SAND2016-10686C

# Anasazi & Belos tutorial


**Alicia Klinvex & Mark Hoemmen**
**Sandia National Laboratories**
**24 Oct 2016**

Sandia is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

# Why Anasazi & Belos?

- Very similar interfaces; designed concurrently
- Anasazi: iterative eigensolvers
- Belos: iterative linear solvers

# How to get started

- Look at Doxygen homepage to learn about methods
  - trilinos.org -> Packages -> Anasazi -> Anasazi Doxygen
  - Read the cited papers
- See “Examples” tab on package’s Doxygen page for examples of how to use these methods
- Contact trilinos-users@trilinos.org
  - Please be specific!

# Available Belos solvers

- Fixed Point (Richardson)
- Conjugate Gradient
- MINRES
- GMRES
- BiCGStab
- PETSc KSP solvers (through external interface)
- And more!

**If you pick an inappropriate solver,**
**Belos will produce an error.**
**e.g., trying to use CG on a non-Hermitian problem**
**Warning: Belos cannot detect whether A is SPD**

# Belos solver types

- Block Krylov methods
  - Builds a single Krylov subspace
  - Right-hand sides are linearly dependent
- Pseudo-block Krylov methods
  - Mathematically equivalent to multiple independent solves, but more computationally efficient
  - Right-hand sides are independent
- Recycling solvers
  - Sequences of linear systems
- Flexible variants
  - Nested linear solvers

# Belos components

- Linear problem
- Solver factory
- Solver managers
- Status tests
- Iterations
- Orthogonalization managers
- States
- Errors (Failures)

**You can generally ignore everything but these!**

# Belos linear problem

- setOperator
- setLHS (initial guess; replaced by computed solution)
- setRHS
- setLeftPrec, setRightPrec (optional)
- setHermitian
- setProblem
  - Call when you finished putting in information
  - Computes the initial residual & preconditioned residual

# Common solver options

- Convergence tolerance
- Maximum iterations
- Block size
- Num blocks (maximum subspace dimension)
- Verbosity

# Example: Belos + Ifpack2 preconditioner

- Trilinos/packages/ifpack2/example/ex1.cpp

# What if I don't have a matrix?

- Belos will accept custom operators too
  - Wants two things
    - Apply method
    - Ability to query whether you can apply the transpose (not required)
  - Custom operator should be subclass of
    - Epetra::Operator
    - Tpetra::Operator
    - Belos::Operator
- Also accepts custom multivector types
  - Subclass of Belos::MultiVec

# Available Anasazi eigensolvers

- Block Krylov Schur
  - Similar to Arnoldi, Lanczos
- TraceMin, TraceMin-Davidson
  - similar to Jacobi-Davidson
- LOBPCG
- Riemannian Trust Region method
- Block Davidson, Generalized Davidson

**If you pick an inappropriate solver,**
**Anasazi will produce an error.**
**e.g., trying to use LOBPCG when A is non-Hermitian**
**Warning: Anasazi cannot detect whether A is SPD**

# Anasazi components

- Eigenproblem
- Solver managers
- Status tests
- Iterations
- Orthogonalization managers
- States
- Errors (Failures)

**You can generally ignore everything but these!**

# Anasazi eigenproblem

- setA
- setM
- setPrec
- setInitVec
- setAuxVecs
- setNEV
- setHermitian
- setProblem

$$Ax = \lambda Mx$$

# Common solver options

- Which
- Convergence Tolerance
- Maximum Iterations/Maximum Restarts
- Block Size
- Num Blocks
- Use Locking
- Verbosity

# Example: TraceMin-Davidson

- Trilinos/packages/anasazi/tpetra/example/TraceMinDavidson/TraceMinDavidsonUserOpEx.cpp

# What if I don't have a matrix?

- Same as Belos
- Anasazi accepts user-defined operators and multivector types
- Operator must be subclass of Epetra::Operator, Tpetra::Operator, or Anasazi::Operator
  - Anasazi only needs apply method
- Multivector must be subclass of Anasazi::MultiVec

# What if I want to use BKS with shift-and-invert?

- Need to perform spectral transformation
  - $Ax = \lambda Mx \rightarrow A^{-1}Mx = \lambda^{-1}x$
- Suggestion: Modify existing examples
- Please see Trilinos/packages/anasazi/epetra/example/BlockKrylovSchur/
  - BlockKrylovSchurEpetraExGenAmesos.cpp
  - BlockKrylovSchurEpetraExGenAztecOO.cpp
  - BlockKrylovSchurEpetraExGenBelos.cpp


# Questions?