IWAVE++: a Framework for Imaging and Inversion based on Regular-Grid Finite Difference Modeling

William W. Symes

The Rice Inversion Project

Department of Computational and Applied Mathematics
Rice University, Houston, TX

symes@caam.rice.edu

Rice Vector Library

Abstract Optimization: the Rice Vector Library ("RVL")

C++ classes expressing calculus in Hilbert Space

Design Paper: Padula, Scott & S, ACM TOMS 2009

High level abstractions – Space, Vector, (linear or nonlinear) Operator, Functional

Evaluation objects – organize the value of function & derivatives at a point, enforce coherency

Abstract Optimization: the Rice Vector Library (“RVL”)

Typical use: migration looks like

```
MyKindaDataSpace dsp(…);
MyKindaModelSpace msp(…);
Vector m(msp); Vector g(msp); Vector d(dsp);
…
MyKindaModelingOp op(….);
OperatorEvaluation opeval(op,m);
opeval.getDeriv().applyAdj(d,g);
```

“THE MATH IS THE API”

Abstract Optimization: the Rice Vector Library ("RVL")

Built on RVL:

Optimization Library `UMin`: LBFGS, trust region G-N-K, CG, Arnoldi…

Abstract time-stepping library `TSOpt`, including universal implementation of optimal checkpointing (Griewank 92)

Plans: additional algorithms (L1, TV, matrix-free TR-SQP, …)

Abstract Optimization: the Rice Vector Library ("RVL")

Critical component: standard interface to concrete data types – in-core, out-of-core, distributed,…

- `DataContainer` – data abstraction, forms Visitor pattern with
- `FunctionObject` – encapsulates all actions on data

RVL Objects = *intrusive handles* – underlying data not exposed, limited access, usually no `operator new`

Examples: FunctionObjects to perform array operations for linear algebra, associate out-of-core data with Vectors, etc.

New data types – build these components!

RVL + IWAVE = IWAVE++

Reverse Time Migration and Full Waveform Inversion

Requirements for any inversion implementation:

- modeling
- linearized (“Born”) modeling
- adjoint (transposed) linearized modeling [= RTM]
- optimization algorithm, implementation
- interface modeling and optimization

Our approach:

- maximize code re-use
- high-quality abstract optimization, linear algebra library (RVL)
- middleware layer forms interface

Reverse Time Migration and Full Waveform Inversion

Code re-use - build on IWAVE:

- define additional interfaces needed using IWAVE types, minimal extensions:
    - `gts_adj(RDOM * p, RDOM * r, int iv, void * gfd_pars)`
- re-use parallel automation, job control, i/o from IWAVE

However IWAVE data structures (RDOM etc.) are not RVL vectors, and simulators are not operators…

Reverse Time Migration and Full Waveform Inversion

Middleware layer – IWAVE++

• C++ classes encapsulating high-level drivers for IWAVE functions, Born & adjoint extensions
• delegates checkpointing, optimization functions to TSOpt
• RVL data types translated to IWAVE in/outputs
[WWS, Enriquez & Sun, Geophys. Prosp. 11]

Release with acoustic staggered grid app: Q2 12

Claims to fame: works just like IWAVE, passes dot product test […demo]