SAND2007-7201P

Intrepid

INteroperable Tools for Rapid dEveloPment of compatIble Discretizations

Pavel Bochev (1414) and Denis Ridzal (1411)
Sandia National Laboratories

Trilinos User Group Meeting

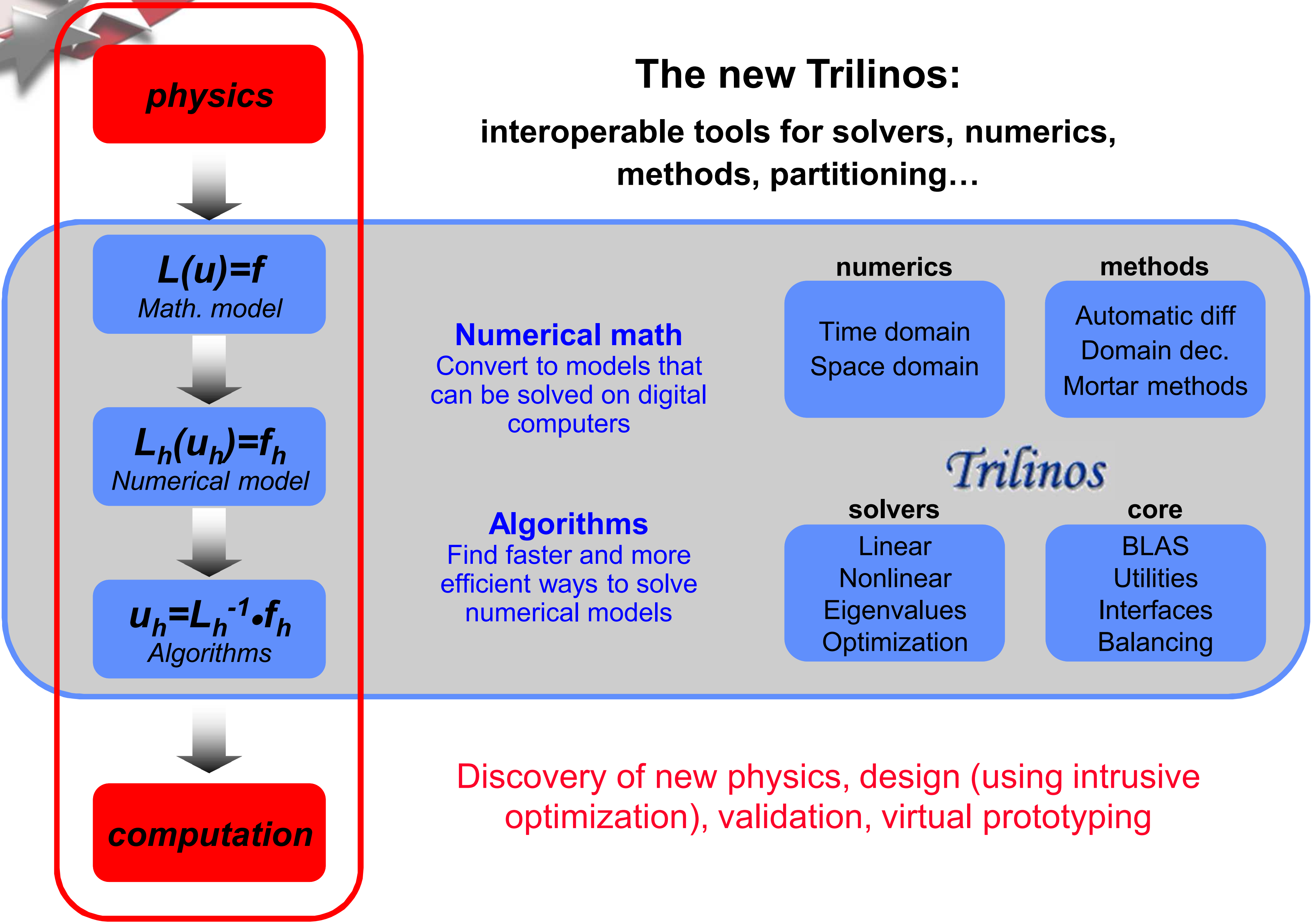
November 6, 2007

- David Day
- Ulrich Hetmaniuk
- Roger Pawlowski
- Misha Shashkov (LANL)
- Konstantin Lipnikov (LANL)
- Pavel Solin (UTEP)
- Pavel Kus (UTEP)
- Rob Kirby (Texas Tech)
- Allen Robinson
- Alan Schiemenz
- Eric Phipps

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

The new Trilinos:

interoperable tools for solvers, numerics, methods, partitioning…



physics

$L(u)=f$
Math. model

$L_h(u_h)=f_h$
Numerical model

$u_h=L_h^{-1}\bullet f_h$
Algorithms

computation

Numerical math
Convert to models that can be solved on digital computers

Algorithms
Find faster and more efficient ways to solve numerical models

numerics
Time domain
Space domain

methods
Automatic diff
Domain dec.
Mortar methods

Trilinos

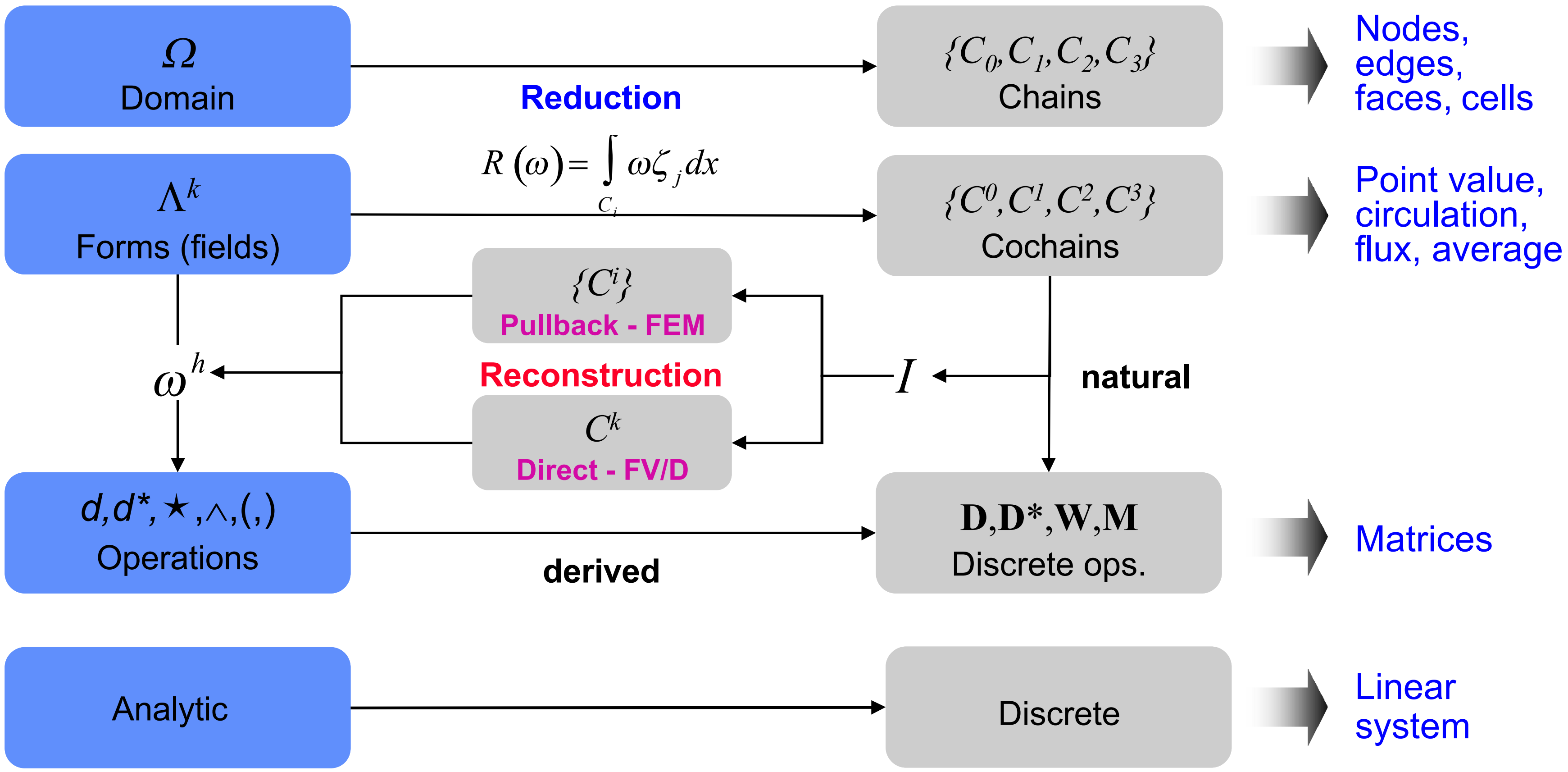
solvers
Linear
Nonlinear
Eigenvalues
Optimization

core
BLAS
Utilities
Interfaces
Balancing

Discovery of new physics, design (using intrusive optimization), validation, virtual prototyping

Mathematical Background

All discrete structures induced by **2 basic operations**



Ω
Domain

Reduction

$R(\omega) = \int_{C_j} \omega \zeta_j dx$

$\{C_0, C_1, C_2, C_3\}$
Chains

Nodes, edges, faces, cells

Λ^k
Forms (fields)

$\{C^0, C^1, C^2, C^3\}$
Cochains

Point value, circulation, flux, average

$\{C^i\}$
Pullback - FEM

ω^h

Reconstruction

I

natural

C^k
Direct - FV/D

$d, d^*, \star, \wedge, (,)$
Operations

derived

$\mathbf{D}, \mathbf{D}^*, \mathbf{W}, \mathbf{M}$
Discrete ops.

Matrices

Analytic

Discrete

Linear system

Software Design

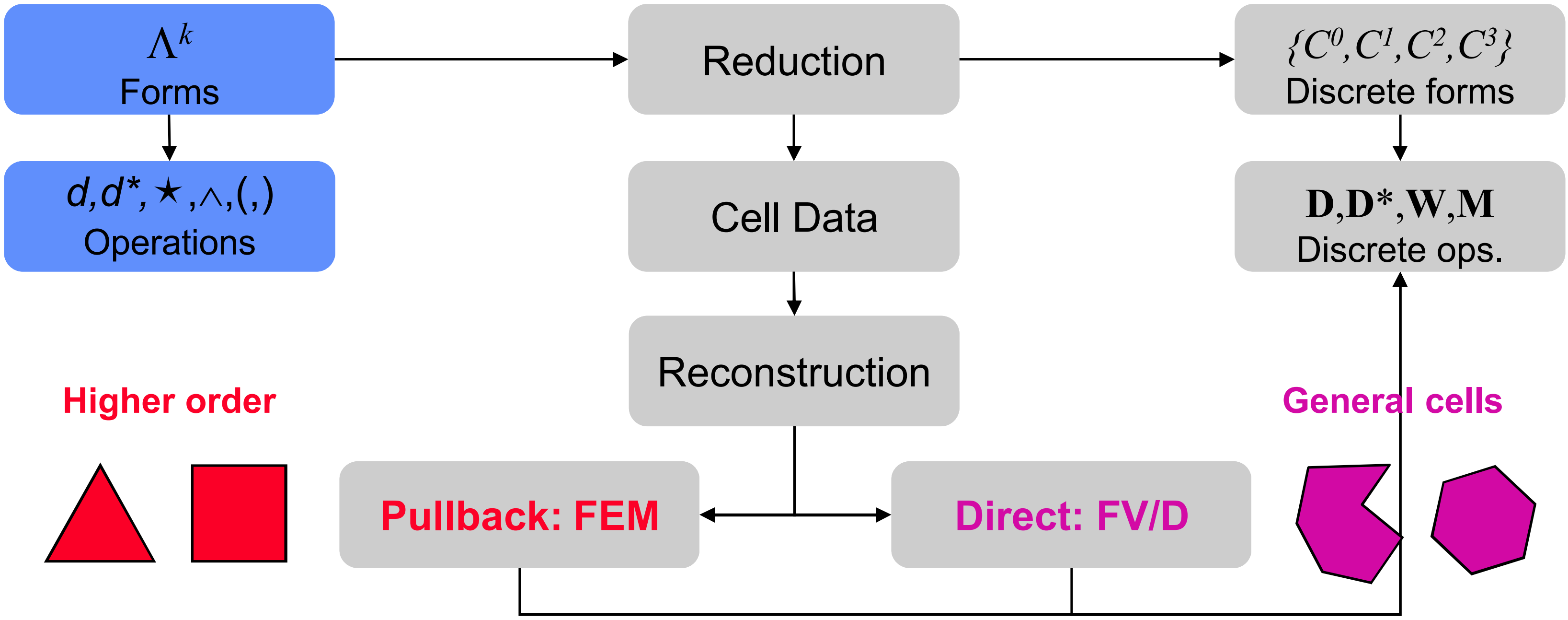
Intrepid promotes an **innovative software design** for compatible discretizations:

- allows access to FEM, FV and FD methods using a common API
- supports hybrid discretizations (FEM, FV and FD) on unstructured grids

FEM, **FV** and **FD methods** are defined by choosing a specific **reconstruction** operator $\mathcal{I}$:

Direct: $\mathcal{I}$ is **low order**, easily extendable to **arbitrary cells**

Pullback: $\mathcal{I}$ is defined on **standard cells,** easily extendable to **high orders**



Λ^k
Forms

$d,d^*,\star,\wedge,(,)$
Operations

Reduction

Cell Data

Reconstruction

$\{C^0,C^1,C^2,C^3\}$
Discrete forms

$\mathbf{D},\mathbf{D}^*,\mathbf{W},\mathbf{M}$
Discrete ops.

Higher order

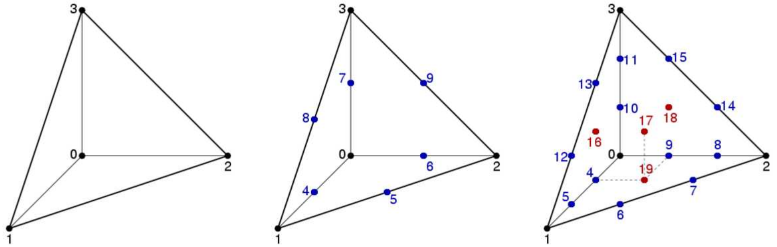
Pullback: FEM

Direct: FV/D

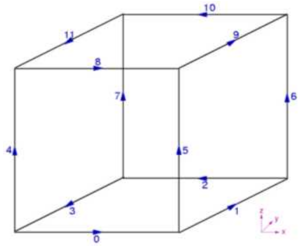
General cells

Status of Intrepid

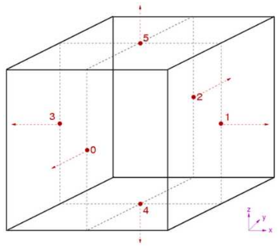
Completed development of **basic finite element** reconstruction operators (Bochev, Ridzal):



Lagrange elements of order 1,2,3

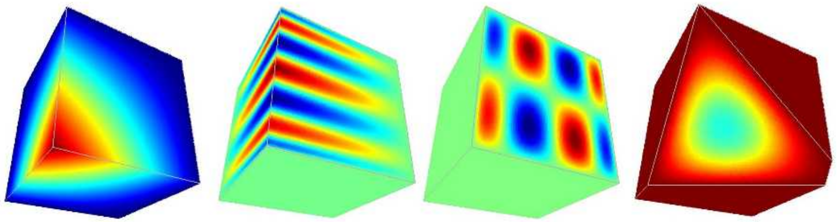


Nedelec element

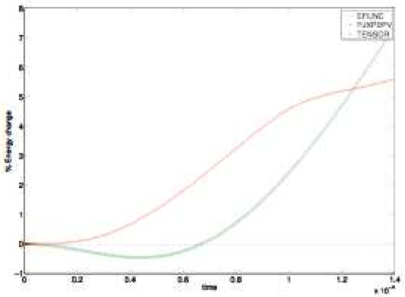


Raviart-Thomas element

Completed development of **hierarchic FE basis** functions (collaboration with Kus, Solin, UTEP)

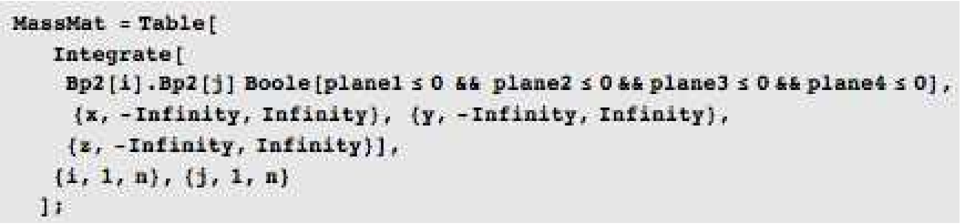


Demonstrated **Intrepid-Sacado interoperability** on prototype magnetic diffusion problem (work by Alan Schiemenz, SIP, and Allen Robinson)

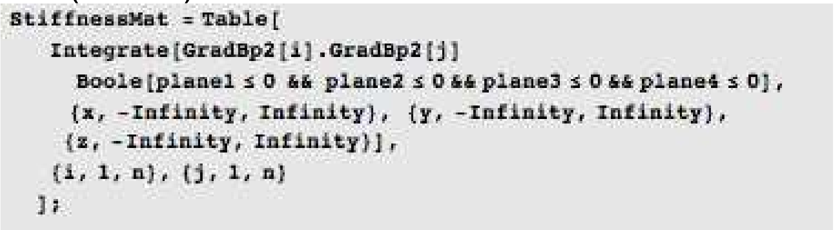


Provided and tested extensive collection of cubature rules on TET, TRI, HEX, QUAD (Day and Kus)

Developed extensive **unit test suite** using symbolic computation. (Ridzal)

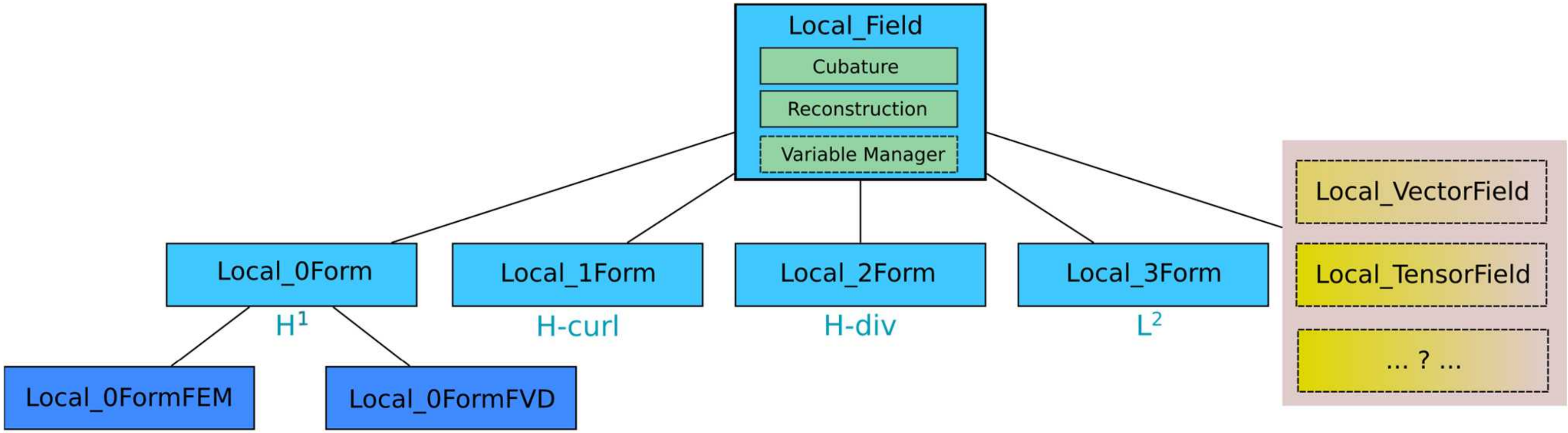


```
MassMat = Table[
   Integrate[
    Bp2[i].Bp2[j] Boole[plane1 ≤ 0 && plane2 ≤ 0 && plane3 ≤ 0 && plane4 ≤ 0],
     {x, -Infinity, Infinity}, {y, -Infinity, Infinity},
    {z, -Infinity, Infinity}],
   {i, 1, n}, {j, 1, n}
  ];
```



```
StiffnessMat = Table[
   Integrate[GradBp2[i].GradBp2[j]
     Boole[plane1 ≤ 0 && plane2 ≤ 0 && plane3 ≤ 0 && plane4 ≤ 0],
    {x, -Infinity, Infinity}, {y, -Infinity, Infinity},
    {z, -Infinity, Infinity}],
   {i, 1, n}, {j, 1, n}
  ];
```

Intrepid – Local Functionality



Local_Field

Cubature

Reconstruction

Variable Manager

Local_0Form

H^1

Local_1Form

H-curl

Local_2Form

H-div

Local_3Form

L^2

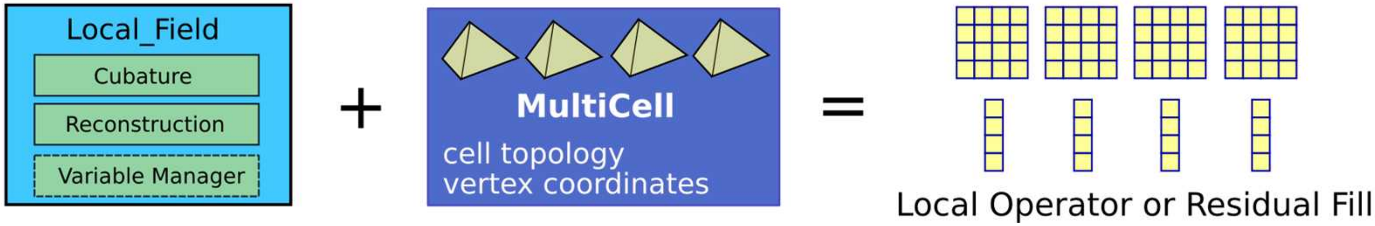
Local_0FormFEM

Local_0FormFVD

Local_VectorField

Local_TensorField

... ? ...



Local_Field

Cubature

Reconstruction

Variable Manager

+

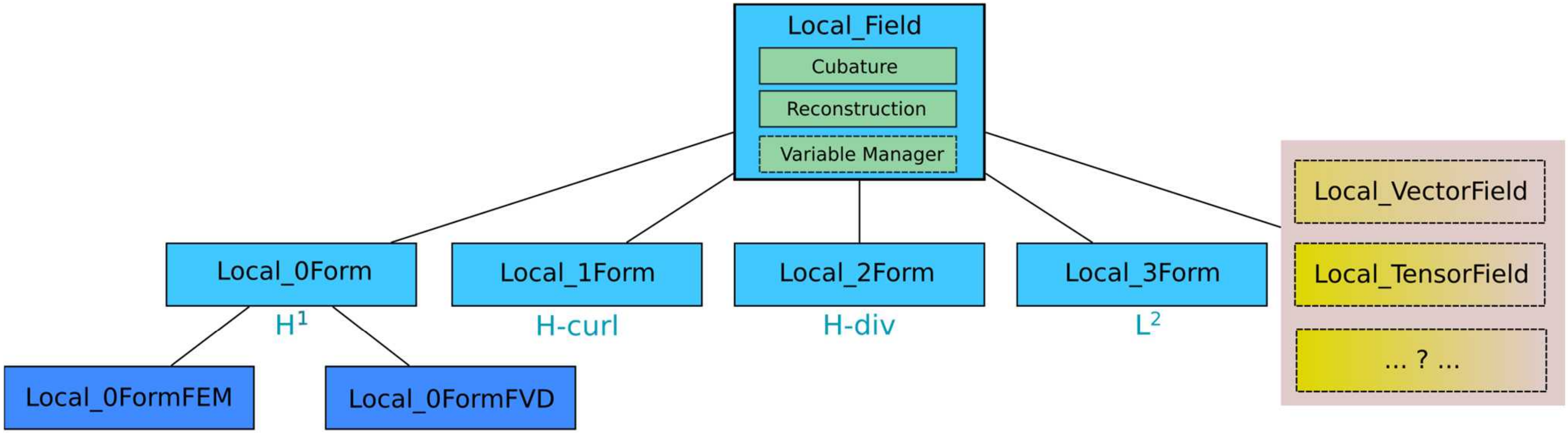
MultiCell

cell topology
vertex coordinates

=

Local Operator or Residual Fill

Intrepid – Local Functionality



Local_Field

Rec. Type / Flavor / Order

setNumDof()

setDofAllocation()

setCubatureType()

getOperator()
----
in: MultiCell, Var. Manager
out: array of dense matrices

getFunctional()

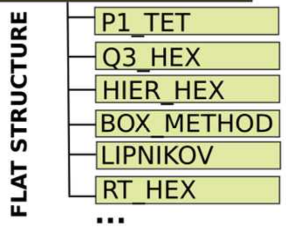
getResidual()

Cubature

collection of arrays of points and weights on various reference cells:

getPointsAndWeights()
----
in: Cubature Type,
Cell Type, Order (hp)
out: array of points and weights

Reconstruction



FEM: arrays of BF values and gradients on the reference cell

FVD: ?

computeRefValues()

reconstructField()

Variable Manager

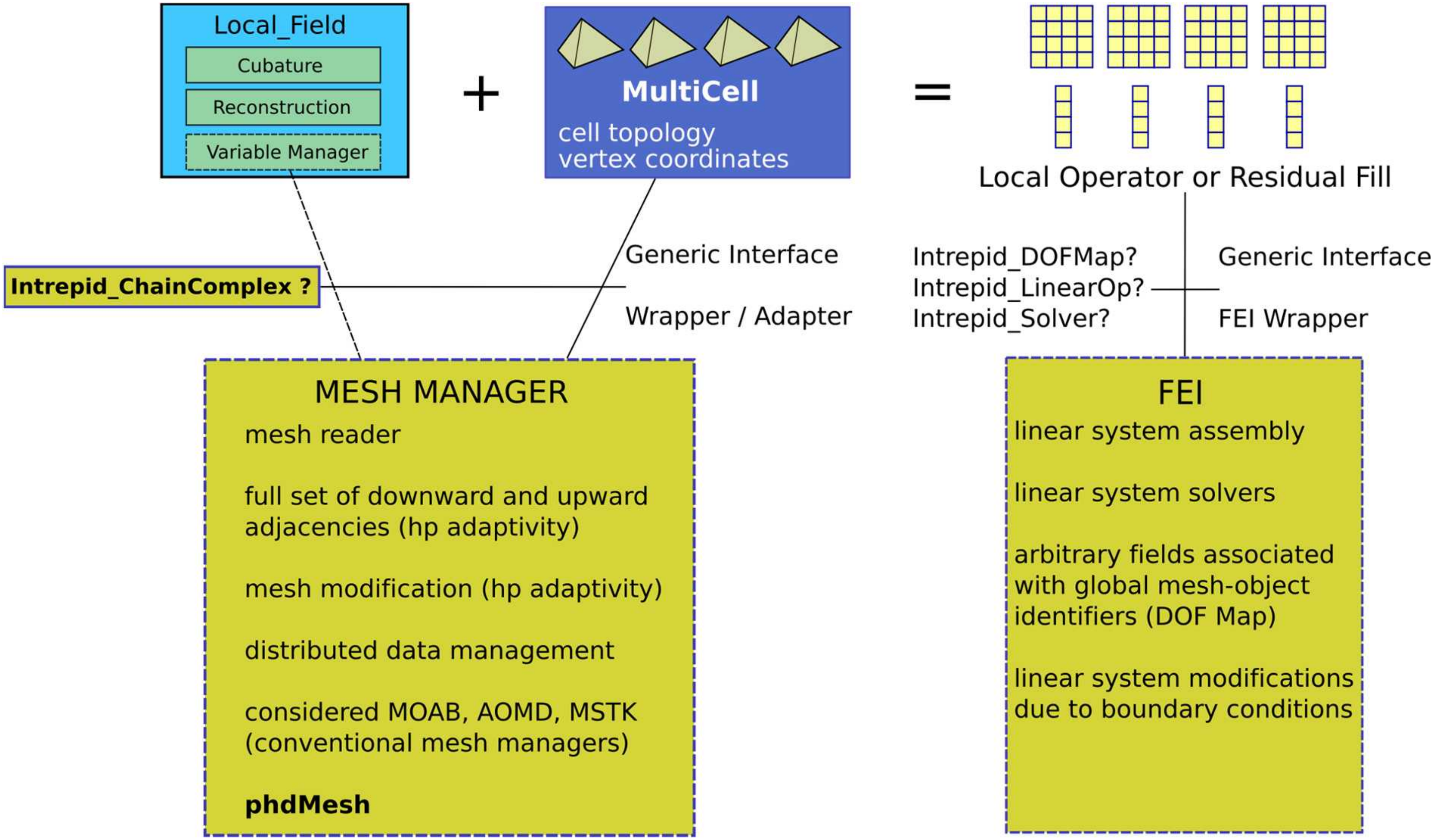
provides problem-dependent quantities, needed for complete field definitions, such as material properties, reaction rates, etc.

supports *LOCAL* fields

think: local, multicell-based PDE model

Sandia National Laboratories

Intrepid – Global Functionality



Local_Field

Cubature

Reconstruction

Variable Manager

+

MultiCell

cell topology
vertex coordinates

=

Local Operator or Residual Fill

Intrepid_ChainComplex ?

Generic Interface

Wrapper / Adapter

Intrepid_DOFMap?
Intrepid_LinearOp?
Intrepid_Solver?

Generic Interface

FEI Wrapper

MESH MANAGER

mesh reader

full set of downward and upward adjacencies (hp adaptivity)

mesh modification (hp adaptivity)

distributed data management

considered MOAB, AOMD, MSTK (conventional mesh managers)

phdMesh

FEI

linear system assembly

linear system solvers

arbitrary fields associated with global mesh-object identifiers (DOF Map)

linear system modifications due to boundary conditions

Plan for the First Release (Trilinos 9.0)

- **Refactor:** use transition to Trilinos as an opportunity to apply "lessons learned" and provide better OO design
  - Interface design to reflect a wider class of potential PDE apps: fluids, elasticity, conservation laws for which form encoding is not always natural
  - Basic functionality is the same, but wider range of "fields" required, e.g., vector, tensor fields, etc.
- **Add polyhedral cell functionality:**
  - Shashkov's group has given us access to their research code
- **Add hp-functionality for 1- and 2-forms:**
  - continue collaboration with Solin (UTEP)
- **Local Intrepid:**
  - develop "variable manager" or "PDE model" class (Pawlowski)
  - Incorporate "FIAT"-like solution to tabulate FE shape functions (R. Kirby)
- **Global Intrepid:** What is the best way to move forward?
  - MOAB (T. Tautges) or phdMesh (C. Edwards)
  - FEI (A. Williams)