Objectives

Develop a next-generation atmosphere model by leveraging Albany, a C++, parallel, implicit, unstructured-grid finite element code from Sandia National Laboratories that demonstrates agile components and enables rapid prototyping. Unique features will include embedded uncertainty quantification, performance portability, and a wide variety of discretizations.

Component-Based Strategy

Component-based approach enables rapid development of new production codes embedded with transformational capabilities

"Components" = Libraries Software Tools Interfaces Demonstration Applications

Sandia’s components effort includes ~100 interoperable libraries

Verification Tools

Dependency graph of finite element assembly kernels for the Shallow Water equations. By use of operator overloading-based automatic differentiation (AD), the same code base is used for implicit and explicit calculations. Most kernels shown here are general-purpose finite element calculations that were provided by Albany.

UQ Approach for Aeras

Aeras utilizes the uncertainty quantification capabilities available in Albany through Dakota and the Trilinos package Stokhos.

Stokhos provides tools to enable:
- Stochastic Galerkin Methods
- Concurrent Ensemble Sample Evaluation

Concurrent Ensemble Sample Evaluation

Aeras uses the concurrent ensemble sample evaluation capability of Stokhos to improve the computational efficiency of uncertainty quantification.

In this approach an ensemble of samples is propagated concurrently at the scalar level of the simulation code. This allows for:
- Improved memory access patterns
- Increased vectorization of the operations
- Reduced communication overhead in MPI codes
- Increased opportunities for fine-grained parallelism

Numerical Results

Shallow Water Test Case 5 with mountain height treated as a random variable

Concurrent Ensemble Sample Propagation can dramatically reduce the computation time for uncertainty quantification compared to standard sampling methods in which each sample is evaluated independently.

For this test case, using a concurrent ensemble size of 32 decreases the execution time by a factor of 4 to 6 compared to standard sample evaluations.