UNFLATTENING A SCALAR DATA VOLUME – PROGRAM unflatten

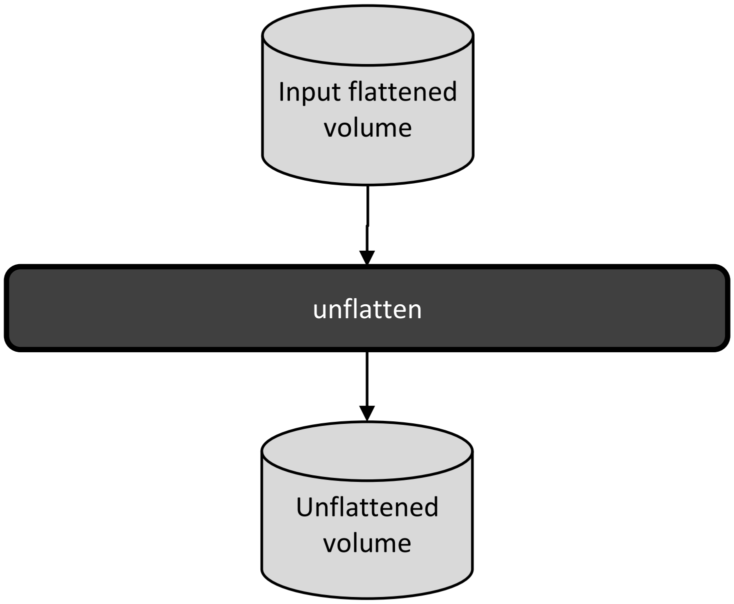
Contents

Overview

Extracting phantom horizon slices and stratal (or proportional) slices are some of the more common interpretation activities performed in interpretation workstation software. Since the interpretation workstation is where you picked your horizons, it is the obvious place to do such slicing and subsequent analysis. Nevertheless, there are reasons to create flattened or stratal sliced subvolumes in the AASPI software. Generating flattened volumes have value if your commercial software does not consist of a state-of-the-art spectral decomposition algorithm and you wish to generate a suite of volumes about a target horizon. Similarly, AASPI provides horizon-based clustering (also called classification) algorithms. AASPI programs **flatten**, **complex_spectral_flatten**, and **vector_flatten**, flatten a user-defined window of input data defined about a picked horizon. AASPI programs **unflatten**, **complex_spectral_unflatten,** and **vector_unflatten**, reverse this process. AASPI programs **stratal_slice**, **complex_spectral_stratal_slice**, and **vector_stratal_slice**, generate a suite of stratal (proportional slices) between two user-defined horizons. Flattened slices are computed by interpolating the input data using a $\varphi=2\pi f\Delta t$ Fourier phase shift of each Fourier component. Because the distance between stratal slices varies from trace to trace, it is more efficient to compute stratal slices using a simple six-point interpolation in the time (or depth) domain. Programs **flatten**, **unflatten**, and **stratal_slice**, work on scalar data volumes. Because there is a discontinuity about ±180°, phase and azimuth cannot be interpolated along the vertical axes. For this reason, complex spectra and vector quantities such as dip magnitude and dip azimuth should be flattened as pairs using programs **complex_spectral_flatten**, and **vector_flatten.** The same argument holds for interpolating to generate complex and vector stratal slices.

Computation Flow Chart

Program **unflatten** reads in a previously flattened seismic or attribute volume where the original horizon picks are stored in the trace headers and generates an unflattened output volume:



Output file naming convention

Program convolutional_modeling will always generate the following output files:

Output file description	File name syntax
Unflattened data volume	unflattened_*unique_project_name_suffix*.H
Program log information	unflatten_*unique_project_name_suffix*.log
Program error/completion information	unflatten_*unique_project_name_suffix*.err

where the values in red are defined by the program GUI. The errors we anticipated will be written to the **.err* file and be displayed in a pop-up window upon program termination. These errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the **.log* file.

Theory

Unflattening of a scalar volumes *d*(*t*) is most easily done using its frequency components *D*(*ω*) computed using a Fourier transform:

$$D\left(\omega_j\right)=\sum_{k=0}^{K} d(t_k)\exp\left(-i\omega_j t_k\right). \qquad (1)$$

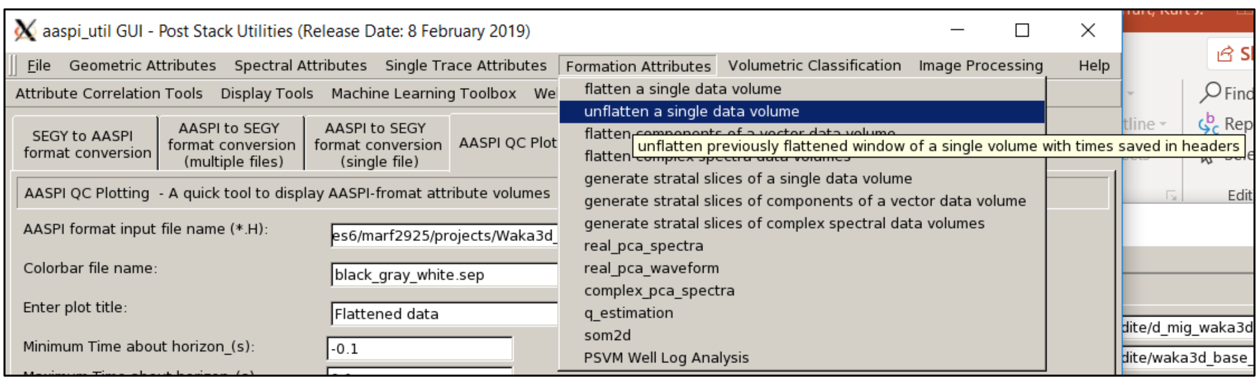
where the time shift t_k will have the opposite sign of that used in the original flattening. To shift the data centered a time *t*=0 back to time *T*, we simply use the shifting theorem in the inverse Fourier transform

$$d(t_k+T)=\sum_{j=0}^{J} D\left(\omega_j\right)\exp\left(+i\omega_j T\right)\exp\left(+i\omega_j t_k\right) \qquad (2)$$

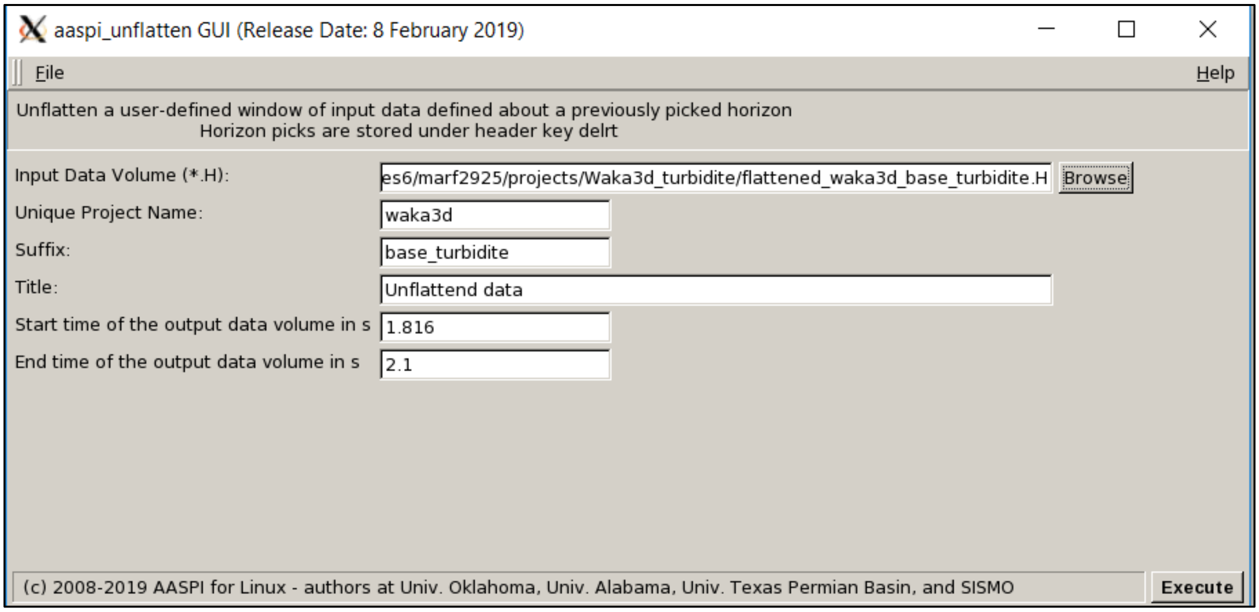
where the normalization constants have been left out.

Invoking the unflatten GUI

Program **unflatten** is launched from the *Formation Attributes* tab within in the main **aaspi_util** GUI:

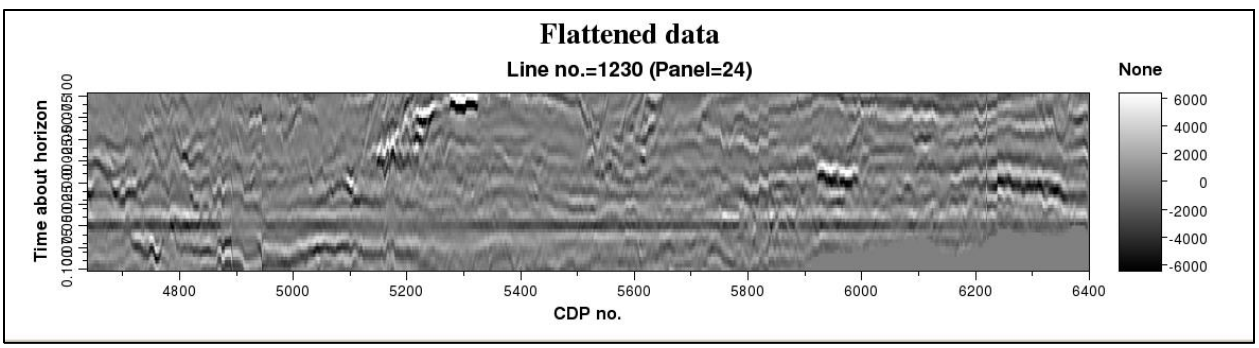


The following GUI appears:



Because the horizon has been previously stored by program **flatten** in header word *delrt*, program **unflatten** has less parameters than program **flatten**. In addition, the limits of the data used to generate the flattened horizon are stored in the *.H header file as *t_first_sample_out* and *t_last_sample_out* and read in to define defaults for the start and end time of the output data volume.

The input data flattened data volume looks like the following image:



While the corresponding output unflattened volume looks like next image.

Formation_Attributes: Program **unflatten**