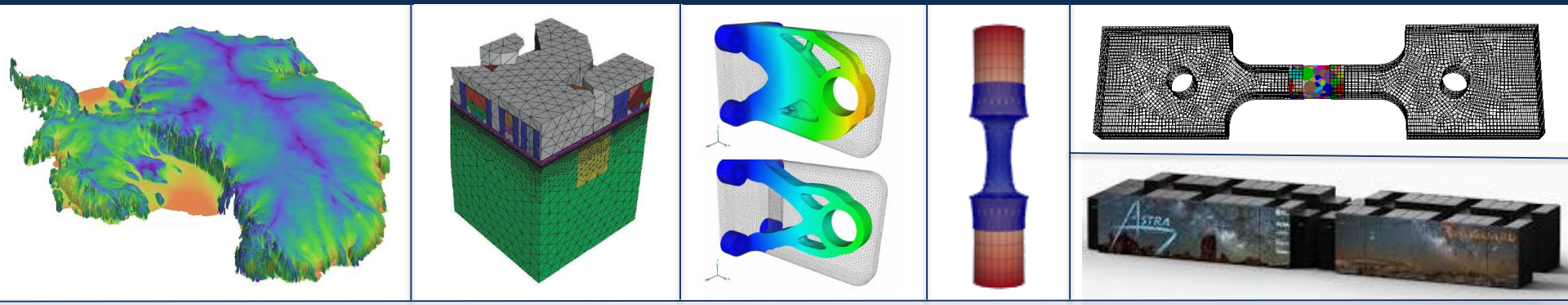
Sandia National Laboratories



# Albany : a Trilinos-based multi-physics partial differential equation research tool created using the AgileComponents code development strategy

**Irina Tezaur**

Quantitative Modeling & Analysis Dept., Sandia National Laboratories, Livermore, CA.

SIAM CSE 2019 Spokane, WA Feb. 25 - Mar. 1 2019

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and 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.

SAND2019-1842 C

# Acknowledgements

**“Father” of Albany, early advocate for AgileComponents strategy:**

- Andy Salinger [SNL]

**Albany contributors (58, from github):**

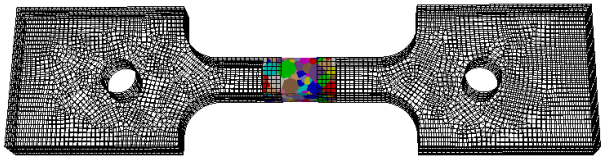
- I. Tezaur [SNL]
- A. Mota [SNL]
- D. Ibanez [RPI/SNL]
- G. Hansen [SNL]
- M. Perego [FSU/SNL]
- L. Bertagna [FSU/SNL]
- A. Bradley [SNL]
- J. Ostien [SNL]
- B. Granzow [RPI/SNL]
- A. Salinger [SNL]
- D. Littlewood [SNL]
- J. Foulk [SNL]
- C. Alleman [SNL]
- J. Overfelt [SNL]
- E. Nielsen [SNL]
- I. Demeshko [LANL]
- O. Guba [SNL]
- M. Juha [U de La Sabana]
- E. Phipps [SNL]
- J. Frederick [SNL]
- J. Watkins [SNL]
- J. Robbins [SNL]
- T. Smith [SNL]
- R. Jones [SNL]
- B. Spotz [SNL]
- S. Besu [RPI]
- J. Foucar [SNL]
- Q. Chen [Clemson]
- P. Lindsay [SNL]
- J. Fike [SNL]
- S. Gao [SNL]
- J. Redhorse [SNL]
- S. Bova [SNL]
- P. Bosler [SNL]
- H. Yuan [Tokyo]
- C. Siefert [SNL]
- T. Voth [SNL]
- T. Wildey [SNL]
- C. Smith [RPI]
- J. Clough [USC]
- W. Sun [Columbia]
- Z. Wang [USC]
- T. Fuller [SNL]
- A. Vacanti [Kitware]
- R. Tuminaro [SNL]
- M. Parks [SNL]
- J. Robbins [SNL]
- M. Hoffman [LANL]
- R. Pawlowski [SNL]
- M. Bloomfield [RPI]
- G. Phlipot [CalTech]
- B. Perschbacher [SNL]
- J. Willenbring [SNL]
- …

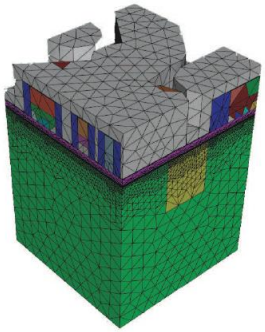
# Outline

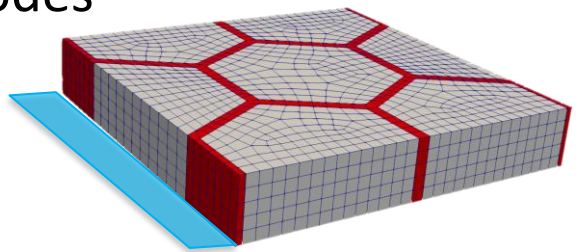
Sandia National Laboratories

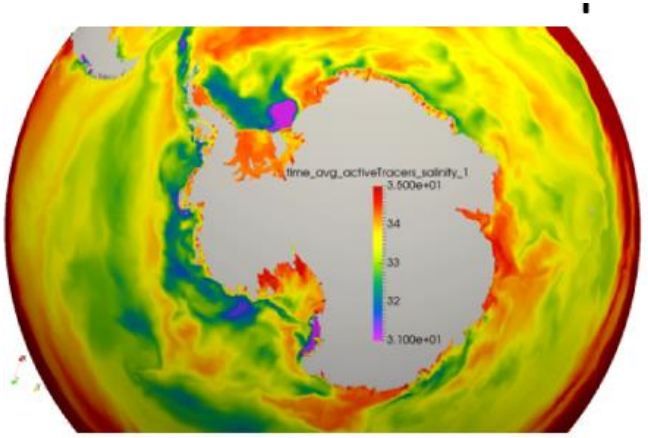
Albany

1. AgileComponents code-development strategy
2. What is Albany?
3. Albany code design
   - Global discretization & libraries
   - Problem abstraction & finite element assembly
   - Nonlinear model abstraction & libraries
   - Linear model abstraction & libraries
   - Software quality tools
4. Applications hosted by Albany
5. Algorithmic projects hosted by Albany
6. Coupling with other codes
7. Summary









# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.

- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.


- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

| | | |
|---|---|---|
| "**Components**" = | ☑ Libraries | ☑ Software Quality Tools |
| | ☑ Interfaces | ☑ Demonstration Applications |

# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.


- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

| "**Components**" = | ☑ Libraries | ☑ Software Quality Tools |
|---|---|---|
| | ☑ Interfaces | ☑ Demonstration Applications |

- ***Business strategy:***

# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.


- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

| "**Components**" = | ☑ Libraries | ☑ Software Quality Tools |
|---|---|---|
| | ☑ Interfaces | ☑ Demonstration Applications |

- ***Business strategy:***

Trilinos, SIERRA, DAKOTA, PUMI

Base of Software Components

# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.


- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

"**Components**" = ☑ Libraries ☑ Software Quality Tools ☑ Interfaces ☑ Demonstration Applications

- ***Business strategy:***

ASC, National Nuclear Security Administration, Office of Science U.S. Department of Energy, SciDAC Scientific Discovery through Advanced Computing, LDRD, ECP Exascale Computing Project

Trilinos, SIERRA, DAKOTA, PUMI Parallel Unstructured Mesh Infrastructure

Base of Software Components

Projects

# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.


- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

"**Components**" = ☑ Libraries ☑ Software Quality Tools ☑ Interfaces ☑ Demonstration Applications

- ***Business strategy:***

Trilinos SIERRA DAKOTA PUMI

Base of Software Components

Leverage the base →

ASC NNSA National Nuclear Security Administration, Office of Science U.S. Department of Energy, SciDAC Scientific Discovery through Advanced Computing, LDRD, ECP Exascale Computing Project

Projects

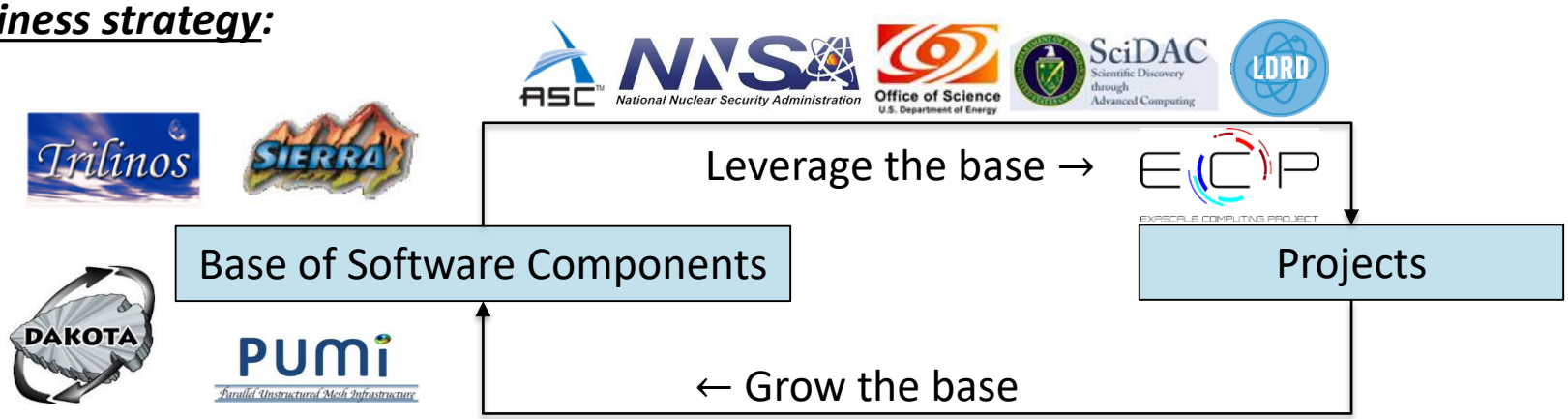
# AgileComponents: a PDE code strategy

***Strategic Goal:*** To enable the **rapid** development of new **production** codes embedded with **transformational** capabilities.


- ***Technical strategy:*** projects create, use, and improve a common base of modular, independent-yet-interoperable, software **components**
  - **2012 white paper by A. Salinger**: "Component-Based Scientific Application Development" (right).

"**Components**" = ☑ Libraries ☑ Software Quality Tools ☑ Interfaces ☑ Demonstration Applications

- ***Business strategy:***



# What is Albany? (high-level description)

Sandia National Laboratories

"**Components**" =

- ☑ Libraries
- ☑ Software Quality Tools
- ☑ Interfaces
- ☑ **Demonstration Applications**

2008

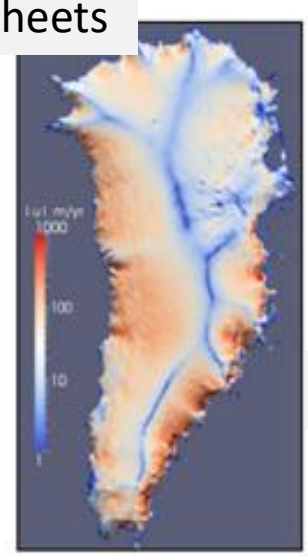
# What is Albany? (high-level description)

Sandia National Laboratories

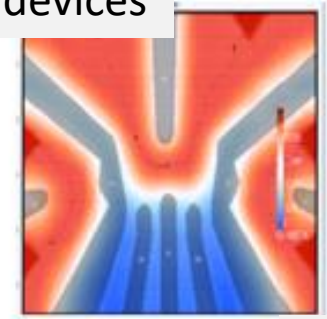
***Albany***: open-source*, parallel, C++, unstructured-grid, mostly-implicit multi-physics finite element code that demonstrates **AgileComponents** vision.

Albany houses a variety of **diverse algorithmic projects** and **applications**:

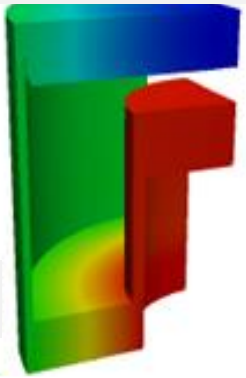
Ice sheets



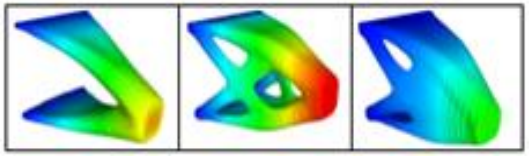
Quantum devices



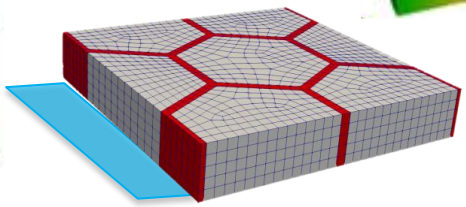
Computational mechanics



Additive manufacturing



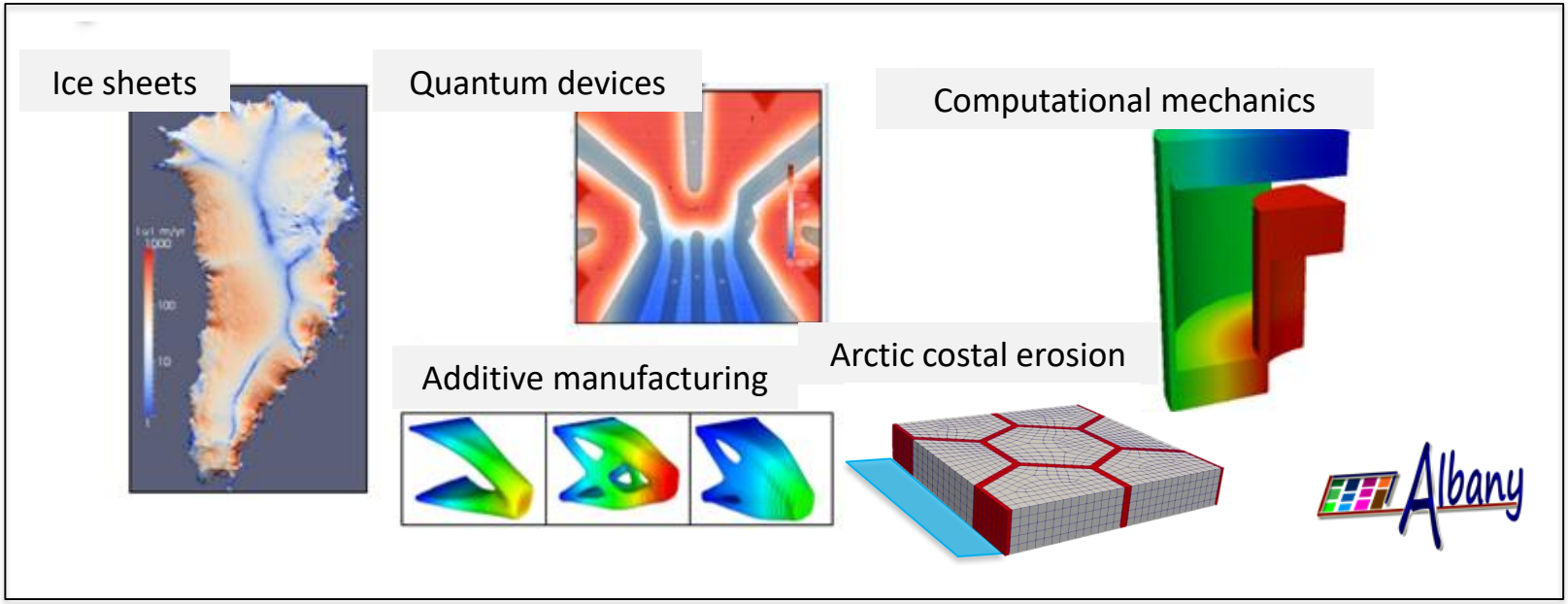
Arctic costal erosion



Albany

* **Albany github repo:** https://github.com/SNLComputation/Albany.

# What is Albany? (high-level description)




***Distinguishing features of Albany:***

- **Funded** entirely **by applications** residing within.
- Both a **"sand-box"** for prototyping **new approaches** and a **production** code.
- **Algorithms/software** are developed/matured directly on **applications**.
- Applications are "**born**" scalable, fast, robust, and...
- Equipped with **embedded advanced analysis capabilities**: sensitivities, bifurcation analysis, adjoint-based inversion, *embedded UQ*, model reduction*.*

* Italicized capabilities are in feature branches/tags.

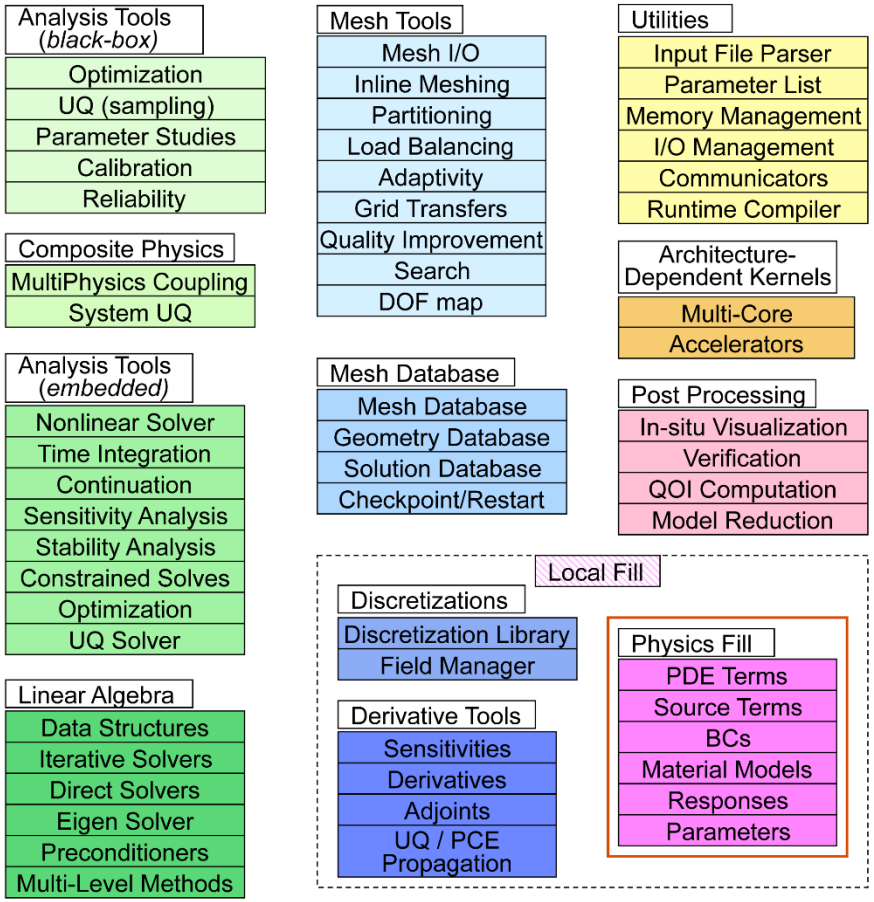
# The components effort: libraries & tools

Sandia National Laboratories

***Components in Albany*** = cutting-edge technology from Trilinos, SCOREC, SierraToolKit, DAKOTA, FASTMath, QUEST, Kitware, etc.

Many components are **Trilinos* packages**:

- Distributed linear algebra (*Tpetra*)
- Mesh tools (*STK*)
- Discretization tools (*Intrepid2*)
- Nonlinear solver (*NOX*)
- Linear solver (*Belos*)
- Preconditioners (*Ifpack2*)
- Automatic differentiation (*Sacado*)
- Shared memory parallelism (*Kokkos*)
- Optimization (*ROL*)
- *Many more…*



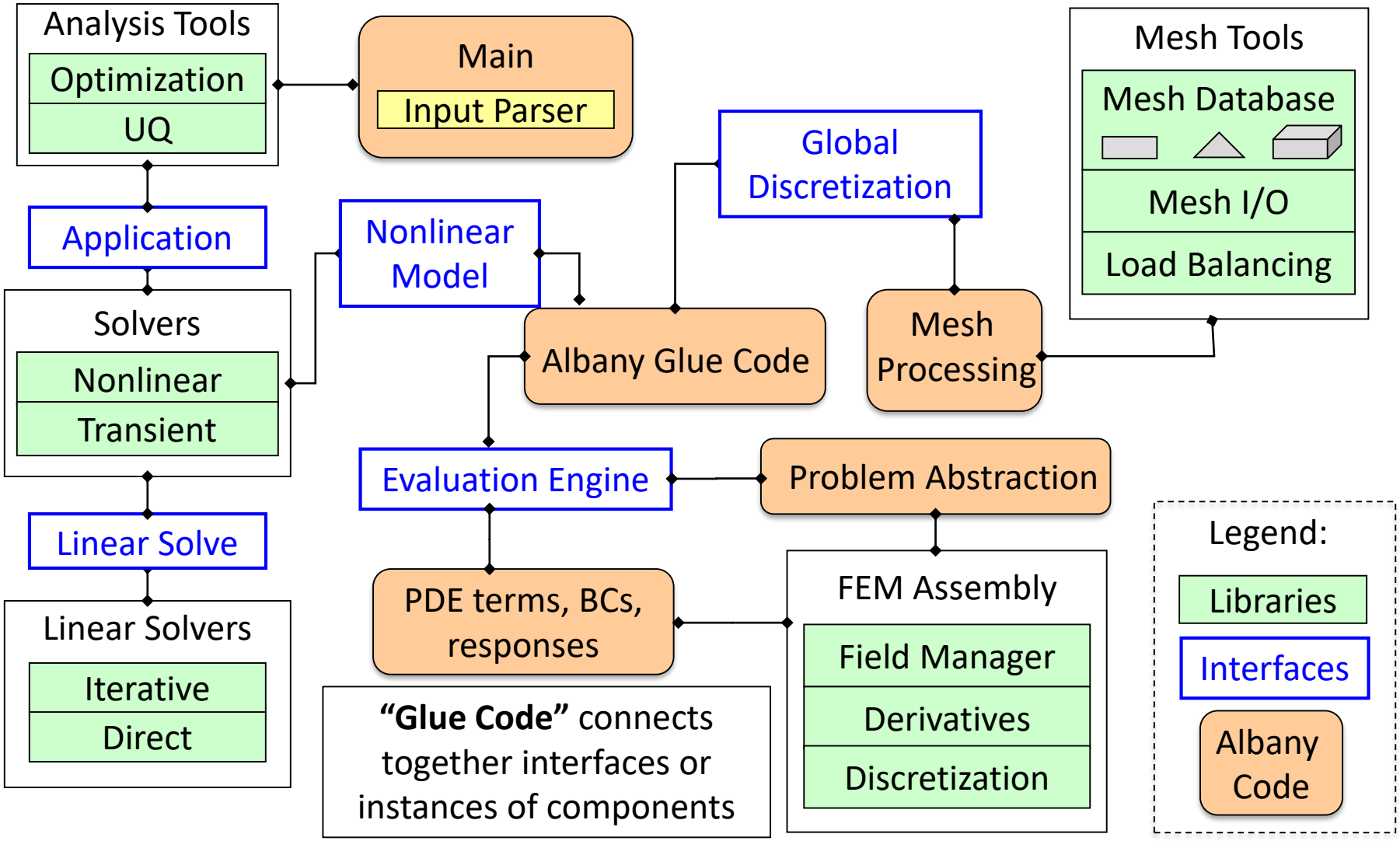

Trilinos : 40+ packages; 120+ libraries

* **Trilinos github repo:** https://github.com/trilinos/Trilinos.

# What is Albany? (under-the-hood)

Sandia National Laboratories

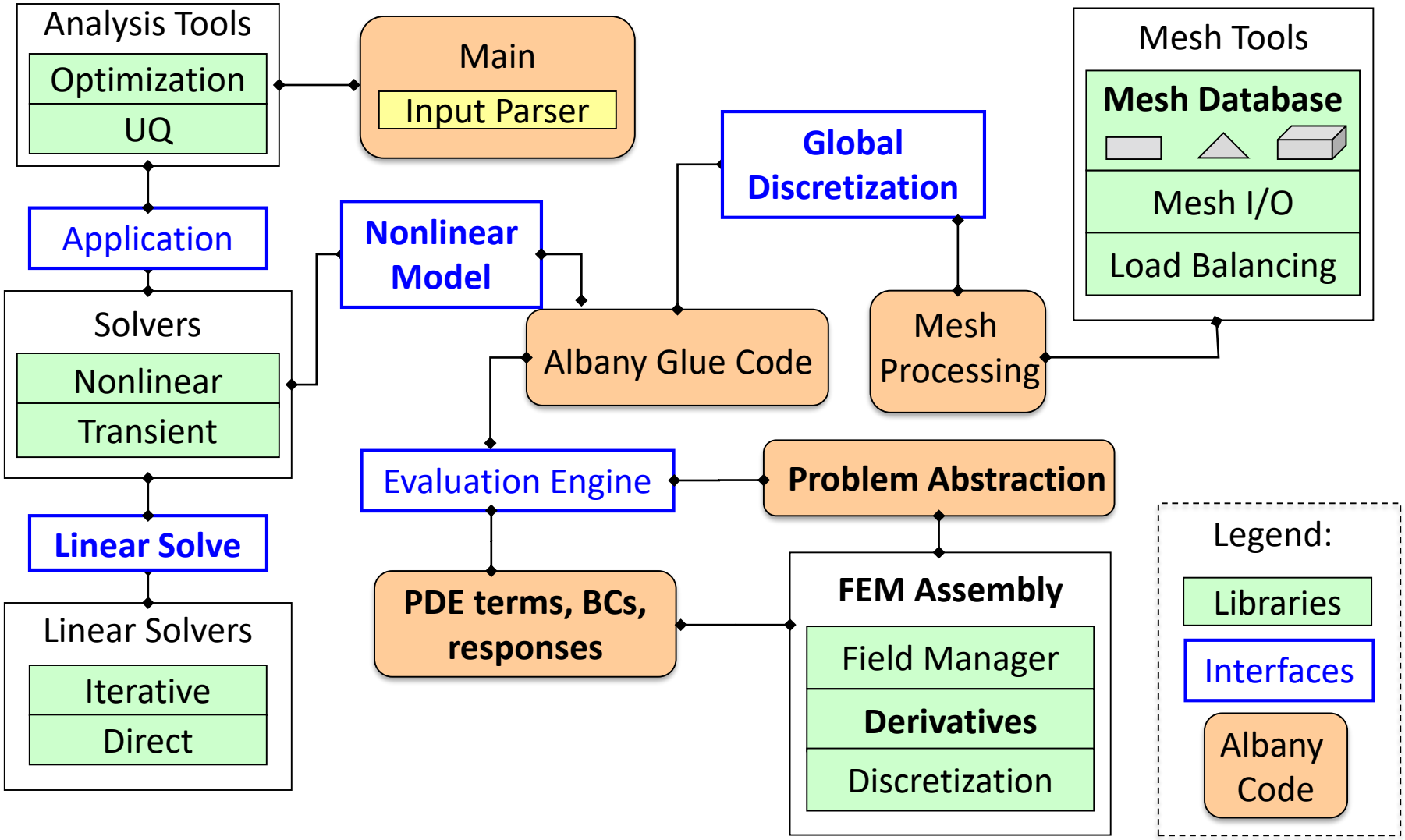
**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# What is Albany? (under-the-hood)

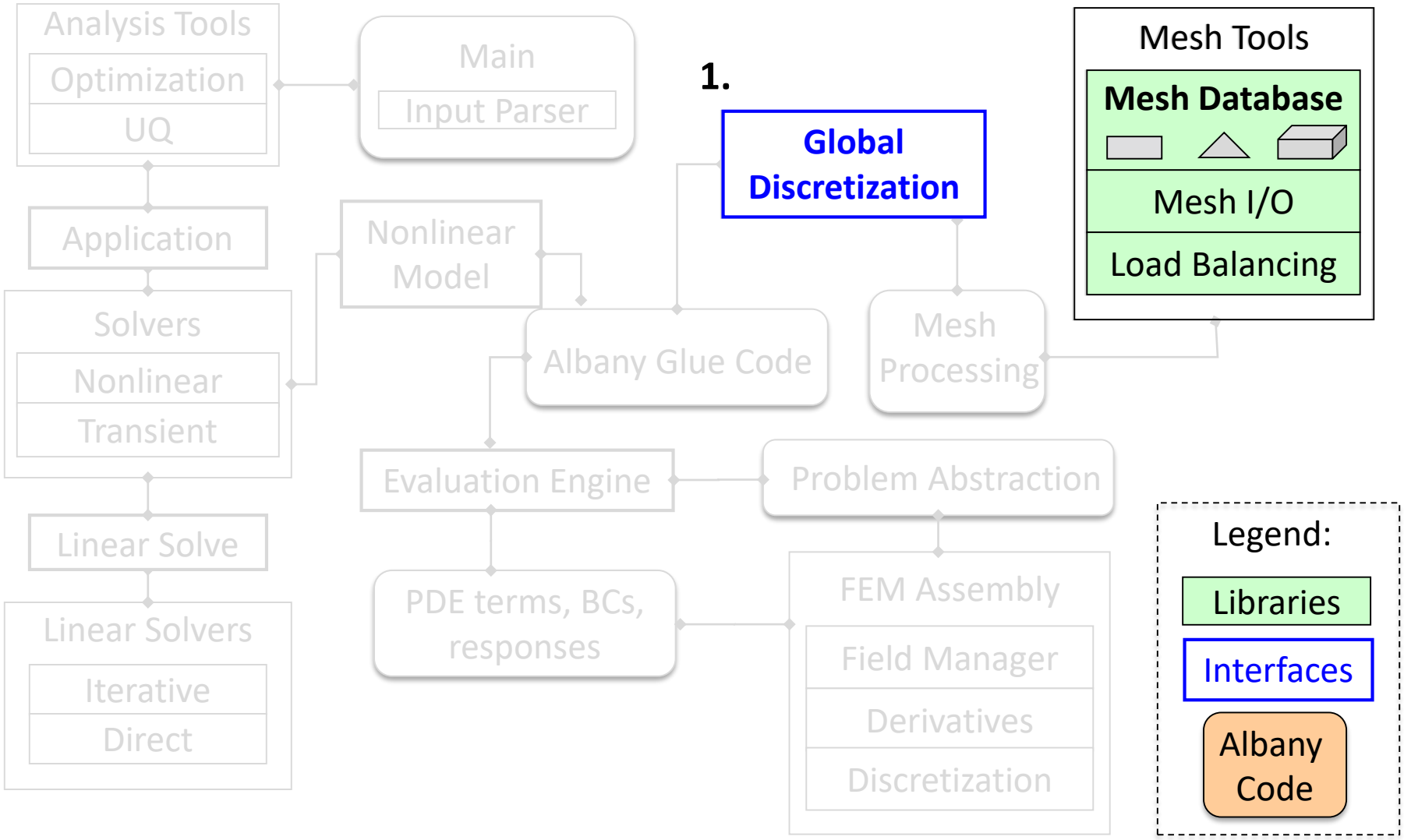
Sandia National Laboratories

**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# What is Albany? (under-the-hood)

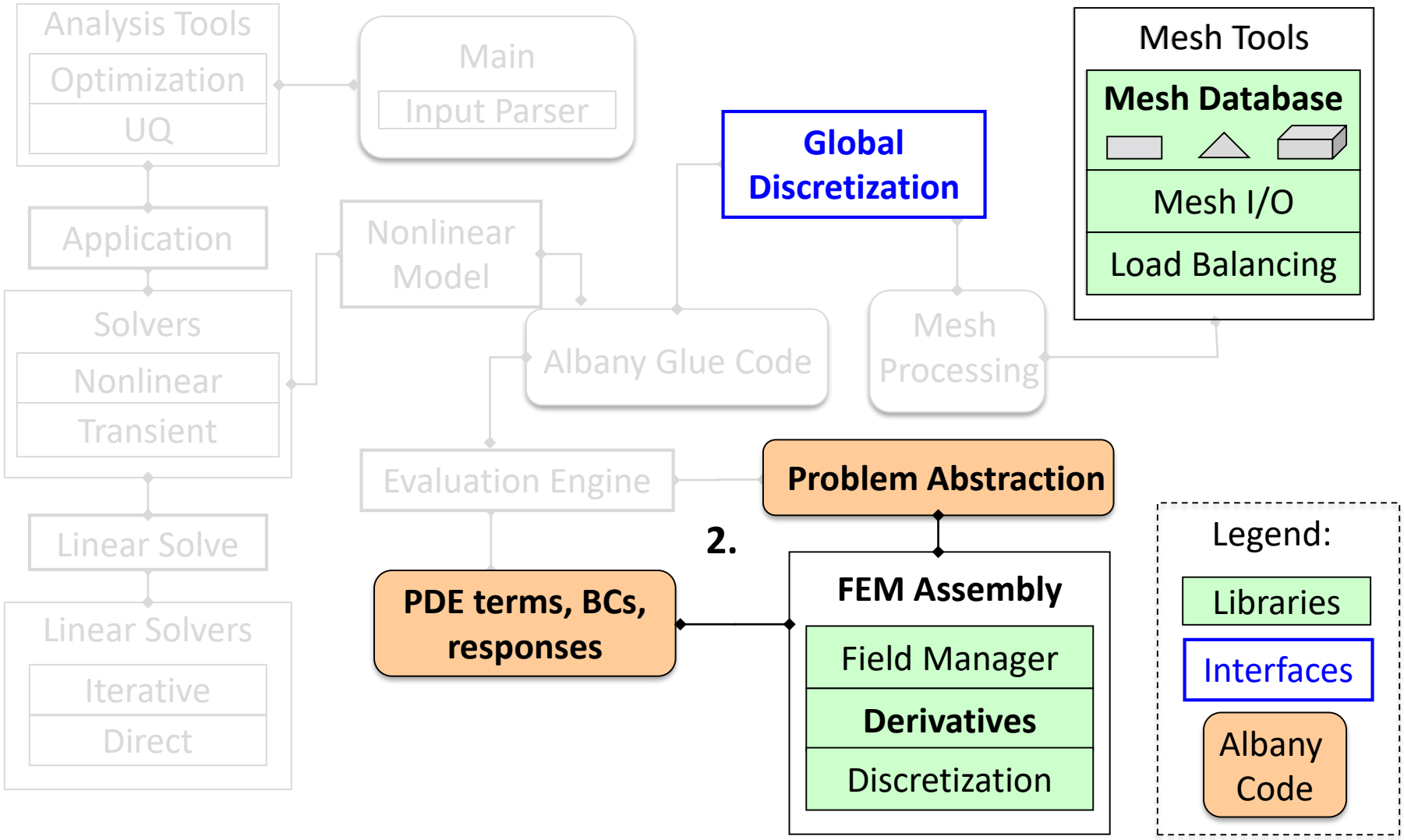
**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# What is Albany? (under-the-hood)

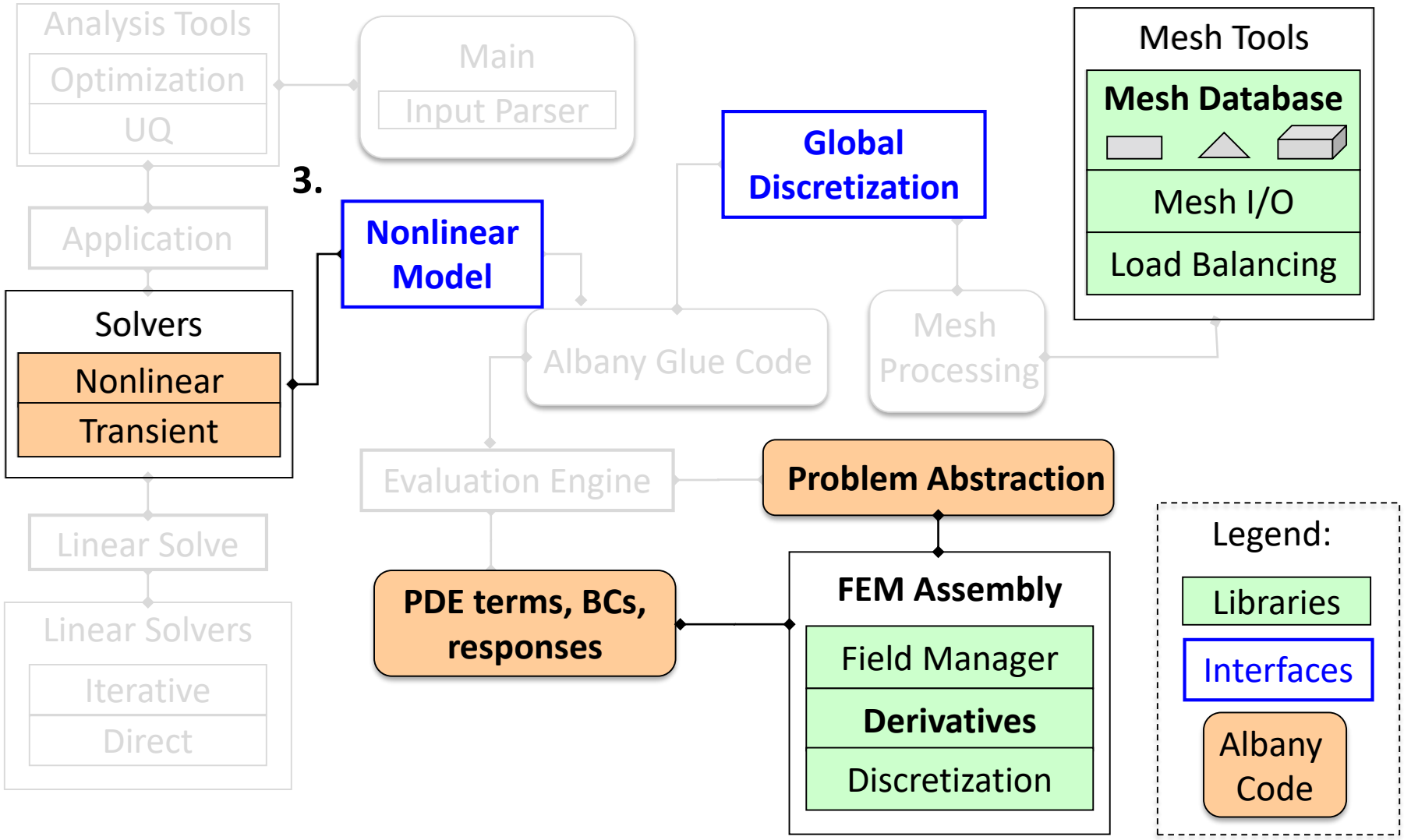
Sandia National Laboratories

**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# What is Albany? (under-the-hood)

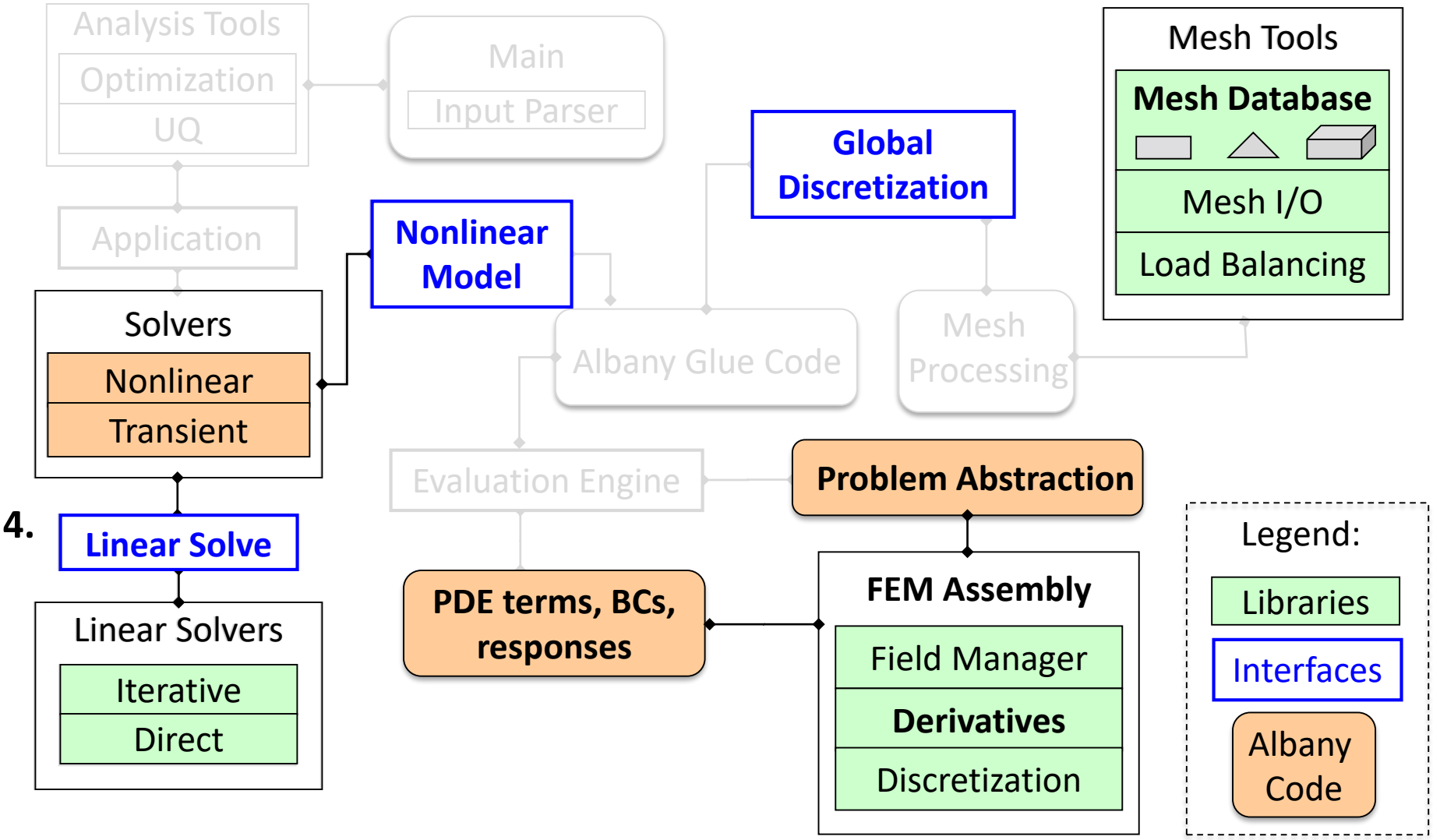
**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# What is Albany? (under-the-hood)

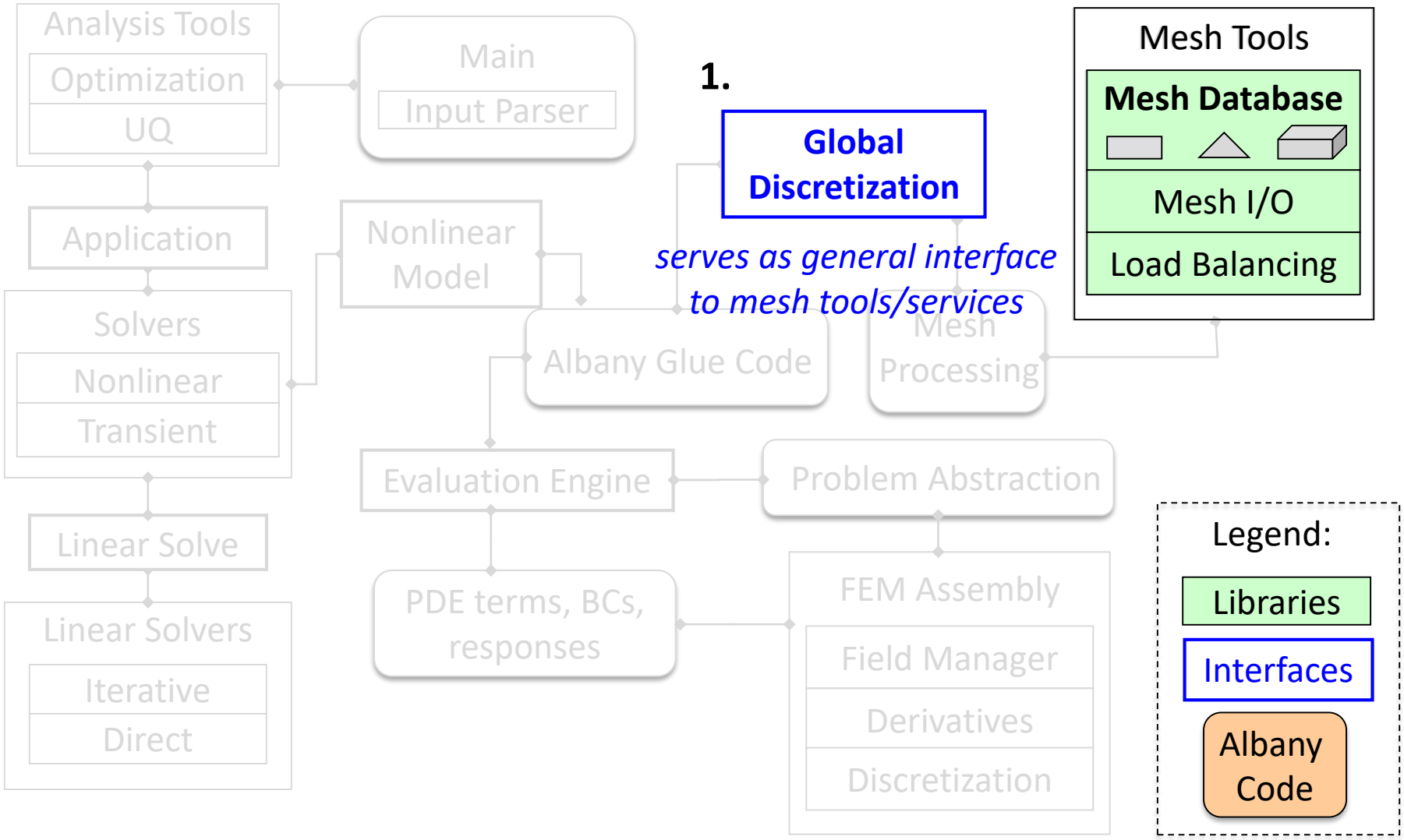
Sandia National Laboratories

**Albany** = Component Libraries + Abstract Interfaces + “Glue Code”



# What is Albany? (under-the-hood)

**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"

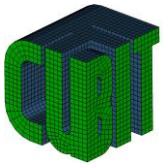


# Global discretization abstraction & libraries

Sandia National Laboratories

**Discretization interface:** currently has ***two independent implementations***

1. **SierraToolKit (STK)** package in Trilinos.
   - Supports reading in **Exodus** mesh files (e.g., from CUBIT), **inline meshing** via Pamgen, **simple rectangular meshes** constructed in Albany.
   - Meshes can be **structured/unstructured** but are **static**.
2. **Parallel Unstructured Mesh Infrastructure (PUMI)** package, developed at the Scientific Computation Research Center (SCOREC) at RPI.
   - Supports **VTK mesh files** (generated by Symmetrix).
   - Goal-oriented generalized **error estimation** and **in-memory mesh adaptation**.






S. Seol *et. al.* PUMI, *High Performance Computing, Networking, Storage and Analysis (SCC), 2012.*

# Global discretization abstraction & libraries

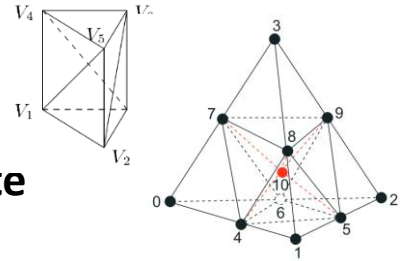
Sandia National Laboratories

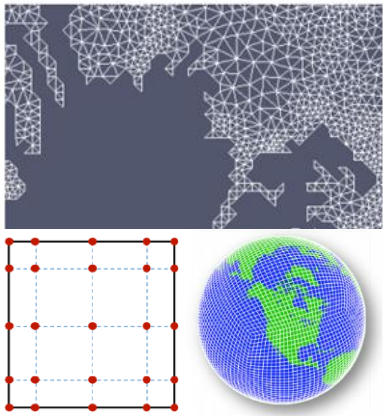
**Discretization interface:** currently has ***two independent implementations***

1. **SierraToolKit (STK)** package in Trilinos.
   - Supports reading in **Exodus** mesh files (e.g., from CUBIT), **inline meshing** via Pamgen, **simple rectangular meshes** constructed in Albany.
   - Meshes can be **structured/unstructured** but are **static**.
2. **Parallel Unstructured Mesh Infrastructure (PUMI)** package, developed at the Scientific Computation Research Center (SCOREC) at RPI.
   - Supports **VTK mesh files** (generated by Symmetrix).
   - Goal-oriented generalized **error estimation** and **in-memory mesh adaptation**.

**Element types**: variety of element types supported, with basis functions/quadrature routines from ***Intrepid2*** Trilinos library:

- **Isoparametric** elements (tet, hex, wedge, …).
- 2D **spectral elements** of arbitrary orders.
- Some physics-specific elements, e.g., **composite 10-node tetrahedron** for solid mechanics.



# Global discretization abstraction & libraries

Sandia National Laboratories

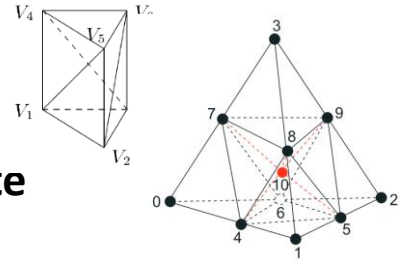
**Discretization interface:** currently has ***two independent implementations***

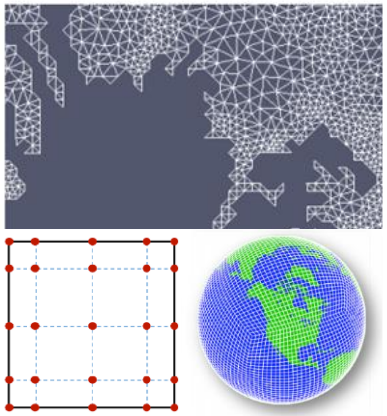
1. **SierraToolKit (STK)** package in Trilinos.
   - Supports reading in **Exodus** mesh files (e.g., from CUBIT), **inline meshing** via Pamgen, **simple rectangular meshes** constructed in Albany.
   - Meshes can be **structured/unstructured** but are **static**.
2. **Parallel Unstructured Mesh Infrastructure (PUMI)** package, developed at the Scientific Computation Research Center (SCOREC) at RPI.
   - Supports **VTK mesh files** (generated by Symmetrix).
   - Goal-oriented generalized **error estimation** and **in-memory mesh adaptation**.

Albany is a **Continuous Galerkin (CG)** unstructured grid finite element code.

**Element types**: variety of element types supported, with basis functions/quadrature routines from ***Intrepid2*** Trilinos library:

- **Isoparametric** elements (tet, hex, wedge, …).
- 2D **spectral elements** of arbitrary orders.
- Some physics-specific elements, e.g., **composite 10-node tetrahedron** for solid mechanics.

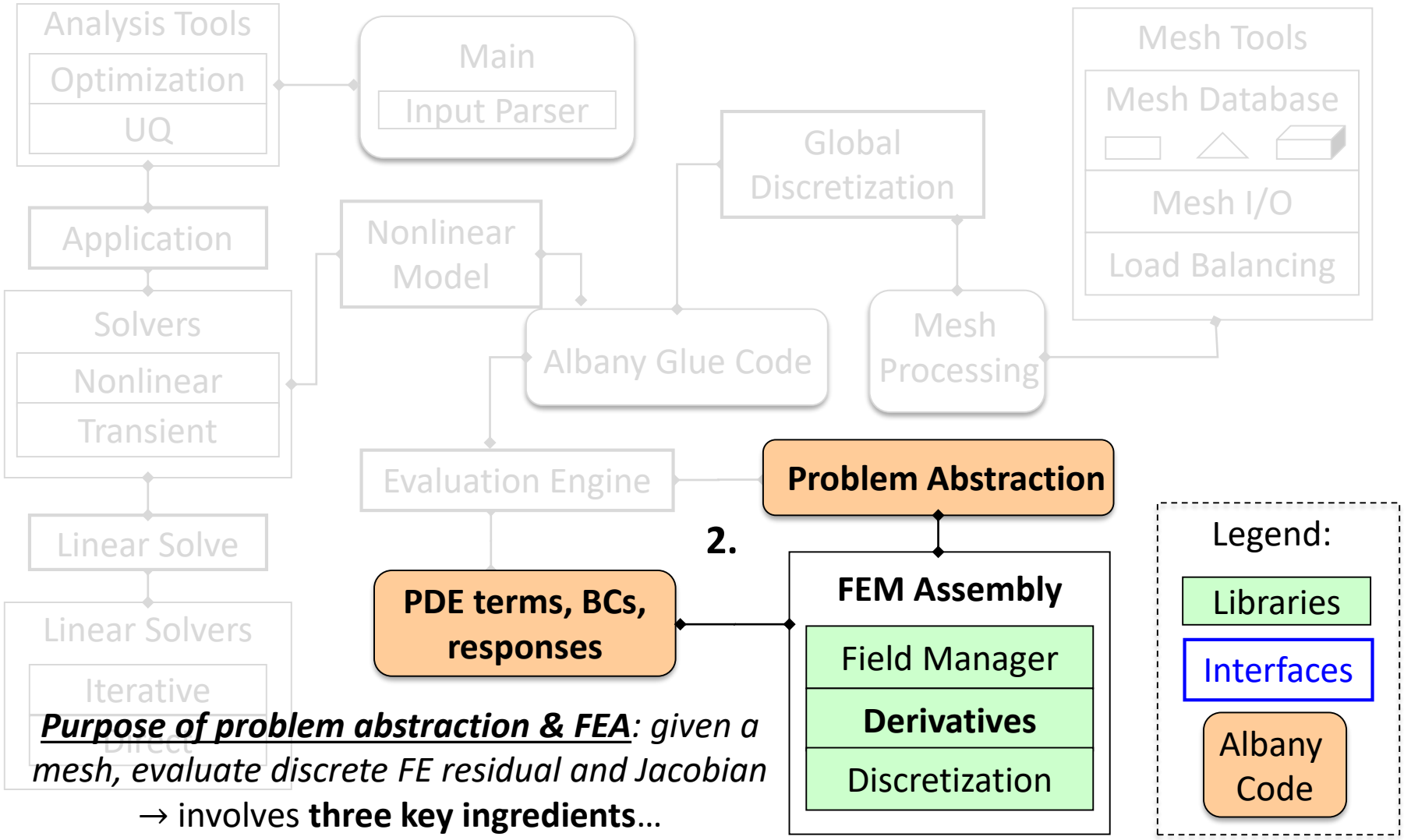


# What is Albany? (under-the-hood)

Sandia National Laboratories

**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"




***Purpose of problem abstraction & FEA**: given a mesh, evaluate discrete FE residual and Jacobian*

→ involves **three key ingredients**…

# 1. Templated-based automatic differentiation

Sandia National Laboratories

**Automatic differentiation (AD)** provides exact derivatives without time/effort of deriving and hand-coding them.

- Template equation implementation on scalar type.
- Libraries (Sacado) provides new scalar types that ***overload the math operators*** to propagate embedded quantities via chain rule.
  - *Derivatives:* DFad<double>
  - *Hessians*: DFad<SFad<double,N>>
  - *Stochastic Galerkin resid*: PCE<double>
  - *Stochastic Galerkin Jac*: DFad<PCE<double>
  - *Sensitivities*: DFad<double>

***No finite difference truncation error!***

- Great for ***multi-physics codes*** (e.g., many Jacobians) and ***advanced analysis*** (e.g., sensitivities, optimization)

| | double | DFad<double> |
|---|---|---|
| **Operation** | $c = a \pm b$ | $\dot{c} = \dot{a} \pm \dot{b}$ |
| | $c = ab$ | $\dot{c} = a\dot{b} + \dot{a}b$ |
| | $c = a/b$ | $\dot{c} = (\dot{a} - c\dot{b})/b$ |
| | $c = a^r$ | $\dot{c} = ra^{r-1}\dot{a}$ |
| | $c = \sin(a)$ | $\dot{c} = \cos(a)\dot{a}$ |
| | $c = \cos(a)$ | $\dot{c} = -\sin(a)\dot{a}$ |
| | $c = \exp(a)$ | $\dot{c} = c\dot{a}$ |
| | $c = \log(a)$ | $\dot{c} = \dot{a}/a$ |

```
template <typename ScalarT>
void computeF(ScalarT* x, ScalarT* f)
{
  f[0] = 2.0 * x[0] + x[1] * x[1];
  f[1] = x[0] * x[0] * x[0] + sin(x[1]);
}
```

```
double* x;
double* f;
…
computeF(x,f);
```

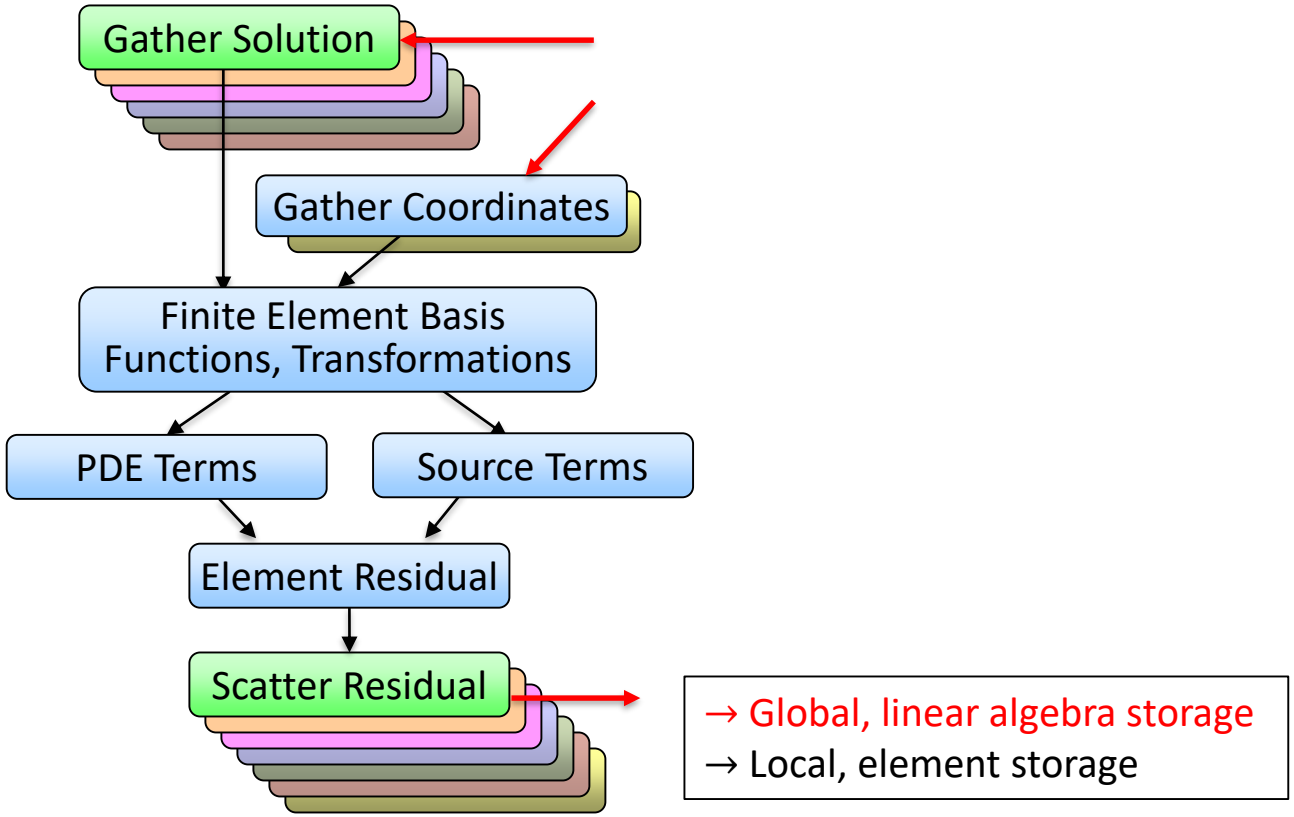
```
DFad<double>* x;
DFad<double>* f;
…
computeF(x,f);
```

E. Phipps *et. al*. Efficient expression templates for operator overloading-based AD, LNCSE, 309–319, 2012.

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***

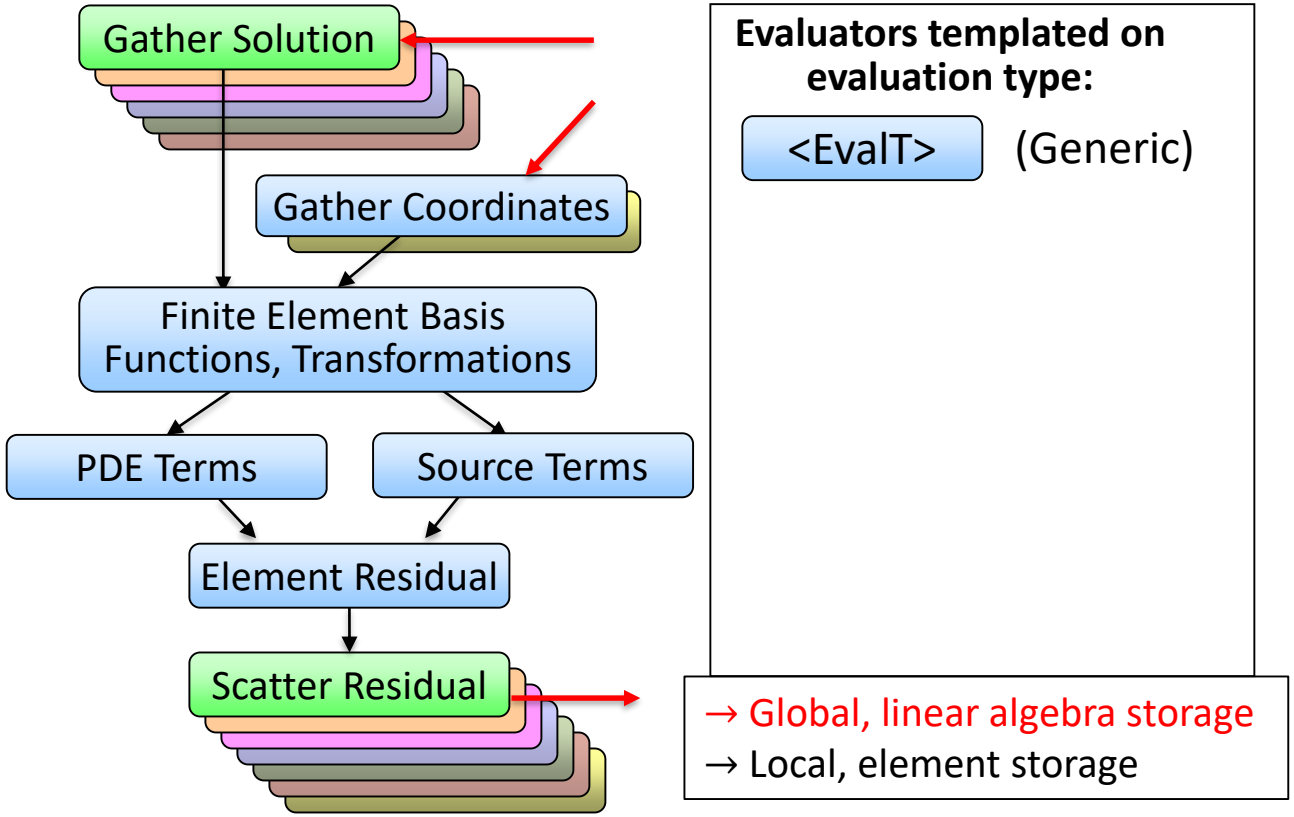



- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***

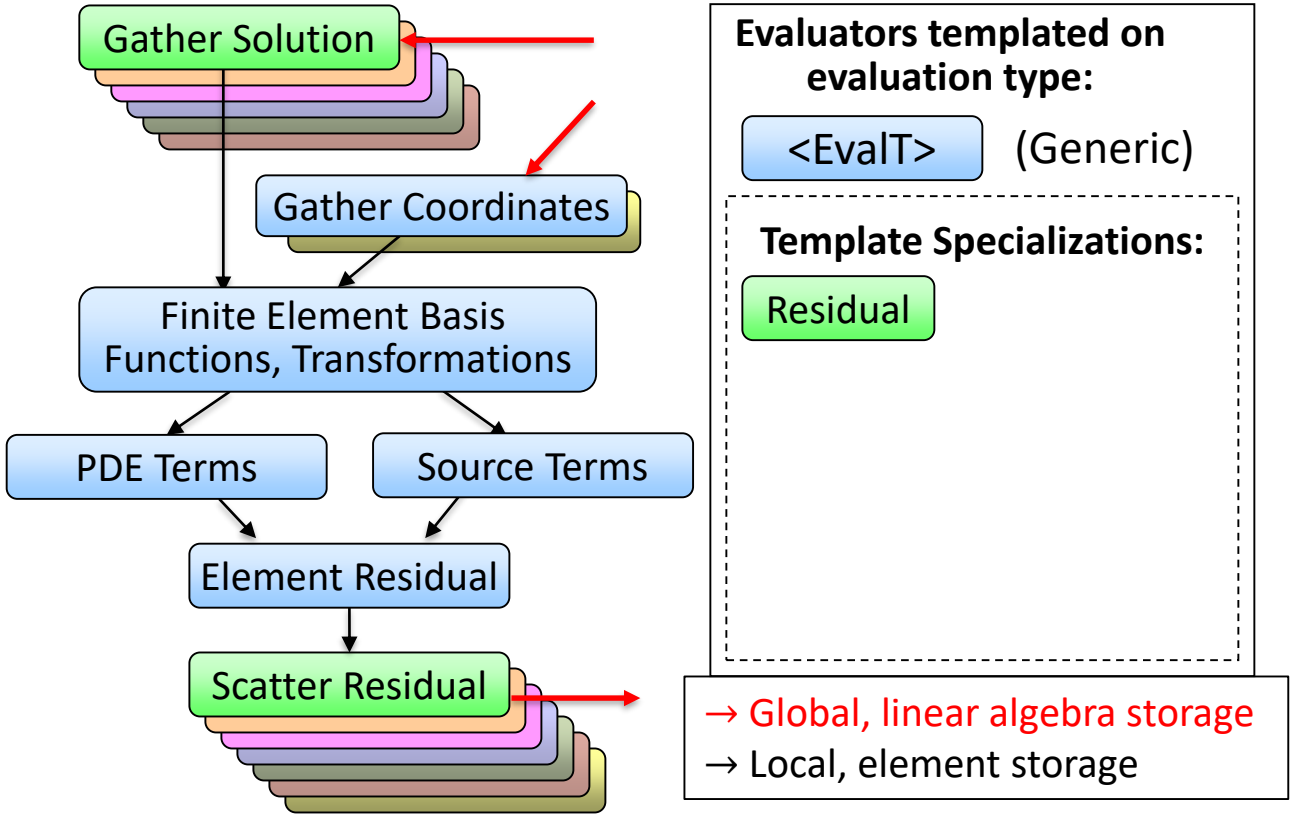



- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***

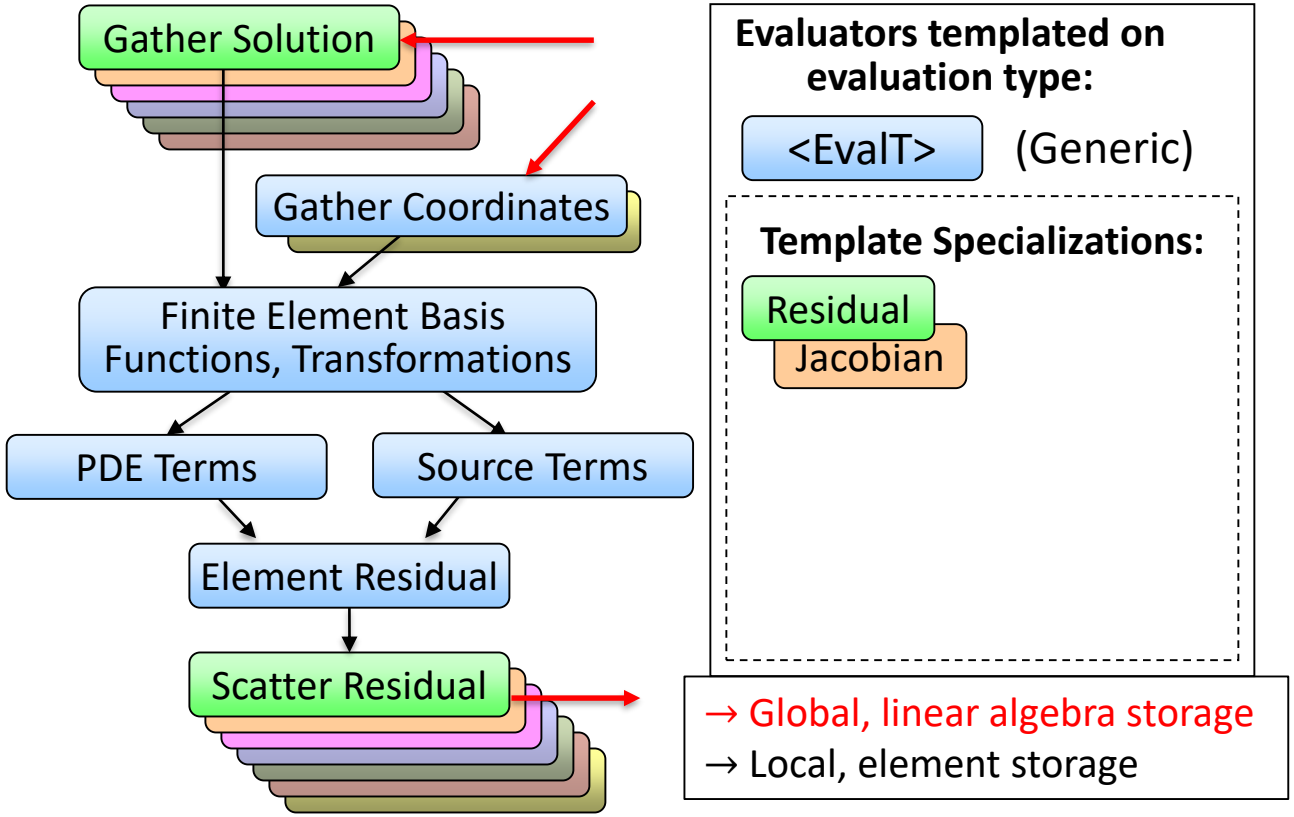



- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***

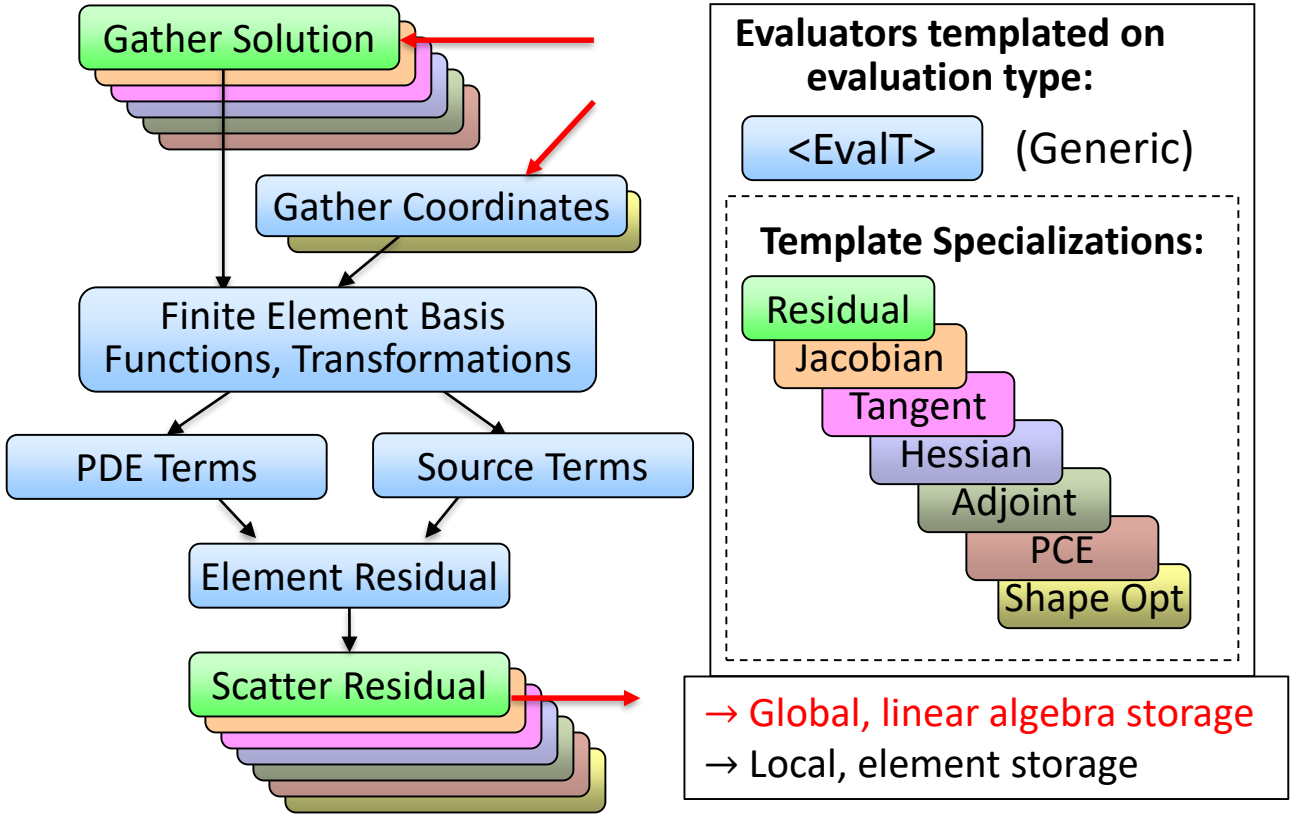



- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***

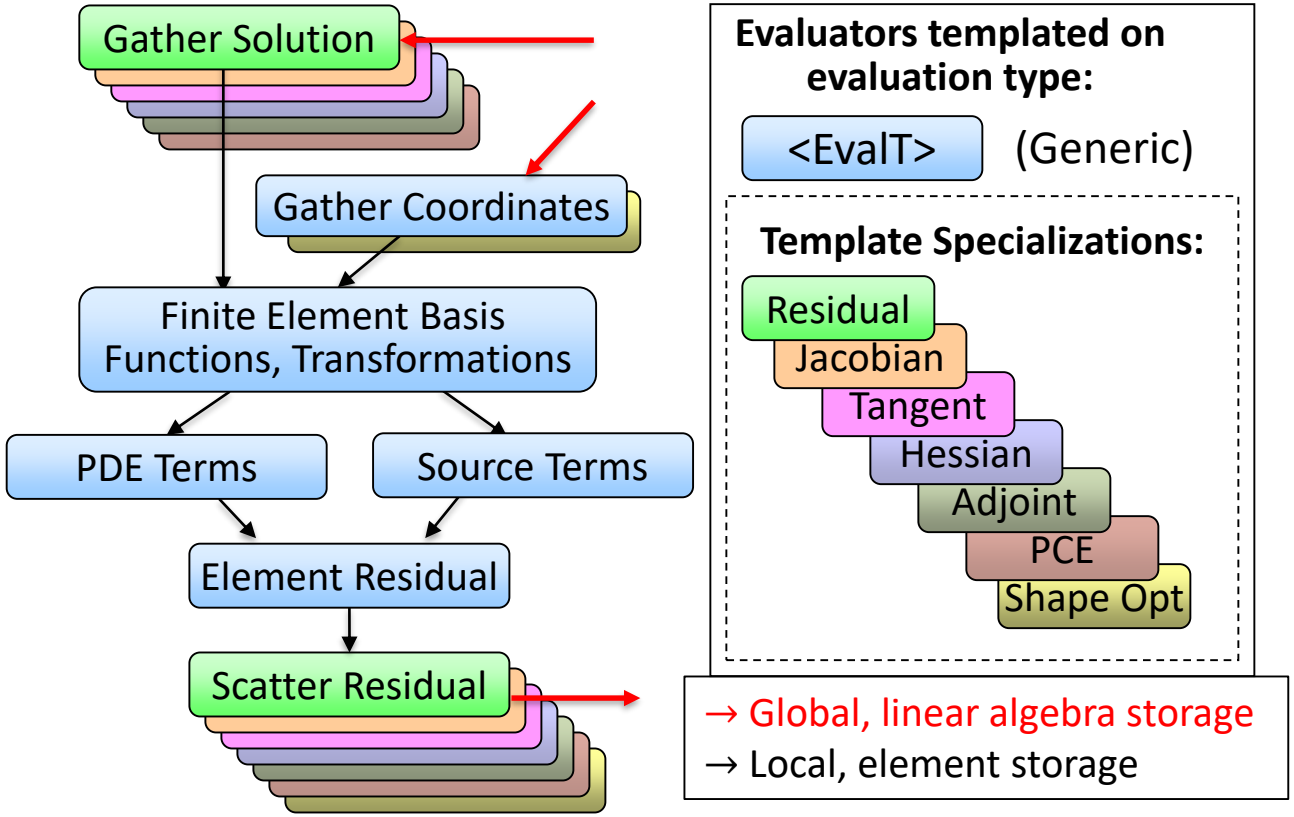



- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***




- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

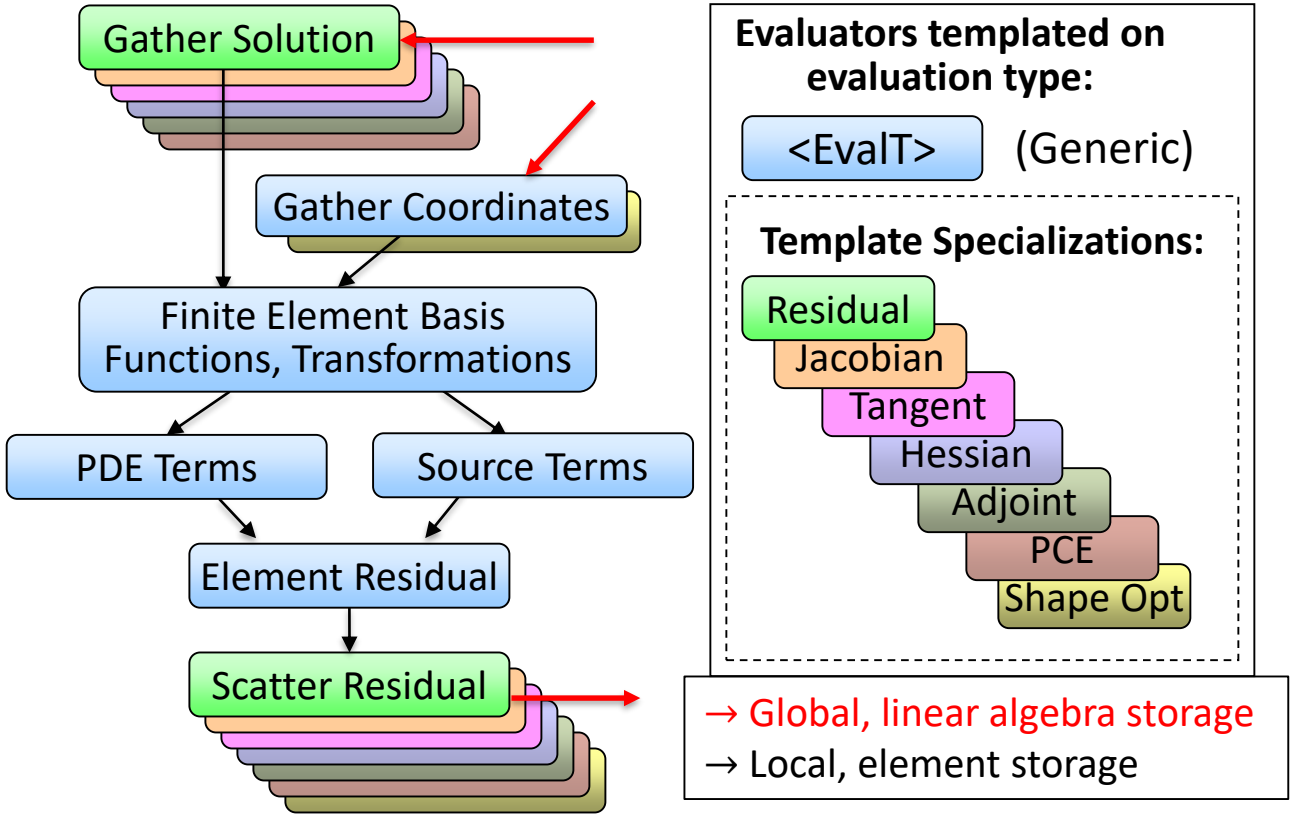
Enables **advanced analyses** (sensitivities, optimization, …)

R. Pawlowski et. al. *Sci. Program.*, 2012.

# 2. Template-based generic programming (TBGP)

Sandia National Laboratories

***Albany Finite Element Assembly (FEA):***




Employs **AD overloads** (Sacado, Stokhos)

- **Gather Solution** extracts values from global structures, puts in element local structures
- **Evaluators** operate on element local data structures
- **Scatter** adds local contributions to global structures

Enables **advanced analyses** (sensitivities, optimization, …)

# 3. Graph-based finite element assembly (FEA)

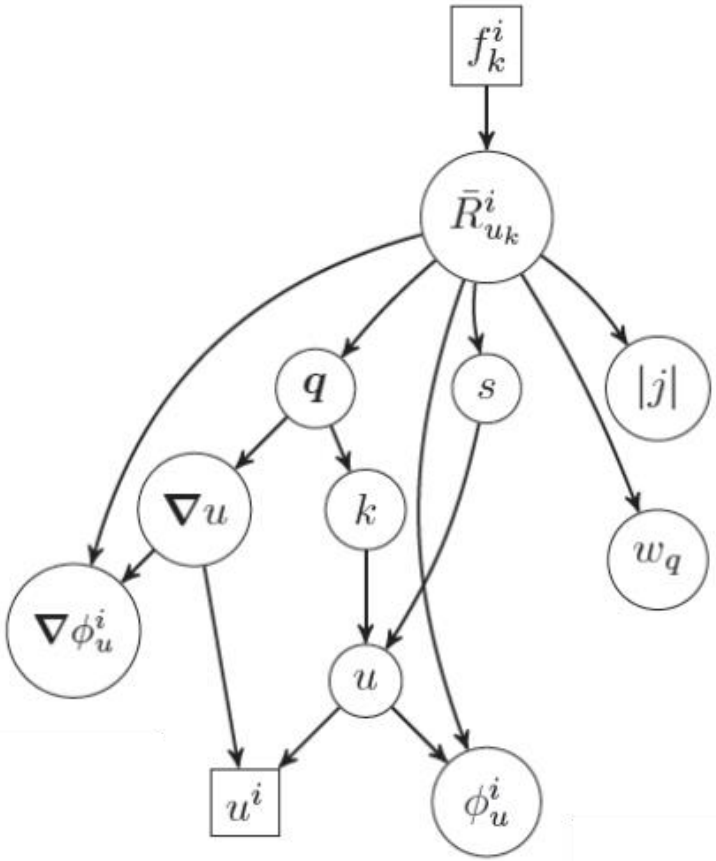
Sandia National Laboratories

**Assembly** of physics pieces comes down to the evaluation of a **directed acyclic graph (DAG)** of computations of field data.

$$R_u^i = \int_\Omega \left[ \phi_u^i \dot{u} - \boldsymbol{\nabla} \phi_u^i \cdot \boldsymbol{q} + \phi_u^i s \right] \, \mathrm{d}\Omega$$

**Phalanx package:** Local field evaluation kernel designed for assembly of arbitrary equation sets (i.e. evaluating residuals/Jacobians).

- **Decomposes** a complex model into a graph of **simple kernels** (functors)
- A node in the graph evaluates one or more **temporary fields**
- **Runtime** DAG construction of graph
- Achieves **flexible multi-physics assembly**



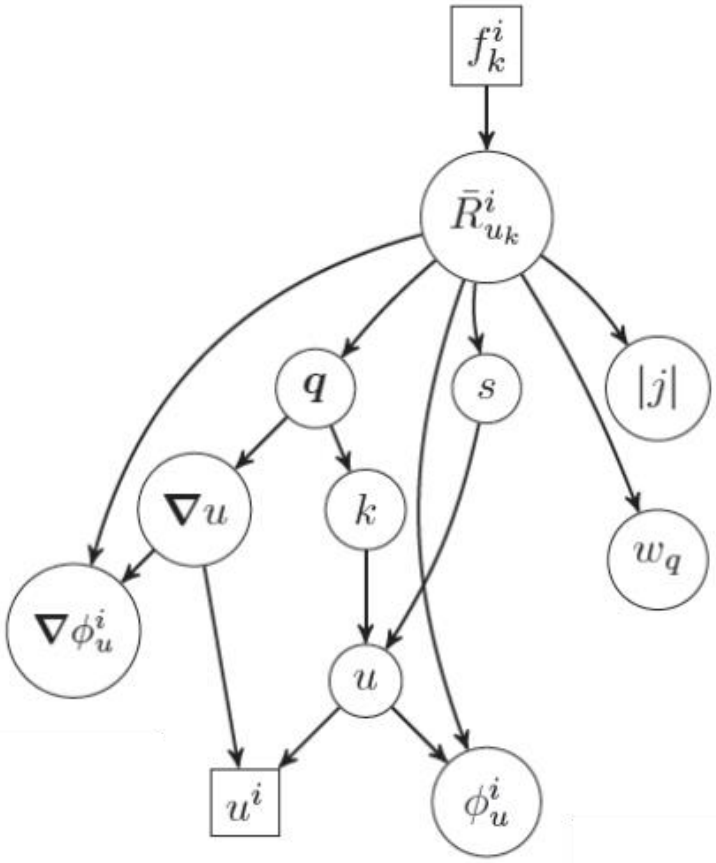
# 3. Graph-based finite element assembly (FEA)

Sandia National Laboratories

**Assembly** of physics pieces comes down to the evaluation of a **directed acyclic graph (DAG)** of computations of field data.

$$R_u^i = \int_\Omega \left[ \phi_u^i \dot{u} - \boldsymbol{\nabla} \phi_u^i \cdot \boldsymbol{q} + \phi_u^i s \right] \mathrm{d}\Omega$$

**Phalanx package:** Local field evaluation kernel designed for assembly of arbitrary equation sets (i.e. evaluating residuals/Jacobians).



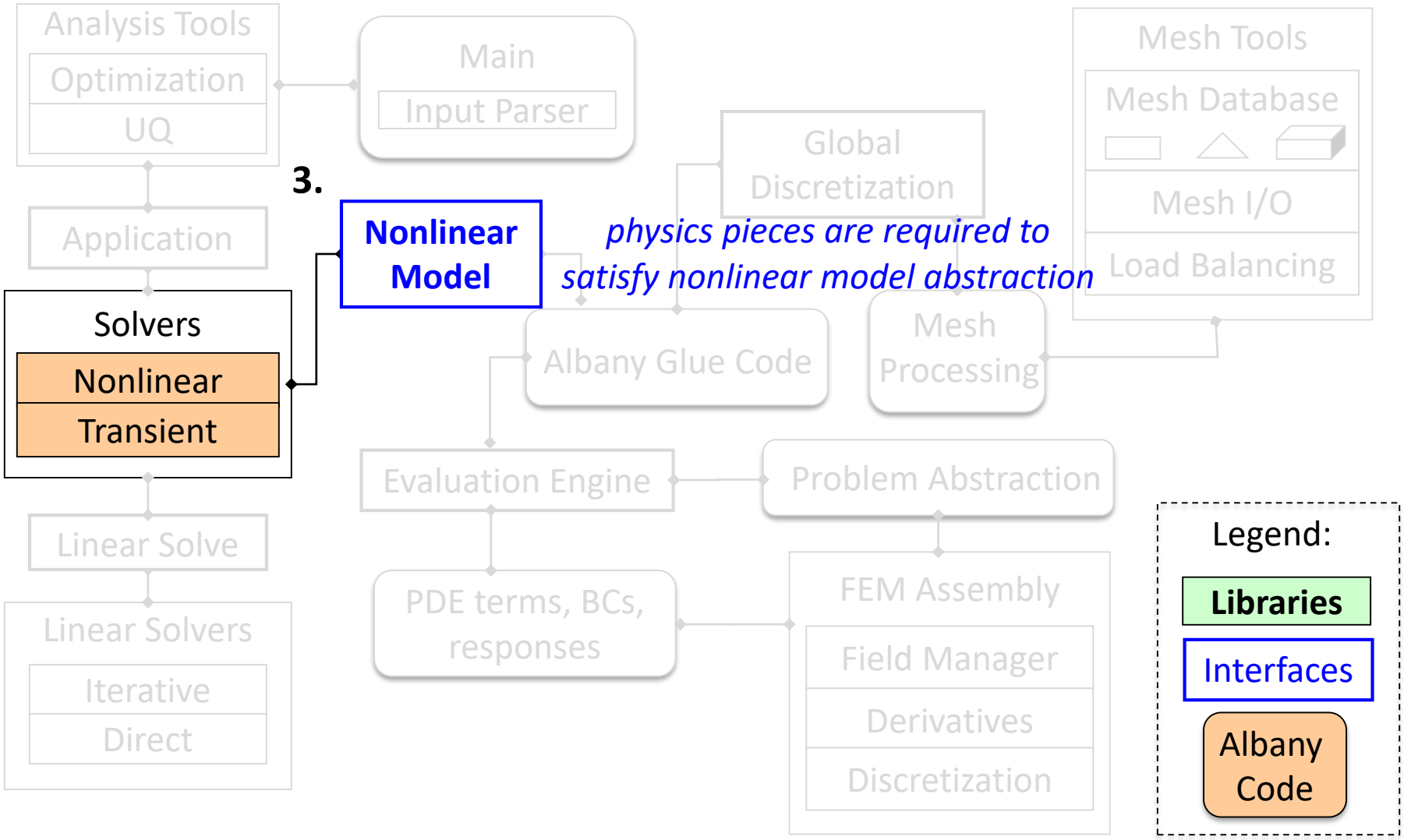

- **Decomposes** a complex model into a graph of **simple kernels** (functors)
- A node in the graph evaluates one or more **temporary fields**
- **Runtime** DAG construction of graph
- Achieves **flexible multi-physics assembly**

DAG-based assembly enables **flexibility**, **extensibility**, **rapid development:** to add new PDE, all you need to code is problem-specific residual $R_u^i$!

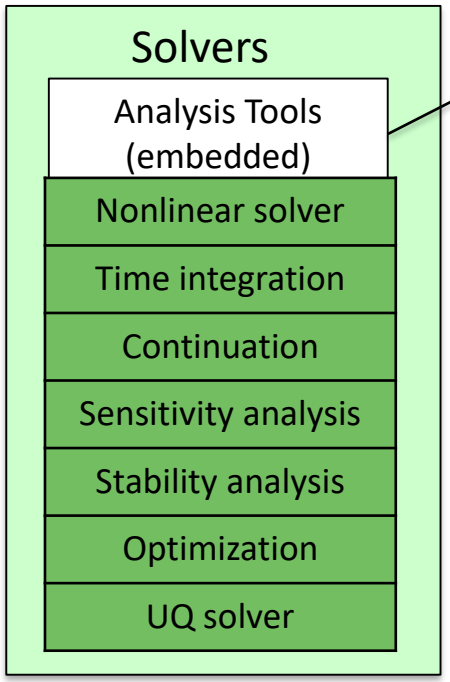
R. Pawlowski, Phalanx web site, http://trilinos.sandia.gov/packages/phalanx/ 2015.

# What is Albany? (under-the-hood)

**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# Nonlinear model abstraction & libraries




**"ModelEvaluator":**

| Given: | Computes: |
|---|---|
| $x$ | $f(\dot{x}, \ddot{x}, x, p, t)$ |
| $\dot{x}$ | $W = \alpha \frac{df}{d\dot{x}} + \beta \frac{df}{dx} + \omega \frac{df}{d\ddot{x}}, W_{prec}$ |
| $\ddot{x}$ | $\frac{df}{dp}$ |
| $p$ | $g$ |
| $t$ | $\frac{dg}{dx}, \frac{dg}{dp}$ |

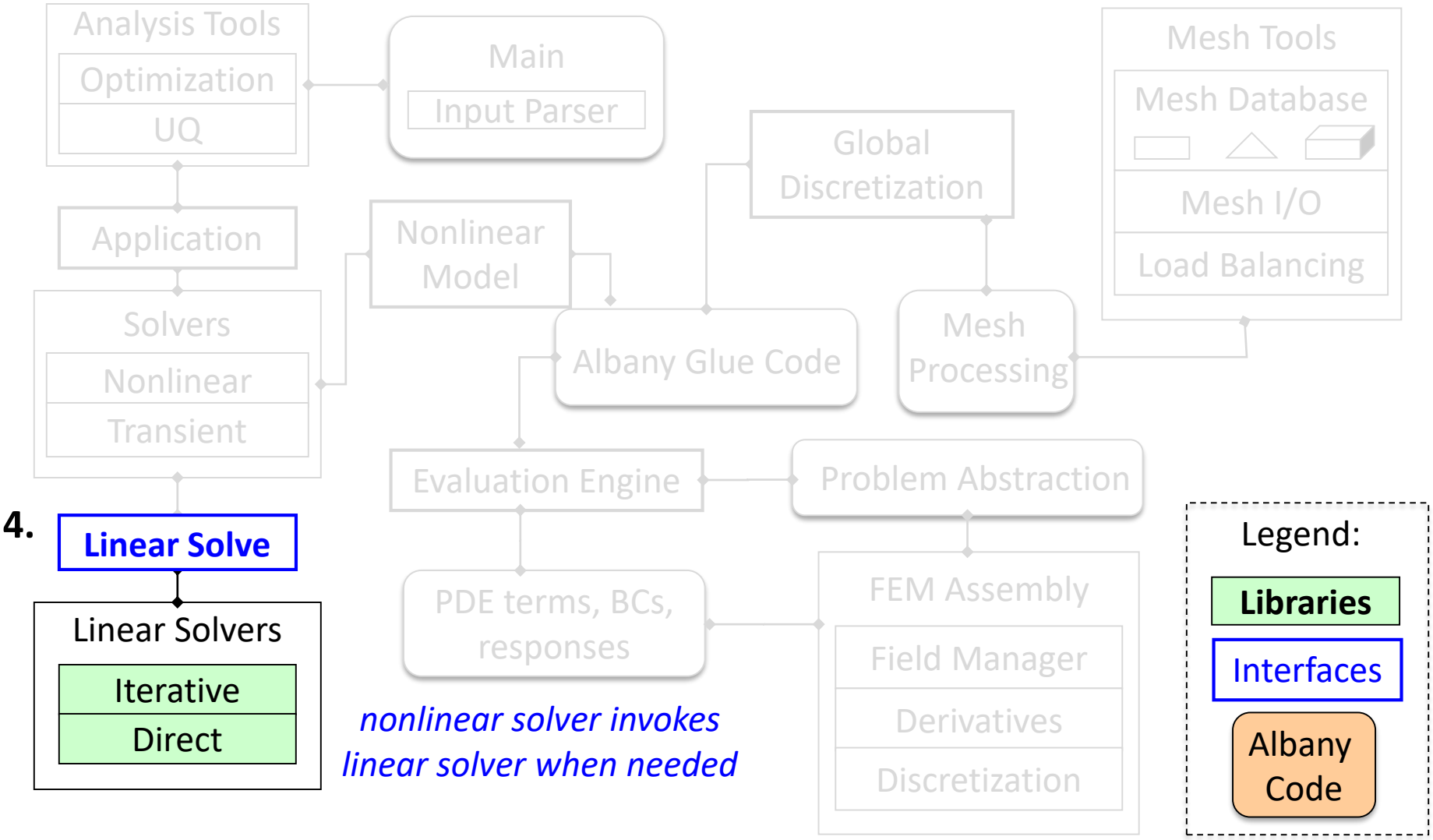
Access to Trilinos **embedded solvers** requires satisfaction of **ModelEvaluator** (nonlinear model) abstraction.

$f$ = residual; $x$ = solution; $p$ = params;
$g$ = responses; $t$ = time;
$W$ = Jacobian; $W_{prec}$ = Preconditioner

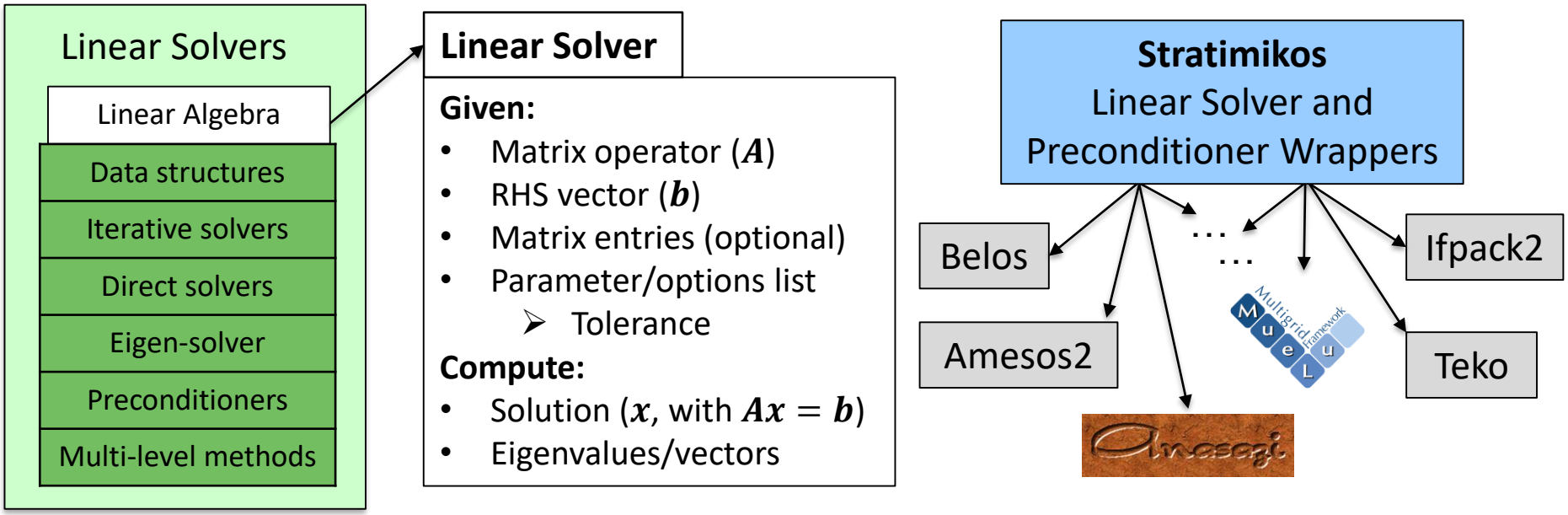
- Interface is **general** to accommodate computation of Jacobians, user-defined preconditioners, and stochastic Galerkin expansions.
- Enables "**beyond-forward analysis**": analysts/physics experts are not burdened with analysis algorithm requirements, i.e., programming sensitivities for implicit solvers, optimization, stability, bifurcation analysis.
  - **Advanced capabilities**: optimization (ROL), homotopy continuation (LOCA), embedded UQ (Stokhos).

# What is Albany? (under-the-hood)

**Albany** = Component Libraries + Abstract Interfaces + "Glue Code"



# Linear solver abstraction & libraries

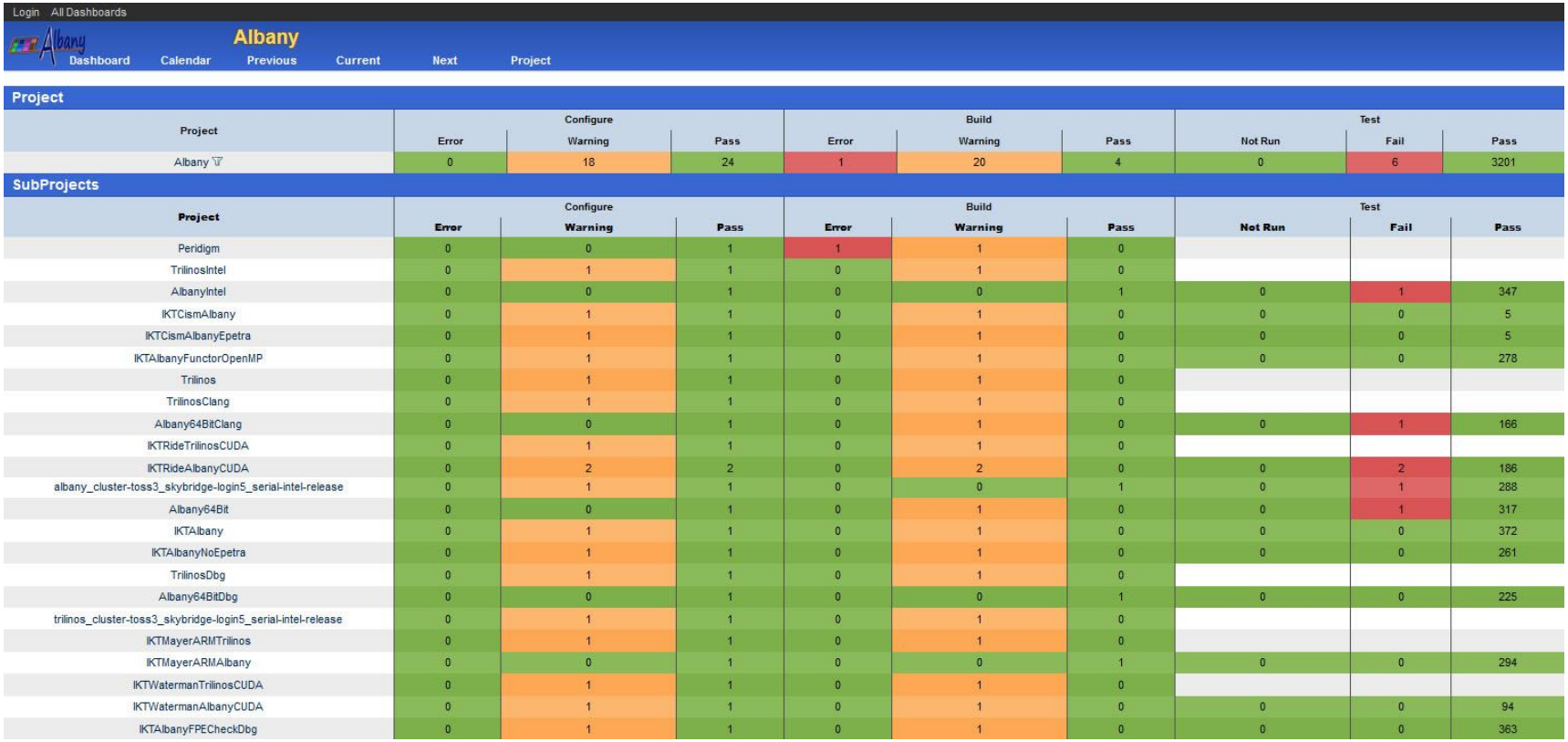



- **Linear solver abstraction** provides full access to all Trilinos linear solvers (direct and iterative), eigensolvers and preconditioners through Stratimikos interface.
- **Factory class** supports run-time solution configuration through input file options.
- **Available direct solvers**: Amesos, Amesos2 (UMFPACK, MUMPS, SuperLU, SCALAPACK, etc.).
- **Available iterative solvers**: AztecOO, Belos (CG, GMRES)
- **Available preconditioners**: Ifpack, Ifpack2 (ILU); ML, MueLu (AMG); Teko (block)
- **Eigensolvers**: Anasazi

# Software quality tools & processes

| | | |
|---|---|---|
| Repository* | Nightly test harness | Mailing lists |
| Version control | Unit tests | Issue tracking |
| Build system | Verification tests | Web pages |
| Config mgmt | Code coverage | Licensing |
| Regression tests | Performance tests | Release process |

Performance monitored via **CDash nightly testing** on a variety of architecture including GPU (P100, V100), Xeon Phi, Skylake, ARM platforms.



Login All Dashboards

Albany

Dashboard Calendar Previous Current Next Project

**Project**

| Project | Configure | | | Build | | | Test | | |
|---|---|---|---|---|---|---|---|---|---|
| | Error | Warning | Pass | Error | Warning | Pass | Not Run | Fail | Pass |
| Albany | 0 | 18 | 24 | 1 | 20 | 4 | 0 | 6 | 3201 |

**SubProjects**

| Project | Configure | | | Build | | | Test | | |
|---|---|---|---|---|---|---|---|---|---|
| | Error | Warning | Pass | Error | Warning | Pass | Not Run | Fail | Pass |
| Peridigm | 0 | 0 | 1 | 1 | 1 | 0 | | | |
| TrilinosIntel | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| AlbanyIntel | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 347 |
| IKTCismAlbany | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5 |
| IKTCismAlbanyEpetra | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5 |
| IKTAlbanyFunctorOpenMP | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 278 |
| Trilinos | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| TrilinosClang | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| Albany64BitClang | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 166 |
| IKTRideTrilinosCUDA | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTRideAlbanyCUDA | 0 | 2 | 2 | 0 | 2 | 0 | 0 | 2 | 186 |
| albany_cluster-toss3_skybridge-login5_serial-intel-release | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 288 |
| Albany64Bit | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 317 |
| IKTAlbany | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 372 |
| IKTAlbanyNoEpetra | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 261 |
| TrilinosDbg | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| Albany64BitDbg | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 225 |
| trilinos_cluster-toss3_skybridge-login5_serial-intel-release | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTMayerARMTrilinos | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTMayerARMAlbany | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 294 |
| IKTWatermanTrilinosCUDA | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTWatermanAlbanyCUDA | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 94 |
| IKTAlbanyFPECheckDbg | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 363 |

*** Albany github repo:** https://github.com/SNLComputation/Albany.

# Software quality tools & processes

| | | |
|---|---|---|
| Repository* | Nightly test harness | Mailing lists |
| Version control | Unit tests | Issue tracking |
| Build system | Verification tests | Web pages |
| Config mgmt | Code coverage | Licensing |
| Regression tests | Performance tests | Release process |

Performance monitored via **CDash nightly testing** on a variety of architecture including GPU (P100, V100), Xeon Phi, Skylake, ARM platforms.

Login All Dashboards

Albany

Dashboard Calendar Previous Current Next Project

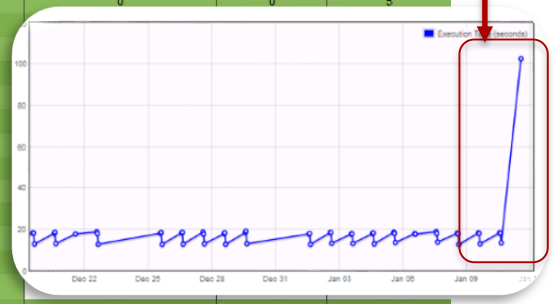
**Project**

| Project | Configure | | | Build | | | Test | | |
|---|---|---|---|---|---|---|---|---|---|
| | Error | Warning | Pass | Error | Warning | Pass | Not Run | Fail | |
| Albany | 0 | 18 | 24 | 1 | 20 | 4 | 0 | 6 | |

**SubProjects**

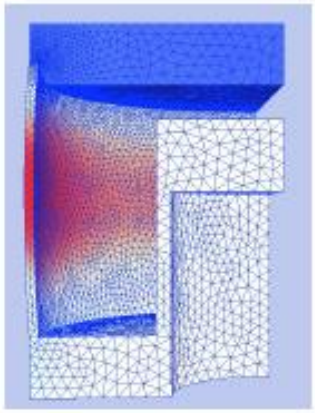
| Project | Configure | | | Build | | | Test | | |
|---|---|---|---|---|---|---|---|---|---|
| | Error | Warning | Pass | Error | Warning | Pass | Not Run | Fail | |
| Peridigm | 0 | 0 | 1 | 1 | 1 | 0 | | | |
| TrilinosIntel | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| AlbanyIntel | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 347 |
| IKTCismAlbany | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5 |
| IKTCismAlbanyEpetra | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTAlbanyFunctorOpenMP | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| Trilinos | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| TrilinosClang | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| Albany64BitClang | 0 | 0 | 1 | 0 | 1 | 0 | | | |
| IKTRideTrilinosCUDA | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTRideAlbanyCUDA | 0 | 2 | 2 | 0 | 2 | 0 | | | |
| albany_cluster-toss3_skybridge-login5_serial-intel-release | 0 | 1 | 1 | 0 | 0 | 1 | | | |
| Albany64Bit | 0 | 0 | 1 | 0 | 1 | 0 | | | |
| IKTAlbany | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTAlbanyNoEpetra | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| TrilinosDbg | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| Albany64BitDbg | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 225 |
| trilinos_cluster-toss3_skybridge-login5_serial-intel-release | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTMayerARMTrilinos | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTMayerARMAlbany | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 294 |
| IKTWatermanTrilinosCUDA | 0 | 1 | 1 | 0 | 1 | 0 | | | |
| IKTWatermanAlbanyCUDA | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 94 |
| IKTAlbanyFPECheckDbg | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 363 |

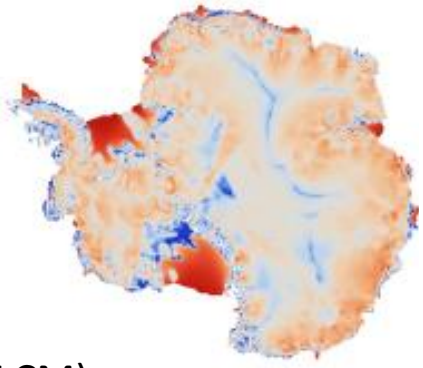
Slowdown identified by nightly CDash testing!

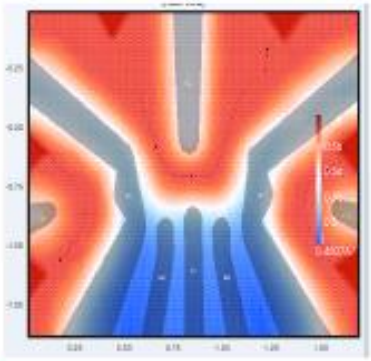


* **Albany github repo:** https://github.com/SNLComputation/Albany.

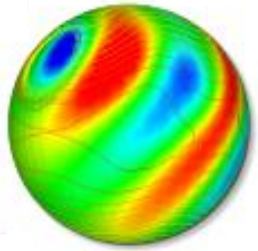
# Applications hosted by Albany

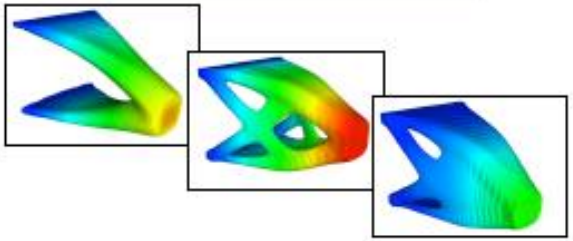


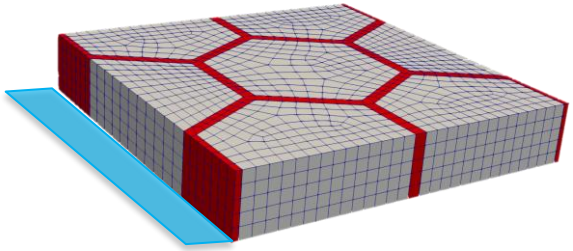




- Quantum Devices (QCAD)
- Ice Sheets (Albany Land-Ice)
- Mechanics (LCM)
- Atmosphere Dynamics (Aeras)
- Particle-continuum coupling (Peridigm-LCM)
- Additive Manufacturing Design (ATO)
- Additive Manufacturing Processing (AMP)
- Arctic Coastal Erosion (ACE)
- Coupled Geomechanics (Albotran).

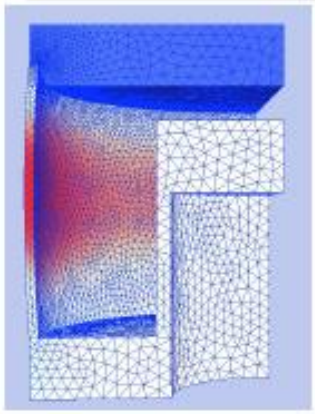


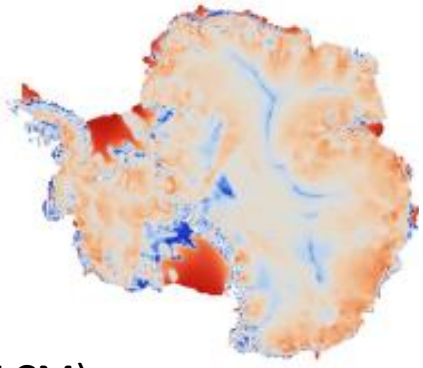




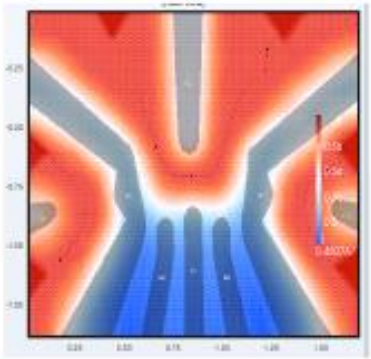
# Applications hosted by Albany

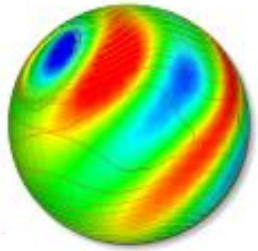
Applications are "**born**" scalable, fast, robust, and equipped with advanced analysis capabilities!

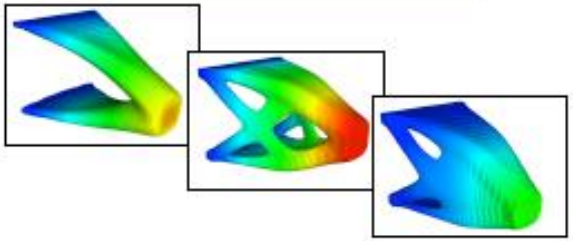


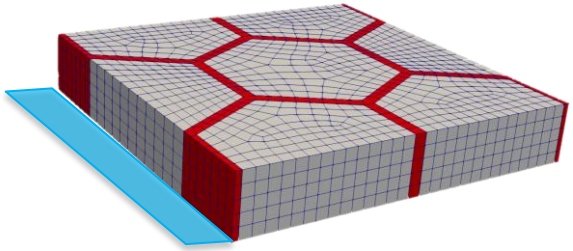


- Quantum Devices (QCAD)
- Ice Sheets (Albany Land-Ice)
- Mechanics (LCM)
- Atmosphere Dynamics (Aeras)
- Particle-continuum coupling (Peridigm-LCM)
- Additive Manufacturing Design (ATO)
- Additive Manufacturing Processing (AMP)
- Arctic Coastal Erosion (ACE)
- Coupled Geomechanics (Albotran).



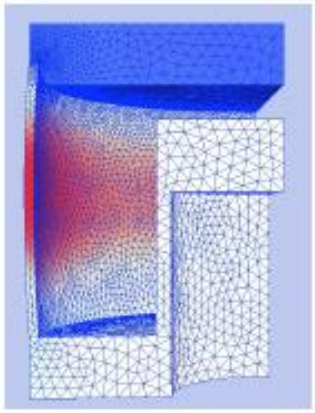


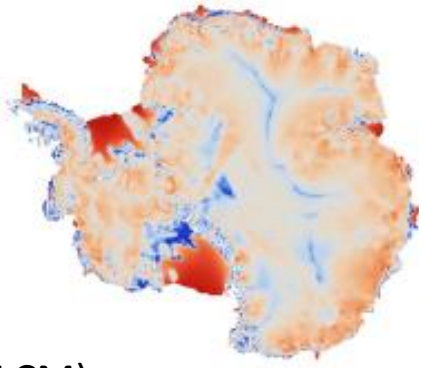




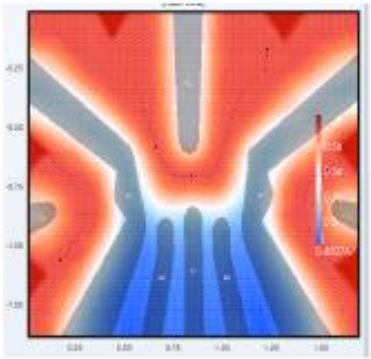
# Applications hosted by Albany

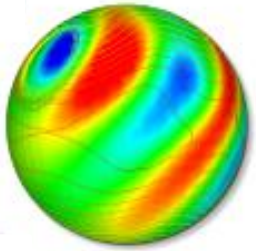
Applications are "**born**" scalable, fast, robust, and equipped with advanced analysis capabilities!

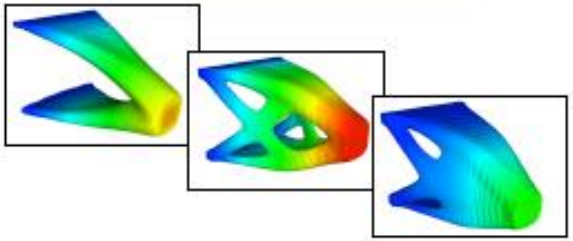


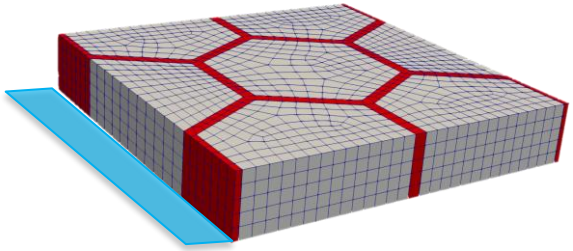


- Quantum Devices (QCAD)
- **Ice Sheets (Albany Land-Ice)**
- **Mechanics (LCM)**
- Atmosphere Dynamics (Aeras)
- Particle-continuum coupling (Peridigm-LCM)
- Additive Manufacturing Design (ATO)
- Additive Manufacturing Processing (AMP)
- **Arctic Coastal Erosion (ACE)**
- Coupled Geomechanics (Albotran).









# Ice sheets: Albany Land-Ice (ALI)

Sandia National Laboratories

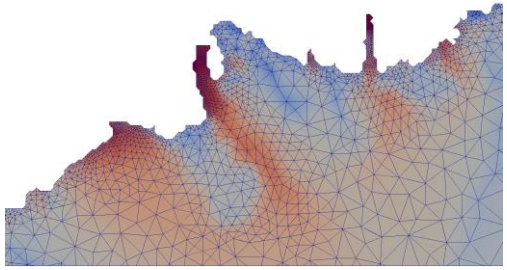
Albany enabled the **rapid development** of a production **land-ice dycore** for providing **actionable predictions** of **21st century sea-level rise** as a part of the DOE Energy Exascale Earth System Model (E3SM).

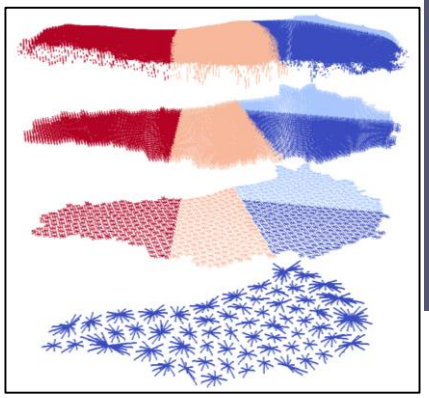
## *Capabilities:*

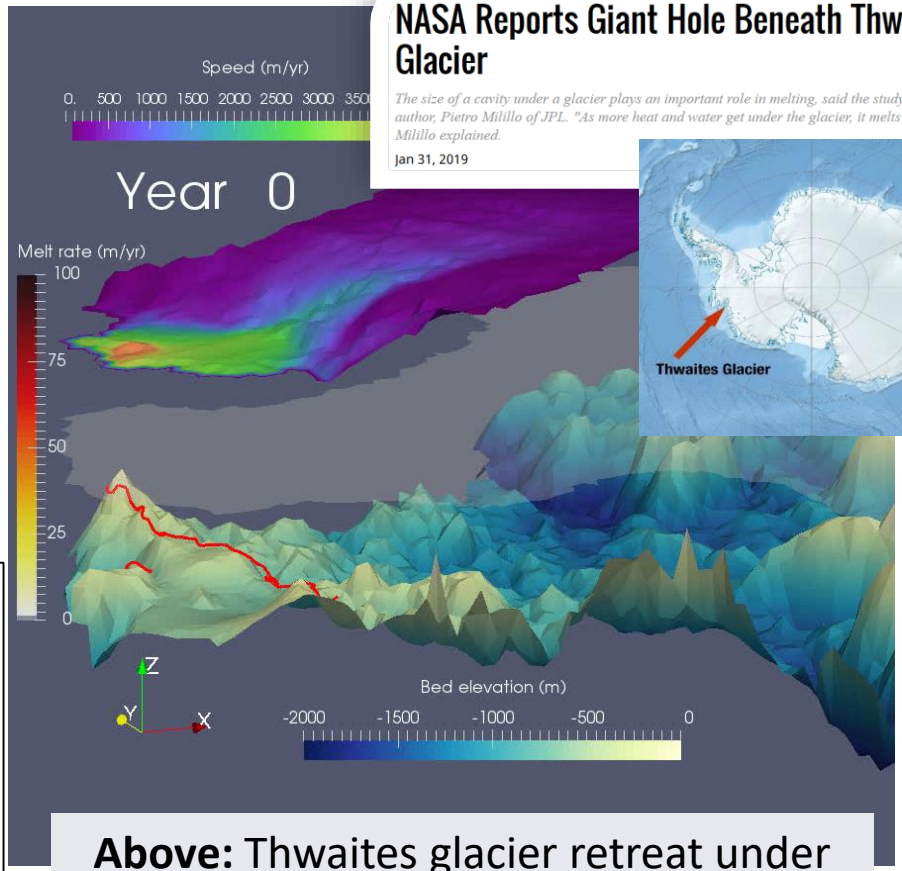
- ***Unstructured grid*** finite elements.
- ***Scalable, fast*** and ***robust***
- ***Verified*** and ***validated***
- ***Advanced analysis***: inversion, UQ
- ***Portable*** to GPU, KNL, … via Kokkos
- ***Multi-physics:*** velocity-temperature, velocity-thickness, velocity-hydrology

E3SM Energy Exascale Earth System Model

SciDAC Scientific Discovery through Advanced Computing








NASA Reports Giant Hole Beneath Thwaites Glacier

*The size of a cavity under a glacier plays an important role in melting, said the study's lead author, Pietro Milillo of JPL. "As more heat and water get under the glacier, it melts faster," Milillo explained.*

Jan 31, 2019

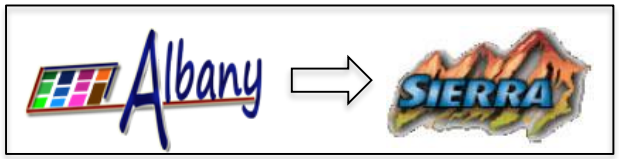



**Above:** Thwaites glacier retreat under parametrized submarine melting

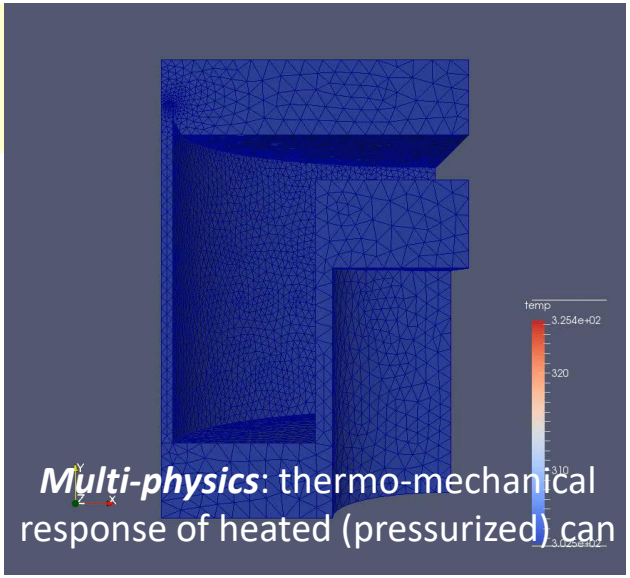
# Laboratory for computational mechanics (LCM)

Sandia National Laboratories

The Albany LCM suite contains sophisticated **material models**, **physics** and **technologies** for solid mechanics.

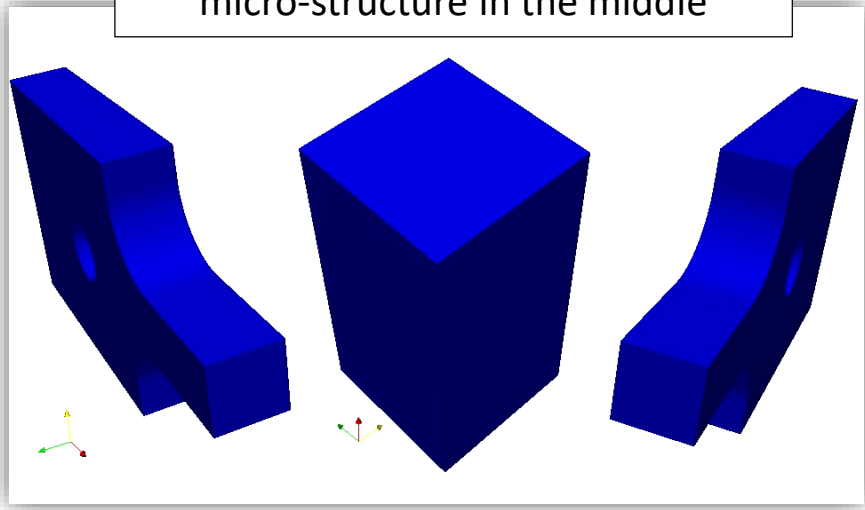



- "**Sand-box**" for new algorithms/methods:
  - Composite 10-node tetrahedron
  - Pressure projection stabilization
  - Multi-scale coupling via Schwarz
  - In-memory mesh adaptation
- **Models**: elasticity, Neohookean, $J_2$ plasticity, crystal plasticity, elasto-visco-plastic, …
- **Physics (PDEs)**: elasticity, mechanics, electro-mechanics, thermo-mechanics, thermo-poro-mechanics, …
- **Fracture** and **damage** simulation capabilities




***Multi-physics***: thermo-mechanical response of heated (pressurized) can

***Multi-scale***: tensile specimen with micro-structure in the middle

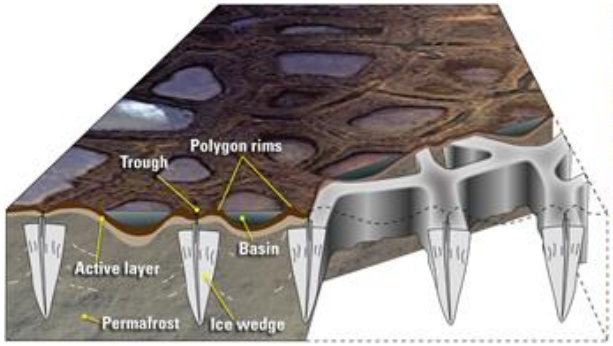


# Arctic Coastal Erosion (ACE)

Mechanistic modeling within Albany is **advancing state-of-the-art coastal erosion/permafrost modeling**.

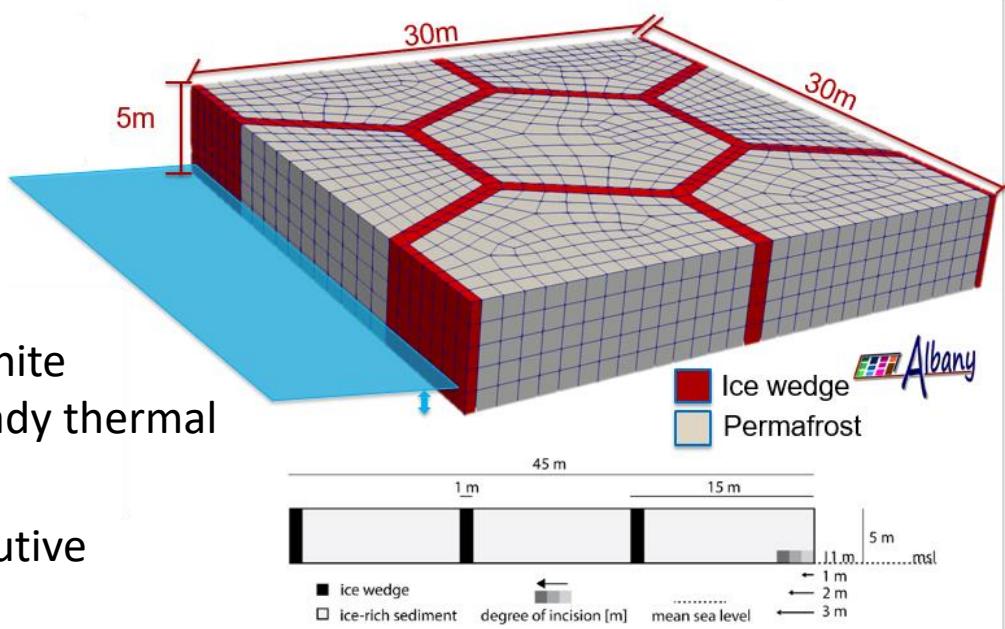
**Permafrost modeling background:**

- Predominant geomorphology is **ice-wedge polygons** (right).
- **State-of-the-art erosion modeling**: trend projection, empirical relation-ships, 1D steady-state heat flow, …




**Albany modeling of degradation:**

- **Leverages** years of LCM R&D.
- **Time-varying input variables** over the duration of a storm (water level, temperature, salinity)
- **Multi-physics FEM model of coastline**: finite deformation plasticity model + 3D unsteady thermal flow + chemical characteristics
  - **Failure modes** develop from constitutive relationships (**no empirical model**!)



# Algorithmic projects hosted by Albany

Sandia National Laboratories

**Algorithms** and **software** are **matured directly** on **applications.**

## Algorithmic projects within Albany:

- Scalable multi-level solvers (PISCEES/ProSPect) – R. Tuminaro, I. Tezaur.
- Nonlinear solvers (FASTMath) – R. Pawlowski, M. Perego
- In-memory mesh adaptation (FASTMath) – M. Sheppard, M. Bloomfield (RPI), A. Oberai, J. Smith (USC), D. Ibanez, B. Granzow, G. Hansen
- Multi-scale coupling via Schwarz (P&EM) – A. Mota, I. Tezaur, C. Alleman, G. Phlipot (CalTech)
- Stabilized mechanics (FASTMath) – A. Bradley, J. Ostien, G. Hansen
- Adjoint-based inversion (FASTMath) – M. Perego, E. Phipps, ROL team
- Optimization-based coupling (ASCR) – M. Perego, M. D'Elia, D. Littlewood, P. Bochev
- UQ workflow (PISCEES) – J. Jakeman, I. Tezaur, M. Perego
- Embedded UQ (Equinox) – E. Phipps, J. Fike
- Performance portable FEM (PISCEES/ProSPect/ATDM) – I. Demeshko (LANL), E. Phipps, R. Pawlowski, E. Cyr, I. Tezaur, A. Bradley, J. Watkins
- Development of composite tet-10 for solid mechanics (PE&M) – J. Foulk, J. Ostien, A. Mota

# Algorithmic projects hosted by Albany

Sandia National Laboratories

**Algorithms** and **software** are **matured directly** on **applications.**

## Algorithmic projects within Albany:

- Scalable multi-level solvers (PISCEES/ProSPect) – R. Tuminaro, I. Tezaur.
- Nonlinear solvers (FASTMath) – R. Pawlowski, M. Perego
- **In-memory mesh adaptation** (FASTMath) – M. Sheppard, M. Bloomfield (RPI), A. Oberai, J. Smith (USC), D. Ibanez, B. Granzow, G. Hansen
- **Multi-scale coupling via Schwarz** (P&EM) – A. Mota, I. Tezaur, C. Alleman, G. Phlipot (CalTech)
- Stabilized mechanics (FASTMath) – A. Bradley, J. Ostien, G. Hansen
- **Adjoint-based inversion** (FASTMath) – M. Perego, E. Phipps, ROL team
- Optimization-based coupling (ASCR) – M. Perego, M. D'Elia, D. Littlewood, P. Bochev
- UQ workflow (PISCEES) – J. Jakeman, I. Tezaur, M. Perego
- Embedded UQ (Equinox) – E. Phipps, J. Fike
- Performance portable FEM (PISCEES/ProSPect/ATDM) – I. Demeshko (LANL), E. Phipps, R. Pawlowski, E. Cyr, I. Tezaur, A. Bradley, J. Watkins
- Development of composite tet-10 for solid mechanics (PE&M) – J. Foulk, J. Ostien, A. Mota

# Adjoint-based optimization/inversion

Sandia National Laboratories

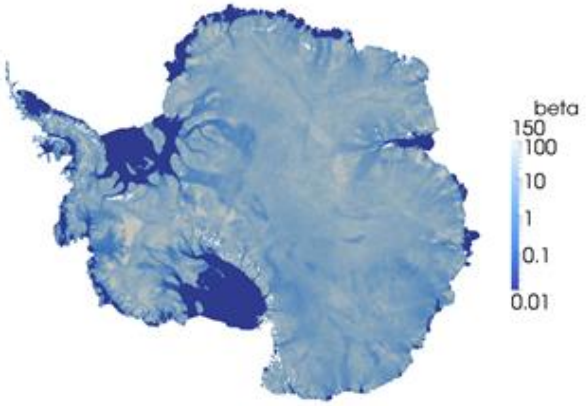
AD & TBGP in Albany enabled the efficient solution of **adjoint-based PDE-constrained optimization/inversion** problems.


$$\begin{aligned} &\text{Find } p \text{ that minimizes} && g(u,p) \\ &\text{subject to} && f(u,p)=0 \quad \leftarrow \quad \text{PDE} \end{aligned}$$

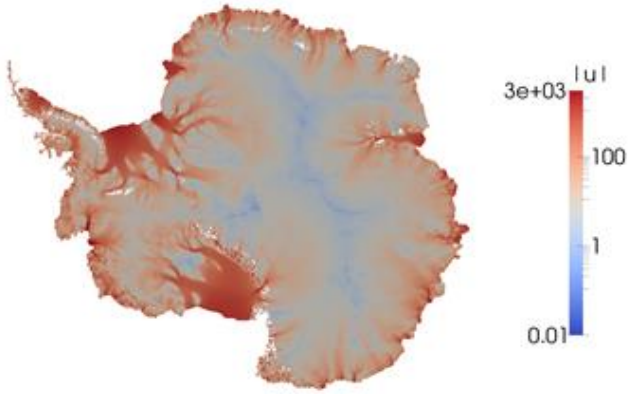
Inversion problem solved robustly for ***O*(100K) parameters**!

***Application:*** inversion for basal friction and ice thickness in Albany Land-Ice model to initialize dynamic simulation.

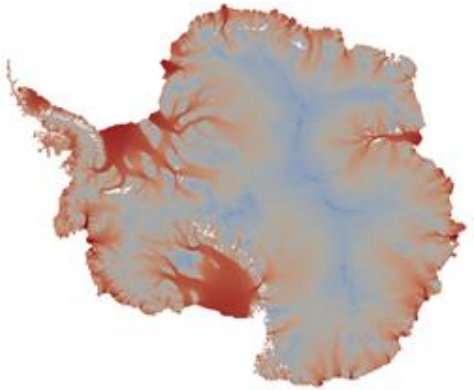
- Inversion approach **significantly reduces non-physical transients**.




$\beta$ (kPa y/m) obtained through inversion



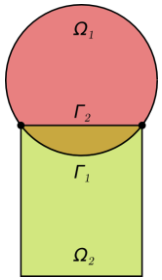

$|\boldsymbol{u}|$ (m/yr) computed with estimated $\beta$



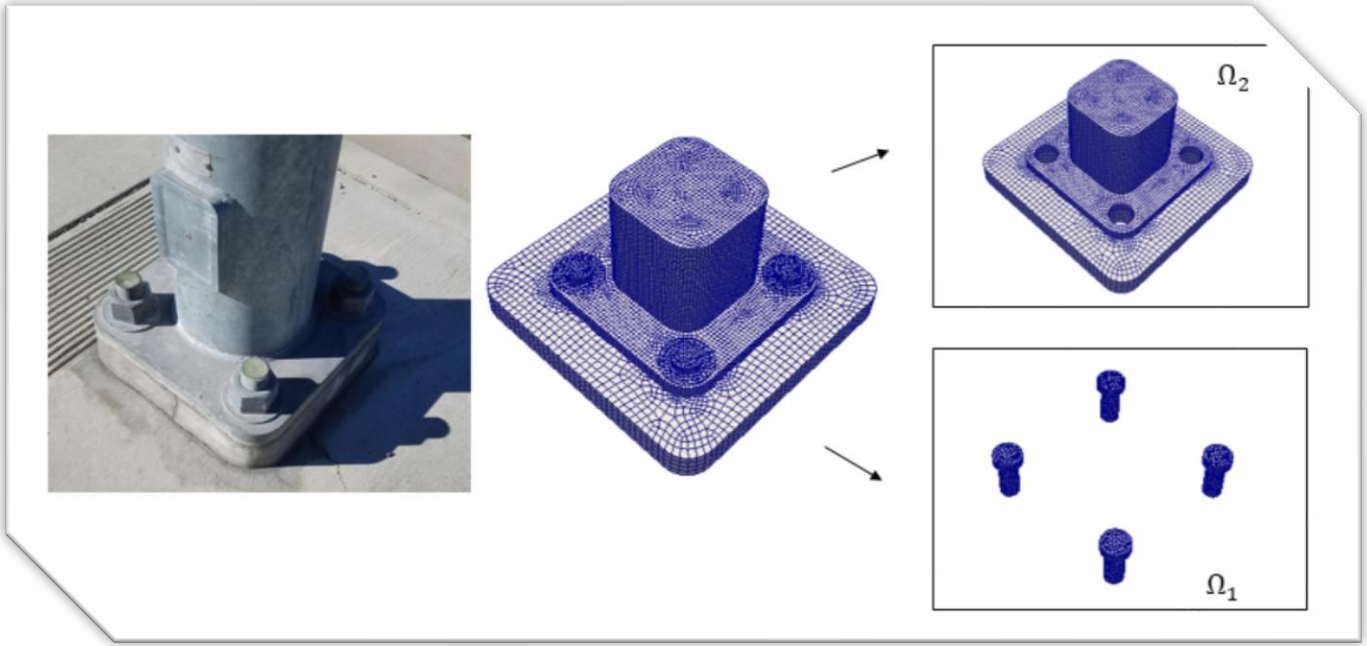
$|\boldsymbol{u}|$ (m/yr) for observed surface velocity

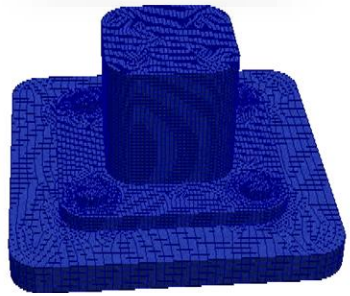
# Multi-scale coupling via Schwarz

A **domain decomposition alternating-Schwarz**-based method has been developed in Albany for **concurrent multi-scale coupling** in solid mechanics*.




- **Targeted application**: failure of **bolted components**.
- **"Plug-and-play" framework**: simplifies meshing complex geometries!
  - ➢ Couple regions with **different non-conformal meshes, element types, levels of refinement, solvers/time-integrators.**



**Schwarz coupling:**
hex + composite tet 10

*** See talk by A. Mota**: MS 350, Fri. Mar. 1, 10:10-10:30AM, Room 302B

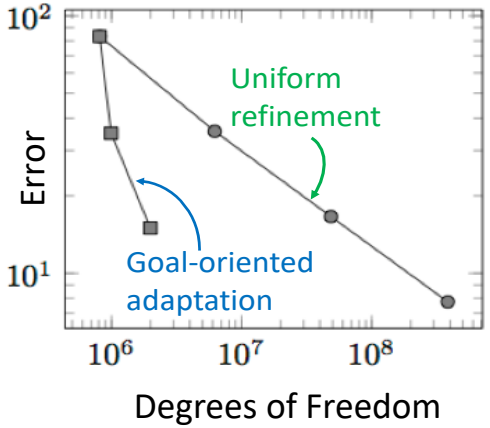
# In-memory mesh adaptation

Sandia National Laboratories

Collaboration with **SCOREC***: development of **mesh adaptation** capabilities in Albany to enable **multi-scale/multi-physics adaptive simulation**

- Automated ***parallel goal-oriented adaptive simulation***
  - Use **adjoint solution** to drive mesh adaptation
  - ~100× **DoF-efficiency** observed
  - **Scaling** out to at least 8K MPI ranks
  - **Performance portable** via Kokkos

PUMI
Parallel Unstructured Mesh Infrastructure

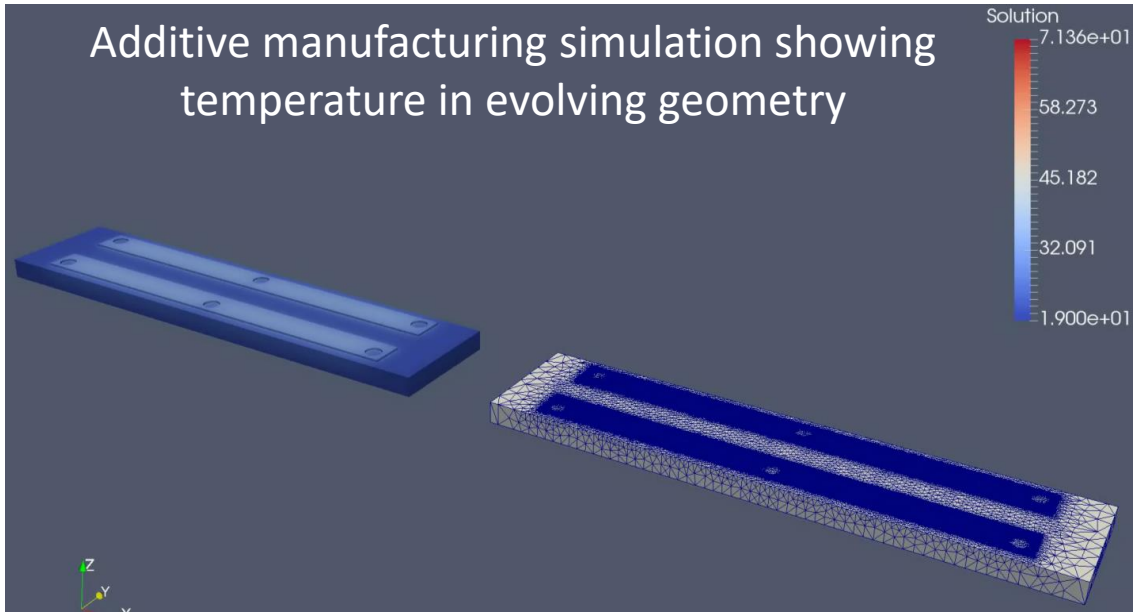



SCOREC

Rensselaer

***Some applications:***

- **3D manufacturing (right)**
- Creep/plasticity in large solder joint arrays
- Coupled dislocation dynamics



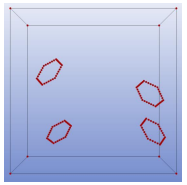

* Scientific Computation Research Center at RPI (Mark Shephard *et al.*)

# Coupling with other codes

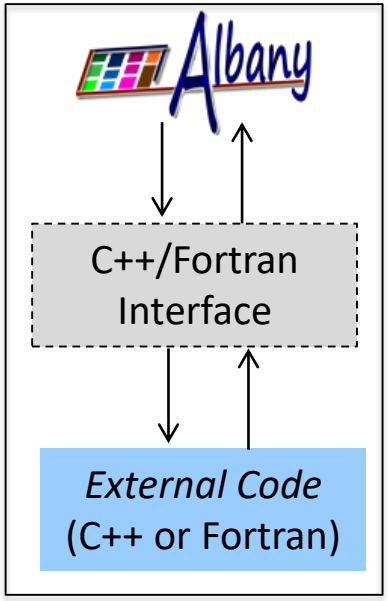
Sandia National Laboratories

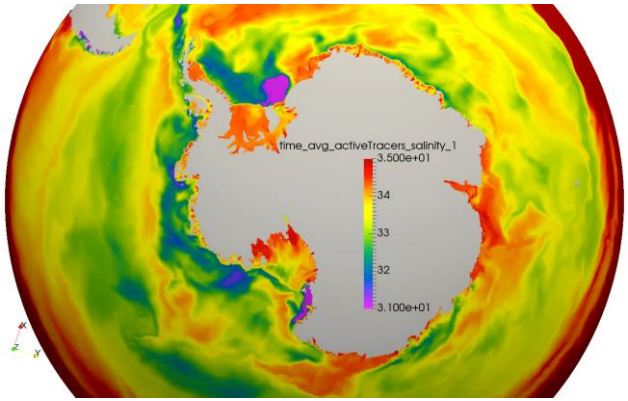
Albany has been **interfaced/coupled** with a number of other codes.

- ***Albany-Peridigm:*** local-nonlocal coupling of continuum mechanics + peridynamics
- ***Albany-ParaDis:*** coupled dislocation dynamics
- ***Albany-PLATO***: Advanced Topology Optimization (ATO)
- ***MPAS-Albany-Land-Ice (MALI)***: ice sheets/coupling to E3SM
- ***CISM-Albany-Land-Ice (CALI)***: ice sheets/coupling to CESM
- ***Albany-PFLOTRAN (Albotran)***: coupled geomechanics problems

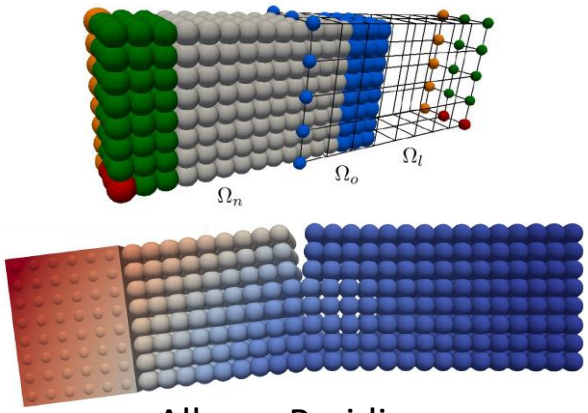


Albany-ParaDis

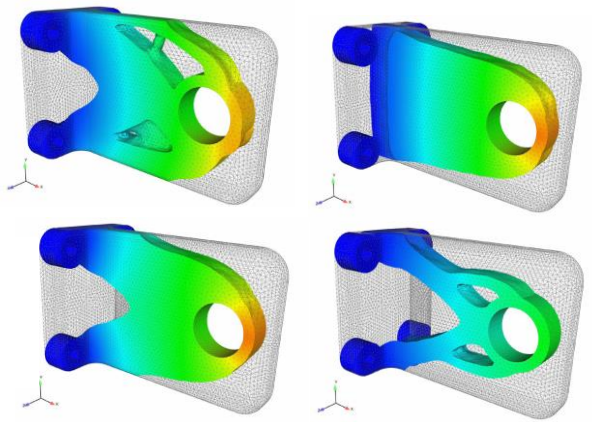



MPAS-Albany (MALI)




Albany-Peridigm

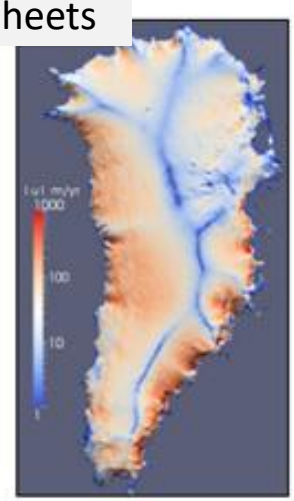


Albany-PLATO (ATO)

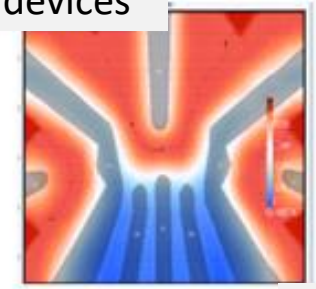
# Summary

***Albany***: open-source, parallel, C++, unstructured-grid, mostly-implicit multi-physics finite element code that demonstrates **AgileComponents** vision and can enable **rapid development** of new physics/algorithms.

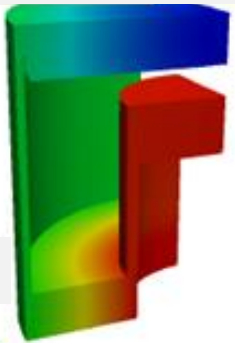
Ice sheets



Quantum devices

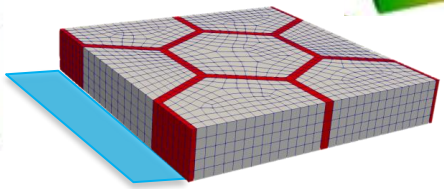


Computational mechanics



Additive manufacturing

Arctic costal erosion



**Github:** https://github.com/SNLComputation/Albany

**Paper:** A. Salinger *et al*. "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", *IJMCE* 14(4) (2016) 415-438.

**Albany User Meeting (AUM):** every ~2 years (TBD)

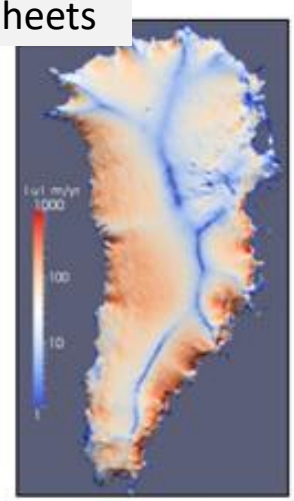
# Summary

***Thank you! Questions?***

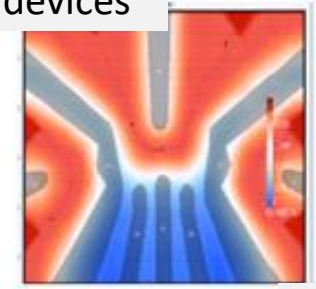
Sandia National Laboratories

***Albany***: open-source, parallel, C++, unstructured-grid, mostly-implicit multi-physics finite element code that demonstrates **AgileComponents** vision and can enable **rapid development** of new physics/algorithms.

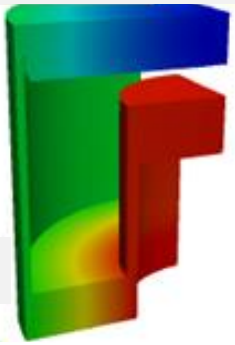
Ice sheets



Quantum devices

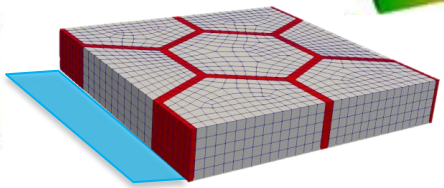


Computational mechanics



Additive manufacturing

Arctic costal erosion



Albany

Trilinos

**Github:** https://github.com/SNLComputation/Albany

**Paper:** A. Salinger *et al*. "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", *IJMCE* 14(4) (2016) 415-438.

**Albany User Meeting (AUM):** every ~2 years (TBD)

# References


[1] A. Salinger *et al*. "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", *Int. J. Multiscale Comput. Engng* 14(4) (2016) 415-438.

[2] M. Heroux *et al.* An overview of the Trilinos project, *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 397–423, 2005.

[3] I. Tezaur, M. Perego, A. Salinger, R. Tuminaro, S. Price. "*Albany/FELIX*: A Parallel, Scalable and Robust Finite Element Higher-Order Stokes Ice Sheet Solver Built for Advanced Analysis", *Geosci. Model Develop.* 8 (2015) 1-24.

[4] M. Perego, S. Price, G. Stadler. "Optimal Initial Conditions for Coupling Ice Sheet Models to Earth System Models", *J. Geophys. Res.* 119 (2014) 1894-1917.

[5] I. Demeshko, J. Watkins, I. Tezaur, O. Guba, W. Spotz, A. Salinger, R. Pawlowski, M. Heroux. "Towards performance-portability of the Albany finite element analysis code using the Kokkos library", *J. HPC Appl*. (2018) 1-23.

[6] J. Watkins, I. Tezaur, I. Demeshko. "A study on the performance portability of the finite element assembly process within the Albany land ice solver", *Lecture Notes in Computational Science and Engineering* (accepted).

[7] A. Mota, I. Tezaur, C. Alleman. "The Schwarz alternating method in solid mechanics", *Comput. Meth. Appl. Mech. Engng.* 319 (2017), 19-51.

[8] W. Spotz, et al., Aeras: A next generation global atmosphere model, Proc. Comput. Sci. 51 (2015) 2097–2106

# References (cont'd)


[9] X. Gao, et al., Quantum computer aided design simulation and optimization of semiconductor quantum dots, J. Appl. Phys. 114 (2013) 1–19.

[10] S. Slattery, P. Wilson, R. Pawlowski, The data transfer kit: a geometric rendezvous-based tool for multiphysics data transfer, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, American Nuclear Society, Sun Valley, ID, 2013,

[11] E. Cyr, J. Shadid, R. Tuminaro, Teko: A block preconditioning capability with concrete example applications in Navier–Stokes and MHD, SIAM J. Sci. Comput. 38 (5) (2016) S307-331.

[12] J. Ostien, et al., A 10-node composite tetrahedral finite element for solid mechanics, Internat. J. Numer. Methods Engrg. (ISSN: 1097-0207) 107 (2016) 1145–1170.

[13] R. Pawlowski, E. Phipps, A. Salinger, Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part I: Template-based generic programming, *Sci. Program.*, vol. 20, pp. 197–219, 2012a.

[14] E. Phipps, R. Pawlowski, Efficient expression templates for operator overloading-based automatic differentiation, in *Recent Advances in Algorithmic Differentiation*, S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther, Eds., Lecture Notes in Computer Science, Springer, pp. 309–319, 2012.

[15] P. Bochev, H. Edwards, R. Kirby, K. Peterson, D. Ridzal, Solving PDEs with Intrepid, *Sci. Program.*, vol. 20, no. 2, pp. 151–180, 2012.

# References (cont'd)

[16] S. Seol, C. Smith, D. Ibanez, M. Shephard, A parallel unstructured mesh infrastructure, *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion*, pp. 1124–1132, doi: 10.1109/SC.Companion.2012.135, 2012.

[17] H. Edwards, A. Williams, G. Sjaardema, D. Baur, W. Cochran, SIERRA toolkit computational mesh conceptual model. Tech. Rep. SAND2010-1192, Sandia National Laboratories, 2010.

Sandia National Laboratories

# Backup Slides

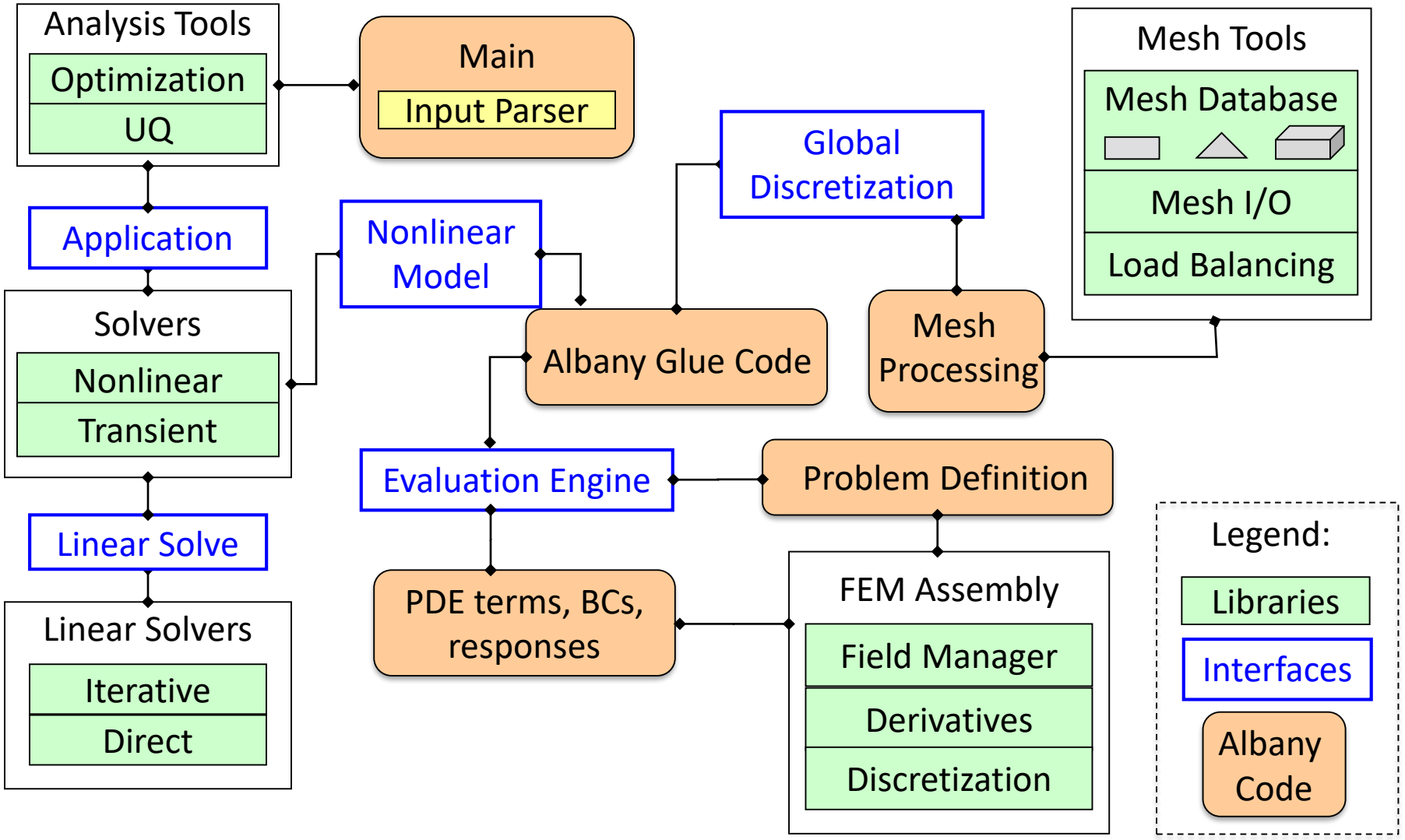
# Albany : a component-based finite element code

2008

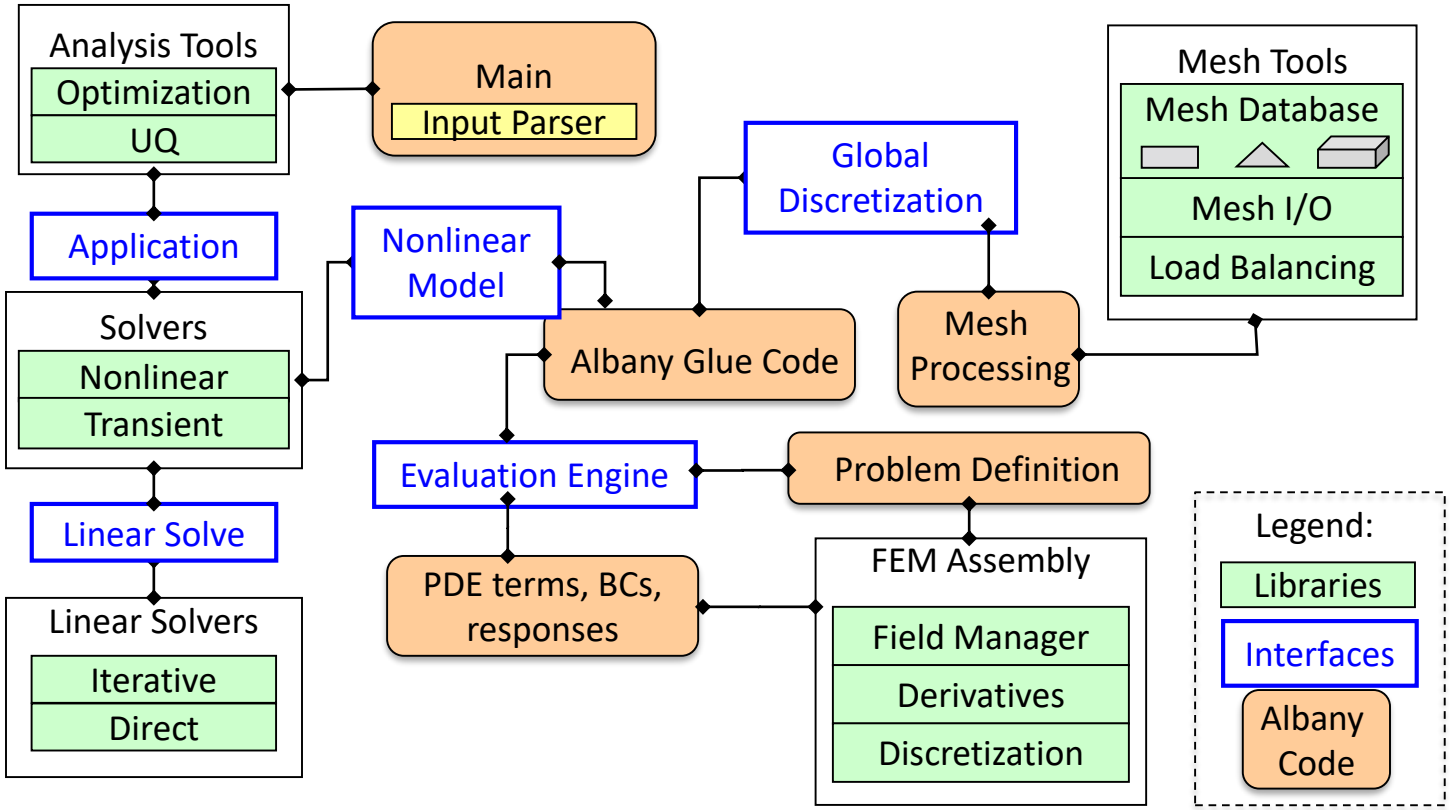
- Started by A. Salinger in 2008 as **first DemoApp** for AgileComponents code development strategy.
- During **next 10 years**, Albany became:
  - a friendly **early adopter** of cutting-edge technology from Trilinos, SCOREC, SierraToolKit, DAKOTA, FASTMath, QUEST, Kitware.
  - a model for a **Trilinos-based** and **Office of Science** application.
  - a demonstration of **transformational analysis** spanning template-based generic programming to optimization and UQ
- **11 years later**, Albany is:
  - an open-source parallel, mostly-implicit **unstructured-grid multi-physics finite element code** that demonstrates the AgileComponents vision by using, maturing, and spinning-off reusable libraries/abstract interfaces.
  - an **attractive environment** for the development of open-source application codes and research.
  - a Meso-App for maturation of **MPI+X programming model** for next generation architecture
  - the code base underlying a number of **research projects** and **applications**.

2019

# Albany code design

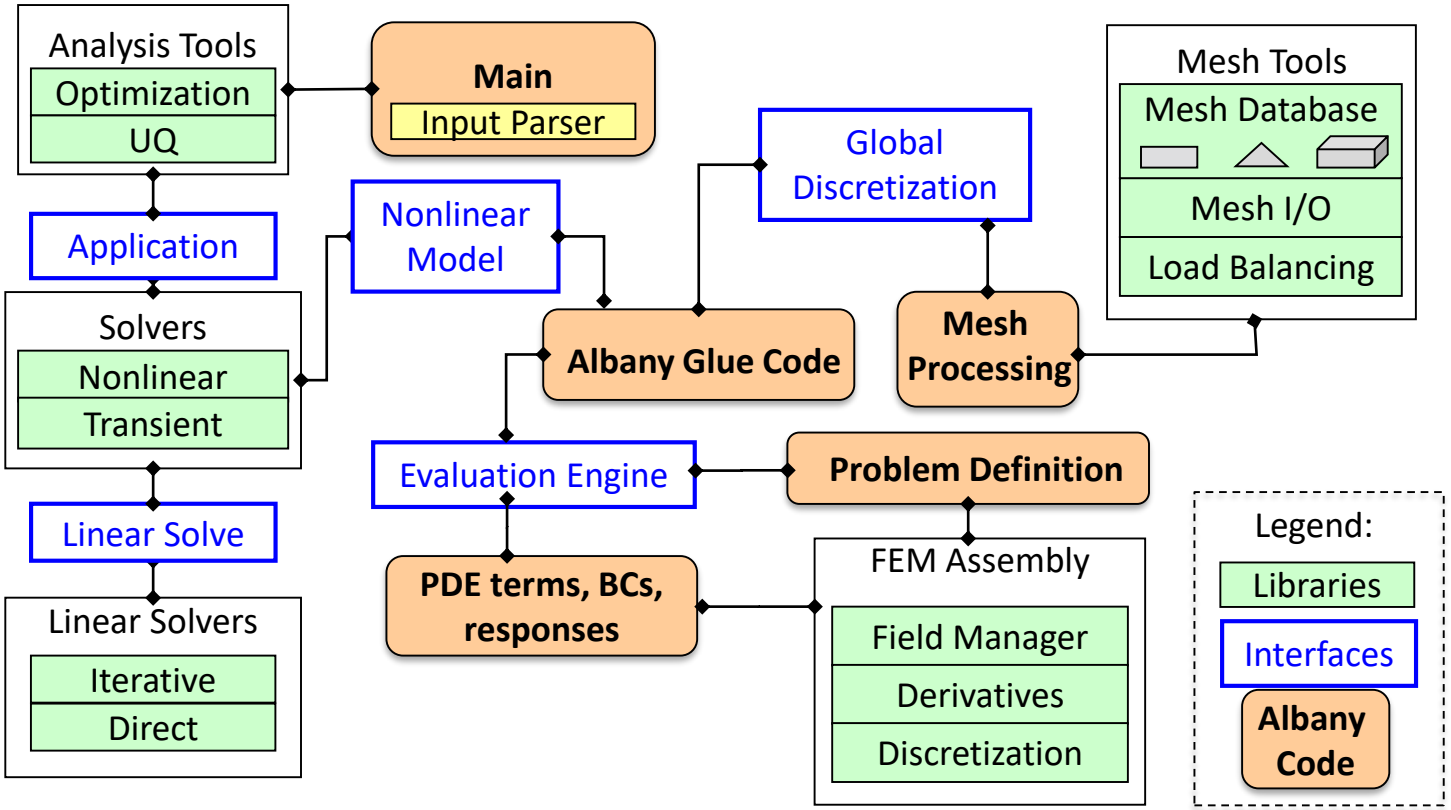


# Albany code design



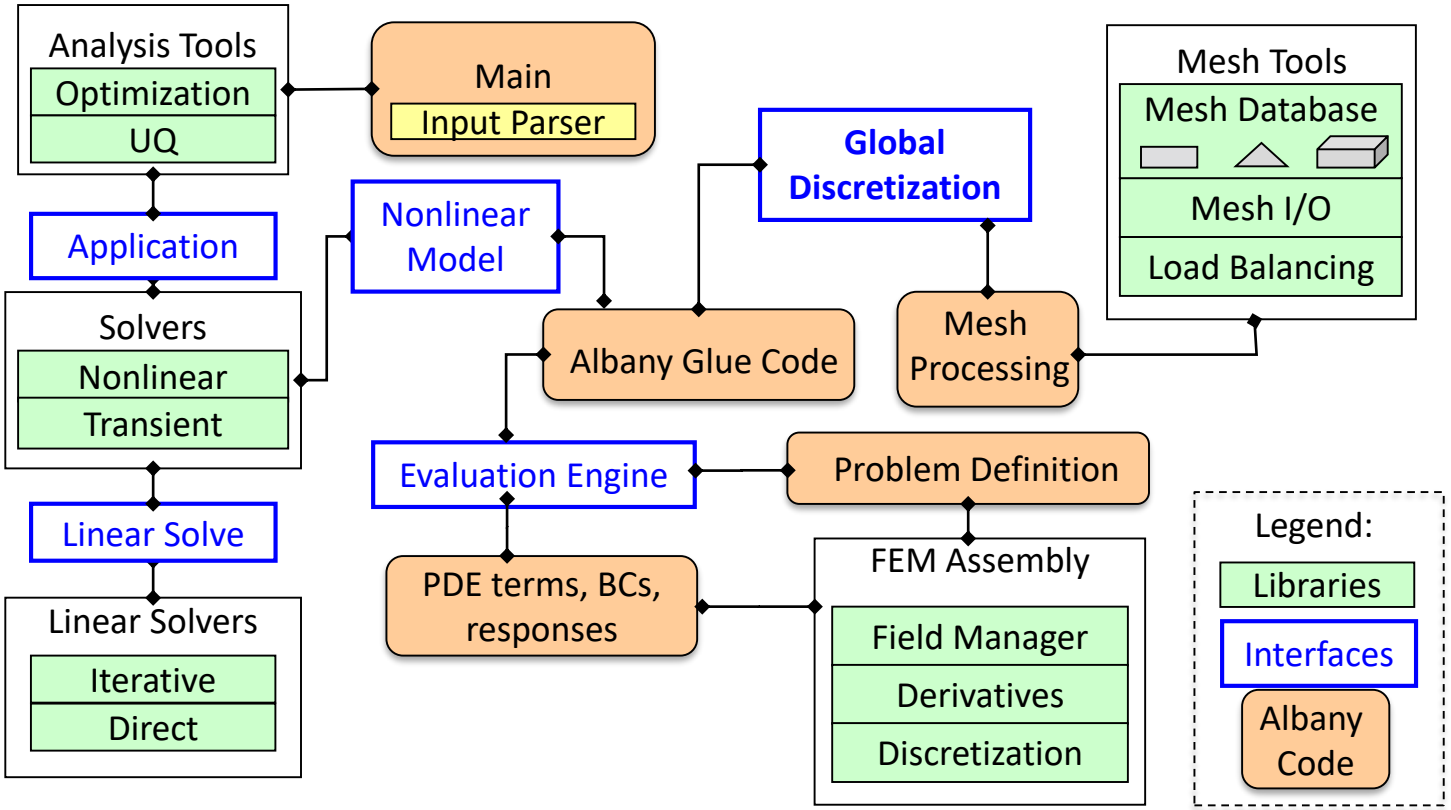
# Albany “glue code”

**“Glue Code”:** driver code integrating components + providing overall capabilities



# Albany “glue code”

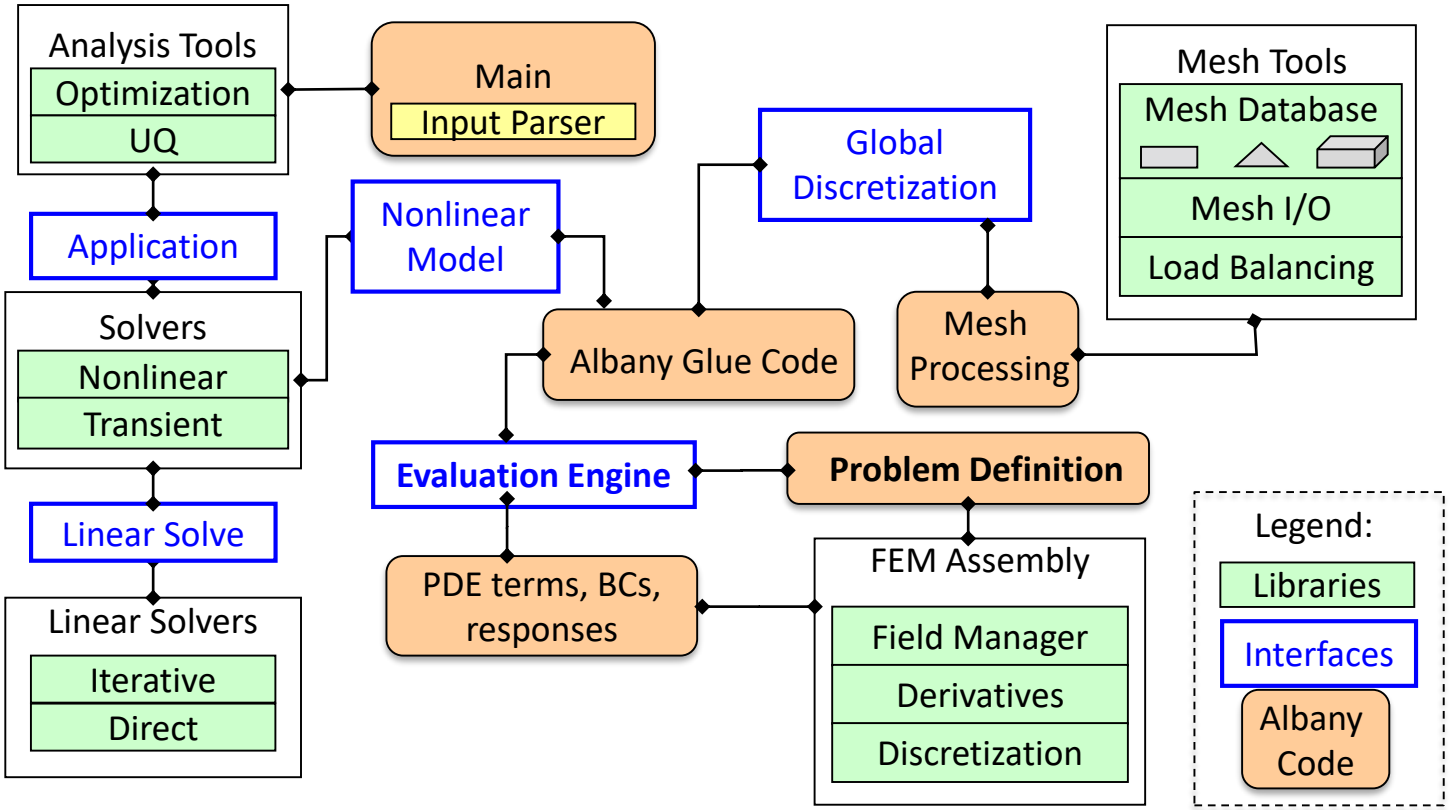
**“Glue Code”:** driver code integrating components + providing overall capabilities




- Depends on **discretization abstraction** (serves as general interface to a mesh service)

# Albany "glue code"

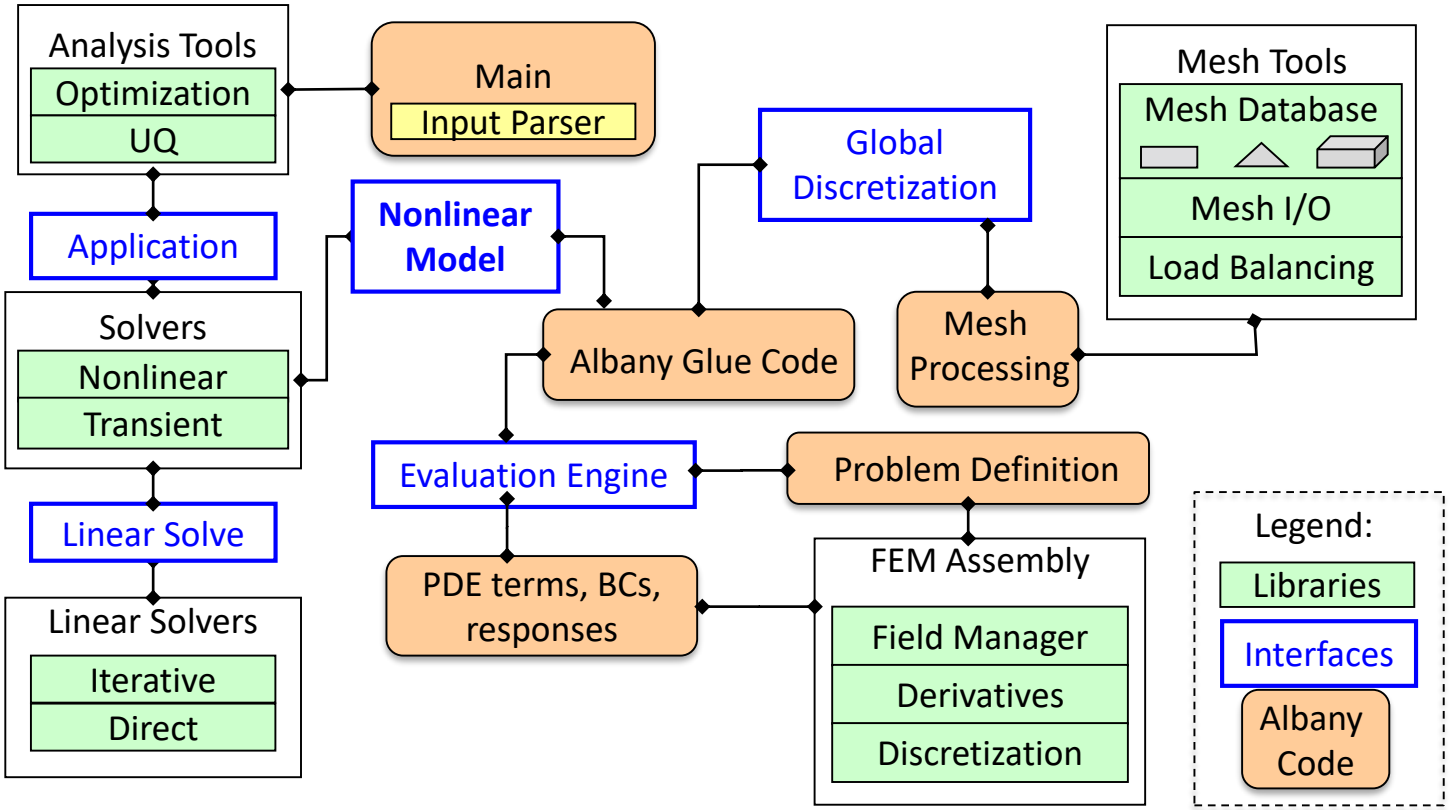
**"Glue Code":** driver code integrating components + providing overall capabilities




- Depends on **discretization abstraction** (serves as general interface to a mesh service)
- Employs **evaluation engine** to construct PDEs, BCs, and response calculations

# Albany "glue code"

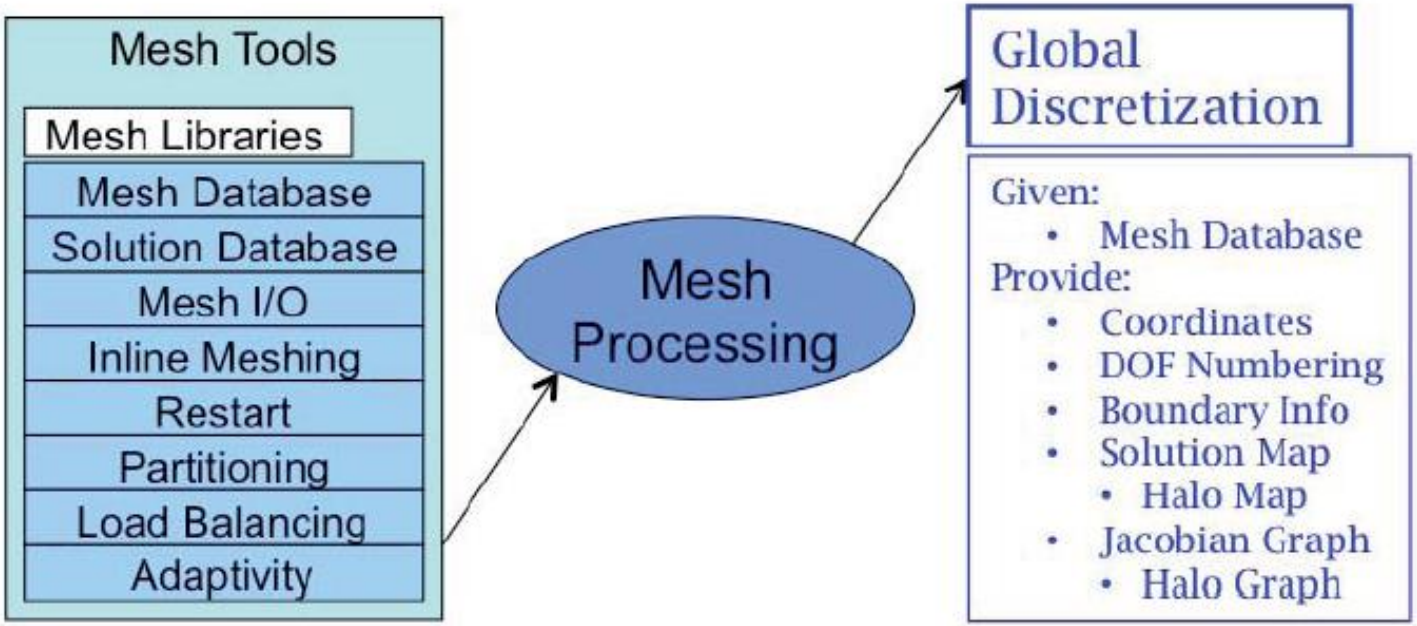
**"Glue Code":** driver code integrating components + providing overall capabilities




- Depends on **discretization abstraction** (serves as general interface to a mesh service)
- Employs **evaluation engine** to construct PDEs, BCs, and response calculations
- Uses physics pieces to satisfy **nonlinear model abstraction** (e.g., compute resid/Jac)

# Global discretization abstraction

- **Mesh framework**: defines geometry, element topologies, connectivities, boundary info, mesh-dependent fields.
- **Global discretization abstraction**: gives the finite element assembly process access to all of the data distribution information required by the linear algebra objects.
- Mesh info is contained in in-memory mesh database accessed through abstract **global discretization interface** class.



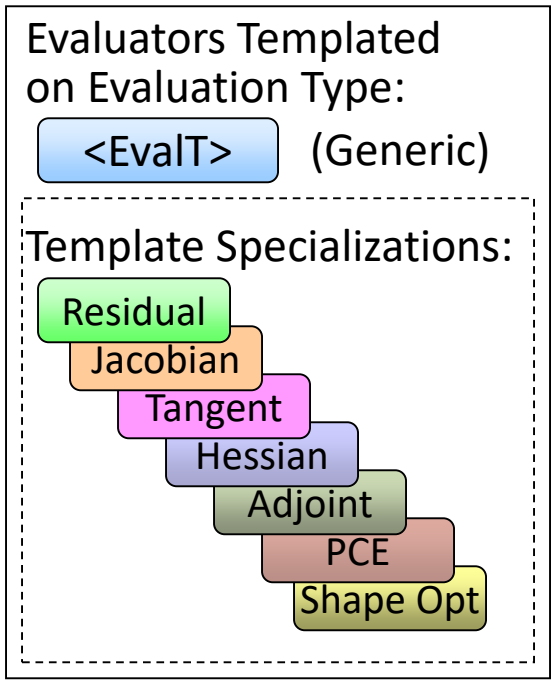
# Problem abstraction & finite element assembly

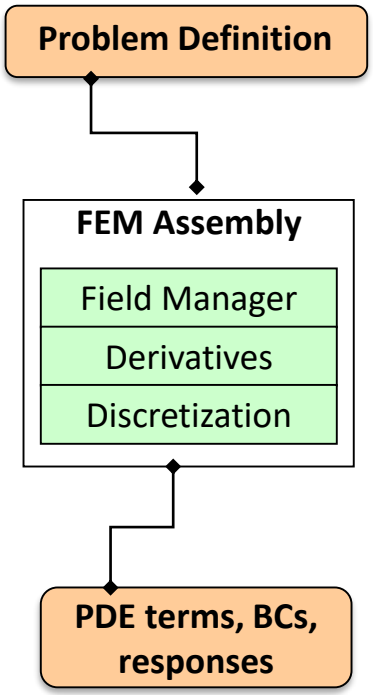
Sandia National Laboratories

**Purpose of problem abstraction & finite element assembly**: given a finite element mesh, evaluate discrete finite element residual, Jacobian, and (if applicable) parameter derivatives.

**3 key ingredients facilitating multi-physics implementations in Albany:**

1. Template-based generic programming (TBGP)
2. Graph-based finite element assembly (FEA)
   - ➢ Handled by Phalanx package
3. Templated-based automatic differentiation
   - ➢ Handled by Sacado package



# Automatic differentiation via Sacado

**Automatic Differentiation (AD**) provides exact derivatives w/o time/effort of deriving and hand-coding them!

- How does AD work? → **freshman calculus**!
  - Computations are composition of simple operations (+, *, sin(), etc.)
  - Derivatives computed line by line then combined via chain rule.
- Great for **multi-physics codes** (e.g., many Jacobians) and **advanced analysis** (e.g., sensitivities)
- Albany uses Trilinos package **Sacado** for AD
  - AD accomplished via **operator overloading** + **templating**: floats/double data types replaced by AD types.

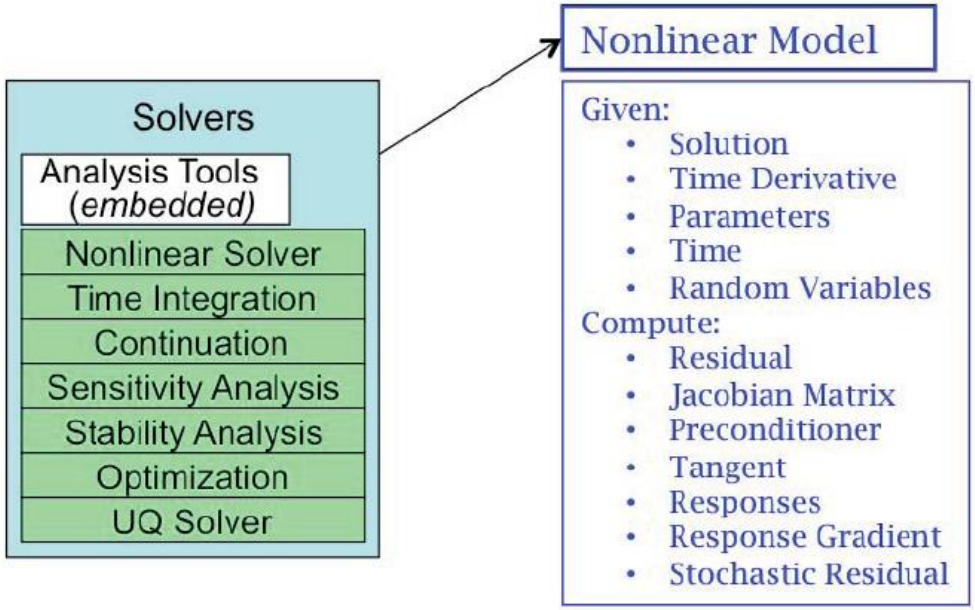
***Automatic Differentiation Example:***

$$y = \sin(e^x + x \log x), \quad x = 2$$

| | | $\frac{d}{dx}$ |
|---|---|---|
| $x \leftarrow 2$ | $\frac{dx}{dx} \leftarrow 1$ | 1.000 |
| $t \leftarrow e^x$ | $\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$ | 7.389 |
| $u \leftarrow \log x$ | $\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$ | 0.500 |
| $v \leftarrow xu$ | $\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$ | 1.301 |
| $w \leftarrow t + v$ | $\frac{dw}{dx} \leftarrow \frac{dt}{dx} + \frac{dv}{dx}$ | 8.690 |
| $y \leftarrow \sin w$ | $\frac{dy}{dx} \leftarrow \cos(w)\frac{dw}{dx}$ | -1.188 |

Derivatives are ***as accurate as analytic computation*** – no finite difference truncation error!

# Nonlinear model abstraction & libraries




**"ModelEvaluator" Abstraction:**

| Given: | Computes: |
|---|---|
| $x$ | $f(\dot{x}, \ddot{x}, x, p, t)$ |
| $\dot{x}$ | $W = \alpha \frac{df}{d\dot{x}} + \beta \frac{df}{dx} + \omega \frac{df}{d\ddot{x}}$ |
| $\ddot{x}$ | $\frac{df}{dp}$ |
| $p$ | $g$ |
| $t$ | $\frac{dg}{dx}$ |
| | $\frac{df}{dp}$ |

- Access to the **embedded solvers** in Trilinos requires satisfaction of **ModelEvaluator** (nonlinear model) abstraction.
- Interface is **general** to accommodate computation of Jacobians, user-defined preconditioners, and stochastic Galerkin expansions.
- **Advanced capabilities**: embedded UQ (Stokhos), optimization (ROL), homotopy continuation (LOCA).

$f$ = residual; $x$ = solution vec;
$p$ = parameters; $g$ = responses;
$t$ = time; $W$ = Jacobian

# Core building blocks of Albany

Sandia National Laboratories

- Component-based design.
- Template-based generic programming.
- Assembly/field evaluation via Phalanx.
- Automatic differentiation.
- Discretizations/meshes, mesh adaptivity.
- Solvers, time-integration schemes.
- Performance-portable kernels.
- Software quality tools: git cmake, ctest, CDash.

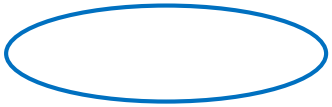
# Graph-based finite element assembly (FEA)

Sandia National Laboratories

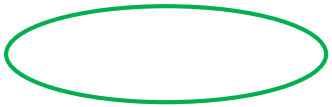
**Assembly** of physics pieces comes down to the evaluation of a **directed acyclic graph (DAG)** of computations of field data.

**Phalanx package:** Local field evaluation kernel designed for assembly of arbitrary equation sets (i.e. evaluating residuals/Jacobians).

Template-specialized evaluators

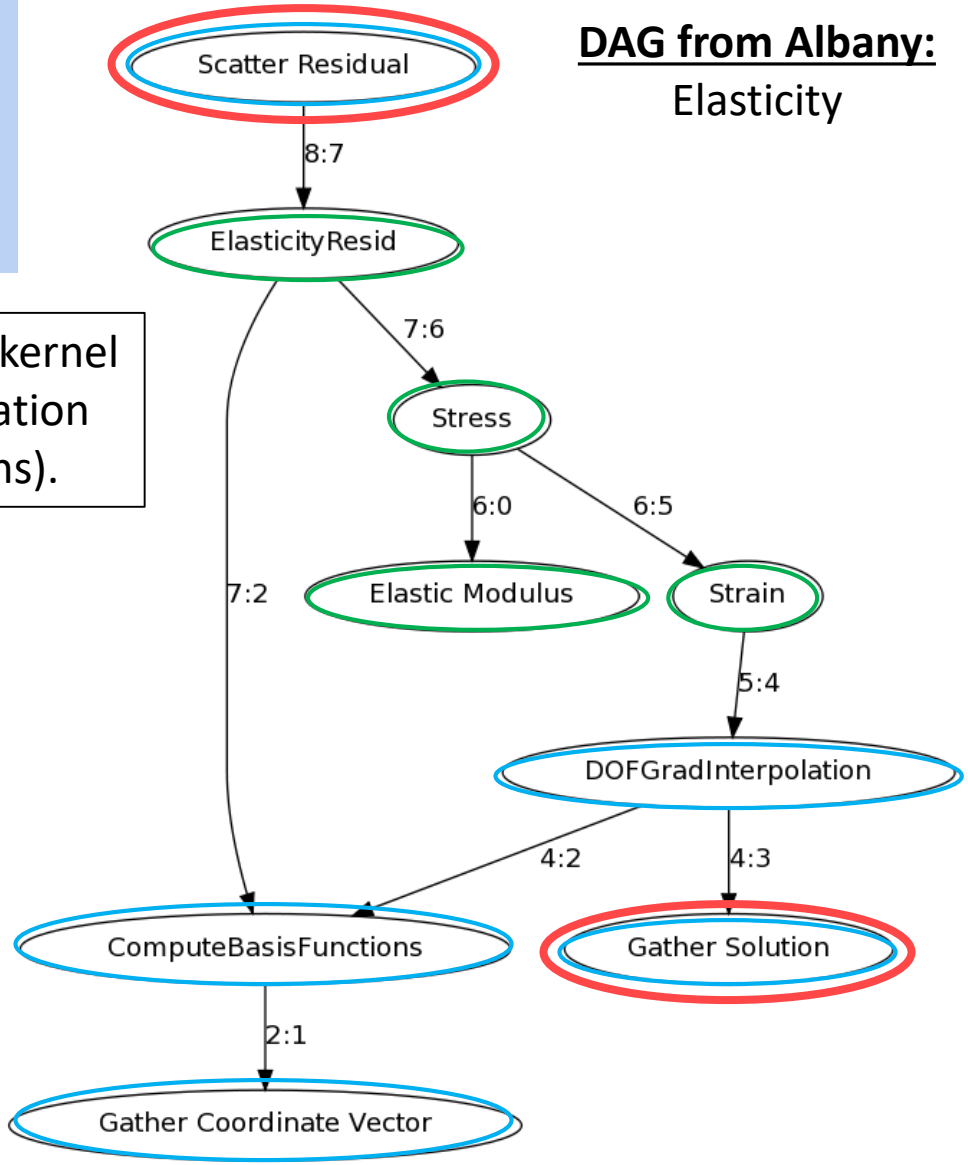


Evaluator common to all FEAs



Problem-specific evaluator

**DAG from Albany:**
Elasticity




R. Pawlowski, Phalanx web site, http://trilinos.sandia.gov/packages/phalanx/ 2015.

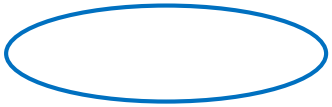
# Graph-based finite element assembly (FEA)

Sandia National Laboratories

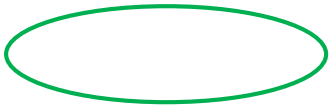
**Assembly** of physics pieces comes down to the evaluation of a **directed acyclic graph (DAG)** of computations of field data.

**Phalanx package:** Local field evaluation kernel designed for assembly of arbitrary equation sets (i.e. evaluating residuals/Jacobians).

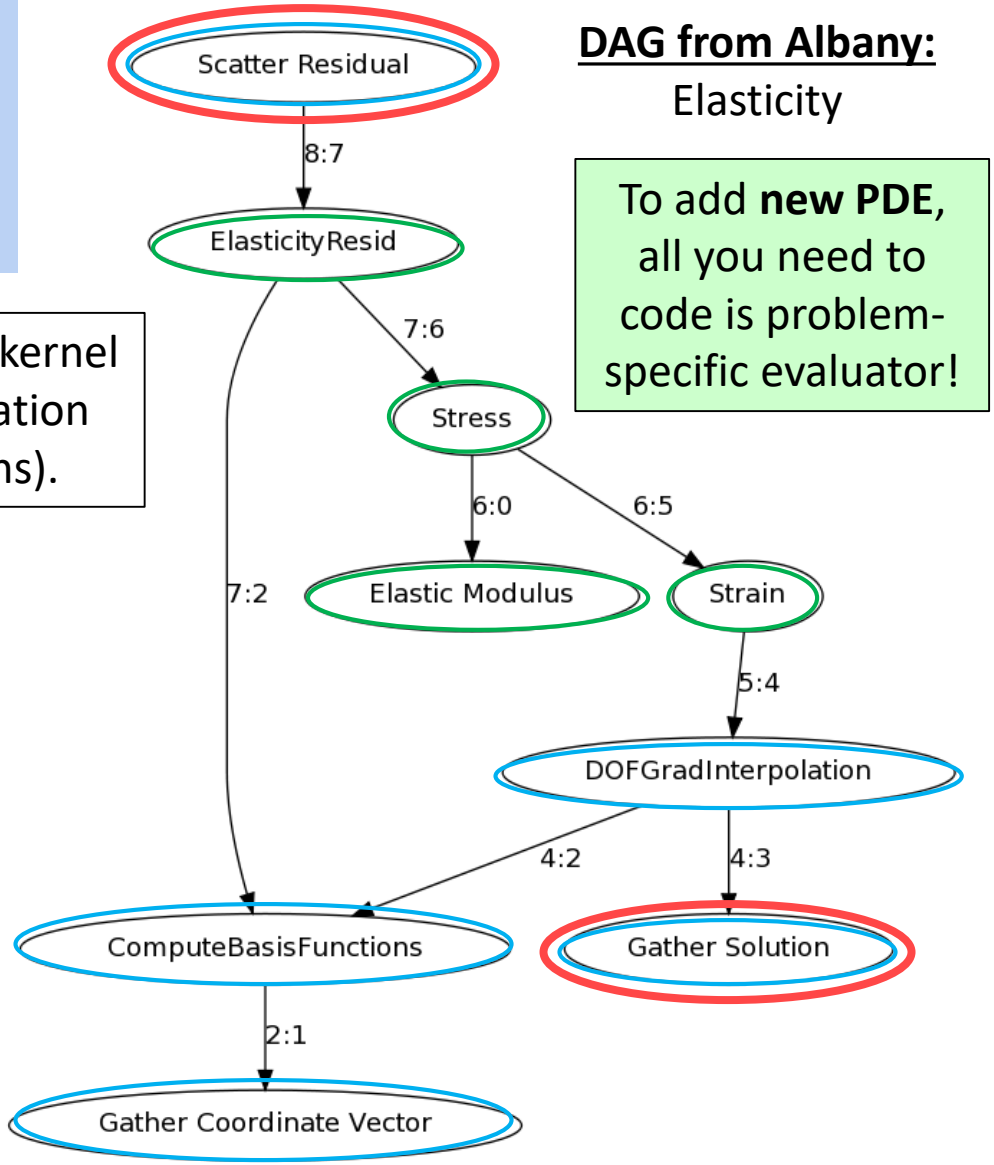
Template-specialized evaluators



Evaluator common to all FEAs



Problem-specific evaluator

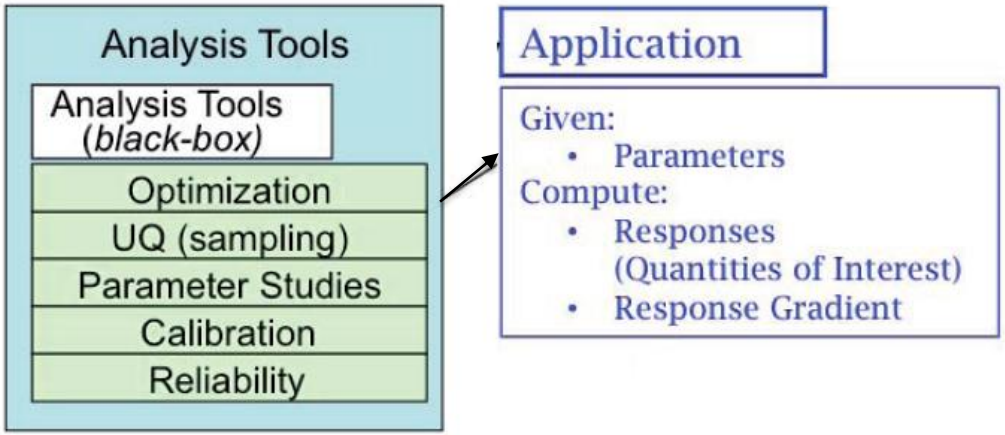



**DAG from Albany:**
Elasticity

To add **new PDE**, all you need to code is problem-specific evaluator!

R. Pawlowski, Phalanx web site, http://trilinos.sandia.gov/packages/phalanx/ 2015.

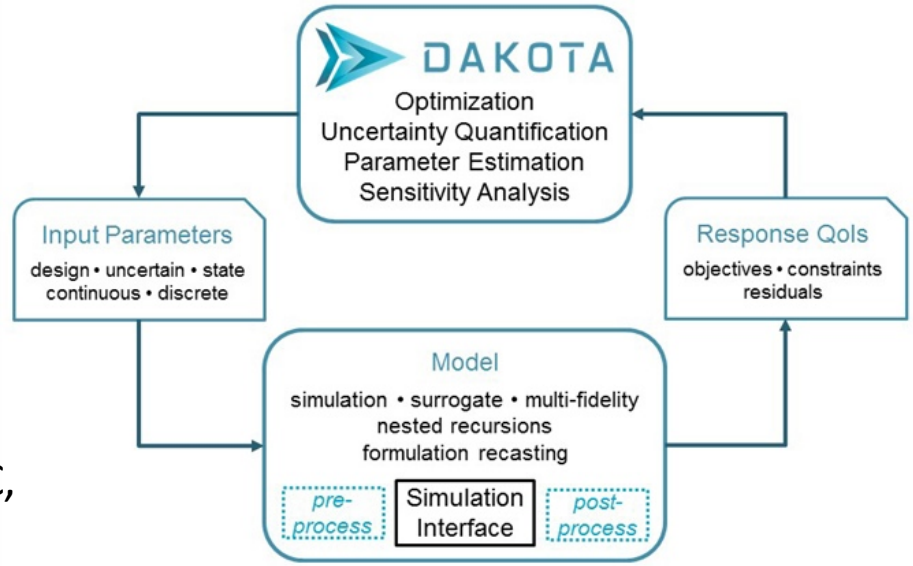
# Analysis tools abstraction & libraries




"**Black box**" analysis tools at top level of software stack can perform a single forward solve, sensitivity analysis, parameter studies, bifurcation analysis, optimization, and UQ runs.

**Optimization** and **UQ** can be done through DAKOTA:

- **Optimization algorithms**: gradient-based local algorithms, pattern searches, and genetic algorithms, etc.
- **UQ algorithms**: Latin hypercube stochastic sampling, stochastic collocation, PCE, MCMC, etc.



# Libraries/algorithms whose development was significantly aided by Albany

Sandia National Laboratories

**Libraries Developed in Albany:**

- Piro
- TriKota
- MiniTensor
- Razor (MOR)
- buildAgainstTrilinos

**Libraries Driven by Albany:**

- Stokhos Embedded UQ
- Semi-Coarsening AMG
- PAALS
- Advanced Topological Opt
- Embedded Ensembles
- CUBIT Mesh-Morpher

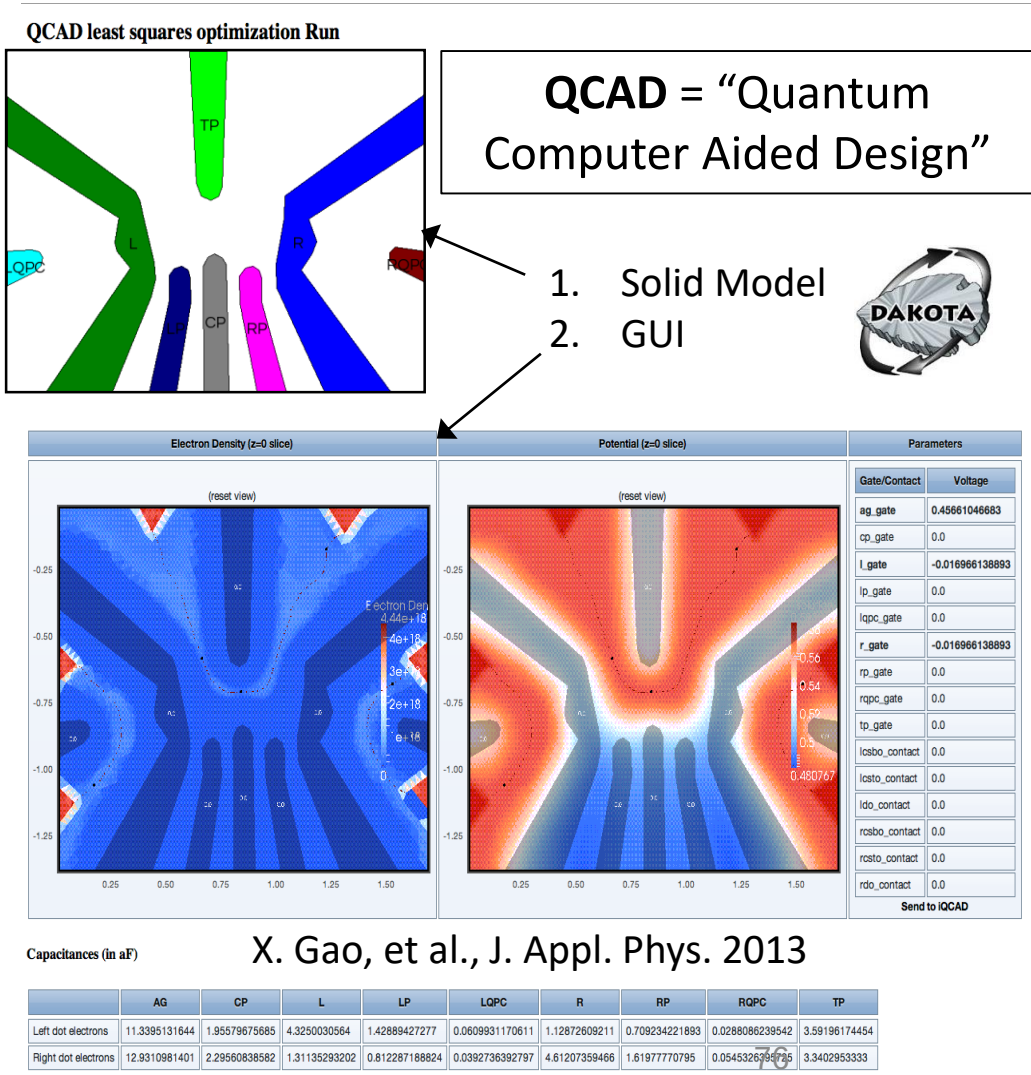
**Libraries Matured in Albany:**

- Tempus
- Phalanx
- STK
- ModelEvaluator
- Stratimikos
- TPetra
- PUMI
- ROL
- DTK
- Intrepid2/Kokkos
- DynRankView
- *And counting…*

# Quantum device modeling (QCAD)

Albany enabled the rapid stand-up of a world-class **quantum device design tool.**

- **Application**: quantum computing
- **Objective:** simulation/optimization of semiconductor quantum double dots
  - ➢ Provide fast feedback on which device layouts are most likely to lead to **few-electron** behavior
- **Key to QCAD's success**: interfaces
  - ➢ Various **multi-physics couplings** of Poisson + Schrodinger
  - ➢ DAKOTA* for **optimization**.
- QCAD is used by experimentalist in Sandia's **world-class experimental facilities (CINT)** as design tool for quantum **device fabrication**
  - ➢ **iQCAD**: GUI for experimentalists

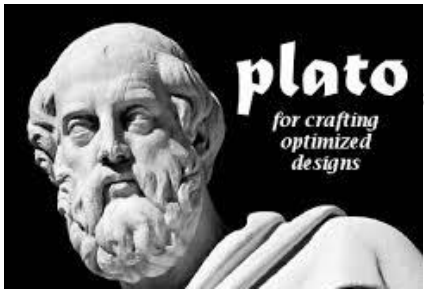



| | AG | CP | L | LP | LQPC | R | RP | RQPC | TP |
|---|---|---|---|---|---|---|---|---|---|
| Left dot electrons | 11.3395131644 | 1.95579675685 | 4.3250030564 | 1.42889427277 | 0.0609931170611 | 1.12872609211 | 0.709234221893 | 0.0288086239542 | 3.59196174454 |
| Right dot electrons | 12.9310981401 | 2.29560838582 | 1.31135293202 | 0.812287188824 | 0.0392736392797 | 4.61207359466 | 1.61977770795 | 0.0545326395725 | 3.3402953333 |



* https://dakota.sandia.gov

# Advanced Topology Optimization (ATO)




Coupling of Albany code and **PLATO*** engine for **optimization-based topology optimization**.

## Goals:

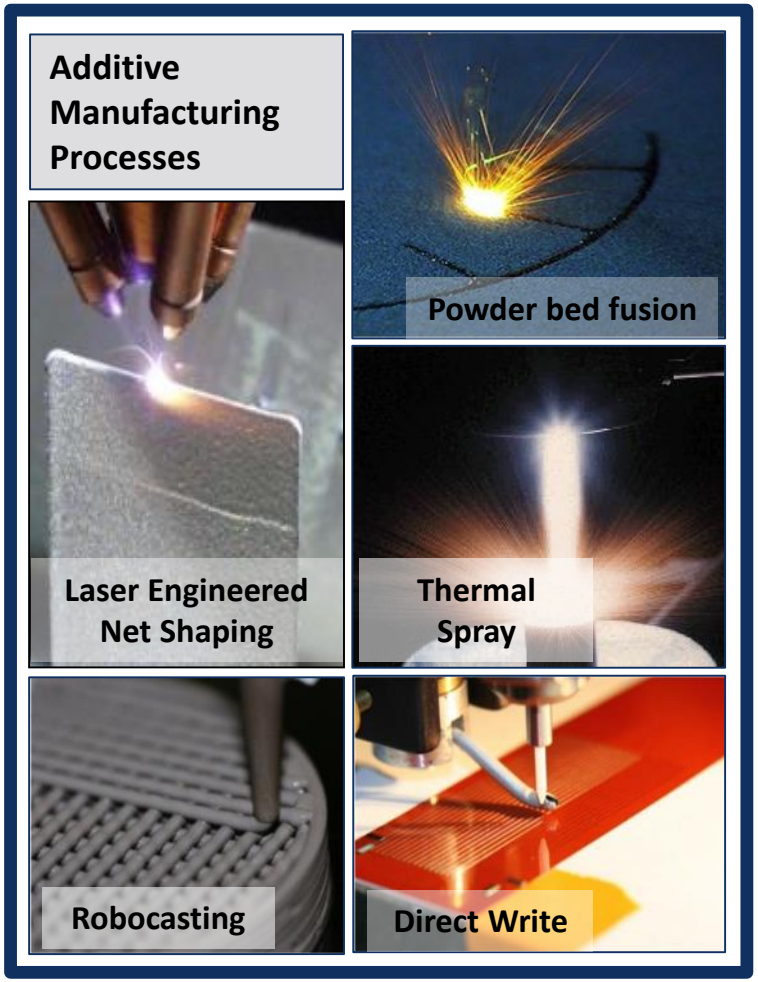
- **Qualification**: assure quality
- **Design**: effectively utilize AM

## PDE-constrained optimization:

- **Physics**: elastostatics, Poisson
- **Objectives**: compliance, p-norm
- **Constraints**: volume/mass

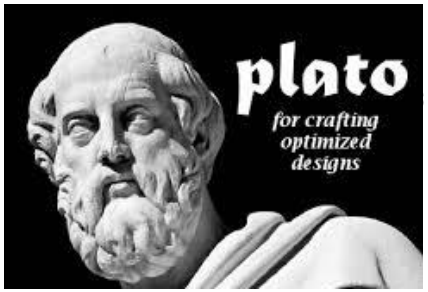
## Multiple simultaneous Albany runs can inform a single design optimization by PLATO:

- Albany implements objective + gradient evaluation, optimization loop
- **New "meshless" ATO capability**: allows user to include arbitrarily many simultaneous load cases (linear thermal/electrical, mechanical)




* Topology optimization-based design environment developed by SNL.

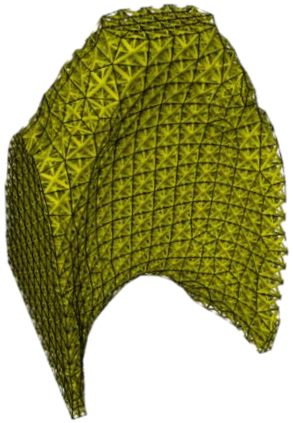
# Advanced Topology Optimization (ATO)

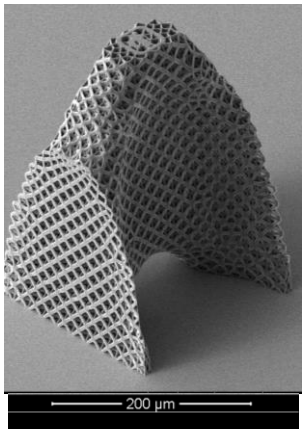



Coupling of Albany code and **PLATO*** engine for **optimization-based topology optimization**.

## Goals:

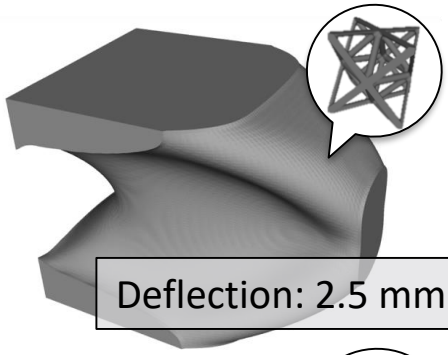
- **Qualification**: assure quality
- **Design**: effectively utilize AM

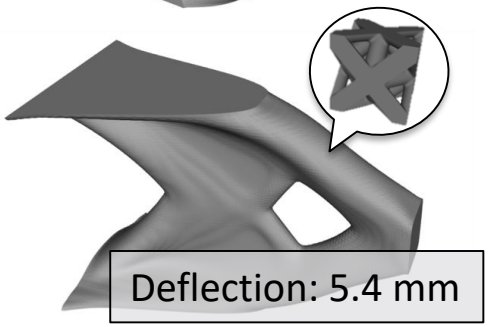


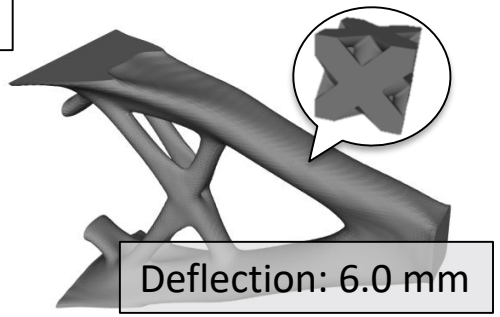



## PDE-constrained optimization:

- **Physics**: elastostatics, Poisson
- **Objectives**: compliance, p-norm
- **Constraints**: volume/mass




***Right:*** cellular structures with optimized stiffness

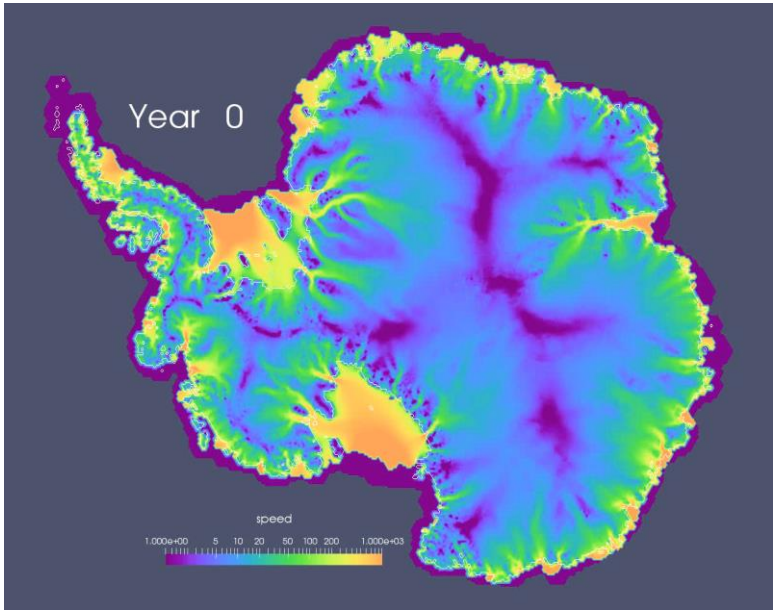
## Multiple simultaneous Albany runs can inform a single design optimization by PLATO:

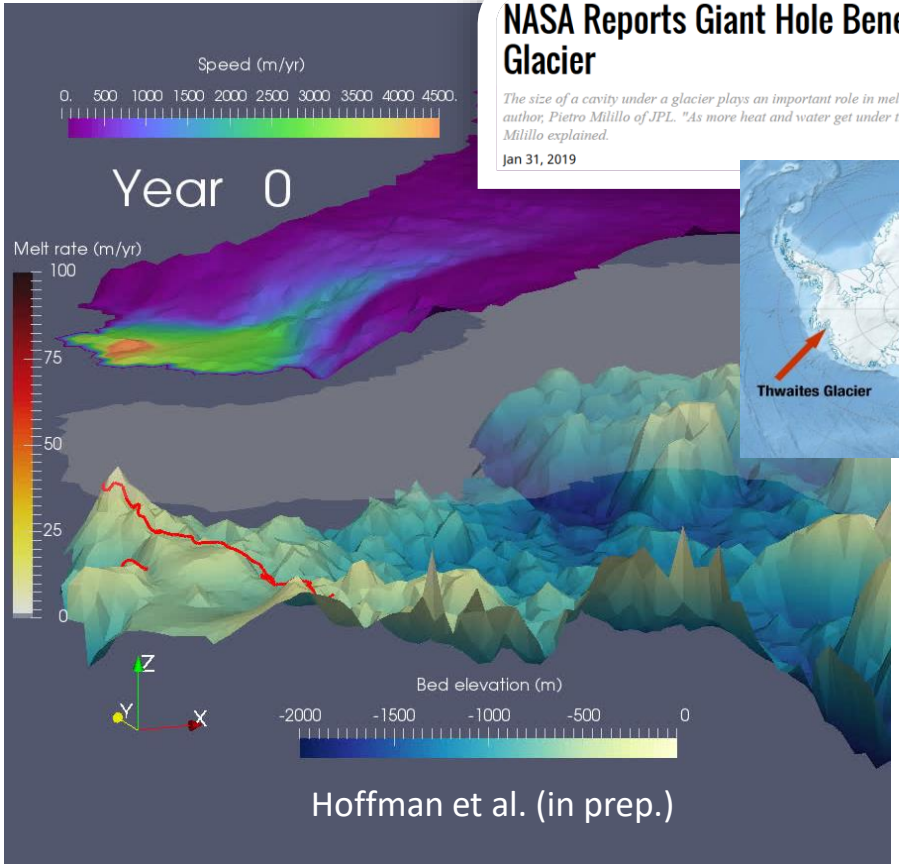
- Albany implements objective + gradient evaluation, optimization loop
- **New "meshless" capability**: geometry defined in constructive solid geometry (CSG) format and meshed inline
  - Allows user to include arbitrarily many simultaneous load cases (linear thermal/electrical, mechanical, etc.)

* PLAtform for Topology Optimization: topology optimization-based design environment developed by SNL.

# Ice sheets: Albany Land-Ice (ALI)

Albany enabled the **rapid development** of a production **land-ice dycore** for providing **actionable predictions** of **21st century sea-level rise** as a part of the DOE Energy Exascale Earth System Model (E3SM).







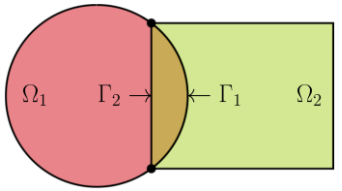
**Above**: ABUMIP-Antarctica experiment

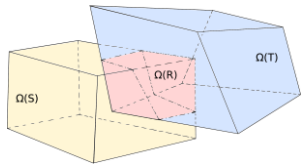
**Right**: Thwaites glacier retreat under parametrized submarine melting.

# Multi-scale coupling via Schwarz

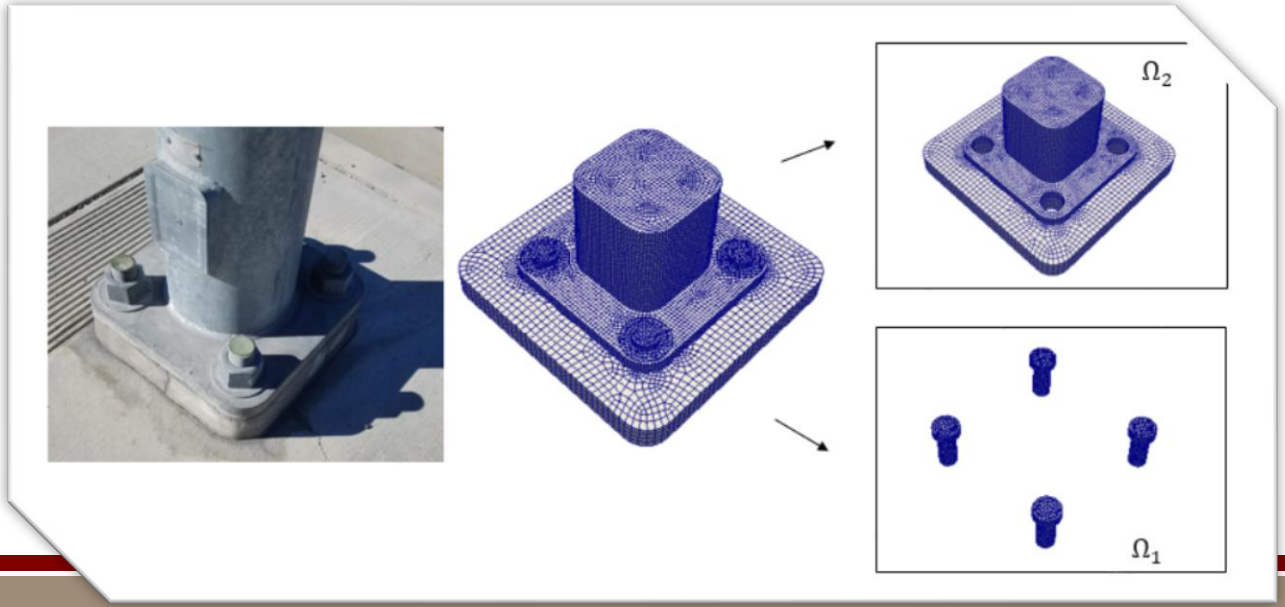
A **domain decomposition alternating-Schwarz**-based method has been developed in Albany for **concurrent multi-scale coupling** in solid mechanics.

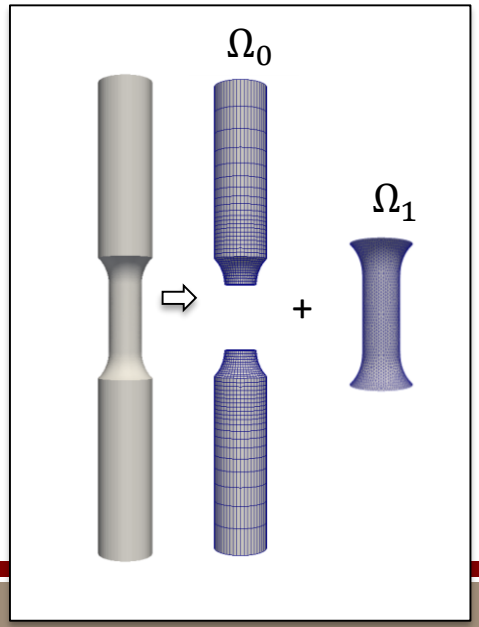
- **Crux of Method:** use solutions in simple domains to iteratively build a solution for the more complex domain.
- **Targeted application**: failure of **bolted components**.
- **"Plug-and-play" framework**: simplifies meshing complex geometries!
  - Couple regions with **different non-conformal meshes**, **element types, levels of refinement, solvers/time-integrators.**




https://github.com/ORNL-CEES/DataTransferKit

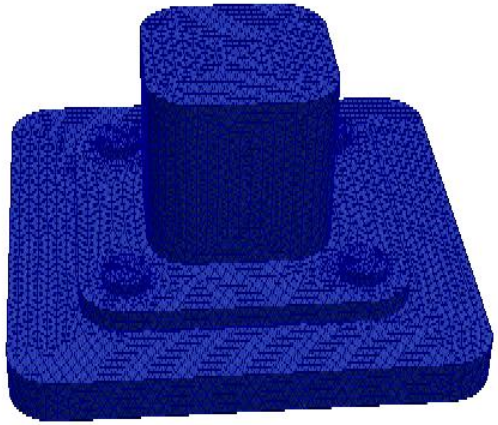


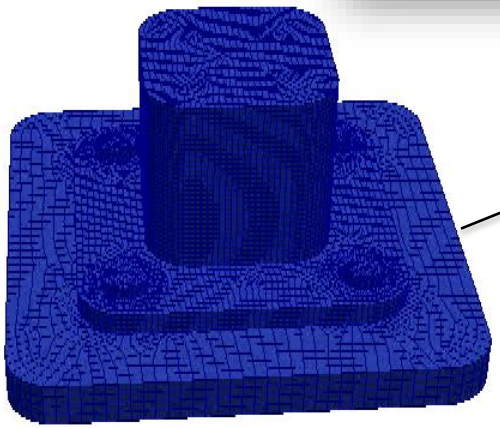
# Multi-scale coupling via Schwarz

A **domain decomposition alternating-Schwarz**-based method has been developed in Albany for **concurrent multi-scale coupling** in solid mechanics.

**See talk by A. Mota**: MS 350, Fri. Mar. 1, 10:10-10:30AM, Room 302B



**Single domain**



**Schwarz coupling** of hex (parts) + composite tet 10 (bolts) elements ($J_2$ material model from LCM suite)

**Future work:** extend method for **multi-physics** coupling

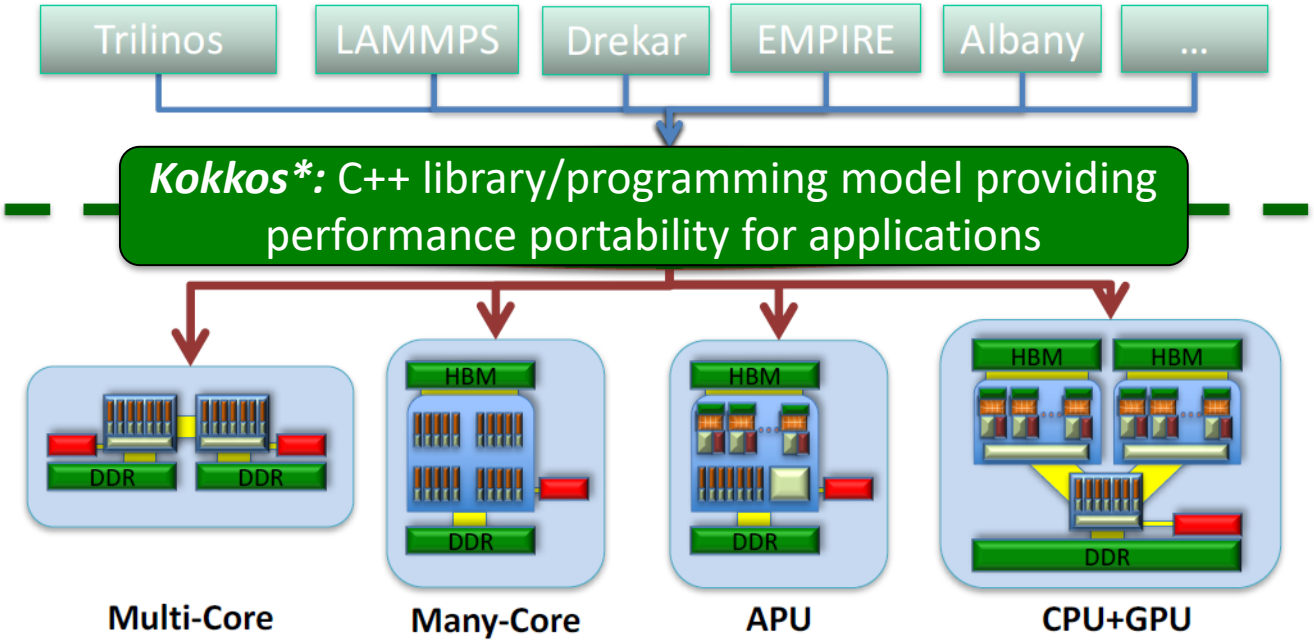
A. Mota *et. al.* "The Schwarz alternating method in solid mechanics", *CMAME.* 319 (2017), 19-51.

# Performance-portable FEM

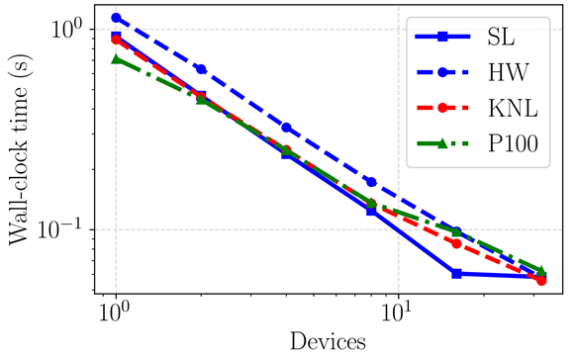
**Talk by J. Watkins,**
MS 121, Tues. Feb. 26

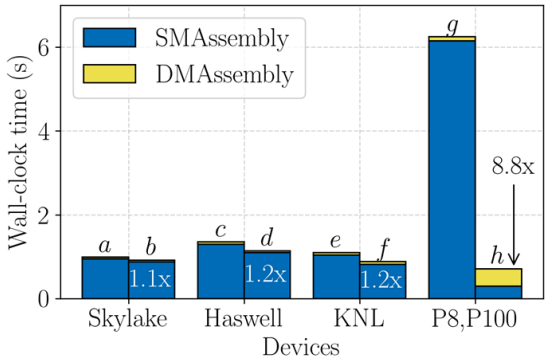
Sandia National Laboratories






- Algorithm is written **once** for multiple architectures
- Template parameters specify **optimal data layout** for a given architecture.



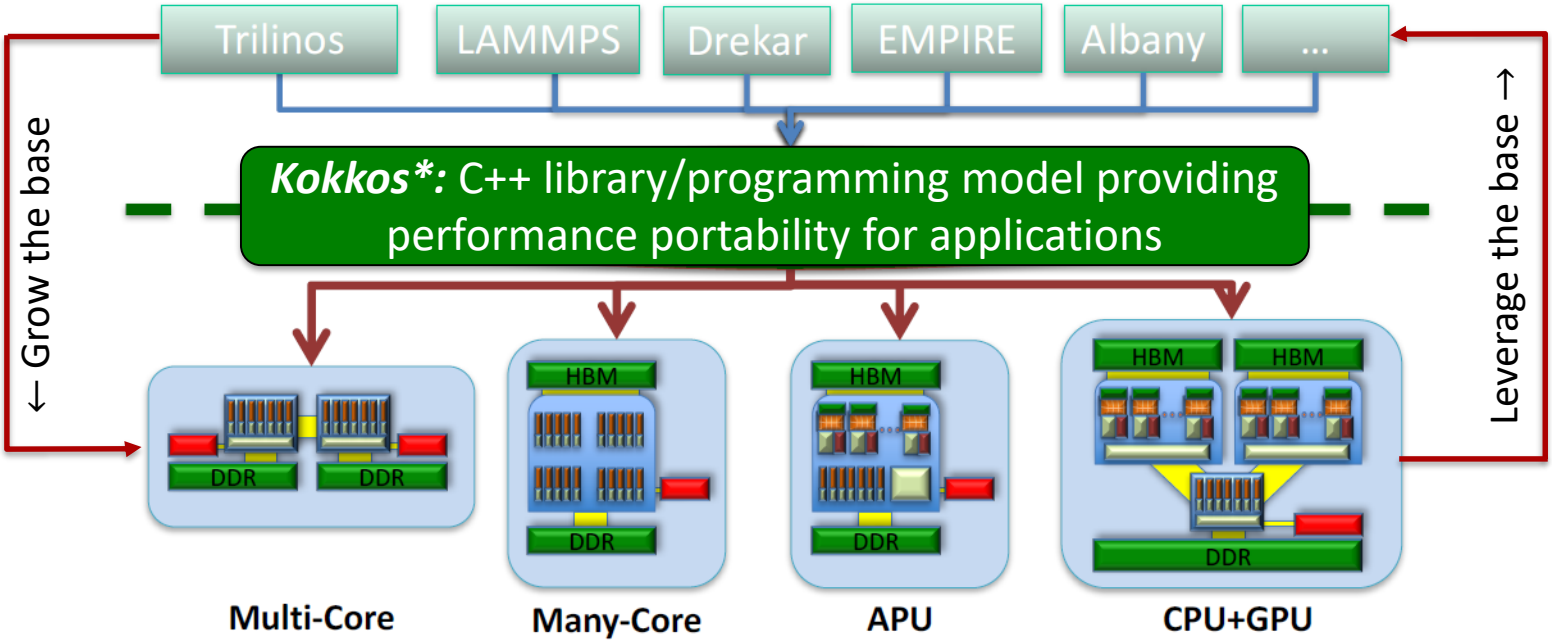

* **Kokkos github repo:** https://github.com/kokkos/Kokkos I. Demeshko *et. al.*, 2018.

# Performance-portable FEM

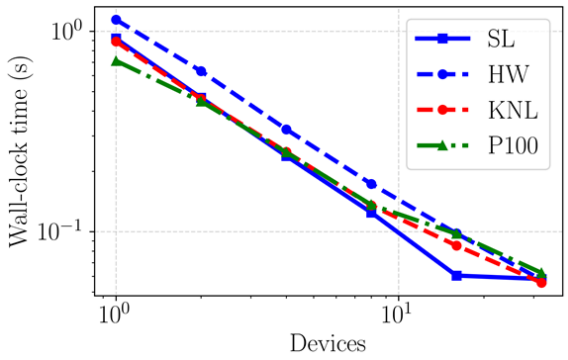
**Talk by J. Watkins,**
MS 121, Tues. Feb. 26

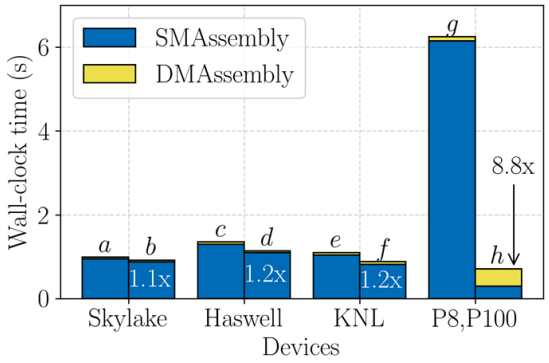
Sandia National Laboratories

**Performance-portability** of Albany FEA achieved using ***Kokkos;***
Albany usage has in turn led to ***Kokkos improvements***




- Algorithm is written **once** for multiple architectures
- Template parameters specify **optimal data layout** for a given architecture.

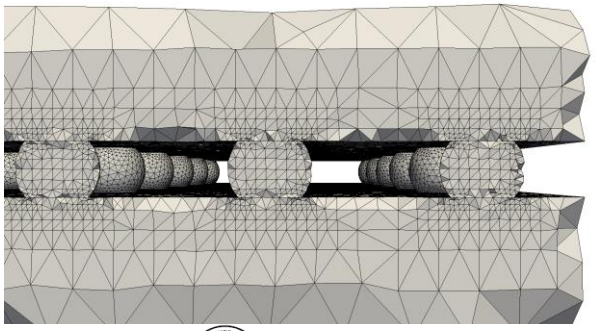


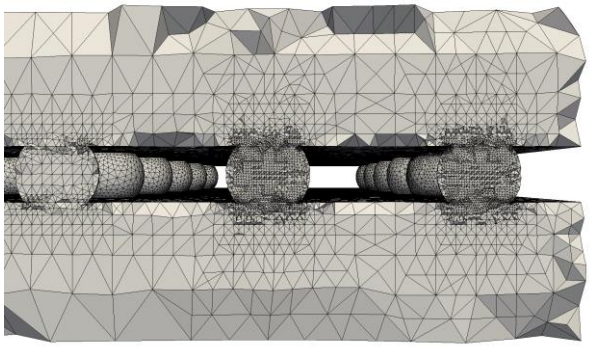

* **Kokkos github repo:** https://github.com/kokkos/Kokkos I. Demeshko *et. al.*, 2018.

# In-memory mesh adaptation

Collaboration with **SCOREC***: development of **mesh adaptation** capabilities in Albany to enable **multi-scale/multi-physics adaptive simulation**



SCOREC Rensselaer

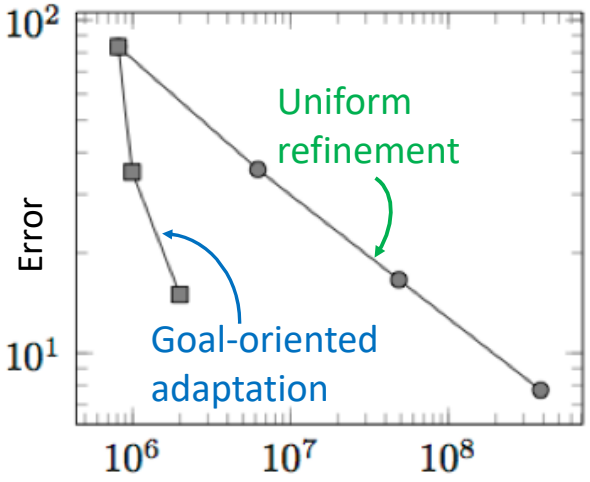


**PAALS (Parallel Albany Adaptive Loop with SCOREC)**

- Fully-coupled ***in-memory adaptation, solution transfer***.
- ***Parallel mesh infrastructure*** and services via ***PUMI***.
- ***Dynamic load balancing*** (ParMetis/Zoltan, ParMA).
- Automated ***parallel goal-oriented adaptive simulation***
  - Use **adjoint solution** to drive mesh adaptation
  - ~100× DoF-efficiency observed
  - Scaling out to at least 8K MPI ranks
- ***Performance portability*** to GPUs via ***Kokkos***.
- ***Applications:*** 3D manufacturing, creep/plasticity in large solder joint arrays, coupled dislocation dynamics (Albany + ParaDis), …

PUMI
Parallel Unstructured Mesh Infrastructure

Error in von Mises stress

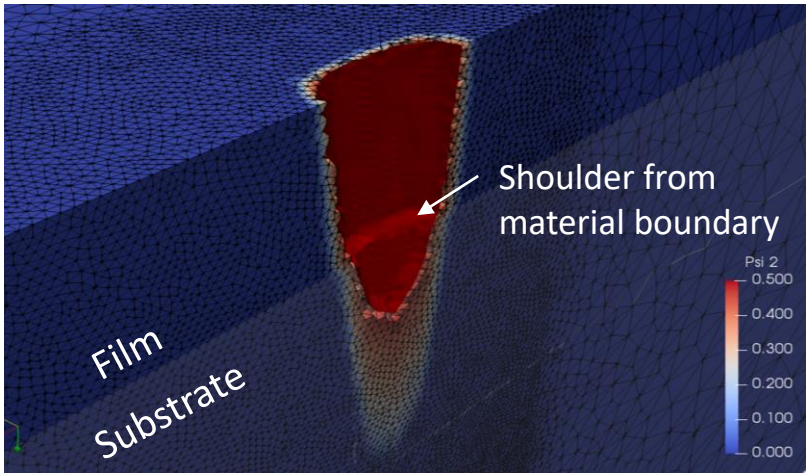


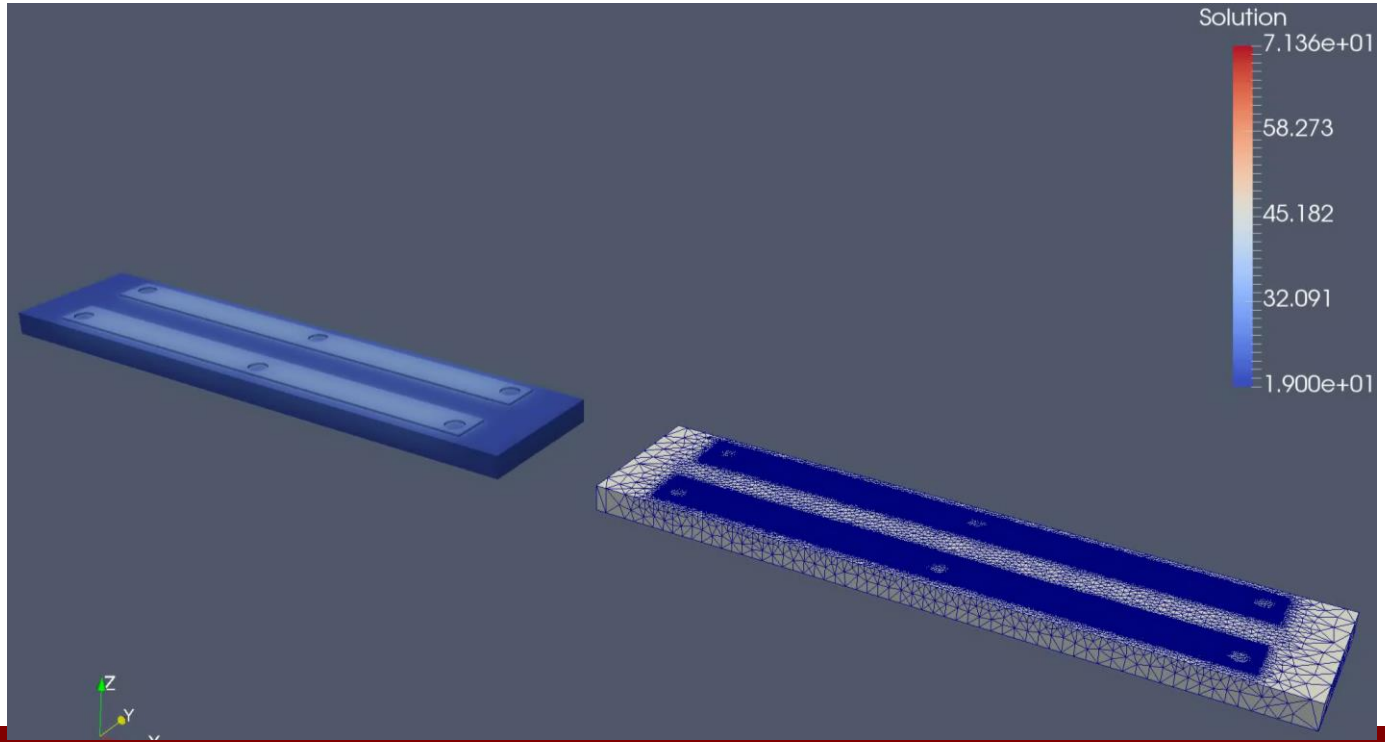

* Scientific Computation Research Center at RPI (Mark Shephard *et al.*)

# 3D manufacturing

- **Additive** & **subtractive** capabilities
- Employs **advanced adaptive meshing** and **evolving geometries** (using Symmetrix)
- Coupling with **feedback control**

***Right***: simulation of subtractive manufacturing with picosecond laser*



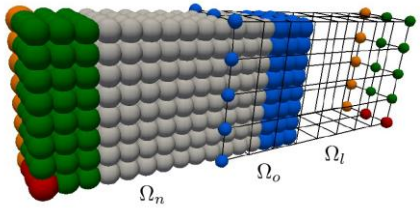

***Left:*** Additive manufacturing simulation showing temperature in evolving geometry

* Maniatty, et al., *Comp. Mech.*, 62:273, 2018.

# Albany-Peridigm

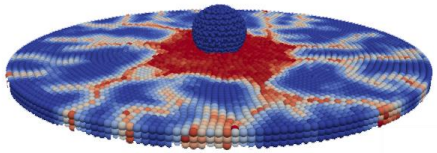
Peridigm Albany ROL

Sandia National Laboratories

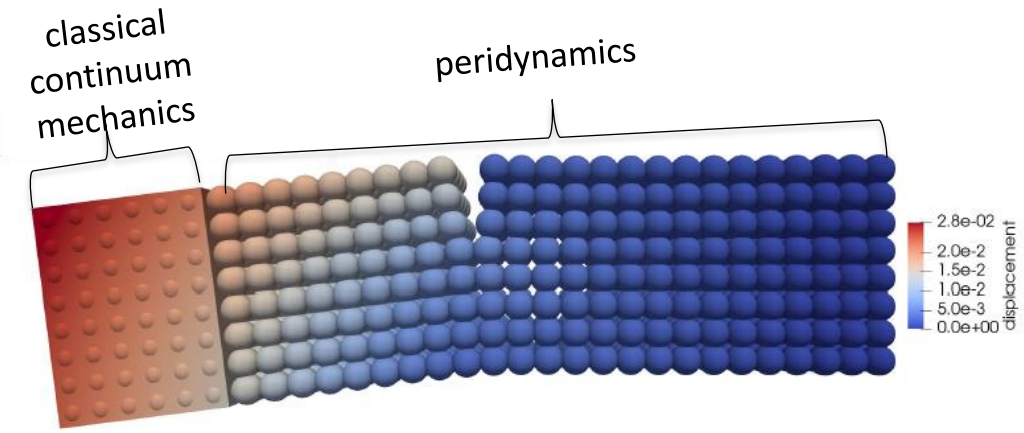


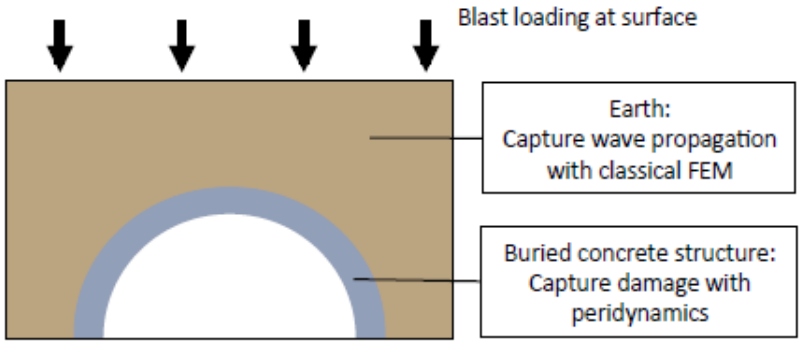

**Local-nonlocal** coupling for **integrated fracture modeling & multi-physics peridynamics** simulations

- **Peridynamics**: nonlocal extension of continuum mechanics that remains valid at discontinuities/cracks



- **Peridigm** = open-source* peridynamics code
  - **Nonlocal meshfree approach** (Silling *et al.*, 2005).
- "**Best of both worlds**" by combining FEM + peridynamics: peridynamics applied in regions susceptible to material failure, easy delivery to applications via FEM.
- **Optimization-based** local-to-nonlocal **coupling** using ROL (D'Elia *et al.* 2016).



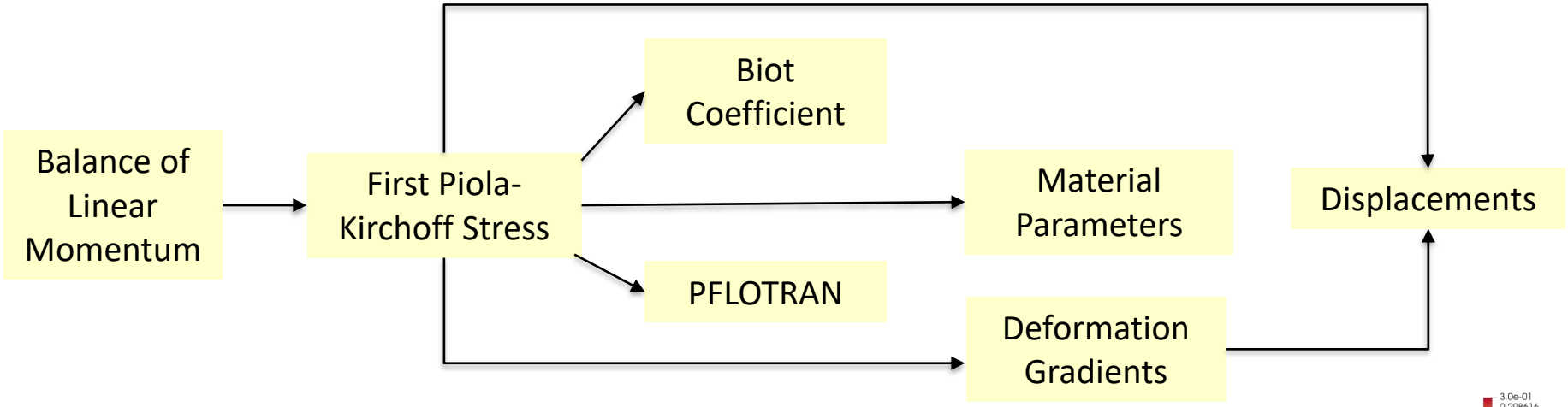




* **Peridigm github repo**: https://github.com/peridigm/peridigm.

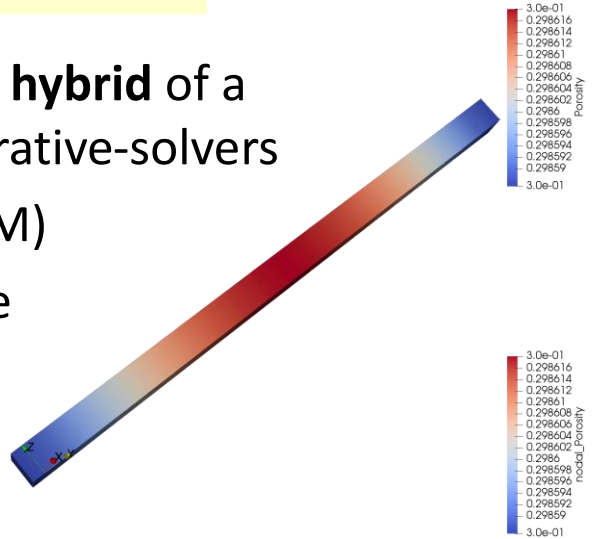
# Albany-PFLOTRAN (Albotran)

Sandia National Laboratories

**Albotran** seeks to create a **multi-physics** geomechanical application that couples the flow response in **PFLOTRAN** with a mechanical response from **Albany**.




- Albany + PFLOTRAN coupling strategy can be viewed as **hybrid** of a fully-coupled implicit solver + more loosely coupled iterative-solvers
  - ➢ Integrates extensive **domain expertise** (Albany/LCM)
  - ➢ Specialized solvers are **not required** for either code

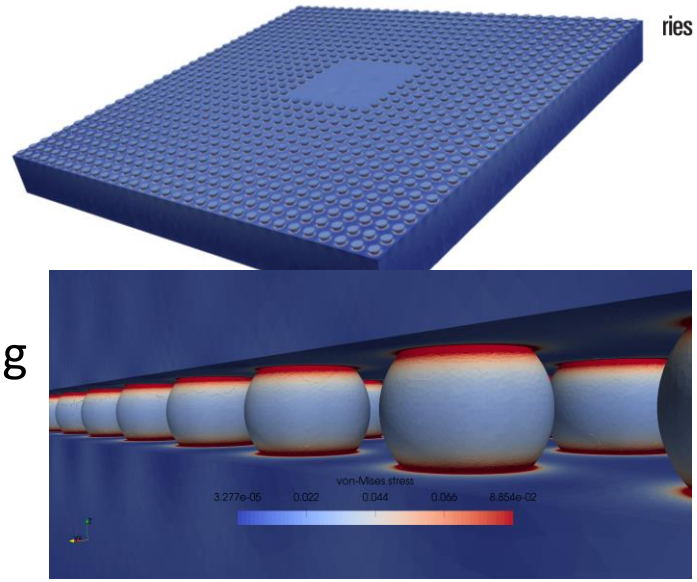


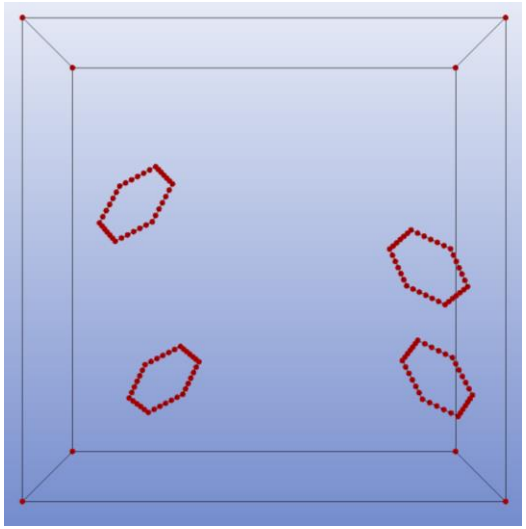

**Right**: Albotran consolidation problem results. Porosities tracked independently in each code are identical

# Mesh adaptation applications

## Creep/Plasticity in Large Solder Joint Arrays*

- Strategic reliability process in semiconductor manufacturing
- Automated workflows with locally refined meshing
- Novel materials models
- Scaling out to 16K processors, 1B+ elements



## Coupled Dislocation Dynamics

- Integrates Albany and ParaDis
- Computes dislocation dynamics (DD) in complex geometry
- Allows intersection of dislocations with free surfaces
- ***Left***: prismatic dislocation loops in finite domain

* Li, et al. *Comp. Mech.*, 62:323, 2018. Bloomfield, et al. *Eng. with* Comp., 33: 509, 2017.

# Work in progress

Sandia National Laboratories

## Performance portability:

- **Code optimizations** for finite element assembly
- Performance portable **solvers** [WIP by Trilinos team]




## Infrastructure work:

- Refactor of code to use **block data structures** to facilitate **multi-physics** coupling
  - "**Plug-and-play**" different PDEs within Albany
  - Ability to use **block preconditioners** (Teko*)
- Add support for **mixed finite elements**
  - Can be accomplished via incorporation of **Panzer** and **DOFManager.**

* E. Cyr *et. al.* SIAM J. Sci. Comput. 2016.

# Work in progress


## Application-driven development:

***Land-ice:***

- Improved **basal hydrology** models for land-ice.
- **Level set** formulation to track better the calving.
- **Uncertainty quantification** (Bayesian inference, forward propagation).
- Seeking funding for developing **solid-mechanics-based ice fracture/ calving models** for improved ice sheet models.

***LCM:***

- Modeling of structural components in **hypersonic vehicles** with large mechanical and thermal loads (USC).
- Enhancement of **subtractive manufacturing** capabilities in Albany (RPI).

***ATO:***

- **Meshless** topology optimization using Albany-PLATO.

***Much more…!***