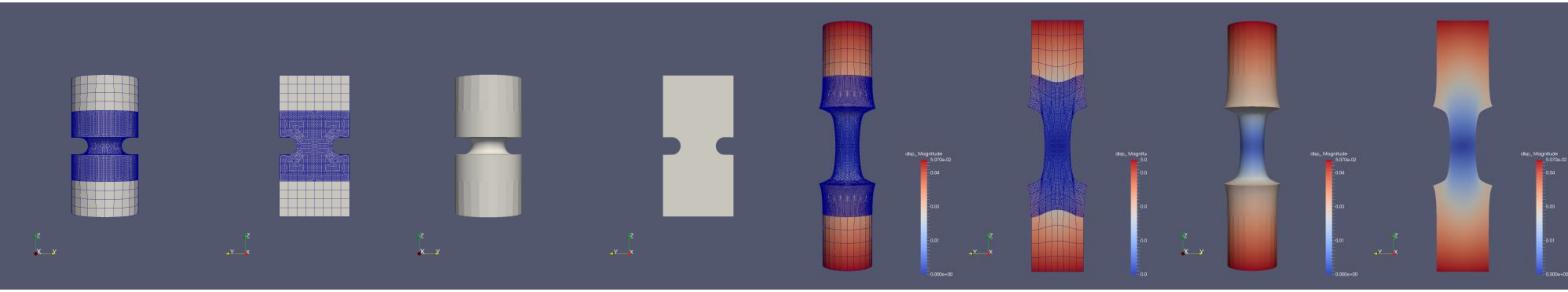
Sandia National Laboratories



The Schwarz alternating method as a means for concurrent multi-scale coupling of conventional and data-driven models

Irina Tezaur[1], Alejandro Mota[1], Coleman Alleman[1], Greg Phlipot[2], Joshua Barnett[1,3]

[1]Sandia National Labs [2]MSC Software [3]Stanford University

University of Nevada, Reno DMS Colloquium February 2, 2023

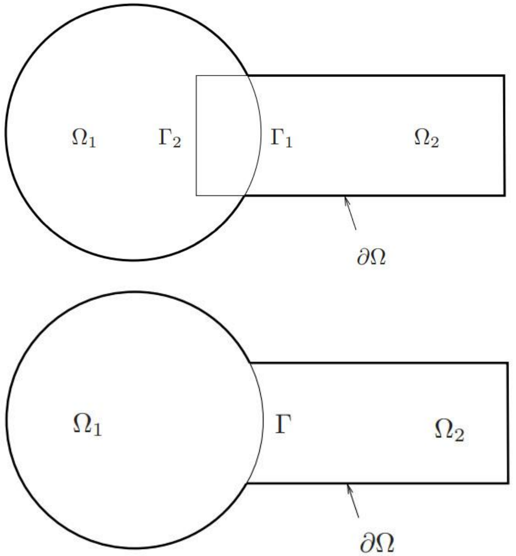
U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

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.

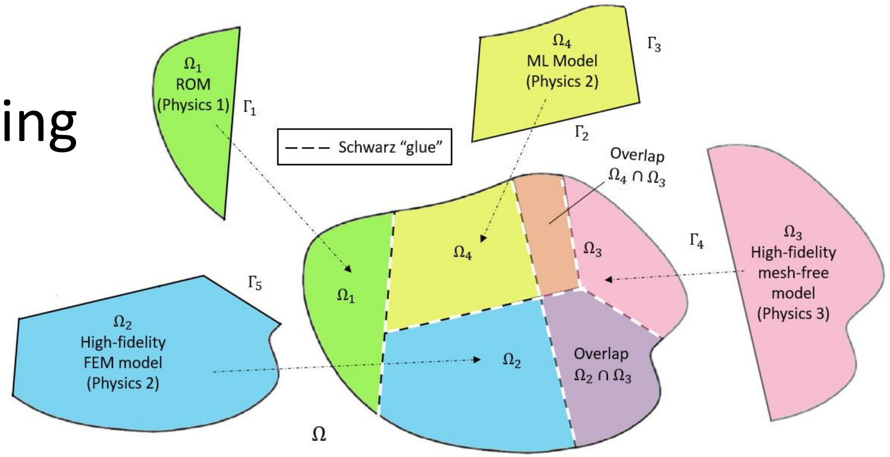
SAND2023-11341A

Outline

1. Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics
   - Motivation & Background
   - Quasistatics
   - Extension to Dynamics
   - Summary & Next Steps



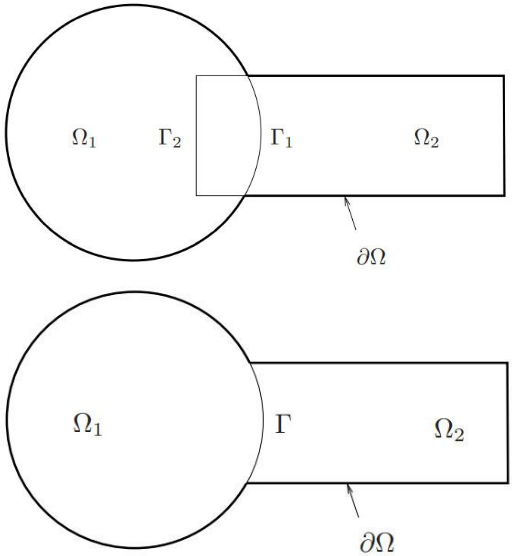
2. Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling
   - Motivation & Background
   - Demonstration
   - Ongoing & Future Work



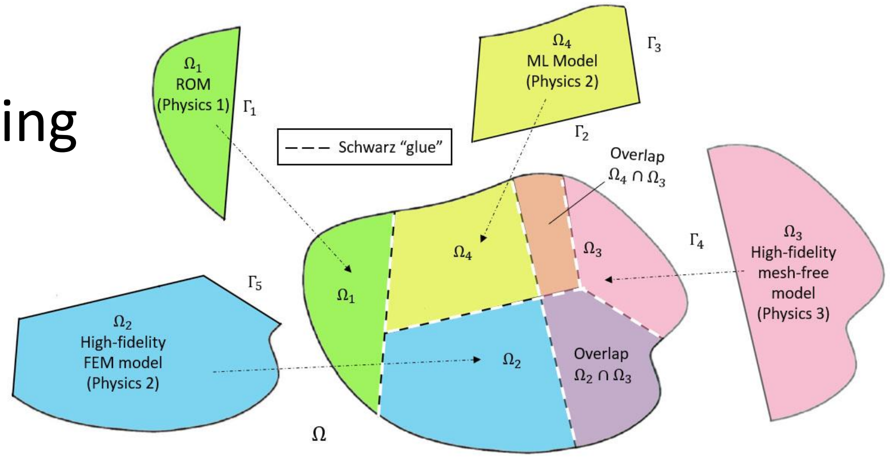
* Projection-based Reduced Order Model

Outline

1. **Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics**
   - **Motivation & Background**
   - Quasistatics
   - Extension to Dynamics
   - Summary & Next Steps



2. Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling
   - Motivation & Background
   - Demonstration
   - Ongoing & Future Work



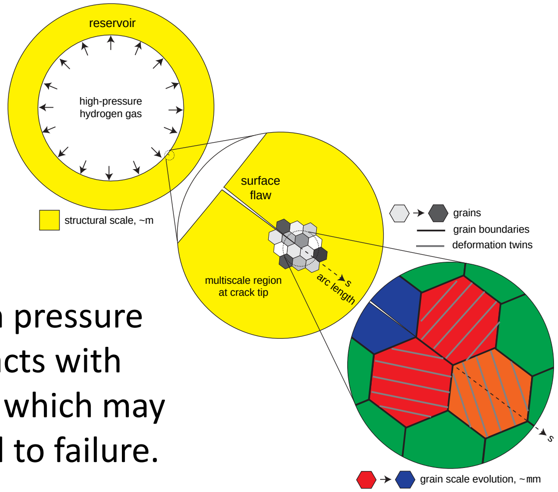
Motivation for Concurrent Multiscale Coupling

- ***Large scale*** structural ***failure*** frequently originates from ***small scale*** phenomena such as defects, microcracks, inhomogeneities and more, which grow quickly in unstable manner.

- Failure occurs due to ***tightly coupled interaction*** between small scale (stress concentrations, material instabilities, cracks, etc.) and large scale (vibration, impact, high loads and other perturbations).

Concurrent multiscale methods are ***essential*** for understanding and prediction of behavior of engineering systems when a ***small scale failure*** determines the performance of the entire system.

Roof failure of Boeing 737 aircraft due to fatigue cracks. From *imechanica.org*



Surface flaw in pressure vessel: interacts with microstructure, which may or may not lead to failure.

Previous Multiscale Coupling Work

Comput Mech (2014) 54:803–820
DOI 10.1007/s00466-014-1034-0

ORIGINAL PAPER

A multiscale overlapped coupling formulation for large-deformation strain localization

WaiChing Sun · Alejandro Mota

Received: 18 September 2013 / Accepted: 7 April 2014 / Published online: 3 May 2014

Abstract We generalize the multiscale overlapped domain framework to couple multiple rate-independent standard dissipative material models in the finite deformation regime across different length scales. We show that a fully coupled multiscale incremental boundary-value problem can be recast as the stationary point that optimizes the partitioned incremental work of a three-field energy functional. We also establish inf-sup tests to examine the numerical stability issues that arise from enforcing weak compatibility in the three-field formulation. We also devise a new block solver for the domain coupling problem and demonstrate the performance of the formulation with one-dimensional numerical examples. These simulations indicate that it is sufficient to introduce a localization limiter in a confined region of interest to regularize the partial differential equation if loss of ellipticity occurs.

strain localization may lead to the eventual failure of materials, this phenomenon is of significant importance to modern engineering applications.

The objective of this work is to introduce concurrent coupling between sub-scale and macro-scale simulations for inelastic materials that are prone to strain localization. Since it is not feasible to conduct sub-scale simulations on macroscopic problems, we use the domain coupling method such that computational resources can be efficiently allocated to regions of interest [14,23,24,30]. To the best of our knowledge, this is the first work focusing on utilizing the domain coupling method to model strain localization in inelastic materials undergoing large deformation.

Nevertheless, modeling strain localization with the conventional finite element method may lead to spurious mesh-dependent results due to the loss of ellipticity at the onset of strain localization [31]. To circumvent the loss of mate-

Previous Multiscale Coupling Work

Comput Mech (2014) 54:803–820
DOI 10.1007/s00466-014-1034-0

ORIGINAL PAPER

A multiscale overlapped coupling formulation for large-deformation strain localization

WaiChing Sun · Alejandro Mota

Received: 18 September 2013 / Accepted: 7 April 2014 / Published online: 3 May 2014

Three-field **multiscale coupling formulation with compatibility enforced weakly using** ***Lagrange multipliers*****.**

Abstract We generalize the multiscale overlapped domain framework to couple multiple rate-independent standard dissipative material models in the finite deformation regime across different length scales. We show that a fully coupled multiscale incremental boundary-value problem can be recast as the stationary point that optimizes the partitioned incremental work of a three-field energy functional. We also establish inf-sup tests to examine the numerical stability issues that arise from enforcing weak compatibility in the three-field formulation. We also devise a new block solver for the domain coupling problem and demonstrate the performance of the formulation with one-dimensional numerical examples. These simulations indicate that it is sufficient to introduce a localization limiter in a confined region of interest to regularize the partial differential equation if loss of ellipticity occurs.

strain localization may lead to the eventual failure of materials, this phenomenon is of significant importance to modern engineering applications.

The objective of this work is to introduce concurrent coupling between sub-scale and macro-scale simulations for inelastic materials that are prone to strain localization. Since it is not feasible to conduct sub-scale simulations on macroscopic problems, we use the domain coupling method such that computational resources can be efficiently allocated to regions of interest [14,23,24,30]. To the best of our knowledge, this is the first work focusing on utilizing the domain coupling method to model strain localization in inelastic materials undergoing large deformation.

Nevertheless, modeling strain localization with the conventional finite element method may lead to spurious mesh-dependent results due to the loss of ellipticity at the onset of strain localization [31]. To circumvent the loss of mate-

Previous Multiscale Coupling Work

Comput Mech (2014) 54:803–820
DOI 10.1007/s00466-014-1034-0

ORIGINAL PAPER

A multiscale overlapped coupling formulation for large-deformation strain localization

WaiChing Sun · Alejandro Mota

Received: 18 September 2013 / Accepted: 7 April 2014 / Published online: 3 May 2014

Abstract We generalize the multiscale overlapped domain framework to couple multiple rate-independent standard dissipative material models in the finite deformation regime across different length scales. We show that a fully coupled multiscale incremental boundary-value problem can be recast as the stationary point that optimizes the partitioned incremental work of a three-field energy functional. We also establish inf-sup tests to examine the numerical stability issues that arise from enforcing weak compatibility in the three-field formulation. We also devise a new block solver for the domain coupling problem and demonstrate the performance of the formulation with one-dimensional numerical examples. These simulations indicate that it is sufficient to introduce a localization limiter in a confined region of interest to regularize the partial differential equation if loss of ellipticity occurs.

strain localization may lead to the eventual failure of materials, this phenomenon is of significant importance to modern engineering applications.

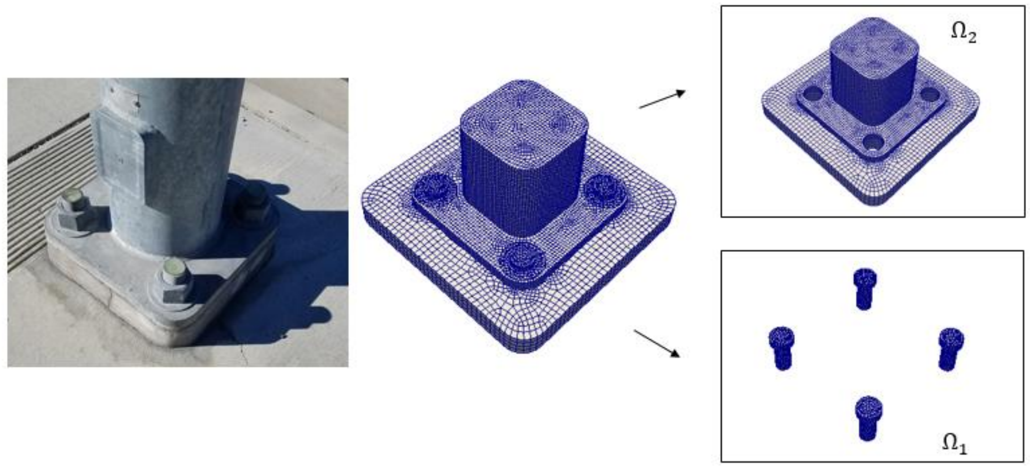
The objective of this work is to introduce concurrent coupling between sub-scale and macro-scale simulations for inelastic materials that are prone to strain localization. Since it is not feasible to conduct sub-scale simulations on macroscopic problems, we use the domain coupling method such that computational resources can be efficiently allocated to regions of interest [14,23,24,30]. To the best of our knowledge, this is the first work focusing on utilizing the domain coupling method to model strain localization in inelastic materials undergoing large deformation.

Nevertheless, modeling strain localization with the conventional finite element method may lead to spurious mesh-dependent results due to the loss of ellipticity at the onset of strain localization [31]. To circumvent the loss of mate-

Method works well, but is ***difficult to implement*** into existing codes.

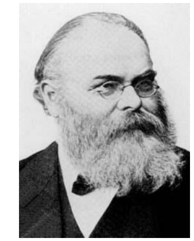
Requirements for Multiscale Coupling Method

- Coupling is ***concurrent*** (two-way).
- ***Ease of implementation*** into existing massively-parallel HPC codes.
- ***Scalable, fast, robust*** (we target ***real*** engineering problems, e.g., analyses involving failure of bolted components!).
- ***"Plug-and-play" framework***: simplifies task of meshing complex geometries
  - Ability to couple regions with ***different non-conformal meshes***, ***different element types*** and ***different levels of refinement.***
  - Ability to use ***different solvers/time-integrators*** in different regions.
- Coupling does not introduce ***nonphysical artifacts.***
- ***Theoretical*** convergence properties/guarantees.



Schwarz Alternating Method for Domain Decomposition

- Proposed in 1870 by H. Schwarz for solving Laplace PDE on irregular domains.



H. Schwarz
(1843–1921)

Crux of Method: if the solution is known in regularly shaped domains, use those as pieces to iteratively build a solution for the more complex domain.

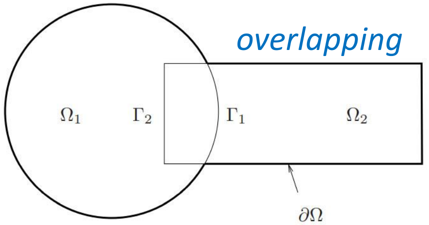
Basic Schwarz Algorithm

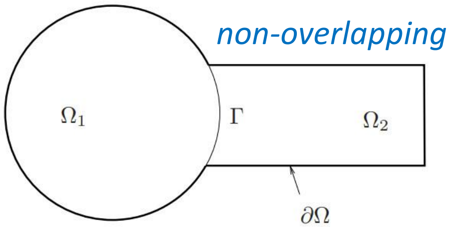
Initialize:

- Solve PDE by any method on Ω_1 w/ initial guess for transmission BCs on Γ_1.

Iterate until convergence:

- Solve PDE by any method on Ω_2 w/ transmission BCs on Γ_2 based on values just obtained for Ω_1.
- Solve PDE by any method on Ω_1 w/ transmission BCs on Γ_1 based on values just obtained for Ω_2.



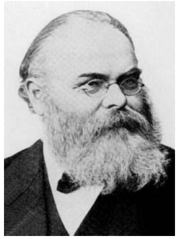


Schwarz Alternating Method for Domain Decomposition

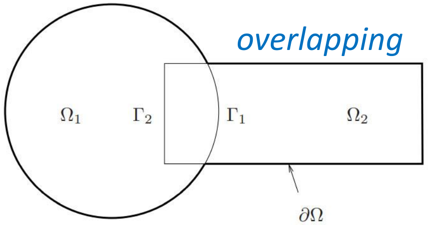
Sandia National Laboratories

- Proposed in 1870 by H. Schwarz for solving Laplace PDE on irregular domains.

Crux of Method: if the solution is known in regularly shaped domains, use those as pieces to iteratively build a solution for the more complex domain.



H. Schwarz (1843–1921)



Basic Schwarz Algorithm

Initialize:

- Solve PDE by any method on Ω_1 w/ initial guess for transmission BCs on Γ_1.

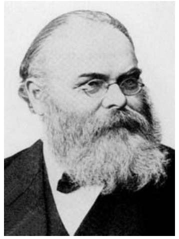
Iterate until convergence:

- Solve PDE by any method on Ω_2 w/ transmission BCs on Γ_2 based on values just obtained for Ω_1.
- Solve PDE by any method on Ω_1 w/ transmission BCs on Γ_1 based on values just obtained for Ω_2.

Overlapping Schwarz: convergent with all-Dirichlet transmission BCs[1] if $\Omega_1 \cap \Omega_2 \neq \emptyset$.

[1]Schwarz, 1870; Lions, 1988.

Schwarz Alternating Method for Domain Decomposition



H. Schwarz
(1843–1921)

- Proposed in 1870 by H. Schwarz for solving Laplace PDE on irregular domains.

Crux of Method: if the solution is known in regularly shaped domains, use those as pieces to iteratively build a solution for the more complex domain.

Basic Schwarz Algorithm

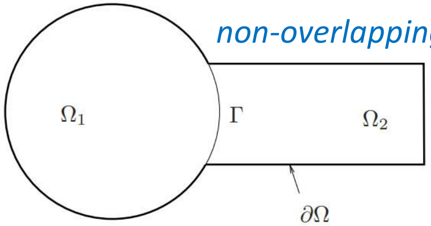
Initialize:

- Solve PDE by any method on Ω_1 w/ initial guess for transmission BCs on Γ_1.

Iterate until convergence:

- Solve PDE by any method on Ω_2 w/ transmission BCs on Γ_2 based on values just obtained for Ω_1.
- Solve PDE by any method on Ω_1 w/ transmission BCs on Γ_1 based on values just obtained for Ω_2.

non-overlapping



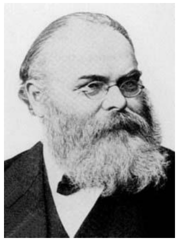
Non-overlapping Schwarz: convergent with Robin-Robin[2] or alternating Neumann-Dirichlet[3] transmission BCs.

[2]Lions, 1990. [3]Zanolli *et al.*, 1987.

Schwarz Alternating Method for Domain Decomposition

- Proposed in 1870 by H. Schwarz for solving Laplace PDE on irregular domains.

Crux of Method: if the solution is known in regularly shaped domains, use those as pieces to iteratively build a solution for the more complex domain.



H. Schwarz
(1843–1921)

Basic Schwarz Algorithm

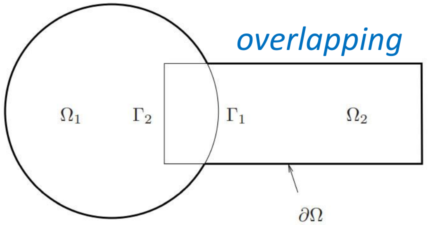
Initialize:

- Solve PDE by any method on Ω_1 w/ initial guess for transmission BCs on Γ_1.

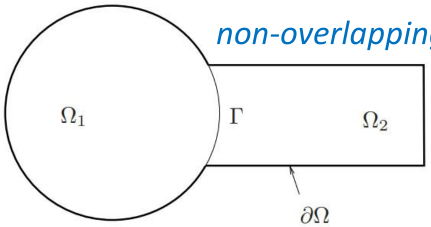
Iterate until convergence:

- Solve PDE by any method on Ω_2 w/ transmission BCs on Γ_2 based on values just obtained for Ω_1.
- Solve PDE by any method on Ω_1 w/ transmission BCs on Γ_1 based on values just obtained for Ω_2.

overlapping



non-overlapping



Overlapping Schwarz: convergent with all-Dirichlet transmission BCs[1] if $\Omega_1 \cap \Omega_2 \neq \emptyset$.

Non-overlapping Schwarz: convergent with Robin-Robin[2] or alternating Neumann-Dirichlet[3] transmission BCs.

Common use of Schwarz: preconditioner for Krylov iterative methods to solve linear systems.

[1]Schwarz, 1870; Lions, 1988. [2]Lions, 1990. [3]Zanolli *et al.*, 1987.

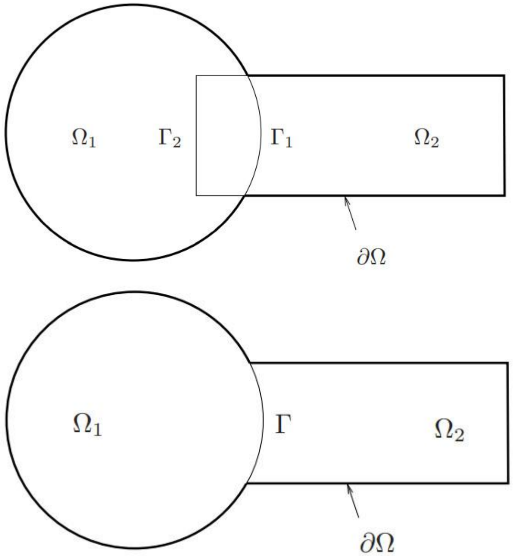
How we use the Schwarz Alternating Method

AS A *PRECONDITIONER* FOR THE LINEARIZED SYSTEM

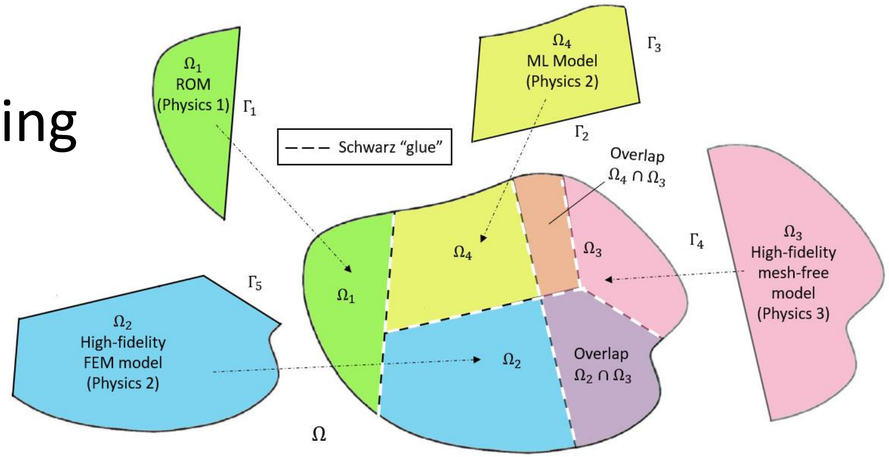
AS A *SOLVER* FOR THE COUPLED FULLY NONLINEAR PROBLEM

Outline

1. **Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics**
   - Motivation & Background
   - **Quasistatics**
   - Extension to Dynamics
   - Summary & Next Steps



2. Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling
   - Motivation & Background
   - Demonstration
   - Ongoing & Future Work



* Projection-based Reduced Order Model

Quasistatic Solid Mechanics Formulation

- **Energy functional** defining weak form of the governing PDEs

$$\boxed{\Phi[\boldsymbol{\varphi}] := \int_\Omega A(\boldsymbol{F},\boldsymbol{Z})dV - \int_\Omega \rho\boldsymbol{B}\cdot\boldsymbol{\varphi}dV}$$

  - $A(\boldsymbol{F},\boldsymbol{Z})$: Helmholtz free-energy density
  - $\boldsymbol{F} := \nabla\boldsymbol{\varphi}$: deformation gradient
  - $\boldsymbol{Z}$: collection of internal variables (for plastic materials)
  - ρ: density, $\boldsymbol{B}$: body force

- **Euler-Lagrange equations**, obtained by minimizing $\Phi[\boldsymbol{\varphi}]$: $\begin{cases}\mathrm{Div}\,\boldsymbol{P} + \rho\boldsymbol{B} = \boldsymbol{0}, & \text{in } \Omega \\ \boldsymbol{\varphi} = \boldsymbol{\chi}, & \text{on } \partial\Omega\end{cases}$

- Quasistatics solves **sequence of problems** in which loading (body force) $\boldsymbol{B}$ is incremented **quasistatically** w.r.t. **pseudo time** t_i:

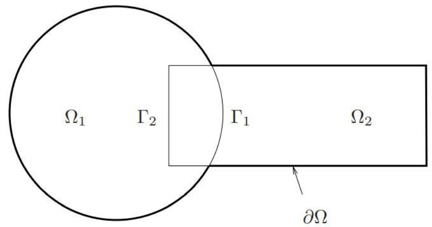
For $i = 1, \dots, n$
  Solve $\mathrm{Div}\,\boldsymbol{P} + \rho\boldsymbol{B}(t_i) = \boldsymbol{0}$ with appropriate boundary conditions (BCs)
  Increment pseudo time t_i to obtain t_{i+1}

Spatial Coupling via Alternating Schwarz

Overlapping Domain Decomposition

$$\begin{cases} \mathrm{Div}\, \boldsymbol{P}_1^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0} \text{ , in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, \quad \text{on } \partial\Omega_1 \backslash \Gamma_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\varphi}_2^{(n)} \quad \text{on } \Gamma_2 \end{cases}$$

$$\begin{cases} \mathrm{Div}\, \boldsymbol{P}_2^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0} \text{ , in } \Omega_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\chi}, \quad \text{on } \partial\Omega_2 \backslash \Gamma_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\varphi}_1^{(n+1)} \quad \text{on } \Gamma_2 \end{cases}$$



Model PDE: $\begin{cases} \mathrm{Div}\, \boldsymbol{P} + \rho \boldsymbol{B} = \boldsymbol{0} \text{ , in } \Omega \\ \boldsymbol{\varphi} = \boldsymbol{\chi}, \quad \text{on } \partial\Omega \end{cases}$

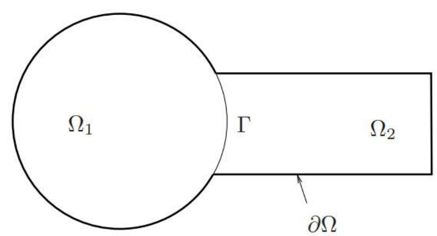
- Dirichlet-Dirichlet transmission BCs [Schwarz, 1870; Lions, 1988]

Non-overlapping Domain Decomposition

$$\begin{cases} \mathrm{Div}\, \boldsymbol{P}_1^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0} \text{ , in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, \quad \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\lambda}_{n+1} \quad \text{on } \Gamma \end{cases}$$

$$\begin{cases} \mathrm{Div}\, \boldsymbol{P}_2^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0} \text{ , in } \Omega_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\chi}, \quad \text{on } \partial\Omega_2 \backslash \Gamma \\ \boldsymbol{P}_2^{(n+1)} \boldsymbol{n} = \boldsymbol{P}_2^{(n+1)} \boldsymbol{n}, \quad \text{on } \Gamma \end{cases}$$

$$\boldsymbol{\lambda}_{n+1} = \theta \boldsymbol{\varphi}_2^{(n)} + (1-\theta)\boldsymbol{\lambda}_n, \text{on } \Gamma, \text{for } n \geq 1$$



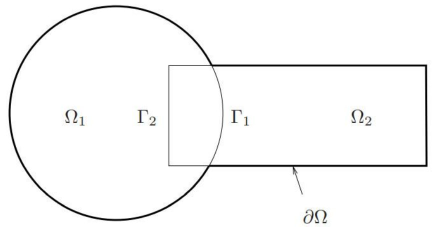
- Relevant for multi-material and multi-physics coupling
- Alternating Dirichlet-Neumann transmission BCs [Zanolli *et al.*, 1987]
- Robin-Robin transmission BCs also lead to convergence [Lions,1990]
- $\theta \in [0,1]$: relaxation parameter (can help convergence)

Spatial Coupling via Alternating Schwarz

Overlapping Domain Decomposition

$$\begin{cases} \operatorname{Div} \boldsymbol{P}_1^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0}\,, & \text{in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_1 \backslash \Gamma_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\varphi}_2^{(n)} & \text{on } \Gamma_2 \end{cases}$$

$$\begin{cases} \operatorname{Div} \boldsymbol{P}_2^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0}\,, & \text{in } \Omega_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_2 \backslash \Gamma_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\varphi}_1^{(n+1)} & \text{on } \Gamma_2 \end{cases}$$



Model PDE: $\begin{cases} \operatorname{Div} \boldsymbol{P} + \rho \boldsymbol{B} = \boldsymbol{0}\,, & \text{in } \Omega \\ \boldsymbol{\varphi} = \boldsymbol{\chi}, & \text{on } \partial\Omega \end{cases}$

- Dirichlet-Dirichlet transmission BCs [Schwarz, 1870; Lions, 1988]

Part 1 of talk

Non-overlapping Domain Decomposition

Part 2 of talk

$$\begin{cases} \operatorname{Div} \boldsymbol{P}_1^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0}\,, & \text{in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\lambda}_{n+1} & \text{on } \Gamma \end{cases}$$

$$\begin{cases} \operatorname{Div} \boldsymbol{P}_2^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0}\,, & \text{in } \Omega_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_2 \backslash \Gamma \\ \boldsymbol{P}_2^{(n+1)} \boldsymbol{n} = \boldsymbol{P}_2^{(n+1)} \boldsymbol{n}, & \text{on } \Gamma \end{cases}$$

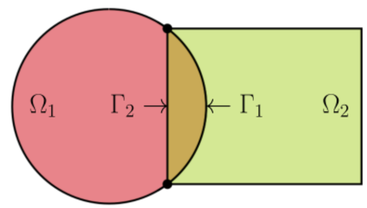
$$\boldsymbol{\lambda}_{n+1} = \theta \boldsymbol{\varphi}_2^{(n)} + (1-\theta)\boldsymbol{\lambda}_n, \text{on } \Gamma, \text{for } n \geq 1$$

- Relevant for multi-material and multi-physics coupling
- Alternating Dirichlet-Neumann transmission BCs [Zanolli *et al.*, 1987]
- Robin-Robin transmission BCs also lead to convergence [Lions,1990]
- $\theta \in [0,1]$: relaxation parameter (can help convergence)

Overlapping Schwarz Coupling in Quasistatics

outer quasistatic loop

$\varphi^{(0)} \leftarrow \mathrm{id}_{\boldsymbol{X}}$ in Ω_2 ▷ initialize to zero displacement or a better guess in Ω_2
$n \leftarrow 1$
repeat ▷ Schwarz loop
  $\varphi^{(n)} \leftarrow \chi$ on $\partial_{\varphi}\Omega_i$ ▷ Dirichlet BC for Ω_i
  $\varphi^{(n)} \leftarrow P_{\Omega_j \to \Gamma_i}[\varphi^{(n-1)}]$ on Γ_i ▷ Schwarz BC for Ω_i
  $\varphi^{(n)} \leftarrow \arg\min_{\varphi \in S_i} \Phi_i[\varphi]$ in Ω_i ▷ solve in Ω_i
  $n \leftarrow n + 1$
until converged



Advantages:

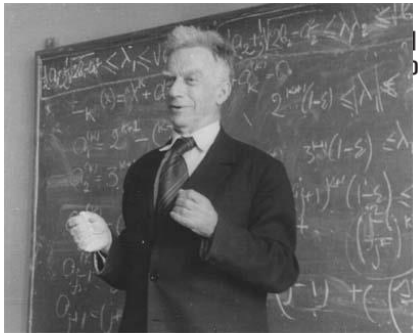
- Conceptually very ***simple***.
- Allows the coupling of regions with ***different non-conforming meshes***, ***different element types***, and ***different levels of refinement***.
- Information is exchanged among two or more regions, making coupling ***concurrent***.
- ***Different solvers*** can be used for the different regions.
- ***Different material models*** can be coupled if they are compatible in the overlap region.
- Simplifies the task of ***meshing complex geometries*** for the different scales.

Theoretical Foundation

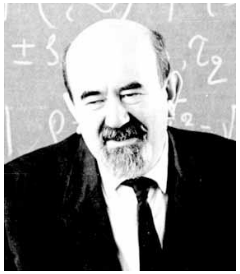
Using the Schwarz alternating as a ***discretization method*** for PDEs is natural idea with a sound ***theoretical foundation***.

- S.L. Sobolev (1936): posed Schwarz method for ***linear elasticity*** in variational form and ***proved method's convergence*** by proposing a convergent sequence of energy functionals.
- **S.G. Mikhlin (1951):** ***proved convergence*** of Schwarz method for general linear elliptic PDEs.
- **P.-L. Lions (1988):** studied convergence of Schwarz for ***nonlinear monotone elliptic problems*** using max principle.
- **A. Mota, I. Tezaur, C. Alleman (2017):** proved ***convergence*** of the alternating Schwarz method for ***finite deformation quasi-static nonlinear PDEs*** (with energy functional $\boldsymbol{\Phi}[\boldsymbol{\varphi}]$) with a ***geometric convergence rate.***

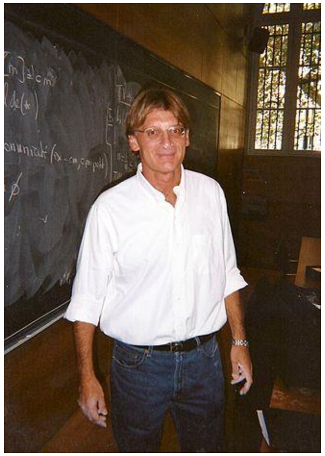
$$\boldsymbol{\Phi}[\boldsymbol{\varphi}] = \int_B A(\boldsymbol{F},\boldsymbol{Z})\, dV - \int_B \boldsymbol{B}\cdot\boldsymbol{\varphi}\, dV$$
$$\nabla\cdot\boldsymbol{P}+\boldsymbol{B}=\boldsymbol{0}$$



S.L. Sobolev (1908 – 1989)



S.G. Mikhlin (1908 – 1990)



P.- L. Lions (1956-)

A. Mota, I. Tezaur, C. Alleman

Convergence Proof*

Theorem 1. *Assume that the energy functional $\Phi[\varphi]$ satisfies properties 1–5 above. Consider the Schwarz alternating method of Section 2 defined by (9)–(13) and its equivalent form (39). Then*

(a) $\Phi[\tilde{\varphi}^{(0)}] \geq \Phi[\tilde{\varphi}^{(1)}] \geq \cdots \geq \Phi[\tilde{\varphi}^{(n-1)}] \geq \Phi[\tilde{\varphi}^{(n)}] \geq \cdots \geq \Phi[\varphi]$, *where φ is the minimizer of $\Phi[\varphi]$ over $\mathcal{S}$.*

(b) *The sequence $\left\{\tilde{\varphi}^{(n)}\right\}$ defined in (39) converges to the minimizer φ of $\Phi[\varphi]$ in $\mathcal{S}$.*

(c) *The Schwarz minimum values $\Phi[\tilde{\varphi}^{(n)}]$ converge monotonically to the minimum value $\Phi[\varphi]$ in $\mathcal{S}$ starting from any initial guess $\tilde{\varphi}^{(0)}$.*

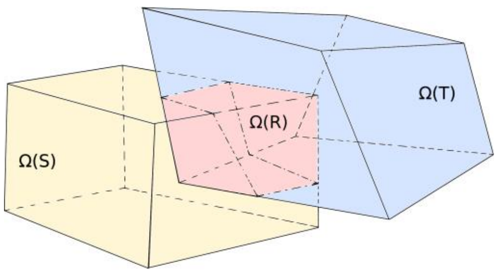
Implementation in *Albany LCM* Code

The proposed ***quasistatic alternating Schwarz method*** is implemented within the ***Albany LCM*** open-source parallel, C++, multi-physics, finite element code.

- ***Component-based*** design for rapid development.
- Contains a wide variety of ***constitutive models***.
- Extensive use of libraries from the open-source ***Trilinos*** project.
  - Use of the ***Phalanx*** package to decompose complex problem into simpler problems with managed dependencies.
  - Use of the ***Sacado*** package for ***automatic differentiation.***
  - Use of ***Teko*** package for block preconditioning for Modified and Monolithic Schwarz variants.
- ***Parallel*** implementation of Schwarz alternating method uses the ***Data Transfer Kit*** (***DTK***).
- All software available on ***GitHub***.

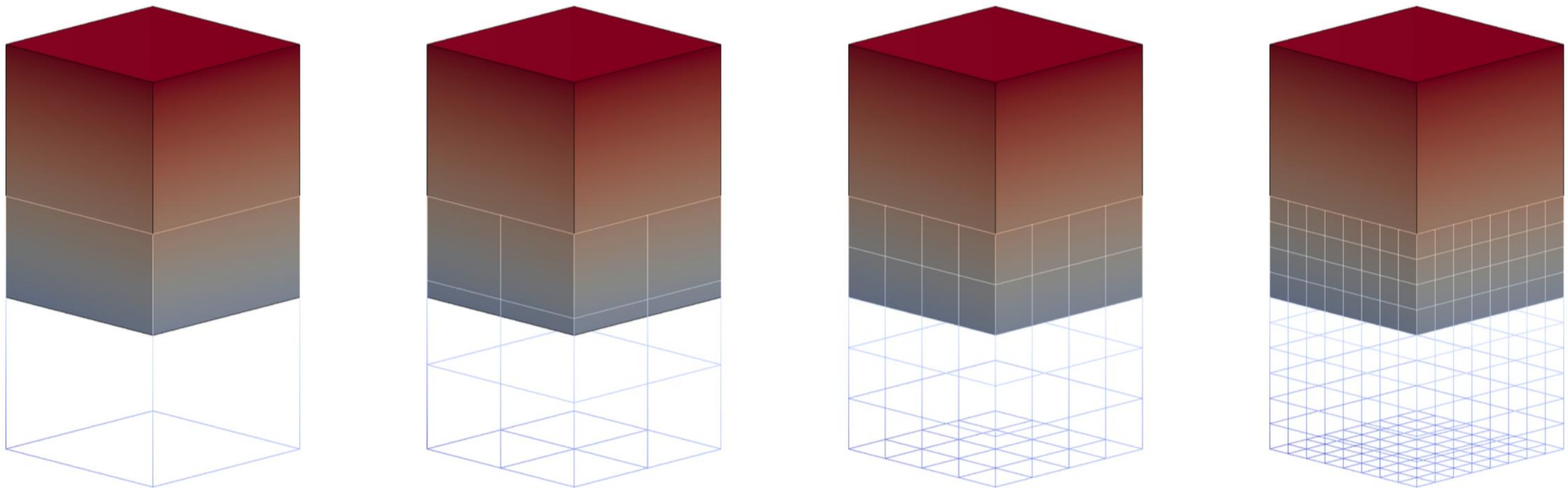
https://github.com/SNL Computation/LCM

https://github.com/trilinos/trilinos

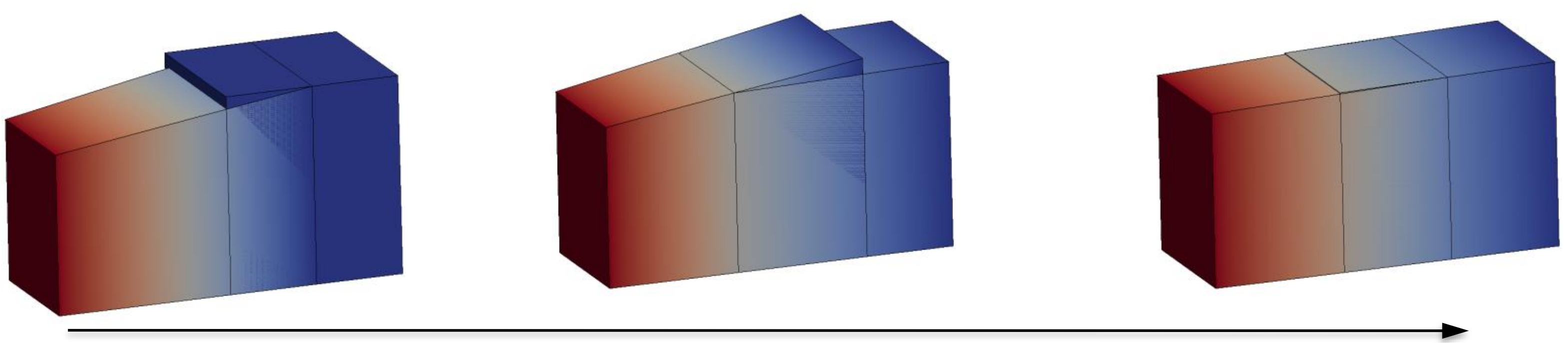


https://github.com/ORNL-CEES/DataTransferKit

Quasistatic Example #1: Cuboid Problem

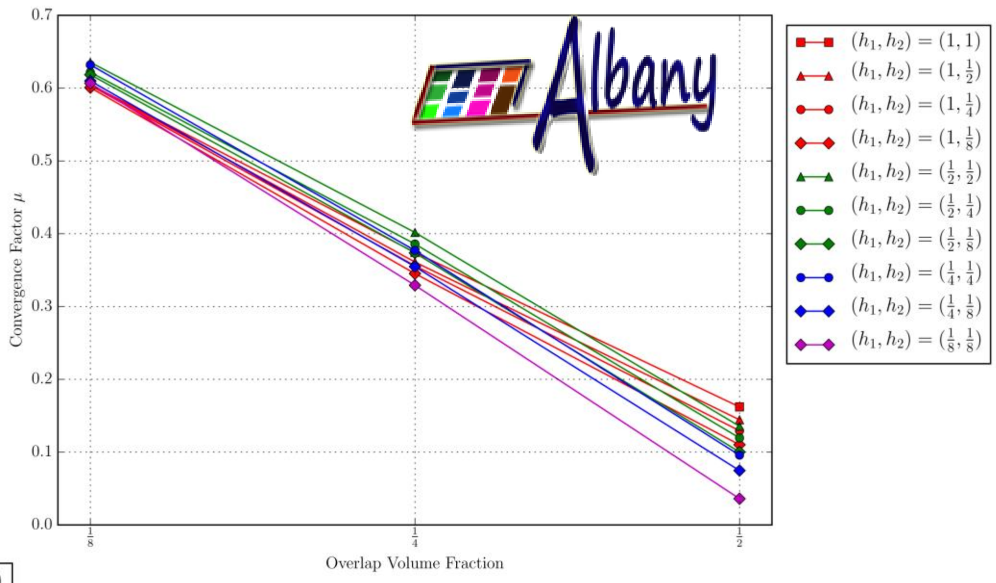


- Coupling of ***two cuboids*** with square base (above).
- ***Neohookean***-type material model.

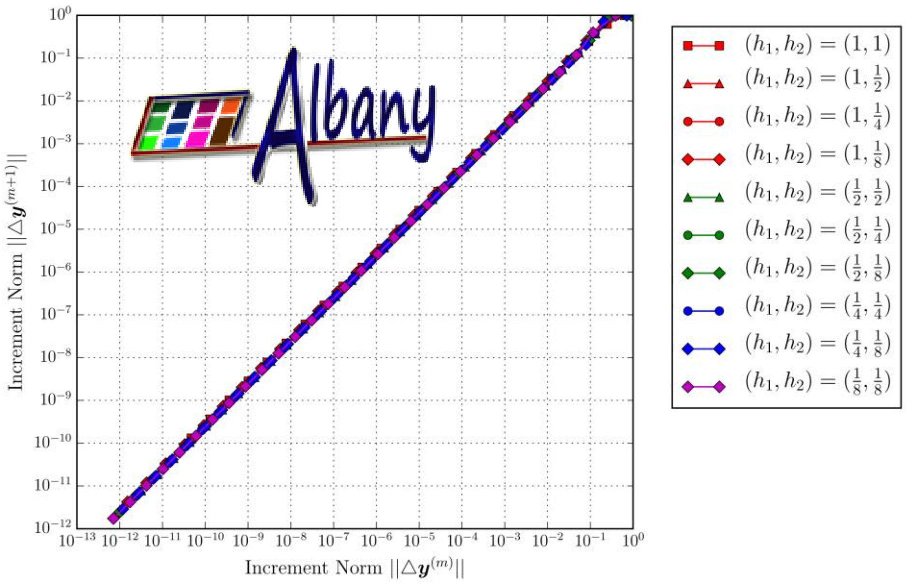


Schwarz Iteration

Cuboid Problem: Convergence with Overlap & Refinement



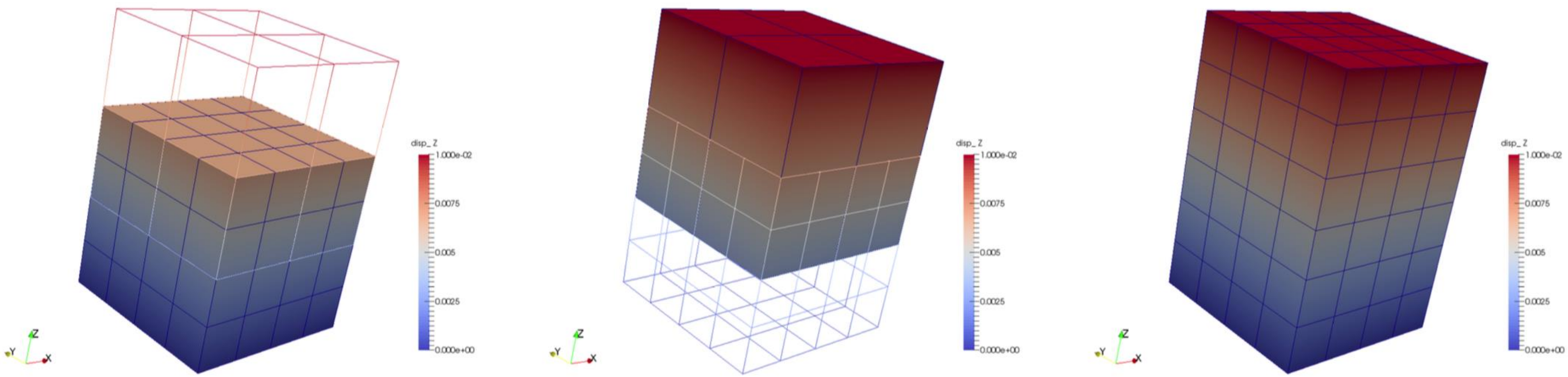
Below: Convergence of the cuboid problem for different mesh sizes and fixed overlap volume fraction. The Schwarz alternating method converges ***linearly***.



Above: Convergence factor μ as a function of overlap volume and different mesh. There is ***faster linear convergence*** with increasing ***overlap volume fraction***.

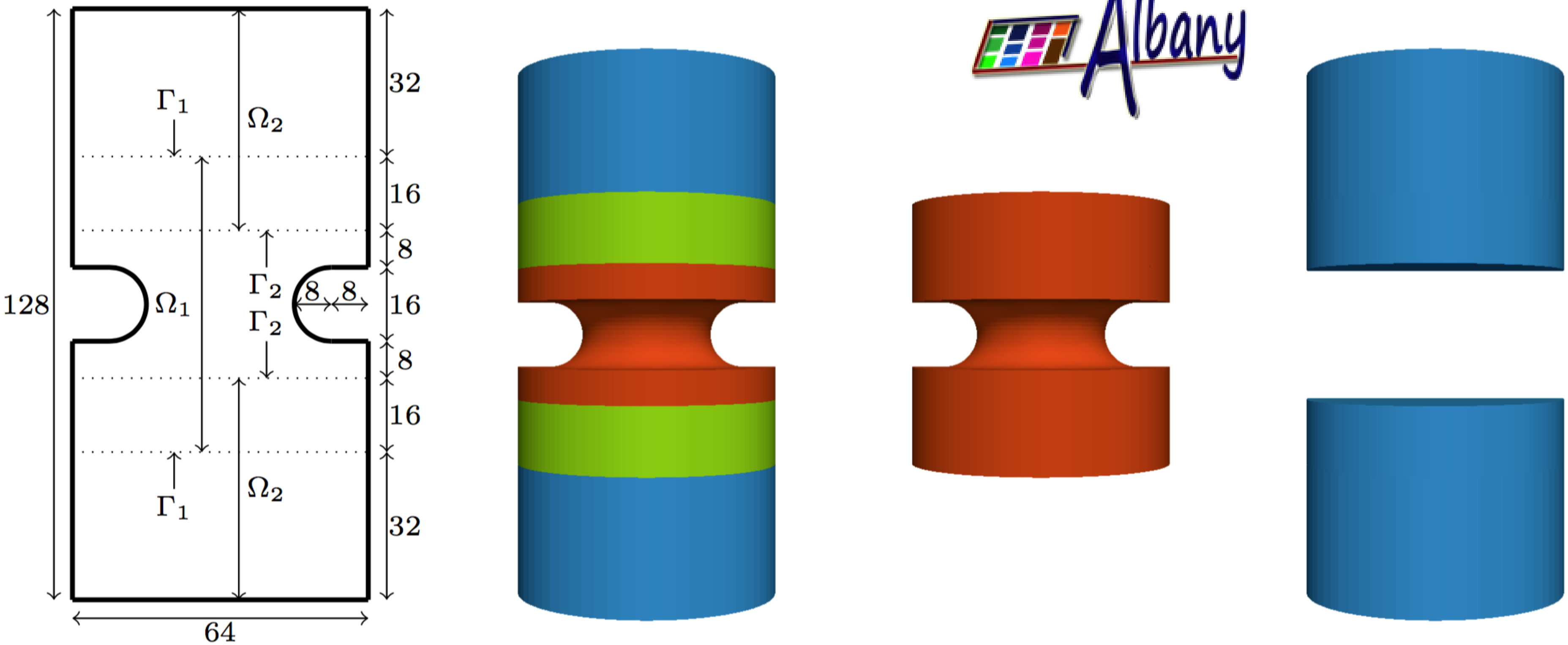
$$\Delta y^{(m+1)} \leq \mu \Delta y^{(m)}$$

Cuboid Problem: Schwarz Error



Subdomain	u_3 relative error	σ_{33} relative error
Ω_1	1.24×10^{-14}	2.31×10^{-13}
Ω_2	7.30×10^{-15}	3.06×10^{-13}

Quasistatic Example #2: Notched Cylinder

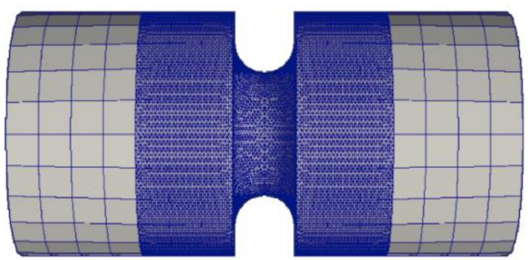


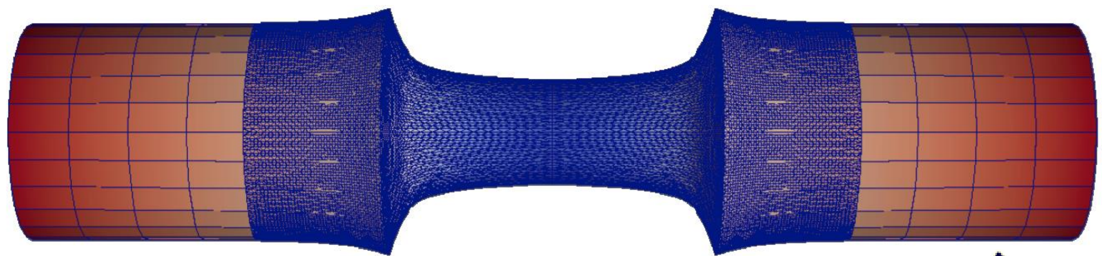
(a) Schematic **(b)** Entire Domain Ω **(c)** Fine Region Ω_1 **(d)** Coarse Region Ω_2

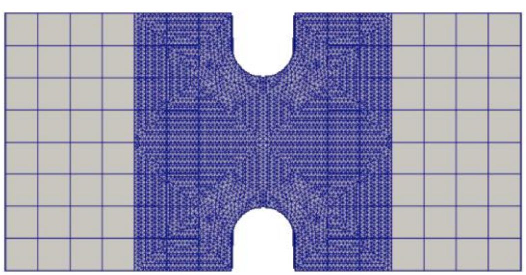
- ***Notched cylinder*** that is stretched along its axial direction.
- Domain decomposed into ***two subdomains***.
- ***Neohookean***-type material model.

Notched Cylinder: TET - HEX Coupling

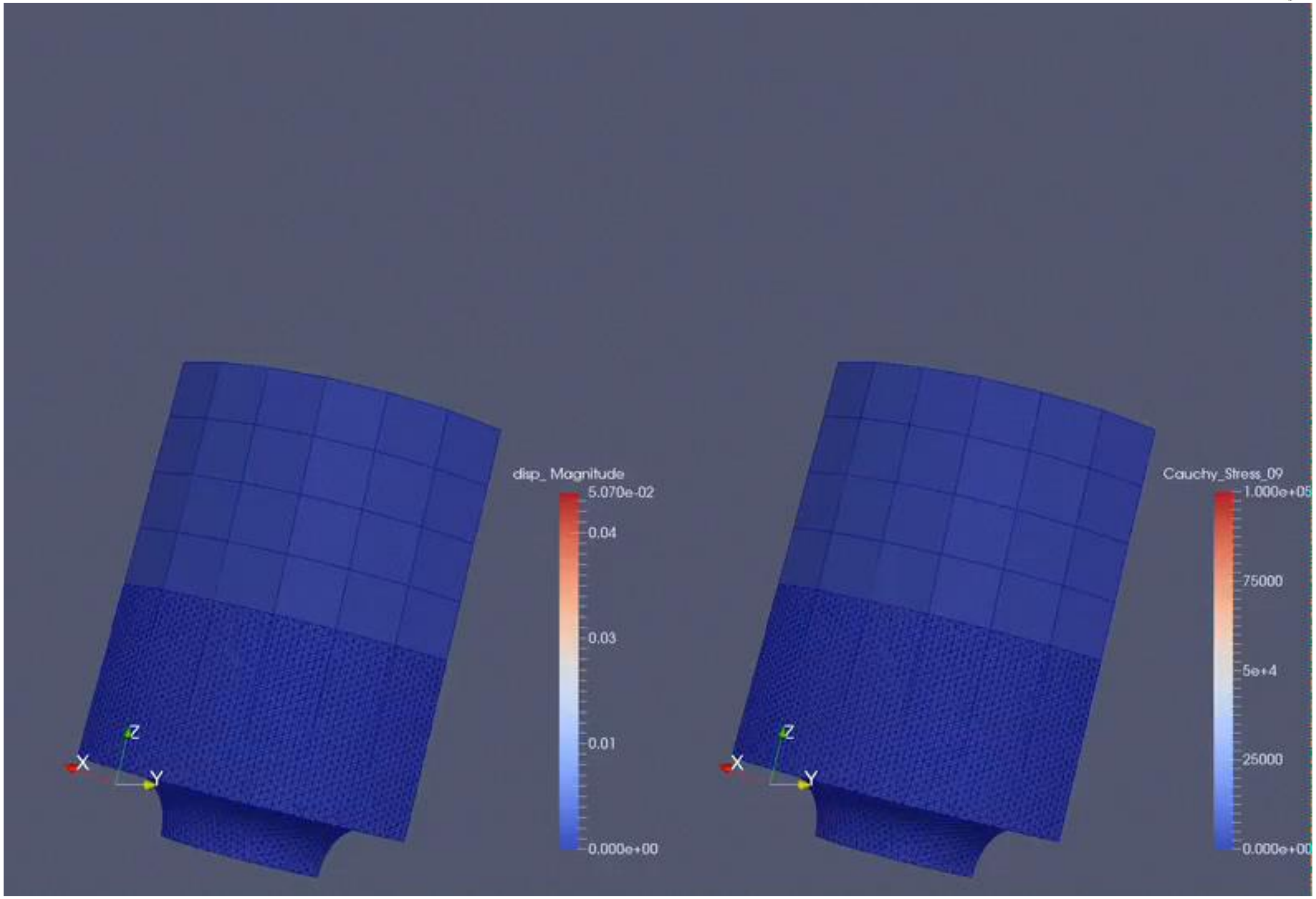
- The Schwarz alternating method is capable of coupling ***different mesh topologies***.
- The notched region, where stress concentrations are expected, is ***finely*** meshed with ***tetrahedral*** elements.
- The top and bottom regions, presumably of less interest, are meshed with ***coarser hexahedral*** elements.



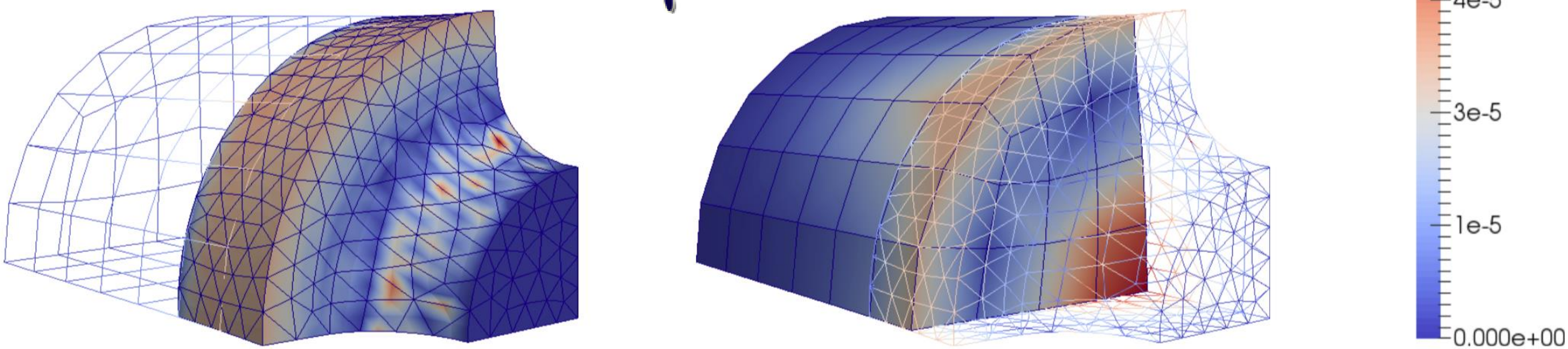




Notched Cylinder: TET - HEX Coupling

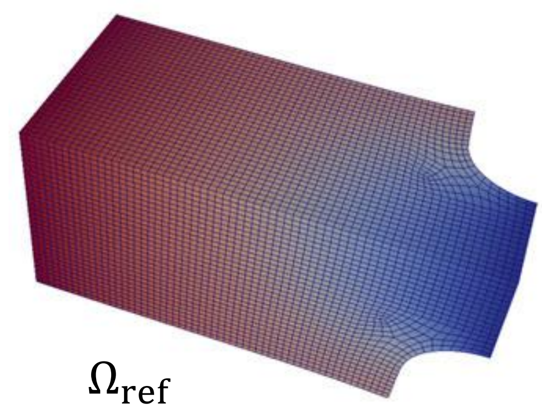


Notched Cylinder: TET - HEX Coupling



(a) Ω_1 **(b)** Ω_2

Absolute residual tolerance	u_3 relative error Ω_1	Ω_2
1.0×10^{-14}	9.27×10^{-3}	3.70×10^{-3}



Ω_{ref}

Notched Cylinder: Coupling Different Materials

The Schwarz method is capable of coupling regions with ***different material models***.

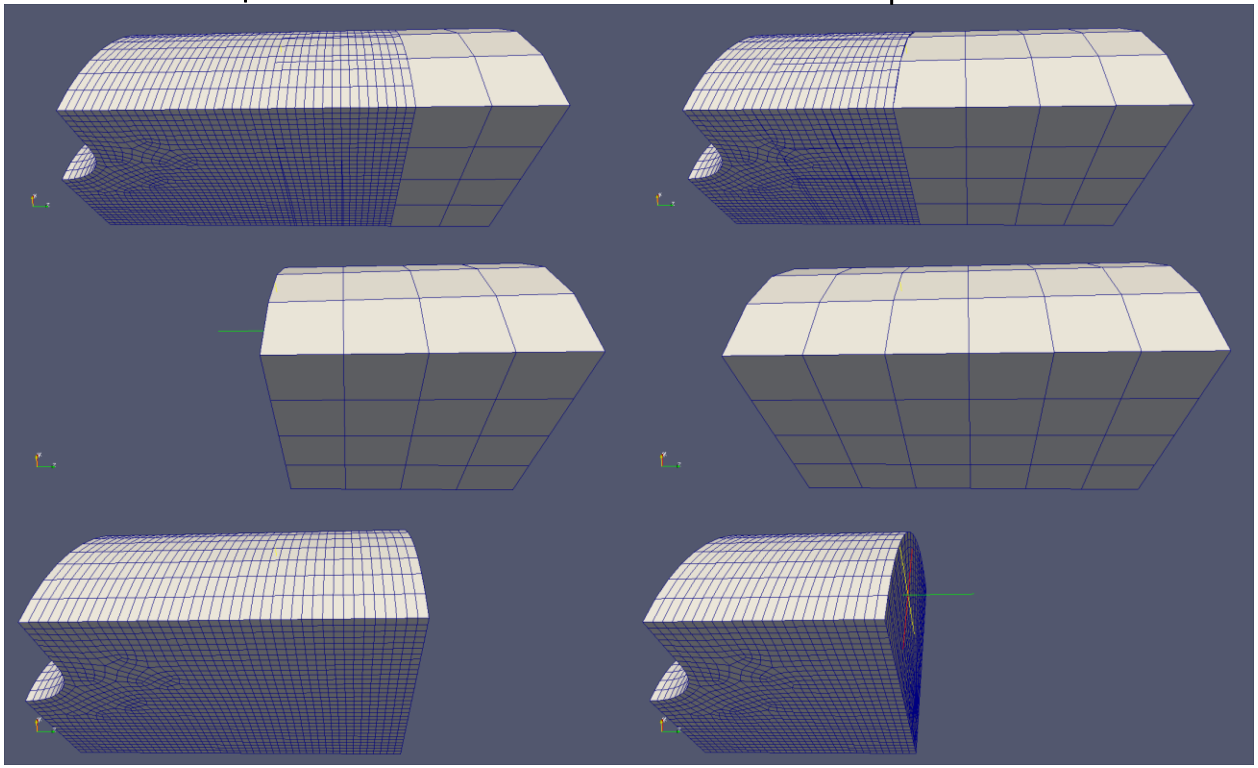
- Notched cylinder subjected to tensile load with an ***elastic*** and ***J2 elasto-plastic*** regions.
- ***Coarse*** region is ***elastic*** and ***fine*** region is ***elasto-plastic***.
- The ***overlap region*** in the first mesh is nearer the notch, where plastic behavior is expected.

Overlap far from notch. Overlap near notch.

Coupled regions

Coarse, elastic region

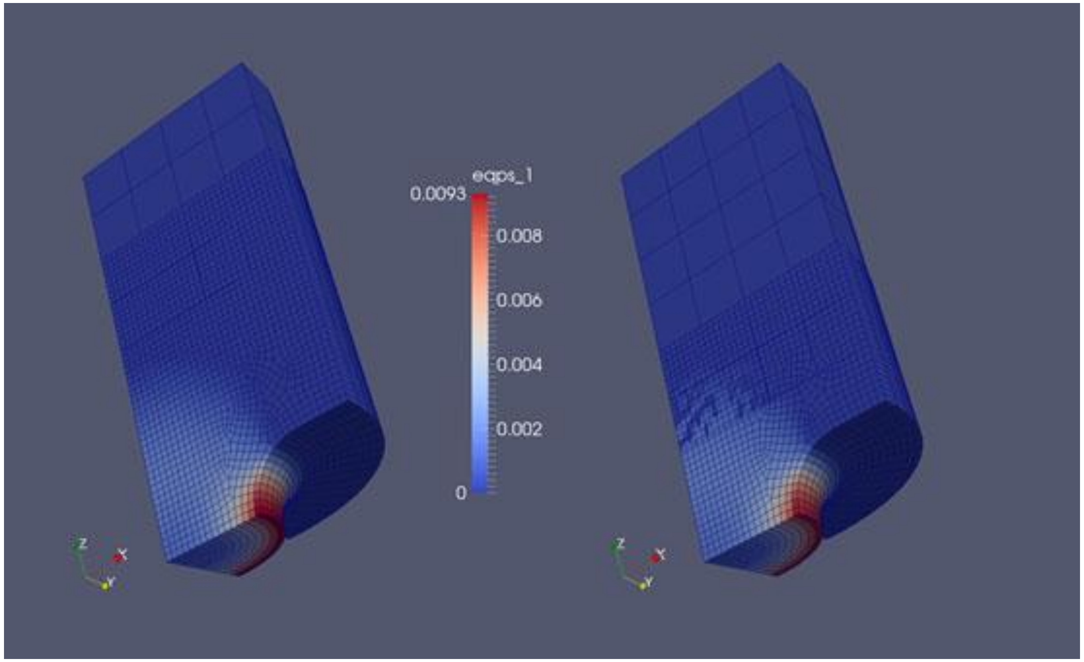
Fine, elasto-plastic region



Notched Cylinder: Coupling Different Materials

Need to be careful to do domain decomposition so that material models are ***consistent*** in overlap region.

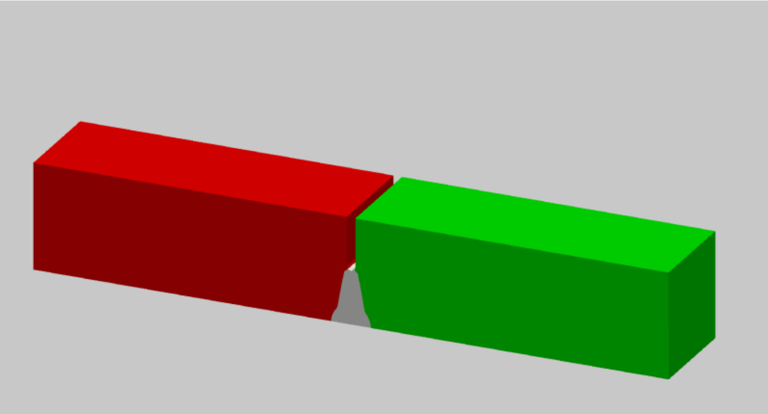
- When the ***overlap*** region is ***far from the notch***, no plastic deformation exists in it: the coarse and fine regions predict the ***same behavior***.
- When the ***overlap*** region is ***near the notch***, plastic deformation spills onto it and the two models predict different behavior, affecting convergence ***adversely***.



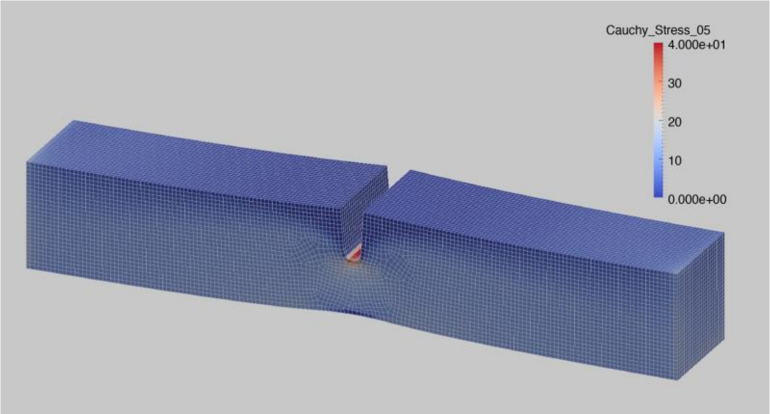
Overlap far from notch. Overlap near notch.

Quasistatic Example #3: Laser Weld

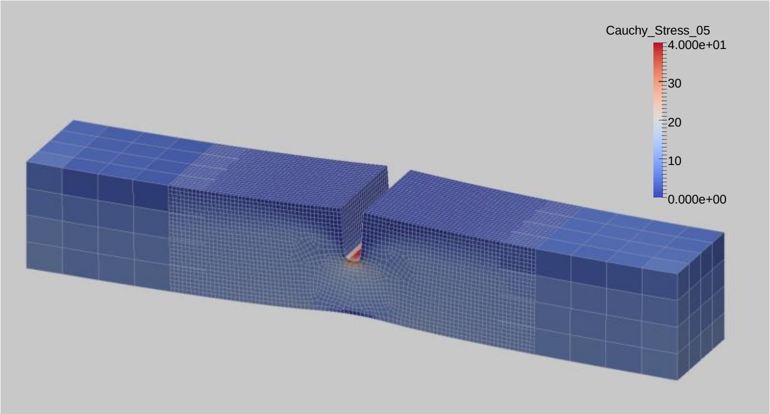
Laser weld specimen



Single domain discretization



Coupled Schwarz discretization
(50% reduction in model size)

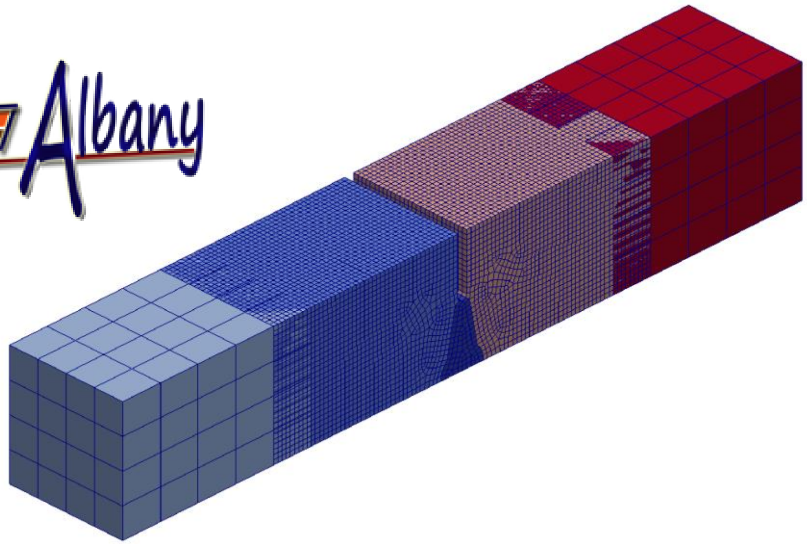


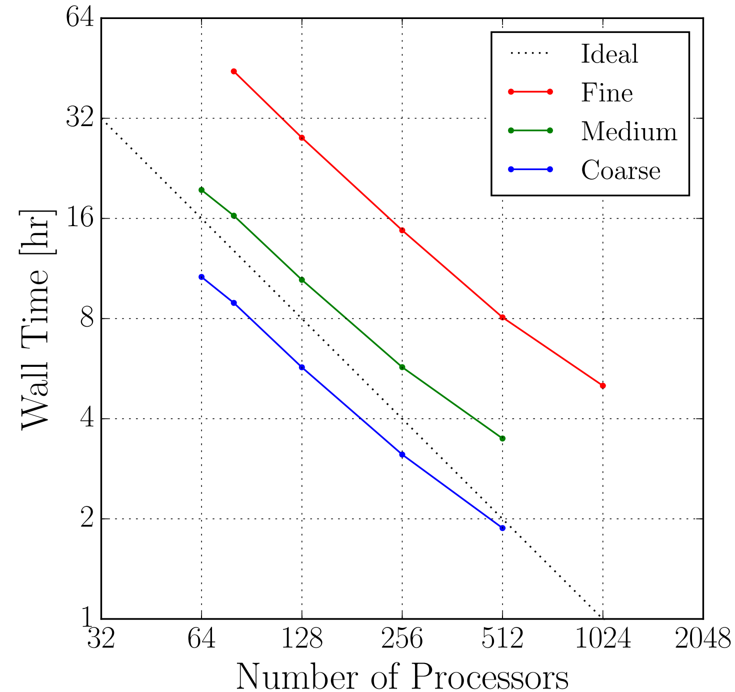
- Problem of ***practical scale.***
- ***Isotropic elasticity*** and ***J_2 plasticity*** with linear isotropic hardening.
- ***Identical parameters*** for weld and base materials for proof of concept, to become independent models.

Laser Weld: Strong Scalability of Parallel Schwarz with DTK

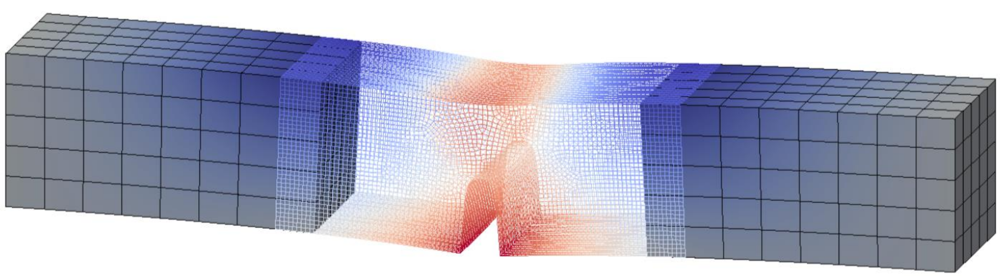
Sandia National Laboratories

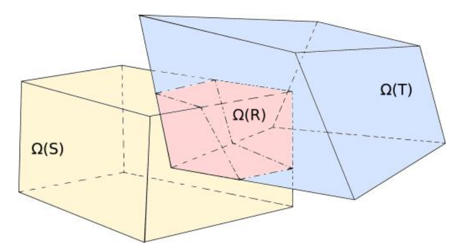
Albany





- ***Near-ideal linear speedup*** (64-1024 cores).



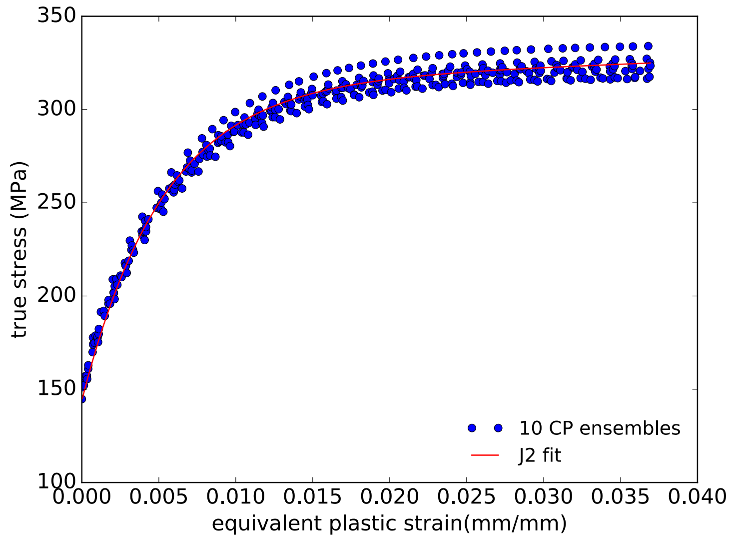


Data Transfer Kit (DTK)

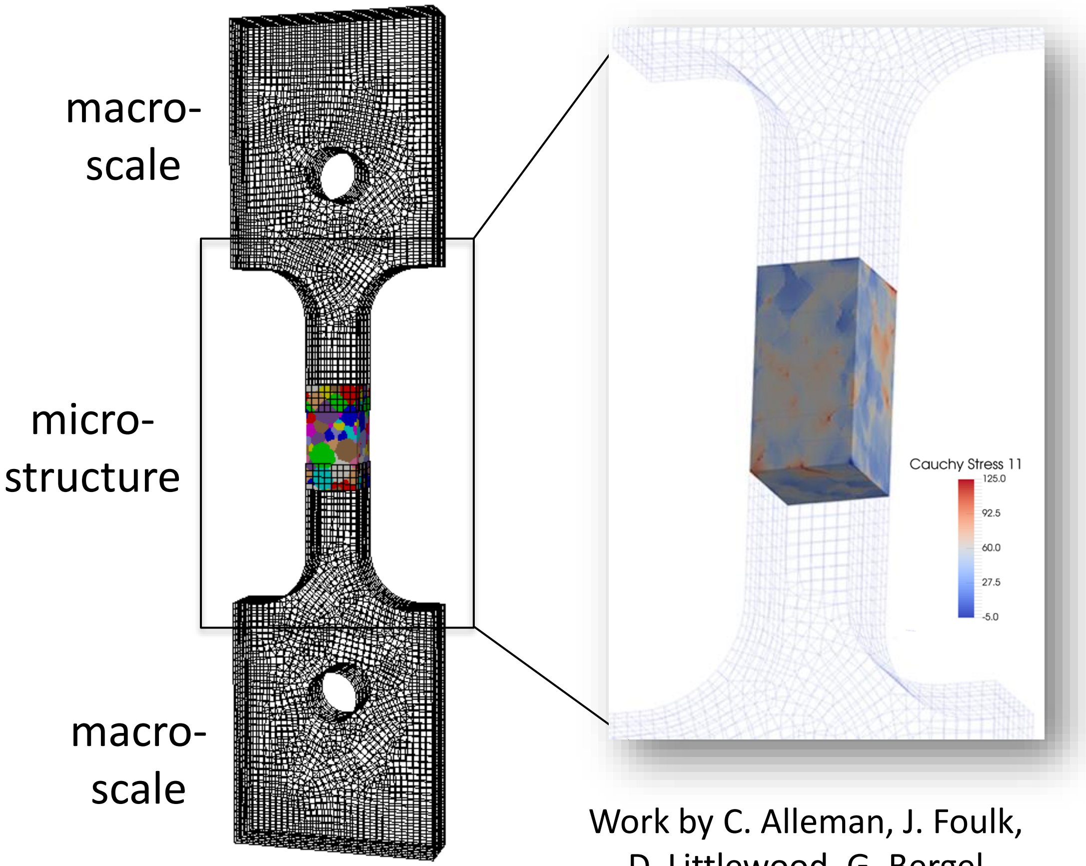
Quasistatic Example #4: Tensile Bar

The alternating Schwarz method can be used as part of a ***homogenization*** (upscaling) process to bridge gap b/w ***microscopic*** and ***macroscopic*** regions

- ***Microstructure*** embedded in ASTM tensile geometry (right).
- Fix microstructure, investigate ***ensemble*** of uniaxial loads.
- Fit flow curves with a ***macroscale*** J_2 plasticity model (below).



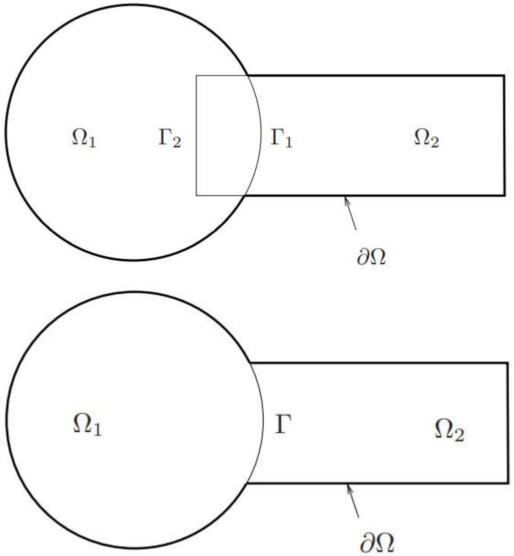
Goal: *study strain localization in microstructure.*



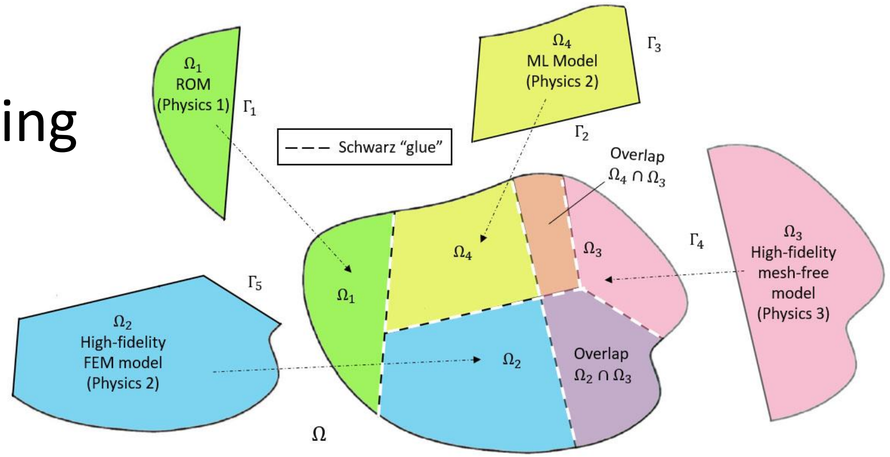
Work by C. Alleman, J. Foulk, D. Littlewood, G. Bergel

Outline

1. **Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics**
   - Motivation & Background
   - Quasistatics
   - **Extension to Dynamics**
   - Summary & Next Steps



2. Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling
   - Motivation & Background
   - Demonstration
   - Ongoing & Future Work



* Projection-based Reduced Order Model

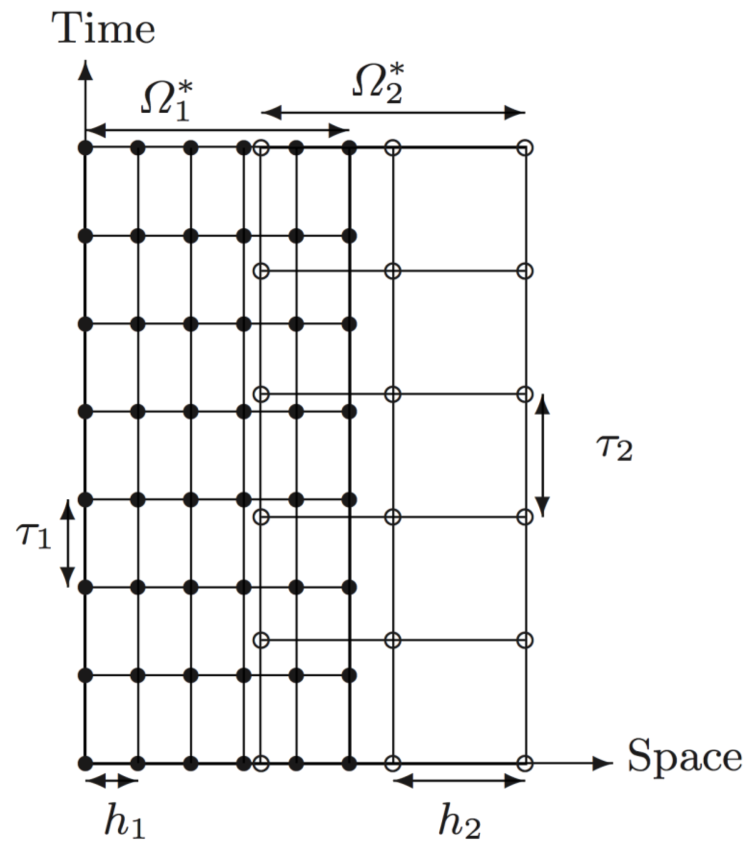
Dynamic Solid Mechanics Formulation

- Kinetic energy: $T(\dot{\boldsymbol{\varphi}}) \coloneqq \frac{1}{2}\int_{\Omega} \rho\dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\varphi}}\, dV$
- Potential Energy: $V(\boldsymbol{\varphi}) \coloneqq \int_{\Omega} A(\boldsymbol{F}, \boldsymbol{Z}) dV - \int_{\Omega} \rho\boldsymbol{B} \cdot \boldsymbol{\varphi}\, dV$
- Lagrangian: $L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) \coloneqq T(\dot{\boldsymbol{\varphi}}) - V(\boldsymbol{\varphi})$
- Action functional: $S[\boldsymbol{\varphi}] \coloneqq \int_{I} L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})\, dt$
- Euler-Lagrange equations: $\begin{cases} \mathrm{Div}\, \boldsymbol{P} + \rho\boldsymbol{B} = \rho\ddot{\boldsymbol{\varphi}}, & \text{in } \Omega \times I \\ \boldsymbol{\varphi}(\boldsymbol{X}, t_0) = \boldsymbol{x}_0, & \text{in } \Omega \\ \dot{\boldsymbol{\varphi}}(\boldsymbol{X}, t_0) = \boldsymbol{v}_0, & \text{in } \Omega \\ \boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{\chi}, & \text{on } \partial\Omega \times I \end{cases}$
- Semi-discrete problem following FEM discretization in space:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{f}_{\mathrm{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\mathrm{ext}}$$

Schwarz Alternating Method for Dynamics

- In the literature the Schwarz method is applied to dynamics by using ***space-time discretizations***.



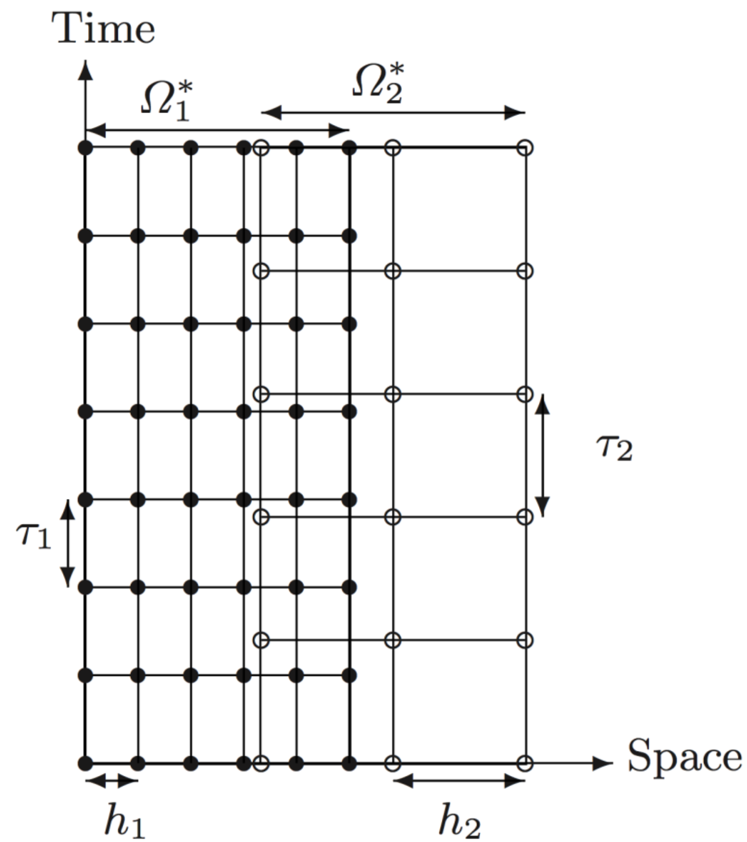
Overlapping non-matching meshes and time steps in dynamics.

Schwarz Alternating Method for Dynamics

- In the literature the Schwarz method is applied to dynamics by using ***space-time discretizations***.

Pro ☺: Can use ***non-matching*** meshes and time-steps (see right figure).

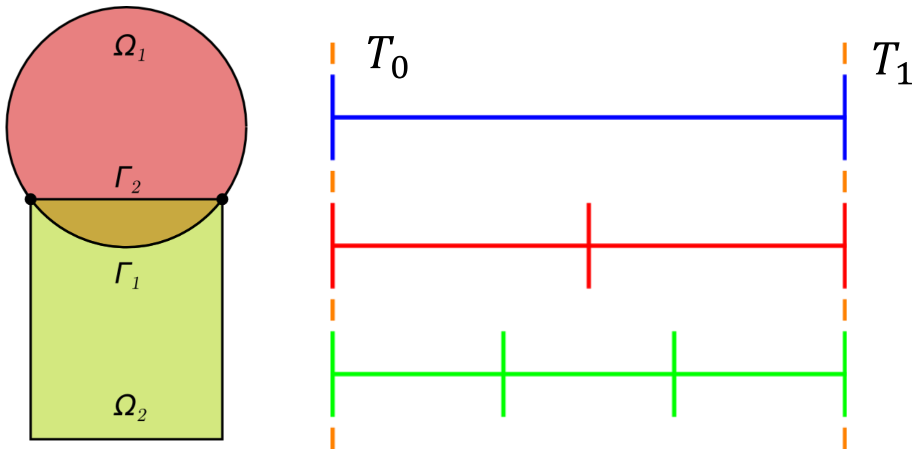
Con ☹: ***Unfeasible*** given the design of our current codes and size of simulations.



Overlapping non-matching meshes and time steps in dynamics.

Schwarz Alternating Method for Dynamic Multiscale Coupling

Controller time stepper: *defines global ΔTs at which subdomains are synchronized*



Controller time stepper

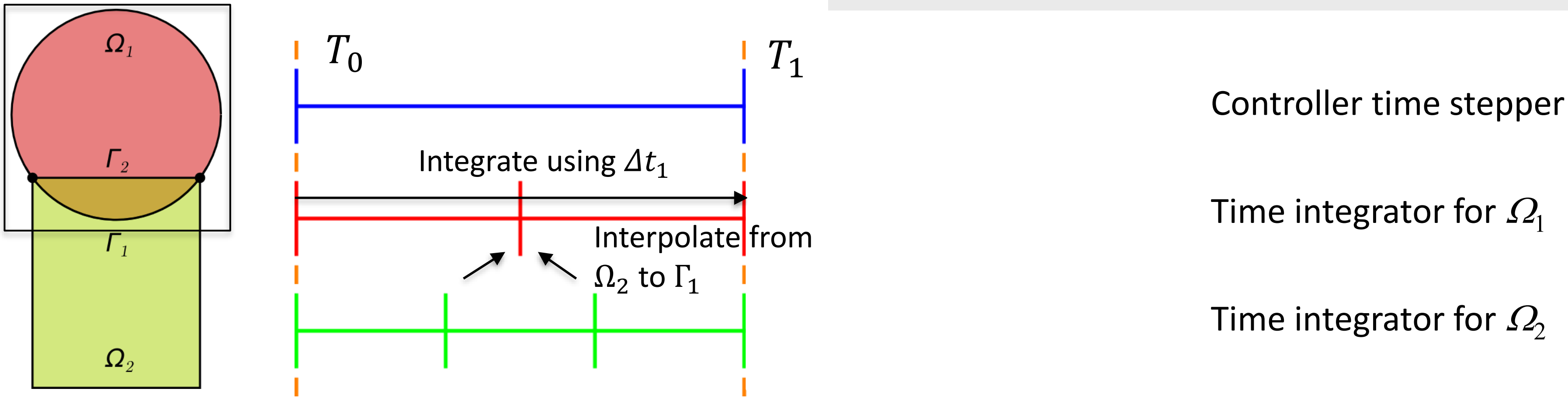
Time integrator for Ω_1

Time integrator for Ω_2

Step 0: Initialize $i = 0$ (controller time index).

Schwarz Alternating Method for Dynamic Multiscale Coupling

***Controller time stepper:** defines global ΔTs at which subdomains are synchronized*

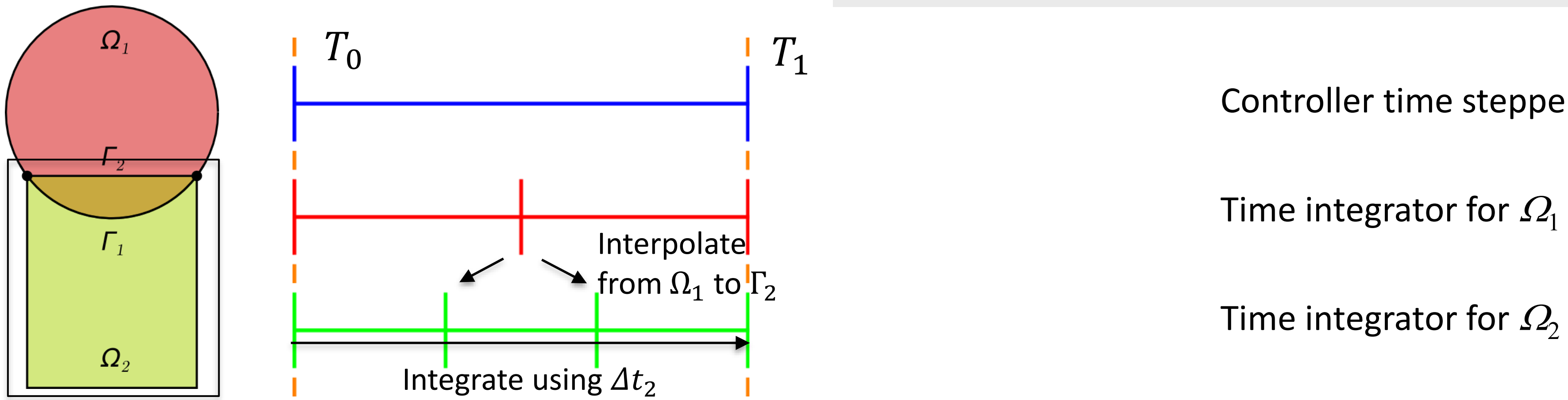


Step 0: Initialize $i = 0$ (controller time index).

Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

Schwarz Alternating Method for Dynamic Multiscale Coupling

***Controller time stepper:** defines global ΔTs at which subdomains are synchronized*



Step 0: Initialize $i = 0$ (controller time index).

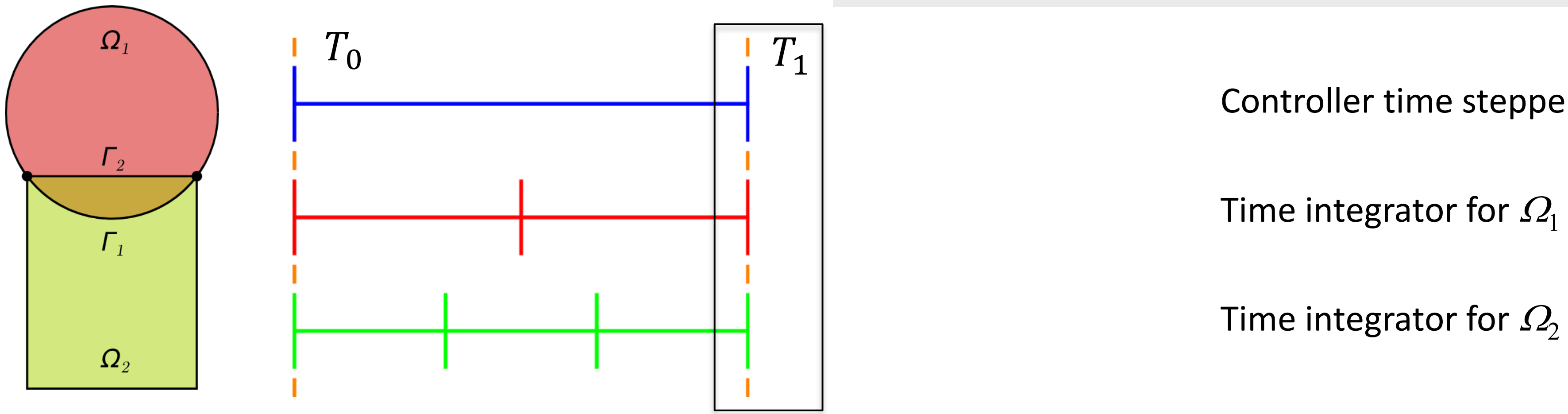
Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

Step 2: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2, using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

Schwarz Alternating Method for Dynamic Multiscale Coupling

Sandia National Laboratories

***Controller time stepper:** defines global ΔTs at which subdomains are synchronized*



Step 0: Initialize $i = 0$ (controller time index).

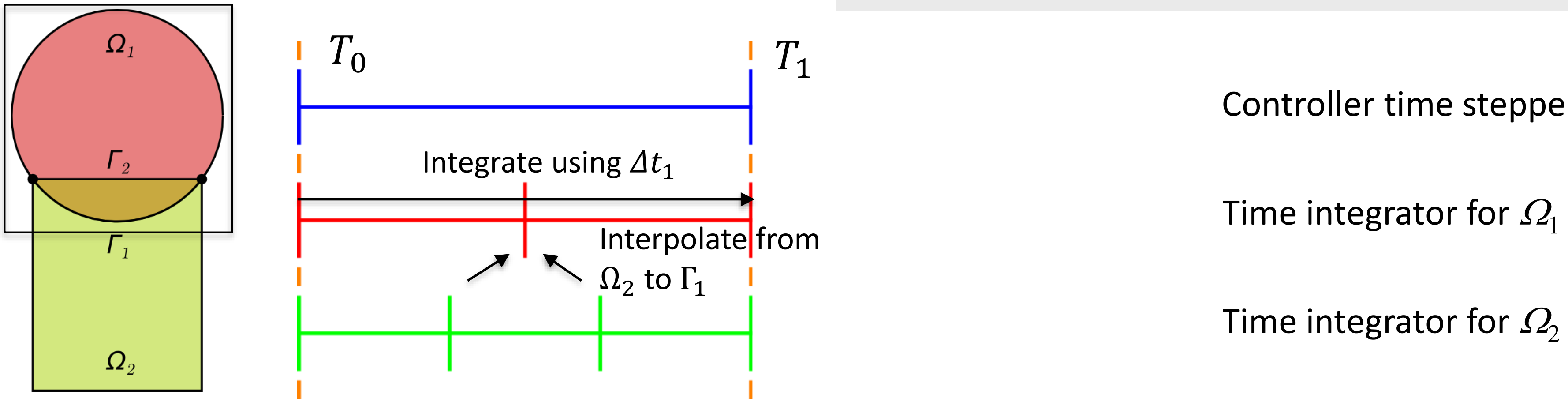
Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

Step 2: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2, using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

Step 3: Check for convergence at time T_{i+1}.

Schwarz Alternating Method for Dynamic Multiscale Coupling

Controller time stepper: *defines global ΔTs at which subdomains are synchronized*



<u>Step 0</u>: Initialize $i = 0$ (controller time index).

<u>Step 1</u>: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

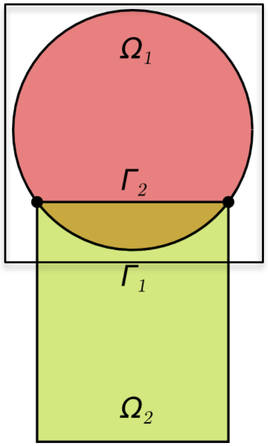
<u>Step 2</u>: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2, using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

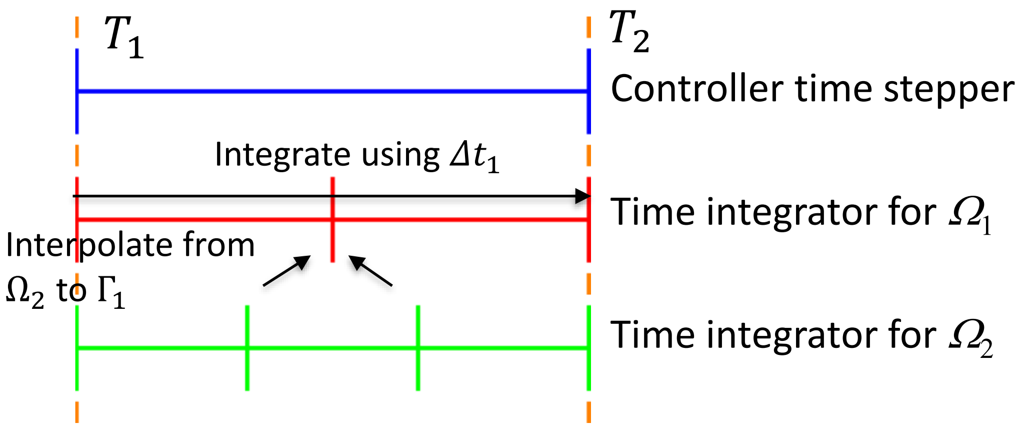
<u>Step 3</u>: Check for convergence at time T_{i+1}.

- If unconverged, return to Step 1.

Schwarz Alternating Method for Dynamic Multiscale Coupling

Controller time stepper: *defines global ΔTs at which subdomains are synchronized*





Step 0: Initialize $i = 0$ (controller time index).

Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

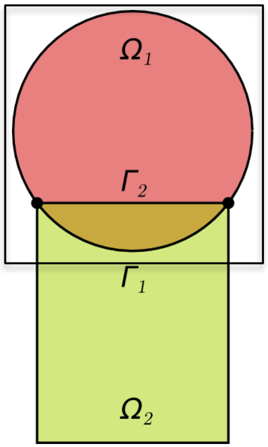
Step 2: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2, using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

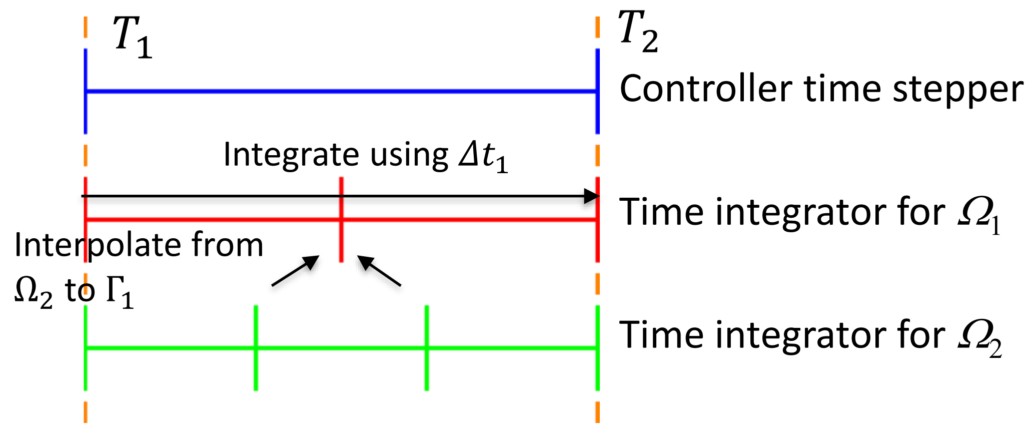
Step 3: Check for convergence at time T_{i+1}.

- If unconverged, return to Step 1.
- If converged, set $i = i + 1$ and return to Step 1.

Schwarz Alternating Method for Dynamic Multiscale Coupling

***Controller time stepper:** defines global ΔTs at which subdomains are synchronized*





<u>Step 0</u>: Initialize $i = 0$ (controller time index).

<u>Step 1</u>: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

<u>Step 2</u>: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2, using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

<u>Step 3</u>: Check for convergence at time T_{i+1}.

- If unconverged, return to Step 1.
- If converged, set $i = i + 1$ and return to Step 1.

Can use ***different integrators*** with ***different time steps*** within each domain!

Schwarz Alternating Method for Dynamic Multiscale Coupling: Theory

- Like for quasistatics, dynamic alternating Schwarz method converges provided each single-domain problem is ***well-posed*** and ***overlap region*** is ***non-empty***, under some ***conditions*** on Δt.
- ***Well-posedness*** for the dynamic problem requires that action functional $S[\boldsymbol{\varphi}] := \int_I \int_\Omega L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) dV dt$ be ***strictly convex*** or ***strictly concave***, where $L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) := T(\dot{\boldsymbol{\varphi}}) + V(\boldsymbol{\varphi})$ is the Lagrangian.
  - ➢ This is studied by looking at its second variation $\delta^2 S[\boldsymbol{\varphi}_h]$
- We can show assuming a ***Newmark*** time-integration scheme that for the ***fully-discrete*** problem:

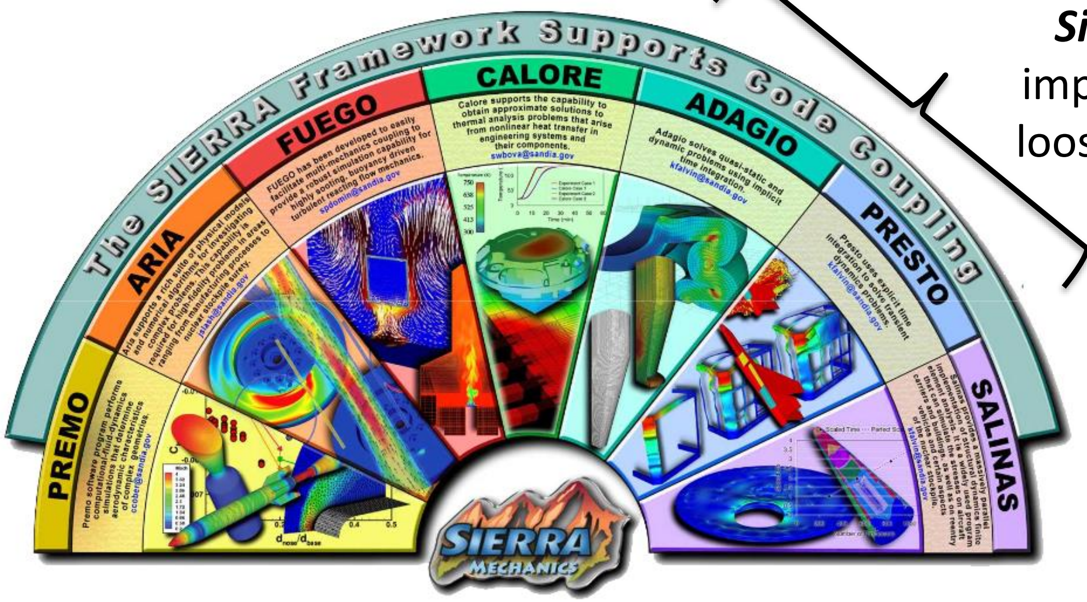
$$\delta^2 S[\boldsymbol{\varphi}_h] = \boldsymbol{x}^T \left[\frac{\gamma^2}{(\beta \Delta t)^2} \boldsymbol{M} - \boldsymbol{K} \right] \boldsymbol{x}$$

  - ➢ $\delta^2 S[\boldsymbol{\varphi}_h]$ can always be made positive by choosing a ***sufficiently small*** Δt
  - ➢ Numerical experiments reveal that Δt requirements for ***stability/accuracy*** typically lead to automatic satisfaction of this bound.

Implementation in *Albany LCM* & *Sierra/SM*

Numerical results shown here for ***dynamic*** Schwarz are from two codes:
Albany LCM and ***Sierra/Solid Mechanics*** (***Sierra/SM***)

- ***Albany LCM:*** Trilinos-based open-source* parallel, C++, multi-physics, FE code.
- ***Sierra Mechanics Framework****:* Sandia Lagrangian 3D code for FEA of solids & structures.

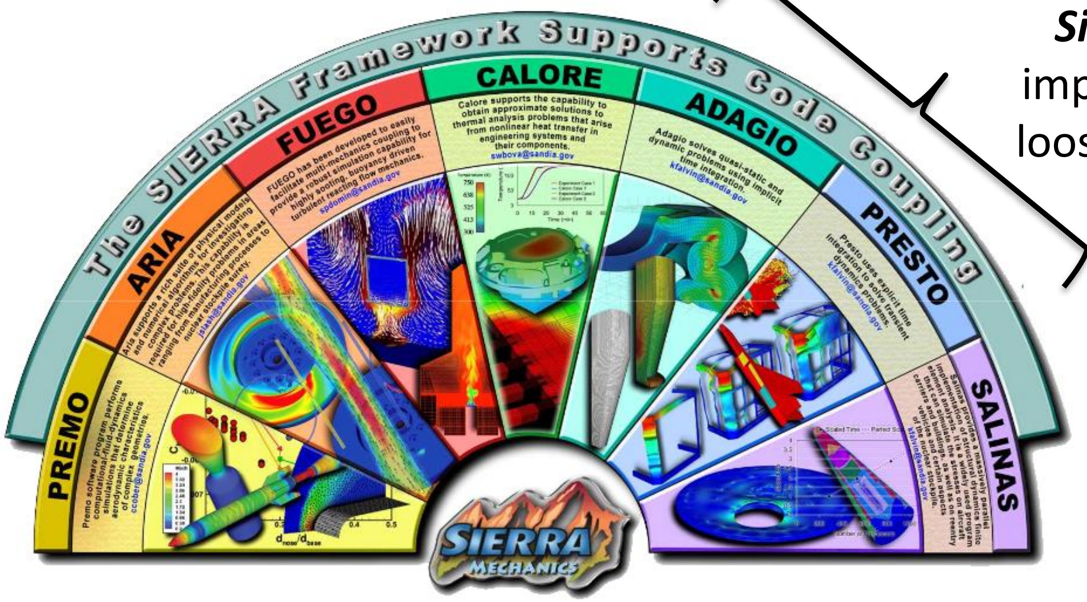


Sierra/SM: quasistatics, implicit/explicit dynamics + loose coupling via ***Arpeggio***

Implementation in *Albany LCM & Sierra/SM*

Numerical results shown here for ***dynamic*** Schwarz are from two codes:
Albany LCM and ***Sierra/Solid Mechanics*** (***Sierra/SM***)

- ***Albany LCM:*** Trilinos-based open-source* parallel, C++, multi-physics, FE code.
- ***Sierra Mechanics Framework**:* Sandia Lagrangian 3D code for FEA of solids & structures.



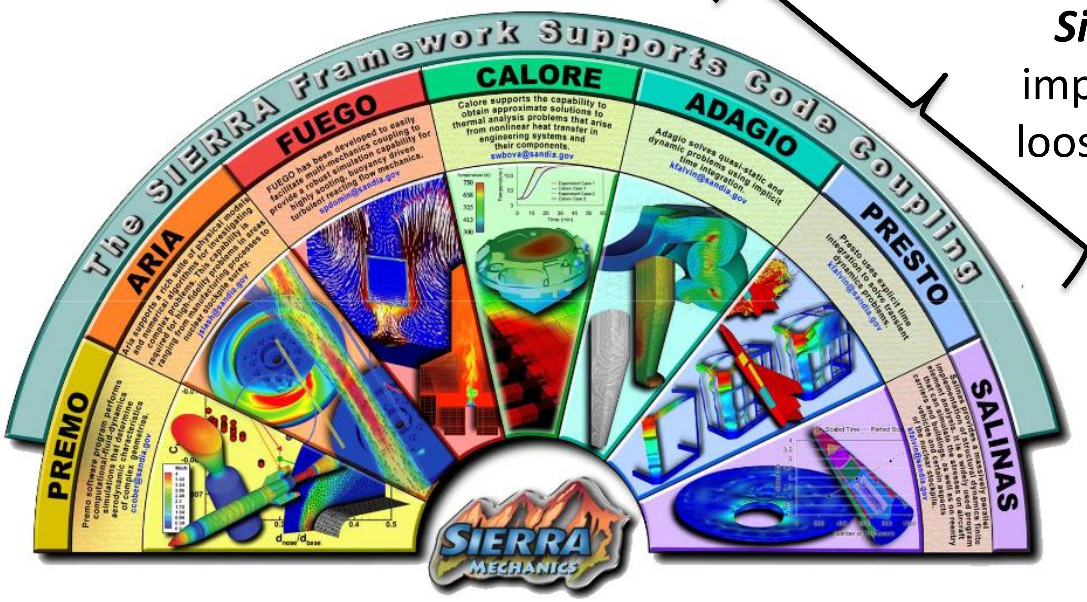
Sierra/SM: quasistatics, implicit/explicit dynamics + loose coupling via ***Arpeggio***

Schwarz alternating method was **"implemented"** in ***Sierra/SM*** using ***Arpeggio*** loose coupling framework

Implementation in *Albany LCM* & *Sierra/SM*

Numerical results shown here for ***dynamic*** Schwarz are from two codes: ***Albany LCM*** and ***Sierra/Solid Mechanics*** **(*Sierra/SM*)**

- ***Albany LCM:*** Trilinos-based open-source* parallel, C++, multi-physics, FE code.
- ***Sierra Mechanics Framework****:* Sandia Lagrangian 3D code for FEA of solids & structures.



Sierra/SM: quasistatics, implicit/explicit dynamics + loose coupling via ***Arpeggio***

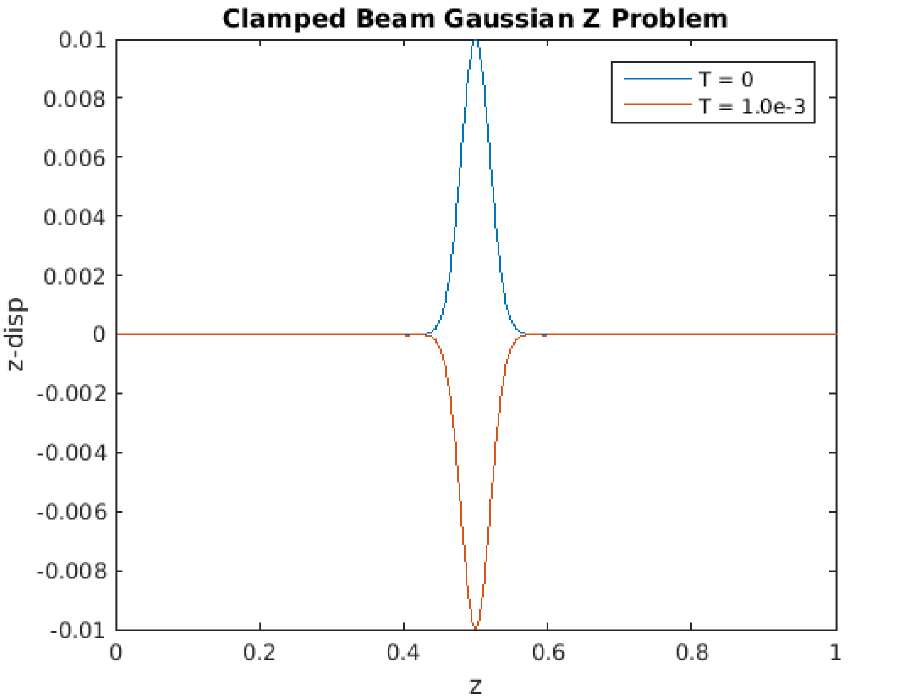
Schwarz alternating method was **"implemented"** in ***Sierra/SM*** using ***Arpeggio*** loose coupling framework

We did not have to write any code in Sierra/SM to implement Schwarz!

* https://github.com/SNLComputation/LCM

Dynamic Example #1: Elastic Wave Propagation

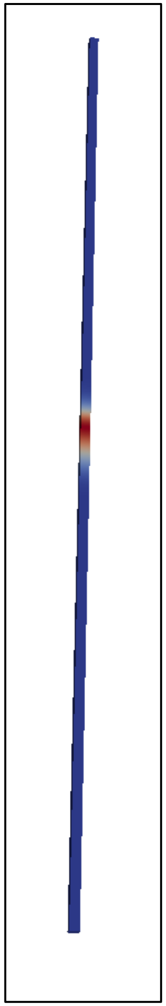
- Linear elastic ***clamped beam*** with Gaussian initial condition for the z-displacement.
- Simple problem with analytical exact solution but very ***stringent test*** for discretization methods.
- Test Schwarz with ***2 subdomains***: $\Omega_0 = (0,0.001) \times (0,0.001) \times (0,0.75), \Omega_1 = (0,0.001) \times (0,0.001) \times (0.25,1)$.



Left: Initial condition (blue) and final solution (red). Wave profile is negative of initial profile at time T = 1.0e-3.

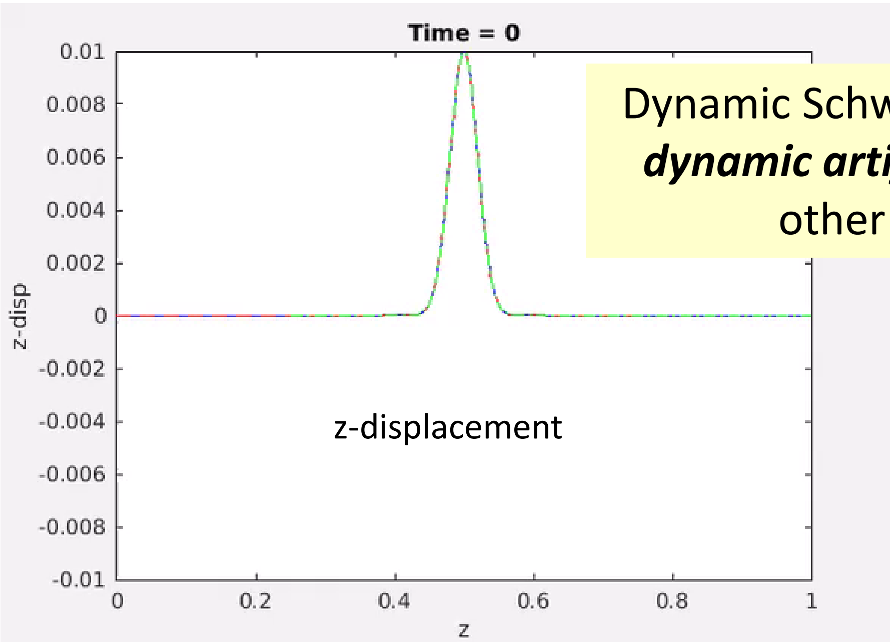
Time-discretizations: Newmark (implicit, explicit).

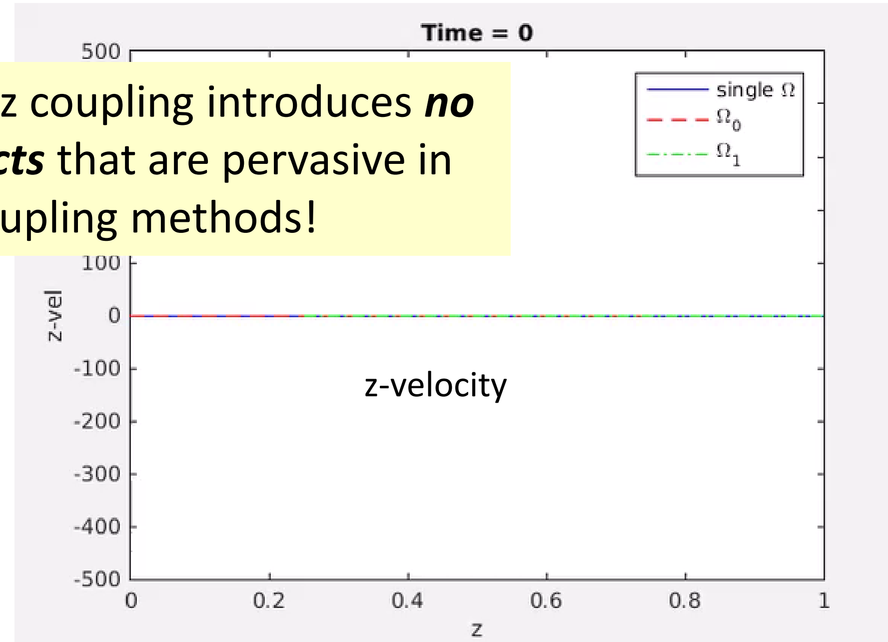
Meshes: HEX, TET



Elastic Wave: Diff. Integrators, Same Δts

Sandia National Laboratories





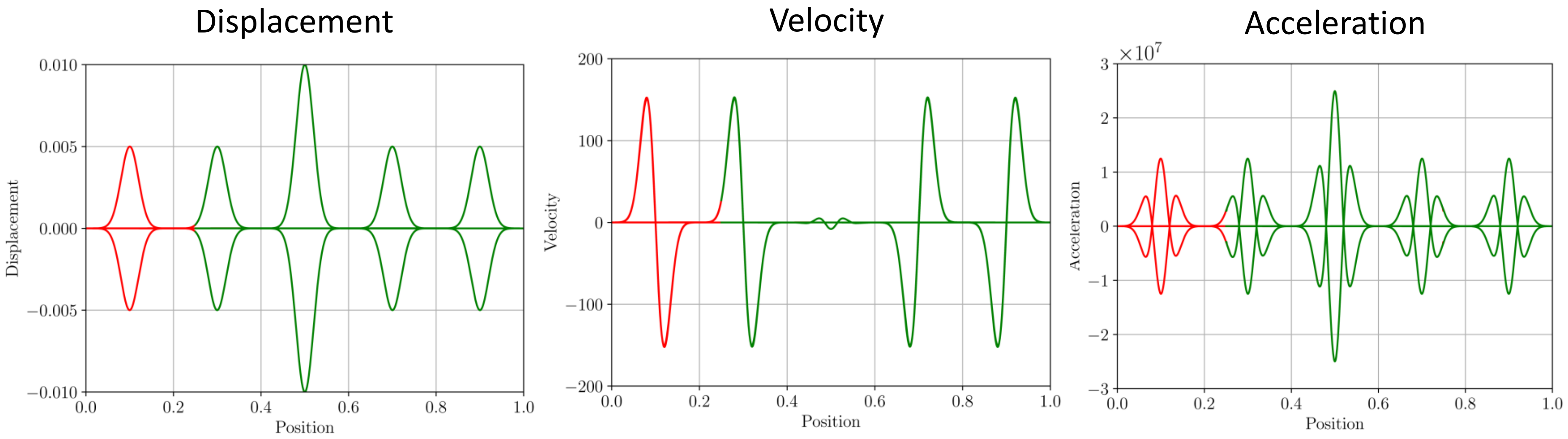
Dynamic Schwarz coupling introduces ***no dynamic artifacts*** that are pervasive in other coupling methods!

Table 1: Averaged (over times + domains) relative errors in **z–displacement** (blue) and **z-velocity** (green) for several different Schwarz couplings, 50% overlap volume fraction

	Implicit-Implicit		Explicit(CM)-Implicit		Explicit(LM)-Implicit	
Conformal HEX - HEX	2.79e-3	7.32e-3	3.53e-3	8.70e-3	4.72e-3	1.19e-2
Nonconformal HEX - HEX	2.90e-3	7.10e-3	2.82e-3	7.29e-3	2.84e-3	7.33e-3
TET - HEX	2.79e-3	7.58e-3	3.52e-3	8.92e-3	4.72e-3	1.19e-2

LM = Lumped Mass, CM = Consistent Mass

Elastic Wave: Diff. Integrators, Diff. Δts

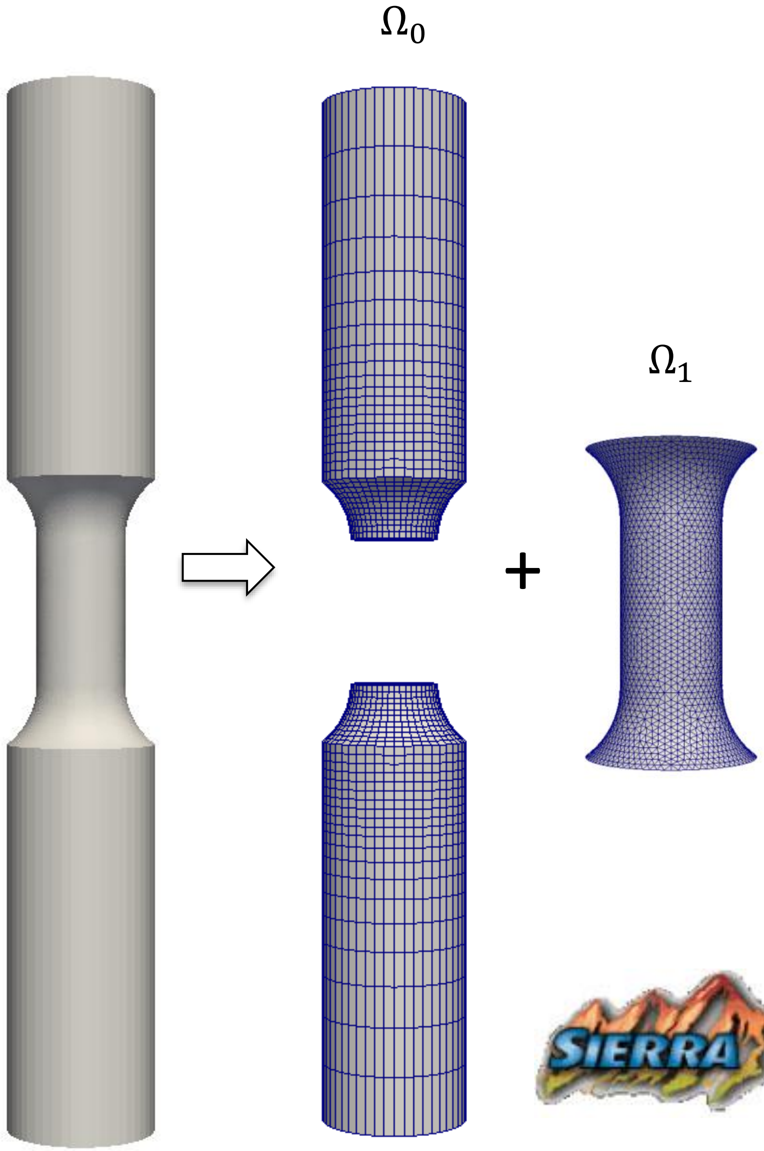


Figures above: Plots of displacement, velocity and acceleration for the elastic wave propagation problem using different time integrators (implicit and explicit) and different time steps (1e-2s and 2e-7s) for each subdomain, superimposed over the analytic single domain solution.

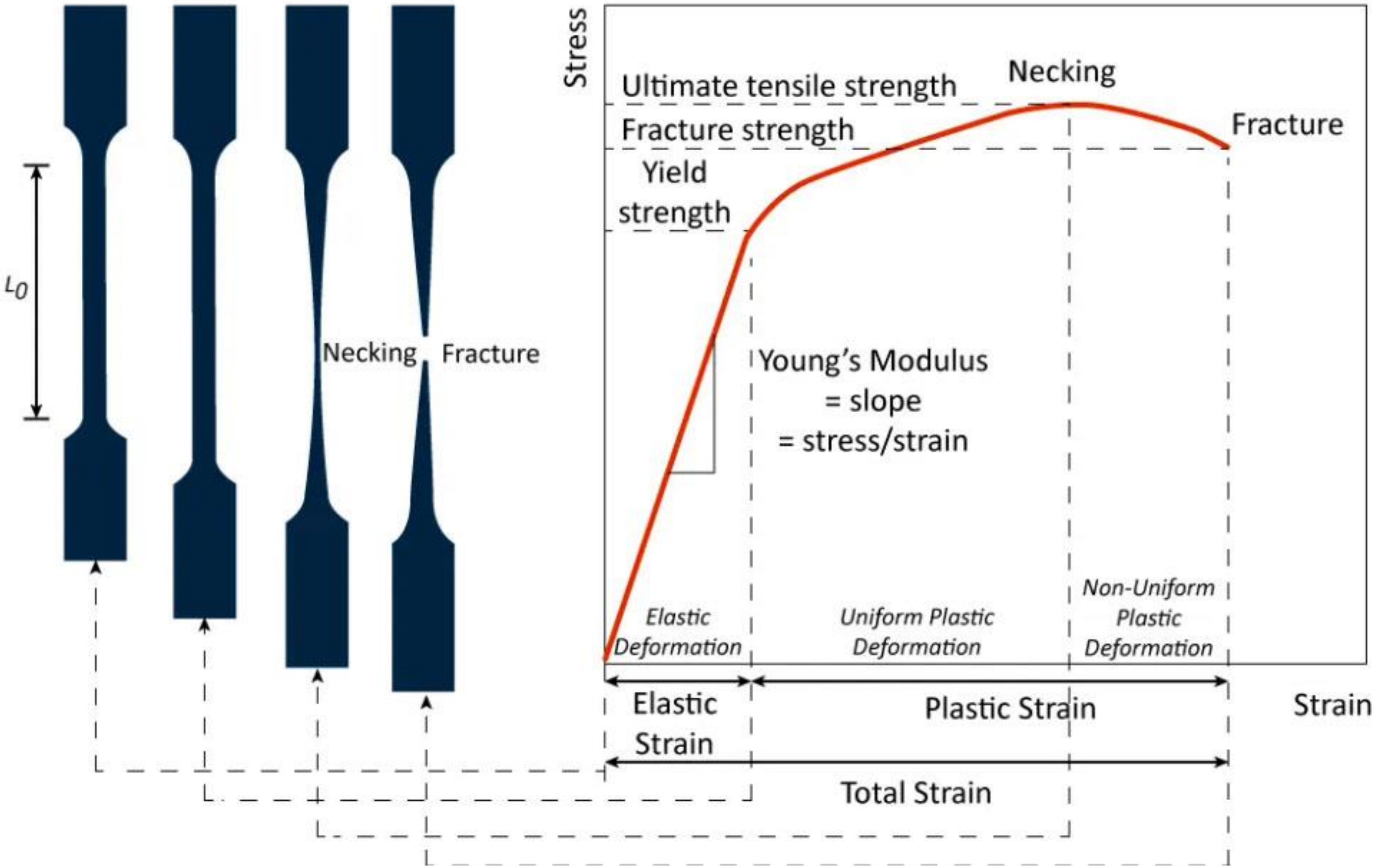
The analytic solution is ***indistinguishable*** from Schwarz solutions (hidden behind the solutions for Ω_0 (red) and Ω_1 (green))!

Dynamic Example #2: Tension Specimen

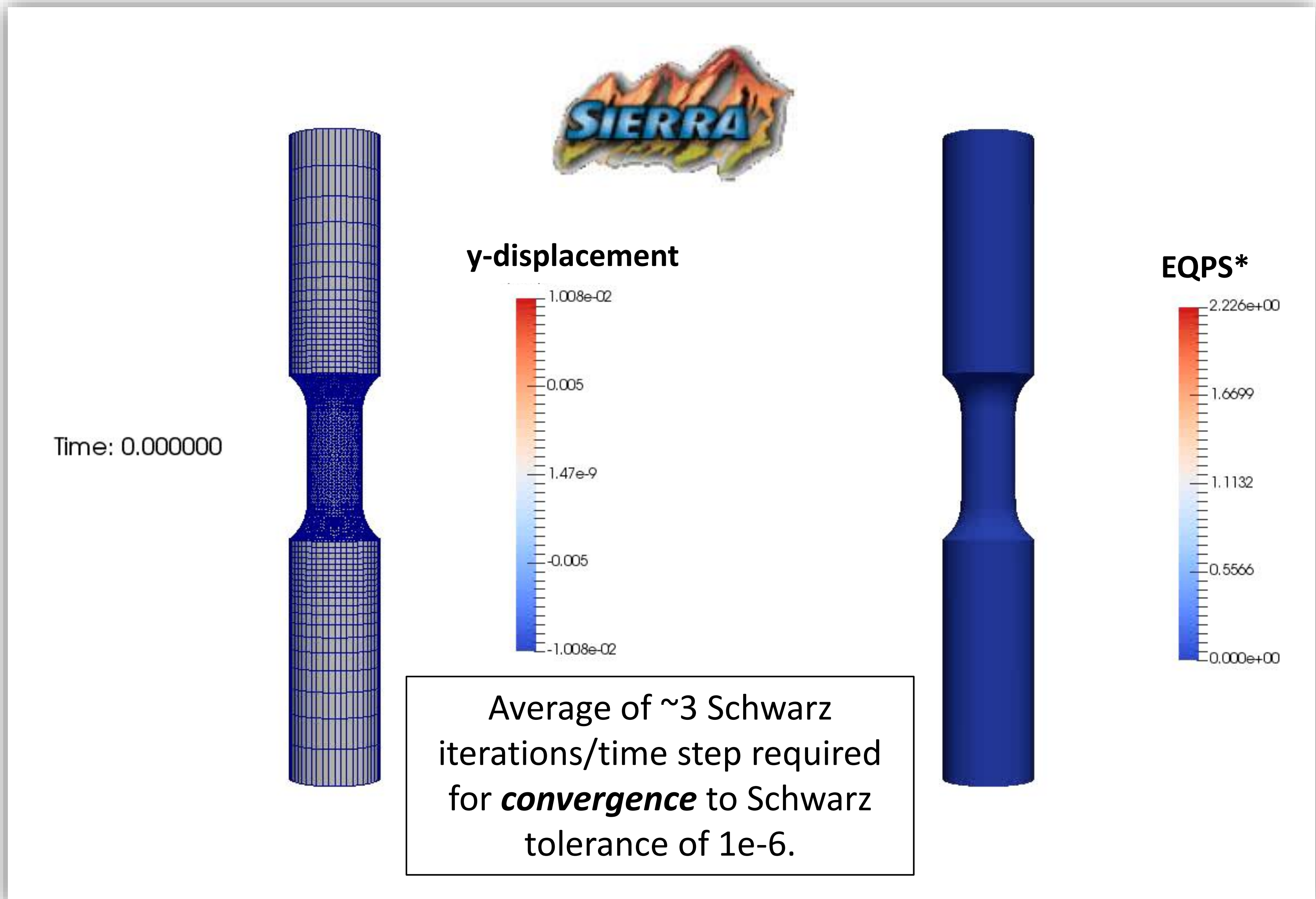
- Uniaxial aluminum cylindrical tensile specimen with ***inelastic J_2 material model***.
- Domain decomposition into ***two subdomains*** (right): Ω_0 = ends, Ω_1 = gauge.
- ***Nonconformal HEX + composite TET10*** coupling via Schwarz.
- ***Implicit*** Newmark time-integration with ***adaptive time-stepping*** algorithm employed in both subdomains.
- Slight ***imperfection*** introduced at center of gauge to force ***necking*** upon pulling in vertical direction.



Tension Specimen: Expected Result



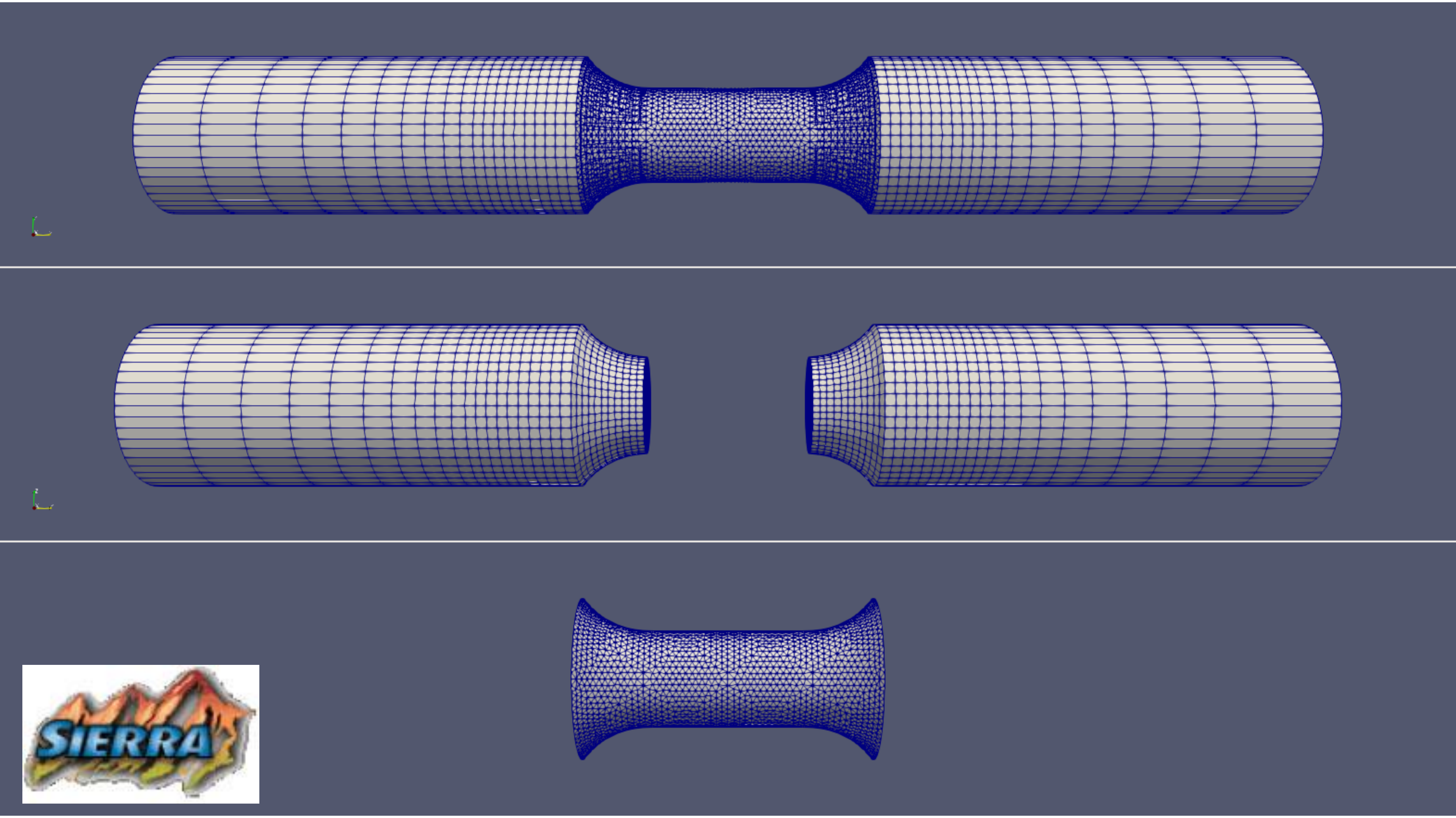
Tension Specimen: Disp. & EQPS



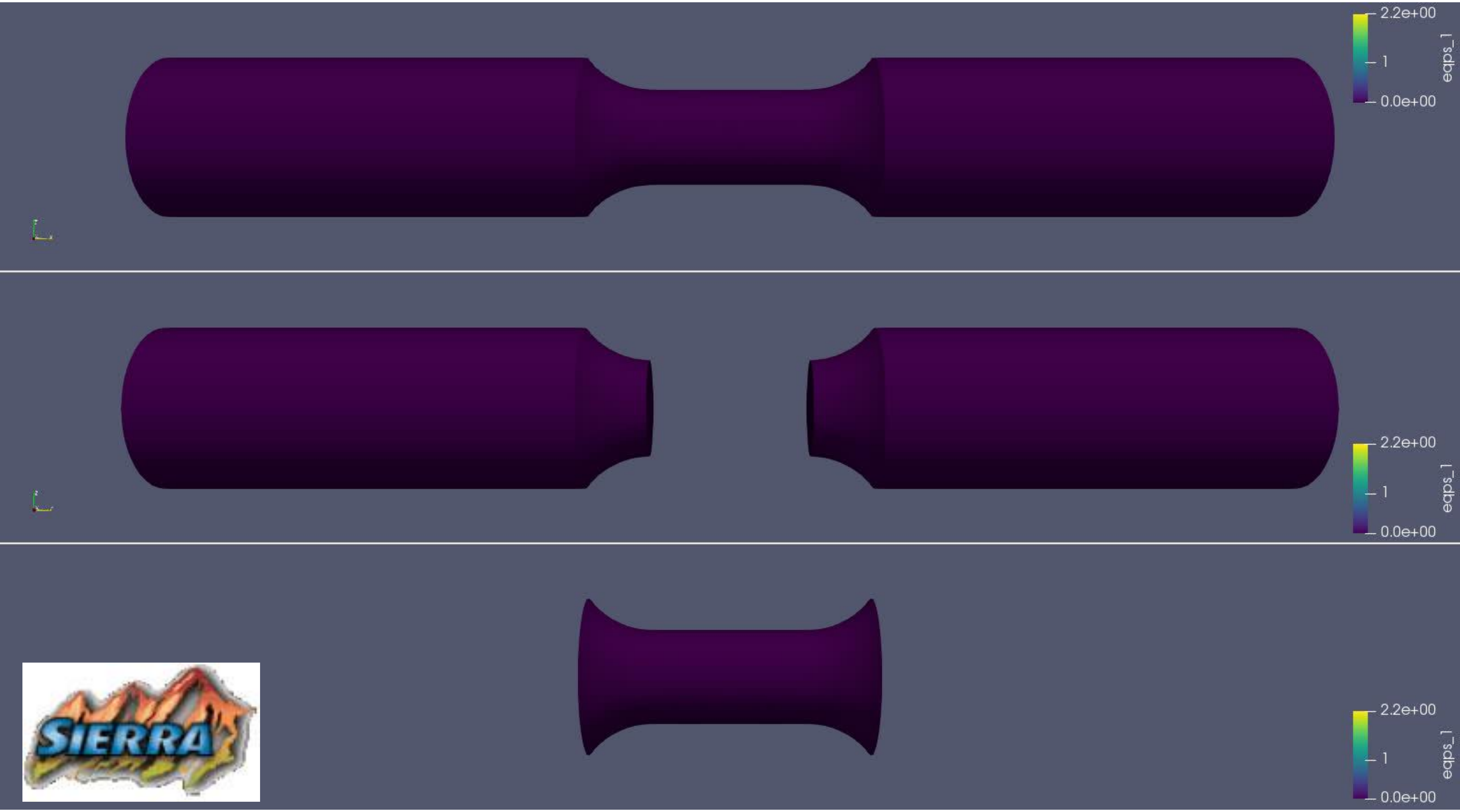
Average of ~3 Schwarz iterations/time step required for ***convergence*** to Schwarz tolerance of 1e-6.

*EQPS = Equivalent Plastic Strain

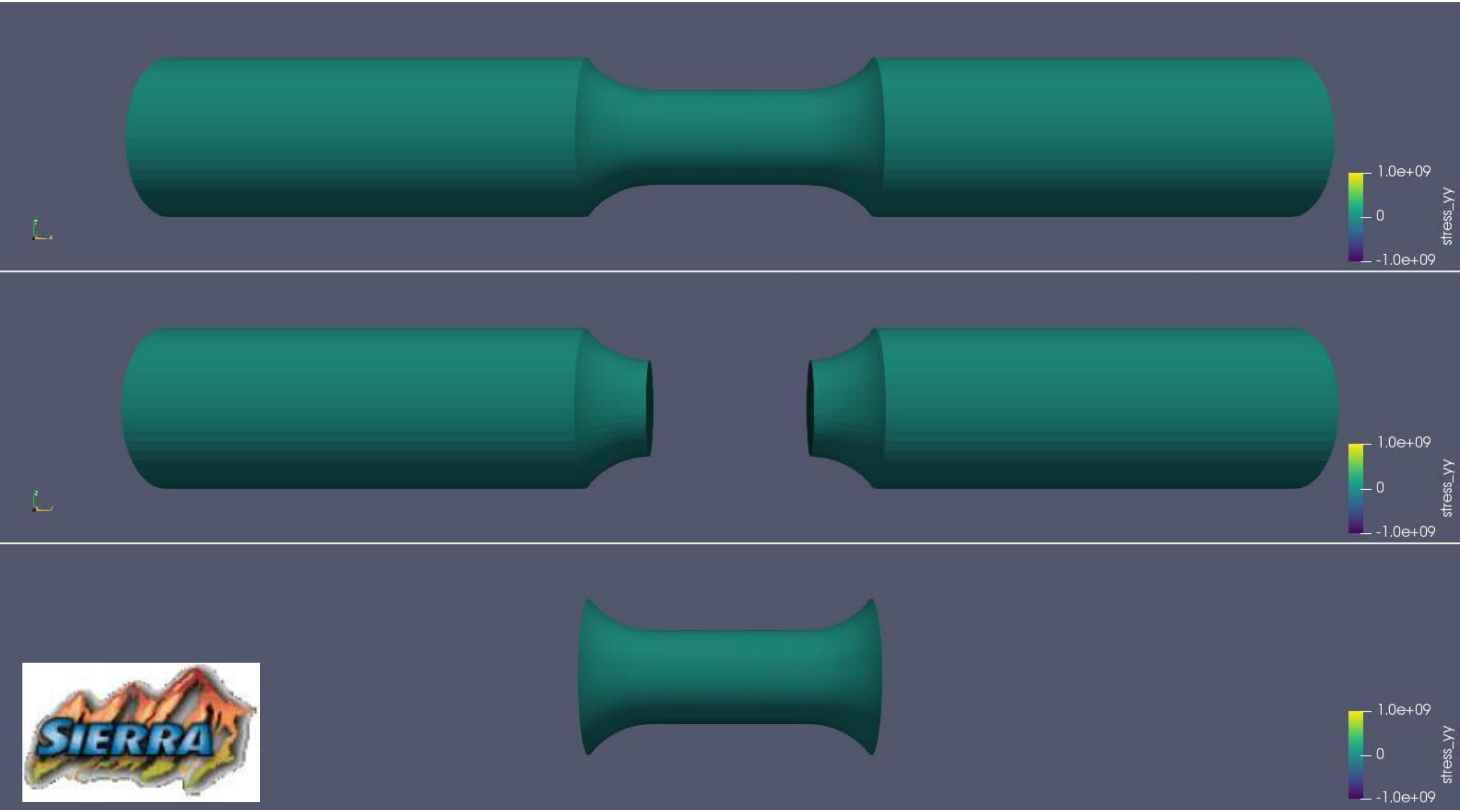
Tension Specimen: Domain Decomposition



Tension Specimen: EQPS



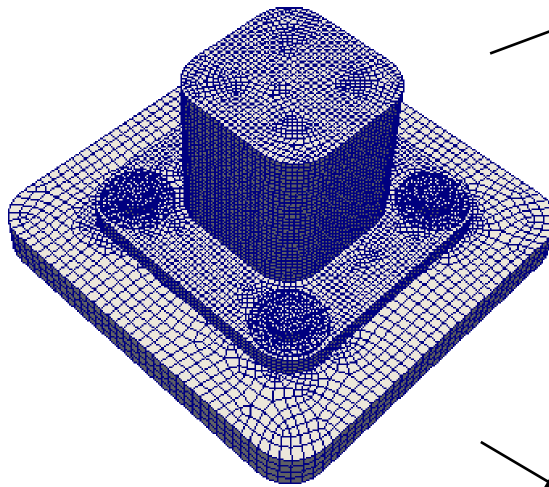
Tension Specimen: Cauchy Stress

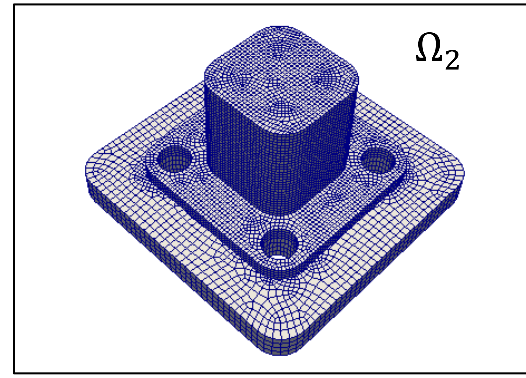


Dynamic Example #3: Bolted Joint Problem

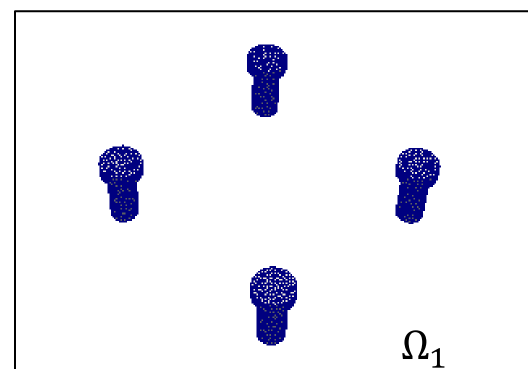
Problem of ***practical scale***.

- Schwarz solution compared to single-domain solution on composite TET10 mesh.



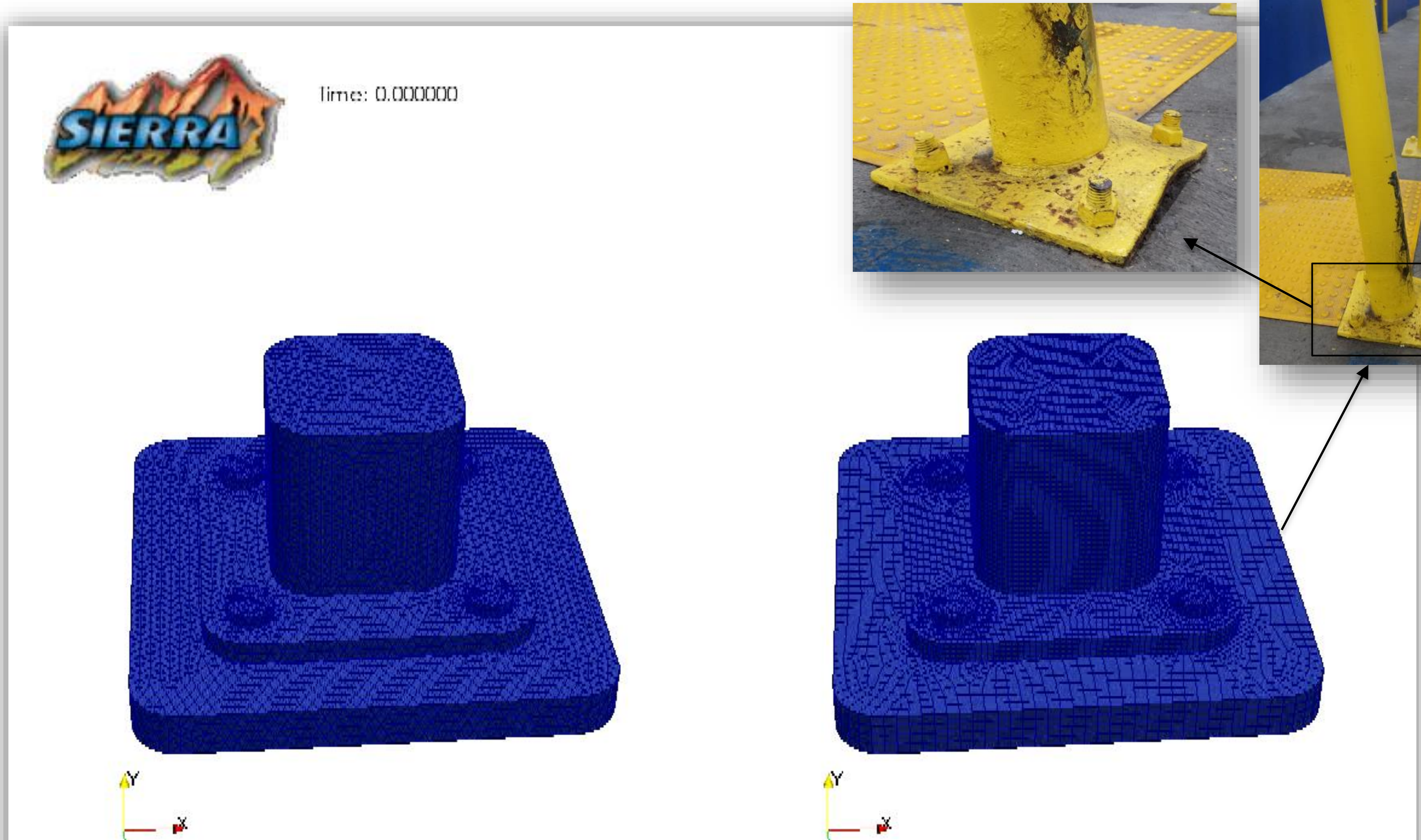


- BC: x-disp = 0.02 at T = 1.0e-3 on top of parts.
- Run until T = 5.0e-4 w/ dt = 1e-5 + implicit Newmark with analytic mass matrix for composite tet 10s.



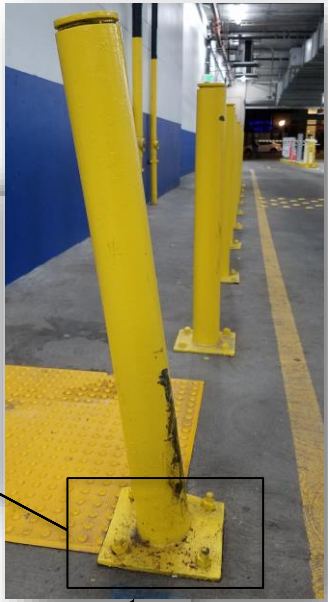
- Ω_1 = bolts (Composite TET10), Ω_2 = parts (HEX).
- ***Inelastic J_2 material model*** in both subdomains.
  - Ω_1: steel
  - Ω_2: steel component, aluminum (bottom) plate

Bolted Joint Problem: Displacement

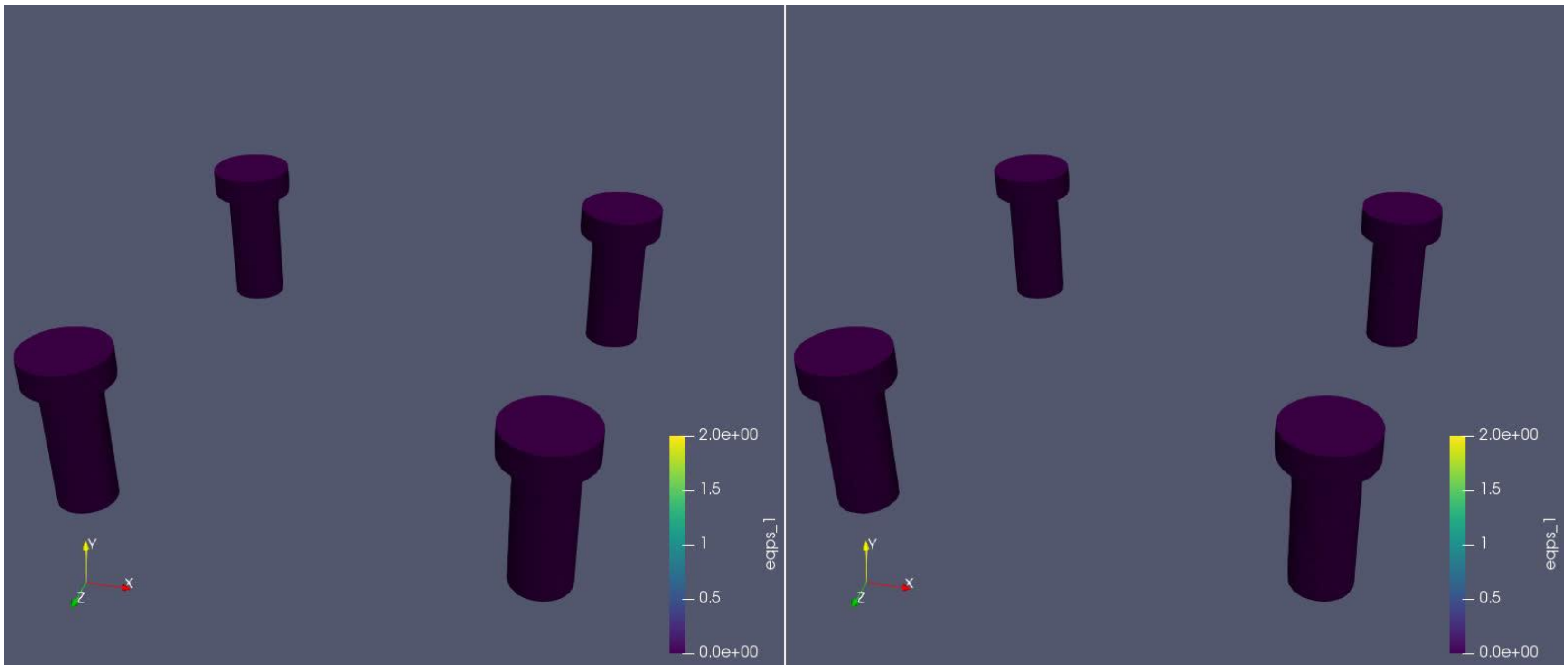


Single Ω

Schwarz



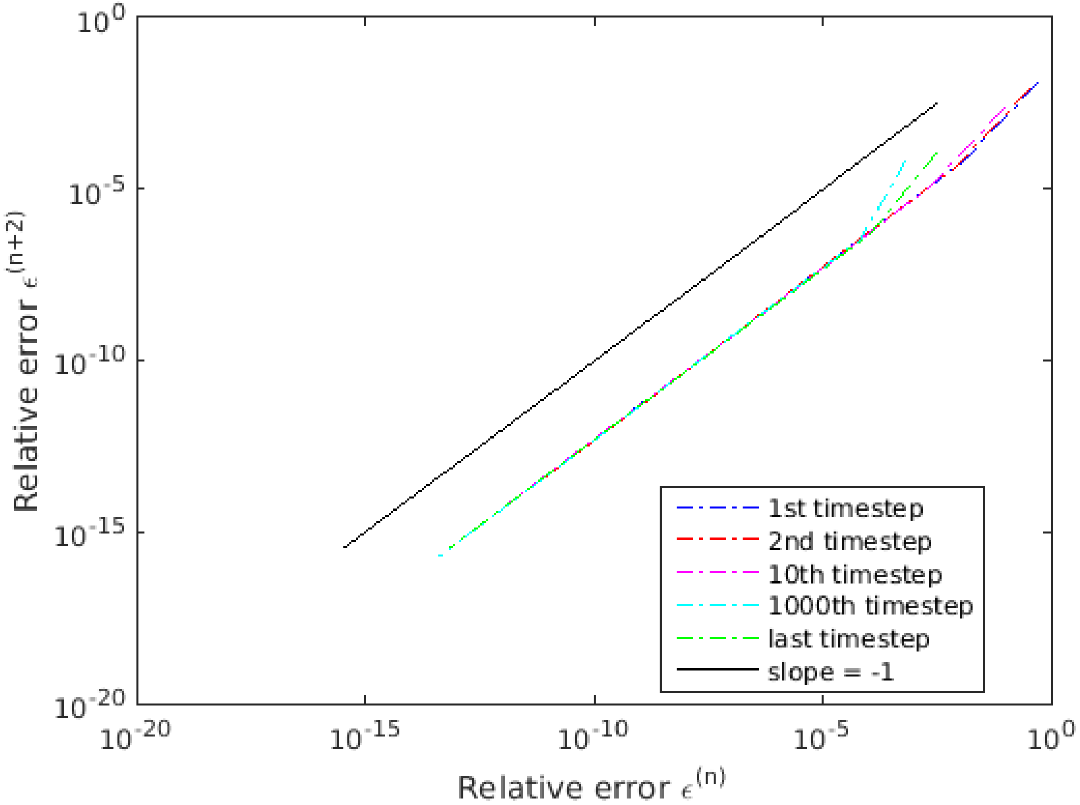
Bolted Joint Problem: EQPS



Single Ω

Schwarz

Bolted Joint Problem: Convergence Rate



Linear convergence rate is observed for ***dynamic*** Schwarz algorithm, as for the quasistatic Schwarz algorithm.

Figure above: Convergence behavior of the dynamic Schwarz algorithm for the bolted joint problem

Bolted Joint Problem: Performance

	CPU times (64 procs*)	Avg # Schwarz iters	Max # Schwarz iters
Single Domain	3h 34m	–	–
Schwarz	2h 42m	3.22	4
Single Domain (finer)	17h 00m	–	–
Schwarz (finer mesh of bolts)	29h 29m	3.28	4

* On SNL ascicgpu15, 16, 17 machines (Intel Skylake CPU processor), Schwarz tol = 1e-6.

Bolted Joint Problem: Performance

	CPU times (64 procs*)	Avg # Schwarz iters	Max # Schwarz iters
Single Domain	3h 34m	–	–
Schwarz	2h 42m	3.22	4
Single Domain (finer)	17h 00m	–	–
Schwarz (finer mesh of bolts)	29h 29m	3.28	4

- Despite its iterative nature, Schwarz can actually be ***faster*** than single domain run for discretizations having comparable # of elements in the bolts.

* On SNL ascicgpu15, 16, 17 machines (Intel Skylake CPU processor), Schwarz tol = 1e-6.

Bolted Joint Problem: Performance

	CPU times (64 procs*)	Avg # Schwarz iters	Max # Schwarz iters
Single Domain	3h 34m	–	–
Schwarz	2h 42m	3.22	4
Single Domain (finer)	17h 00m	–	–
Schwarz (finer mesh of bolts)	29h 29m	3.28	4

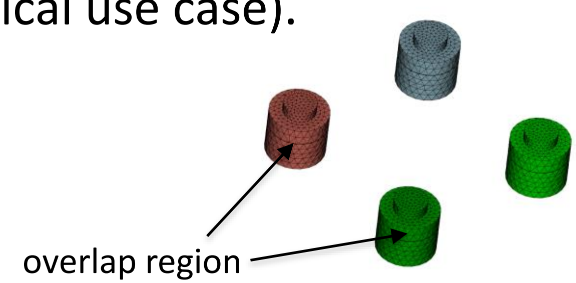
- Despite its iterative nature, Schwarz can actually be ***faster*** than single domain run for discretizations having comparable # of elements in the bolts.
  - Even if the method is more computationally expensive for some resolutions, it may be preferred for its ability to ***rapidly change*** and ***evaluate*** a ***variety*** of ***engineering designs*** (our typical use case).

* On SNL ascicgpu15, 16, 17 machines (Intel Skylake CPU processor), Schwarz tol = 1e-6.

Bolted Joint Problem: Performance

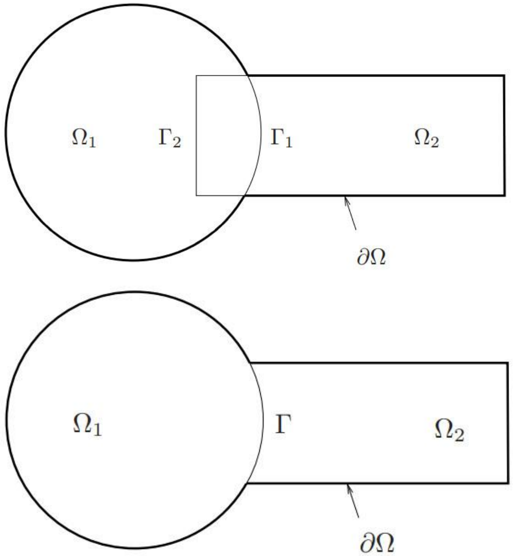
	CPU times (64 procs*)	Avg # Schwarz iters	Max # Schwarz iters
Single Domain	3h 34m	–	–
Schwarz	2h 42m	3.22	4
Single Domain (finer)	17h 00m	–	–
Schwarz (finer mesh of bolts)	29h 29m	3.28	4

- Despite its iterative nature, Schwarz can actually be ***faster*** than single domain run for discretizations having comparable # of elements in the bolts.
  - ➢ Even if the method is more computationally expensive for some resolutions, it may be preferred for its ability to ***rapidly change*** and ***evaluate*** a ***variety*** of ***engineering designs*** (our typical use case).
- Dynamic Schwarz converges in between ***2-4 Schwarz iterations*** per time-step despite the ***overlap*** region being ***very small*** for this problem.



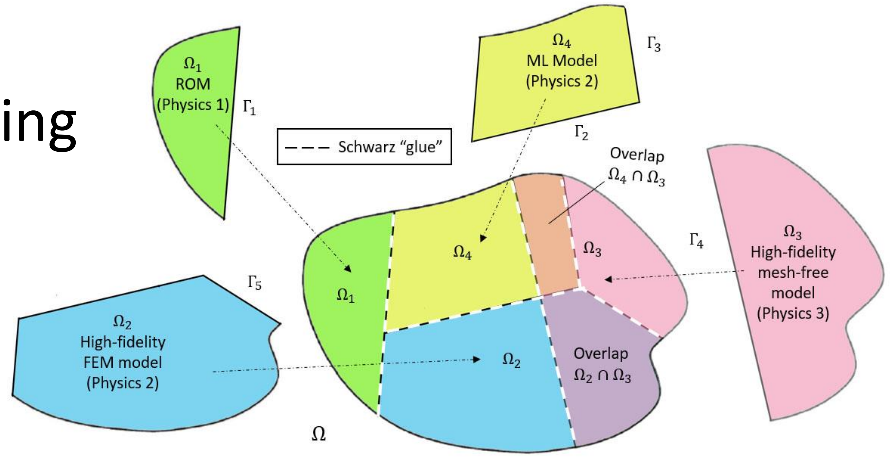
* On SNL ascicgpu15, 16, 17 machines (Intel Skylake CPU processor), Schwarz tol = 1e-6.

Outline



1. **Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics**
   - Motivation & Background
   - Quasistatics
   - Extension to Dynamics
   - **Summary & Next Steps**

2. Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling
   - Motivation & Background
   - Demonstration
   - Ongoing & Future Work



Summary

The **Schwarz alternating method** has been developed/implemented for **concurrent multiscale quasistatic & dynamic modeling** in Sandia's *Albany LCM* and *Sierra/SM* codes.

- ☺ Coupling is ***concurrent*** (two-way).
- ☺ ***Ease of implementation*** into existing massively-parallel HPC codes.
- ☺ ***Scalable, fast, robust*** (we target ***real*** engineering problems, e.g., analyses involving failure of bolted components!).
- ☺ ***"Plug-and-play" framework***: simplifies task of meshing complex geometries!
  - ☺ Ability to couple regions with ***different non-conformal meshes***, ***different element types*** and ***different levels of refinement.***
  - ☺ Ability to use ***different solvers/time-integrators*** in different regions.
- ☺ Coupling does not introduce ***nonphysical artifacts.***
- ☺ ***Theoretical*** convergence properties/guarantees.

Ongoing Work and Next Steps

- Continuing to ***apply*** the Schwarz alternating method to problems of interest to ***production*** using *Sierra/SM*.
- Advancing the Schwarz alternating method to enable ***coupling*** of ***structural elements*** to ***continuum elements***.
- Developing a ***Schwarz-like algorithm*** for simulating ***contact*** to address well-known challenges (contact constraint enforcement, multiple scales).
  - Done by introducing a ***combination*** of ***Dirichlet*** and ***Neumann*** boundary conditions into different subdomains in a ***non-overlapping*** alternating fashion.
  - See the following ***reference*** for more info:

J. Hoy, **I. Tezaur**, A. Mota. "The Schwarz alternating method for multiscale contact mechanics". in *Computer Science Research Institute Summer Proceedings 2021,* J.D. Smith and E. Galvan, eds., Technical Report SAND2021-0653R, Sandia National Labs, 360-378, 2021. (https://www.sandia.gov/ccr/csri-summer-programs/2021-proceedings/)

  - ***Journal article*** with D. Koliesnikova in preparation.

References – Part 1

[1] M.A. Heroux *et al.* "An overview of the Trilinos project." *ACM Trans. Math. Softw.* 31(3) (2005)

[2] 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.

[3] H. Schwarz. "Über einen Grenzübergang durch alternierendes Verfahren". In: Vierteljahrsschriftder Naturforschenden Gesellschaft in Zurich 15 (1870), pp. 272–286.

[4] S.L. Sobolev. "Schwarz's Algorithm in Elasticity Theory". In: Selected Works of S.L Sobolev. Volume I: equations of mathematical physics, computational mathematics and cubature formulats. Ed. By G.V. Demidenko and V.L. Vaskevich. New York: Springer, 2006.

[5] S. Mikhlin. "On the Schwarz algorithm". In: Proceedings of the USSR Academy of Sciences (in Russian) 77 (1951), pp. 569–571.

[6] D.J. Evans et al. "The convergence rate of the Schwarz alternating procedure (II): For two-dimensional problems". In: International Journal for Computer Mathematics 20.3–4 (1986), pp. 325–339.

[7] P.L. Lions. "On the Schwarz alternating method I." In: 1988, First International Symposium on Domain Decomposition methods for Partial Differential Equations, SIAM, Philadelphia.

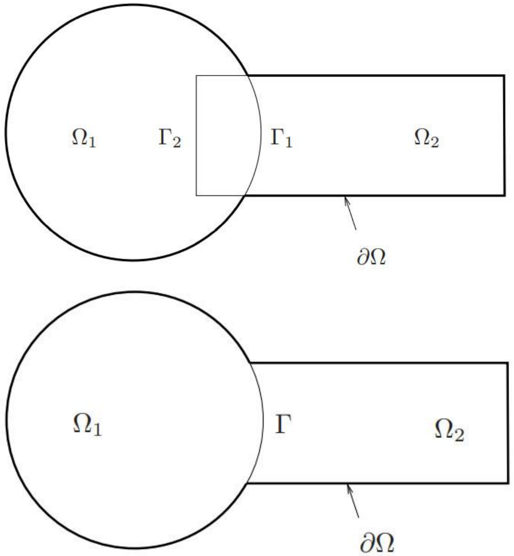
[8] SIERRA Solid Mechanics Team. Sierra/Solid Mechanics 4.48 User's Guide. Tech. rep. SAND2018-2961. Sandia National Laboratories Report, Oct. 2018.

[9] A. Mota, **I. Tezaur**, C. Alleman. "The Schwarz Alternating Method in Solid Mechanics", *Comput. Meth. Appl. Mech. Engng.* 319 (2017), 19-51.

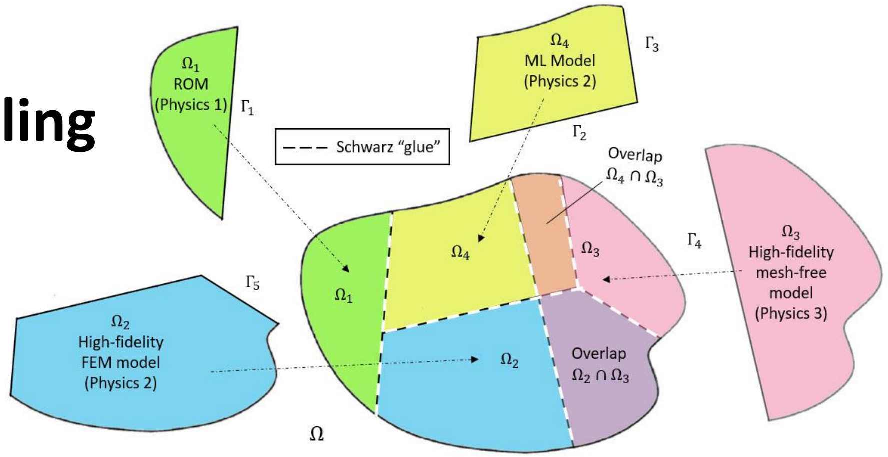
[10] A. Mota, **I. Tezaur**, G. Phlipot. "The Schwarz Alternating Method for Dynamic Solid Mechanics", *Comput. Meth. Appl. Mech. Engng.* 121 (21) (2022) 5036-5071.

Outline

1. Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics
   - Motivation & Background
   - Quasistatics
   - Extension to Dynamics
   - Summary & Next Steps

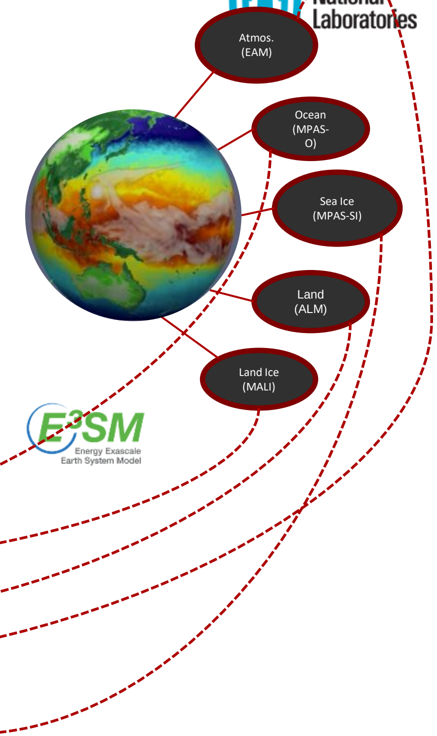


2. **Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling**
   - **Motivation & Background**
   - Demonstration
   - Ongoing & Future Work

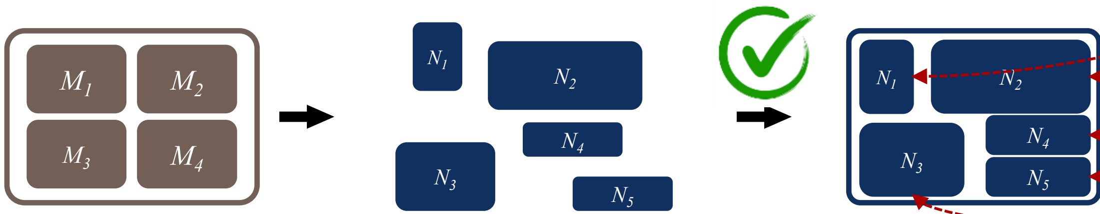


Motivation

The past decades have seen tremendous investment in **simulation frameworks** for **coupled multi-scale** and **multi-physics** problems.



- Frameworks rely on **established mathematical theories** to couple physics components.
- Most existing coupling frameworks are based on **traditional discretization methods**.



Complex System Model
- PDEs, ODEs
- Nonlocal integral
- Classical DFT
- Atomistic, …

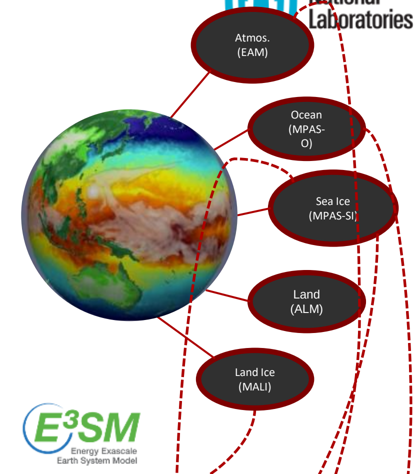
Traditional Methods
- Mesh-based (FE, FV, FD)
- Meshless (SPH, MLS)
- Implicit, explicit
- Eulerian, Lagrangian, …

Coupled Numerical Model
- Monolithic (Lagrange multipliers)
- Partitioned (loose) coupling
- Iterative (Schwarz, optimization)

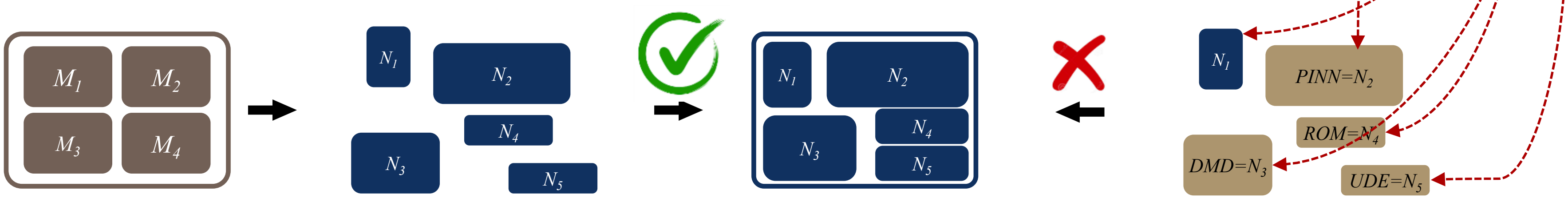
Traditional + Data-Driven Methods
- PINNs
- Neural ODEs
- Projection-based ROMs, …

Motivation

The past decades have seen tremendous investment in simulation frameworks for coupled multi-scale and multi-physics problems.



- Frameworks rely on **established mathematical theories** to couple physics components.
- Most existing coupling frameworks are based on **traditional discretization methods**.



Complex System Model
- PDEs, ODEs
- Nonlocal integral
- Classical DFT
- Atomistic, …

Traditional Methods
- Mesh-based (FE, FV, FD)
- Meshless (SPH, MLS)
- Implicit, explicit
- Eulerian, Lagrangian, …

Coupled Numerical Model
- Monolithic (Lagrange multipliers)
- Partitioned (loose) coupling
- Iterative (Schwarz, optimization)

Traditional + Data-Driven Methods
- PINNs
- Neural ODEs
- Projection-based ROMs, …

- There is currently a big push to integrate **data-driven methods** into modeling & simulation toolchains.

Unfortunately, existing algorithmic and software infrastructures are **ill-equipped** to handle plug-and-play integration of **non-traditional, data-driven models**!

Flexible Heterogeneous Numerical Methods (fHNM) Project

$\int \mathcal{M}^2 \mathrm{d}t$

Principal research objective:

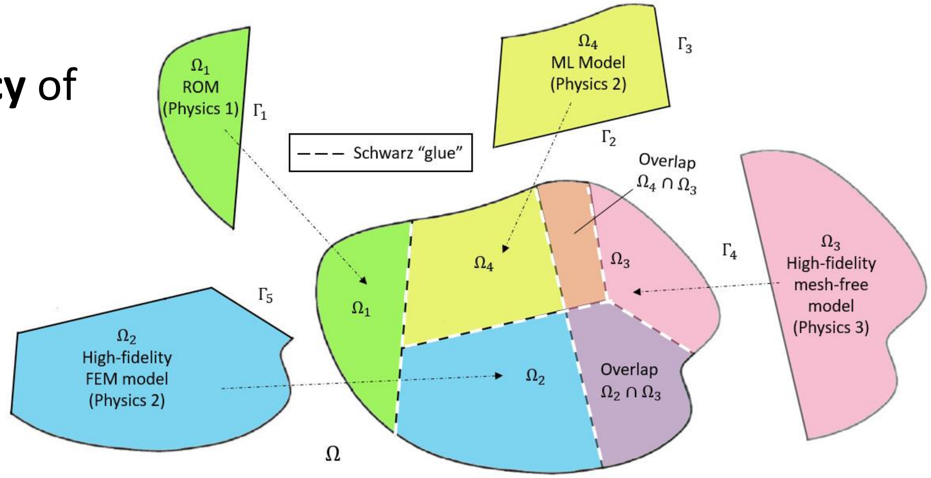
- **Discover mathematical principles** guiding the assembly of **standard** and **data-driven** numerical models in stable, accurate and physically consistent ways.

Principal research challenges: we lack mathematical and algorithmic understanding of how to

- **"Mix-and-match" standard** and **data-driven models** from three-classes
  - *Class A:* projection-based reduced order models (ROMs) ***This talk.***
  - *Class B:* machine-learned models, i.e., Physics-Informed Neural Networks (PINNs)
  - *Class C:* flow map approximation models, i.e., dynamic model decomposition (DMD) models
- Ensure **well-posedness & physical consistency** of resulting **heterogeneous models**.
- **Solve** such heterogeneous models efficiently.

Three coupling methods: ***This talk.***

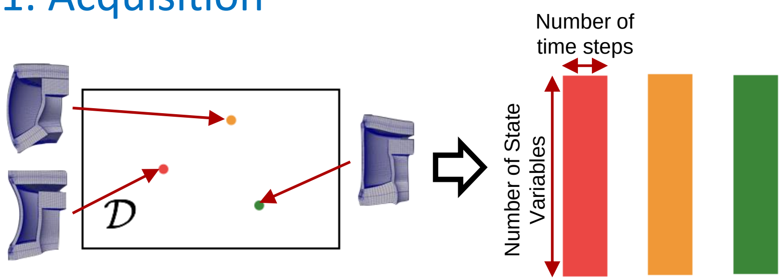
- Alternating Schwarz-based coupling
- Optimization-based coupling
- Coupling via generalized mortar methods



Projection-Based Model Order Reduction via the POD/Galerkin Method

Full Order Model (FOM): $\boldsymbol{M}\frac{d^2\boldsymbol{x}}{dt^2} + \boldsymbol{f}_{\text{int}}(\boldsymbol{x}) = \boldsymbol{f}_{\text{ext}}$

1. Acquisition

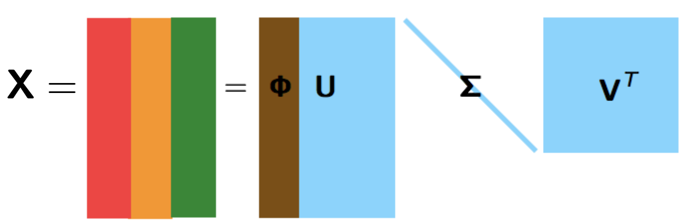


Solve ODE at different design points

Save solution data

2. Learning

Proper Orthogonal Decomposition (POD):



3. Projection-Based Reduction

Reduce the number of unknowns

$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{\Phi}\,\hat{\mathbf{x}}(t)$

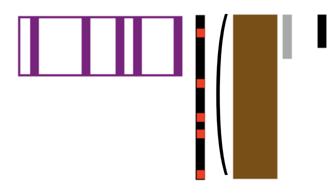
Perform Galerkin projection

$\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{\Phi}\frac{d^2\hat{\boldsymbol{x}}}{dt^2} + \boldsymbol{\Phi}^T\boldsymbol{f}_{\text{int}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}) = \boldsymbol{\Phi}^T\boldsymbol{f}_{\text{ext}}$

Hyper-reduce nonlinear terms

$\boldsymbol{f}_{\text{int}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}}) \approx \boldsymbol{A} \quad \boldsymbol{f}_{\text{int}}(\boldsymbol{\Phi}\hat{\boldsymbol{x}})$

Hyper-reduction/sample mesh



ROM = Projection-Based Reduced Order Model. HROM = Hyper-Reduced ROM

Schwarz Extensions to FOM-ROM and ROM-ROM Couplings

Enforcement of Dirichlet boundary conditions (DBCs) in ROM at indices $\boldsymbol{i}_{\text{Dir}}$

- Method I in [Gunzburger *et al.* 2007] is employed

$$\boldsymbol{u}(t) \approx \overline{\boldsymbol{u}} + \boldsymbol{\Phi}\hat{\boldsymbol{u}}(t), \quad \boldsymbol{v}(t) \approx \overline{\boldsymbol{v}} + \boldsymbol{\Phi}\hat{\boldsymbol{v}}(t), \quad \boldsymbol{a}(t) \approx \overline{\boldsymbol{a}} + \boldsymbol{\Phi}\hat{\boldsymbol{a}}(t)$$

  - POD modes made to satisfy homogeneous DBCs: $\boldsymbol{\Phi}(\boldsymbol{i}_{\text{Dir}}, :) = \boldsymbol{0}$
  - BCs imposed by modifying $\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}, \overline{\boldsymbol{a}}$: $\overline{\boldsymbol{u}}(\boldsymbol{i}_{\text{Dir}}) \leftarrow \boldsymbol{\chi}_u, \overline{\boldsymbol{v}}(\boldsymbol{i}_{\text{Dir}}) \leftarrow \boldsymbol{\chi}_v, \overline{\boldsymbol{a}}(\boldsymbol{i}_{\text{Dir}}) \leftarrow \boldsymbol{\chi}_a$

Choice of domain decomposition

- Error-based indicators that help decide in what region of the domain a ROM can be viable should drive domain decomposition (future work) [Bergmann *et al.* 2018]

Snapshot collection and reduced basis construction

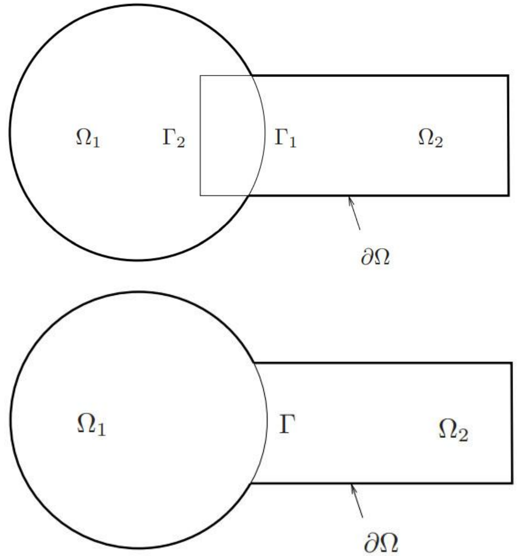
- Ideally, generate snapshots/reduced bases separately in each subdomain Ω_i [Hoang *et al.* 2021, Smetana *et al.*, 2022]
- POD results presented herein use snapshots obtained via FOM-FOM coupling on $\Omega = \cup_i \ \Omega_i$

For nonlinear problems, hyper-reduction needs to preserve Hamiltonian structure

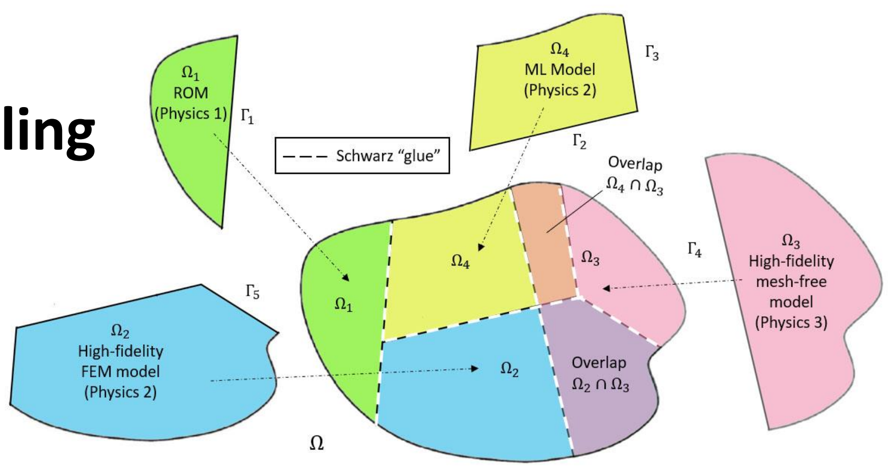
- We employ Energy-Conserving Sampling & Weighting Method (ECSW) [Farhat *et al.* 2015]
- Investigating structure-preserving coupling is near-term future work.

Outline

1. Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics
    - Motivation & Background
    - Quasistatics
    - Extension to Dynamics
    - Summary & Next Steps



2. **Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling**
    - Motivation & Background
    - **Demonstration**
    - Ongoing & Future Work



* Projection-based Reduced Order Model

Numerical Example: 1D Dynamic Wave Propagation Problem

- ***1D beam*** geometry $\Omega = (0,1)$, clamped at both ends, with Gaussian initial condition discretized using FEM + Newmark-β
- Simple problem but very ***stringent test*** for discretization/ coupling methods.
- Two **constitutive models** considered:
  - ➢ Linear elastic (problem has exact analytical solution)
  - ➢ Nonlinear hyperelastic Henky

This talk.

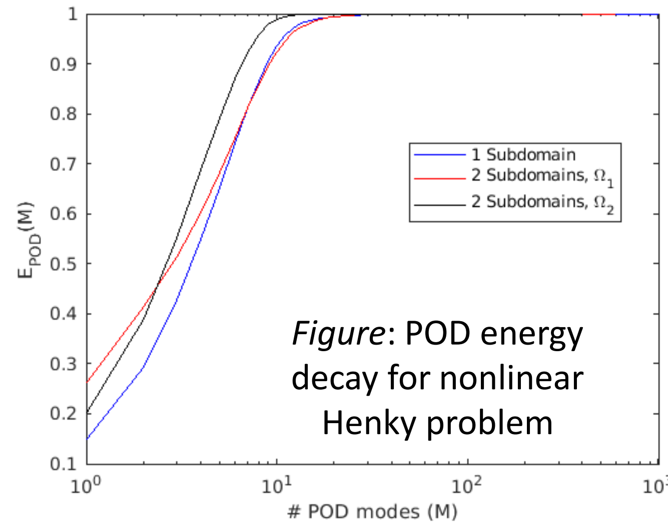


Figure: POD energy decay for nonlinear Henky problem

- ROMs results are ***reproductive*** and ***predictive***, and are based on the ***POD/Galerkin*** method, with POD calculated from FOM-FOM coupled model.
  - ➢ 50 POD modes capture ~100% snapshot energy for linear variant of this problem.
  - ➢ 536 POD modes capture ~100% snapshot energy for Henky variant of this problem.
- Hyper-reduced ROMs (HROMs) perform ***hyper-reduction*** using ECSW [Farhat *et al.*, 2015]
  - ➢ Ensures that ***Lagrangian structure*** of problem is preserved in HROM.
- ***Couplings tested:*** overlapping, non-overlapping, FOM-FOM, FOM-ROM, ROM-ROM, FOM-HROM, HROM-HROM, implicit-explicit, implicit-implicit, explicit-explicit. ***This talk.***

Numerical Example: 1D Dynamic Wave Propagation Problem

- **Two variants** of problem, with different initial conditions (ICs):
  - ➢ Symmetric Gaussian IC (top right)
  - ➢ Rounded Square IC (bottom right)
- **Non-overlapping domain decomposition (DD)** of $\Omega = \Omega_1 \cup \Omega_2$, where Ω_1 = [0, 0.6] and Ω_2 = [0.6, 1.0]
  - ➢ DD based on heuristics: during time-interval considered $(0 \leq t \leq 1 \times 10^3)$, sharper gradient forms in Ω_2, suggesting FOM or larger ROM is needed there.
- **Reproductive problem**:
  - ➢ Displacement snapshots collected using FOM-FOM non-overlapping coupling with **Symmetric Gaussian IC**
  - ➢ FOM-ROM, FOM-HROM, ROM-ROM and HROM-HROM run with **Symmetric Gaussian IC**
- **Predictive problem**:
  - ➢ Snapshots same as for reproductive problem.
  - ➢ FOM-ROM, FOM-HROM, ROM-ROM and HROM-HROM run with **Rounded Square IC**

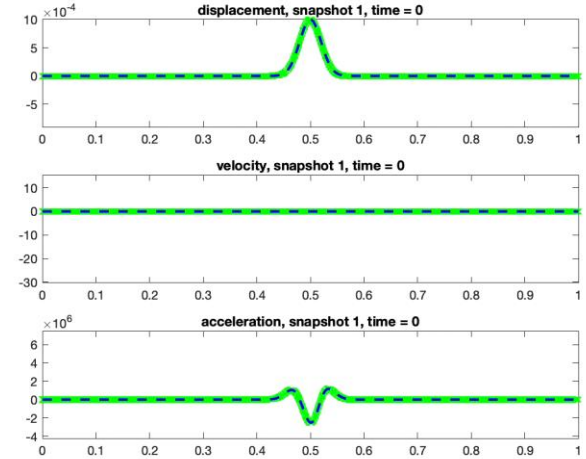
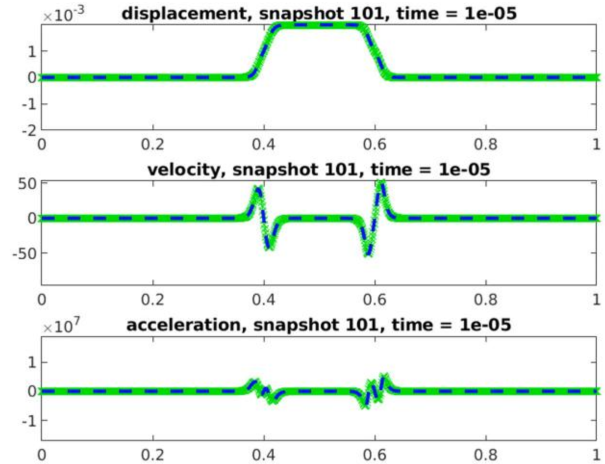


Figure above: Symmetric Gaussian IC
Figure below: Rounded Square IC



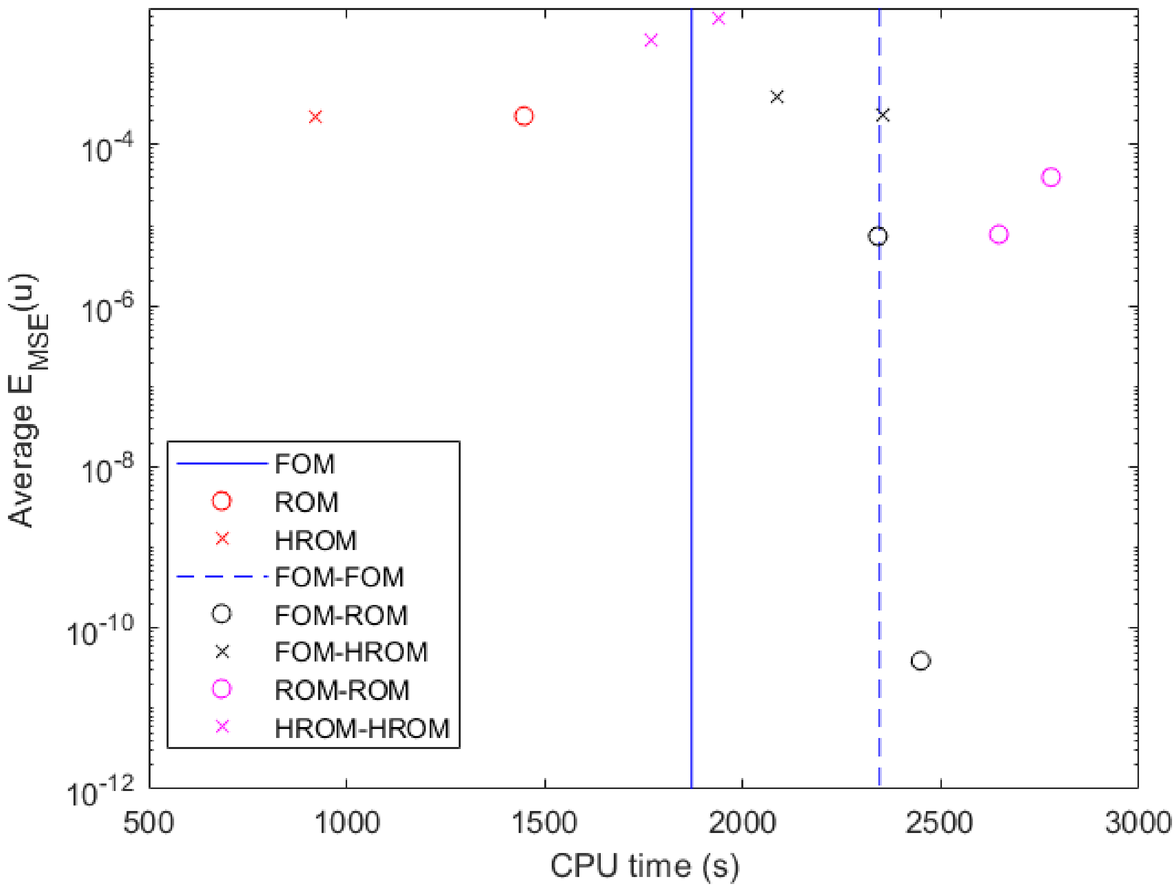
Numerical Example: Reproductive Results

Model	M_1/M_2	$N_{e,1}/N_{e,2}$	CPU time (s)	$\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{u}}_1)/$ $\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{u}}_2)$	$\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{v}}_1)/$ $\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{v}}_2)$	$\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{a}}_1)/$ $\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{a}}_2)$	N_S
FOM	$-/-$	$-/-$	1.871×10^3	$-/-$	$-/-$	$-/-$	$-$
ROM	$60/-$	$-/-$	1.398×10^3	$1.659 \times 10^{-2}/-$	1.037×10^{-1}	$4.681 \times 10^{-1}/-$	$-$
HROM	$60/-$	$155/-$	5.878×10^2	$1.730 \times 10^{-2}/-$	$1.063 \times 10^{-1}/-$	$4.741 \times 10^{-1}/-$	$-$
ROM	$200/-$	$-/-$	1.448×10^3	$2.287 \times 10^{-4}/-$	$4.038 \times 10^{-3}/-$	$4.542 \times 10^{-2}/-$	$-$
HROM	$200/-$	$428/-$	9.229×10^2	$8.396 \times 10^{-4}/-$	$8.947 \times 10^{-3}/-$	$7.462 \times 10^{-2}/-$	$-$
FOM-FOM	$-/-$	$-/-$	2.345×10^3	$-$	$-$	$-$	24,630
FOM-ROM	$-/80$	$-/-$	2.341×10^3	$2.171 \times 10^{-6}/$ 1.253×10^{-5}	$3.884 \times 10^{-5}/$ 2.401×10^{-4}	$2.982 \times 10^{-4}/$ 2.805×10^{-3}	25,227
FOM-HROM	$-/80$	$-/130$	2.085×10^3	$2.022 \times 10^{-4}/$ 5.734×10^{-4}	$1.723e \times 10^{-3}/$ 5.776×10^{-3}	$7.421 \times 10^{-3}/$ 3.791×10^{-2}	29,678
FOM-ROM	$-/200$	$-/-$	2.449×10^3	$4.754 \times 10^{-12}/$ 7.357×10^{-11}	$1.835 \times 10^{-10}/$ 4.027×10^{-9}	$5.550 \times 10^{-9}/$ 1.401×10^{-7}	24,630
FOM-HROM	$-/200$	$-/252$	2.352×10^3	$1.421 \times 10^{-5}/$ 4.563×10^{-4}	$1.724 \times 10^{-4}/$ 2.243×10^{-3}	$9.567 \times 10^{-4}/$ 1.364×10^{-2}	27,156
ROM-ROM	$200/80$	$-/-$	2.778×10^3	$4.861 \times 10^{-5}/$ 3.093×10^{-5}	$1.219 \times 10^{-3}/$ 4.177×10^{-4}	$1.586 \times 10^{-2}/$ 3.936×10^{-3}	27,810
HROM-HROM	$200/80$	$315/130$	1.769×10^3	$3.410 \times 10^{-3}/$ 6.662×10^{-4}	$4.110 \times 10^{-2}/$ 6.432×10^{-3}	$2.485 \times 10^{-1}/$ 4.307×10^{-2}	29,860
ROM-ROM	$300/80$	$-/-$	2.646×10^3	$2.580 \times 10^{-6}/$ 1.292×10^{-5}	$6.226 \times 10^{-5}/$ 2.483×10^{-4}	9.470×10^{-4} 2.906×10^{-3}	25,059
HROM-HROM	$300/80$	$405/130$	1.938×10^3	6.960×10^{-3} 7.230×10^{-4}	6.328×10^{-2} 7.403×10^{-3}	3.137×10^{-1} 4.960×10^{-2}	29,896

Green shading highlights most competitive coupled models

- All coupled models evaluated converged on average in <**3 Schwarz iterations** per time-step
- FOM-ROM coupling has **same total # Schwarz iters** (N_S) as FOM-FOM coupling
- Other couplings require more Schwarz iters than FOM-FOM coupling to converge
  - ➢ **More Schwarz iters** required when coupling **less accurate models**
  - ➢ Larger 300/80 mode ROM-ROM takes less time than smaller 200/80 mode ROM-ROM
- **FOM-HROM** & **HROM-HROM** couplings **outperform FOM-FOM** coupling in CPU time by 12.5-32.6%
- All couplings involving ROMs/HROMs are **at least as accurate** as single-domain ROMs/HROMs

Numerical Example: Reproductive Results



- Single-domain ROM and HROM are most efficient
- Couplings involving ROMs and HROMs enable one to achieve smaller errors
- Benefits of hyper-reduction are limited on 1D problem

Numerical Example: Reproductive Results

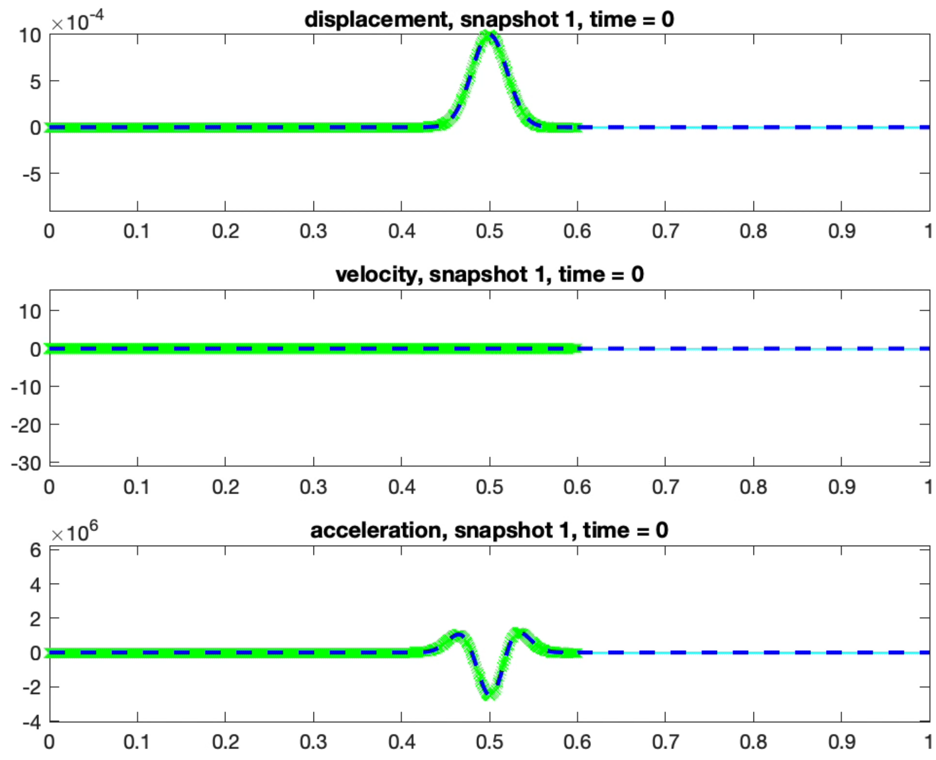
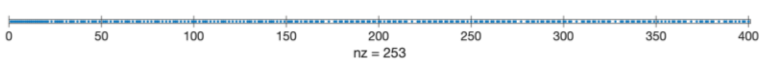


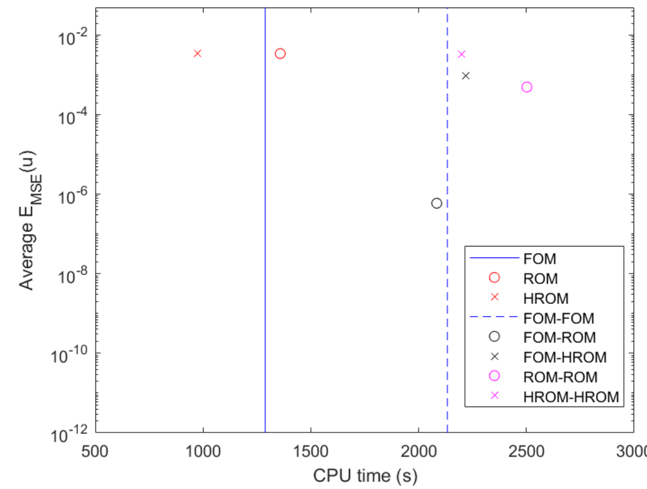
Figure left: FOM (green) – HROM (cyan) coupling compared with single-domain FOM solution (blue). HROM has 200 modes.

Figure below: ECSW algorithm samples 253/400 elements



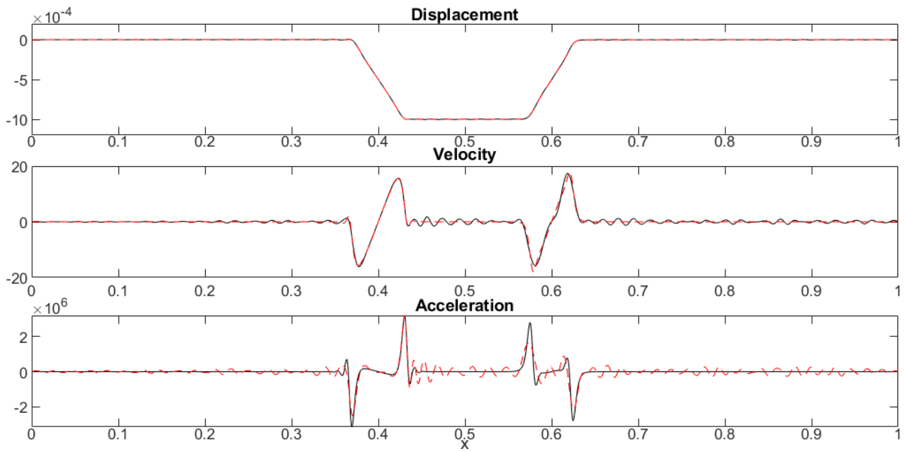
Numerical Example: Predictive Results

Model	CPU time (s)	$N_{e,1}/N_{e,2}$	$\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{u}}_1)$/ $\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{u}}_2)$	$\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{v}}_1)$/ $\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{v}}_2)$	$\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{a}}_1)$/ $\mathcal{E}_{\text{MSE}}(\tilde{\boldsymbol{a}}_2)$	N_S
FOM	1.288×10^3	−/−	−/−	−/−	−/−	−
ROM	1.358×10^3	−/−	3.451×10^{-3}/−	6.750×10^{-2}/−	3.021×10^{-1}/−	−
HROM	9.759×10^2	614/−	3.463×10^{-3}/−	6.750×10^{-2}/−	3.021×10^{-1}/−	−
FOM-FOM	2.133×10^3	−/−	−/−	−/−	−/−	23,280
FOM-ROM	2.084×10^3	−/−	1.907×10^{-8}/ 1.170×10^{-6}	1.461×10^{-6}/ 9.882×10^{-5}	3.973×10^{-5}/ 1.757×10^{-3}	23,288
FOM-HROM	2.219×10^3	−/253	1.967×10^{-4} 1.720×10^{-3}	4.986×10^{-3} 4.185×10^{-2}	2.768×10^{-2} 2.388×10^{-1}	29,700
ROM-ROM	2.502×10^3	−/−	5.592×10^{-4}/ 4.346×10^{-4}	1.575×10^{-2}/ 1.001×10^{-2}	9.197×10^{-2}/ 5.304×10^{-2}	26,220
HROM-HROM	2.200×10^3	405/253	4.802×10^{-3} 1.960×10^{-3}	8.500×10^{-2} 4.630×10^{-2}	3.744×10^{-1} 2.580×10^{-1}	30,067

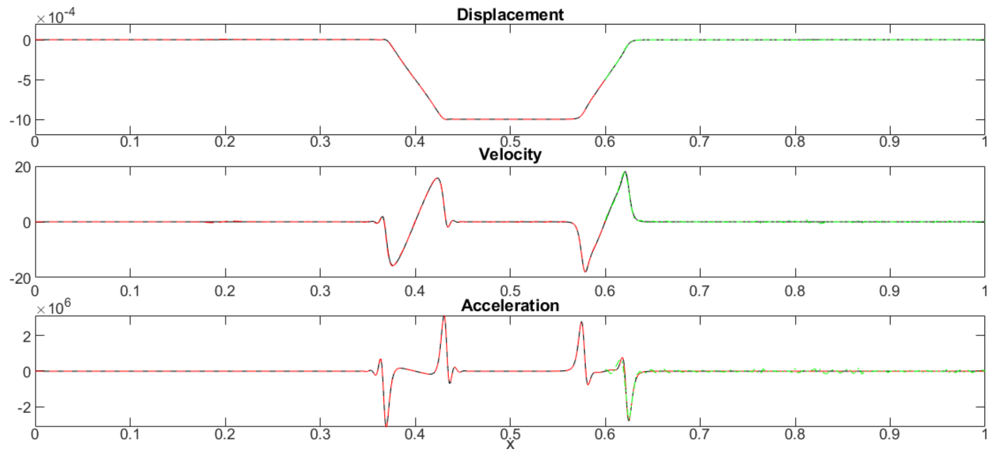


- **Predictive accuracy/robustness** can be **improved** by **coupling** ROM or HROM to FOM
  - FOM-ROM coupling is **remarkably accurate**, achieving displacement error $O(1 \times 10^{-8})$
  - FOM-HROM and ROM-ROM couplings are **more accurate** than single-domain ROMs
  - HROM-HROM **on par** with single-domain HROM in terms of accuracy
- **Wall-clock times** of coupled models can be improved
  - FOM-HROM, ROM-ROM and HROM-HROM models are **slowe**r than FOM-FOM model as **more Schwarz iterations** required to achieve convergence
  - **Hyper-reduction** samples ~60% of total mesh points
    - Greater gains from hyper-reduction anticipated for **2D/3D problems**

Numerical Example: Predictive Results



Predictive single-domain ROM (M_1= 300) solution at final time

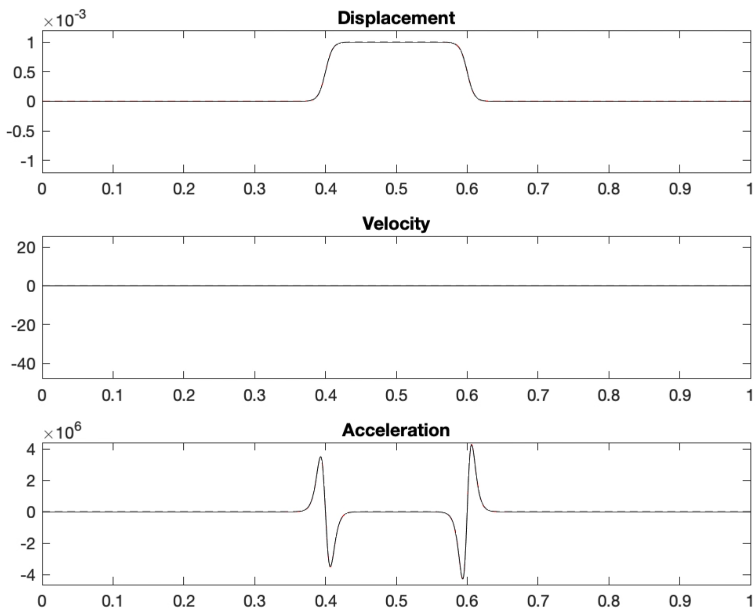


Predictive FOM-HROM (M_2= 200) solution at final time

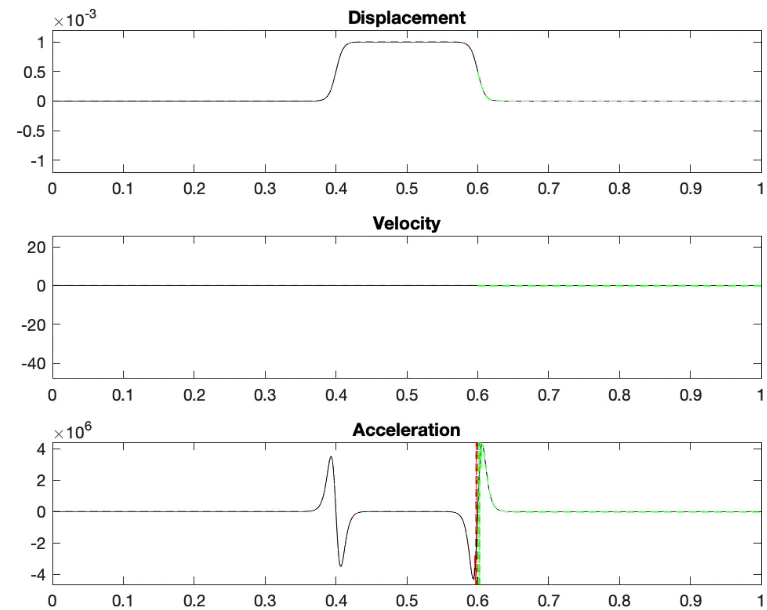
— Single-domain FOM solution — Solution in Ω_1 — Solution in Ω_2

- Predictive **single-domain ROM** solution exhibits **spurious oscillations** in velocity and acceleration
- Predictive **FOM-HROM** solution is **smooth** and **oscillation-free**
  - ➢ Highlights coupling method's ability to **improve ROM predictive accuracy**

Numerical Example: Predictive Results



Predictive single-domain ROM (M_1= 300)

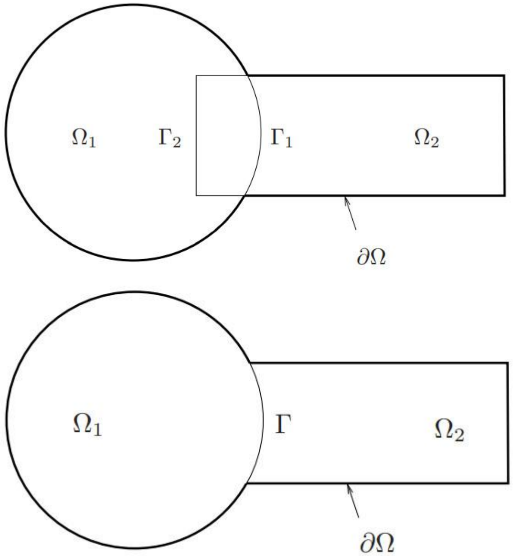


Predictive FOM-HROM (M_2= 200)

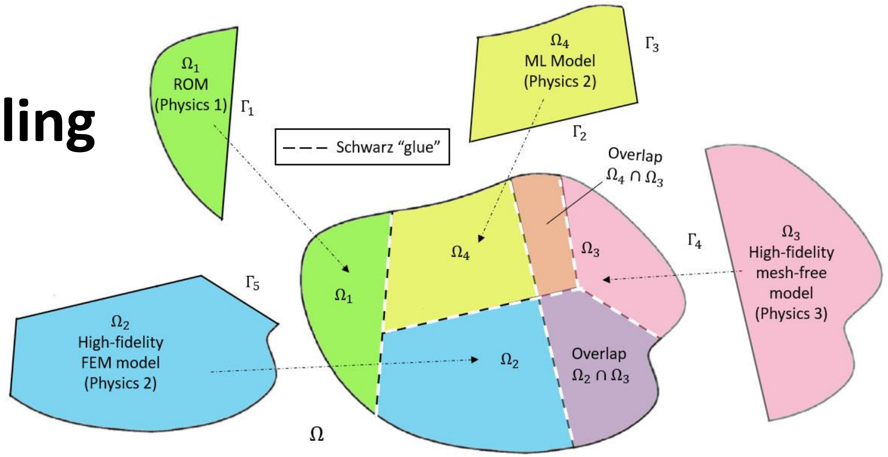
— Single-domain FOM solution — Solution in Ω_1 — Solution in Ω_2

Outline

1. Alternating Schwarz Method for Coupling of Full Order Models (FOMs) in Solid Mechanics
   - Motivation & Background
   - Quasistatics
   - Extension to Dynamics
   - Summary & Next Steps



2. **Alternating Schwarz Method for FOM-ROM* and ROM-ROM Coupling**
   - Motivation & Background
   - Demonstration
   - **Ongoing & Future Work**



Ongoing & Future Work

Ongoing Work

- Extension to **multi-material** problems (method works; results not shown herein).
- Development of **additive Schwarz** variant, which has more potential for parallelization and speed-ups (preliminary FOM-FOM results show promise).
- Extension of method to **multi-D** through newly-developed **Python** code, **pressio-demoapps**[1] **library** and/or **3D solid mechanics Julia code**[2].
  - ➢ Multi-D implementation requires transfer operators for transmission BCs
- **Journal publication** on results presented here + 2D/3D results.
- Alternating Schwarz-based coupling of **Physics-Informed Neural Networks (PINNs)**

Next Steps

- Development of **smart DD approaches** based on error indicators, to determine optimal placement of ROM and FOM (including **on-the-fly ROM-FOM switching**).
- **Analysis** of method's convergence properties for non-overlapping and ROM/HROM coupling cases.
- Extension of coupling approach to **POD modes** built from snapshots on **independently-simulated subdomains**.
- Application to **other problems**, including **multi-physics problems**.
- **Structure-preservation** within ROMs and couplings involving ROMs

$\int \mathcal{M}^2 \mathrm{d}t$

[1] https://github.com/Pressio/pressio-demoapps, [2] https://github.com/lxmota/norma

References – Part 2

[11] M. Gunzburger, J. Peterson, J. Shadid. “Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data”. *CMAME* 196 (2007) 1030-1047.

[12] C. Hoang, Y. Choi, K. Carlberg. “Domain-decomposition least-squares Petrov–Galerkin (DD-LSPG) nonlinear model reduction”. *CMAME* 384 (2021) 113997.

[13] K. Smetana, T. Taddei. “Localized model reduction for nonlinear elliptic partial differential equations: localized training, partition of unity, and adaptive enrichment”, *ArXiV pre-print*, 2022.

[14] C. Farhat, T. Chapman, P. Avery. “Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models”, *IJNME* 102 (2015) 1077-1110.

[15] M. Bergmann, A. Ferrero, A. Iollo, E. Lombardi, A. Scardigli, H. Telib. “A zonal Galerkin-free POD model for incompressible flows.” *JCP* 352 (2018) 301-325.

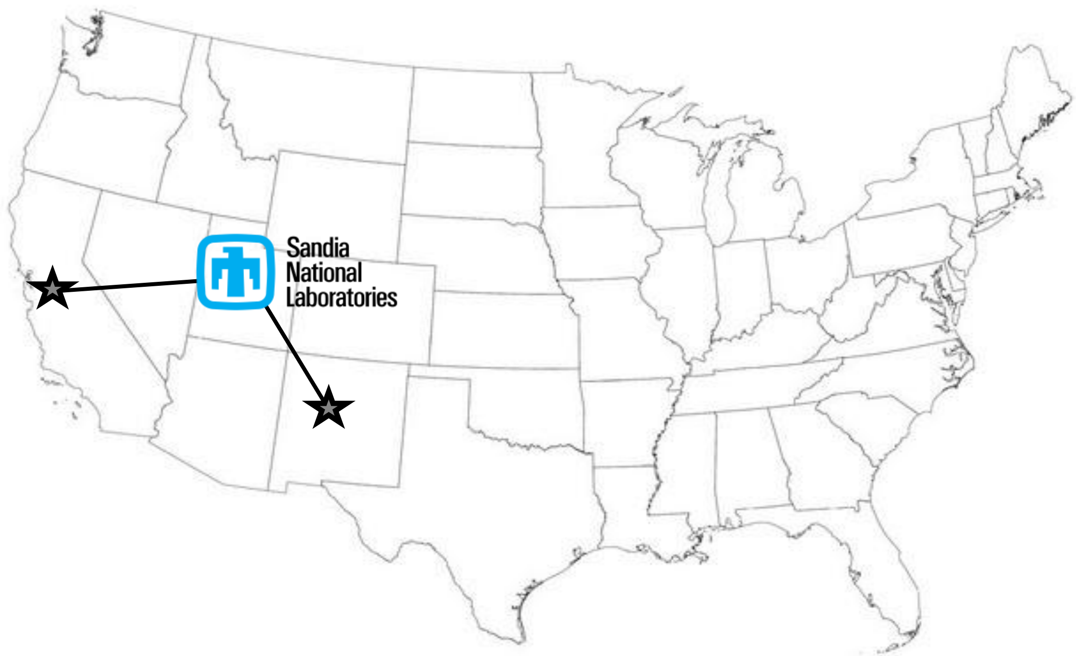
[16] A. Iollo, G. Sambataro, T. Taddei. “A one-shot overlapping Schwarz method for component-based model reduction: application to nonlinear elasticity”, *CMAME* 404 (2023) 115786.

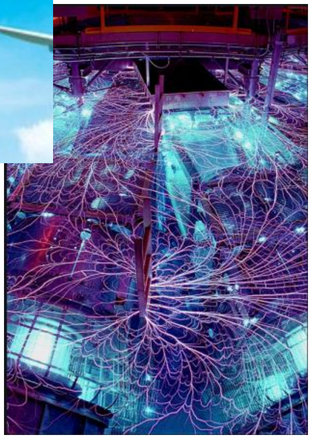
[17] J. Hoy, **I. Tezaur**, A. Mota. "The Schwarz alternating method for multiscale contact mechanics". in *Computer Science Research Institute Summer Proceedings 2021,* J.D. Smith and E. Galvan, eds., Technical Report SAND2021-0653R, Sandia National Labs, 360-378, 2021.

[18] J. Barnett, **I. Tezaur**, A. Mota. "The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models", in *Computer Science Research Institute Summer Proceedings 2022*, S.K. Seritan and J.D. Smith, eds., Technical Report SAND2022-10280R, Sandia National Laboratories, 2022, pp. 31-55.

Careers at Sandia National Labs

Students: please consider Sandia and other national labs as a potential employer for summer internships and when you graduate!





- Sandia is a **multidisciplinary** national lab and Federally Funded Research & Development Center (FFRDC).
- Contractor for U.S. DOE's National Nuclear Security Administration (**NNSA**).
- **Two main sites**: Albuquerque, NM and Livermore, CA

Careers at Sandia National Labs

<u>Students:</u> please consider Sandia and other national labs as a potential employer for summer internships and when you graduate!

- Sandia is a ***great*** place to work!
  - Lots of ***interesting*** problems that require ***fundamental research*** in applied math/computational science and impact ***mission-critical applications***.
  - Great ***work/life balance***.
- ***Opportunities*** at/with Sandia:
  - Interns (summer, year-round)
  - Post docs
  - Several prestigious post doctoral fellowships (von Neumann, Truman, Hruby, Data Science)
  - Staff

Please see: www.sandia.gov/careers for info about current opportunities.

Careers at Sandia National Labs

Sandia National Laboratories

Intern- Scientific Machine Learning Technical Graduate Summer

Job ID	686827
Location	Albuquerque, NM
Department	Scientific Machine Learning
Job Family	Student
Posted Date	11/30/2022

Intern - Computer Science Research Institute (CSRI) - R&D Graduate Summer

Job ID	685380
Location	Multiple
Department	Scalable Algorithms
Job Family	Student
Posted Date	12/14/2022

Intern - Computer Science Research Institute (CSRI) - R&D Undergraduate Summer

Job ID	685381
Location	Multiple
Department	Scalable Algorithms
Job Family	Student
Posted Date	12/01/2022

⋮

Left: current postings from www.sandia.gov/careers for Summer 2023. ***Summer interns are hired by early March.***

I am hiring interns to help with ROM-ROM/ROM-FOM coupling work!

If interested, email me (ikalash@sandia.gov) and apply to the CSRI Internship Posting (left).

Start of Backup Slides

Appendix. Four Variants* of Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Schwarz loop
  $\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check
  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
  repeat ▷ Newton loop for Ω_1
    $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
    $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
  until $\|\triangle\boldsymbol{x}_B^{(1)}\|/\|\boldsymbol{x}_B^{(1)}\| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance
  $\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check
  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
  repeat ▷ Newton loop for Ω_2
    $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system
    $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
  until $\|\triangle\boldsymbol{x}_B^{(2)}\|/\|\boldsymbol{x}_B^{(2)}\| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance
until $\left[\left(\|\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}\|/\|\boldsymbol{x}_B^{(1)}\|\right)^2 + \left(\|\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}\|/\|\boldsymbol{x}_B^{(2)}\|\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Full Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Newton-Schwarz loop
  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
  $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
  $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
  $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system
  $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
until $\left[\left(\|\triangle\boldsymbol{x}_B^{(1)}\|/\|\boldsymbol{x}_B^{(1)}\|\right)^2 + \left(\|\triangle\boldsymbol{x}_B^{(2)}\|/\|\boldsymbol{x}_B^{(2)}\|\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Modified Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Schwarz loop
  $\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check
  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
  repeat ▷ Newton loop for Ω_1
    $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
    $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
  until $\|\triangle\boldsymbol{x}_B^{(1)}\|/\|\boldsymbol{x}_B^{(1)}\| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$
  $\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check
  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
  repeat ▷ Newton loop for Ω_2
    $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ solve linear system
    $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
  until $\|\triangle\boldsymbol{x}_B^{(2)}\|/\|\boldsymbol{x}_B^{(2)}\| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$
until $\left[\left(\|\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}\|/\|\boldsymbol{x}_B^{(1)}\|\right)^2 + \left(\|\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}\|/\|\boldsymbol{x}_B^{(2)}\|\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Inexact Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, ▷ initialize for Ω_2
repeat ▷ Newton-Schwarz loop
  $\begin{Bmatrix} \triangle\boldsymbol{x}_B^{(1)} \\ \triangle\boldsymbol{x}_B^{(2)} \end{Bmatrix} \leftarrow \begin{pmatrix} \boldsymbol{K}_{AB}^{(1)} + \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{11} & \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{12} \\ \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{21} & \boldsymbol{K}_{AB}^{(2)} + \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{22} \end{pmatrix} \backslash \begin{Bmatrix} -\boldsymbol{R}_A^{(1)} \\ -\boldsymbol{R}_A^{(2)} \end{Bmatrix}$ ▷ linear system
  $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
  $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
until $\left[\left(\|\triangle\boldsymbol{x}_B^{(1)}\|/\|\boldsymbol{x}_B^{(1)}\|\right)^2 + \left(\|\triangle\boldsymbol{x}_B^{(2)}\|/\|\boldsymbol{x}_B^{(2)}\|\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Monolithic Schwarz

Appendix. Four Variants* of Schwarz

Modified and ***Monolithic*** Schwarz variants create and solve a ***block system***.

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1

$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2

repeat ▷ Schwarz loop

  $\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check

  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1

  repeat ▷ Newton loop for Ω_1

    $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)}) \backslash \boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system

    $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$

  until $||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}|| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

  $\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check

  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2

  repeat ▷ Newton loop for Ω_2

    $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)}) \backslash \boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system

    $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$

  until $||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}|| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

until $\left[\left(||\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Full Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1

$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2

repeat ▷ Newton-Schwarz loop

  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1

  $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)}) \backslash \boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system

  $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$

  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2

  $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)}) \backslash \boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system

  $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$

until $\left[\left(||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Modified Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1

$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2

repeat ▷ Schwarz loop

  $\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check

  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1

  repeat ▷ Newton loop for Ω_1

    $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)}) \backslash \boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system

    $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$

  until $||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}|| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$

  $\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check

  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2

  repeat ▷ Newton loop for Ω_2

    $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)}) \backslash \boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ solve linear system

    $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$

  until $||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}|| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$

until $\left[\left(||\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Inexact Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, ▷ initialize for Ω_1

$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, ▷ initialize for Ω_2

repeat ▷ Newton-Schwarz loop

  $$\begin{Bmatrix} \triangle\boldsymbol{x}_B^{(1)} \\ \triangle\boldsymbol{x}_B^{(2)} \end{Bmatrix} \leftarrow \begin{pmatrix} \boldsymbol{K}_{AB}^{(1)} + \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{11} & \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{12} \\ \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{21} & \boldsymbol{K}_{AB}^{(2)} + \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{22} \end{pmatrix} \Big\backslash \begin{Bmatrix} -\boldsymbol{R}_A^{(1)} \\ -\boldsymbol{R}_A^{(2)} \end{Bmatrix}$$ ▷ linear system

  $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$

  $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$

until $\left[\left(||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Monolithic Schwarz

*A. Mota, **I. Tezaur**, C. Alleman. "The Schwarz Alternating Method in Solid Mechanics", *CMAME* 319 (2017), 19-51.

Appendix. Four Variants* of Schwarz

Most performant method: monotonic convergence, theoretical convergence guarantee.

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Schwarz loop
$\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check
$\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
repeat ▷ Newton loop for Ω_1
$\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
until $||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}|| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance
$\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check
$\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
repeat ▷ Newton loop for Ω_2
$\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
until $||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}|| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance
until $\left[\left(||\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Full Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Newton-Schwarz loop
$\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
$\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
$\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
$\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
until $\left[\left(||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Modified Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Schwarz loop
$\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check
$\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
repeat ▷ Newton loop for Ω_1
$\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
until $||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}|| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$
$\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check
$\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
repeat ▷ Newton loop for Ω_2
$\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ solve linear system
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
until $||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}|| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$
until $\left[\left(||\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Inexact Schwarz

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, ▷ initialize for Ω_2
repeat ▷ Newton-Schwarz loop

$$\begin{Bmatrix} \triangle\boldsymbol{x}_B^{(1)} \\ \triangle\boldsymbol{x}_B^{(2)} \end{Bmatrix} \leftarrow \begin{pmatrix} \boldsymbol{K}_{AB}^{(1)} + \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{11} & \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{12} \\ \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{21} & \boldsymbol{K}_{AB}^{(2)} + \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{22} \end{pmatrix} \backslash \begin{Bmatrix} -\boldsymbol{R}_A^{(1)} \\ -\boldsymbol{R}_A^{(2)} \end{Bmatrix}$$ ▷ linear system

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
until $\left[\left(||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Monolithic Schwarz

Appendix. Full Schwarz Method

Classical algorithm originally proposed by Schwarz with ***outer Schwarz loop*** and ***inner Newton loop***, each converged to a ***tight tolerance*** ($\epsilon_{machine}$).

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
repeat ▷ Schwarz loop
    $\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check
    $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
    repeat ▷ Newton loop for Ω_1
        $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
        $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
    until $||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}|| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance
    $\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check
    $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
    repeat ▷ Newton loop for Ω_2
        $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system
        $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
    until $||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}|| \leq \epsilon_{\text{machine}}$ ▷ tight tolerance
until $\left[\left(||\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Appendix. Inexact Schwarz Method

Classical algorithm originally proposed by Schwarz with ***outer Schwarz loop*** and ***inner Newton loop***, with Newton step converged to a ***loose tolerance***.

$\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \chi(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1

$\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \chi(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2

repeat ▷ Schwarz loop

  $\boldsymbol{y}^{(1)} \leftarrow \boldsymbol{x}_B^{(1)}$ ▷ for convergence check

  $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1

  repeat ▷ Newton loop for Ω_1

    $\triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system

    $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$

  until $||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}|| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$

  $\boldsymbol{y}^{(2)} \leftarrow \boldsymbol{x}_B^{(2)}$ ▷ for convergence check

  $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2

  repeat ▷ Newton loop for Ω_2

    $\triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ solve linear system

    $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$

  until $||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}|| \leq \epsilon$ ▷ loose tolerance, e.g. $\epsilon \in [10^{-4}, 10^{-1}]$

until $\left[\left(||\boldsymbol{y}^{(1)} - \boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\boldsymbol{y}^{(2)} - \boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\mathrm{machine}}$ ▷ tight tolerance

Appendix. Monolithic Schwarz Method

Combines Schwarz and Newton loop into ***since Newton-Schwarz loop***, with ***elimination of Schwarz boundary DOFs,*** and tight convergence tolerance.

1: $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_1$, ▷ initialize for Ω_1

2: $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_{\boldsymbol{\varphi}}\Omega_2$, ▷ initialize for Ω_2

3: **repeat** ▷ Newton-Schwarz loop

4: $\begin{Bmatrix} \triangle\boldsymbol{x}_B^{(1)} \\ \triangle\boldsymbol{x}_B^{(2)} \end{Bmatrix} \leftarrow \begin{pmatrix} \boldsymbol{K}_{AB}^{(1)} + \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{11} & \boldsymbol{K}_{A\beta}^{(1)}\boldsymbol{H}_{12} \\ \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{21} & \boldsymbol{K}_{AB}^{(2)} + \boldsymbol{K}_{A\beta}^{(2)}\boldsymbol{H}_{22} \end{pmatrix} \backslash \begin{Bmatrix} -\boldsymbol{R}_A^{(1)} \\ -\boldsymbol{R}_A^{(2)} \end{Bmatrix}$ ▷ linear system

5: $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$

6: $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$

7: **until** $\left[\left(||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Advantages:

- By-passes Schwarz loop.

Disadvantages:

- Off-diagonal coupling terms → block linear solver is needed.

Appendix. Modified Schwarz Method

Combines Schwarz and Newton loop into ***since Newton-Schwarz loop***, with ***Schwarz boundaries*** at ***Dirichlet boundaries*** and tight convergence tolerance.

1: $\boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{X}_B^{(1)}$ in Ω_1, $\boldsymbol{x}_b^{(1)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(1)})$ on $\partial_\varphi\Omega_1$, $\boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{X}_\beta^{(1)}$ on Γ_1 ▷ initialize for Ω_1
2: $\boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{X}_B^{(2)}$ in Ω_2, $\boldsymbol{x}_b^{(2)} \leftarrow \boldsymbol{\chi}(\boldsymbol{X}_b^{(2)})$ on $\partial_\varphi\Omega_2$, $\boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{X}_\beta^{(2)}$ on Γ_2 ▷ initialize for Ω_2
3: **repeat** ▷ Newton-Schwarz loop
4: $\quad \boldsymbol{x}_\beta^{(1)} \leftarrow \boldsymbol{P}_{12}\boldsymbol{x}_B^{(2)} + \boldsymbol{Q}_{12}\boldsymbol{x}_b^{(2)} + \boldsymbol{G}_{12}\boldsymbol{x}_\beta^{(2)}$ ▷ project from Ω_2 to Γ_1
5: $\quad \triangle\boldsymbol{x}_B^{(1)} \leftarrow -\boldsymbol{K}_{AB}^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})\backslash\boldsymbol{R}_A^{(1)}(\boldsymbol{x}_B^{(1)};\boldsymbol{x}_b^{(1)};\boldsymbol{x}_\beta^{(1)})$ ▷ linear system
6: $\quad \boldsymbol{x}_B^{(1)} \leftarrow \boldsymbol{x}_B^{(1)} + \triangle\boldsymbol{x}_B^{(1)}$
7: $\quad \boldsymbol{x}_\beta^{(2)} \leftarrow \boldsymbol{P}_{21}\boldsymbol{x}_B^{(1)} + \boldsymbol{Q}_{21}\boldsymbol{x}_b^{(1)} + \boldsymbol{G}_{21}\boldsymbol{x}_\beta^{(1)}$ ▷ project from Ω_1 to Γ_2
8: $\quad \triangle\boldsymbol{x}_B^{(2)} \leftarrow -\boldsymbol{K}_{AB}^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})\backslash\boldsymbol{R}_A^{(2)}(\boldsymbol{x}_B^{(2)};\boldsymbol{x}_b^{(2)};\boldsymbol{x}_\beta^{(2)})$ ▷ linear system
9: $\quad \boldsymbol{x}_B^{(2)} \leftarrow \boldsymbol{x}_B^{(2)} + \triangle\boldsymbol{x}_B^{(2)}$
10: **until** $\left[\left(||\triangle\boldsymbol{x}_B^{(1)}||/||\boldsymbol{x}_B^{(1)}||\right)^2 + \left(||\triangle\boldsymbol{x}_B^{(2)}||/||\boldsymbol{x}_B^{(2)}||\right)^2\right]^{1/2} \leq \epsilon_{\text{machine}}$ ▷ tight tolerance

Advantages:

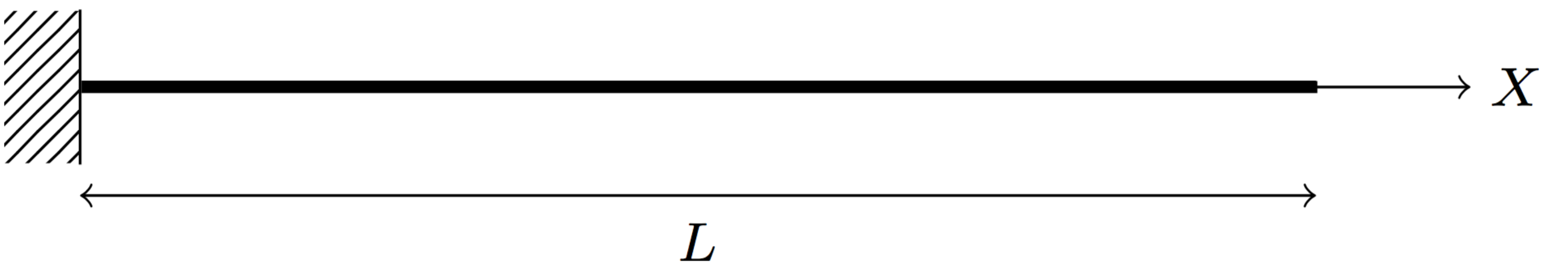
- By-passes Schwarz loop.
- No diagonal coupling (conventional linear solver can be used in each subdomain).

Least-intrusive variant: by-passes Schwarz iteration, no need for block solver.

Appendix. Foulk's Singular Bar

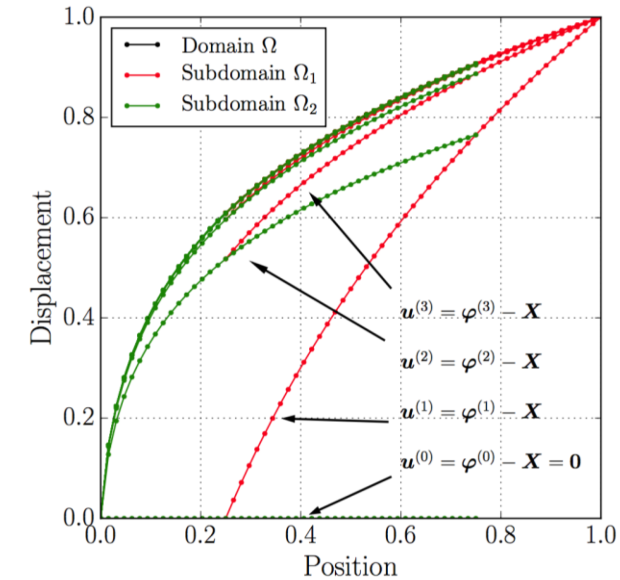
- ***1D proof of concept*** problem:
  - ***1D bar*** with area proportional to square root of length.
  - Strong ***singularity*** on left end of bar.
  - Simple ***hyperelestic*** material model with no damage.
  - ***MATLAB*** implementation.

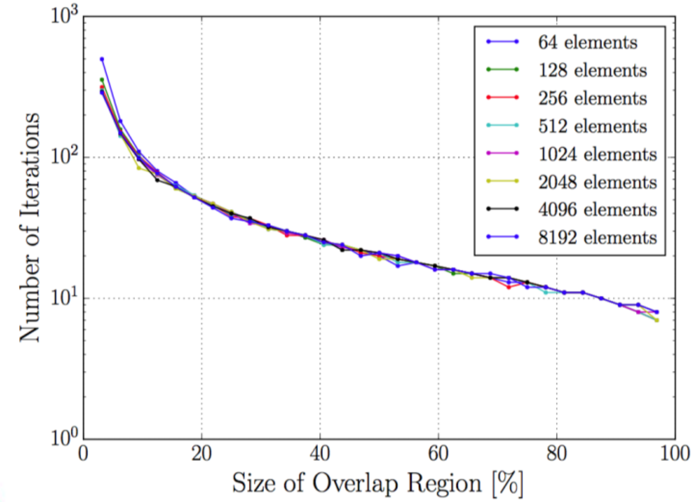
$u(0) = 0$ $\qquad A(X) = A_0\sqrt{X/L}$ $\qquad u(L) = \Delta$

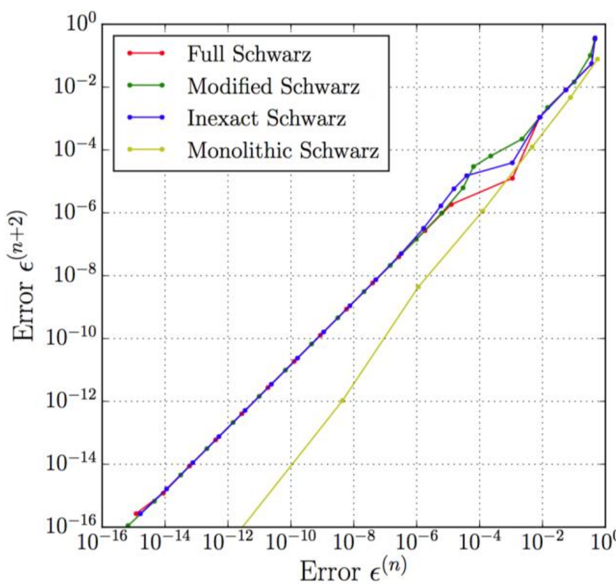


- ***Problem goals***:
  - Explore ***viability*** of ***4 variants*** of the Schwarz alternating method.
  - Test ***convergence*** and compare with literature (Evans, 1986).
    - Expect ***faster convergence*** in ***fewer iterations*** with ***increased overlap***.

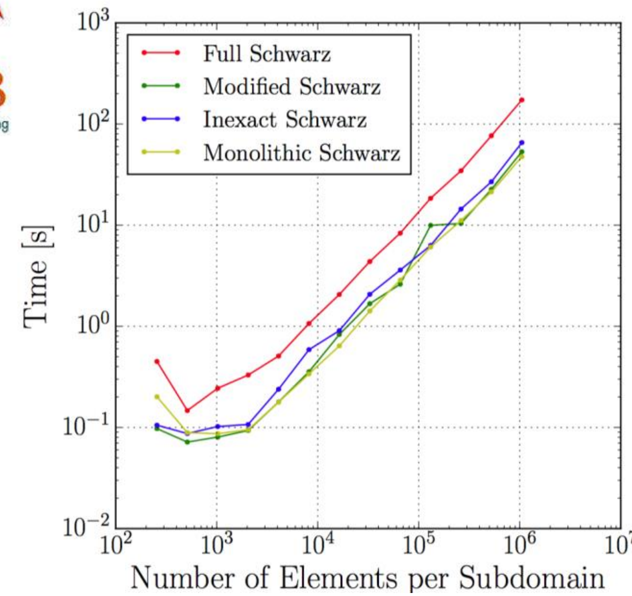
Appendix. Singular Bar and Schwarz Variants



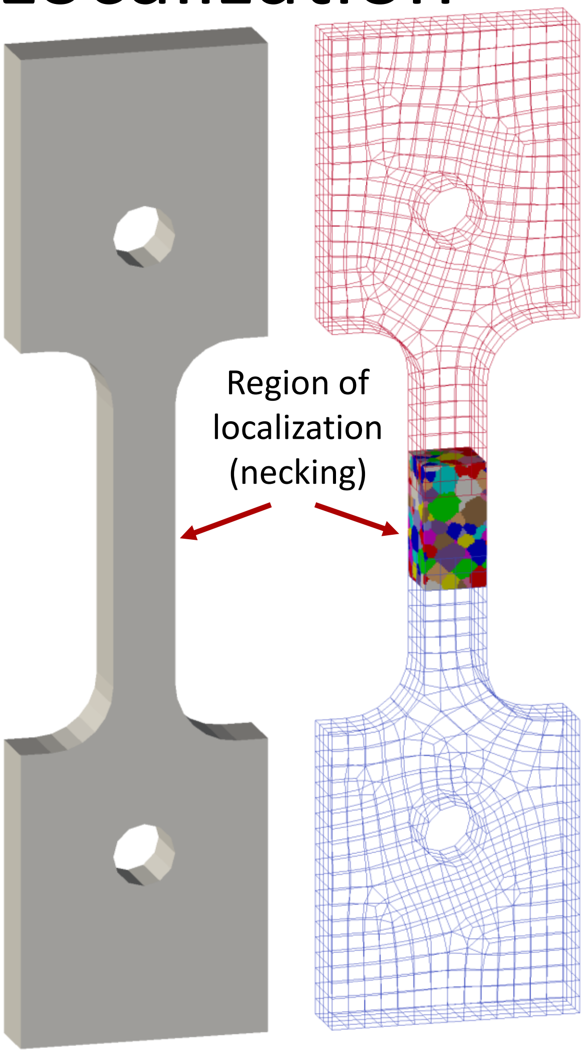


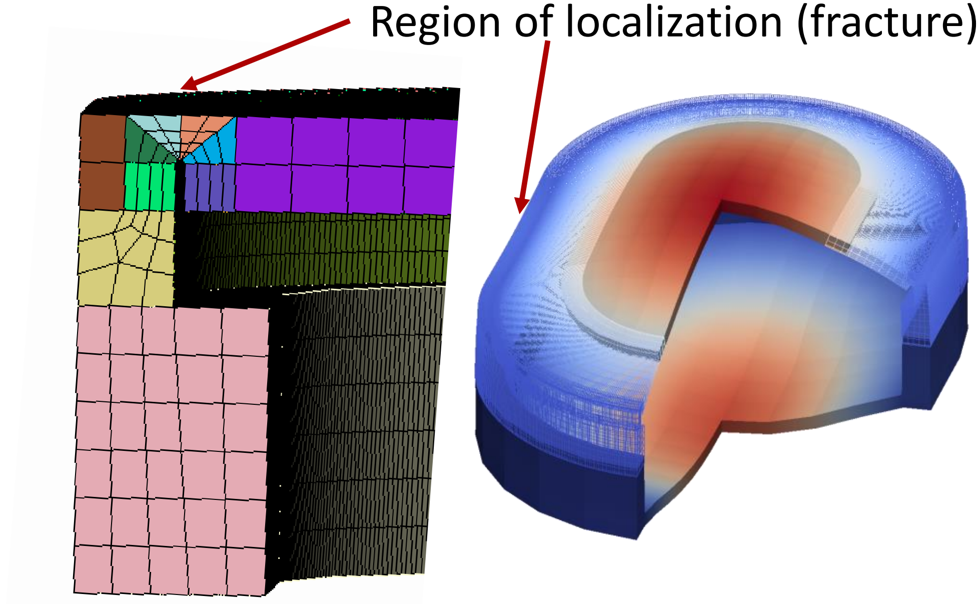


MATLAB
The Language of Technical Computing



Appendix. Multiscale Modeling of Localization



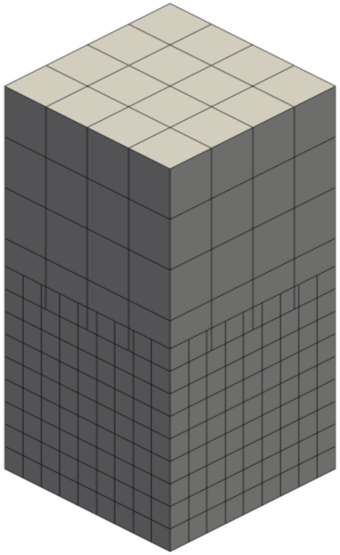


Strain localization can cause ***localized necking*** (left) and ultimately ***fracture*** (above).

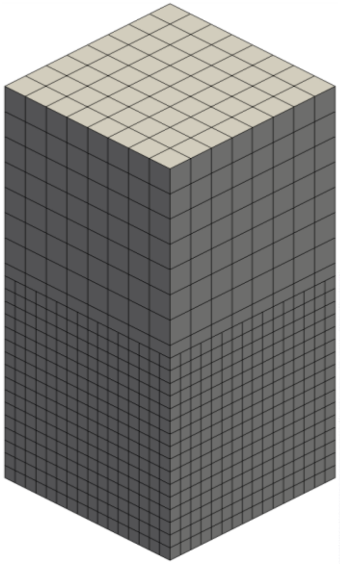
Goals:

- Connect ***physical length scales*** to ***engineering scale models.***
- Investigate importance of ***microstructural detail.***
- Develop bridging technologies for ***spatial multiscale/ multiphysics.***

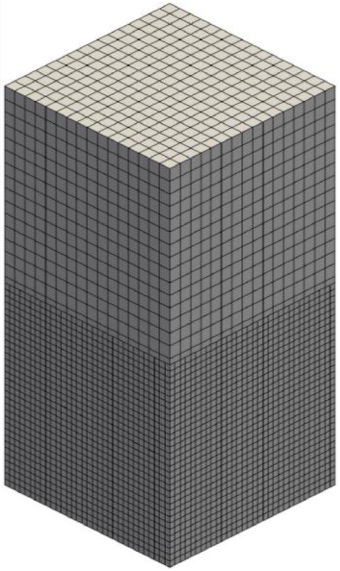
Appendix. Parallelization via DTK: Weak Scaling on Cubes Problem



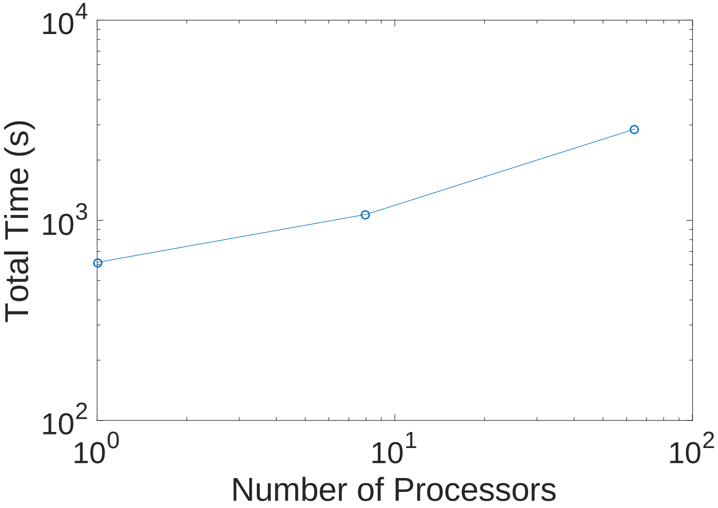
1 Processor,
2.5*10^3 DOF / proc



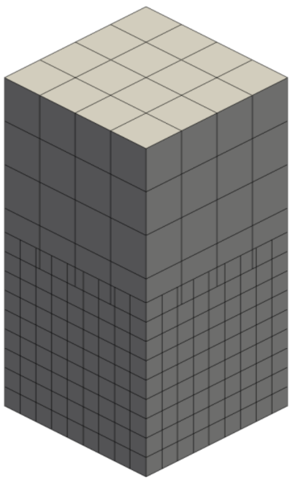
8 Processors,
2.1*10^3 DOF / proc



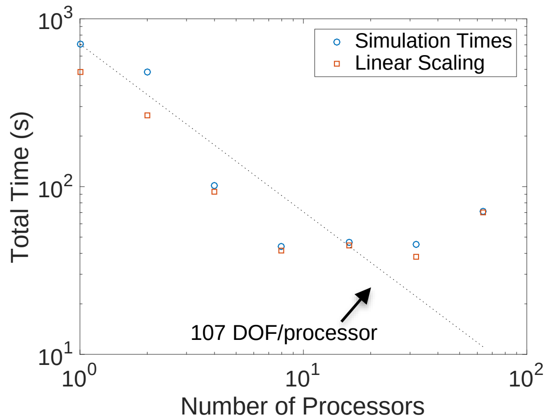
64 Processors,
1.9*10^3 DOF / proc



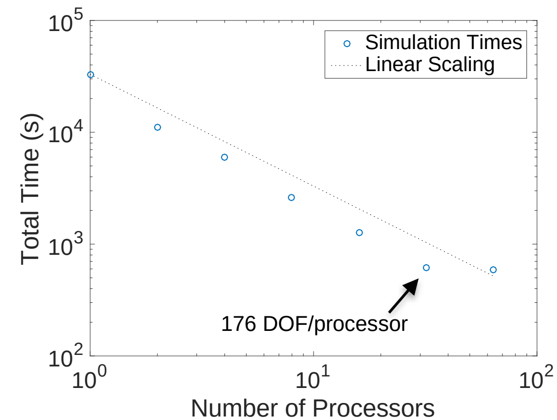
Appendix. Parallelization via DTK: Strong Scaling on Cubes Problem

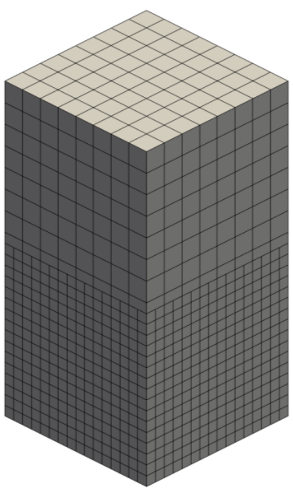


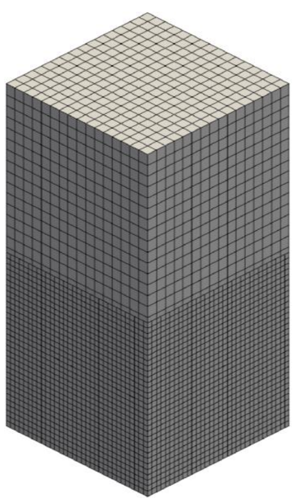
Small problem ($2.5*10^3$ DOFs)



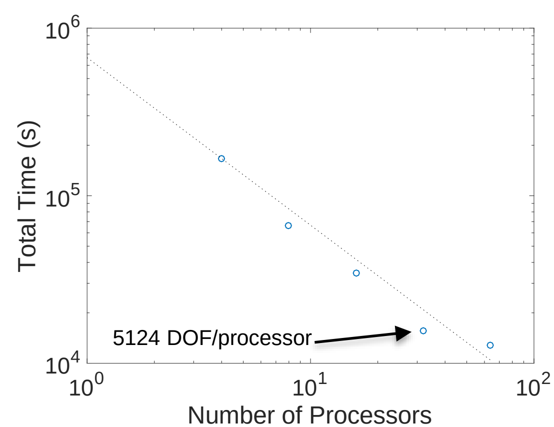
Medium problem ($1.7*10^4$ DOFs)



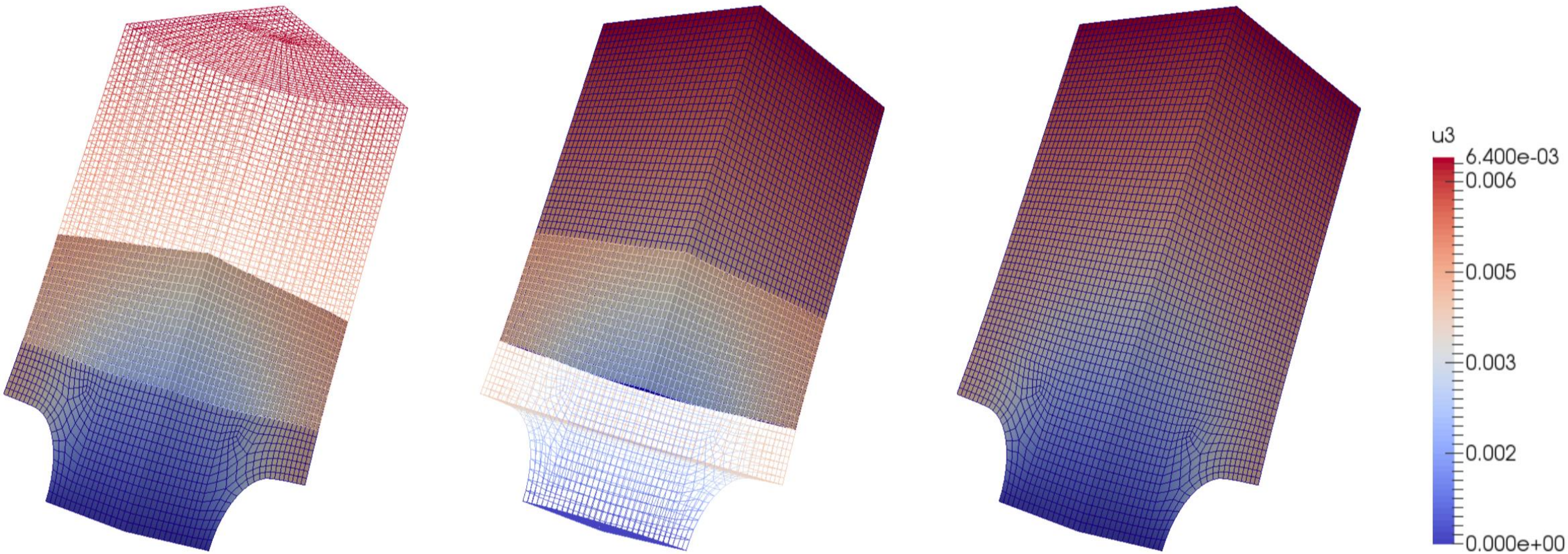




Large problem ($1.6*10^5$ DOFs)



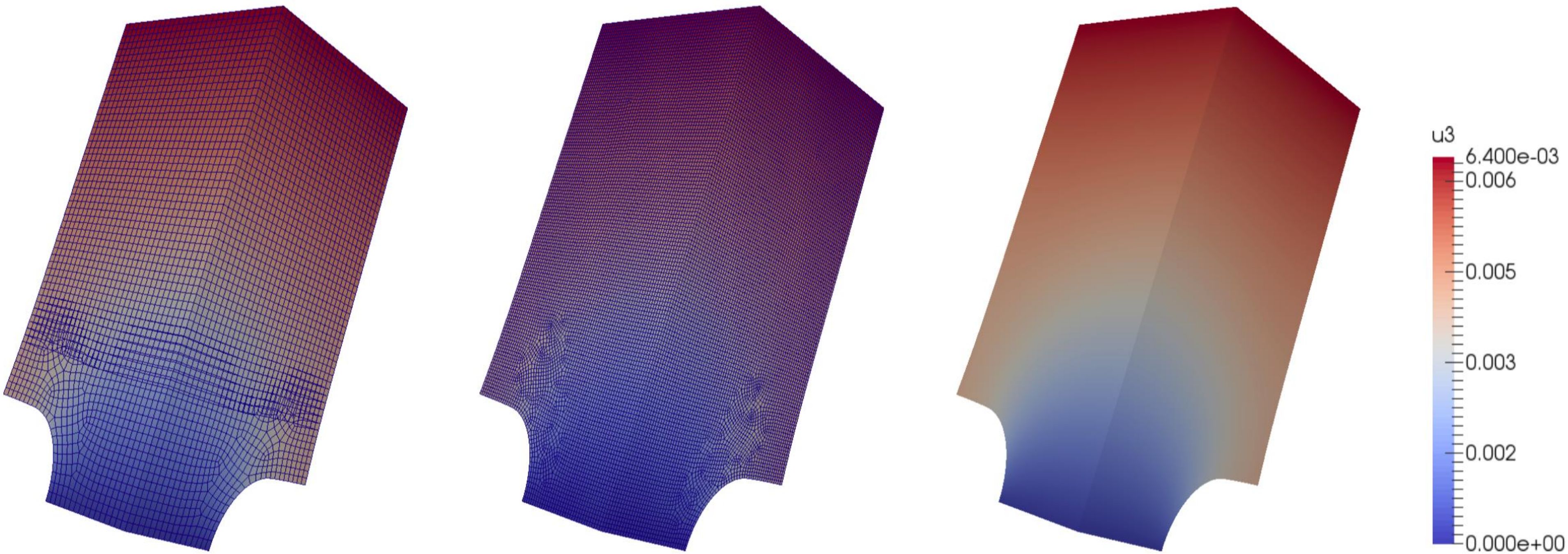
Appendix. Notched Cylinder: HEX-HEX Coupling



(a) Ω_1 **(b)** Ω_2 **(c)** Ω_{ref}

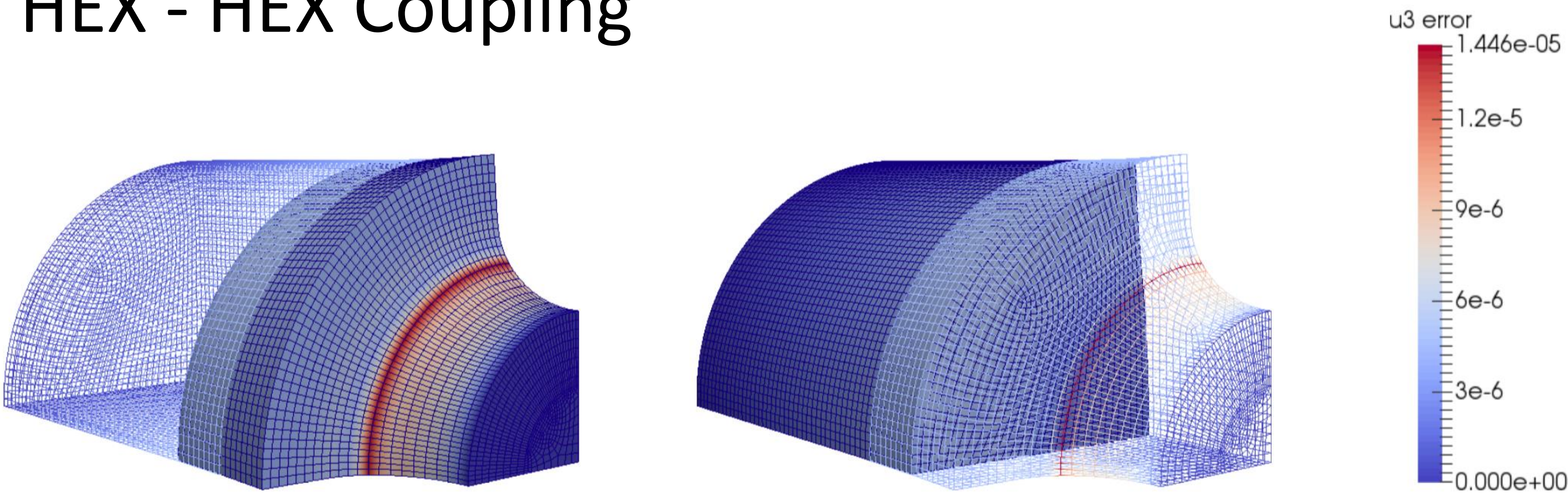
	u_3 relative error	
Absolute residual tolerance	Ω_1	Ω_2
1.0×10^{-4}	7.60×10^{-3}	3.20×10^{-3}
1.0×10^{-8}	3.10×10^{-5}	1.71×10^{-5}
1.0×10^{-12}	1.34×10^{-9}	5.10×10^{-10}
1.0×10^{-14}	1.23×10^{-11}	4.69×10^{-12}
2.5×10^{-16}	1.14×10^{-13}	8.37×10^{-14}

Appendix. Notched Cylinder: Nonconformal HEX - HEX Coupling



(a) Ω_1 and Ω_2 **(b)** Ω_{ref} mesh **(c)** Ω_{ref} solution

Appendix. Notched Cylinder: Nonconformal HEX - HEX Coupling



(a) Ω_1 **(b)** Ω_2

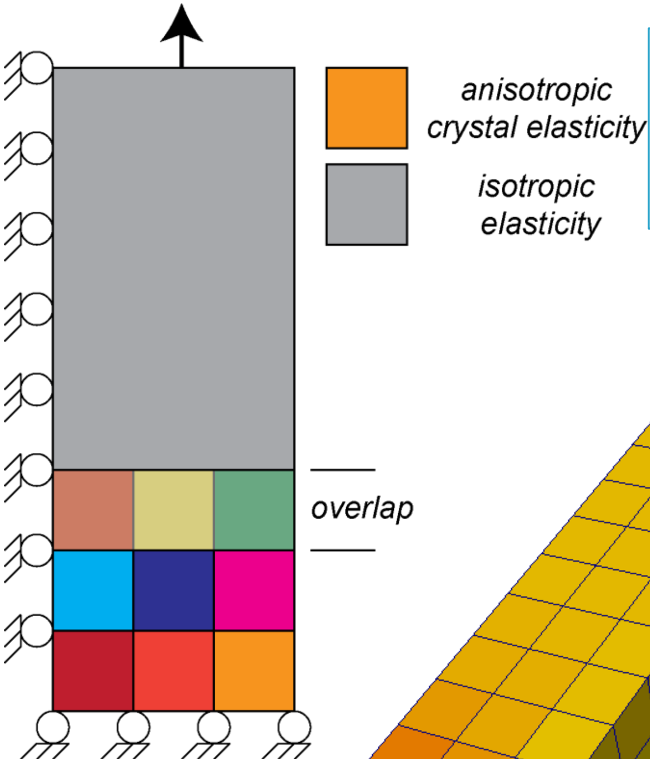
Absolute residual tolerance	u_3 relative error Ω_1	Ω_2
1.0×10^{-8}	1.31×10^{-3}	4.45×10^{-4}
1.0×10^{-12}	1.30×10^{-3}	4.43×10^{-4}
1.0×10^{-14}	1.30×10^{-3}	4.43×10^{-4}
2.5×10^{-16}	1.30×10^{-3}	4.43×10^{-4}

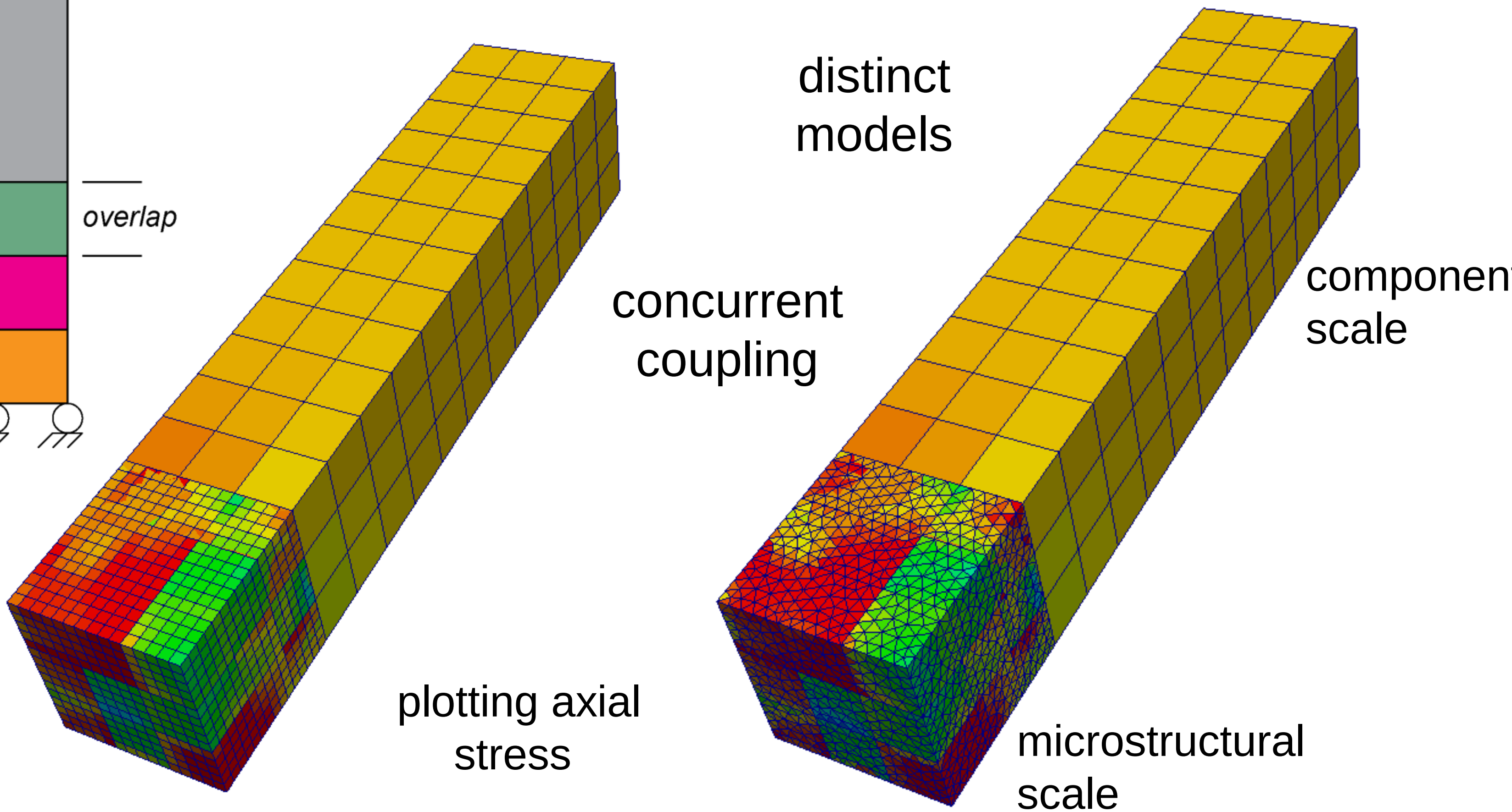
Albany

Appendix. Rubiks Cube Problem

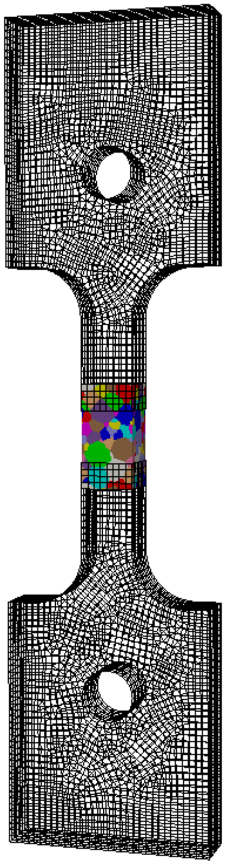
Work by J. Foulk, D. Littlewood, C. Battaile, H. Lim

Two distinct bodies, the component scale and the microstructural scale, are coupled iteratively with alternating Schwarz

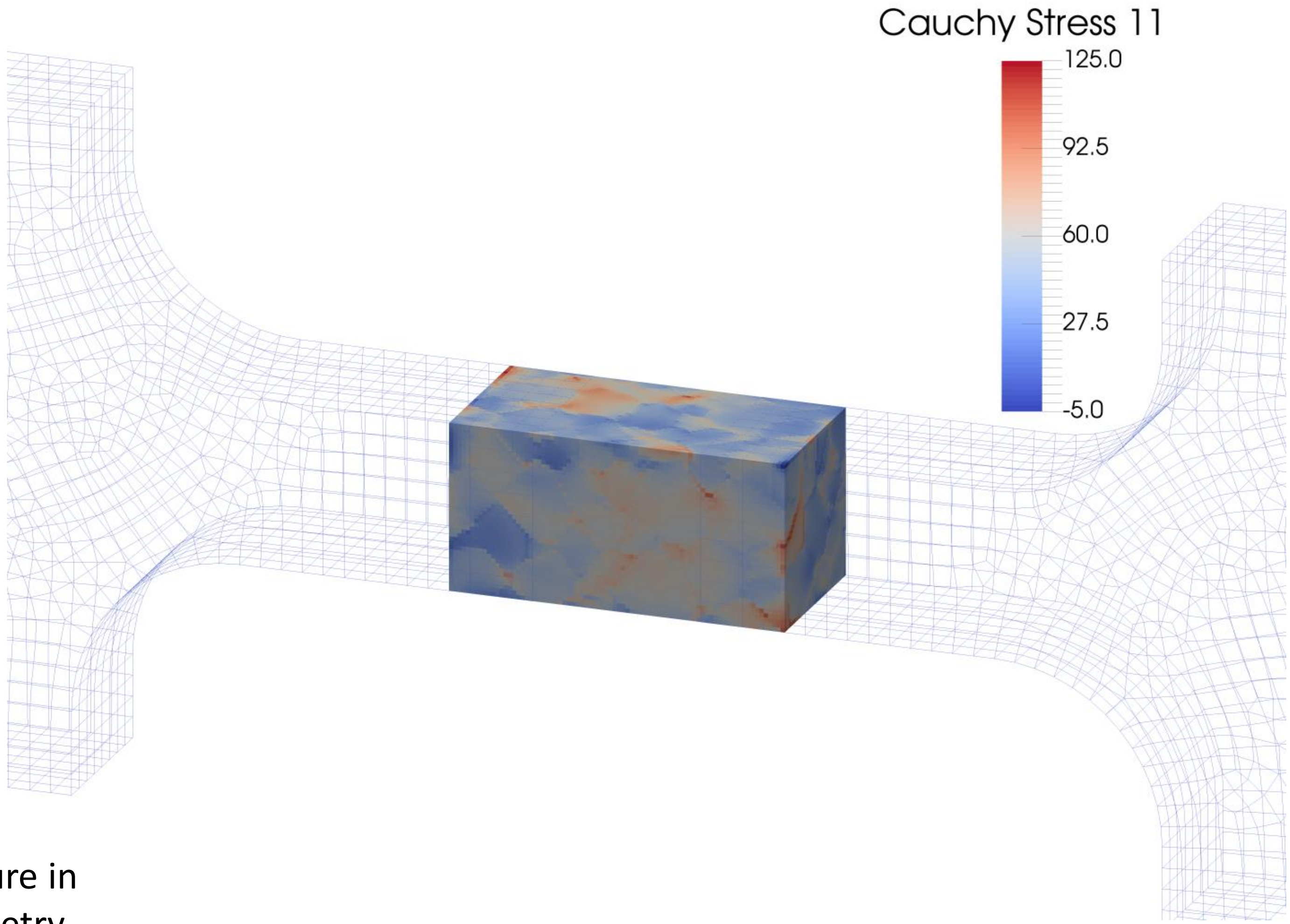




Appendix. Tensile Bar



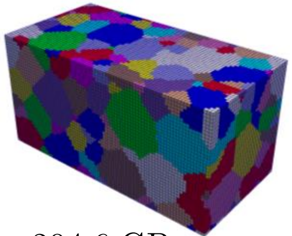
Embed microstructure in ASTM tensile geometry



Appendix. Tensile Bar: Meso-Macroscale Coupling

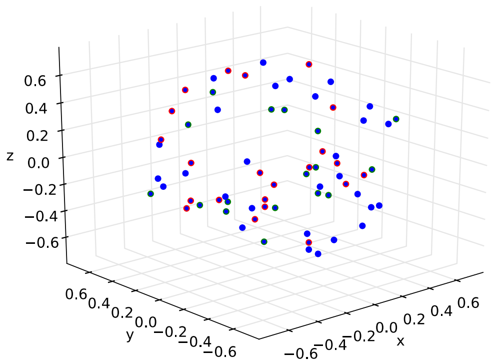
Mesoscale

SPARKS-generated microstructure (F. Abdeljawad)



$$\text{cubic elastic constant}: C_{11} = 204.6\ \text{GPa}$$
$$\text{cubic elastic constant}: C_{12} = 137.7\ \text{GPa}$$
$$\text{cubic elastic constant}: C_{44} = 126.2\ \text{GPa}$$
$$\text{reference shear rate}: \dot{\gamma}_0 = 1.0\ 1/\text{s}$$
$$\text{rate sensitivity factor}: m = 20$$
$$\text{hardening rate parameter}: \dot{g}_0 = 2.0 \times 10^4\ 1/\text{s}$$
$$\text{initial hardness}: g_0 = 90\ \text{MPa}$$
$$\text{saturation hardness}: g_s = 202\ \text{MPa}$$
$$\text{saturation exponent}: \omega = 0.01$$

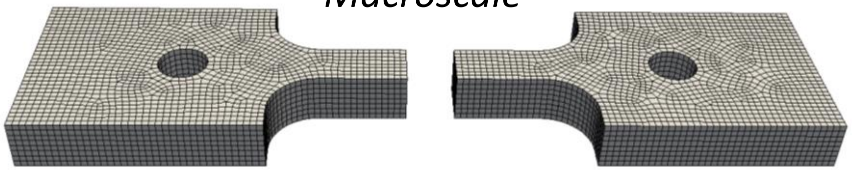
Fix microstructure, investigate ensembles



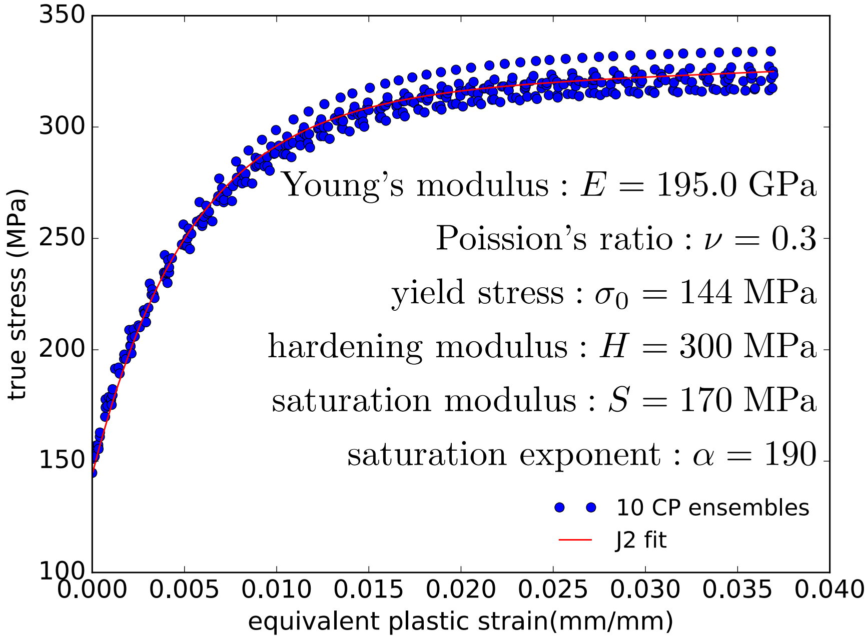
151 axial vectors from 3 of the 10 ensembles of random rotations (blue, green, red)

+

Macroscale



- Load microstructural ensembles in uniaxial stress
- Fit flow curves with a macroscale J_2 plasticity model

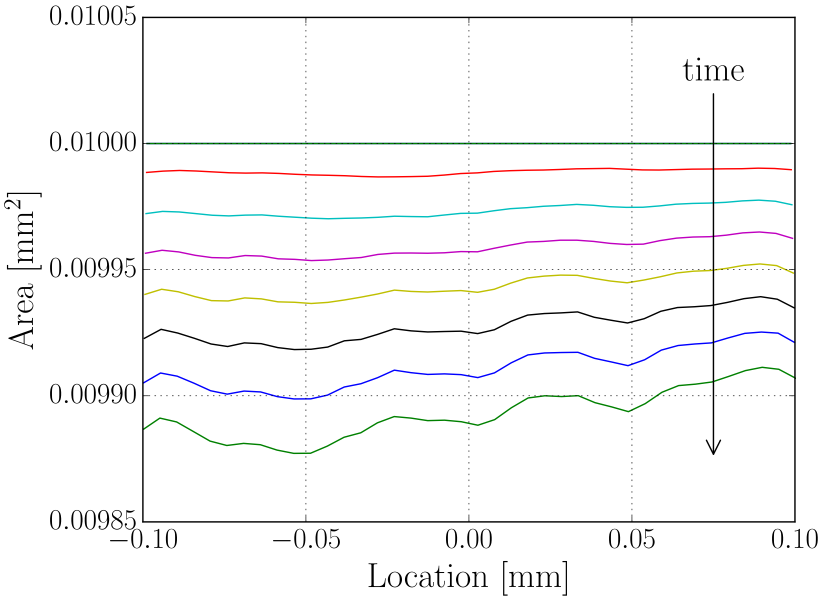


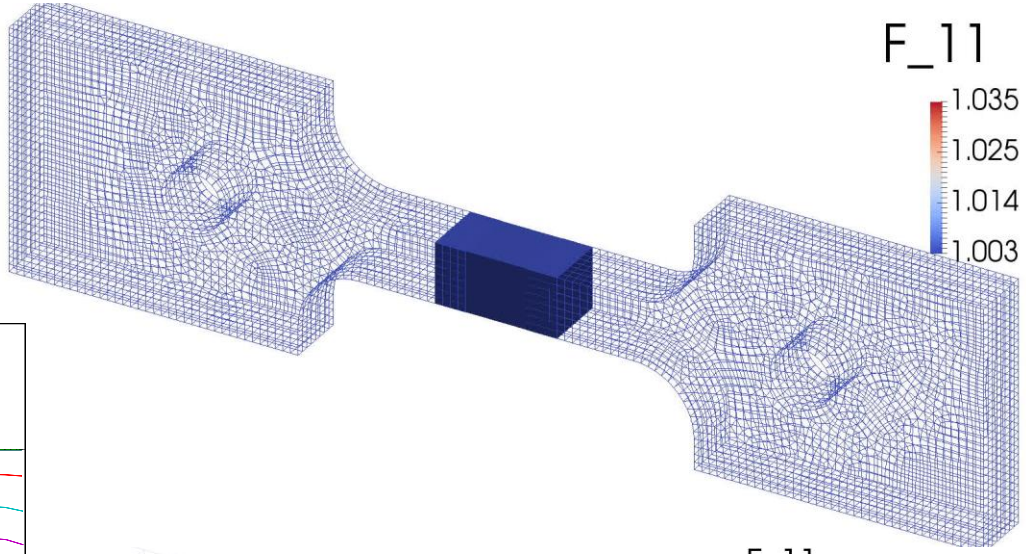
$$\text{Young's modulus}: E = 195.0\ \text{GPa}$$
$$\text{Poission's ratio}: \nu = 0.3$$
$$\text{yield stress}: \sigma_0 = 144\ \text{MPa}$$
$$\text{hardening modulus}: H = 300\ \text{MPa}$$
$$\text{saturation modulus}: S = 170\ \text{MPa}$$
$$\text{saturation exponent}: \alpha = 190$$

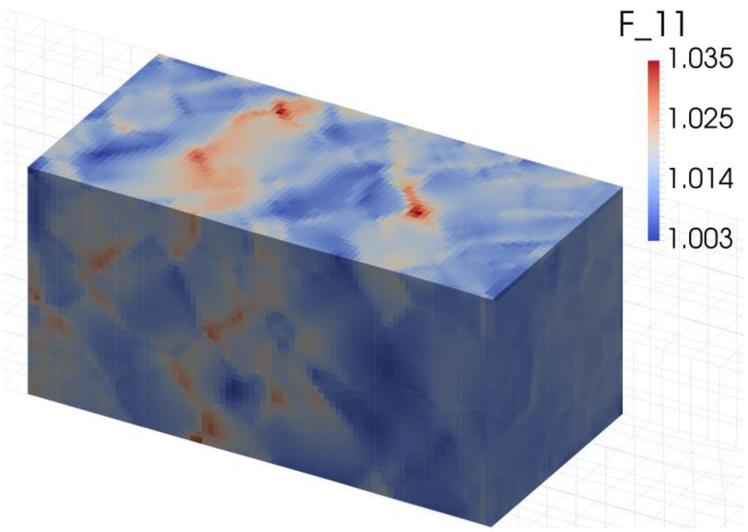
$$\sigma_y = \sigma_0 + H\epsilon_p + S(1 - e^{-\alpha\epsilon_p})$$

Appendix. Tensile Bar: Results

Reduction in cross-sectional area over time

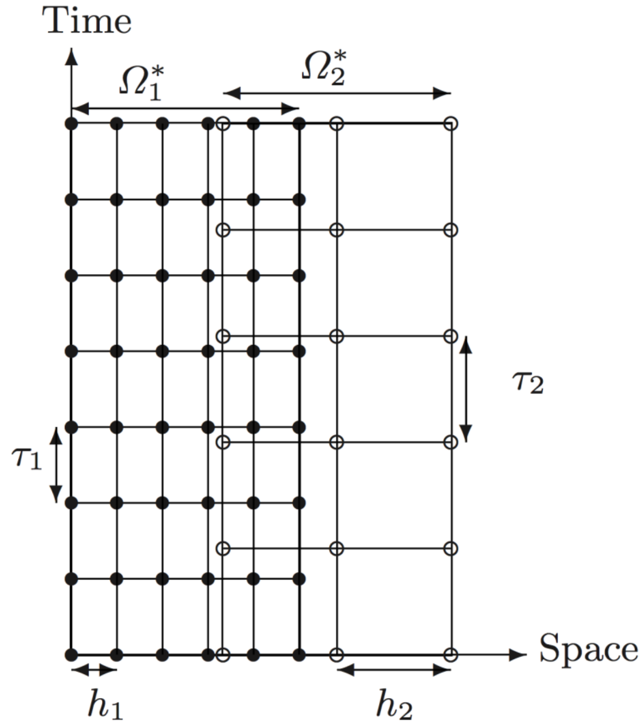






Appendix. Schwarz Alternating Method for Dynamics

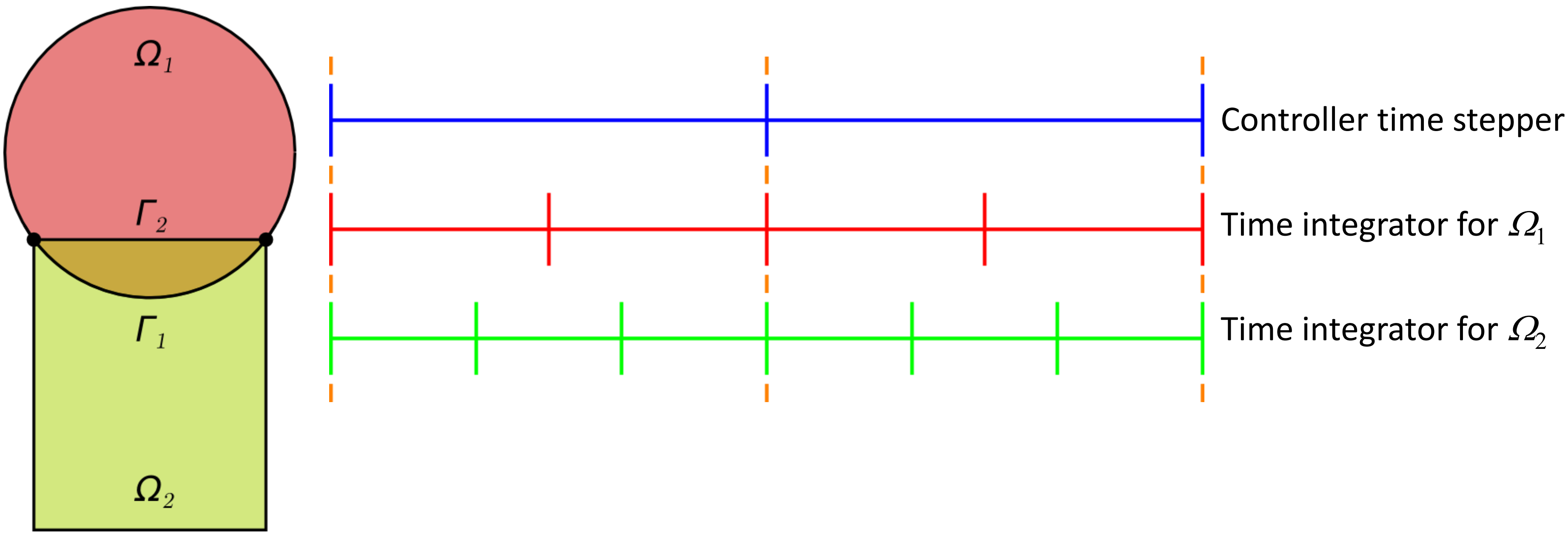
- In the literature the Schwarz method is applied to dynamics by using ***space-time discretizations***.
- This was deemed ***unfeasible*** given the design of our current codes and size of simulations.



Overlapping non-matching meshes and time steps in dynamics.

Appendix. A Schwarz-like Time Integrator

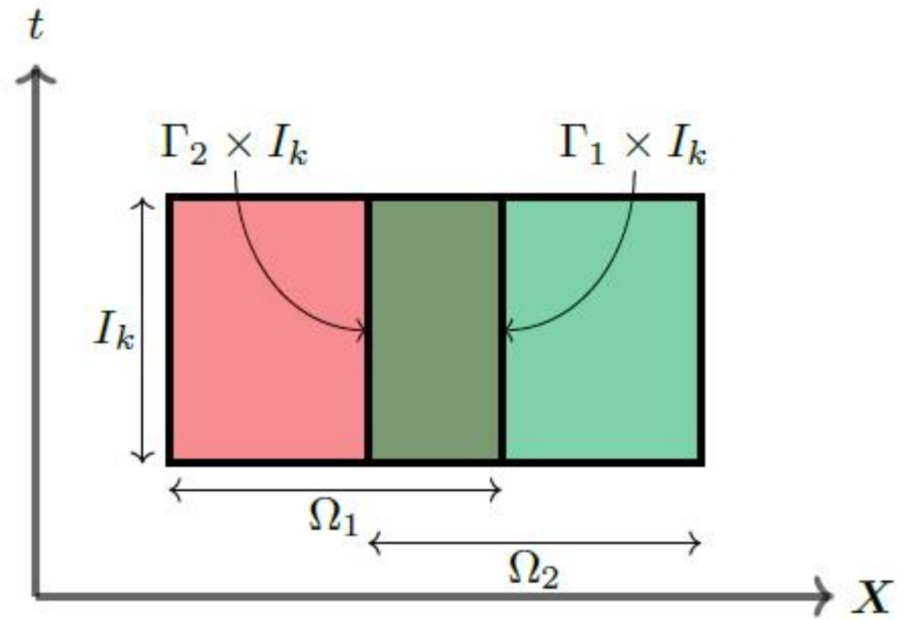
- We developed an ***extension of Schwarz coupling*** to ***dynamics*** using a governing time stepping algorithm that controls time integrators within each domain.
- Can use ***different integrators*** with ***different time steps*** within each domain.
- 1D results show ***smooth coupling without numerical artifacts*** such as spurious wave reflections at boundaries of coupled domains.



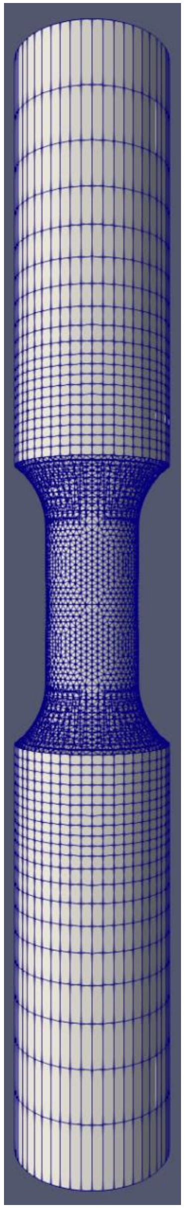
Schwarz Alternating Method for Dynamic Multiscale Coupling: Theory

- Our dynamic alternating Schwarz algorithm can be ***interpreted*** as applying the traditional Schwarz alternating iterations in ***space-time*** between $\Omega_1 \times I_k$ and $\Omega_2 \times I_k$

- This interpretation ***does not*** require the method to be implemented using a space-time framework.



Appendix. Sierra/SM: Dynamic Schwarz Coupling

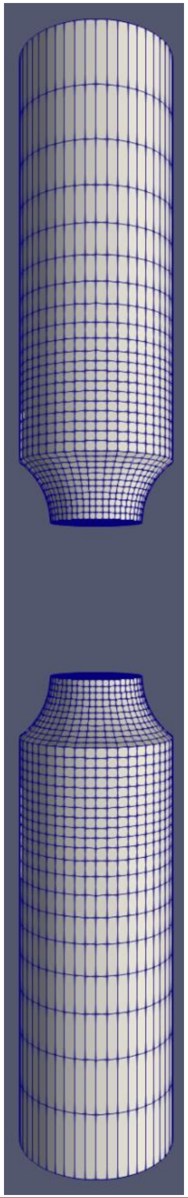


```
begin sierra schwarz_tension

  begin material aluminum
    density = 2700.0
    begin parameters for model fefp
      youngs modulus = 70.0e+09
      poissons ratio = 0.36
      yield stress = 250.0e+06
      hardening model   = linear
      hardening modulus = 0.7e+09
    end
  end

  begin function applied_disp_top_ramp
    type = analytic
    evaluate expression = " 0.01 * cos_ramp(x, 0.0, 1.0)"
  end

  begin function applied_disp_bottom_ramp
    type = analytic
    evaluate expression = "-0.01 * cos_ramp(x, 0.0, 1.0)"
  end

  begin total lagrange section gauge_formulation
    strain incrementation = logarithmic_mapping
    cubature degree = 3
    volume average J = on
    element jacobian = element_probe
  end

  begin total lagrange section ends_formulation
    strain incrementation = logarithmic_mapping
    cubature degree = 3
    volume average J = on
    element jacobian = element_probe
  end

  begin finite element model gauge_mesh
    database name = gauge.g
    database type = exodusII
    begin parameters for block gauge
      material = aluminum
      model = fefp
      section = gauge_formulation
    end
  end

  begin finite element model ends_mesh
    database name = ends.g
    database type = exodusII
    begin parameters for block ends
      material = aluminum
      model = fefp
      section = ends_formulation
    end
  end

  {include("schwarz_funcs.i")}
```

tension.i

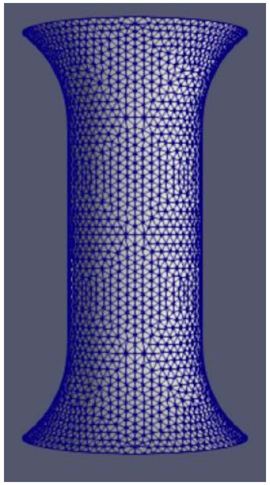
```
  begin procedure schwarz_procedure

    begin solution control description schwarz_loop
      use system main
      begin system main
        begin transient schwarz_solve
          begin nonlinear schwarz_iteration
            advance ends_region
            transfer from_ends_to_gauge
            advance gauge_region
            transfer from_gauge_to_ends
          end nonlinear schwarz_iteration
        end transient schwarz_solve
      end system main

      begin parameters for transient schwarz_solve
        start time = 0.0
        termination time = 1.0
        begin parameters for adagio region ends_region
          time increment = 0.01
        end
        begin parameters for adagio region gauge_region
          time increment = 0.01
        end
      end

      begin parameters for nonlinear schwarz_iteration
        converged when "gauge_region.schwarz_norm < 1.0e-06 && ends_region.schwarz_norm < 1.0e-06"
      end

    end solution control description schwarz_loop

    begin transfer from_ends_to_gauge
      interpolate volume nodes from ends_region to gauge_region
      send block ends to schwarz_upper_transition schwarz_lower_transition
      send field displacement state new to schwarz_disp state none
      send field velocity state new to velocity state new
      send field acceleration state new to acceleration state new
    end

    begin transfer from_gauge_to_ends
      interpolate volume nodes from gauge_region to ends_region
      send block gauge to schwarz_upper_gauge schwarz_lower_gauge
      send field displacement state new to schwarz_disp state none
      send field velocity state new to velocity state new
      send field acceleration state new to acceleration state new
    end

    {include("gauge.i")}
    {include("ends.i")}

  end procedure schwarz_procedure

  begin feti equation solver feti
    residual norm tolerance = 1e-3
    maximum orthogonalization = 5000
  end

end sierra schwarz_tension
```

Appendix. Sierra/SM: Dynamic Schwarz Coupling



```
begin adagio region ends_region
  use finite element model ends_mesh

  begin implicit dynamics
  end

  begin adaptive time stepping
    cutback factor = 0.5
    growth factor = 1.1
    maximum failure cutbacks = 16
  end

  {include("schwarz_vars.i")}

  begin results output
    database name = ends.e
    database type = exodusII
    at time 0, increment = 0.01
    element variables = stress as stress
    element variables = cauchy_stress
    element variables = log_strain
    element variables = eqps
    element variables = von_mises
    nodal variables = displacement
    nodal variables = velocity
    nodal variables = acceleration
    nodal variables = coordinates
    global variables = timestep
  end

  {include("solver.i")}

  begin fixed displacement
    node set = upper_grip
    component = x
  end

  begin fixed displacement
    node set = upper_grip
    component = z
  end

  begin prescribed displacement
    node set = upper_grip
    function = applied_disp_top_ramp
    component = y
  end

  begin fixed displacement
    node set = lower_grip
    component = x
  end

  begin fixed displacement
    node set = lower_grip
    component = z
  end

  begin prescribed displacement
    node set = lower_grip
    function = applied_disp_bottom_ramp
    component = y
  end

  begin prescribed displacement schwarz_bc_x
    node set = schwarz_upper_gauge schwarz_lower_gauge
    component = x
    function = schwarz_disp_x
  end

  begin prescribed displacement schwarz_bc_y
    node set = schwarz_upper_gauge schwarz_lower_gauge
    component = y
    function = schwarz_disp_y
  end

  begin prescribed displacement schwarz_bc_z
    node set = schwarz_upper_gauge schwarz_lower_gauge
    component = z
    function = schwarz_disp_z
  end

end adagio region ends_region
```

ends.i



```
begin adagio region gauge_region
  use finite element model gauge_mesh

  begin implicit dynamics
  end

  begin adaptive time stepping
    cutback factor = 0.5
    growth factor = 1.1
    maximum failure cutbacks = 16
  end

  {include("schwarz_vars.i")}

  begin results output
    database name = gauge.e
    database type = exodusII
    at time 0, increment = 0.01
    element variables = stress as stress
    element variables = cauchy_stress
    element variables = log_strain
    element variables = eqps
    element variables = von_mises
    nodal variables = displacement
    nodal variables = velocity
    nodal variables = acceleration
    nodal variables = coordinates
    global variables = timestep
  end

  {include("solver.i")}

  begin prescribed displacement schwarz_bc_x
    node set = schwarz_upper_transition schwarz_lower_transition
    component = x
    function = schwarz_disp_x
  end

  begin prescribed displacement schwarz_bc_y
    node set = schwarz_upper_transition schwarz_lower_transition
    component = y
    function = schwarz_disp_y
  end

  begin prescribed displacement schwarz_bc_z
    node set = schwarz_upper_transition schwarz_lower_transition
    component = z
    function = schwarz_disp_z
  end

end adagio region gauge_region
```

gauge.i

Appendix. Sierra/SM: Dynamic Schwarz Coupling

schwarz_funcs.i

```
#
# Functions for transfer of Schwarz BCs
#
begin function schwarz_disp_x
  type = analytic
  expression variable: dx = nodal schwarz_disp(x)
  evaluate expression = "dx"
end

begin function schwarz_disp_y
  type = analytic
  expression variable: dy = nodal schwarz_disp(y)
  evaluate expression = "dy"
end

begin function schwarz_disp_z
  type = analytic
  expression variable: dz = nodal schwarz_disp(z)
  evaluate expression = "dz"
end

#
# Define functions for building the Schwarz convergence criterion
#
begin function x_prev
  type = analytic
  expression variable: x_prev = nodal coordinates(x)
  evaluate expression = "x_prev"
end

begin function y_prev
  type = analytic
  expression variable: y_prev = nodal coordinates(y)
  evaluate expression = "y_prev"
end

begin function z_prev
  type = analytic
  expression variable: z_prev = nodal coordinates(z)
  evaluate expression = "z_prev"
end

begin function delta_x
  type = analytic
  expression variable: x_curr = nodal coordinates(x)
  expression variable: x_prev = nodal x_prev
  evaluate expression = "x_curr - x_prev"
end

begin function delta_y
  type = analytic
  expression variable: y_curr = nodal coordinates(y)
  expression variable: y_prev = nodal y_prev
  evaluate expression = "y_curr - y_prev"
end

begin function delta_z
  type = analytic
  expression variable: z_curr = nodal coordinates(z)
  expression variable: z_prev = nodal z_prev
  evaluate expression = "z_curr - z_prev"
end

begin function schwarz_norm
  type = analytic
  expression variable: xn = global curr_x_norm
  expression variable: yn = global curr_y_norm
  expression variable: zn = global curr_z_norm
  expression variable: dx = global delta_x_norm
  expression variable: dy = global delta_y_norm
  expression variable: dz = global delta_z_norm
  evaluate expression = "sqrt(dx*dx+dy*dy+dz*dz)/sqrt(xn*xn+yn*yn+zn*zn)"
end
```

schwarz_vars.i

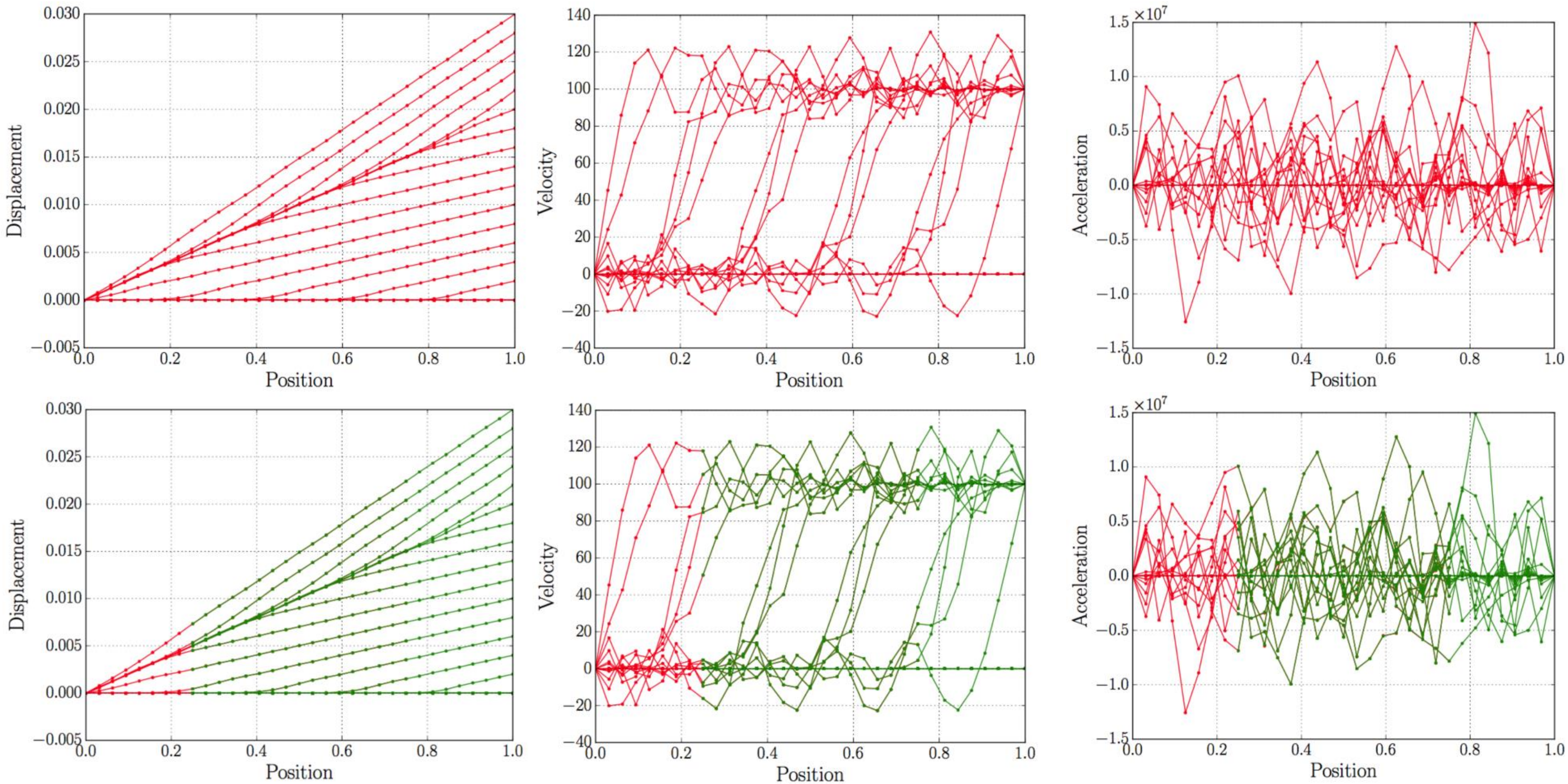
```
begin user variable schwarz_disp
  type = node vector
end

begin user output
  compute nodal delta_x as function delta_x
  compute nodal delta_y as function delta_y
  compute nodal delta_z as function delta_z
  compute global delta_x_norm as l2norm of nodal delta_x
  compute global delta_y_norm as l2norm of nodal delta_y
  compute global delta_z_norm as l2norm of nodal delta_z
  compute global curr_x_norm as l2norm of nodal coordinates(x)
  compute global curr_y_norm as l2norm of nodal coordinates(y)
  compute global curr_z_norm as l2norm of nodal coordinates(z)
  compute global schwarz_norm as function schwarz_norm
  compute at every step
end

begin user output
  compute nodal x_prev as function x_prev
  compute nodal y_prev as function y_prev
  compute nodal z_prev as function z_prev
  compute at every step
end
```

solver.i

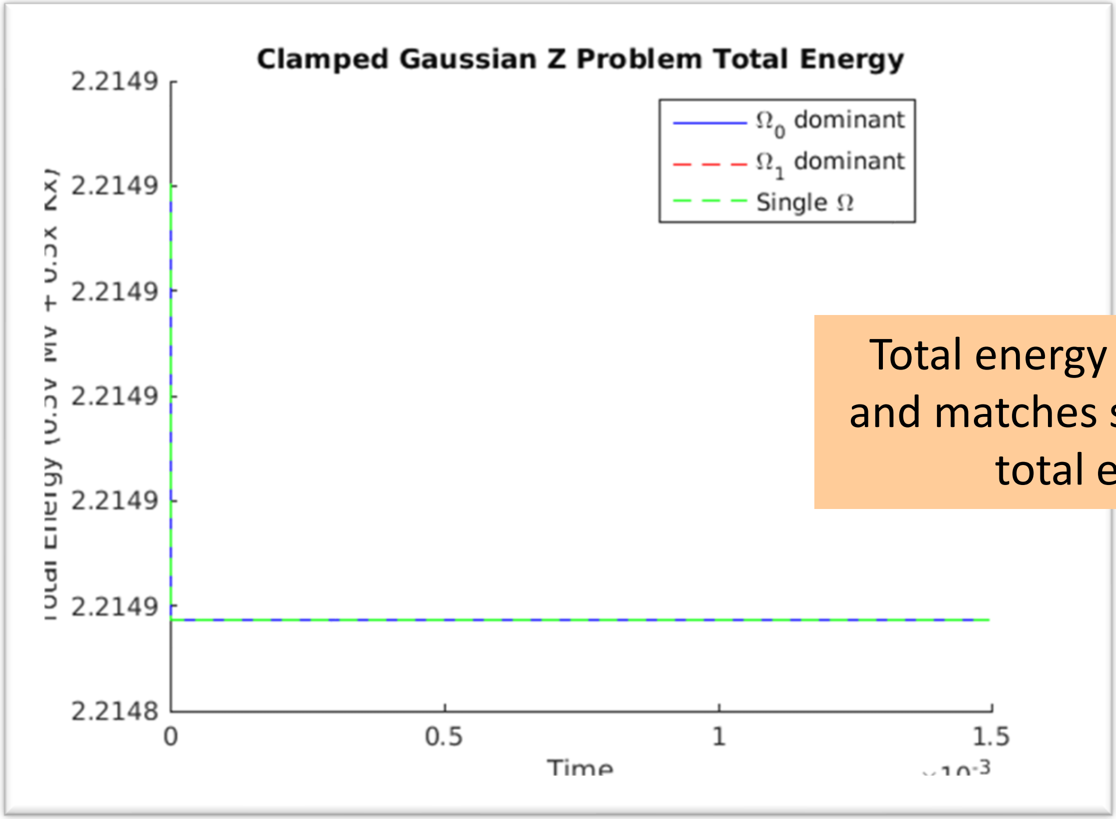
```
${maximum_iterations = 150}
${tangent_iterations = 125}
${smooth__iterations = maximum_iterations - tangent_iterations}

begin solver

  begin cg
    iteration print    = 100
    maximum iterations = {maximum_iterations}
    minimum iterations = 2

    target relative residual     = 1e-6
    target residual              = 1e-8
    acceptable residual          = 1e-3
    acceptable relative residual = 1e-3

    begin full tangent preconditioner
      conditioning = no_check
      linear solver = feti
      minimum smoothing iterations = {smooth__iterations}
      small number of iterations    = 100
    end
  end cg

end solver
```

Appendix. Dynamic Singular Bar

- Inelasticity masks problems by introducing ***energy dissipation***.
- Schwarz does ***not*** introduce ***numerical artifacts***.
- Can couple domains with ***different time integration schemes*** (***Explicit-Implicit*** below).



Appendix. Elastic Wave: Energy Conservation



Total energy is **conserved** and matches single-domain total energy.

- For clamped beam problem, total energy (TE = $0.5\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x} + 0.5\dot{\boldsymbol{x}}^T\boldsymbol{M}\dot{\boldsymbol{x}}$) should be conserved.
- Total energy is calculated in 2 ways: with most of contribution from Ω_0 and from Ω_1.

Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

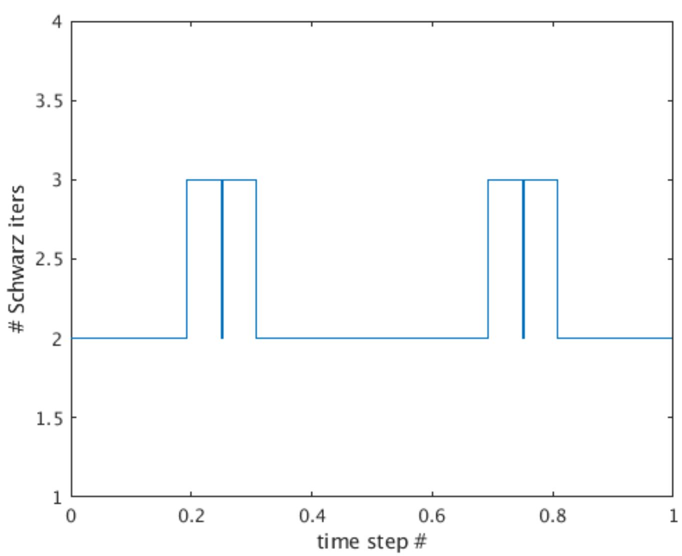
	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

As Schwarz tolerance is **tightened**, number of Schwarz **iterations goes up** (as expected)

Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

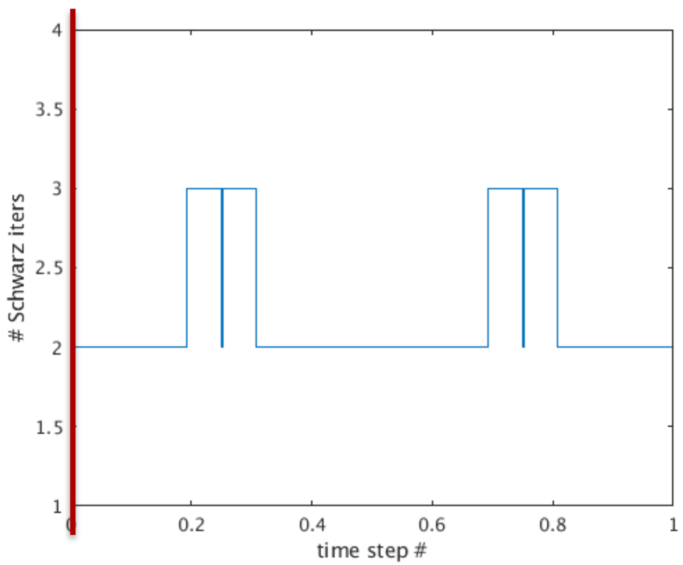


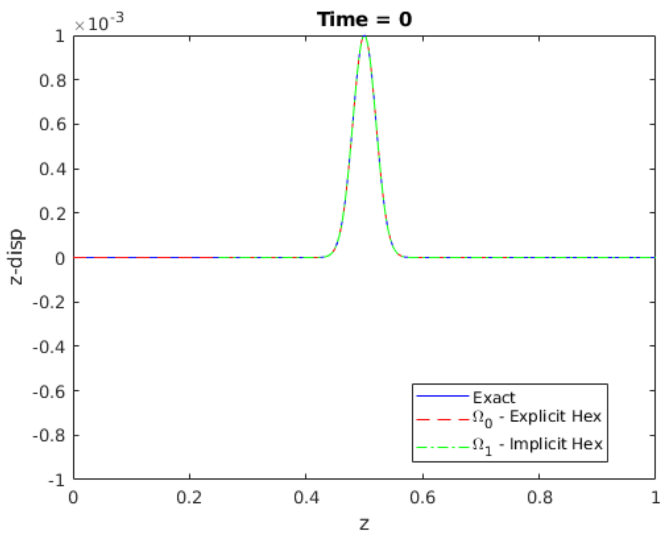
Left: time vs. # Schwarz iterations (per time step) – behavior is as expected given nature of solution/domain decomposition

Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

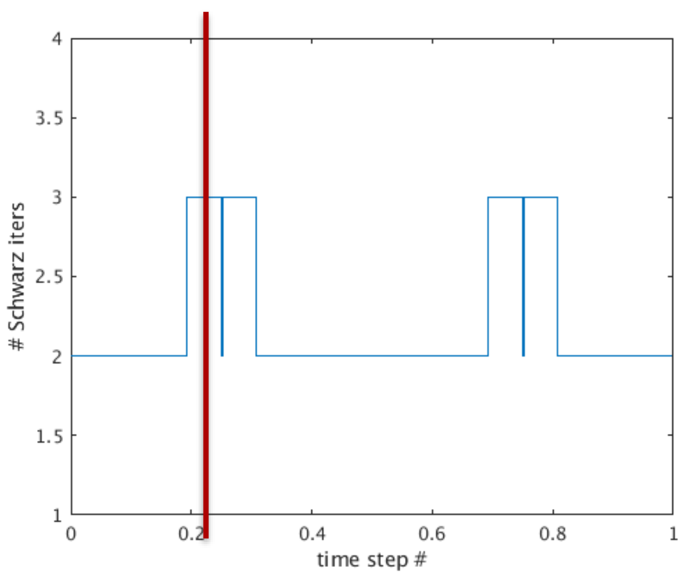


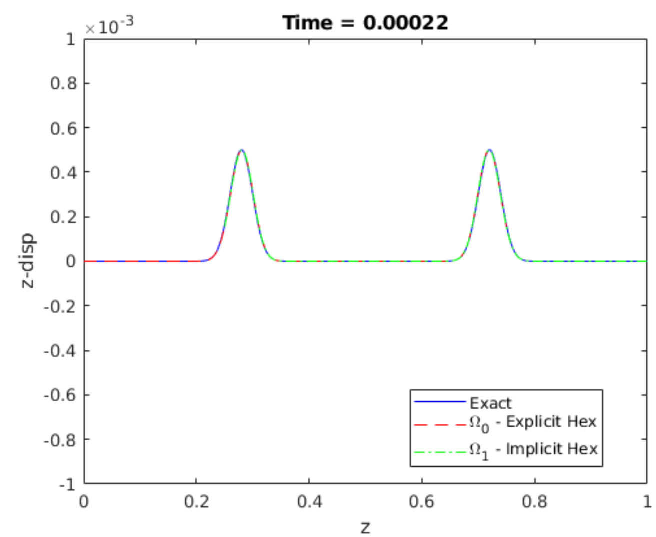


Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

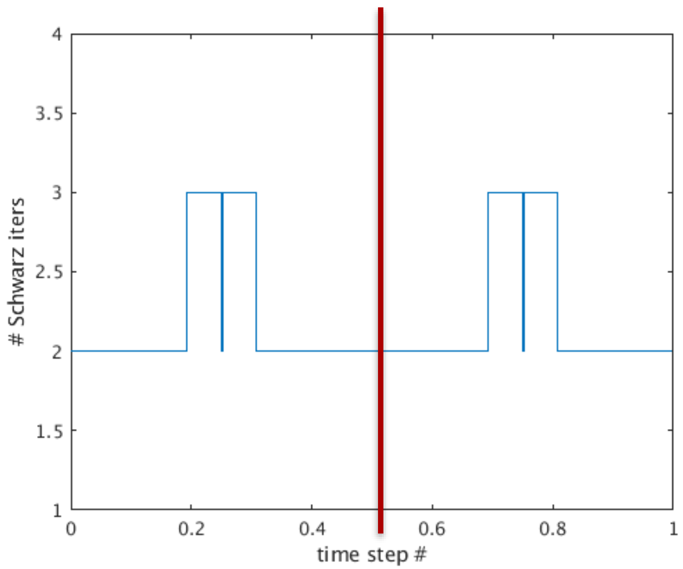


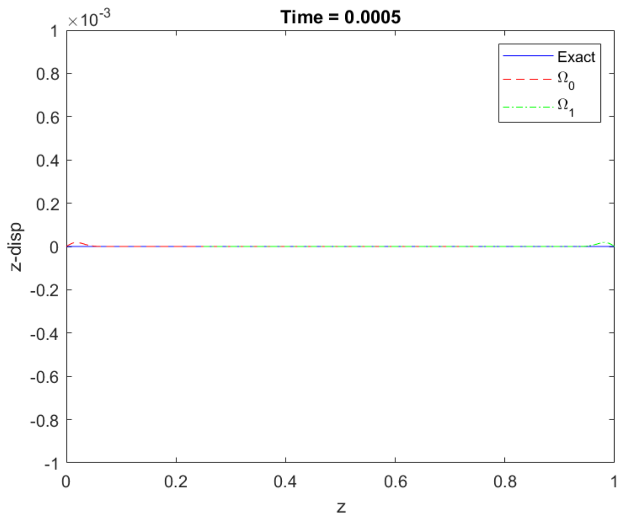


Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

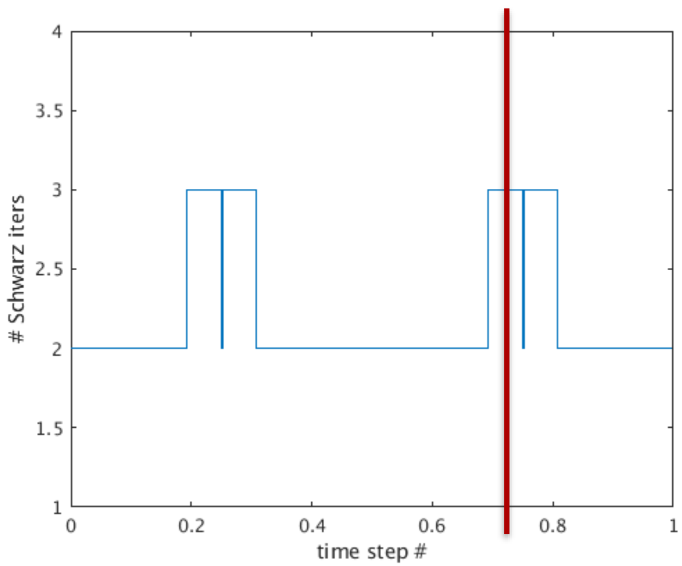


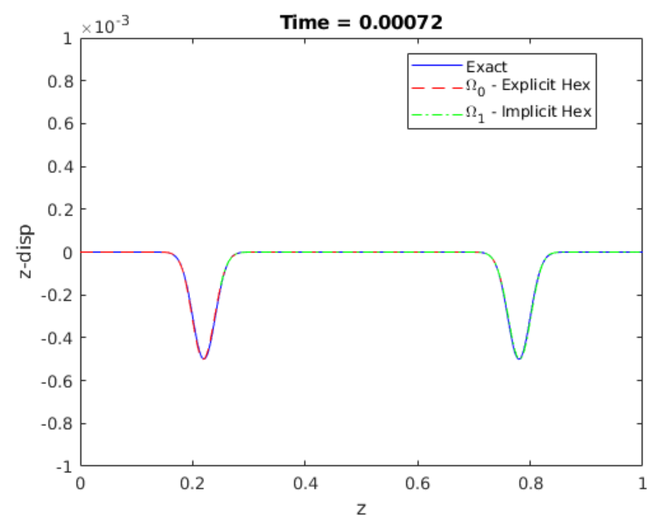


Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54

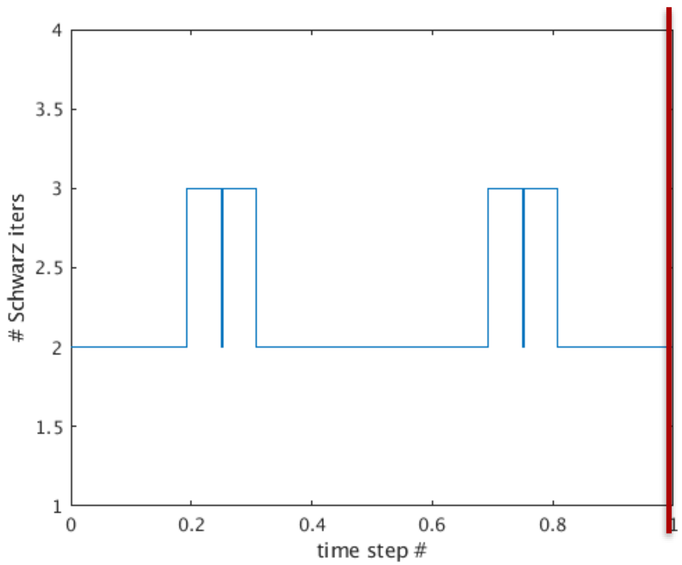


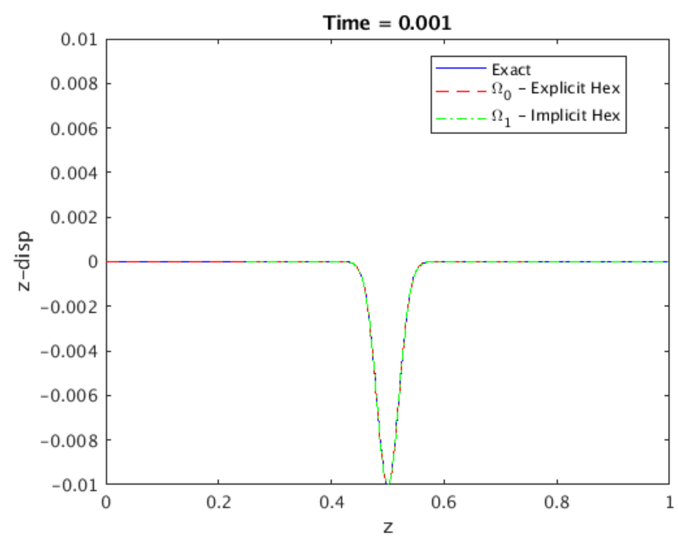


Appendix: Some Schwarz Performance Data

Max / Avg # Schwarz iterations during run

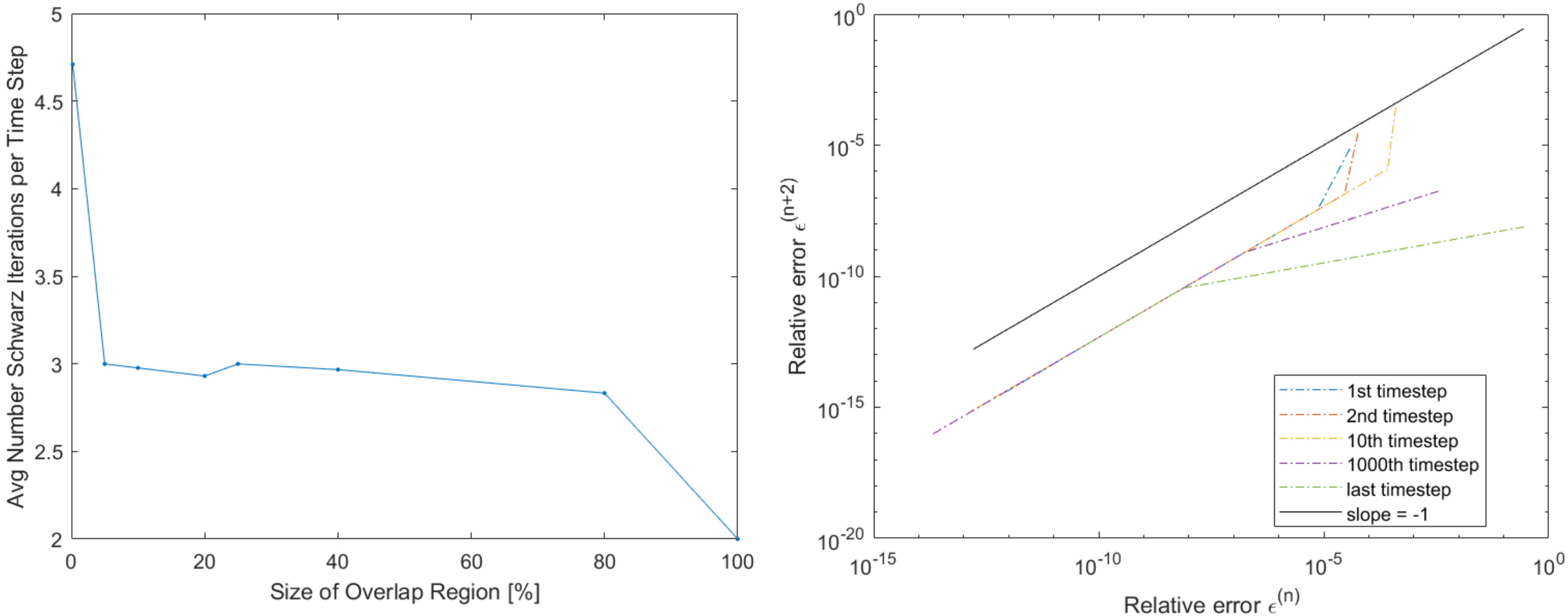
	Impl-impl dt=1e-6, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-6	Impl-impl dt=1e-7, Schwarz tol=1e-10	Impl-impl dt=1e-7, Schwarz tol=1e-15	Expl-impl dt=1e-7 Schwarz tol=1e-6	Expl-impl dt=1e-8 Schwarz tol=1e-6
Hex-Hex	3 / 2.23	3 / 2.08		4 / 2.83	2 / 2.0	
Tet-Hex					2 / 2.0	
Nonconformal Hex-Hex		2 / 2.0	3 / 2.36		2 / 2.0	2 / 1.54





Appendix. Elastic Wave Propagation

Some Performance Results



- Left figure shows ***# of iterations*** as a function of ***overlap region size*** for 2 subdomains. The method does not converge for 0% overlap. If the overlap is 100% then the single-domain solution is recovered for each of the subdomains.

- Right figure shows ***linear convergence rate*** of dynamic Schwarz implementation (for small overlap fraction of 0.2%).

Appendix. Torsion Problem

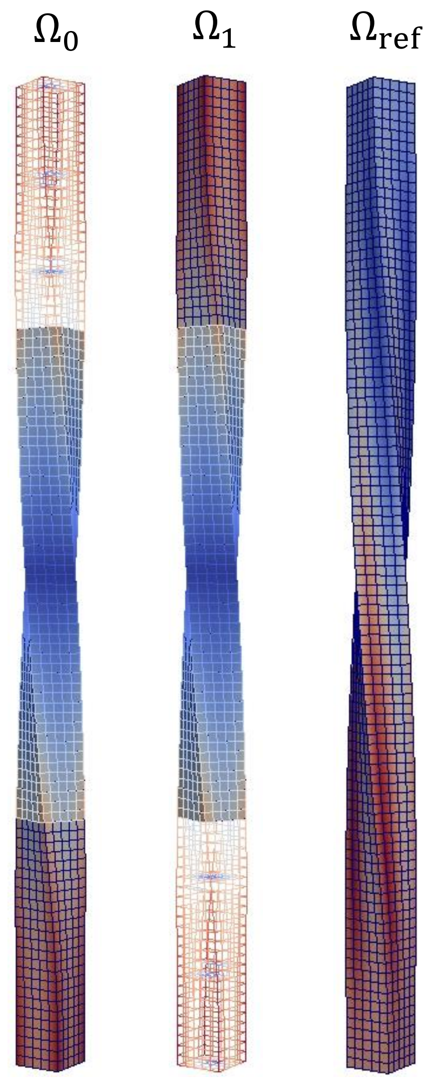
- Nonlinear elastic bar (Neohookean material model) subjected to a high degree of ***torsion***.

- The ***domain*** is $\Omega = (-0.025,0.025) \times (-0.025,0.025) \times (-0.5,0.5)$.

- We evaluate ***dynamic Schwarz*** with 2 subdomains: $\Omega_0 = (-0.025,0.025) \times (-0.025,0.025) \times (-0.5,0.25), \Omega_1 = (-0.025,0.025) \times (-0.025,0.025) \times (-0.25,0.5)$.

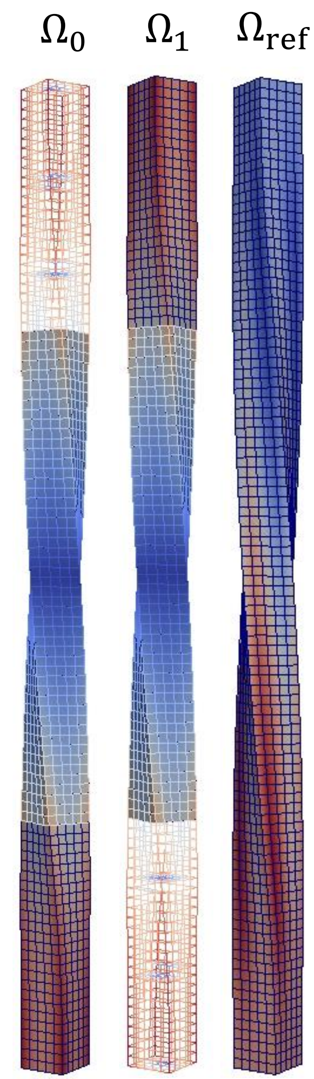
- ***Time-discretizations***: Newmark (implicit, explicit) with same Δt.

- ***Meshes:*** HEX, Composite TET10.

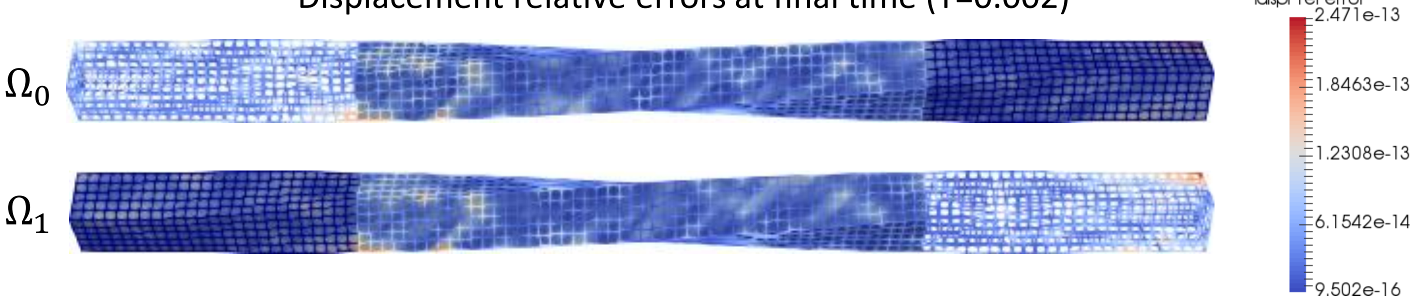


Appendix. Torsion: HEX - HEX Coupling

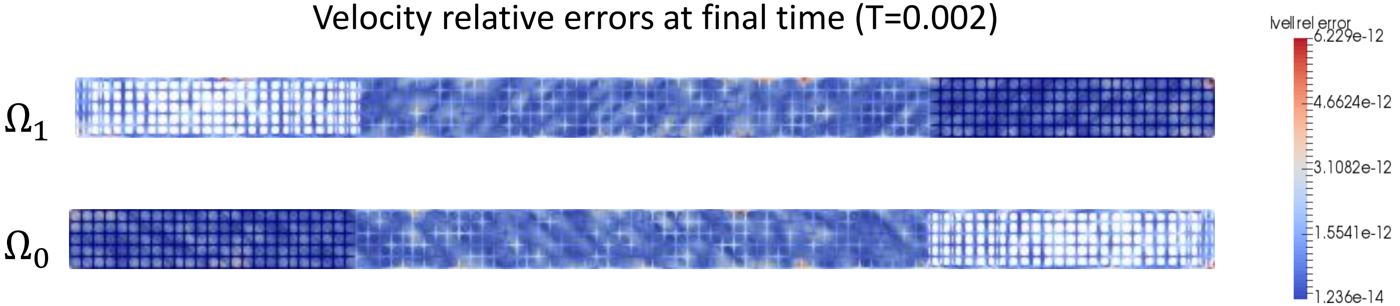
- Each subdomain discretized using **uniform HEX mesh** with $\Delta x_i = 0.01$, and advanced in time using implicit Newmark-Beta scheme with Δt =1e-6.
- Results compared to single-domain solution on mesh **conformal** with Schwarz domain meshes.



Displacement relative errors at final time (T=0.002)



Velocity relative errors at final time (T=0.002)



Schwarz and single-domain results agree to almost ***machine-precision***!

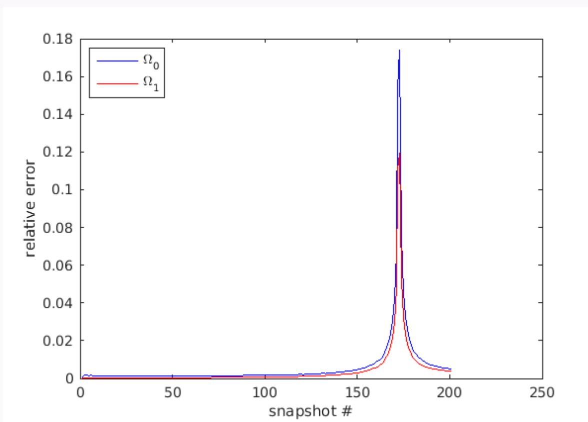
Appendix. Torsion: HEX - CTET10 Coupling

- Coupling of Composite TET10 elements + explicit Newmark with consistent mass in Ω_0 with HEX elements + implicit Newmark in Ω_1.
- Reference solution is computed on fine hex mesh + implicit Newmark Ω_{ref}

Relative error <1% and does not grow in time!

No dynamic artifacts!

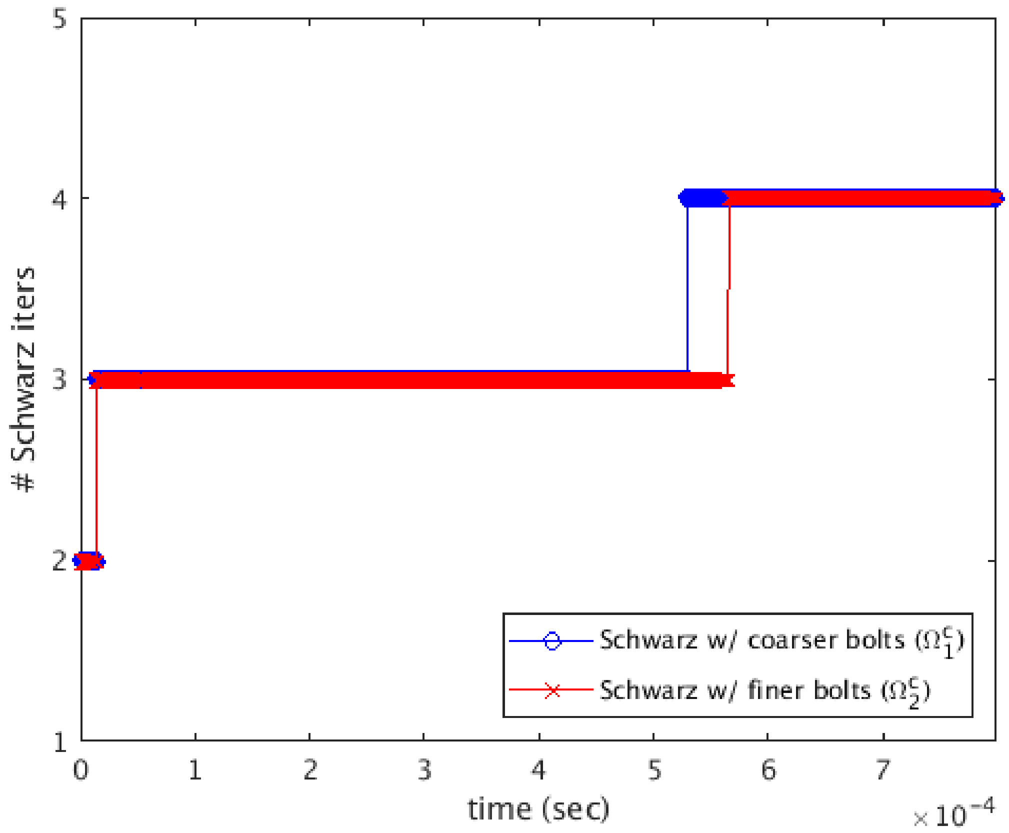
Time: 0.000000



Movie of |displacement|
Left: Single-domain,
Right: Schwarz

Albany

Appendix. Bolted Joint Problem: Performance



- Schwarz ***tolerance*** = 1e-6
- Dynamic Schwarz converges in between ***2-4 Schwarz iterations*** per time-step despite the ***overlap*** region being ***very small*** for this problem.

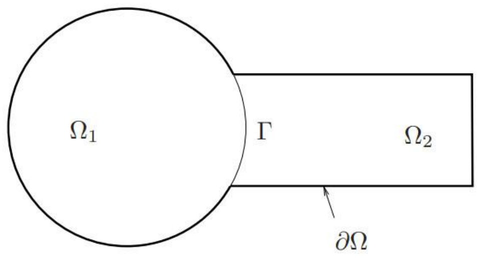
* On SNL ascicgpu15, 16, 17 machines (Intel Skylake CPU processor).

Appendix. Numerical Example: 1D Dynamic Wave Propagation Problem

- Alternating **Dirichlet-Neumann** Schwarz BCs with **no relaxation** ($\theta = 1$) on Schwarz boundary Γ

$$\begin{cases} \text{Div } \boldsymbol{P}_1^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0} \text{ , in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\lambda}_{n+1} & \text{on } \Gamma \end{cases}$$

$$\begin{cases} \text{Div } \boldsymbol{P}_2^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0} \text{ , in } \Omega_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_2 \backslash \Gamma \\ \boldsymbol{P}_2^{(n+1)} \boldsymbol{n} = \boldsymbol{P}_2^{(n+1)} \boldsymbol{n}, & \text{on } \Gamma \end{cases}$$



$$\boldsymbol{\lambda}_{n+1} = \theta \boldsymbol{\varphi}_2^{(n)} + (1-\theta)\boldsymbol{\lambda}_n, \text{on } \Gamma, \text{for } n \geq 1$$

θ	Min # Schwarz Iters	Max # Schwarz Iters	Total # Schwarz Iters
1.10	3	9	59,258
1.00	1	4	24,630
0.99	1	5	35,384
0.95	3	6	45,302
0.90	3	8	56,114

  - ➢ A **parameter sweep study** revealed $\theta = 1$ gave best performance (min # Schwarz iters)
- All couplings were **implicit-implicit** with $\Delta t_1 = \Delta t_2 = \Delta T = 10^{-7}$ and $\Delta x_1 = \Delta x_2 = 10^{-3}$
  - ➢ Time-step and spatial resolution chosen to be small enough to resolve propagating wave
- All reproductive and predictive cases run on the **same RHEL8 and RHEL7 machines.**
- Model **accuracy** evaluated w.r.t. analogous FOM-FOM coupling via **mean square error (MSE)**:

$$\mathcal{E}_{MSE}(\widetilde{\boldsymbol{u}}_i) := \frac{\sqrt{\sum_{n=1}^{S} ||\widetilde{\boldsymbol{u}}_i^n - \boldsymbol{u}_i^n||_2^2}}{\sqrt{\sum_{n=1}^{S} ||\boldsymbol{u}_i^n||_2^2}}$$