Design and specifications for SLIM’s software framework

Road map and progress

SLIM group at UBC EOS

August 2006
About

- Typical approaches to do seismic imaging
- Our way: Python OOP instead of scripting
- SLIM software distribution for SINBAD
What is typically used?

- Seismic Tools
  - RSF, Seismic Unix, SEP, Delphi
- Higher-level languages like Matlab
- Scripting tools:
  - Unix shells, Perl, Python, etc.
- Matlab, SCons
What is the problem?

- Long-learning process – one has to learn a variety of tools and their specific details
- Cumbersome and slow development using inadequate, often low-level programming tools
- Implementation details hard-coded into the applications
- Low-level of abstraction and reusability
- Serial code
Typical developer / user
How to avoid the tragedy?

**ANAs**

An ANA is a numerical algorithm that can be expressed abstractly solely in terms of vectors, vector spaces, and linear operators (i.e. not with direct element access).
Seismic packages

Similarities: `sffdct2 < in.rsf > out.rsf` (RSF)
- out-of-core UNIX command-line operators
- operations on UNIX pipes
- ability to combine operators using UNIX pipes
- serial

Differences:
- data formats
Where do we go?

- Simplicity: \( y = A^*x + b \) (Matlab/Mathematica)
- Abstract Numerical Algorithms (ANA)
- Vectors / Arrays
- Operators (Matrices)

- Universality
  - hidden details of abstract objects/algorithms
  - hidden serial-parallel implementation

- Reusability
How do we get there?

OO Abstraction

- Python object-oriented interface:
  - Abstract Data Types for seismic data
  - Abstract Operators for seismic-tools commands
  - Abstract Numerical Algorithms
How do we get there?

Implementation

- Python wrappers for Python/C++ interfaces to specific seismic-data formats (vectors)
- Python-interface wrapper for specific commands of seismic tools (operators)
- Above implementation details hidden from ANAs and the user
- Choice of serial or parallel algorithm
“Where do we go”
Benefits

- Faster learning curve
- Faster prototyping and development
- Easy implementation of new seismic tools
- Reusability
- Independent of software licenses
Our tools of choice

- RSF (for seismic tools development)
- Python (for OO ANA/user interface)
- NumArray for internal arrays
- PyPar for parallel execution
- PETSc (for distributed overlapping data)
Software development at SLIM

- Command-line operators (SLIM contribution to RSF)
- SLIMpy
  - OO Python interface to seismic packages
  - parallel algorithms for distributed data
  - involving data communication
  - embarrassingly parallel
SLIM for SINBAD
Software components

- CurveLab extension (CurveLab 2.0)
- Extensions to RSF (rev 1808)
- SLIMpy suite (OO Python interface to command-line based seismic tools)
SLIMpy components

- core
- ANAs (algorithms)
- Applications
- Demos
SLIMpy core

- Out-Off-Core objects with RSF support
  - serial
  - parallel
    - window decomposition
    - embarrassingly parallel
- In-Core objects
SLIMpy core – cont.

- Support for other seismic packages
- Sparse matrix representation
- Differential operators
- Link to external solvers like Trylinos
SLIMpy ANAs

- blocksolver.py: Primary-multiple separation through a block relaxation
- landweber.py: Generalized Thresholded Landweber Conjugate Gradient
- Conjugate Gradient
- Least SQuares (linear) Regression (LSQR)
- Stage-wise Orthogonal MAching Pursuit (STOMP)
SLIMpy ANAs

- Ground Roll Prediction and Removal Through Morphological Component Analysis
- New detection-estimation Scheme:
  - Detection with Sparse spike De-convolution
  - Estimation with Image Appearance Manifolds (IAM)
- Seismic image recovery
- Diagonal approximation of migration operator
- Wave-atom transform
SLIMpy applications

- dnoise.py: 2D de-noising using Landweber method
- interpolation.py: 2D/3D interpolation using a sparsity constraint in a transform domain
- pm_separation.py: primary-multiple separation
- pm_separation_noise.py: primary-multiple separation with de-noising

All of above are included in SLIMpy demos
Software distribution

https://wave.eos.ubc.ca/Software/SINBAD/

- SSL protected data transfer
- Access for either:
  - users with password
  - designated IPs
WARNING:

YOU are the beta tester

- SLIMpy is actively developed
- released version is only alpha tested
- there is a number of SLIMpy features that are under development
- only parts of the code that are used in ANAs, apps, and demos are tested
If you ever feel like that:
Software support

Mailing lists:

- **SLIM2RSF-user:**
  - [http://slim.eos.ubc.ca/mailman/listinfo/slim2rsf-user](http://slim.eos.ubc.ca/mailman/listinfo/slim2rsf-user)

- **SLIMpy-user**
  - [http://slim.eos.ubc.ca/mailman/listinfo/slimpy-user](http://slim.eos.ubc.ca/mailman/listinfo/slimpy-user)
Summary

- SLIM software development paths:
  - new algorithms for seismic imaging
  - command-line based seismic tools
  - SLIMpy

- SLIM software distribution for SINBAD members