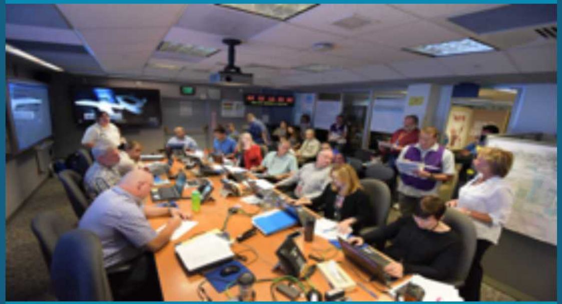
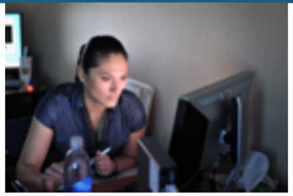
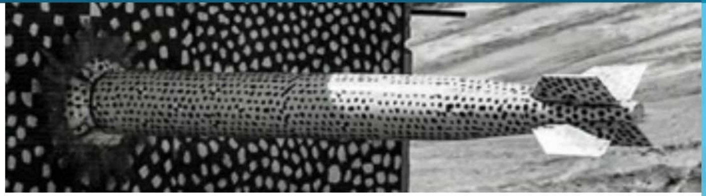
Sandia National Laboratories

SAND2019-12912C

Nonlinear Analysis Product Area Update







Roger Pawlowski

Trilinos User Group Meeting
October 22nd, 2019
Albuquerque, NM

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

The Team

- Roscoe Bartlett
- Sidafa Conde
- Drew Koury
- Sean Miller
- Curtis Ober
- Roger Pawlowski
- Mauro Perego
- Eric Phipps
- Denis Ridzal
- Nate Roberts
- Andrew Steyer
- Irina Tezaur
- Greg von Winckel

Nonlinear Analysis Product

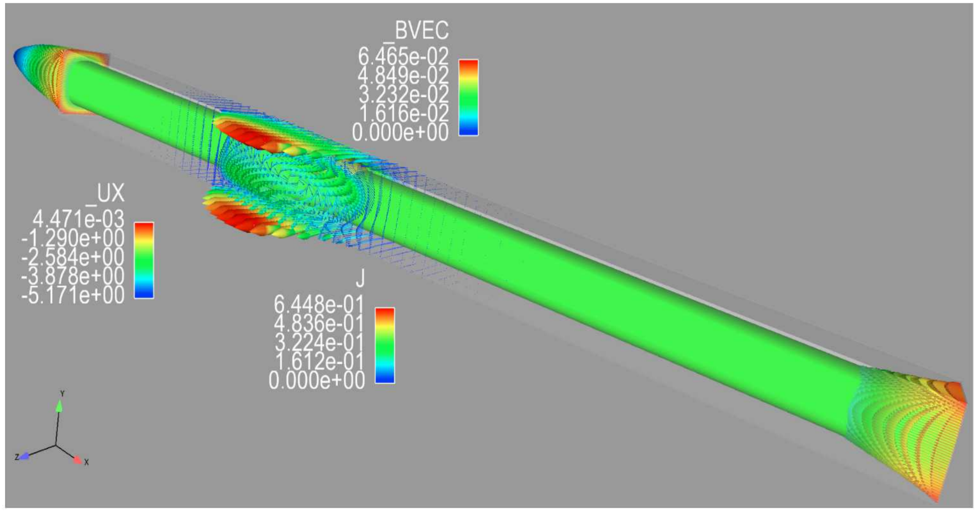
Package	Description	Contact	Status
NOX	Globalized Nonlinear	Pawlowski	Maintenance (minor dev)
LOCA	Bifurcation/Stability Analysis	Phipps	Maintenance (minor dev)
ROL	Optimization	Ridzal	Active development
Tempus	Time Integration	Ober	Active development
Rythmos	Time Integration	Ober	Deprecated (use Tempus)
Sacado	Automatic Differentiation	Phipps	Active development
Stokhos	Ensemble Propagation	Phipps	Active development
PIRO	Black-box (ME) Abstractions	Tezaur, Perego	Maintenance (minor dev)
Thyra	Interface Abstractions	Bartlett	Active development

Sacado: AD Tools for C++ Applications

Automatic differentiation (AD) package
- Compute analytic derivatives in simulation codes without hand-coding
- Relies on chain-rule and known derivatives of all elementary operations

Operator overloading-based approach
- C++ data types storing values and derivatives
- Type of variables in code replaced by AD data type
- Mathematical operations replaced by overloaded versions implementing chain-rule
- Expression templates reduce overhead

New for 2019: new data type/expression template design in Sacado::Fad::Exp namespace
- Leverage modern C++11 features to simplify expression template design
- Enable efficient optimization of expression-template evaluations, particularly for nested data types (e.g., higher derivatives, Sacado+SIMD types)
- Will become the default in the coming months
- Can try yourself using Exp namespace or configuring with -DSacado_NEW_FAD_DESIGN_IS_DEFAULT:BOOL=ON

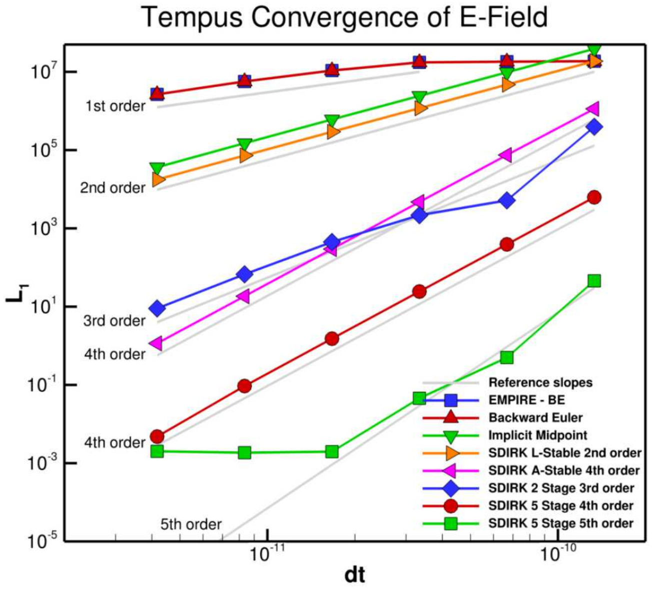


Iso-velocity adjoint surface for fluid flow in a 3D steady MHD generator in Drekar computed via Sacado (Courtesy of T. Wildey)

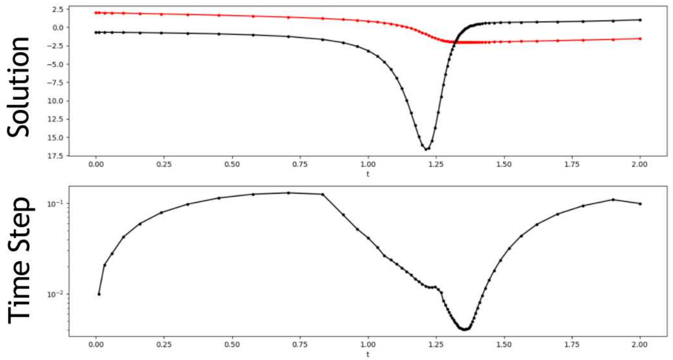
TRILINOS

http://trilinos.org

Tempus Capabilities



Van der Pol Problem



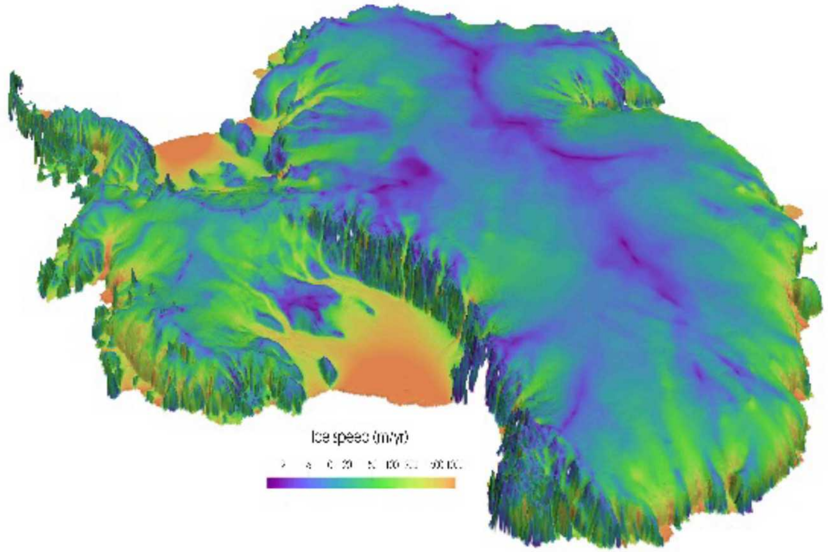
- Many Steppers
  - Implicit and Explicit, e.g., Forward/Backward Euler, BDF2 and Trapezoidal.
  - Runge-Kutta methods, e.g., Explict RK, DIRK and IMEX-RK.
  - 2^{nd} Order ODEs, e.g., Leapfrog, Newmark-β and HHT-α.
  - Stepper-of-Steppers, e.g., 1^{st} Order Operator Splitting and Subcycling.
- Runge-Kutta Embedded Error Analysis
  - Variable time-step error controller, e.g., I, PI and PID.
  - Bogacki-Shampine 3(2) Pair, Merson 4(5) Pair and SDIRK 2(1) Pair.
- Sensitivity Analysis
  - Transient forward/adjoint sensitivities.
  - Enables efficient large-scale optimization and UQ.
- Utilization by several applications
  - Albany/LCM – Enabled Schwarz Alternating Method for Dynamic Multiscale Coupling
  - SPARC – Improved RK method and embedded estimator to run near the stability limit (max time step).
  - Drekar – Developed and using Embedded Error Analysis
  - EMPIRE – Incorporated into suite and have demonstrated verification.

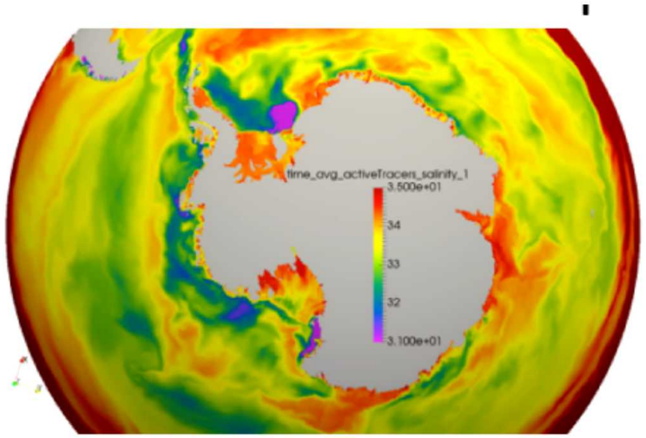
Tempus New Features

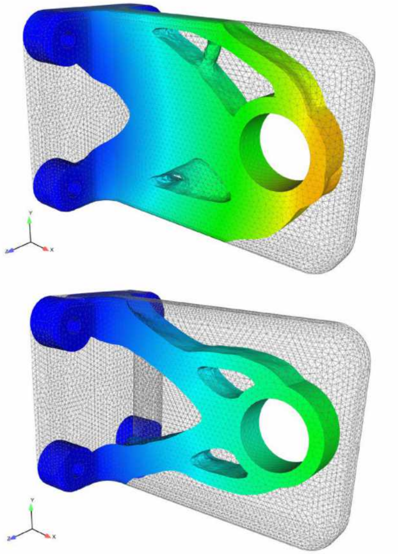
- Subcycling Stepper (i.e., Stepper-of-Steppers)
  - Ability to subcycle one operator in a operator-split scheme, e.g., explicit fast physics.
  - Has all the basic capabilities of IntegratorBasic, e.g., constant/variable time stepping and error control.
  - Can use any Stepper as the subcycling stepper, e.g., implicit/explicit, RK methods, and even Stepper-of-Steppers.
- "New and Improved" Stepper Constructors
  - Basic default constructors provide default settings, which can be reset, and only requires a ModelEvaluator.
  - Full specification constructors provide applications ability to set all the steppers parameters.
  - Still have ParameterList construction through the StepperFactory.
- Added methods to make the initial conditions (ICs) consistent.
  - Ensure the solution, x, $\dot{x}$ and $\ddot{x}$, satisfies the governing equation at ICs, e.g., $f(x, \dot{x}, \ddot{x}, t = 0) = 0$.
  - Ability to do "nothing", "zero" ICs, use application's ICs or solve for "consistent" ICs.
  - Important for DAEs to obtain the correct solution.
- Added First-Step-As-Last (FSAL) principle.
  - Some steppers can use the last evaluation from previous time step as the first evaluation this time step.
  - Saves the cost of the first evaluation.
  - Have the ability to turn off, e.g., FSAL does not generally work with operator-splitting.
- Many minor improvements
  - Solution output after "passing output time" and not change timestep to hit output time.
  - Many accessors to member data.
  - Miscellaneous bug and warning fixes.

PIRO

- Unifies top level packages (NOX, Tempus, Rythmos, LOCA, ROL) into a single interface (Model Evaluator)
- Current customers: Albany, Charon, Drekar and Panzer
- FY19 Development:
  - Added observer to monitor changes to optimization parameters (on branch)
  - Support for sim-opt interface to ROL (on branch)
- FY20 Plans:
  - Adjoint support for PIRO/Tempus wrapper for land ice project







ROL

- **Dynamic optimization interface**
  1. Problem definition at time-step level through DynamicObjective and DynamicConstraint.
  2. Improved adjoint interface for Tempus (discussed at 2018 TUG).
  3. New parallel-in-time optimization capabilities (DITTO-X LDRD).
  4. Randomized sketching to reduce memory footprint of state storage.

- **Optimal experimental design (OED)**
  1. New capabilities for various conventional optimality measures (A, I, D optimality).
  2. New methods for risk-based design of experiments.

- **Optimization vectors as template parameters**
  1. ROL classes are now templated on vector types.
  2. Enabling fast optimization algorithms on advanced architectures, such as GPUs.

- **Python interface through PyROL**

Teuchos Stacked Timer

- Added two years ago as an alternative to TimeMonitor
  - Preferred by application developers (wanted explicit start/stop)
  - Disambiguates multiple uses of same object
  - Not exception safe (will not stop a timer when unrolling the stack)
  - In TeuchosComm subpackage

- Developer recommendation:
  - Don't use in trilinos libraries, continue to use TimeMonitor
  - Project consistency and exception safety
  - TimeMonitor automatically dumps into StackedTimer.
    - Can disable with configure flag:

      Teuchos_ENABLE_STACKED_TIMER_IN_TIME_MONITOR=OFF

Teuchos Stacked Timer

```
Panzer MixedPoisson Test: 4.39393 [1]
|  Mixed Poisson: 4.39382 - 99.9976% [1]
|  |  panzer::CubeHexMeshFactory::buildUncomittedMesh(): 0.000528984 - 0.0120393% [1]
|  |  panzer::CubeHexMeshFactory::completeMeshConstruction(): 0.765863 - 17.4305% [1]
|  |  panzer::DOFManagerFactory::buildUnqueGlobalIndexer: 0.0451156 - 1.0268% [1]
…
|  |  Ifpack2::Relaxation::initialize: 4.403e-06 - 0.000100209% [1]
|  |  Ifpack2::Relaxation::compute: 0.00249703 - 0.0568305% [1]
|  |  Belos: Operation Op*x: 0.00338377 - 0.0770121% [1]
|  |  Belos: PseudoBlockGmresSolMgr total solve time: 0.432027 - 9.83259% [1]
|  |  |  Belos: ICGS[2]: Orthogonalization: 0.0212883 - 4.92755% [31]
|  |  |  |  Belos: ICGS[2]: Ortho (Norm): 0.000489003 - 2.29705% [31]
|  |  |  |  Belos: ICGS[2]: Ortho (Inner Product): 0.00904903 - 42.507% [60]
|  |  |  |  Belos: ICGS[2]: Ortho (Update): 0.00662776 - 31.1333% [60]
|  |  |  |  Remainder: 0.00512255 - 24.0627%
|  |  |  Belos: Operation Prec*x: 0.0420421 - 9.73136% [31]
|  |  |  |  Ifpack2::Relaxation::apply: 0.0418983 - 99.658% [31]
|  |  |  |  Remainder: 0.000143774 - 0.341977%
|  |  |  Belos: Operation Op*x: 0.104452 - 24.1772% [31]
|  |  |  Remainder: 0.264244 - 61.1639%
…
```

Some Useful Features

- Many output options: time, percentage, mpi min, max average, binned mpi distributions, remainders, print depth, output formats

Default:

```
TM:Interoperability: 5.18814 [1] <2, 0, 2>
|   Total Time: 5.18813 - 99.9996% [1] <2, 0, 2>
|   |   Assembly: 2.08128 - 40.1163% [10] <1, 1, 2>
|   |   |   Diffusion Term: 0.278306 - 13.3719% [10] <1, 2, 1>
|   |   |   Reaction Term: 0.779077 - 37.4326% [10] <2, 0, 2>
|   |   |   Remainder: 1.0239 - 49.1955%
|   |   Solve: 2.08407 - 40.1699% [10] <1, 1, 2>
|   |   |   Prec: 0.520275 - 24.9644% [10] <2, 0, 2>
|   |   |   GMRES: 0.539632 - 25.8932% [10] <1, 0, 3>
|   |   |   Remainder: 1.02416 - 49.1423%
|   |   Remainder: 1.02278 - 19.7138%
|   Remainder: 1.95e-05 - 0.000375857%
```

Aligned columns, labels on right:

```
5.18814                       [1]  <2, 0, 2> TM:Interoperability:
5.18813  - 99.9996%           [1]  <2, 0, 2> |   Total Time:
2.08128  - 40.1163%           [10] <1, 1, 2> |   |   Assembly:
0.278306 - 13.3719%           [10] <1, 2, 1> |   |   |   Diffusion Term:
0.779077 - 37.4326%           [10] <2, 0, 2> |   |   |   Reaction Term:
1.0239   - 49.1955%                          |   |   |   Remainder:
2.08407  - 40.1699%           [10] <1, 1, 2> |   |   Solve:
0.520275 - 24.9644%           [10] <2, 0, 2> |   |   |   Prec:
0.539632 - 25.8932%           [10] <1, 0, 3> |   |   |   GMRES:
1.02416  - 49.1423%                          |   |   |   Remainder:
1.02278  - 19.7138%                          |   |   Remainder:
1.95e-05 - 0.000375857%                      |   Remainder:
```

Aligned columns:

```
TM:Interoperability:        5.18814                 [1]  <2, 0, 2>
|   Total Time:             5.18813  - 99.9996%     [1]  <2, 0, 2>
|   |   Assembly:           2.08128  - 40.1163%     [10] <1, 1, 2>
|   |   |   Diffusion Term: 0.278306 - 13.3719%     [10] <1, 2, 1>
|   |   |   Reaction Term:  0.779077 - 37.4326%     [10] <2, 0, 2>
|   |   |   Remainder:      1.0239   - 49.1955%
|   |   Solve:              2.08407  - 40.1699%     [10] <1, 1, 2>
|   |   |   Prec:           0.520275 - 24.9644%     [10] <2, 0, 2>
|   |   |   GMRES:          0.539632 - 25.8932%     [10] <1, 0, 3>
|   |   |   Remainder:      1.02416  - 49.1423%
|   |   Remainder:          1.02278  - 19.7138%
|   Remainder:              1.95e-05 - 0.000375857%
```

Alternative: Nate Roberts has a python parser for stacked timer output parsing.

Some Useful Features

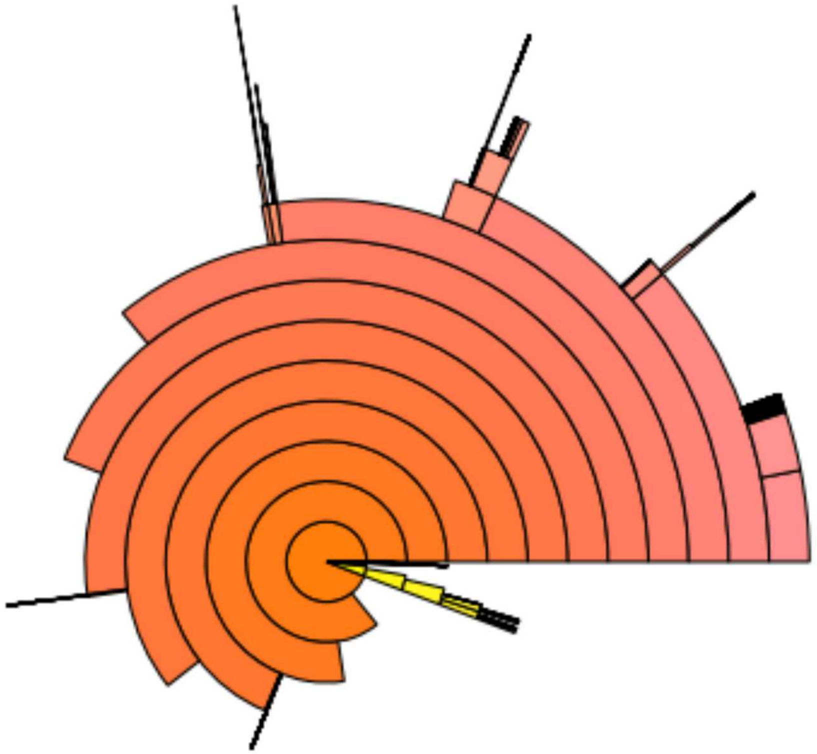
```
My New Timer: 2.11147 [1]
|   Total Time: 2.11147 [1]
|   |   Assembly: 1.03913 [10]
|   |   Solve: 1.0723 [10]
|   |   |   Prec: 0.53726 [10]
|   |   |   Rank 0 ONLY: 0.535003 [10]
|   |   |   Not Rank 0: 0.53467 [10]
|   |   |   Remainder: -0.534634
|   |   Remainder: 3.7e-05
|   Remainder: 1.25e-06

### Printing aligned_column with timers names on left ###
My New Timer:           2.11147                 [1]  (0) {min=2.11141 , max=2.11154 , std dev=6.07913e-05} <2, 0, 2>
|   Total Time:         2.11147   - 99.9999%    [1]  (0) {min=2.11141 , max=2.11154 , std dev=6.10055e-05} <2, 0, 2>
|   |   Assembly:       1.03913   - 49.2138%    [10] (0) {min=1.03894 , max=1.03925 , std dev=0.000141184} <1, 1, 2>
|   |   Solve:          1.0723    - 50.7845%    [10] (0) {min=1.07217 , max=1.07255 , std dev=0.00017294}  <3, 0, 1>
|   |   |   Prec:       0.53726   - 50.1036%    [10] (0) {min=0.537224, max=0.537291, std dev=2.87576e-05} <1, 1, 2>
|   |   |   Rank 0 ONLY: 0.535003 - 49.8931%    [10] (0)
|   |   |   Not Rank 0: 0.53467   - 49.8621%    [10] (0) {min=0       , max=0.534746, std dev=6.78994e-05} <1, 0, 3>
|   |   |   Remainder:  -0.534634 - -49.8588%
|   |   Remainder:      3.7e-05   - 0.00175233%
|   Remainder:          1.25e-06  - 5.92005e-05%
options.print_names_before_values = true == true = true : passed

### Printing aligned_column with timers names on right ###
2.11147                 [1]  (0) {min=2.11141 , max=2.11154 , std dev=6.07913e-05} <2, 0, 2> My New Timer:
2.11147   - 99.9999%    [1]  (0) {min=2.11141 , max=2.11154 , std dev=6.10055e-05} <2, 0, 2> |   Total Time:
1.03913   - 49.2138%    [10] (0) {min=1.03894 , max=1.03925 , std dev=0.000141184} <1, 1, 2> |   |   Assembly:
1.0723    - 50.7845%    [10] (0) {min=1.07217 , max=1.07255 , std dev=0.00017294}  <3, 0, 1> |   |   Solve:
0.53726   - 50.1036%    [10] (0) {min=0.537224, max=0.537291, std dev=2.87576e-05} <1, 1, 2> |   |   |   Prec:
0.535003  - 49.8931%    [10] (0)                                                             |   |   |   Rank 0 ONLY:
0.53467   - 49.8621%    [10] (0) {min=0       , max=0.534746, std dev=6.78994e-05} <1, 0, 3> |   |   |   Not Rank 0:
-0.534634 - -49.8588%                                                                        |   |   |   Remainder:
3.7e-05   - 0.00175233%                                                                      |   |   Remainder:
1.25e-06  - 5.92005e-05%                                                                     |   Remainder:
```

Some Useful Features

- Christian Glusa wrote a script that can create hierarchical pie plots with python
- Trilinos/packages/teuchos/comm/utils/plotStackedTimers.py



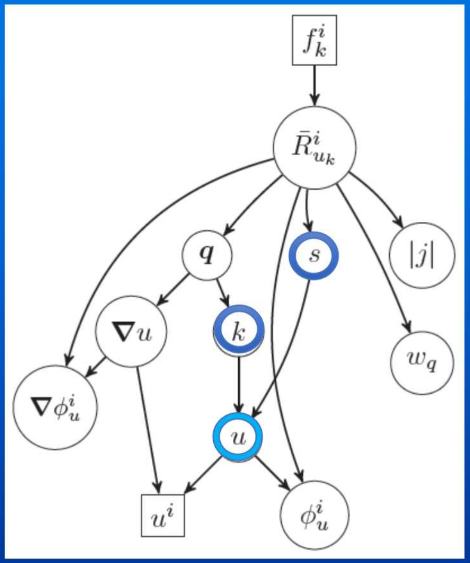
Some Useful Features

- Can inject all StackedTimer calls into Kokkos profiling tool via cmake flag:
  - Teuchos_KOKKOS_PROFILING=ON
- Can add kokkos kernel launches to StackedTimer with a new kokkos profiling tool
  - We can send you this file, it is not in the repo
- Can dump to std::cout to get a traceback into the code if env variable is set:
  - TEUCHOS_ENABLE_VERBOSE_TIMERS=1
  - Fast approximation (bracketing) for test failure triage
  - Good for large-scale HPC runs where interactive debugging tools aren’t effective
    - Can combine with splitting output for each mpi rank

```
Iter    21, [ 1] :    1.377530e-06
STARTING: Belos: Operation Op*x
STOPPING: Belos: Operation Op*x
STARTING: Belos: ICGS[2]: Orthogonalization
STARTING: Belos: ICGS[2]: Ortho (Inner Product)
STOPPING: Belos: ICGS[2]: Ortho (Inner Product)
STARTING: Belos: ICGS[2]: Ortho (Update)
STOPPING: Belos: ICGS[2]: Ortho (Update)
STARTING: Belos: ICGS[2]: Ortho (Inner Product)
STOPPING: Belos: ICGS[2]: Ortho (Inner Product)
STARTING: Belos: ICGS[2]: Ortho (Update)
STOPPING: Belos: ICGS[2]: Ortho (Update)
STARTING: Belos: ICGS[2]: Ortho (Norm)
STOPPING: Belos: ICGS[2]: Ortho (Norm)
STOPPING: Belos: ICGS[2]: Orthogonalization
Iter    22, [ 1] :    1.161976e-06
STARTING: Belos: Operation Op*x
STOPPING: Belos: Operation Op*x
STARTING: Belos: ICGS[2]: Orthogonalization
```

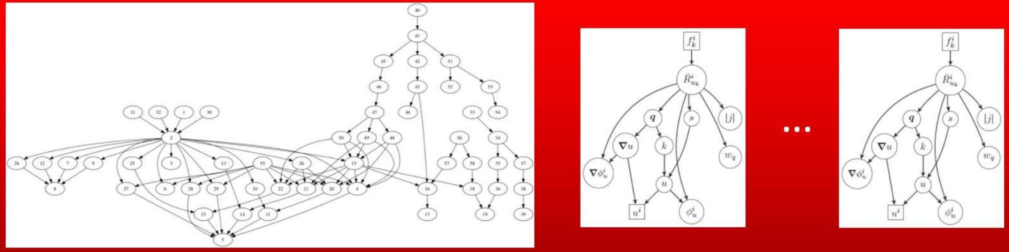
Phalanx

- FY19 Development:
  - Support arbitrary layout and multiple devices in an single Evaluation type
    - Works for MDField, Field and Kokkos::View
    - Layouts complete. Device is 90% complete.
    - Complete rewrite of MDField: leverage c++11 variadic templates, drops code base to ~25% of original impl.
  - Memory manager:
    - Memory management scheme: reuse view memory within DAG (and across DAGs).
    - Drekar plasma (56 PDEs) now 18% of original memory
    - MDField and Field no longer need unmanged allocators, they can be constructed similar to Kokkos::Views
    - On branch for now while debugging apps
- FY20 Plans:
  - Fix application issues
    - Unintentional assumptions in kernels
  - Finish memory management across DAGs



Reclamation

Assembly: Graphs w/ varying AD size

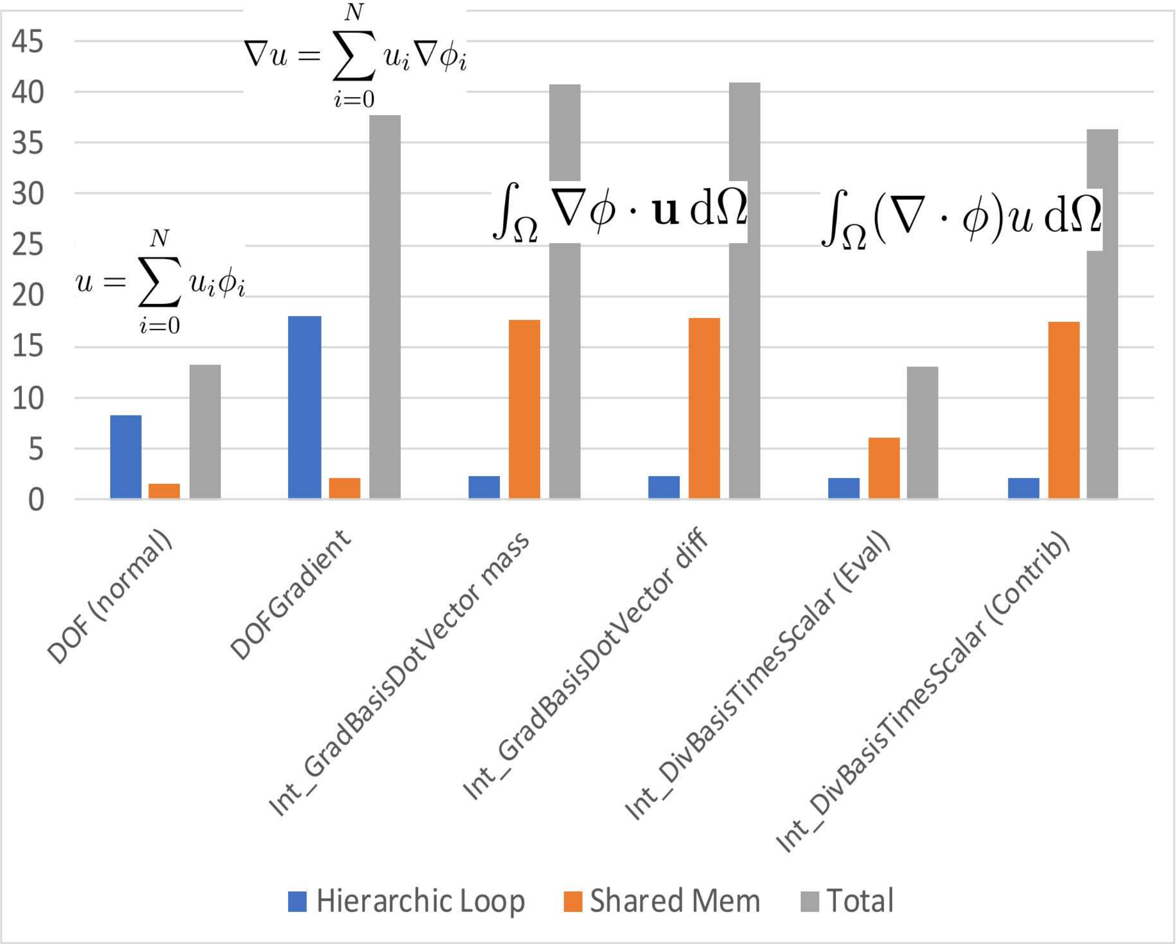


Volume

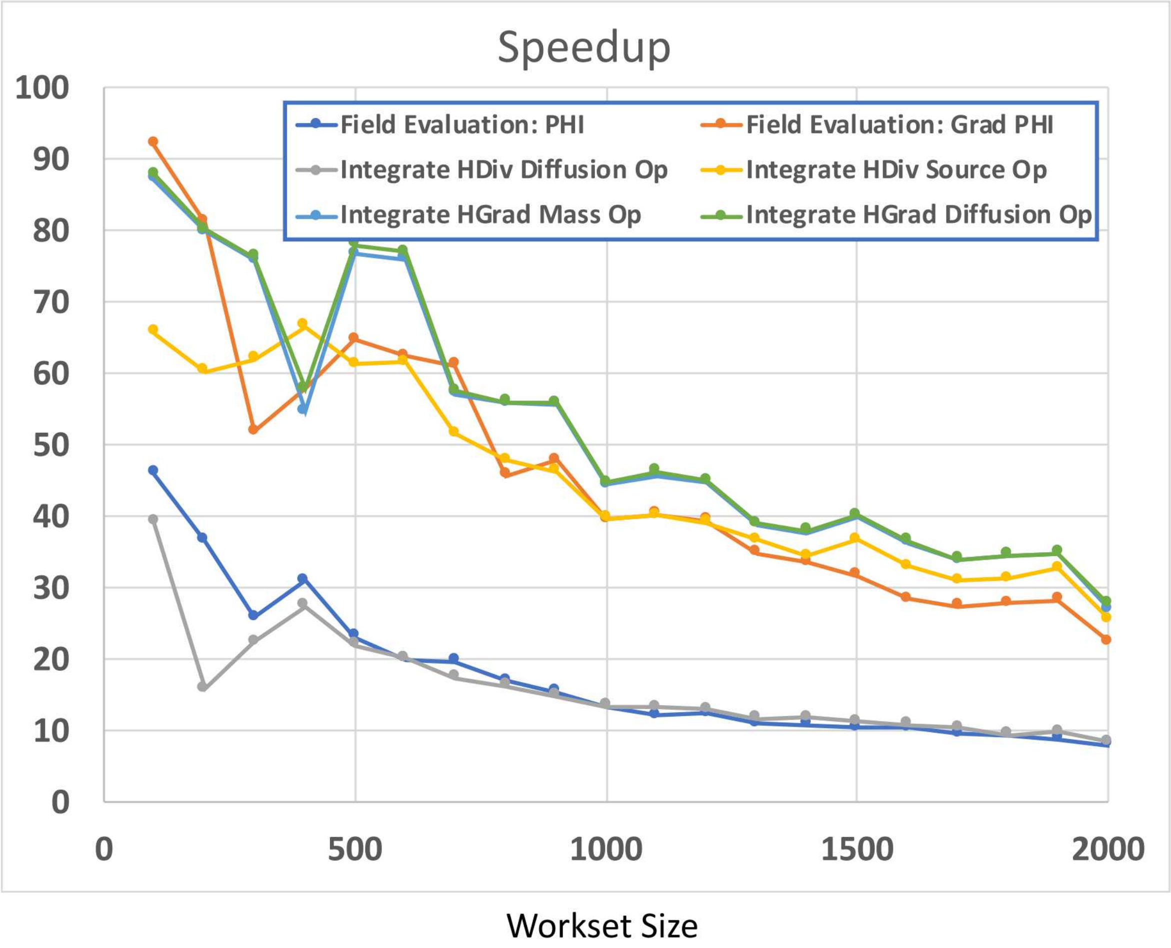
BC 0, …, BC N

Panzer: Hierarchic Parallelism Implemented in Evaluators

- Typical problem speedup is 10-40x, unit tests improved up to 80x
- Unified hierarchic parameter control for kernel launch: Safety mechanism for code reuse that differentiates AD vs non-AD evals
- Memory bound on roofline



Speedup of Panzer Kernels for Mixed HGrad/HDiv Unit Test



Panzer

- L2 Projections extended to (Thyra/Teko) blocked systems
- Ordinal definitions
  - Single GO based on Tpetra configure time choice
  - LO is hard coded to int
  - Elimination of all template parameters from DOF Manager and Connection Manager!
  - Can still build type 1 and type 2 stacks together (currently required)
- Worksets: unifying construction paths and simplifying the interface
- ASC SQE Audit (Sept 2019)
- FY20 Plans:
  - Periodic BC extensions
  - Lazy evaluation of worksets
  - **Remove all use of UVM**
  - New assembly path to eliminate post-assembly communication (requires full one-ring)
  - Disable (possibly drop) type-1 (epetra) stack support