SAND2011-8261C

Panzer: A Finite Element Assembly Engine for Multiphysics Simulation

Roger Pawlowski, Eric Cyr, and John Shadid
Sandia National Laboratories

Trilinos User Group Meeting
November 2nd, 2011

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000

History

- Research over past 7 years in Charon:
  - Export control (4D001) restricted collaborations
  - Complicated build system (some great features including TPL management)
  - Restricted to Nevada mesh database/element library
- Generalize the capabilities explored in Charon into Trilinos packages
  - New code base → flexibility, lessons learned
  - Resulting packages: Phalanx, Panzer

What is Panzer?

- A general finite element assembly engine for multiphysics simulation:
  - Produces quantities need for advanced **solution** and **analysis** algorithms: residuals, Jacobians, parameter sensitivities, stochastic residual/Jacobians, etc.
  - A **unification** of Trilinos discretization tools: Shards, Intrepid, Phalanx, Sacado, Stokhos, (Optionally: STK, SEACAS)
  - Supports 1D, 2D, and 3D calculations
- A library – NOT a terminal application
  - Allows for multiple instantiations
- Contains NO physics specific code
  - Generic assembly tools
- Leverages Template-based Generic Programming to assemble quantities of interest

Research Requirements

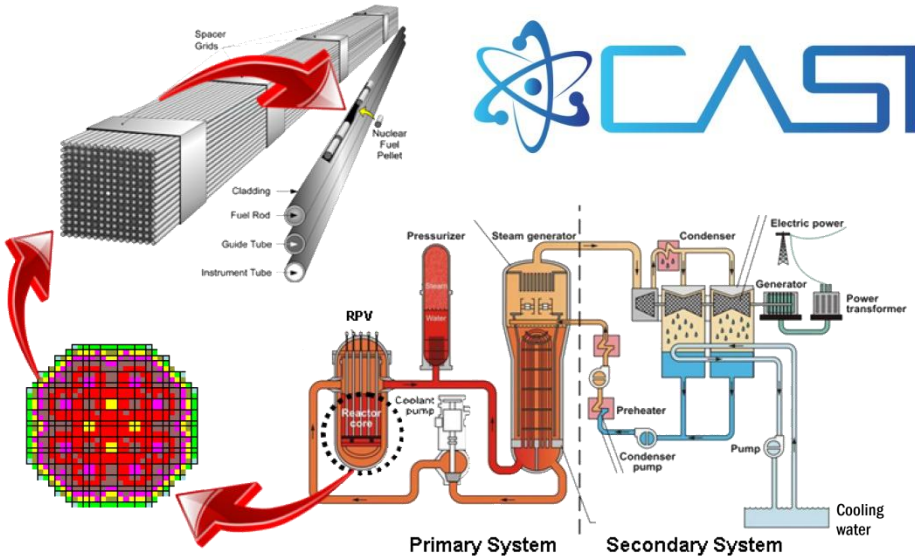
A Research Tool for DOE/OS: ASCR/AMR, ASCR/Multiscale

- **Formulations: fully coupled fully implicit, semi-implicit, FCT**
- **Compatible discretizations:**
  - Mixed basis for DOFs within element block
  - Arbitrary element types (not restricted to nodal basis)
- **Multiphysics:**
  - Fully coupled systems composed of different equation sets in different element blocks
  - Preconditioning: Approximate block factorization/physics based
- **Supports advanced analysis techniques:**
  - Supports Template-based Generic Programming
  - Adjoint-based error analysis
  - Stability, bifurcation, embedded (SAND) optimization, embedded uncertainty quantification (Stokhos/PCE)

Production Requirements

Production Quality Software (ASC, CASL)

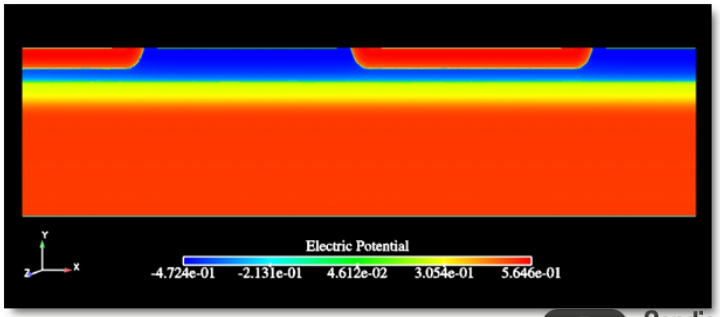
- Strict and extensive unit testing (TDD)
- Integration with legacy code components
- NOT restricted to any mesh database or I/O format
- Control over granularity of assembly process (efficiency vs flexibility)
- Applications:
  - ASC: Semiconductor Device (Next-generation Charon) for QASPR
  - CASL: CFD component for VERA simulator



CASL

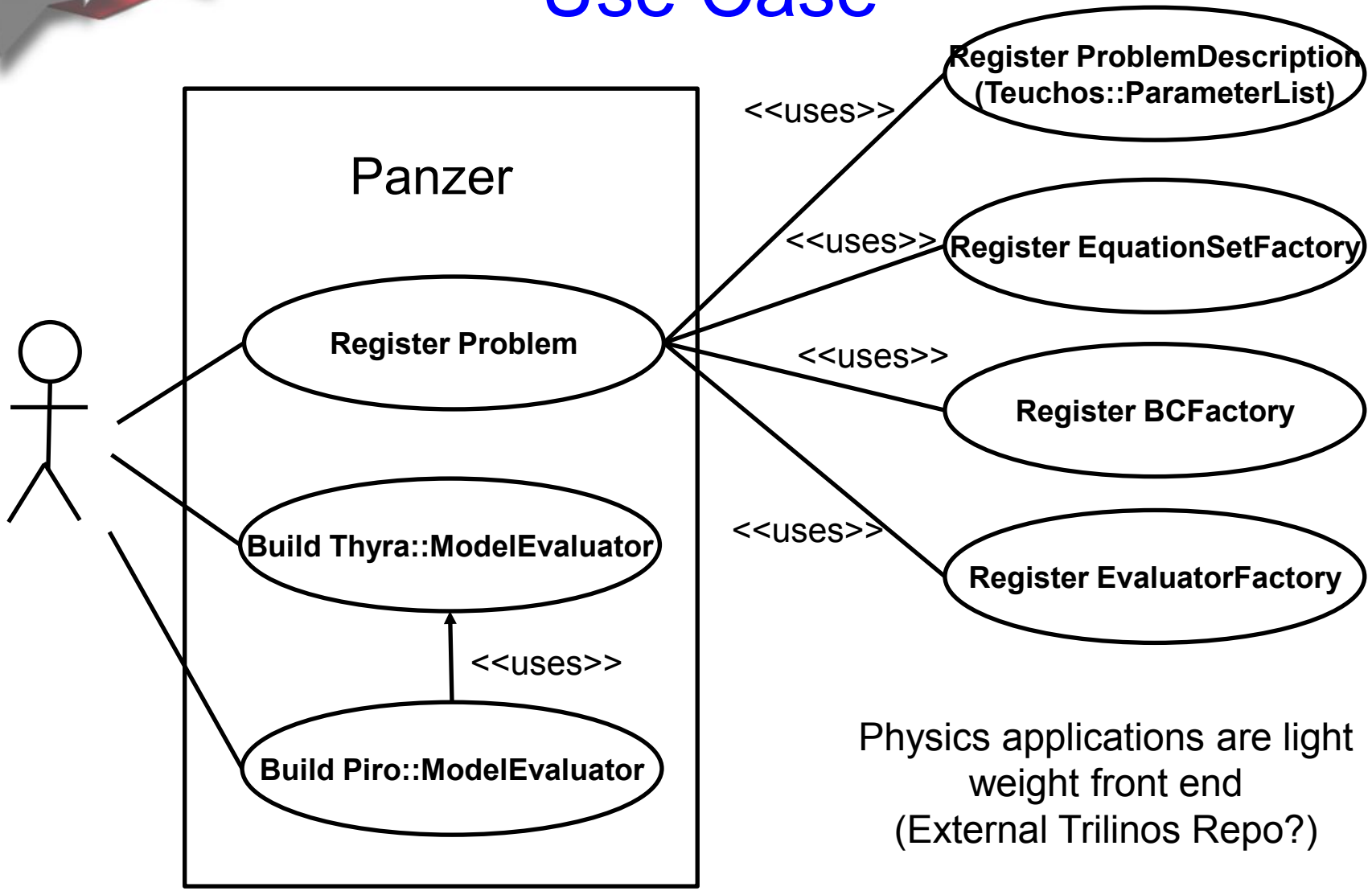
ASC™

DOE / NNSA



Sandia National Laboratories

Use Case



Physics applications are light weight front end
(External Trilinos Repo?)

Examples of Nonlinear Analysis (Supported by ModelEvaluator)

Nonlinear equations:	Solve $f(x) = 0$ for $x \in \mathbf{R}^n$
Stability/Bifurcation analysis:	For $f(x,p) = 0$ find space $p \in \mathcal{P}$ such that $\frac{\partial f}{\partial x}$ is singular
Explicit ODEs:	Solve $\dot{x} = f(x,t) = 0, t \in [0,T]$, $x(0) = x_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
DAEs/Implicit ODEs:	Solve $f(\dot{x}(t), x(t), t) = 0, t \in [0,T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$ for $x(t) \in \mathbf{R}^n, t \in [0,T]$
Explicit ODE Forward Sensitivities:	Find $\frac{\partial x}{\partial p}(t)$ such that: $\dot{x} = f(x,p,t) = 0, t \in [0,T]$, $x(0) = x_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
DAE/Implicit ODE Forward Sensitivities:	Find $\frac{\partial x}{\partial p}(t)$ such that: $f(\dot{x}(t), x(t), p, t) = 0, t \in [0,T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
Unconstrained Optimization:	Find $p \in \mathbf{R}^m$ that minimizes $g(p)$
Constrained Optimization:	Find $x \in \mathbf{R}^n$ and $p \in \mathbf{R}^m$ that: minimizes $g(x,p)$ such that $f(x,p) = 0$
ODE Constrained Optimization:	Find $x(t) \in \mathbf{R}^n$ in $t \in [0,T]$ and $p \in \mathbf{R}^m$ that: minimizes $\int_0^T g(x(t),p)$ such that $\dot{x} = f(x(t),p,t) = 0$, on $t \in [0,T]$ where $x(0) = x_0$

Sandia National Laboratories

Graph-based Assembly Process

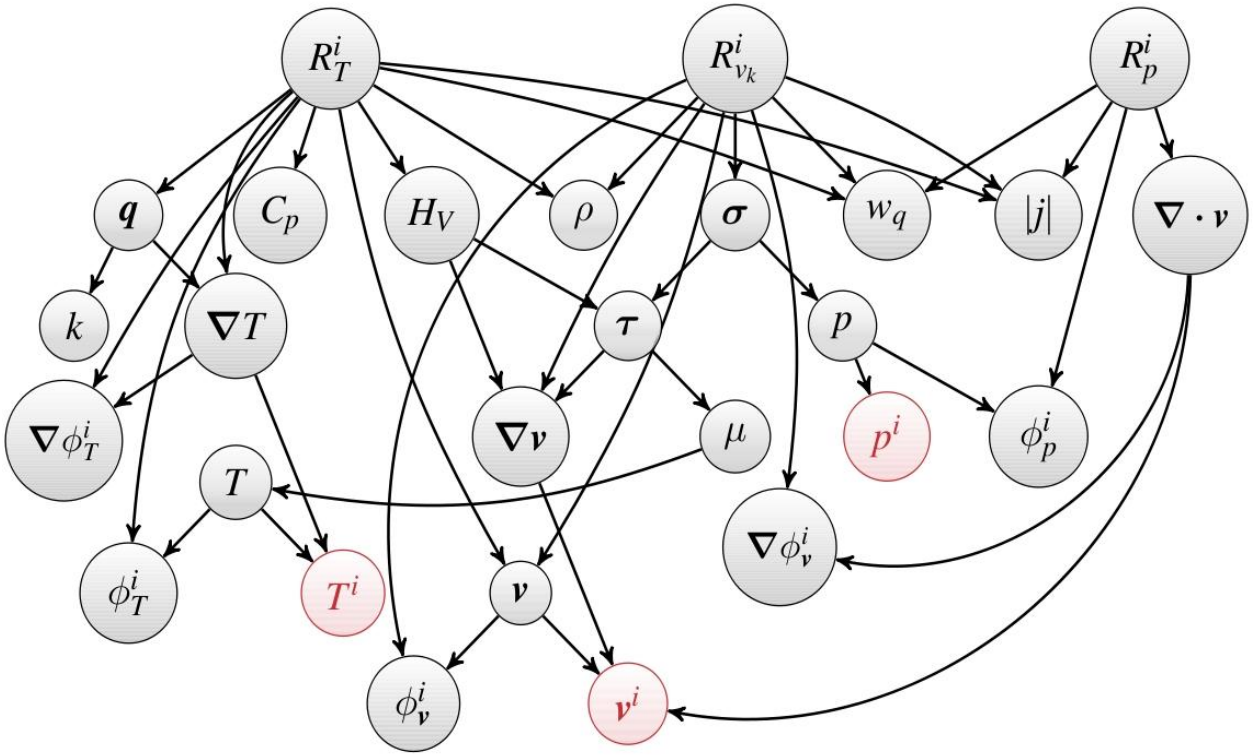
(Notz, Pawlowski, Sutherland; submitted to TOMS)

- Graph-based equation description
  - Automated dependency tracking (Topological sort to order the evaluations)
  - Each node is a point of extension that can be swapped out
  - Easy to add equations
  - Easy to change models
  - Easy to test in isolation
  - User controlled granularity
  - No unique decomposition
- Worksets (blocks of cells)
- User controlled memory allocation of Field data
- Multi-core research:
  - Spatial decomposition
  - Algorithmic decomposition

$$R_T^i = \sum_{e=1}^{N_e} \sum_{q=1}^{N_q} \left[\left(\rho C_p \boldsymbol{v} \cdot \nabla T - H_V \right) \phi_T^i - \boldsymbol{q} \cdot \nabla \phi_T^i \right] w_q |j| = 0$$

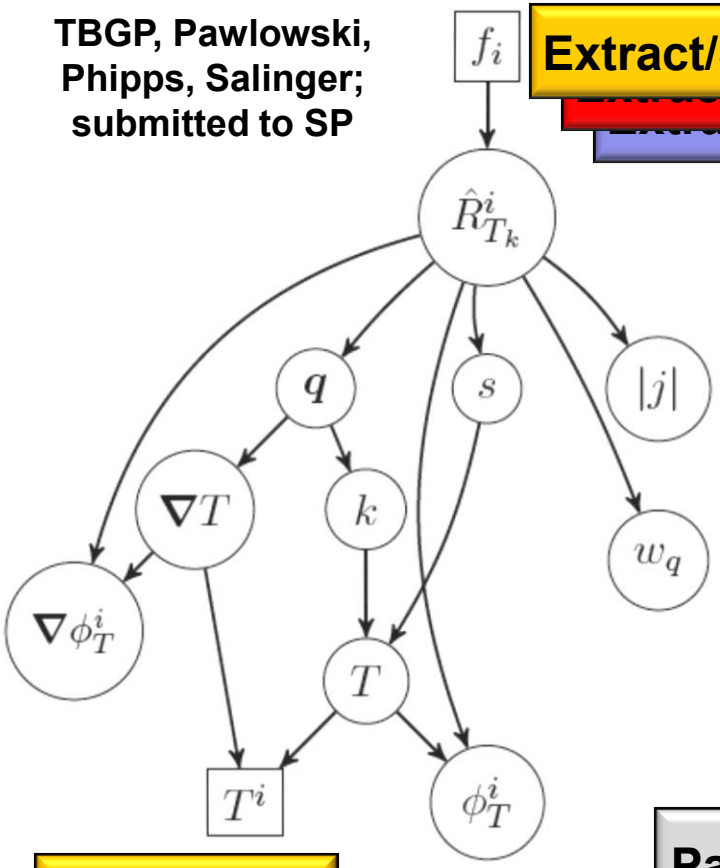
$$R_{v_k}^i = \sum_{e=1}^{N_e} \sum_{q=1}^{N_q} \left[\rho \boldsymbol{v} \cdot \nabla \boldsymbol{v} \phi_{\boldsymbol{v}}^i + \boldsymbol{\sigma} : \nabla \left(\phi_{\boldsymbol{v}}^i \boldsymbol{e}_k \right) \right] w_q |j| = 0$$

$$R_p^i = \sum_{e=1}^{N_e} \sum_{q=1}^{N_q} \nabla \cdot \boldsymbol{v} \phi_p^i w_q |j| = 0$$



Phalanx Handles Multiphysics Complexity using Template-based Generic Programming

TBGP, Pawlowski, Phipps, Salinger; submitted to SP



$$f(x) = \sum_{k-1}^{N_w} f_k = \sum_{k-1}^{N_w} Q_k^T \hat{R}^i_{T_k}(P_k x)$$

$$\hat{R}^i_T = \sum_{e=1}^{N_e} \sum_{q=1}^{N_q} \left[-\nabla\phi^i_T \cdot \boldsymbol{q} + \phi^i_T s \right] w_q |j| = 0$$

Evaluation Type	Scalar Type
$f(x,p)$	`double`
$J = \frac{\partial f}{\partial x}$	`DFad<double>`
$\frac{\partial^2 f}{\partial x_i \partial x_j}$	`DFad< DFad<double> >`

Param. Sens., Jv, Adjoint, PCE (SGF, SGJ), Arb. Prec.

Take Home Message:
Reuse the same code base!
Equations decoupled from algorithms!
Machine precision accuracy!

`PCE::OrthogPoly<double>`

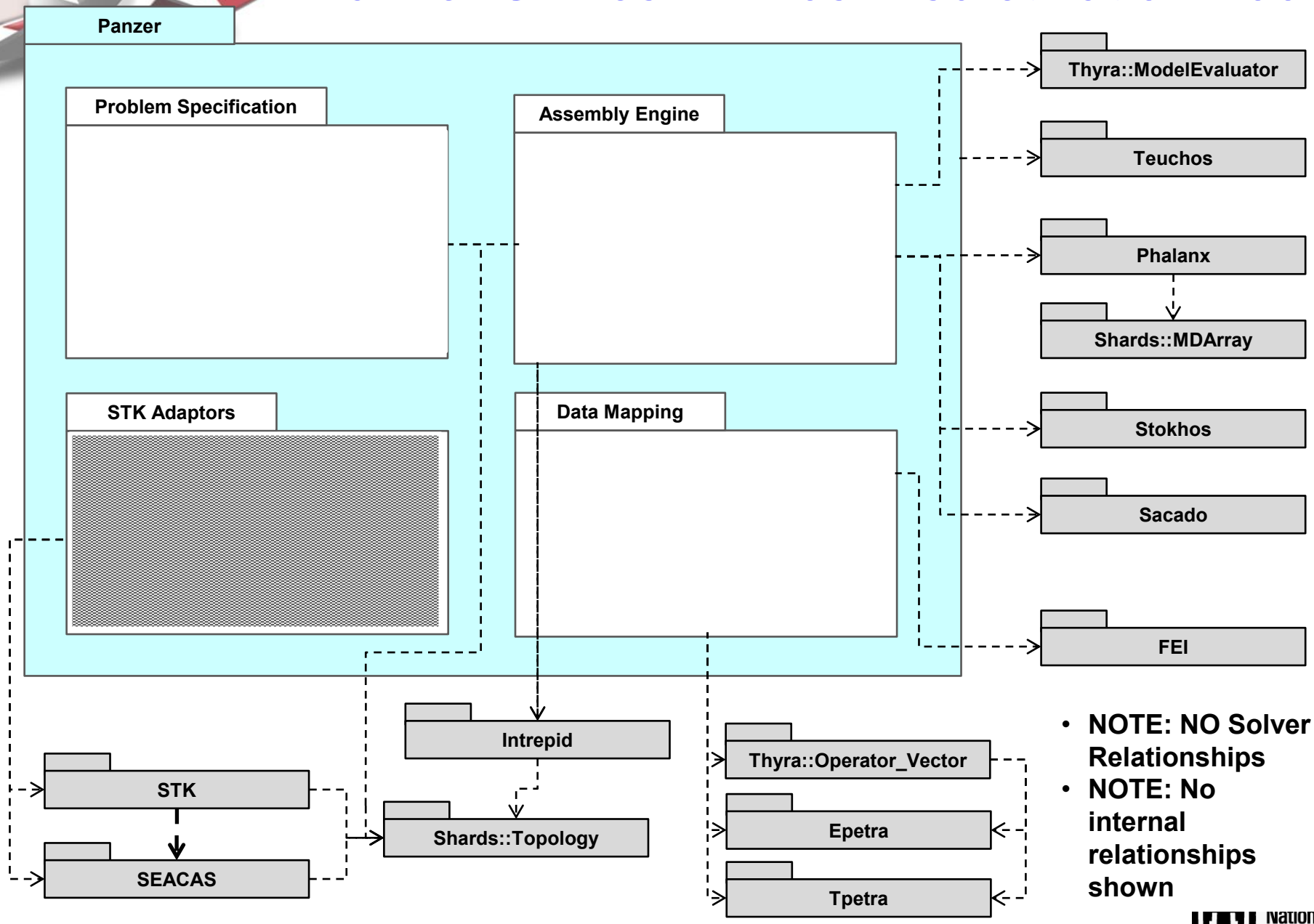
`DFad<PCE::OrthogPoly<double> >`

Sandia National Laboratories

Panzer Components

- Problem Description
  - Maps equations sets and boundary conditions into evaluators for Phalanx assembly
- Assembly Engine
  - A collection of Phalanx Field Managers to control assembly
  - Produces a Model Evaluator for User
- Data Mapping Utilities
  - DOF Manager for mapping field values into linear algebra
  - Connection Manager: Abstraction of Mesh
- STK Adaptors
  - Concrete implementation Panzer objects for using STK::Mesh and SEACAS for I/O
  - Specialized evaluators

Panzer Unifies Trilinos Discretization Tools



- NOTE: NO Solver Relationships
- NOTE: No internal relationships shown

National Laboratories

Data Mapping

Computes global unknown indices

1. Serves as interface to mesh
2. Allows Panzer to be mesh agnostic
3. Handles unknowns for mixed discretizations
4. Handles unknowns for multiphysics (multiple element blocks)
5. Uses FEI for producing unknowns

Composed of 3 primary pieces

1. FieldPattern – Describes the basis layout and continuity of fields
2. DOFManager – Manages and computes unknown numbers on fields
3. ConnManager – (User implemented) Mesh topology from field pattern

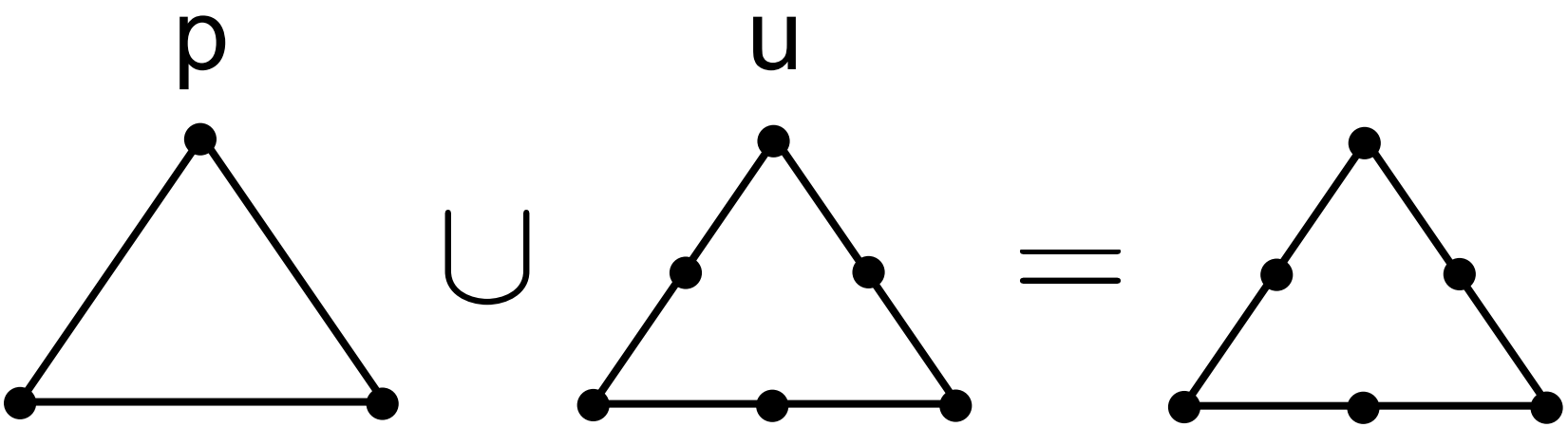
Features not implemented but supported by design

1. Higher order discretizations – geometric symmetries
2. Heterogeneous meshes – quadrilaterals and triangles

Data Mapping: Field Pattern

For stable Navier-Stokes pair:

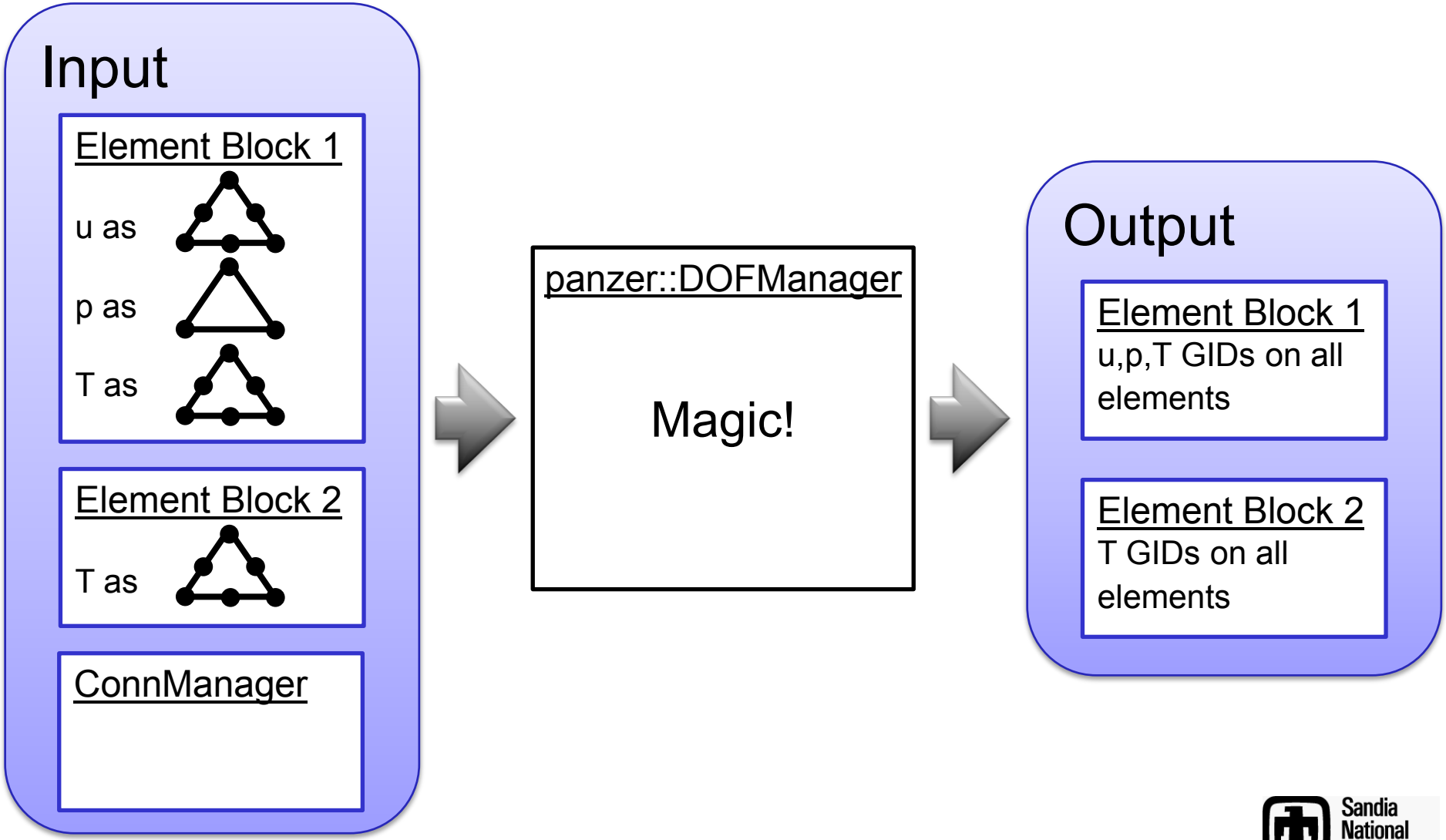
- Linear pressures
- Quadratic velocities



Field Pattern specifies basis layout

- Continuity across subcells (continuity of field)
- Unknowns on each element
- Communicates required topology

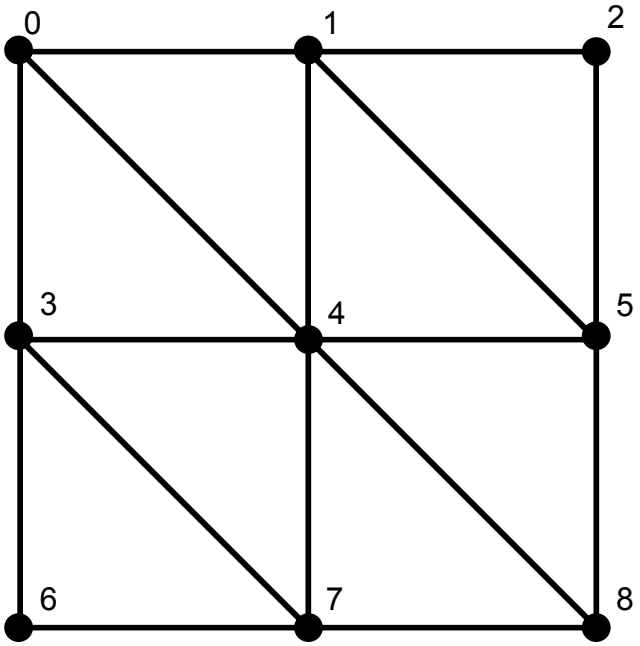
Data Mapping: DOFManager

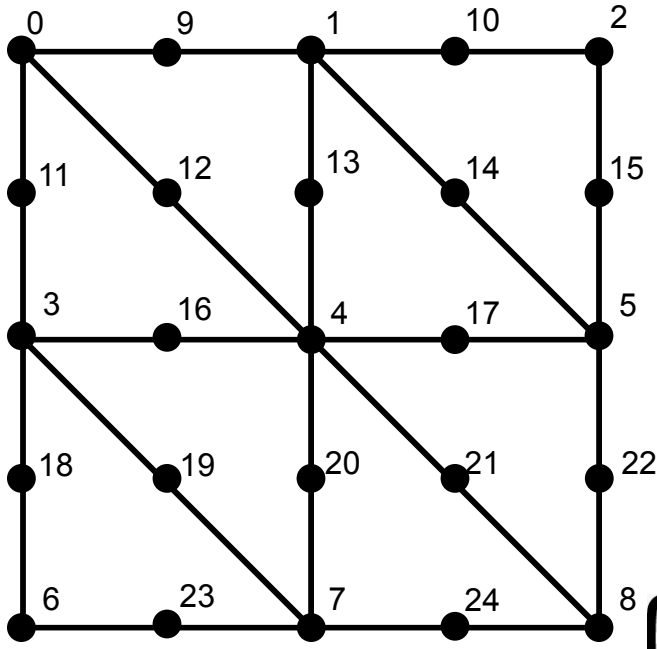


Data Mapping: ConnManager

Must generate mesh connectivity

- DOFManager passes in field pattern
- Provides unique global node, edge, volume ids for each element
- Uniform field pattern across all element blocks
  - Makes multiphysics easy

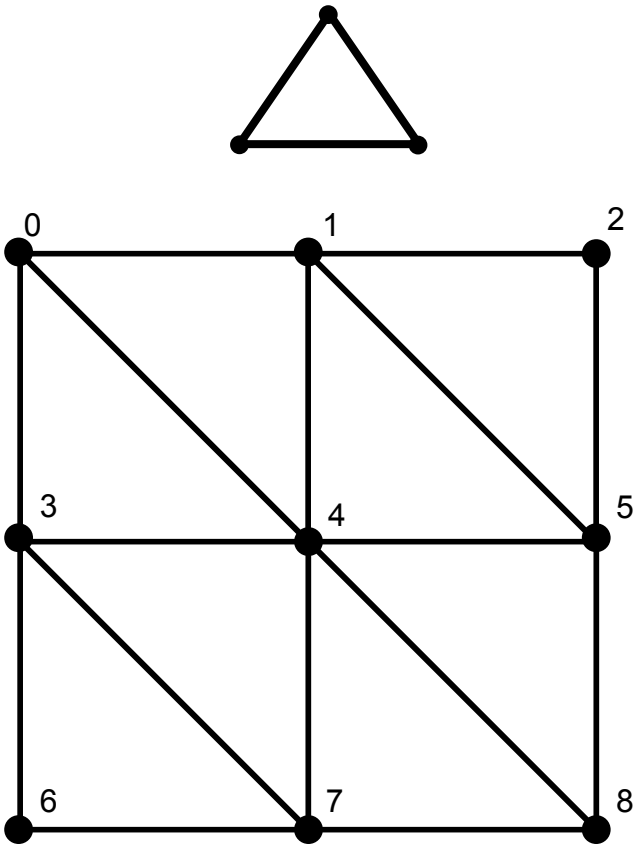




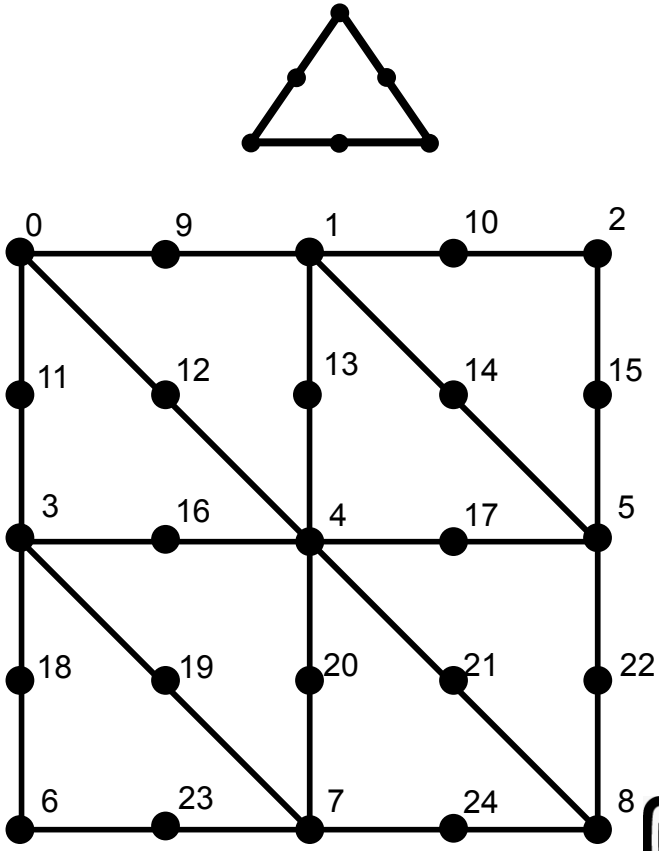
Sandia National Laboratories

Data Mapping: ConnManager

Piecewise linear p
Piecewise linear u



Piecewise linear p
Piecwise quadratic u



Sandia National Laboratories

The Future

- Stokhos integration (almost complete)
- Adjoint capability
- Use of Kokkos MDArray for multi-/many-core/GPGPU support
- Expression templates for MDFields