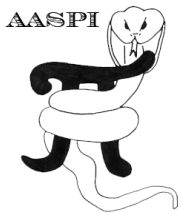


INTRODUCTION AND SOFTWARE OVERVIEW

Document Objectives

The objective of this document is to show how to submit and QC attribute jobs on a multi-processor Linux system using simple Graphical User Interfaces (GUIs) and a minimum number of Linux commands. While we hope that some insight on how the programs work can be gained, we will defer detailed descriptions of parameters and how they effect the output to separate documents for each application. Not suprisingly, the optimum choice of parameters depends not only on the seismic data quality but also on the geologic objectives to be imaged. Most of our work to date that addresses this sensitivity can be found through case studies on the AASPI web site *www.geology.ou.edu/aaspi* under the *Publications* area. However, sponsors have told us that they would also like to see a suite of studies that examine sensitivity to specific parameters, such as analysis window size, number of iterations, and so forth. In this manual, we strive to generate such documentation.

Major Improvements in the September 21, 2012 Release

Software structure improvements

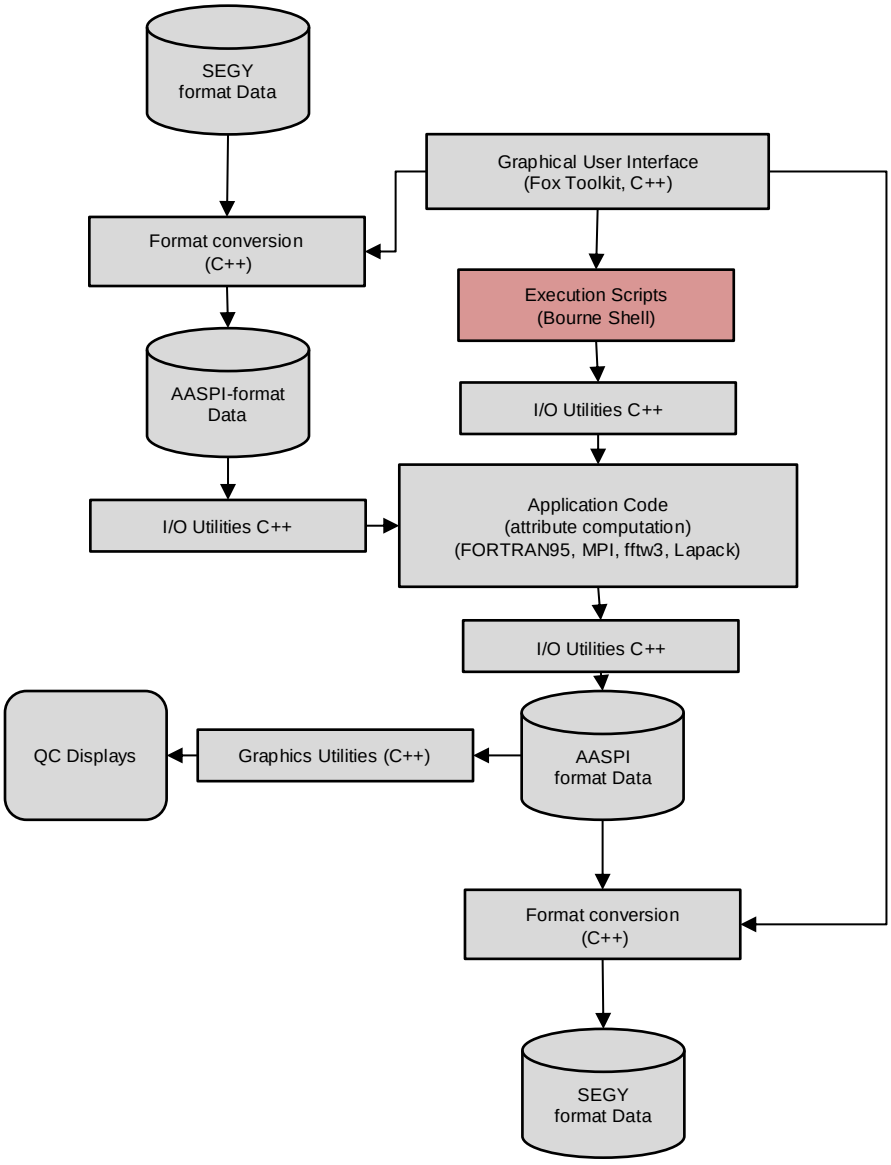
The initial 2007 AASPI software was designed to run on a multiprocessor Linux environment and was built on the Stanford Exploration Project (SEP) file format (http://sepwww.stanford.edu/doku.php?id=sep:software:seplib). Because of both long life and continuous contribution by our academic colleagues at Stanford to Geophysical algorithm development, SEP formats and utilities are well understood and accepted by the geophysical research community, particularly those involved in seismic data processing and migration. During the past three years, a more portable, open source "Madagascar" or *rsf* variation of the SEP framework was developed (http://www.reproducibility.org/wiki/Main_Page). The file formats for both SEP and *rsf* formats are very similar, though at present, the Madagascar community does not have extensive trace header manipulation utilities. Like SEP, Madagascar is tightly linked to Linux constructs, such as those that allow the use of pipes to communicate between modules.

The major improvement in the AASPI software during 2011 has been the construction of what we denote **aaspi_io** utilities. Written in portable C++ by Tim Kwiatkowski, **aaspi_io** compiles under both Linux and the Microsoft Windows operating systems. These utilities support conversion to and from the complete 2002 SEGY standard (including 16-bit integer, 32-bit integer, IBM floating point, and IEEE floating point data formats), command line arguments, as well as reading and writing both binary files and trace headers. Several annoying but difficult to fix "features" buried in the SU and SEP frameworks are now circumvented, the two most annoying of which were the truncation of long file names and conversion of the binary trace header to the little-endian if the

environmental variable F_UFMTENDIAN was unknowingly set by some other 3rd party application.

As of September 21, 2012 the following AASPI components all work on Windows 7:

- All graphical user interfaces (GUIs) which are written in C++,
- All data conversion, command line argument handling, and i/o (using **aaspi_io** written in C++),
- All Fortran90 application codes,
- The MPI 3rd party library,
- The fftw-3 3rd party library, and
- The FX toolkit (used in the GUIs and graphics), and a
- The **aaspi_plot** utility to display seismic amplitude and attribute data.



The major remaining component that needs to be addressed is replacing Linux (Bourne) shell scripts with a more portable scripting language such as Python. We have

addressed this task in two steps. First, we have replaced the command line interface between the GUIs and shell scripts with simpler and more portable parameter files. The second (and remaining) step is to replace each (now standalone) shell script with an equivalent python script. MPI will present challenges on Windows because of the need to install specialized security systems. Running across nodes (different physical computers) may require changes by sponsor IT staff. Running in parallel locally (on say 8 processors) may not have security issues since the job is on the same compute node.

Workflows

In 2007 we broke two large geometric attribute programs into their primary components – program **dip3d**, **similarity3d**, and **curvature3d**, which allowed inserting filtering operations using newer programs such as **image_filt3d** and **sof3d**. While providing maximum flexibility, such jobs needed to be run sequentially, with several hours required for the computationally more intensive components, such as program dip3d. Our workflow is based on the assumption that the user will have selected appropriate parameters on a smaller cropped subvolume of the original data. These parameters can then be defined in a new **geometric attribute workflow** GUI, which builds a large shell script that runs each of the components in a predefined order in a batch 'overnight' mode.

Default parameters

We have provided a means for both the system administrator and each user to modify most of the more commonly reused parameters in an *aaspi_default_parameters* file that will reside in the ${AASPIHOME}/par, user home, and local directories. Commonly reused parameters include a list of compute node names, the number of processors per node, the number of colors used in 2D color bars, and conversion velocities appropriate for the area of interest. Setting the default parameters is discussed in Section 2 of this manual.

We have also provided a means of generating user-defined names as part of the AASPI to SEGY format conversion resulting in a naming convention that may more easily integrate with a given site's data base or interpretation workstation naming conventions.

Algorithmic additions

Given the new underlying i-o and graphics, all algorithms have been modified during the past nine months with a few undergoing major revision. For example we have added lower-middle-upper (LUM) and multistage-median modified trimmed mean (MSMTM) filters to our programs **image_filt3d** and the LUM filter to programs **sof3d** and **sof3d_prestack**.

Program **aaspi_plot** provides not only portable display, but the flexibility we need to QC multiattribute displays. Multiattribute display programs **hlplot**, **hsplot**, **hlsplot**, and

rgbplot all provide up to 24-bit color output. While most commercial interpretation software packages are limited to 256 (8-bit) color, several of the newer ones are not. For instance, with a small Ocean plug-in, Petrel can display 4096 colors. While we provide most of the color tables as output, we do not have them all, so please provide us with ASCII versions of color tables you use.

Program **crossplot** is simple but very effective in interactive volumetric clustering. Here, the most common application is to crossplot two geomechanical attributes (such as $\lambda\rho$-$\mu\rho$ computed using Hampson-Russell's Strata) against a 2D color bar. The crossplotted data are then loaded into an interpretation workstation along with the 2D color bar. Many workstations provide a crossplot tool that allows one to highlight subregions of the data volume corresponding to user-defined polygons of a given 'rock type' in 3D. These subvolumes can then be exported to further correlate with well measurements, microseismic events, and production.

Program **som2d** is a formation-based clustering algorithm that examines vertical patterns of attributes seen on a suite of stratal slices to generate a seismic facies map. One application here have included clustering of lambda-rho and mu-rho stacking patterns in shale reservoirs.

Programs **sof3d** and **gtm** provide a computer-driven unsupervised clustering of volumetric multiattribute data. Program **sof3d**, first released in 2011, provides a 3D means of facies classification based on Kohonen self-organizing maps useful in a seismic stratigraphy or seismic geomorphology framework (e.g. Roy et al., 2011). Newly-released program **gtm** provides a probabilistic clustering based on generative topographic maps. Both of these algorithms have proven useful in mixing mechanical measures (such as $\lambda\rho$-$\mu\rho$) with structural measures (such as coherence and curvature) and stratigraphic measures (such as spectral components and texture attributes) to differentiate facies using both rock properties and seismic stratigraphy measures.

In prestack applications, program **mpnmo** provides a non-stretch NMO correction algorithm based on the same concepts of wavelet matching pursuit used in spectral decomposition program **spec_cmp**. **nmo_spec** is best used in a Deregowski loop velocity analysis workflow that is based on a complex matching pursuit construct developed by Bo Zhang. Bo has also prototyped high resolution velocity analysis algorithms based on the KL transform. Future components include least-squares interval velocity algorithm as well as long offset analysis, with the goal of using interval velocities and velocity anisotropy as a means of estimating Total Organic Carbon (TOC) from surface seismic data to compare with 40% anisotropy core measurements made by our colleagues in OU's rock physics laboratory.

Program **sof_prestack** applies structure-oriented filtering to flattened common reflection point gathers. The workflow is similar to the edge-preserving post stack structure-oriented filtering program sof3d, with principal component, alpha-trim mean, or LUM filters being independently applied to common offset, common azimuth, or

common vector tile volumes using a consistent structural direction and edge preservation computed from the stacked volume.

We have also released a simple 3D prestack Kirchhoff time migration algorithm, **azim_offset_mig**. While we do not expect AASPI sponsors to use such a simple algorithm rather than those they have developed internally or use through service companies, such a release provides a clean, easy-to-understand framework for subsequent algorithms addressing constrained least-squares migration, diffraction imaging, azimuthal anisotropy analysis, imaging of very long offset data and 2C by 2C processing.

Software Overview

The AASPI software is a suite of products that generate seismic attributes and apply simple filters that currently runs under Linux on a network of distributed processors. The AASPI software is not tightly linked to any commercial interpretation software system. The reasons for not doing so are two-fold. First, our consortium members use a wide range of workstations including Kingdom Suite, Voxelgeo, Geoframe, GeoProbe, Petrel, Geomodeling, Transform, OpenDtect, and so on, with several companies using all of the above! Second, not all of these software products have a developer's tool kit. Products that have attempted to do so (such as OpenSpirit) have not gained wide acceptance.

To maximize portability and facilitate software development, the software is developed in layers, which from top-to-bottom consist of:

a) **Graphical User Interfaces (GUIs**) that provide an easy way to launch jobs, quality control user-defined parameters, and generate simple graphical displays. The GUIs are written in C++ and are found in directory *${AASPIHOME}/src/cpp_poststack/program_name* (for post-stack GUI) and *${AASPIHOME}/src/cpp_poststack/program_name* (for pre-stack GUI) where the value *${AASPIHOME}* contains the name of the directory containing the AASPI software which was set up by the person who installed the software.
b) **Shell scripts** that read parameters from the GUI, do some simple error checking when possible, and submit a job in the background. The shell scripts can be found under *${AASPIHOME}/scripts/* and can be modified to suit your needs.
c) **Python scripts** that essentially function like shell scripts, but are compatible with both Windows and Linux version of AASPI. The python scripts can be found under *${AASPIHOME}/pyscripts/.*

d) **FORTRAN2003** application codes that perform the actual computation. We have strived to make the code internally documented and to maintain a consistent look and feel. Basically, our workgroup starts with a small suite of templates. At the end, Marfurt takes working prototype code developed by students and reworks them so they look as if they were written by the same person. Sponsors who have installed the software have access to the source code. An application called *program_name* will be found in your installation under *${AASPIHOME}/src/f90_poststack/program_name* (for post-stack application) or *${AASPIHOME}/src/f90_prestack/program_name* (for pre-stack application). Please note that some of the experimental codes are located under *${AASPIHOME}/src/linux_only.* Reused AASPI subroutines can be found in a library called*${AASPIHOME}/src/f90_lib/fortran_mod.*

e) **OpenMPI** is a public domain library (included with RedHat) that controls jobs that run across a distributed network (e.g. if you have eight dual-processor workstations down your hall on the same network you would be able to run the job on 16 processors). Your IT administrator will need to set up *passwordless secure shell* for you to do this.

f) **fft-w** (**the fastest Fourier transforms in the West**) is a suite of Fourier transform libraries developed by MIT. We currently use version 3.

g) **LAPack** is a suite of matrix inversion and eigen decomposition routines developed by the US National Laboratories.

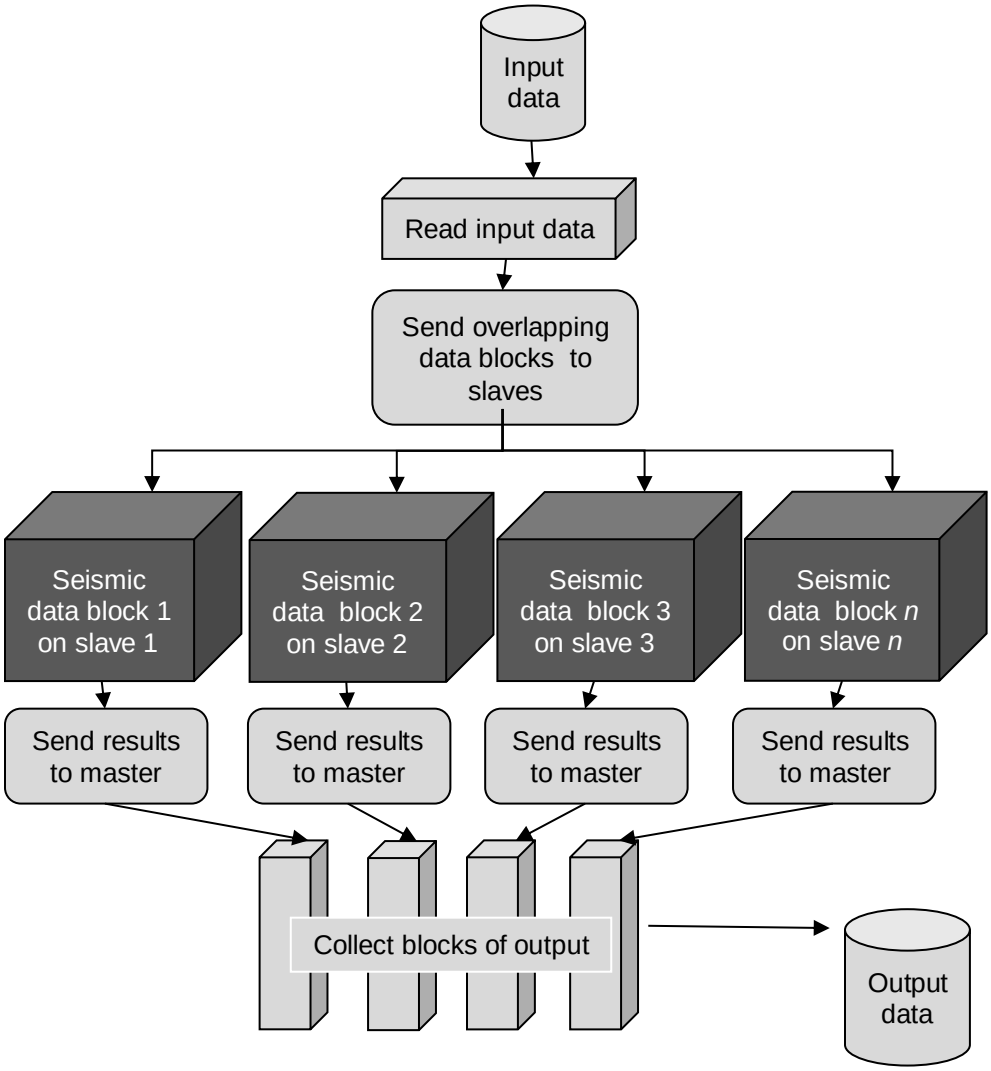
h) **aaspi_io** is a suite of portable I/O and command line libraries written in C++ by Tim Kwiatkowski that supplant those provided by SEPlib. This portability will allow us to move towards deploying the AASPI software on a Microsoft Windows platform.

i) **SEPlib** is a suite of utilities provided by the Stanford Exploration Project (a geophysical consortium running since the mid 1970s). As of September 21, 2011, all SEP I/O and command line subroutines as well as all but three or four rarely used programs (**Merge, Attr, Graph**, and **Math**) have been removed from the AASPI software. While all of the other applications run on Windows7, SEPlib only runs Linux. Its potential successor, Madagascar, has a similar Linux pipe construct and so that it too does not currently run on Windows7 either. We have maintained the SEPLib data structure so that all SEP and Madagascar applications will be able to read these files. AASPI applications can read all SEPLib and Madagascar formats that retain the trace headers using SEP historical header format (hff) construct.

At present, nearly all the AASPI applications run in parallel on distributed Linux computation processors. These processors may be a single 64-node compute server, several 8-node systems in a computer room, or a suite of single- or dual-node processors in various offices down the hall. The AASPI software is written primarily in Fortran2003, uses libraries written in C and C++ (such as **FFTW** and **aaspi_io**) with parallelization implemented using the Message Passing Interface (**OpenMPI**). The input

and output data are padded to be 3D or 4D rectangular volumes in order to simply parallelization.

For programs **dip3d, image_filt3d, sof3d, similarity3d**, **curvature3d, glcm3d,** each seismic sample requires the same amount of work. In the figure below, the seismic data are sent by the master to four slaves. Upon completion, the slaves send the data back to the master, which writes the results out to disk. Since matching pursuit is a trace-independent nonlinear process whose computation time depends on the seismic waveforms, **spec_cmp** uses a different, trace by trace parallelization template.



References

Roy, A., M. Matos, and K. J. Marfurt, 2011, Application of 3D clustering analysis for deep marine seismic facies classification – an example from deep water northern Gulf of Mexico: to be presented at the GCSSEPM 31st Annual Bob. F. Perkins Research Conference.