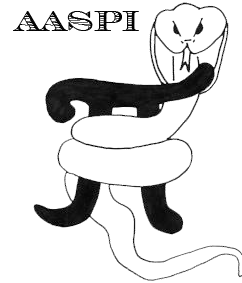


Vertically interpolate/decimate a volume – Program **interpolate**



Contents

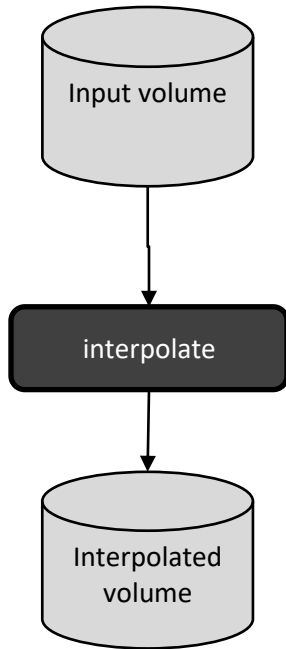
Overview	1
Computation Flow Chart.....	1
Output file naming convention.....	2
Theory	3
Interpolation using the Fourier transform.....	3
Decimation using the Fourier transform	3
Invoking the interpolate GUI	3
Examples	4

Overview

In general, seismic data are properly sampled in the field and subsequent attributes preserve the original sampling increment. However, sometimes we may wish to decrease or increase the sampling increment. For example, if the data are sampled at 4 ms and there is useable information near 125 Hz, then subsequently spectrally balanced data may plot better than when using a 2 ms sample interval rather than the original coarser sample interval. Program **aaspi_interpolate** uses fast Fourier transforms to achieve this purpose. For this reason, the interpolation and decimation factors must be factors of 2, 3, 5, and/or 7.

Computation Flow Chart

Program **interpolate** reads in a seismic amplitude and outputs either an interpolated or decimated volume:



Output file naming convention

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

Output file description	File name syntax
Interpolated/decimated volume	d_interpolated_ <i>unique_project_name_suffix</i> .H or d_decimated_ <i>unique_project_name_suffix</i> .H
Program log information	interpolate_ <i>unique_project_name_suffix</i> .log
Program error/completion information	interpolate_ <i>unique_project_name_suffix</i> .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

Interpolation using the Fourier transform

Interpolation is easily done in the frequency (wavenumber for depth-migrated data) domain by zero-padding the seismic spectrum. However, before taking the Fourier transform, there are two important steps:

- 1) Compute and subtract the mean value from each trace,
- 2) Taper the beginning and the end of each trace,
- 3) Add a short amount of padding to minimize wrap-around,
- 4) Add zeroes to end of the trace to form a trace length that is a power of 2, 3, 5, and/or 7,
- 5) Compute the forward FFT up to frequency n_f ,
- 6) Zero-padding the complex spectrum from frequency n_f+1 to Mn_f where M is the interpolation factor,
- 7) Compute the inverse FFT, and
- 8) Truncate the output to the desired length.

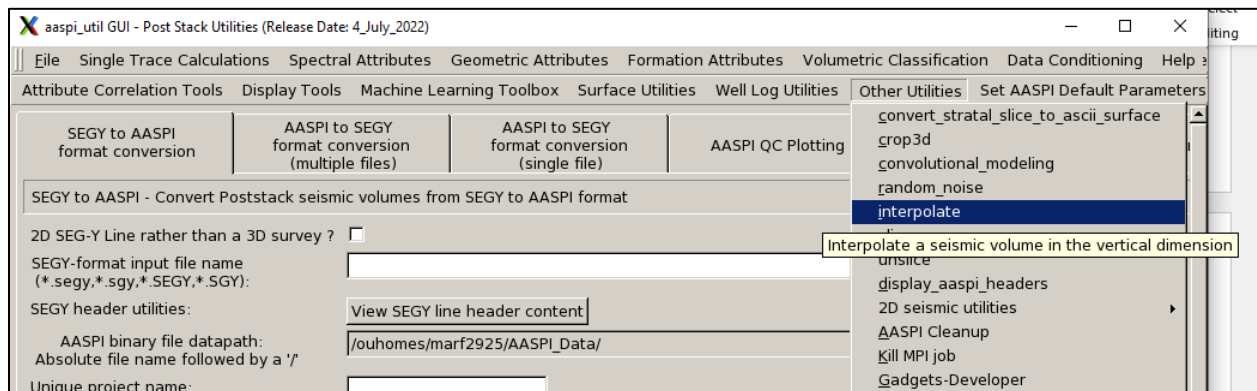
Tapering avoids generating Gibb’s phenomena (ringing) and is a standard preprocessing step for almost all filters. However, tapering input data with a DC bias such as impedance volumes is problematic. By first subtracting the mean value, the resulting trace can be tapered to zero and then zero-padded to the length of the FFT used. After interpolation, this mean value is simply added back. (An alternative approach would be to taper the data to the mean value, and then pad with the mean value to the length of the FFT used). The amplitude range of the interpolated data will be nearly identical to that of the input data.

Decimation using the Fourier transform

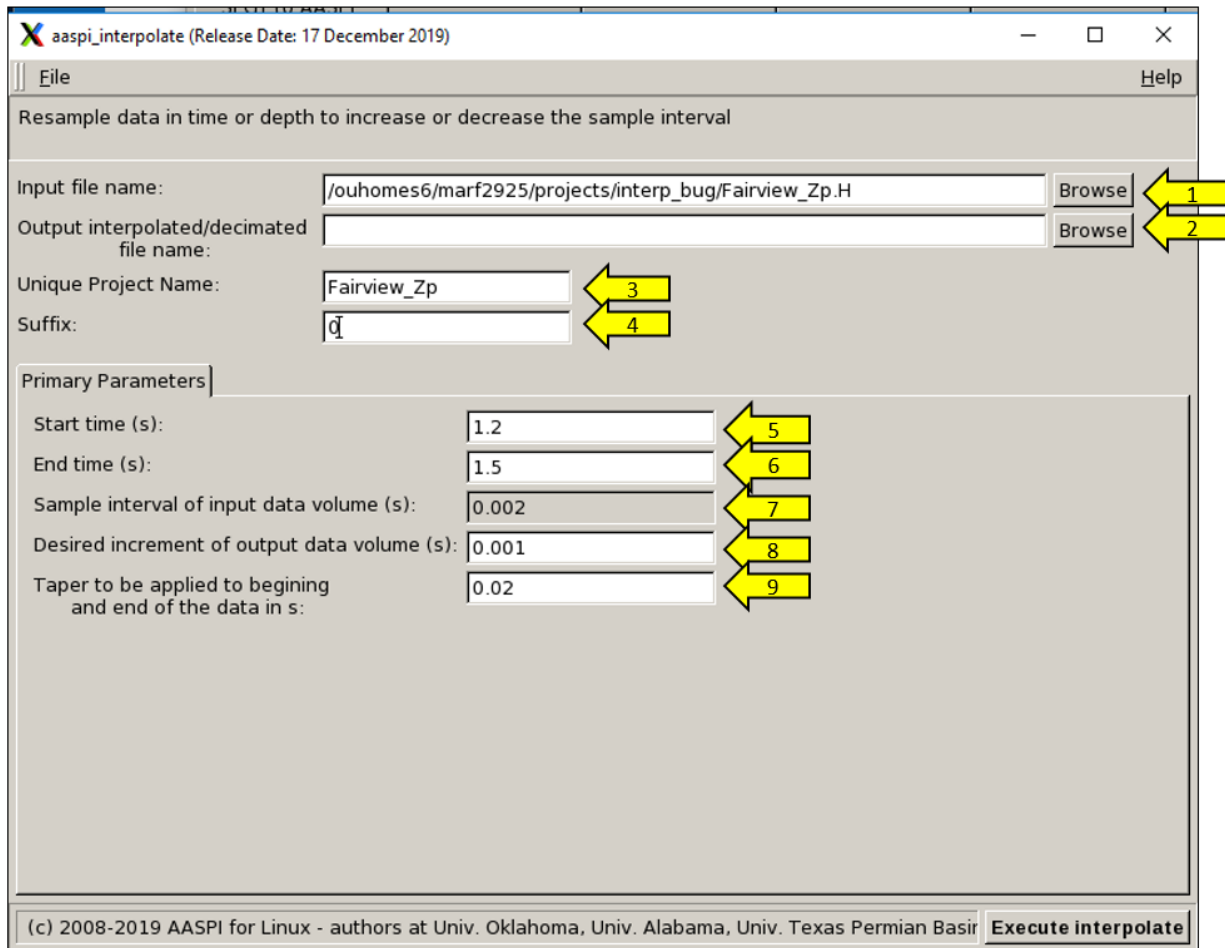
Decimation is achieved in much the same manner. The mean is computed and subtracted, and the trace tapered to zero and zero-padded. After computing the spectrum, the spectrum is simply truncated to length to n_f/M , where M is the decimation factor. Implicitly, all higher frequencies have been set to zero, thus avoiding aliasing of the higher frequencies. The mean amplitude is then added back to the result before writing out the results. Note that the amplitude range of decimated data will be slightly less than of the input data.

Invoking the interpolate GUI

Program **interpolate** is found under the **aaspi_util** GUI *Other Utilities* tab:



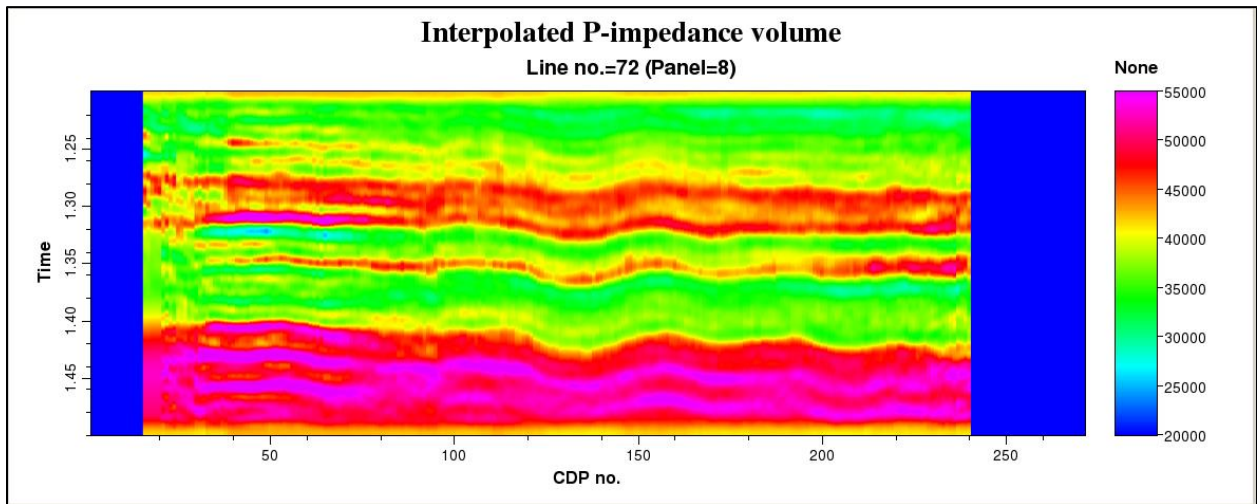
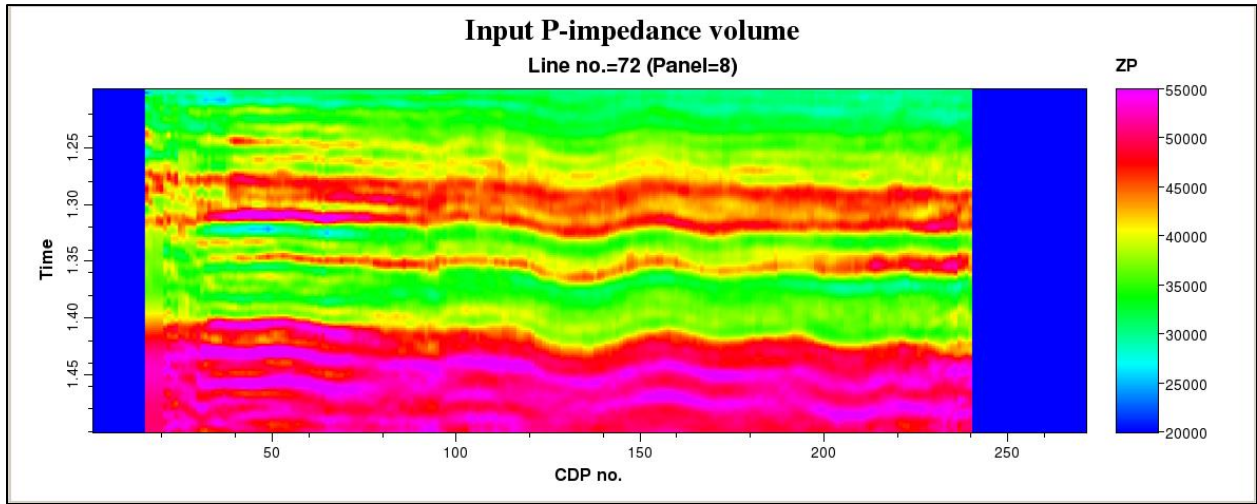
The following GUI appears:



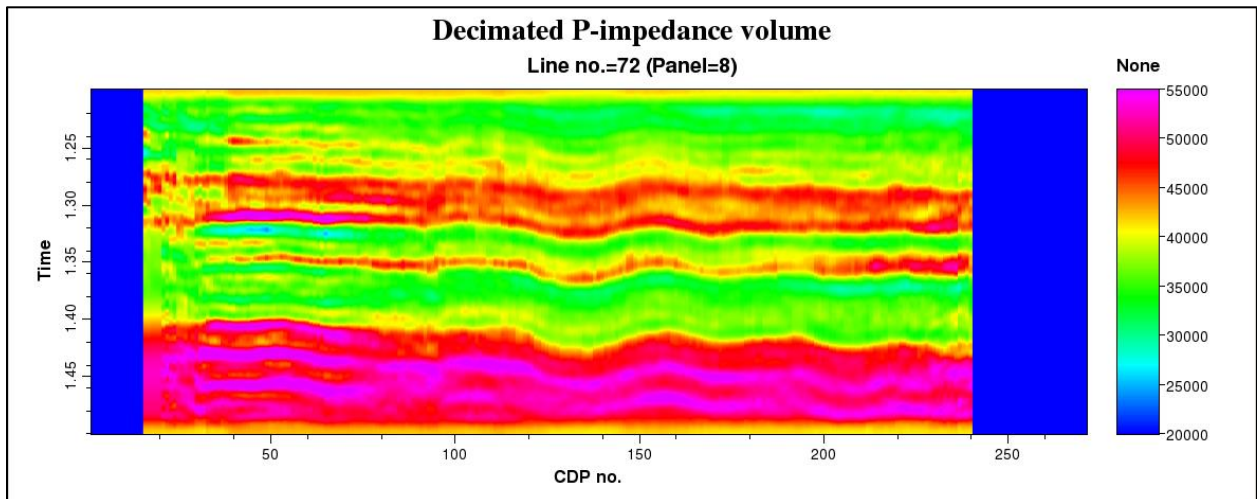
Browse (1) and choose an input file. If you wish, (2) define the name of your output file. If not, the software will define an output file name for you as described above based on (3) a *Unique Project Name* and (4) a *Suffix*. Note in this example that the data to be interpolated are a P-wave impedance volume. The default values of the (5) *Start time* and (6) *End time* are those of the input data volume but can be more restrictive. The value of the (7) *Input sample increment* is for reference, is grayed-out and cannot be changed. In contrast, the value of the (8) *Output sample increment* defines whether the program interpolates (if the *Output sample increment* is less than the *Input sample increment*) or decimates (if the *Output sample increment* is greater than the *Input sample increment*). In either case, the interpolation or decimation factor must be a power of 2, 3, 5, and/or 7. Finally, (9) a default taper of 10 samples (in this case 0.020 s) is applied to the beginning and end of the mutes for each trace to suppress any Gibb's phenomena.

Examples

The example below shows an input P-impedance volume that has been interpolated by a factor of two (from 0.002 s to 0.001 s) using the parameters shown above:



Next, let's decimate the same data volume by a factor of three to a sample rate of 0.006 s to obtain the following result:



Other Utilities – Vertically interpolate/decimate a volume – Program **aaspi_interpolate**

Note the taper at the beginning and ends of both the interpolated and decimated volumes.