Contact constraint enforcement using the Tribol interface physics library

MFEM Community Workshop 2023

October 26, 2023

Eric B. Chin (LLNL)

LLNL-PRES-856390

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Tribol

Contact interface physics library

https://github.com/LLNL/Tribol

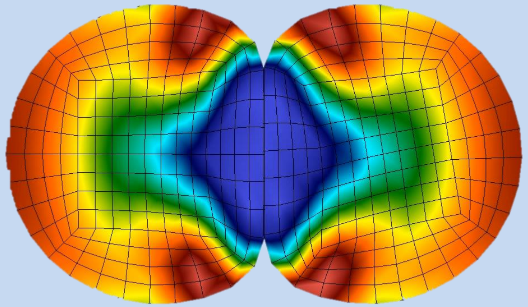
- Current Tribol team:
  - SR Wopschall
  - EB Chin
  - K Weiss
- Code is written in C++
  - Dependencies on **mfem** and **axom**
- Features C and Fortran API compatibility
- Tribol is integrated into many codebases:
  - ale3d
  - blast
  - diablo
  - serac

Tribol

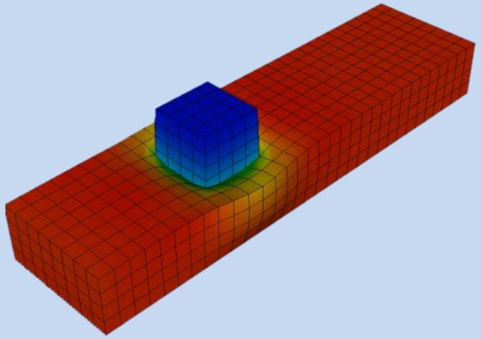
Contact interface physics library

https://github.com/LLNL/Tribol

Contact methodologies

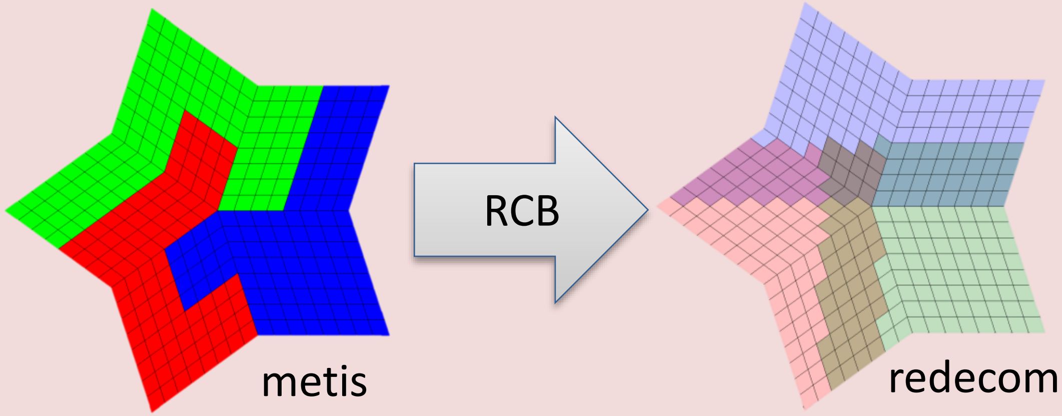


Common plane



Mortar

Domain redecomposer (redecomp)



MFEM interface

Submesh

- Create a surface ParMesh from boundary attributes
- Elements/dofs do not change rank

Low-order refined transfers

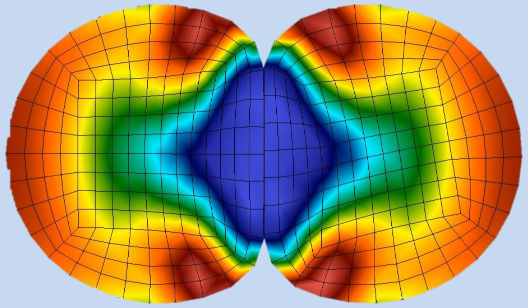
- Transfer higher-order mesh data to a refined low-order mesh
- Mass-conservative mapping

Tribol

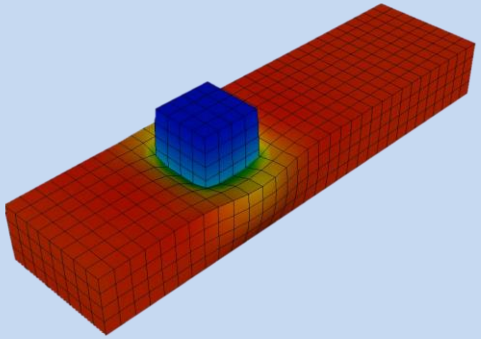
Contact interface physics library

https://github.com/LLNL/Tribol

Contact methodologies

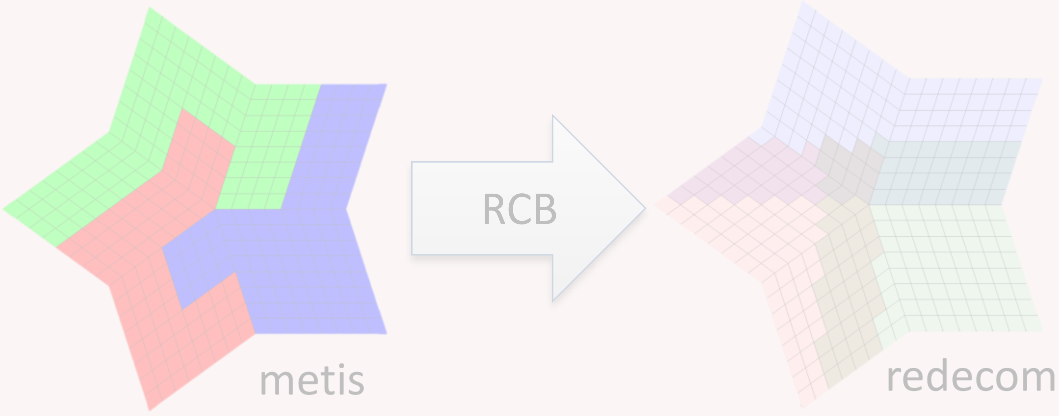


Common plane



Mortar

Domain redecomposer (redecomp)



MFEM interface

Submesh

- Create a surface ParMesh from boundary attributes
- Elements/dofs do not change rank

Low-order refined transfers

- Transfer higher-order mesh data to a refined low-order mesh
- Mass-conservative mapping

Tribol contact methodologies

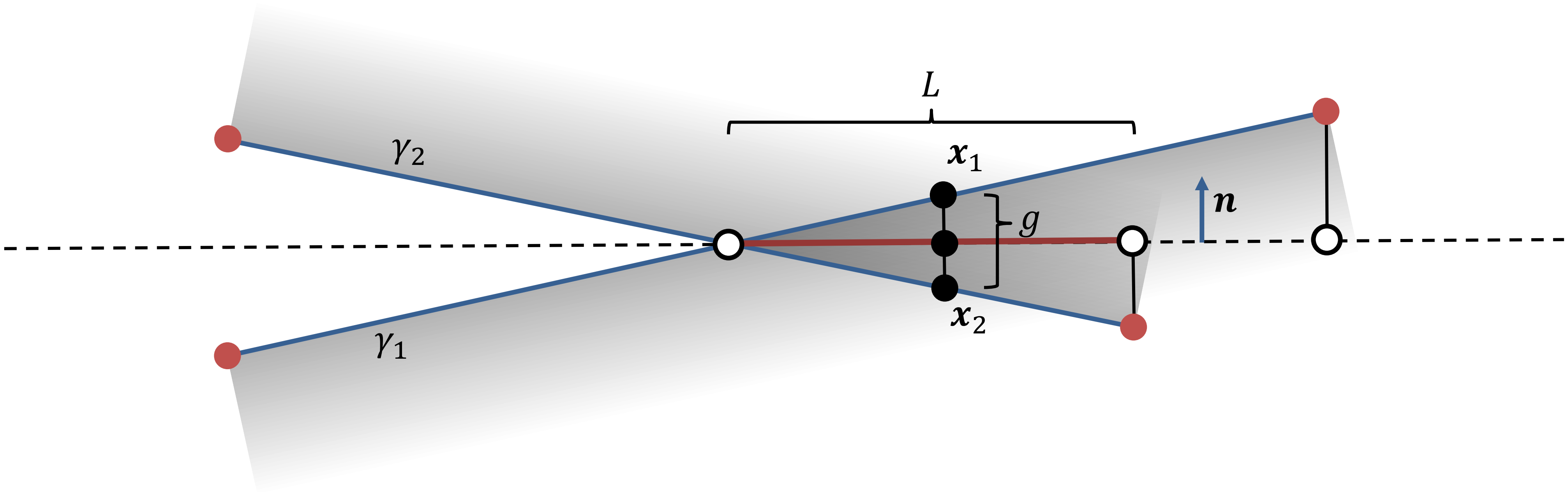
- Contact energy: $\Pi_c = \int_{\Omega_c} p\ g\ dS$

- Pressure (tied contact: enforces $g = 0$ or contact: enforces $g \geq 0$):
  - Penalty
  - Lagrange multiplier
  - Augmented Lagrange

- Gap:
  - How to define?
    - In terms of one surface?
    - In terms of an average over both surfaces?
  - Identifying contact pairs
  - Must also define a normal direction

- Surface of integration:
  - Identifying surfaces in contact (computational geometry)
  - Domain
    - One surface of the pair?
    - Average of both surfaces?
    - Both surfaces?
  - Integration rule
    - Segmentation?
    - Element-based?
    - Nodal? (i.e. node-to-surface contact)

Tribol contact methodologies: common plane



Contact force on pair: $f = \kappa g L$

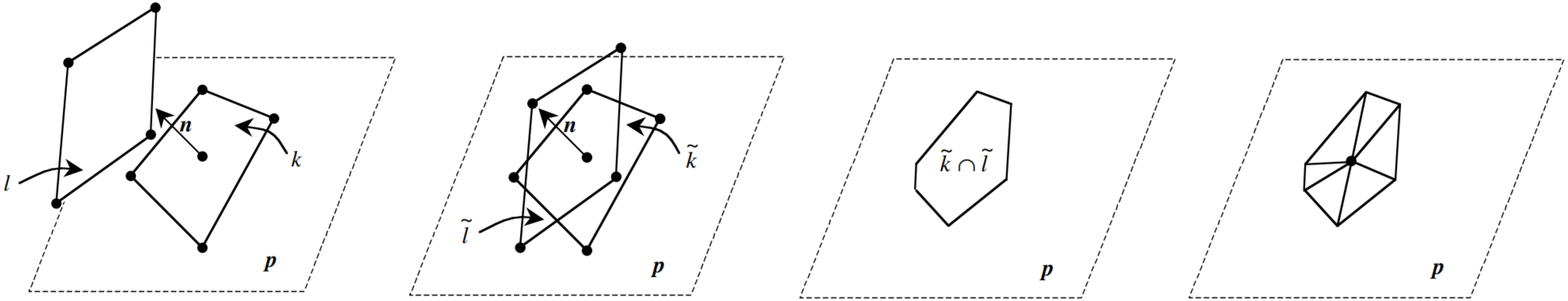
- κ: penalty parameter (force/area)
- $g = (\boldsymbol{x}_2 - \boldsymbol{x}_1) \cdot \boldsymbol{n}$: gap at overlap centroid
- $\boldsymbol{n}$: unit normal of common plane (average of each normal in pair)
- L: length of pair overlap

Tribol contact methodologies: mortar

Puso and Laursen, Comput Methods Appl Mech Engrg (2004)

Contact force on node B on side i from p_A: $\quad f_{AB}^{i} = p_A \boldsymbol{n}_A \int_{\gamma} N_A^1 N_B^i \, ds$

- p_A: pressure value on node A (non-mortar side; side 1)
- $\boldsymbol{n}_A$: nodally averaged unit normal vector on non-mortar side
- $\int_{\gamma} N_A^1 N_B^i \, ds$: mortar integral
  - If $i = 2$, integrand usually isn't smooth over domain of integration
  - Use segmentation to define integration domain and rule:

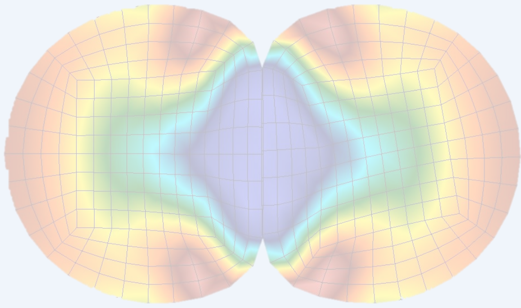


Tribol

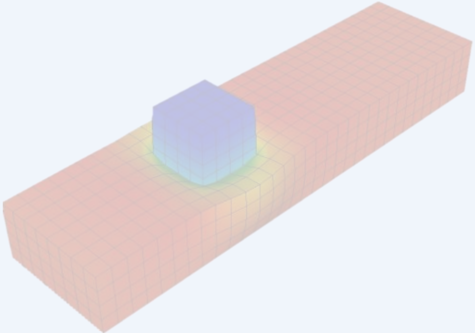
Contact interface physics library

https://github.com/LLNL/Tribol

Contact methodologies

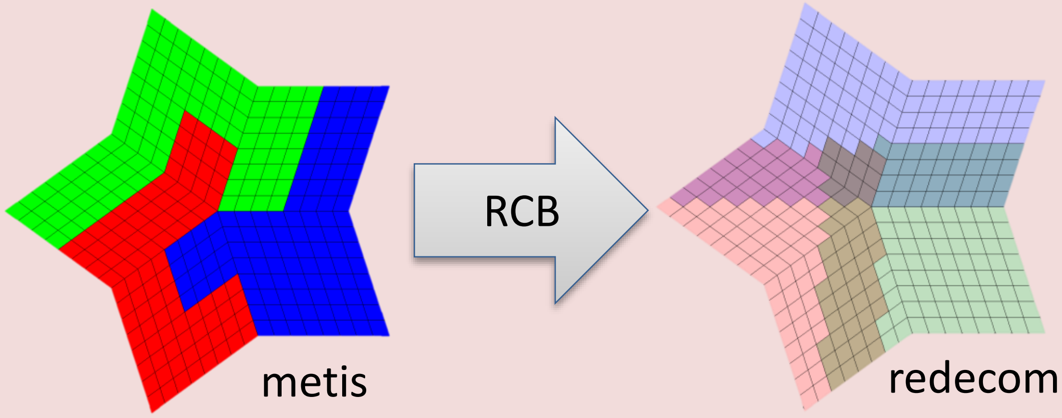


Common plane



Mortar

Domain redecomposer (redecomp)



MFEM interface

Submesh

- Create a surface ParMesh from boundary attributes
- Elements/dofs do not change rank

Low-order refined transfers

- Transfer higher-order mesh data to a refined low-order mesh
- Mass-conservative mapping

Domain redecomposer

Tribol’s contact algorithms

SERIAL

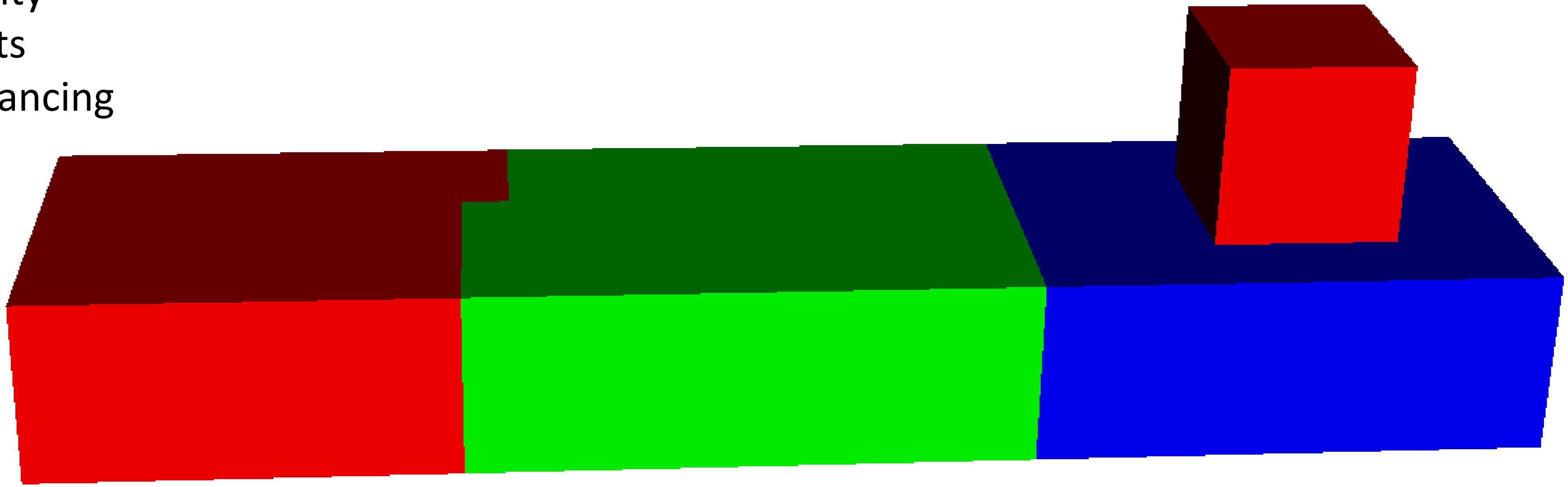
Domain redecomposer
(redecomp)

Typical MFEM problem of interest

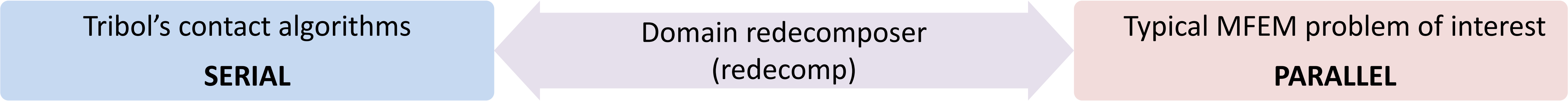
PARALLEL

- Why redecompose?
  - Increase code modularity
  - Spatial proximity
  - Ghost elements
  - Problem rebalancing

Contact ironing problem
Metis decomposed domain



Domain redecomposer



Tribol’s contact algorithms

SERIAL

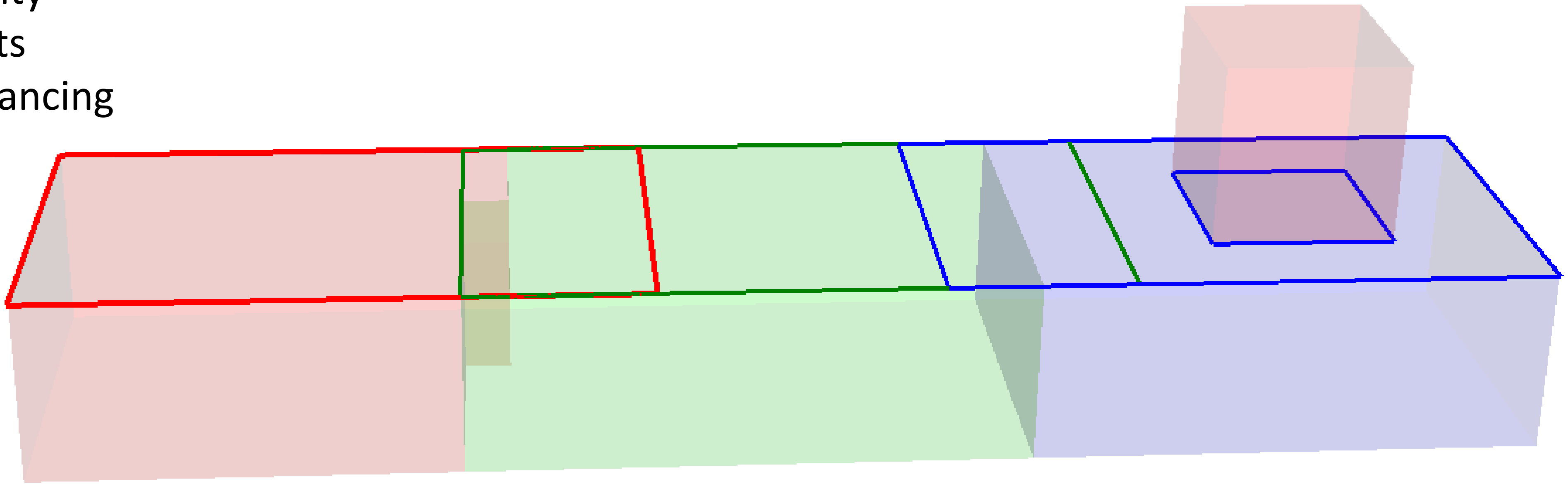
Domain redecomposer
(redecomp)

Typical MFEM problem of interest

PARALLEL

- Why redecompose?
  - Increase code modularity
  - Spatial proximity
  - Ghost elements
  - Problem rebalancing

Contact ironing problem
Redecomposed domain (outline)



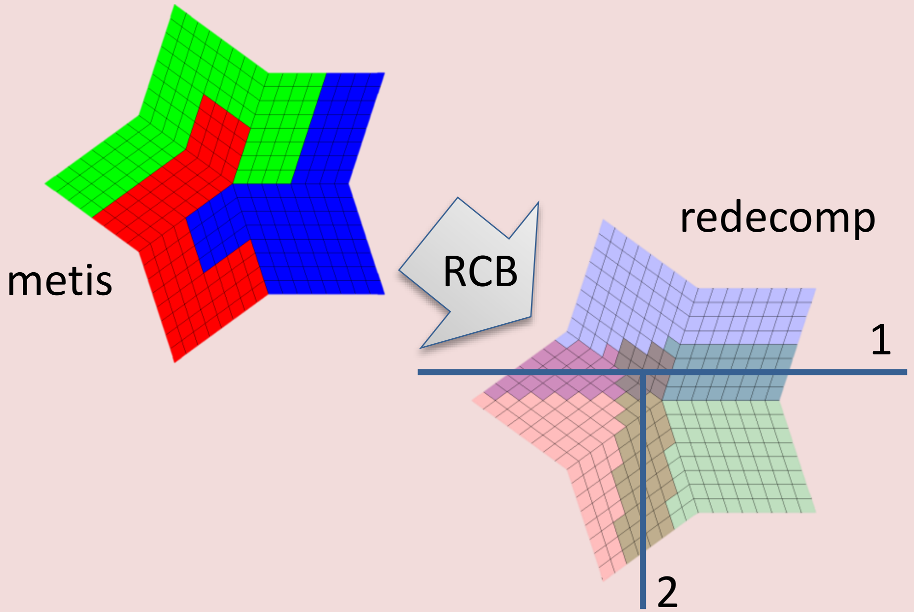
Using the domain redecomposer

See the src/redecomp directory in the Tribol repo

1. Create a RedecompMesh object

   Requires

   - An MFEM ParMesh
   - (Optional) RCB options

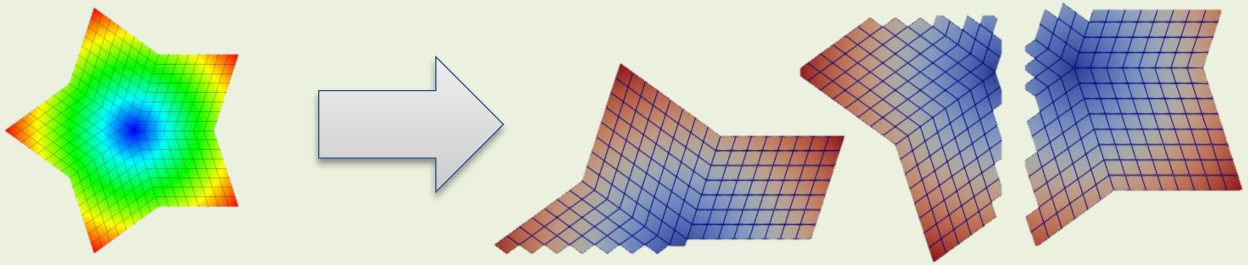


Moving grid functions and quadrature functions

2. Create a RedecompTransfer object

   Optional arguments (optimizes grid function transfer)

   - ParFiniteElementSpace on ParMesh
   - FiniteElementSpace on RedecompMesh



Moving matrices

2. Create a MatrixTransfer object

   Arguments

   - Test and trial ParFiniteElementSpace on ParMesh
   - Test and trial FiniteElementSpace on RedecompMesh

Using the domain redecomposer

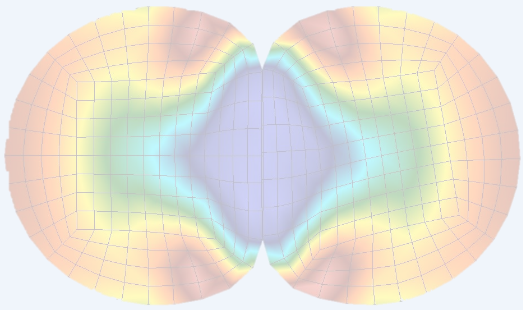
- Example problems
  - src/examples/domain_redecomp.cpp
    - Create a redecomp mesh, transfer grid functions, and transfer quadrature functions to/from a ParMesh
  - src/examples/multidomain_redecomp.cpp
    - Same as the previous example, but with one RCB decomposition over multiple ParMeshes
  - src/examples/matrix_redecomp.cpp
    - Compute mass matrix on a redecomp mesh then move and assemble on a linked ParMesh

Tribol

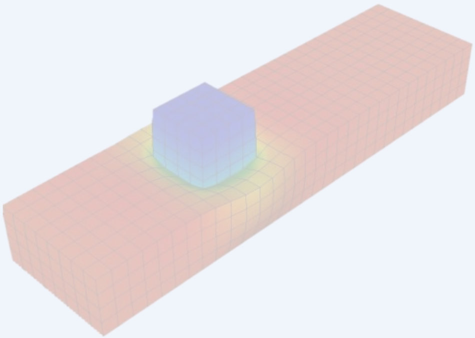
Contact interface physics library

https://github.com/LLNL/Tribol

Contact methodologies

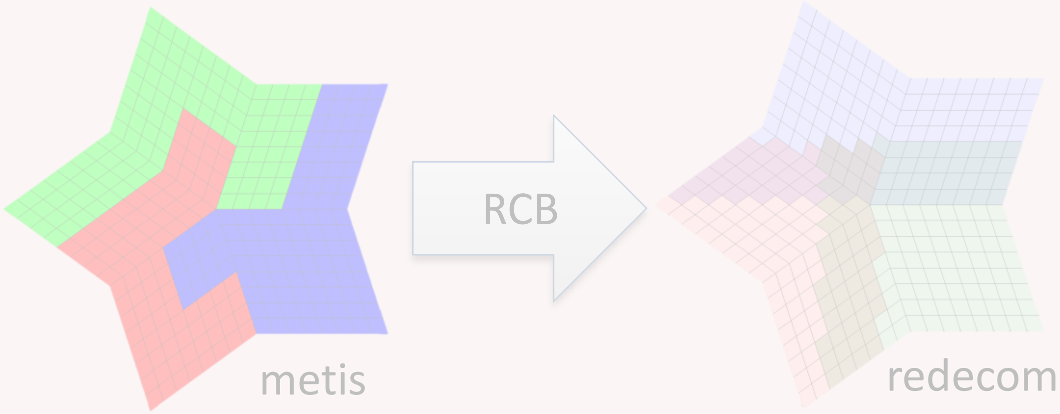


Common plane



Mortar

Domain redecomposer (redecomp)



MFEM interface

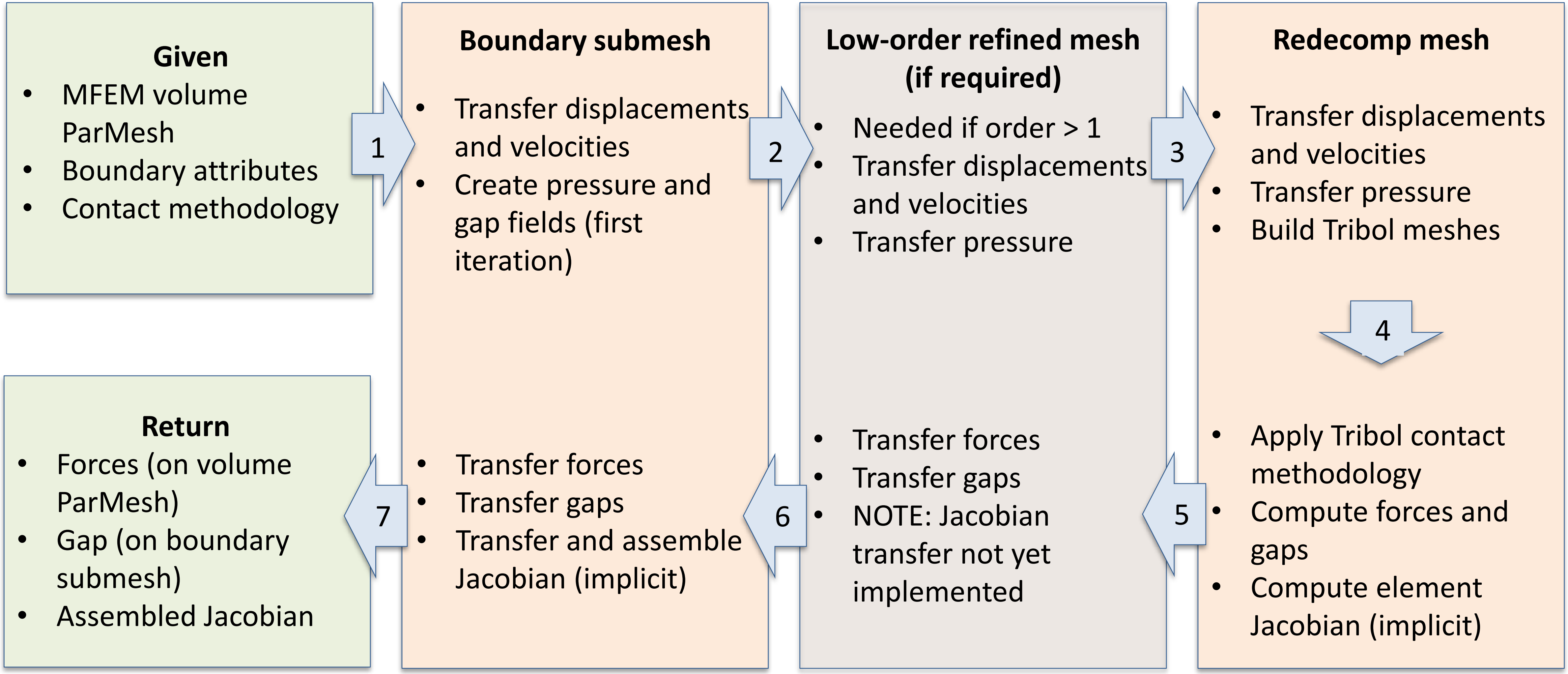
Submesh

- Create a surface ParMesh from boundary attributes
- Elements/dofs do not change rank

Low-order refined transfers

- Transfer higher-order mesh data to a refined low-order mesh
- Mass-conservative mapping

Tribol MFEM interface



Given

- MFEM volume ParMesh
- Boundary attributes
- Contact methodology

1

Boundary submesh

- Transfer displacements and velocities
- Create pressure and gap fields (first iteration)

2

Low-order refined mesh (if required)

- Needed if order > 1
- Transfer displacements and velocities
- Transfer pressure

3

Redecomp mesh

- Transfer displacements and velocities
- Transfer pressure
- Build Tribol meshes

4

- Apply Tribol contact methodology
- Compute forces and gaps
- Compute element Jacobian (implicit)

5

- Transfer forces
- Transfer gaps
- NOTE: Jacobian transfer not yet implemented

6

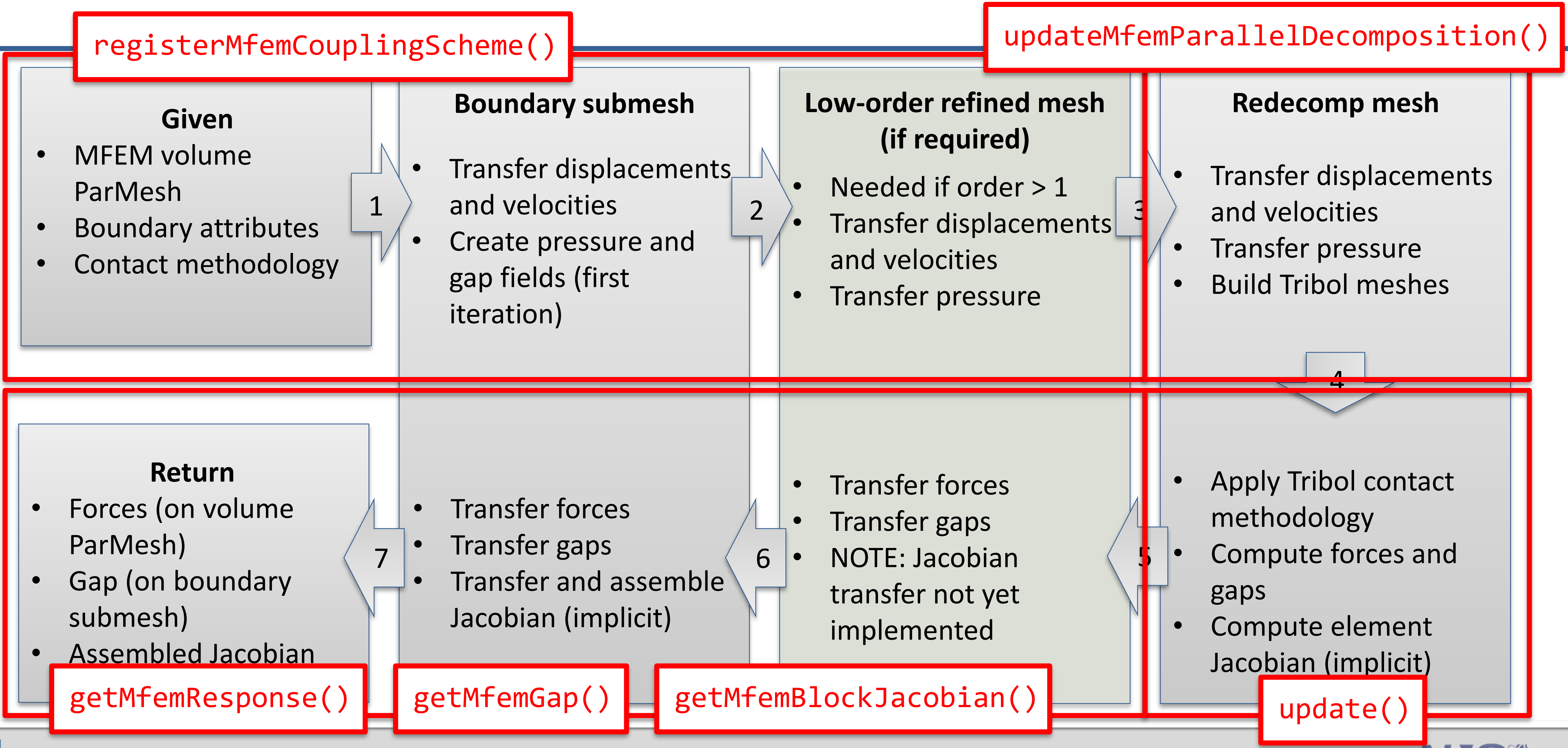
- Transfer forces
- Transfer gaps
- Transfer and assemble Jacobian (implicit)

7

Return

- Forces (on volume ParMesh)
- Gap (on boundary submesh)
- Assembled Jacobian

Tribol MFEM interface



`registerMfemCouplingScheme()`

Given

- MFEM volume ParMesh
- Boundary attributes
- Contact methodology

Boundary submesh

- Transfer displacements and velocities
- Create pressure and gap fields (first iteration)

Low-order refined mesh (if required)

- Needed if order > 1
- Transfer displacements and velocities
- Transfer pressure

`updateMfemParallelDecomposition()`

Redecomp mesh

- Transfer displacements and velocities
- Transfer pressure
- Build Tribol meshes

- Apply Tribol contact methodology
- Compute forces and gaps
- Compute element Jacobian (implicit)

`update()`

- Transfer forces
- Transfer gaps
- NOTE: Jacobian transfer not yet implemented

`getMfemBlockJacobian()`

- Transfer forces
- Transfer gaps
- Transfer and assemble Jacobian (implicit)

`getMfemGap()`

Return

- Forces (on volume ParMesh)
- Gap (on boundary submesh)
- Assembled Jacobian

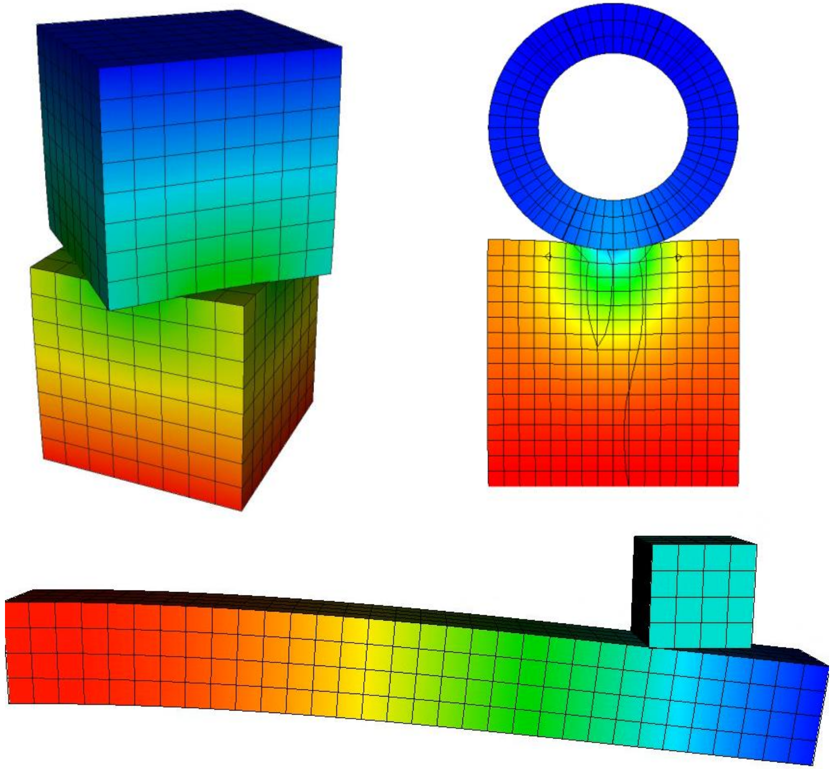
`getMfemResponse()`

Tribol MFEM interface

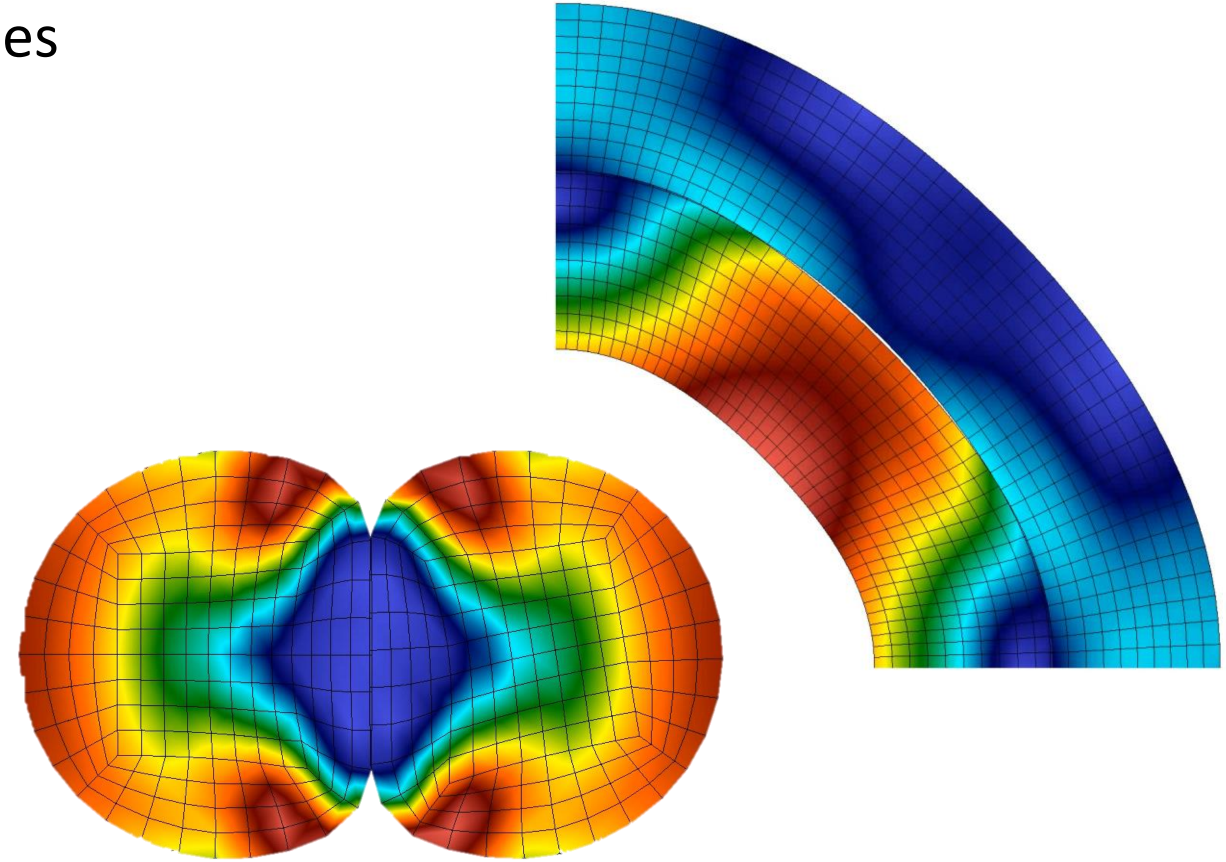
- Example problems
  - `src/examples/mfem_common_plane.cpp`
    - Simple two block impact problem
    - Common plane + penalty contact enforcement, hyperelastic solid, explicit finite elements
  - `src/examples/mfem_mortar_lm_patch.cpp`
    - Two block contact patch test
    - Mortar + Lagrange multiplier enforcement, elastic solid, implicit finite elements

Tribol MFEM interface

- Ongoing integration with multiphysics codes



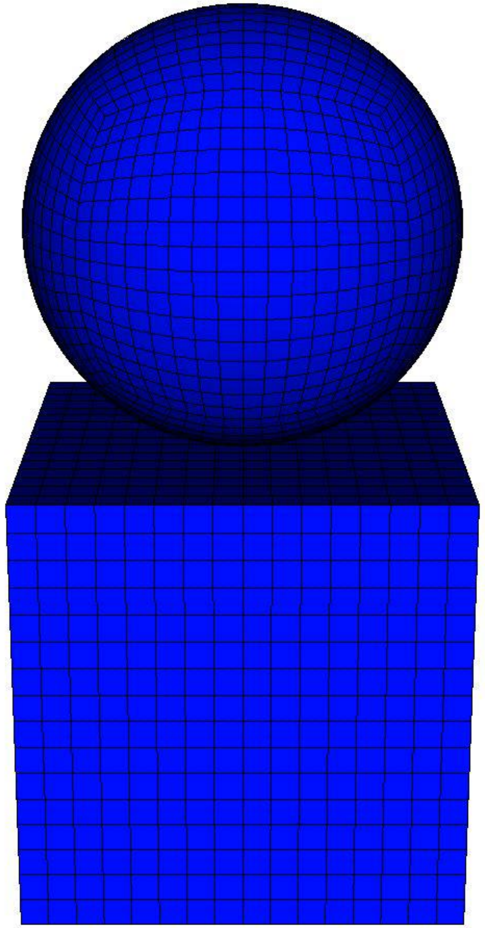
Serac (implicit mortar)

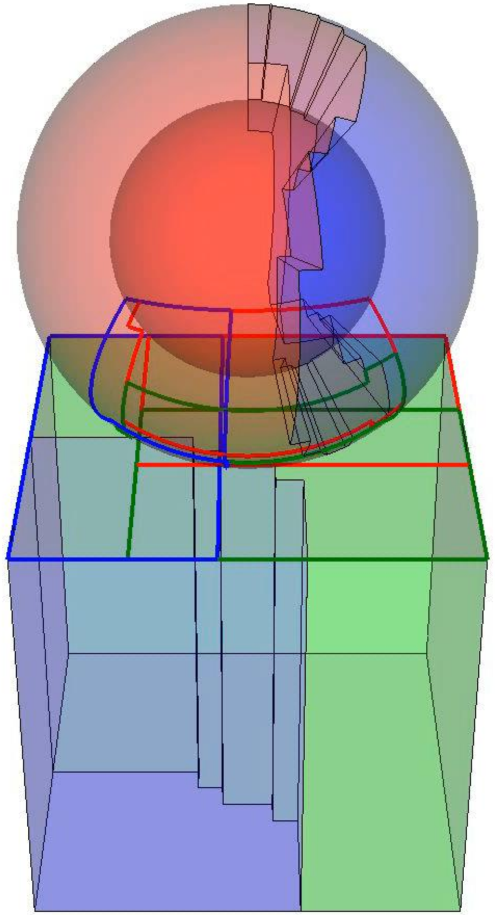


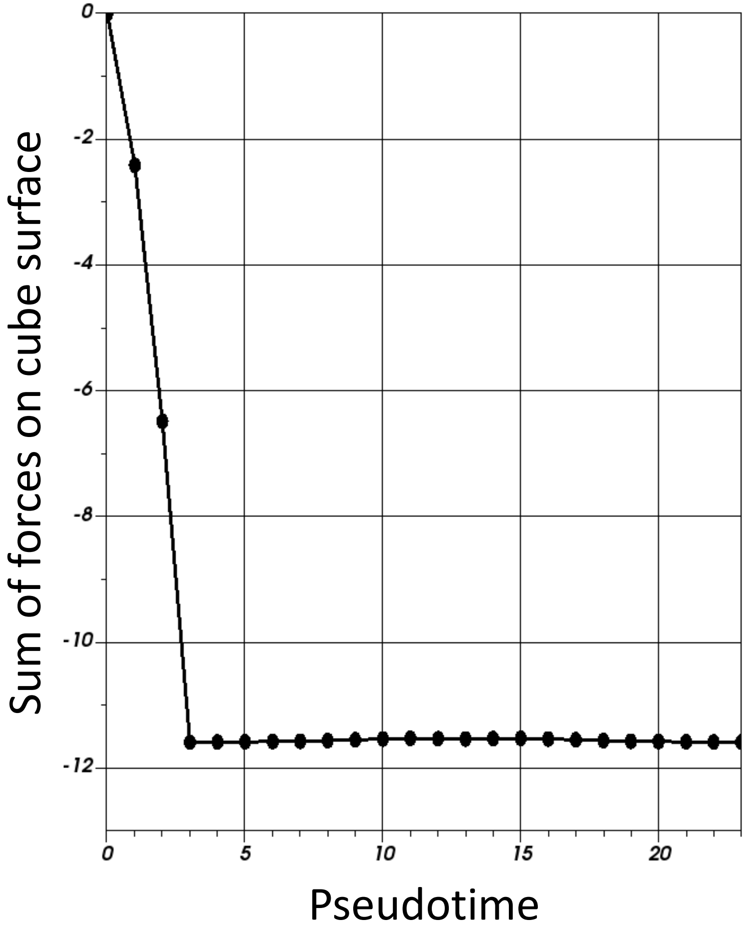
Blast (explicit common plane)

Example problem: quasistatic twisting sphere

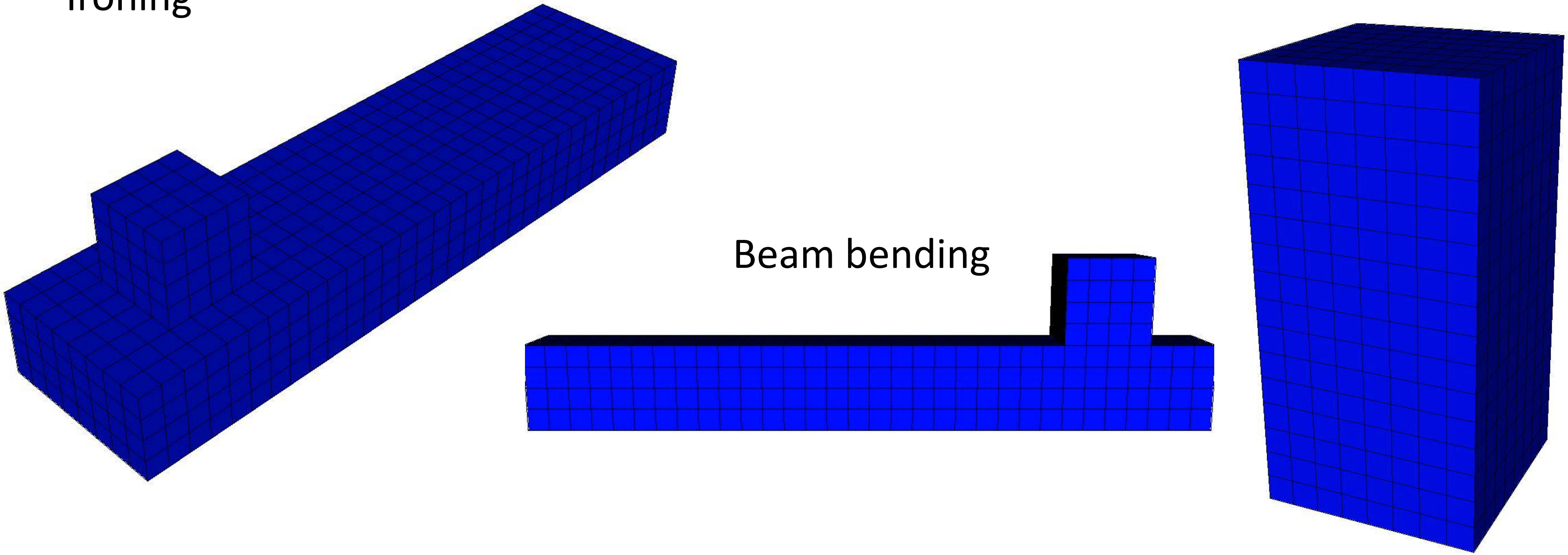
Serac implicit mortar, hyperelastic solid, 3 MPI ranks







More example problems



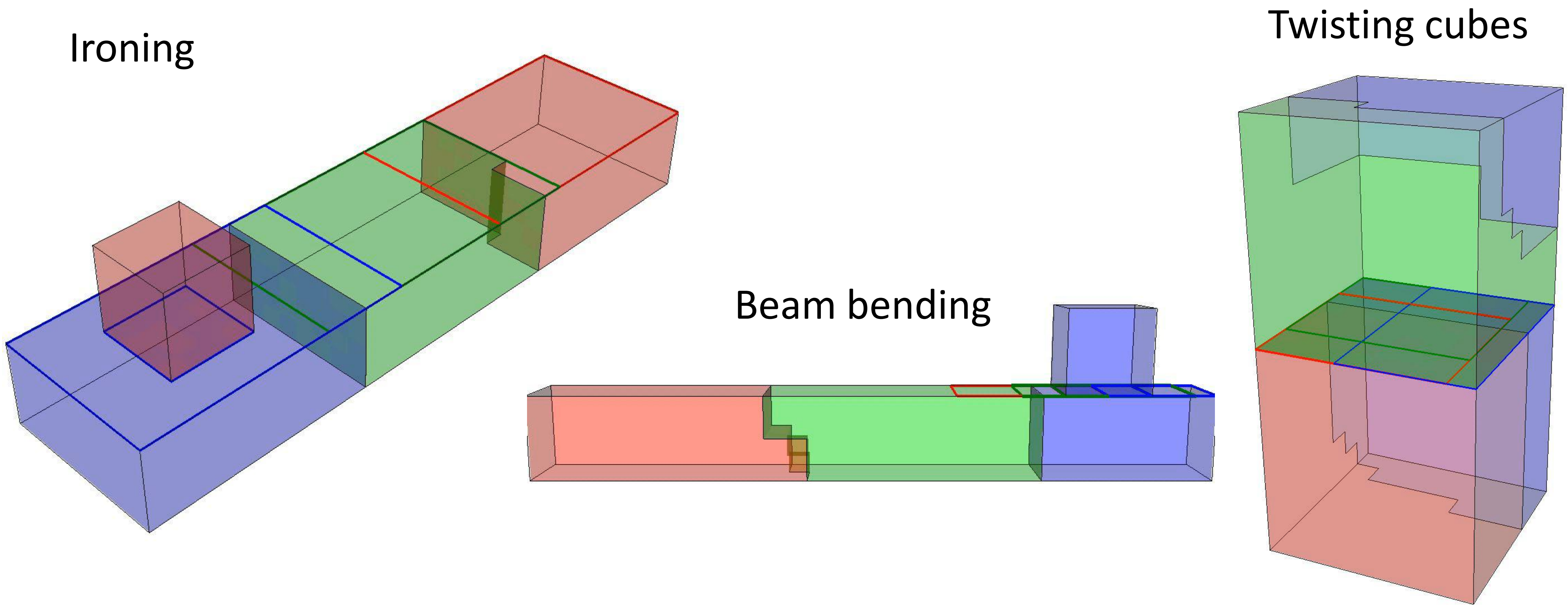
Ironing

Beam bending

Twisting cubes

More example problems

□ □ □ Surface redecomposition on-rank domains 🟥 🟩 🟦 MFEM on-rank domains



Ironing

Beam bending

Twisting cubes

Summary and conclusions

- Three main components of the Tribol interface physics library
  - Common plane and mortar methods
  - Domain redecomposer for MPI parallel contact
  - MFEM interface for simplifying Tribol usage with MFEM data structures
- Code is available on github: https://github.com/LLNL/Tribol
  - Serac integration (see https://github.com/LLNL/serac) coming soon (see PR #982)
- Coming next
  - GPU porting
  - Improved mortar contact algorithms
  - New/improved interface physics capabilities
  - Other code improvements
- **Thank you!**

Lawrence Livermore National Laboratory

Disclaimer
This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.