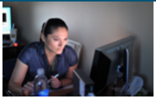
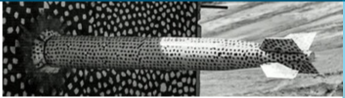
Sandia National Laboratories

The ALEGRA Finite Element Code and Trilinos





Presented by:

Tim Fuller

October 25, 2022

Overview

What is ALEGRA?

What is ALEGRA?

ALEGRA began development circa 1996 to solve the transient equations of Lagrangian solid dynamics (balance laws)

- conservation of mass

$$\frac{d}{dt}(\rho\, dv) = \frac{d}{dt}(dm) = 0$$

- conservation of momentum

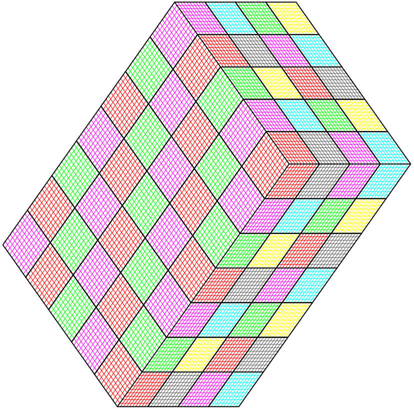
$$\rho \frac{d}{dt}\dot{U} = \rho \ddot{U} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \boldsymbol{f}$$

- conservation of energy

$$\rho \frac{d}{dt} e = \rho\, s = \boldsymbol{\sigma} : \boldsymbol{D} - \nabla \cdot \boldsymbol{q}$$

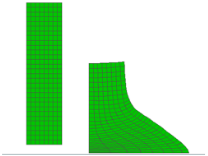
What is ALEGRA?

. . . using a finite element discretization . . .

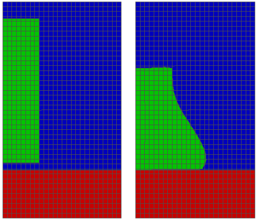


What is ALEGRA?

. . . with Lagrangian, Eulerian, Arbitrary Lagrangian-Eulerian (ALE), or XFEM capability . . .



Lagrangian

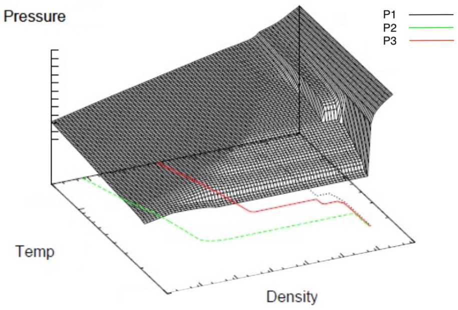


Eulerian

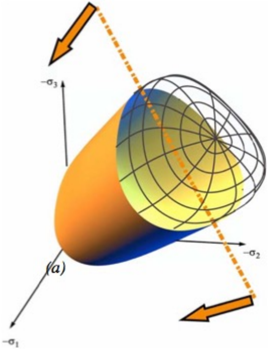
ALEGRA = **ALE G**eneral **R**esearch **A**pplication

What is ALEGRA?

. . . using material data to close the system . . .



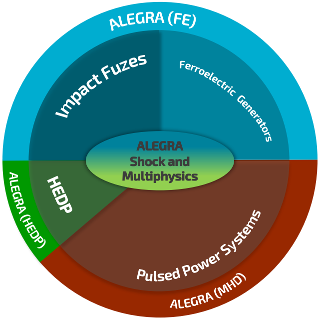
EOS model



Yield model

What is ALEGRA?

. . . and operator splitting to add multiphysics.



Who uses ALEGRA?

Analysts developing models for

- Solid mechanics
- Shock hydrodynamics
- Electromagnetics
- Radiation hydrodynamics

Physicists and engineers interested in

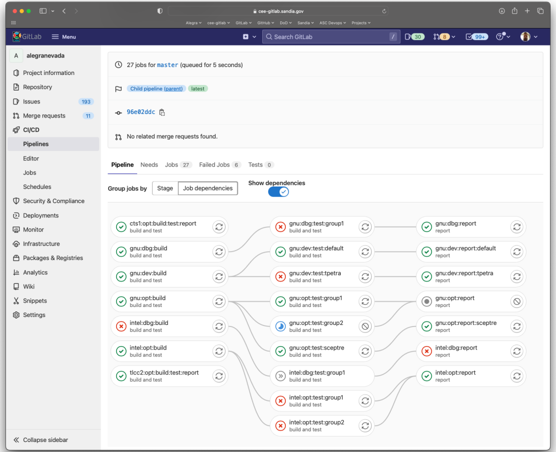
- Pulsed power
- Terminal ballistics
- Radiation effects

Civil servants and contractors in DOE and DoD physics/engineering R&D institutions

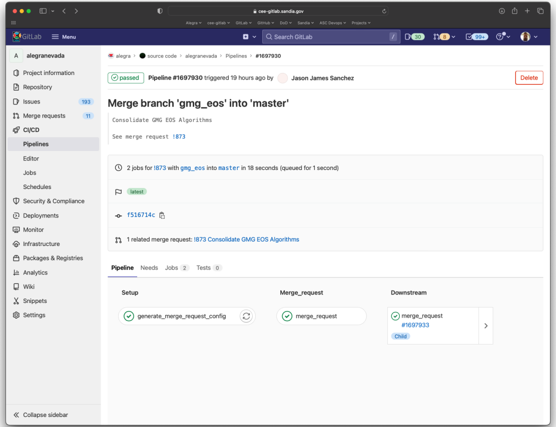
ALEGRA devops

- Git repository served on GitLab
- Spack for dependency management
- vvtest for testing
- CMake build system
- GitLab runners for CI/CD
- Toolset to glue it all together
- Toolset is python library used by developers and users to build, run, and test ALEGRA

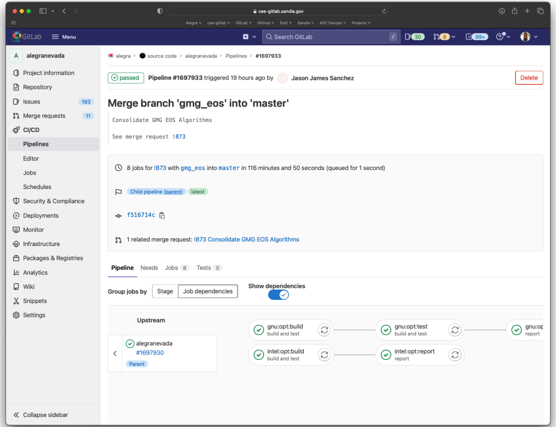
ALEGRA Devops



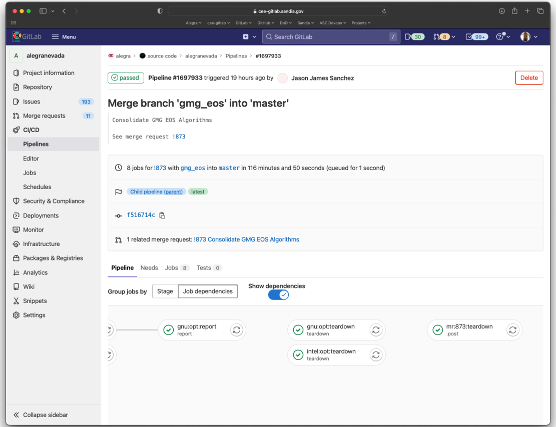
ALEGRA Devops



ALEGRA Devops

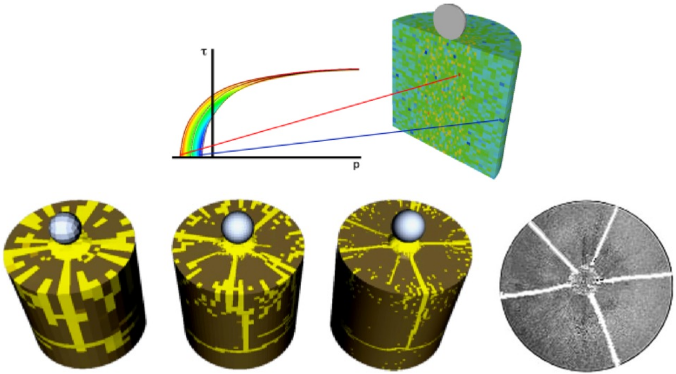


ALEGRA Devops



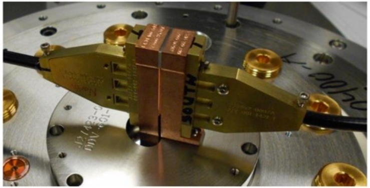
Applications

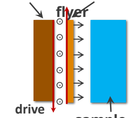
Validating theories of brittle damage

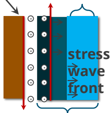


Leavy, Int. J. Appl. Ceram. Technol., 7 [5] 606-615 (2010)

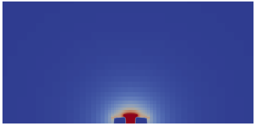
Dynamic material properties (DMP)

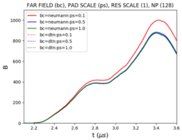






Compute Accurate Magnetic Field Drive History in the Gap at the Material Surface





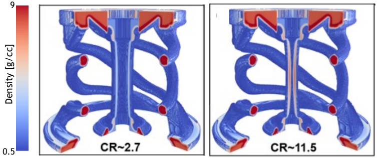
Courtesy Allen Robinson

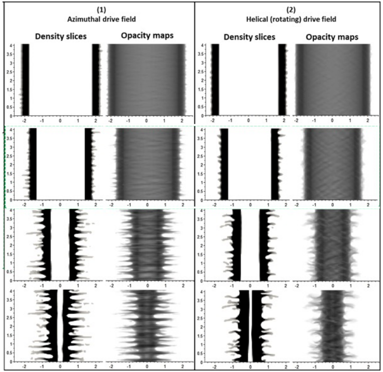
MRTI mitigation via screw pinch mechanism

Azimuthal drive field implosion
vs.
Helical (rotating) drive field produced by Z design return current structure

$$\Xi = \frac{B_z}{B_\phi} \sim 0.5$$

3D ALEGRA

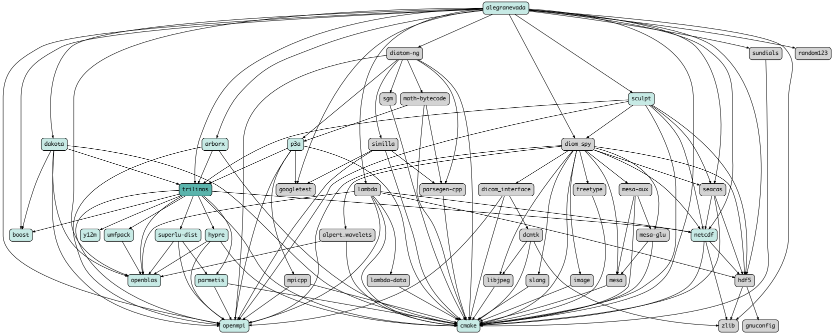




G. A. Shipley, C. A. Jennings, and P. F. Schmit, *Physics of Plasmas* **26**, 102702 (2019)

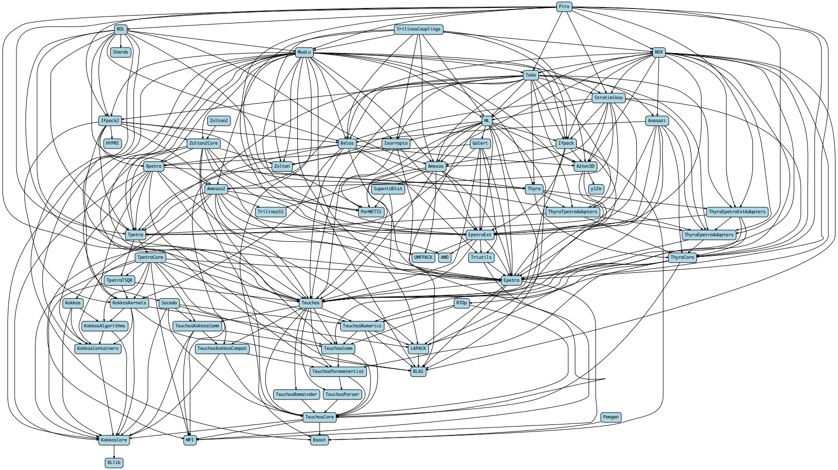
Trilinos usage in ALEGRA

ALEGRA depends on many third-party libraries



Trilinos plays a central role in ALEGRA by providing data structures and solvers required by key physics capabilities.

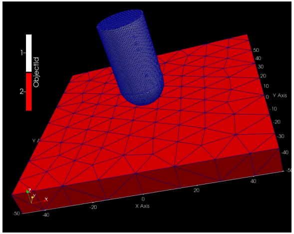
ALEGRA depends on many Trilinos packages



CAD insertion through SEACAS

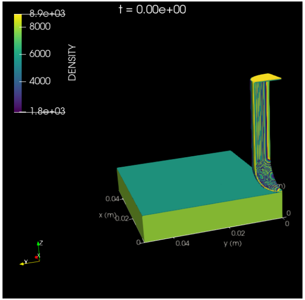
The SEACAS `nas2exo` tool converts Nastran bulk data files to Exodus files that can be imported into an ALEGRA simulation

Exo file produced by nas2exo




```
diatom

scale {mm2m}

  package 'Projectile'
    material 1
    mvelocity 0. 0. -4000.
    insert exodus
      file = 'boss2-mod.exo'
      block = 1
    endi
  endp

  package 'Plate'
    material 2
    insert exodus
      file = 'boss2-mod.exo'
      block = 2
    endi
  endp

endscale

enddiatom
```

ALEGRA initial condition



Moral, Niederhaus

Epetra to Tpetra migration in MHD physics

Requirements

- Transition Epetra/AztecOO/ML to Tpetra/Belos/MueLu
- Enable 64 bit global indices to enable 2^{32} (or more) unknowns
- Ensure same results using Tpetra stack as Epetra stack

Strategy

- Support both Epetra and Tpetra via runtime switch
- Work with Trilinos developers to develop missing functionality in Tpetra stack
- Develop IO capabilities to compare results between Epetra/Teptra stacks
- Nightly regression testing of both stacks

Lessons learned

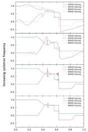
- Epetra does a lot of work under the hood that users of Tpetra must do themselves
- `Tpetra_FECrsGraph`, `Tpetra_FECrsMatrix`, and `Tpetra_FEMultiVector` had a non-intuitive interface (fixed)
- Belos and MueLu lack(ed) implementations for several solvers and preconditioners implemented in AztecOO/ML that users of ALEGRA depend on

3D GMHD(XMHD) ALEGRA

1) Original implementation of GMHD equations (Hall term) was found to contain a near null space but included non-physical oscillatory modes complicating scalable linear solves.

$$\int \Big(\underbrace{\frac{\epsilon}{\Delta t} + \underline{\chi}_{\beta}\sigma^{n+1}\Pi_P}_{\text{when Hall term dominates, } \approx \frac{1}{3} \text{ of spectrum is nearly 0}}\Big)\mathbf{E}^{n+1} \cdot \boldsymbol{\Psi}\, d\Omega + \int \frac{\Delta t}{\mu} \operatorname{curl} \mathbf{E}^{n+1} \cdot \operatorname{curl} \boldsymbol{\Psi}\, d\Omega$$

solve for $\mathbf{E}^{n+1}$



2) Reformulated into 2x2 system which avoids element projections in Ohm's law to produce a matrix more amenable to solver technology.

Block edge element matrix

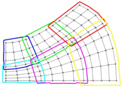
$$\begin{bmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ \mathbb{A}_{21} & \mathbb{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{E}^{n+1} \\ \mathbf{J}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix},$$

$$\mathbb{A}_{11}(\widehat{\mathbf{E}}_i, \widehat{\mathbf{E}}_j) = \frac{\epsilon}{\Delta t}\int_\Omega \widehat{\mathbf{E}}_i \cdot \widehat{\mathbf{E}}_j\, d\Omega + \frac{\Delta t}{\mu}\int_\Omega \operatorname{curl}\widehat{\mathbf{E}}_i \cdot \operatorname{curl}\widehat{\mathbf{E}}_j\, d\Omega \qquad \mathbb{A}_{12}(\widehat{\mathbf{E}}_i, \widehat{\mathbf{J}}_j) = \int_\Omega \widehat{\mathbf{E}}_i \cdot \widehat{\mathbf{J}}_j\, d\Omega$$

$$\mathbb{A}_{21}(\widehat{\mathbf{J}}_i, \widehat{\mathbf{E}}_j) = -\int_\Omega \sigma^{n+1}\left(\widehat{\mathbf{J}}_i \cdot \widehat{\mathbf{E}}_j\right) d\Omega \qquad \mathbb{A}_{22}(\widehat{\mathbf{J}}_i, \widehat{\mathbf{J}}_j) = \int_\Omega \left(\left(1 + \frac{\tau}{\Delta t}\right)\left(\widehat{\mathbf{J}}_i \cdot \widehat{\mathbf{J}}_j\right) - \frac{e\tau}{m_e}\left[\widehat{\mathbf{J}}_i \cdot \left(\mathbf{B}^n \times \widehat{\mathbf{J}}_j\right)\right]\right) d\Omega$$

3) Designed **patch** preconditioners to resolve near null space errors and complement existing algebraic multigrid preconditioners.

Tuminaro Led Late Start LDRD – "Composing preconditioners for multiphysics PDE systems with applications to Generalized MHD"
SAND2022-12164R



Crockatt, Tuminaro, Robinson

3D GMHD(XMHD) ALEGRA

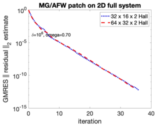
ALEGRA GMHD Solver Strategy going Forward

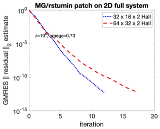
1) Devise and implement a production 3D algebraic multigrid software strategy based on MatLab based geometric multigrid analysis of sample Alegra matrices.

Two types of patch preconditioners have been investigated:

Arnold-Faulk-Winter (MG/AFW) and Adjacent Elements (MG/rstumin)

Scalable





Faster

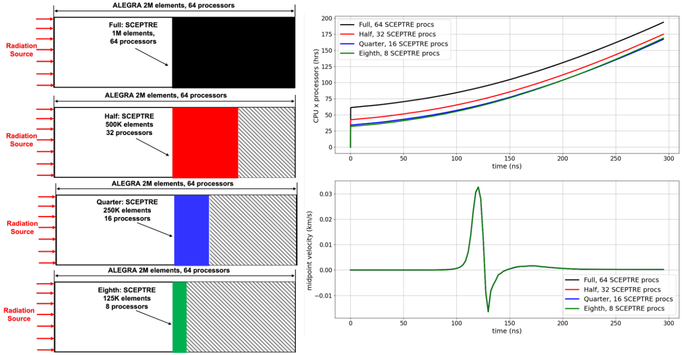
2) Trilinos impact: work should result in a better patch preconditioner infrastructure useful for dealing with multiple near null space preconditioning issues.

3) Will also investigate additional XMHD model extensions and the associated impact on discretization and solver strategy.

Crockatt, Tuminaro, Robinson

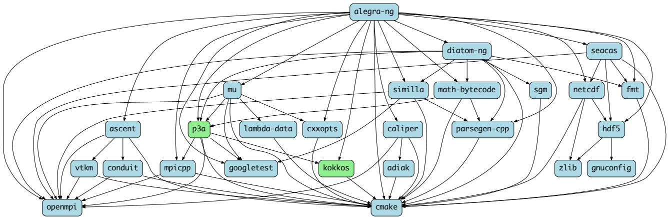
ALEGRA-SCEPTRE coupling through STK

- ALEGRA-SCEPTRE interface allows radiation transport physics subdomains, for which SCEPTRE is only applied to relevant parts of a problem
- Coupling achieved via `STKSearch`, `STKTransfer`, and MPI MPMD interfaces



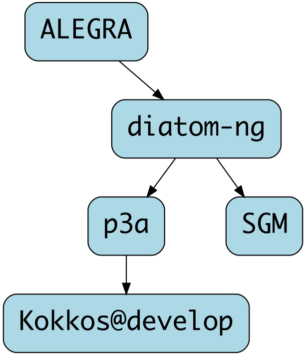
ALEGRA performance portability through Kokkos

The ALEGRA project has a performance portability effort to rewrite ALEGRA functionality using Kokkos and modern C++17

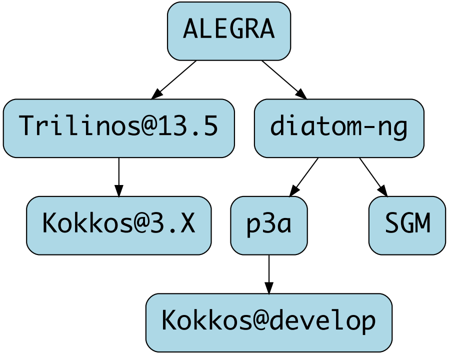


- Achieved across Intel CPUs, NVIDIA V100 GPUs, and NVIDIA A100 GPUs
- Developed SIMD types for guaranteed efficient outer loop vectorization on Intel and ARM CPUs
- SIMD types are being moved into Kokkos to benefit other Sandia applications
- Kokkos built standalone and for GPUs uses the built-in CMake support for CUDA, not the Kokkos compiler wrappers

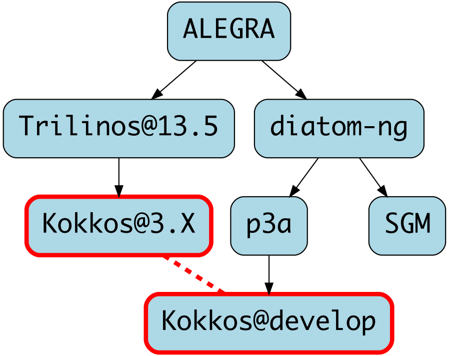
Next-gen geometry insertion



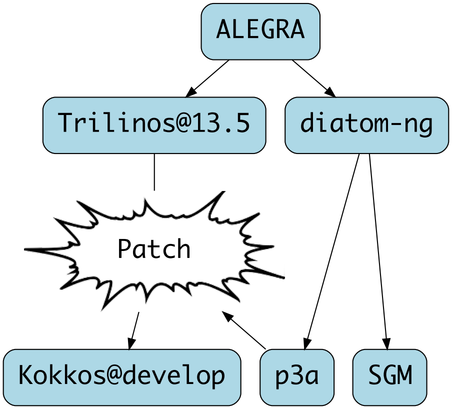
Next-gen geometry insertion



Next-gen geometry insertion



Next-gen geometry insertion



Conclusion

Conclusion

- ALEGRA depends on dozens of Trilinos packages (and sub-packages)
- Trilinos is a key component to ALEGRA delivering value to its customers
- ALEGRA uses both the Epetra and Tpetra stack and is currently migrating to fully Tpetra
- Infrequent Trilinos releases causes issues in a Spack eco system
- Inability to use intrinsic CMake CUDA support is an obstacle for ALEGRA's GPU strategy
- Inability to link external Kokkos causes significant devops issues