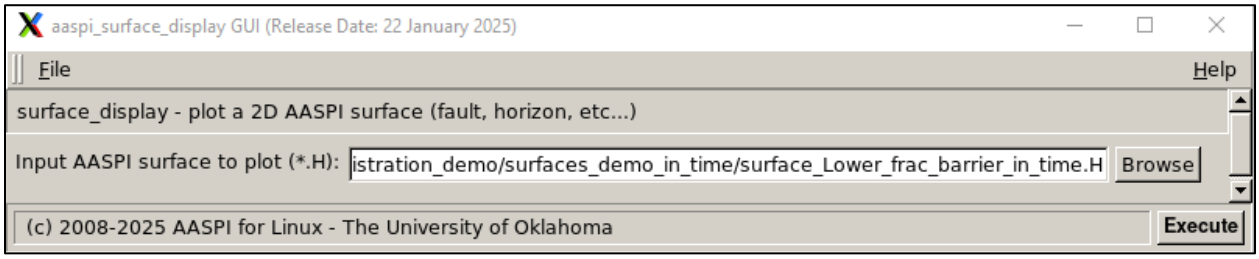
```
kmarfurt@ediacaran:~/projects/data_registration_demo $ ls -ltr surfaces_demo_in_time
surfaces_demo_in_time
total 3016
-rw-rw-r-- 1 kmarfurt kmarfurt       0 Feb  7 13:54 surface_Lower_frac_barrier_in_time.H@@@
-rw-rw-r-- 1 kmarfurt kmarfurt    2127 Feb  7 13:54 surface_Lower_frac_barrier_in_time.H@@
-rw-rw-r-- 1 kmarfurt kmarfurt 1533756 Feb  7 13:54 surface_Lower_frac_barrier_in_time.H@
-rw-rw-r-- 1 kmarfurt kmarfurt    2506 Feb  7 13:54 surface_Lower_frac_barrier_in_time.H
-rw-rw-r-- 1 kmarfurt kmarfurt       0 Feb  7 13:54 surface_Upper_frac_barrier_in_time.H@@@
-rw-rw-r-- 1 kmarfurt kmarfurt    2127 Feb  7 13:54 surface_Upper_frac_barrier_in_time.H@@
-rw-rw-r-- 1 kmarfurt kmarfurt 1533756 Feb  7 13:54 surface_Upper_frac_barrier_in_time.H@
-rw-rw-r-- 1 kmarfurt kmarfurt    2504 Feb  7 13:54 surface_Upper_frac_barrier_in_time.H
```

Plotting the results

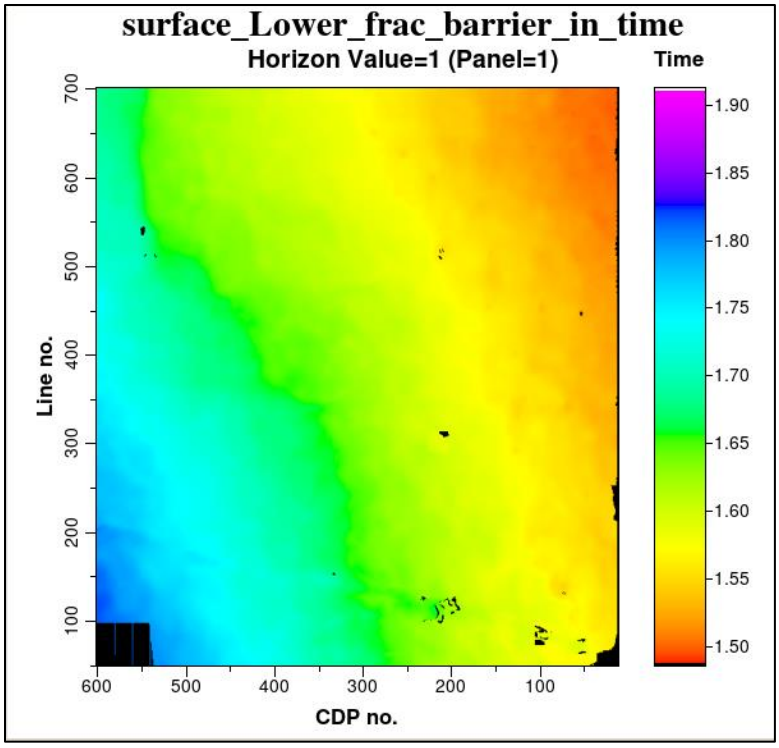
A best practice is to display the AASPI-converted surfaces to be sure the conversion was done correctly. To do so, select *Surface Utilities* in the **aaspi_util** or **aaspi_main** tab and drop down and select D*isplay AASPI-format surfaces*:



The following GUI appears with only one line to select:



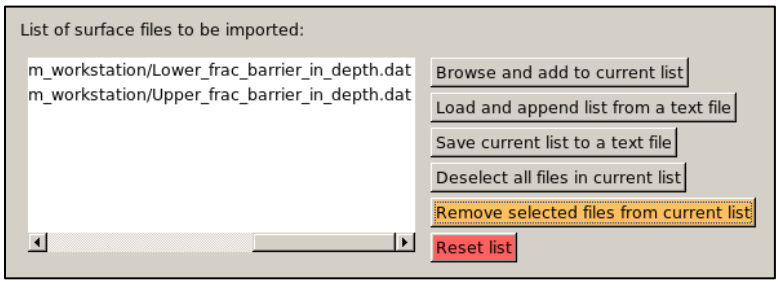
I click *Execute* and obtain the following image:



Where the color bar is in s and the black areas are no permit areas or areas where the surface could not be picked.

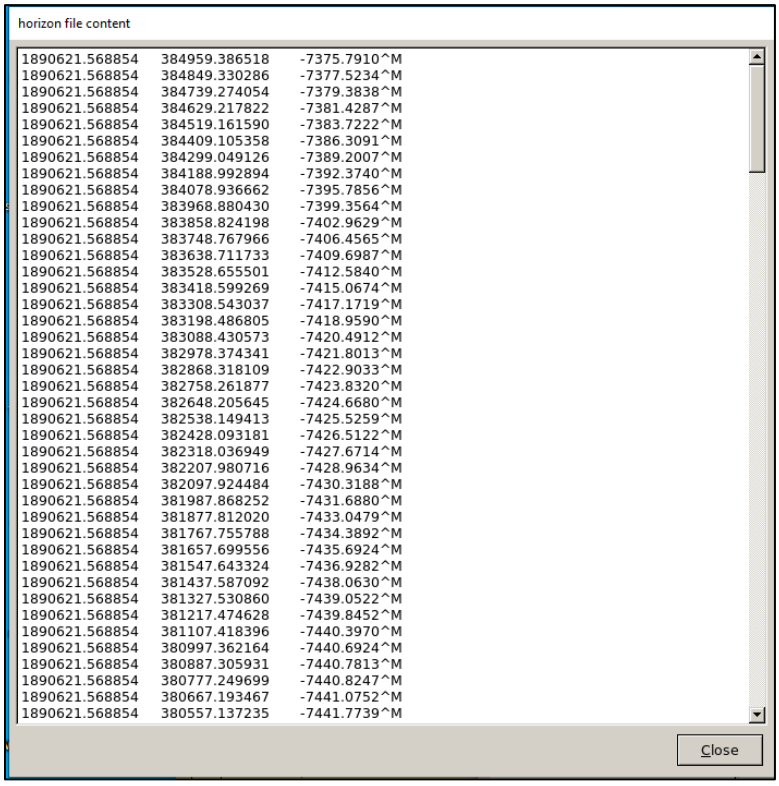
Example 2: Importing surfaces with vertical units in depth and the reference seismic volume in time

If we wish to convert a seismic or attribute data volume to depth using AASPI program **surface_driven_data_registration** we will need to import ASCII text file surfaces that are measured in depth. In this example, the horizon was picked from the input seismic data in time. Then this horizon was converted to depth using well ties of over 100 wells and kriging with external drift using a program in the *Transform* interpretation package. I go to the *List of surfaces to be imported* area and select the two depth surfaces:



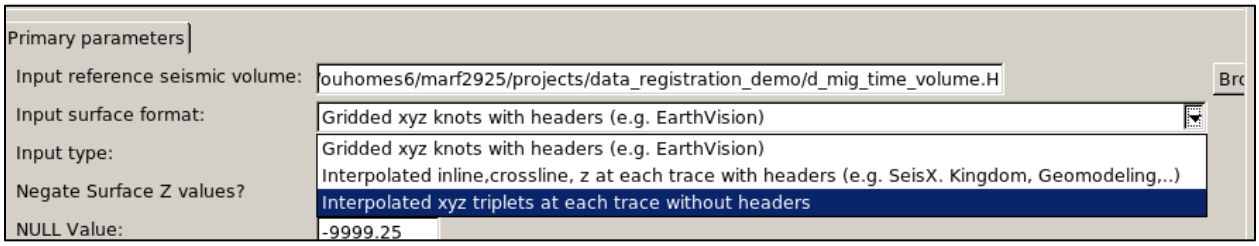
As before I click View a horizon file to be imported:

and select one of the two depth files to examine the results:



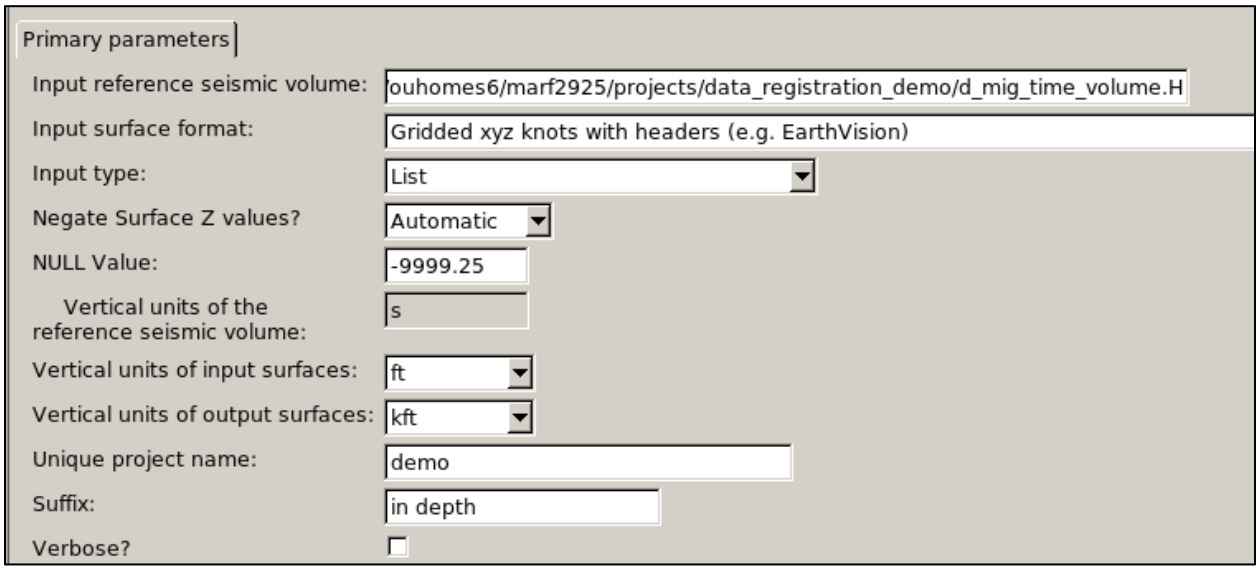
This file only has 3 columns. I can ignore the *^M* carriage return, as it will not affect the conversion. Examining the corner points of the seismic survey, I recognize that column 1 is *x* (Northing), column 2 is *y* (Easting), and column 3 is depth (in ft). I also notice that all of the depths are negative. The survey is from Oklahoma and as I look out the window I don't see and 7000 ft mountains, so I assume that for this file z is positive *up*. Both of my horizons have all negative values so I can leave the (4) *Negate surface z values* as *Automatic*.

I return to the *Input surface format* drop-down menu and select *Interpolated xyz triplets at each trace without headers*:

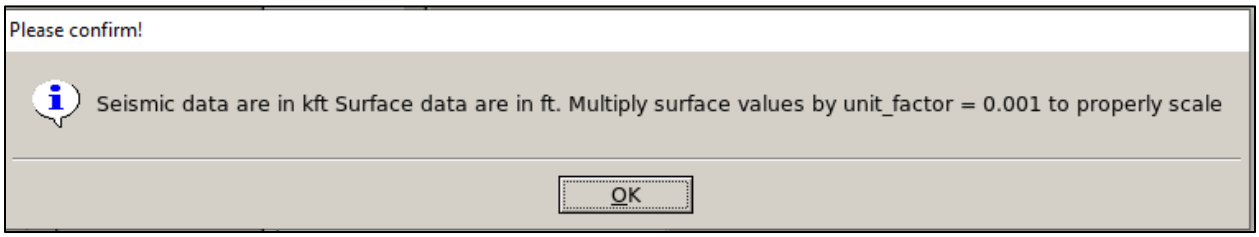


Previously, a warning message says that you need to enter the units of the input ASCII text format fault or horizon. Unlike Example 1 above, my horizons are in depth and measured in ft. I want my

output ASSPI-format horizon to be in kft. I therefore use the drop-down menus under *Vertical units of input surfaces* and *Vertical units of output surfaces* to select these two options:



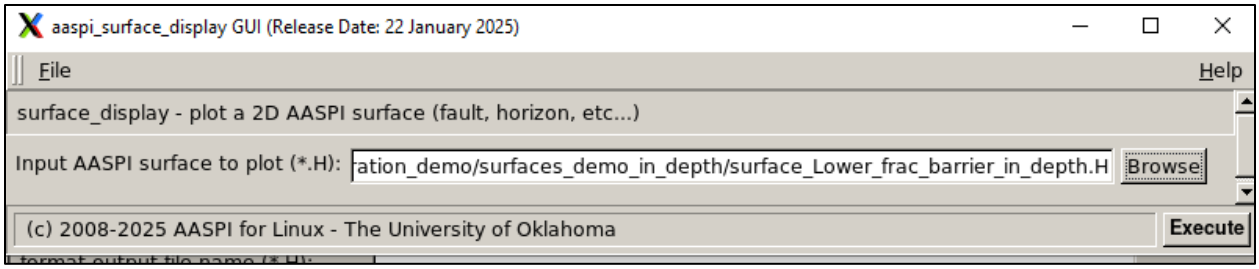
For clarity, I also change my *Suffix* to be in depth, thereby defining a directory called *surfaces_demo_in_depth*. I click Execute and obtain the following warning message:



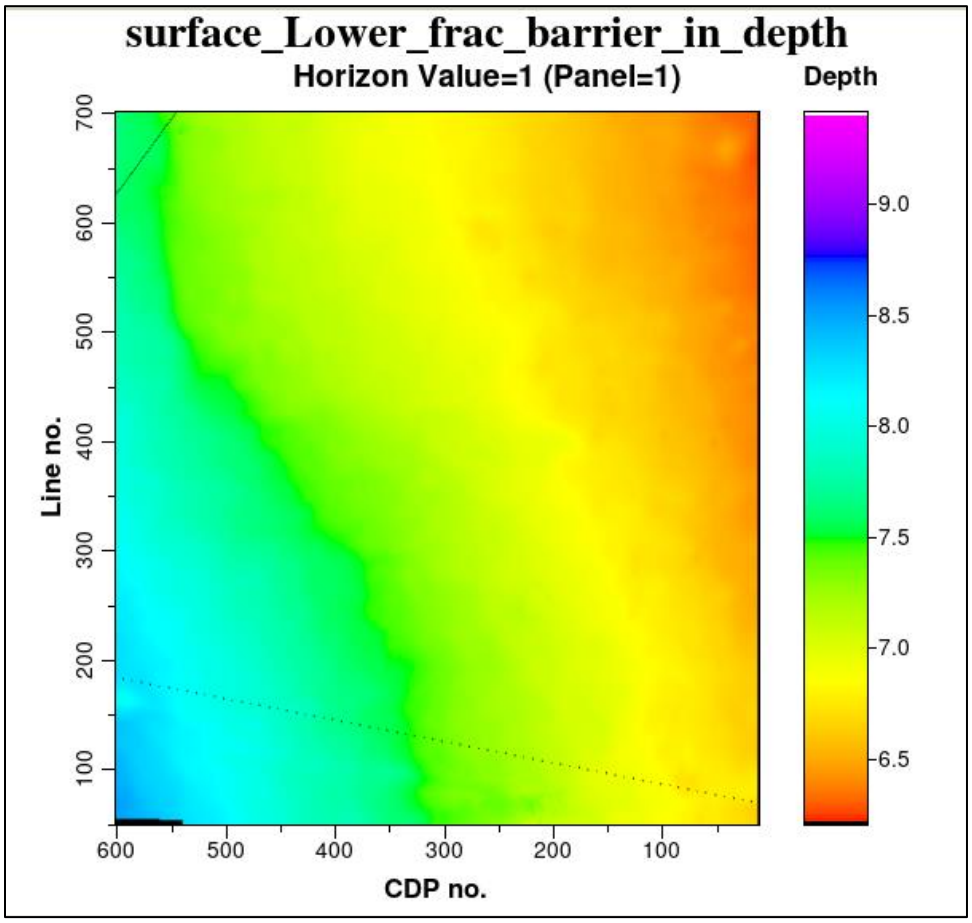
This is what I *want* to do, so I click *OK*. The program completes normally, and I can list two AASPI format depth files in their directory:

```
kmarfurt@ediacaran:~/projects/data_registration_demo $ ls -ltr surfaces_demo_in_depth/
total 3016
-rw-rw-r-- 1 kmarfurt kmarfurt       0 Feb  7 14:54 surface_Lower_frac_barrier_in_depth.H@@@
-rw-rw-r-- 1 kmarfurt kmarfurt    2129 Feb  7 14:54 surface_Lower_frac_barrier_in_depth.H@@
-rw-rw-r-- 1 kmarfurt kmarfurt 1533756 Feb  7 14:54 surface_Lower_frac_barrier_in_depth.H@
-rw-rw-r-- 1 kmarfurt kmarfurt    2502 Feb  7 14:54 surface_Lower_frac_barrier_in_depth.H
-rw-rw-r-- 1 kmarfurt kmarfurt       0 Feb  7 14:55 surface_Upper_frac_barrier_in_depth.H@@@
-rw-rw-r-- 1 kmarfurt kmarfurt    2129 Feb  7 14:55 surface_Upper_frac_barrier_in_depth.H@@
-rw-rw-r-- 1 kmarfurt kmarfurt 1533756 Feb  7 14:55 surface_Upper_frac_barrier_in_depth.H@
-rw-rw-r-- 1 kmarfurt kmarfurt    2500 Feb  7 14:55 surface_Upper_frac_barrier_in_depth.H
```

As before, I go to the **aaspi_util** or **aaspi_main** GUI and the *Surface Utilities* tab and select *surface_display* and choose a file to plot:

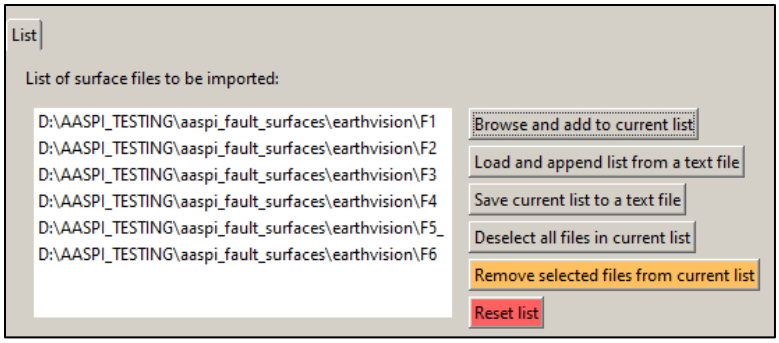


I obtain the following image:

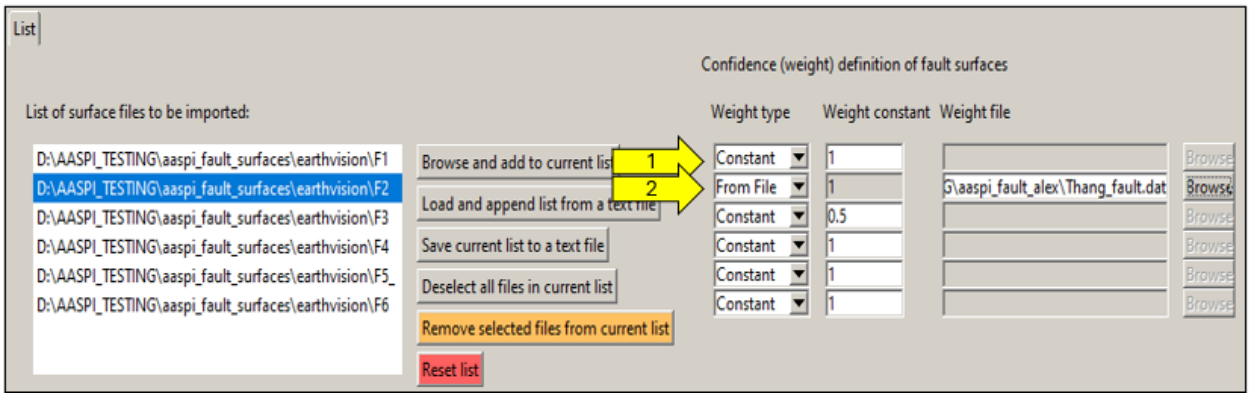


Example 3: Converting fault surfaces in ASCII text format to AASPI format

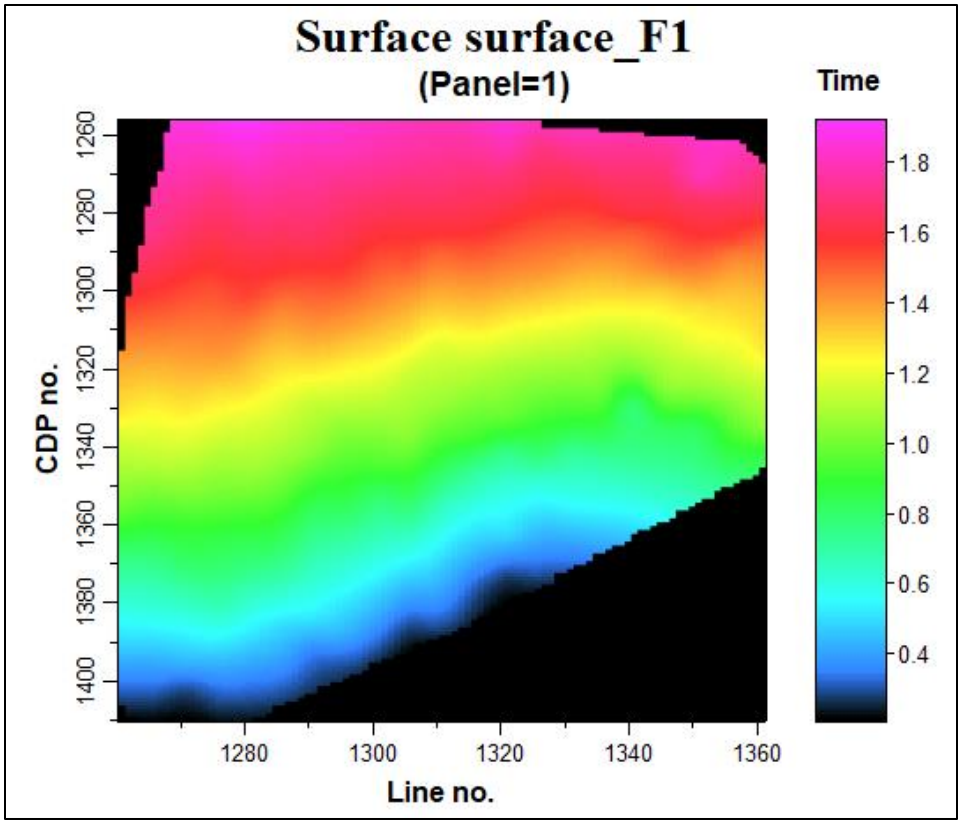
Our third example was created by Thang Ha as he developed the workflow for CNN-based fault prediction. In this application, the interpreted picked fault surfaces serve as training and validation data in the convolutional neural network. Here, he picked and then converted six different fault surfaces that were saved in *Earth Vision* format:



Optionally, for fault surfaces, users can specify the "weight" associated with each fault surface when using the *List* input type. This weight represents how confident a user is when picking a fault. The weights can be set to be (1) a constant for a given surface or (2) as variable weights using a weight file. If weight type is *Constant*, every point on a fault surface has the same "weight" (or confidence) as the weight constant the user specifies. If weight type is *From File* the weight of each point on a fault surface is read from the weight file to the right. This weight file is assumed to have the exact same format as the time/depth fault surface, with the *Z* column ranging from 0.0 (lowest confidence) to 1.0 (highest confidence), and preferably having the same boundary and grid configuration as well.



As a QC, Thang plots fault *F1* using program **surface_display**:


Surface surface_F1
(Panel=1)
Time
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
CDP no.
1260
1280
1300
1320
1340
1360
1380
1400
1280
1300
1320
1340
1360
Line no.