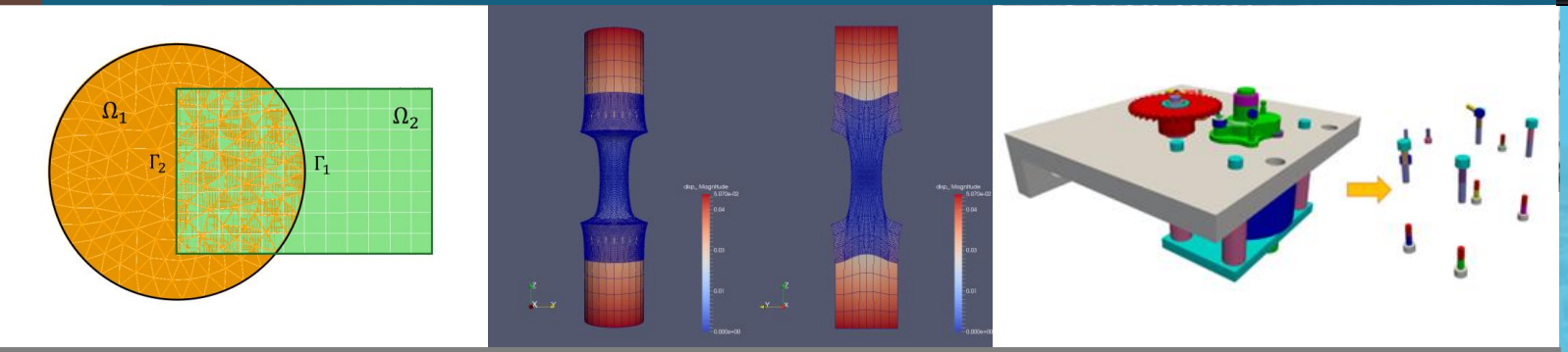
(Towards) Adaptive Hybrid Modeling via Schwarz Alternating Coupling and Operator Inference Reduced Order Models





I. Tezaur[1], E. Parish[1], A. Gruber[1], I. Moore[1,2], A. Mota[1]
C. Wentland[1], A. Diaz[1], C. Rodriguez[3], T. Mondal[4], P. Blonigan[1]

[1]Sandia National Laboratories, USA. [2]Rice University, USA.
[3]Columbia University, USA, [4]Worcester Polytech Institute, USA

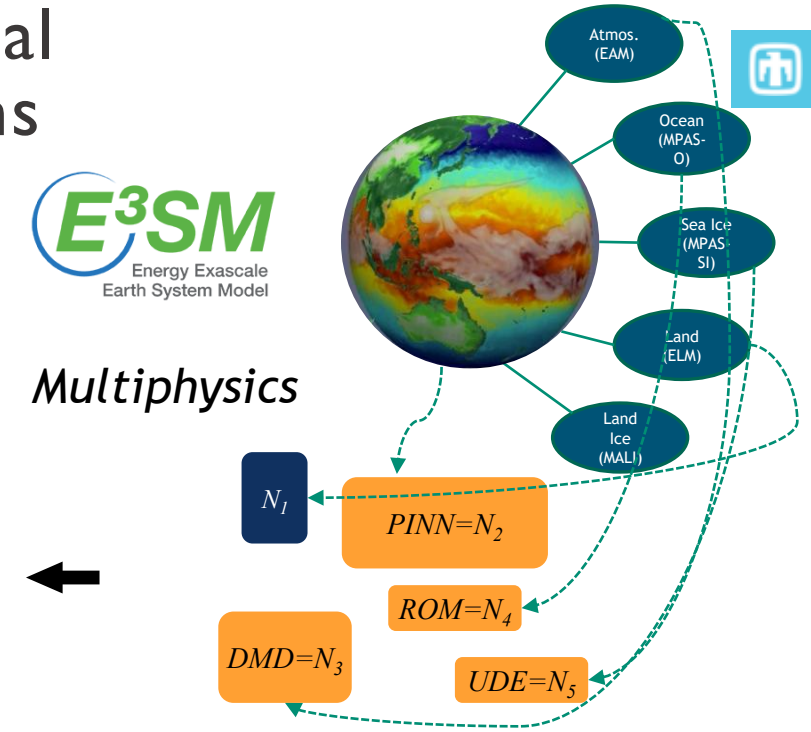
MORe 2026 November 2-6, 2026 Milan, Italy

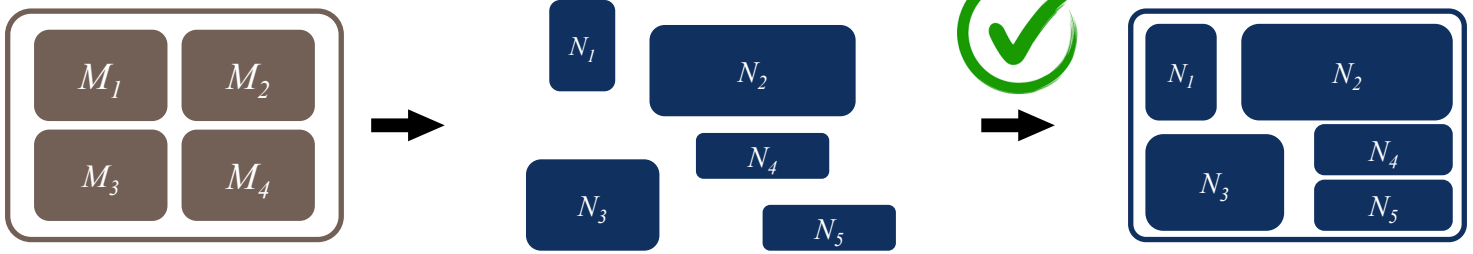
SAND2026-27385C

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

Motivation: Integration of ROMs into Traditional Mod/Sim Workflows for Predictive Digital Twins

There exist established **rigorous mathematical theories** for **coupling** multiscale/multiphysics components based on **traditional discretization methods** (FOMs/HFMs).





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

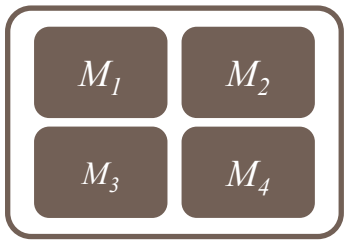
- Monolithic (Lagrange multipliers)
- Partitioned (loose) coupling
- Iterative (Schwarz, optimization)

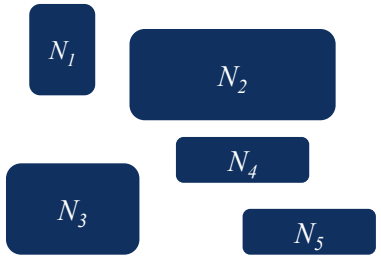
Traditional + Data-Driven Methods

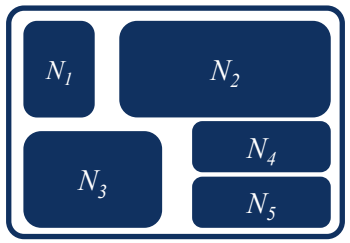
- PINNs
- Neural ODEs
- Projection-based ROMs, …

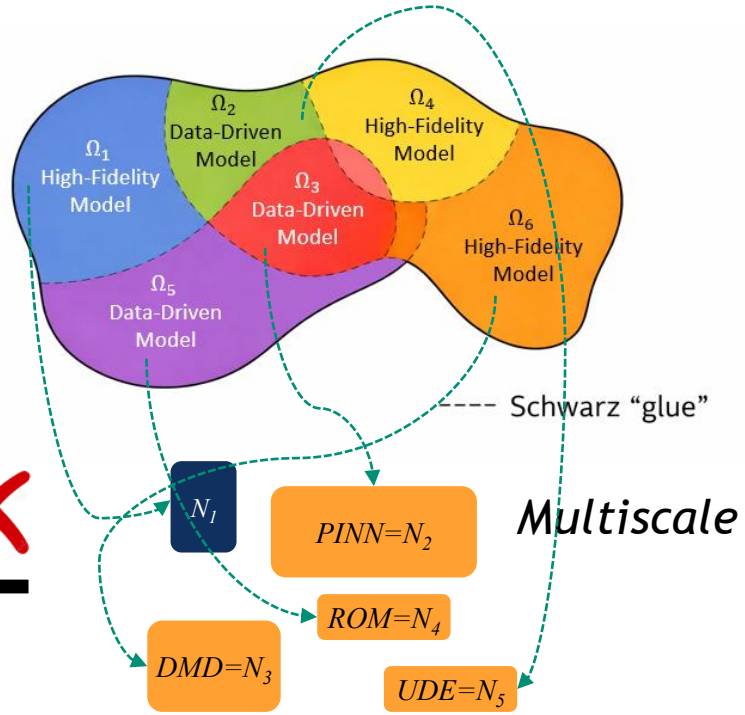
Motivation: Integration of ROMs into Traditional Mod/Sim Workflows for Predictive Digital Twins

There exist established **rigorous mathematical theories** for **coupling** multiscale/multiphysics components based on **traditional discretization methods** (FOMs/HFMs).









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)

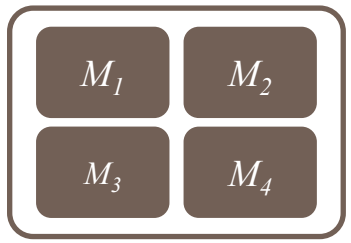
Traditional + Data-Driven Methods

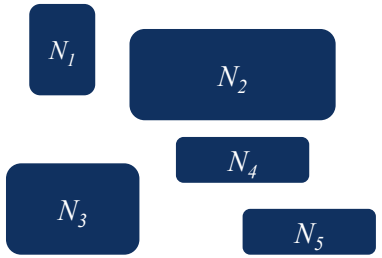
- PINNs
- Neural ODEs
- Projection-based ROMs, …

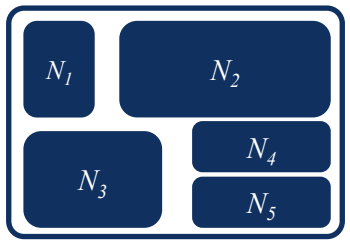
- **Deficiencies of data-driven models and couplings involving them:**
  - Existing coupling methods for high-fidelity models **may not work** out-of-the-box when including **data-driven models**
  - **ROMs** can suffer from **lack of robustness**, **stability** and **accuracy**, and **cannot** be **easily refined** to achieve a specified accuracy

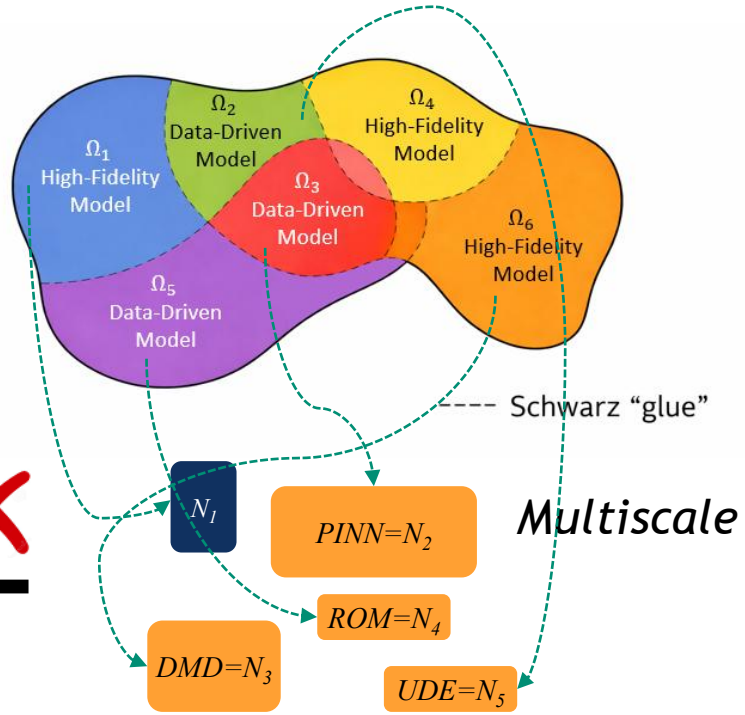
Motivation: Integration of ROMs into Traditional Mod/Sim Workflows for Predictive Digital Twins

There exist established **rigorous mathematical theories** for **coupling** multiscale/multiphysics components based on **traditional discretization methods** (FOMs/HFMs).









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

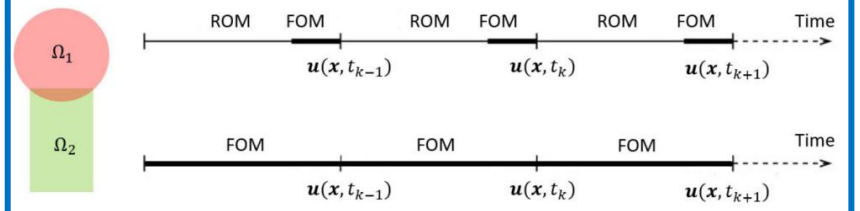
- Monolithic (Lagrange multipliers)
- Partitioned (loose) coupling
- Iterative (Schwarz, optimization)

Traditional + Data-Driven Methods

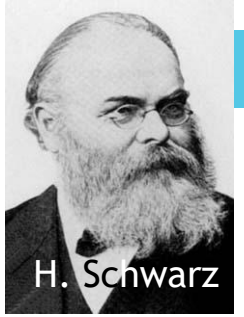
- PINNs
- Neural ODEs
- Projection-based ROMs, …

- **Deficiencies of data-driven models and couplings involving them:**
  - Existing coupling methods for high-fidelity models **may not work** out-of-the-box when including **data-driven models**
  - **ROMs** can suffer from **lack of robustness**, **stability** and **accuracy**, and **cannot** be **easily refined** to achieve a specified accuracy

Proposed solution: domain decomposition-(DD-) based coupling via the *Schwarz alternating method* & *online model switching*



Schwarz Alternating Method (SAM)



- 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.

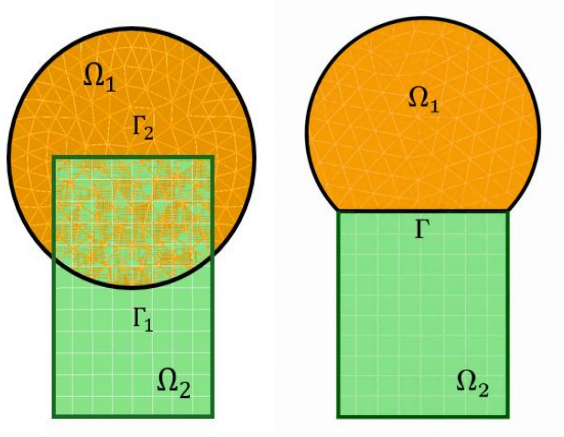
Basic Schwarz Algorithm

Initialize:

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

Iterate until convergence:

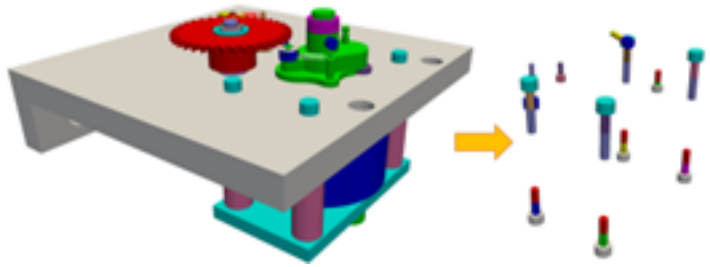
- Solve PDE by any method in Ω_2 w/ BCs on Γ_2 from just-obtained Ω_1 solution.
- Solve PDE by any method in Ω_1 w/ BCs on Γ_1 from just-obtained Ω_2 solution.



Overlapping *Non-overlapping*

- SAM most commonly used as a ***preconditioner*** for linear solvers.

<u>Idea behind this work</u>: using SAM as a ***discretization/ coupling method*** for solving multiscale PDEs.



*<u>Original application</u>: use SAM to simplify meshing in solid mechanics**.

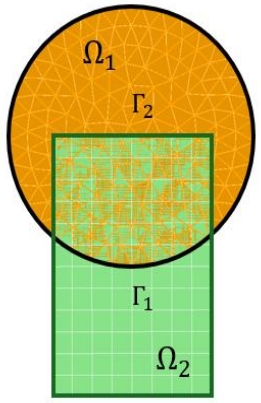
*Mota *et al.*, 2017; Mota *et al.*, 2022.

Domain Decomposition (DD)-based Coupling via SAM

Overlapping Schwarz Alternating Method (O-SAM)

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

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



*Easier **implementation** and **faster convergence***

Model PDE:

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

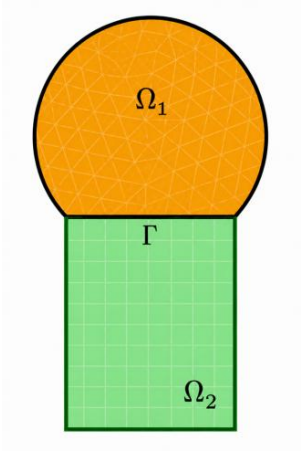
- Dirichlet-Dirichlet transmission BCs [Schwarz, 1870; Lions, 1988]

Non-overlapping Schwarz Alternating Method (NO-SAM)

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

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

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



*More **flexible** but **slower** to converge*

- Relevant for multi-material and multiphysics 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)

Non-Intrusive Model Order Reduction via Polynomial Operator Inference (OpInf)*

Key Idea Behind OpInf: circumvent the burden of implementing intrusive ROMs in HPC codes by **combining projection-based ROM** and **machine learning (ML).**

Nuances:

- OpInf can be applied to **nonlinear problems** by transforming the nonlinear PDEs into PDEs with a polynomial functional form (**"lifting"** [Qian *et al.*, 2019]) or assuming a **polynomial functional form** for the ROM
- The OpInf least-squares (LS) minimization problem often requires **regularization** to be solvable, e.g., Tikhonov regularization
- **Structure preservation** (e.g., symmetry constraints) can be incorporated into the OpInf LS minimization problem

FOM:

$$\dot{\boldsymbol{q}} + \boldsymbol{A}\boldsymbol{q} + \boldsymbol{H}(\boldsymbol{q} \otimes \boldsymbol{q}) + \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0}$$

snapshots of $\boldsymbol{q}$

Reduced basis $\boldsymbol{\Phi}$
(e.g., linear POD or quadratic manifold basis)

$$\min_{\widehat{\boldsymbol{A}},\widehat{\boldsymbol{H}},\widehat{\boldsymbol{B}}} \left\| \dot{\widehat{\boldsymbol{Q}}}^T + \widehat{\boldsymbol{Q}}^T \widehat{\boldsymbol{A}}^T + (\widehat{\boldsymbol{Q}} \otimes \widehat{\boldsymbol{Q}})^T \widehat{\boldsymbol{H}}^T + \widehat{\boldsymbol{U}}^T \widehat{\boldsymbol{B}}^T \right\|_F^2 + \gamma \left(\left\| \widehat{\boldsymbol{A}} \right\|_F + \left\| \widehat{\boldsymbol{H}} \right\|_F + \left\| \widehat{\boldsymbol{B}} \right\|_F \right)$$

$\widehat{\boldsymbol{Q}}$: snapshots generated from projected simulation data $\boldsymbol{q}$

$\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{H}}, \widehat{\boldsymbol{B}}$: low-dimensional operators defining ROM dynamical system

OpInf ROM:

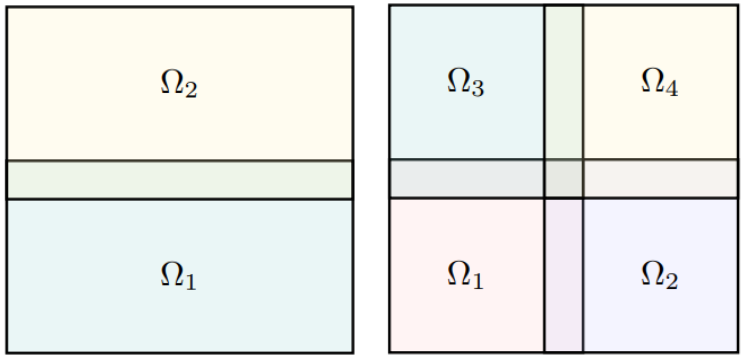
$$\dot{\widehat{\boldsymbol{q}}} + \widehat{\boldsymbol{A}}\widehat{\boldsymbol{q}} + \widehat{\boldsymbol{H}}(\widehat{\boldsymbol{q}} \otimes \widehat{\boldsymbol{q}}) + \widehat{\boldsymbol{B}}\boldsymbol{u} = \boldsymbol{0}$$

* [Peherstorfer & Willcox, 2016] + many subsequent refs

Overlapping SAM-based Coupling for Non-Intrusive OpInf ROMs

Offline stage:

- Create **DD** of Ω into d **overlapping subdomains** Ω_i
- Perform a SAM-coupled **all-FOM simulation** on $\mathrm{U}_i\,\Omega_i$
- Compute **POD basis** $\boldsymbol{\Phi}_i$ on each Ω_i
- Assume a **functional form** for your ROM in Ω_i, informed by the functional form of the corresponding FOM
  - ***Key question**: how to impose Schwarz BCs in OpInf ROMs?*



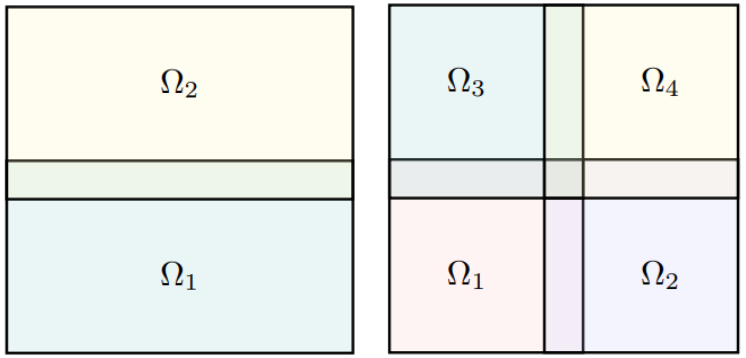
OpInf ROM in Ω_i:

$$\dot{\hat{\boldsymbol{q}}}_i + \hat{\boldsymbol{A}}_i\hat{\boldsymbol{q}}_i + \hat{\boldsymbol{H}}_i(\hat{\boldsymbol{q}}_i \otimes \hat{\boldsymbol{q}}_i) = \boldsymbol{0}$$

Overlapping SAM-based Coupling for Non-Intrusive OpInf ROMs

Offline stage:

- Create **DD** of Ω into d **overlapping subdomains** Ω_i
- Perform a SAM-coupled **all-FOM simulation** on $\mathrm{U}_i\,\Omega_i$
- Compute **POD basis** $\boldsymbol{\Phi}_i$ on each Ω_i
- Assume a **functional form** for your ROM in Ω_i, informed by the functional form of the corresponding FOM
  - ➢ ***Key question**: how to impose Schwarz BCs in OpInf ROMs?*
    - ❖ *Boundary transmission enters through **learned source term** $\widehat{\boldsymbol{B}}_i\boldsymbol{g}_i$ added to OpInf ROM dynamical system*



OpInf ROM + Schwarz BCs in Ω_i:

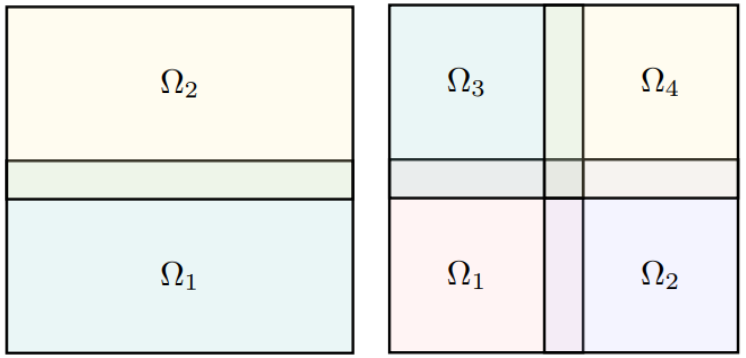
$$\dot{\widehat{\boldsymbol{q}}}_i + \widehat{\boldsymbol{A}}_i\widehat{\boldsymbol{q}}_i + \widehat{\boldsymbol{H}}_i(\widehat{\boldsymbol{q}}_i \otimes \widehat{\boldsymbol{q}}_i) + \underbrace{\widehat{\boldsymbol{B}}_i\boldsymbol{g}_i}_{\text{Schwarz Dirichlet BC term}} = \boldsymbol{0}$$

Motivated by implementation of **Dirichlet BCs** in **FEM**

Overlapping SAM-based Coupling for Non-Intrusive OpInf ROMs

Offline stage:

- Create **DD** of Ω into d **overlapping subdomains** Ω_i
- Perform a SAM-coupled **all-FOM simulation** on $\cup_i \Omega_i$
- Compute **POD basis** $\boldsymbol{\Phi}_i$ on each Ω_i
- Assume a **functional form** for your ROM in Ω_i, informed by the functional form of the corresponding FOM
  - ***Key question**: how to impose Schwarz BCs in OpInf ROMs?*
    - *Boundary transmission enters through **learned source term** $\widehat{\boldsymbol{B}}_i \boldsymbol{g}_i$ added to OpInf ROM dynamical system*
    - ***Further reduction** achieved by expanding $\boldsymbol{g}_i$ in its own POD basis $\boldsymbol{\Phi}_i^g$ and approximating $\widehat{\boldsymbol{B}}_i \boldsymbol{g}_i \approx \widehat{\boldsymbol{B}}_i \widehat{\boldsymbol{g}}_i = \widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i$ where $\widehat{\boldsymbol{g}}_i = \boldsymbol{\Phi}_i^{g} \boldsymbol{g}_i$*



OpInf ROM + Schwarz BCs in Ω_i:

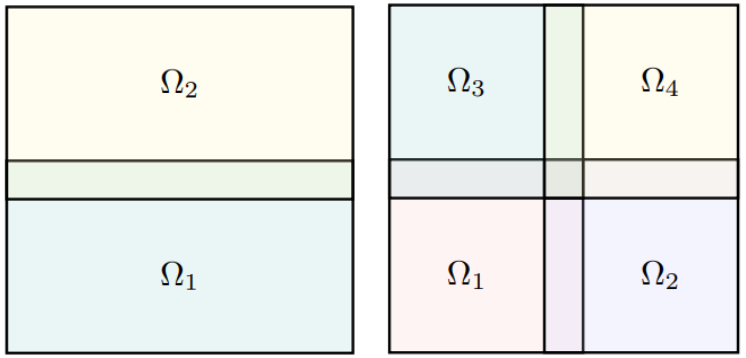
$$\dot{\widehat{\boldsymbol{q}}}_i + \widehat{\boldsymbol{A}}_i \widehat{\boldsymbol{q}}_i + \widehat{\boldsymbol{H}}_i (\widehat{\boldsymbol{q}}_i \otimes \widehat{\boldsymbol{q}}_i) + \underbrace{\widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i}_{\text{Schwarz Dirichlet BC term}} = \boldsymbol{0}$$

Motivated by implementation of **Dirichlet BCs** in **FEM**

Overlapping SAM-based Coupling for Non-Intrusive OpInf ROMs

Offline stage:

- Create **DD** of Ω into d **overlapping subdomains** Ω_i
- Perform a SAM-coupled **all-FOM simulation** on $\cup_i \Omega_i$
- Compute **POD basis** $\boldsymbol{\Phi}_i$ on each Ω_i
- Assume a **functional form** for your ROM in Ω_i, informed by the functional form of the corresponding FOM
  - ***Key question**: how to impose Schwarz BCs in OpInf ROMs?*
    - *Boundary transmission enters through **learned source term** $\widehat{\boldsymbol{B}}_i \boldsymbol{g}_i$ added to OpInf ROM dynamical system*
    - ***Further reduction** achieved by expanding $\boldsymbol{g}_i$ in its own POD basis $\boldsymbol{\Phi}_i^g$ and approximating $\widehat{\boldsymbol{B}}_i \boldsymbol{g}_i \approx \widehat{\boldsymbol{B}}_i \widehat{\boldsymbol{g}}_i = \widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i$ where $\widehat{\boldsymbol{g}}_i = \boldsymbol{\Phi}_i^{g} \boldsymbol{g}_i$*
- Compute OpInf operators $\widehat{\boldsymbol{A}}_i, \widehat{\boldsymbol{H}}_i$ and $\widetilde{\boldsymbol{B}}_i$ in each subdomain Ω_i by solving regularized OpInf least squares **minimization problem**



OpInf ROM + Schwarz BCs in Ω_i:

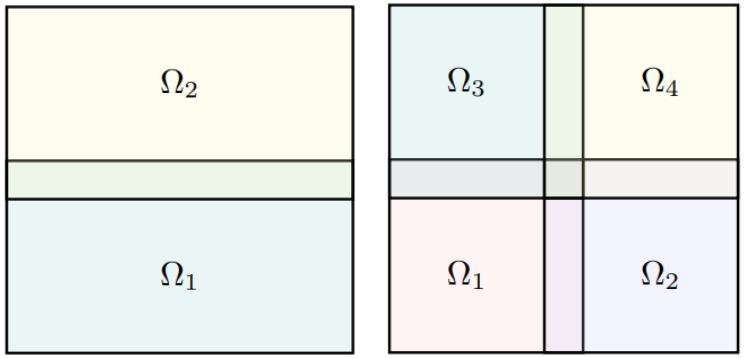
$$\dot{\widehat{\boldsymbol{q}}}_i + \widehat{\boldsymbol{A}}_i \widehat{\boldsymbol{q}}_i + \widehat{\boldsymbol{H}}_i (\widehat{\boldsymbol{q}}_i \otimes \widehat{\boldsymbol{q}}_i) + \underbrace{\widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i}_{\text{Schwarz Dirichlet BC term}} = \boldsymbol{0}$$

Motivated by implementation of **Dirichlet BCs** in **FEM**

Overlapping SAM-based Coupling for Non-Intrusive OpInf ROMs

Offline stage:

- Create **DD** of Ω into d **overlapping subdomains** Ω_i
- Perform a SAM-coupled **all-FOM simulation** on $\cup_i \Omega_i$
- Compute **POD basis** $\boldsymbol{\Phi}_i$ on each Ω_i
- Assume a **functional form** for your ROM in Ω_i, informed by the functional form of the corresponding FOM
  - ***Key question**: how to impose Schwarz BCs in OpInf ROMs?*
    - *Boundary transmission enters through **learned source term** $\widehat{\boldsymbol{B}}_i \boldsymbol{g}_i$ added to OpInf ROM dynamical system*
    - ***Further reduction** achieved by expanding $\boldsymbol{g}_i$ in its own POD basis $\boldsymbol{\Phi}_i^g$ and approximating $\widehat{\boldsymbol{B}}_i \boldsymbol{g}_i \approx \widehat{\boldsymbol{B}}_i \widehat{\boldsymbol{g}}_i = \widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i$ where $\widehat{\boldsymbol{g}}_i = \boldsymbol{\Phi}_i^{g} \boldsymbol{g}_i$*
- Compute OpInf operators $\widehat{\boldsymbol{A}}_i, \widehat{\boldsymbol{H}}_i$ and $\widetilde{\boldsymbol{B}}_i$ in each subdomain Ω_i by solving regularized OpInf least squares **minimization problem**



OpInf ROM + Schwarz BCs in Ω_i:

$$\dot{\widehat{\boldsymbol{q}}}_i + \widehat{\boldsymbol{A}}_i \widehat{\boldsymbol{q}}_i + \widehat{\boldsymbol{H}}_i (\widehat{\boldsymbol{q}}_i \otimes \widehat{\boldsymbol{q}}_i) + \underbrace{\widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i}_{\text{Schwarz Dirichlet BC term}} = \boldsymbol{0}$$

Motivated by implementation of **Dirichlet BCs** in **FEM**

Online stage:

- Apply **Schwarz iteration procedure**, with Schwarz BC transfer via pre-learned boundary contributions $\widetilde{\boldsymbol{B}}_i \boldsymbol{g}_i$

Tension Specimen (Overlapping SAM, Predictive)

- **Hyperelastic** (Neohookean) tension specimen pulled from top
- **TET10 – HEX8 overlapping** coupling with **implicit Newmark** with **same Δt**
- **Quadratic OpInf (QOpInf)** models with M=2 POD modes, capturing 99.9999% snapshot energy
- All results are **predictive** w.r.t. DBC applied at top of holder

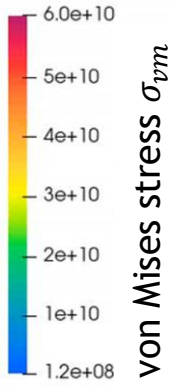
		FOM-FOM	FOM-QOpInf	QOpInf-QOpInf
Ω_1 rel errs	displacement	–	3.44e-4	5.73e-4
	velocity	–	1.72e-2	1.83e-2
	σ_{vm} stress	–	3.41e-4	8.53e-4
Ω_2 rel errs	displacement	–	2.50e-4	4.41e-4
	velocity	–	1.86e-2	1.96e-2
	σ_{vm} stress	–	2.40e-3	6.00e-3
CPU time	–	8h 19m 29.5s	1h 21m 25.9s	4m 42.1s
Mean/max # Schwarz iters	–	32.0/32	7.03/8	7.74/8



FOM-FOM



FOM-QOpInf



QOpInf = Quadratic OpInf

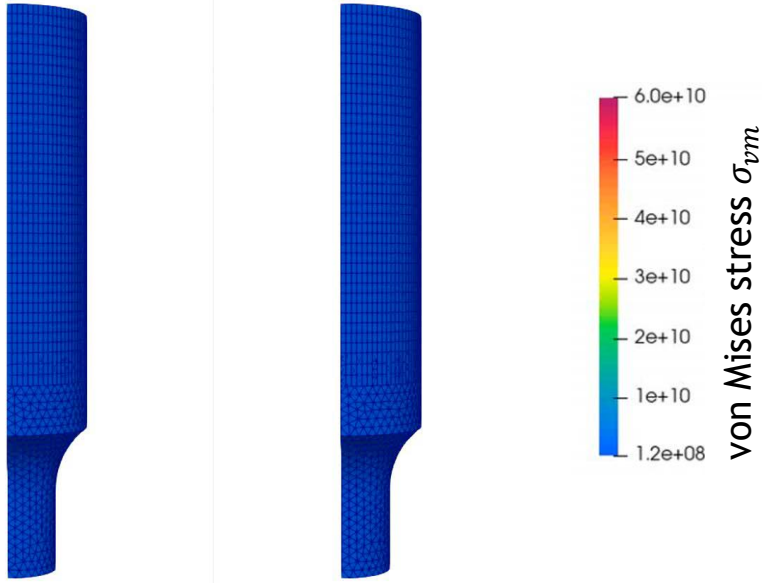
NORMA.JL

Tension Specimen (Overlapping SAM, Predictive)

- **Hyperelastic** (Neohookean) tension specimen pulled from top
- **TET10 - HEX8 overlapping** coupling with **implicit Newmark** with **same Δt**
- **Quadratic OpInf (QOpInf)** models with M=2 POD modes, capturing 99.9999% snapshot energy
- All results are **predictive** w.r.t. DBC applied at top of holder

		FOM-FOM	FOM-QOpInf	QOpInf-QOpInf
Ω_1 rel errs	displacement	–	3.44e-4	5.73e-4
	velocity	–	1.72e-2	1.83e-2
	σ_{vm} stress	–	3.41e-4	8.53e-4
Ω_2 rel errs	displacement	–	2.50e-4	4.41e-4
	velocity	–	1.86e-2	1.96e-2
	σ_{vm} stress	–	2.40e-3	6.00e-3
CPU time	–	8h 19m 29.5s	1h 21m 25.9s	4m 42.1s
Mean/max # Schwarz iters	–	32.0/32	7.03/8	7.74/8

- Relative errors of $\boldsymbol{O}$**(1e-4)-**$\boldsymbol{O}$**(1e-3)** are achieved for the displacement and von Mises stress (σ_{vm})



FOM-FOM FOM-QOpInf

QOpInf = Quadratic OpInf

NORMA.JL

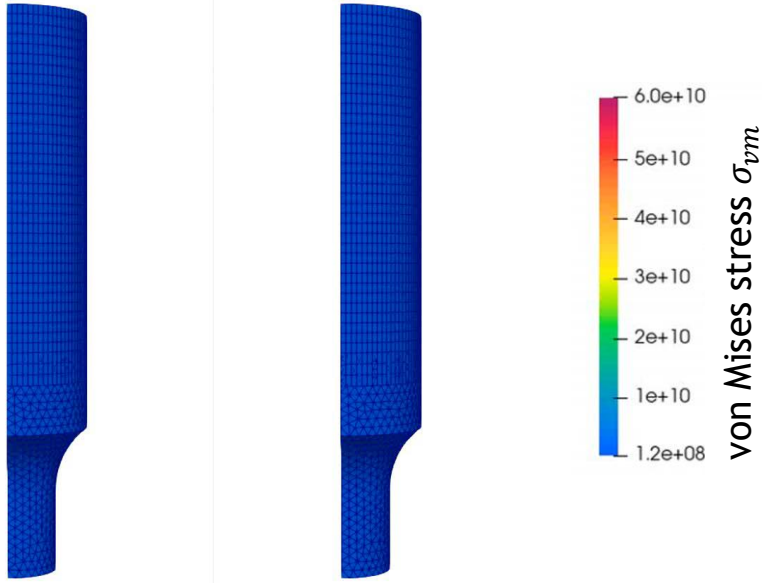
Tension Specimen (Overlapping SAM, Predictive)

- **Hyperelastic** (Neohookean) tension specimen pulled from top
- **TET10 – HEX8 overlapping** coupling with **implicit Newmark** with **same Δt**
- **Quadratic OpInf (QOpInf)** models with M=2 POD modes, capturing 99.9999% snapshot energy
- All results are **predictive** w.r.t. DBC applied at top of holder

		FOM-FOM	FOM-QOpInf	QOpInf-QOpInf
Ω_1 rel errs	displacement	–	3.44e-4	5.73e-4
	velocity	–	1.72e-2	1.83e-2
	σ_{vm} stress	–	3.41e-4	8.53e-4
Ω_2 rel errs	displacement	–	2.50e-4	4.41e-4
	velocity	–	1.86e-2	1.96e-2
	σ_{vm} stress	–	2.40e-3	6.00e-3
CPU time	–	8h 19m 29.5s	1h 21m 25.9s	4m 42.1s
Mean/max # Schwarz iters	–	32.0/32	7.03/8	7.74/8

- Relative errors of ***O*(1e-4)-*O*(1e-3)** are achieved for the displacement and von Mises stress (σ_{vm})

Impressive **6.13×** and **106×** **speedups** are achieved via **FOM-QOpInf** and **QOpInf-QOpInf** couplings, respectively!



FOM-FOM FOM-QOpInf

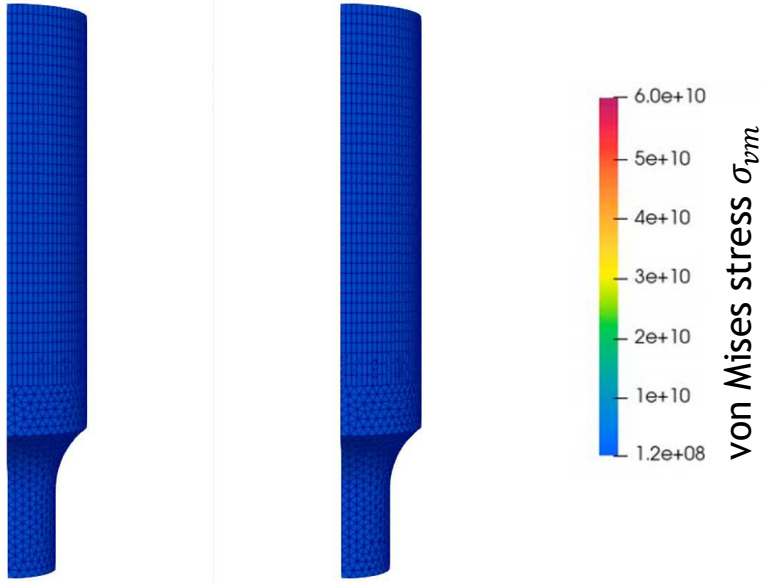
QOpInf = Quadratic OpInf

NORMA.JL

Tension Specimen (Overlapping SAM, Predictive)

- **Hyperelastic** (Neohookean) tension specimen pulled from top
- **TET10 – HEX8 overlapping** coupling with **implicit Newmark** with **same Δt**
- **Quadratic OpInf (QOpInf)** models with M=2 POD modes, capturing 99.9999% snapshot energy
- All results are **predictive** w.r.t. DBC applied at top of holder

		FOM-FOM	FOM-QOpInf	QOpInf-QOpInf
Ω_1 rel errs	displacement	–	3.44e-4	5.73e-4
	velocity	–	1.72e-2	1.83e-2
	σ_{vm} stress	–	3.41e-4	8.53e-4
Ω_2 rel errs	displacement	–	2.50e-4	4.41e-4
	velocity	–	1.86e-2	1.96e-2
	σ_{vm} stress	–	2.40e-3	6.00e-3
CPU time	–	8h 19m 29.5s	1h 21m 25.9s	4m 42.1s
Mean/max # Schwarz iters	–	32.0/32	7.03/8	7.74/8



FOM-FOM FOM-QOpInf

- Relative errors of ***O*(1e-4)-*O*(1e-3)** are achieved for the displacement and von Mises stress (σ_{vm})

Impressive **6.13×** and **106×** **speedups** are achieved via **FOM-QOpInf** and **QOpInf-QOpInf** couplings, respectively!

- Speedup largely due to huge reduction in # Schwarz iterations

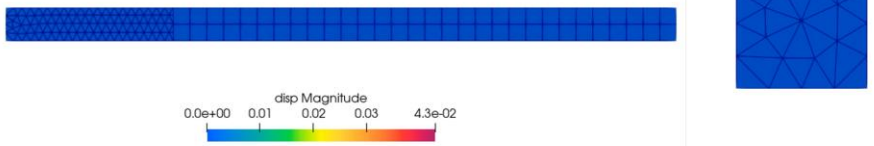
Conjecture: *fewer Schwarz iterations likely b/c **regularization** makes OpInf learn a **weaker-than-physical response** to Schwarz BCs, which makes the Schwarz **fixed-point iteration** contract **faster**.*

QOpInf = Quadratic OpInf

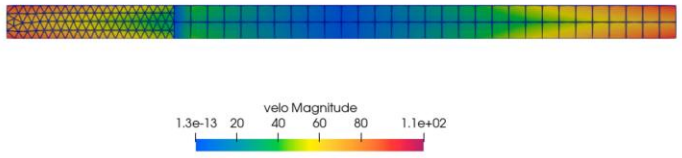
Torsion (Overlapping SAM, Reproductive & Predictive)

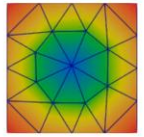
- **Neohookean** material model
- **TET4 - HEX8 overlapping** coupling with **implicit-explicit Newmark** having **different Δt**
- **QOpInf-FOM** coupled models with M=27 and M=30 POD modes, capturing 99.9999% snapshot energy
- **Prediction** w.r.t. **initial velocity** (rotation speed/direction)
- (Linear) **OpInf-FOM** coupling **insufficient**

		FOM-FOM	QOpInf-FOM reproductive	QOpInf-FOM predictive
Ω_1 rel errs	displacement	–	2.67e-3	4.32e-2
	velocity	–	3.56e-2	1.49e-1
Ω_2 rel errs	displacement	–	1.13e-3	2.44e-2
	velocity	–	1.12e-2	9.52e-2
CPU time	–	39m 11.8s	1m 40.2s	1m 39.5s
Mean/max # Schwarz iters	–	3.0/3	2.0/2	2.0/2

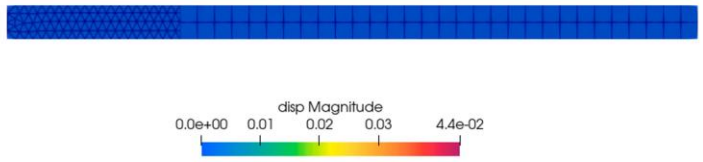


FOM-FOM



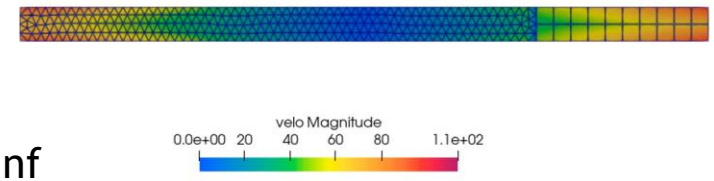


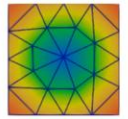
QOpInf = Quadratic OpInf

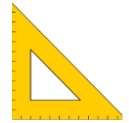




QOpInf-FOM







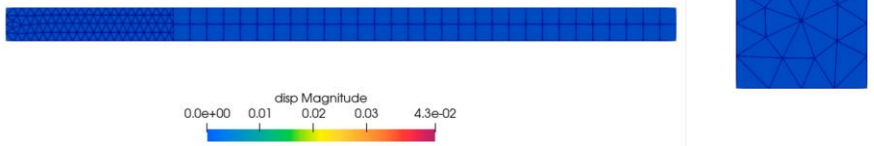
NORMA.JL

Torsion (Overlapping SAM, Reproductive & Predictive)

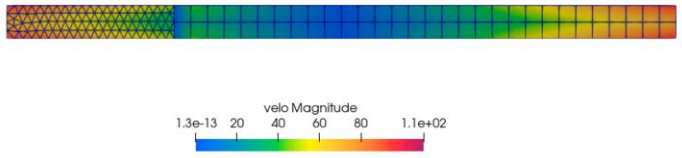
- **Neohookean** material model
- **TET4 - HEX8 overlapping** coupling with **implicit-explicit Newmark** having **different Δt**
- **QOpInf-FOM** coupled models with M=27 and M=30 POD modes, capturing 99.9999% snapshot energy
- **Prediction** w.r.t. **initial velocity** (rotation speed/direction)
- (Linear) **OpInf-FOM** coupling **insufficient**

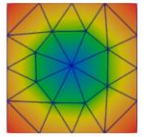
		FOM-FOM	QOpInf-FOM reproductive	QOpInf-FOM predictive
Ω_1 rel errs	displacement	–	2.67e-3	4.32e-2
	velocity	–	3.56e-2	1.49e-1
Ω_2 rel errs	displacement	–	1.13e-3	2.44e-2
	velocity	–	1.12e-2	9.52e-2
CPU time	–	39m 11.8s	1m 40.2s	1m 39.5s
Mean/max # Schwarz iters	–	3.0/3	2.0/2	2.0/2

Relative errors are of $\boldsymbol{O}$**(1e-3)-**$\boldsymbol{O}$**(1e-2)** for displacement and velocity!

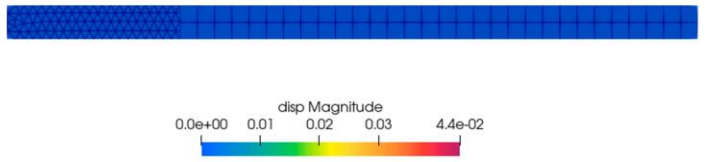


FOM-FOM



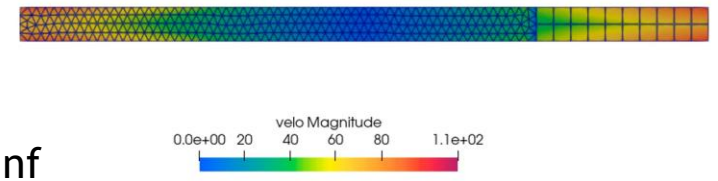


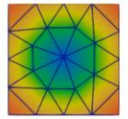
QOpInf = Quadratic OpInf

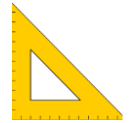




QOpInf-FOM







NORMA.JL

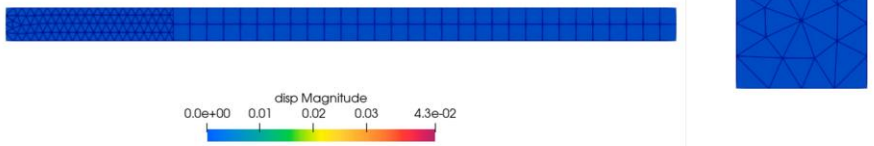
Torsion (Overlapping SAM, Reproductive & Predictive)

- **Neohookean** material model
- **TET4 - HEX8 overlapping** coupling with **implicit-explicit Newmark** having **different Δt**
- **QOpInf-FOM** coupled models with M=27 and M=30 POD modes, capturing 99.9999% snapshot energy
- **Prediction** w.r.t. **initial velocity** (rotation speed/direction)
- (Linear) **OpInf-FOM** coupling **insufficient**

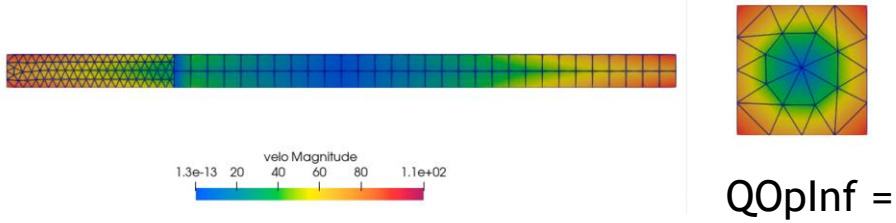
		FOM-FOM	QOpInf-FOM reproductive	QOpInf-FOM predictive
Ω_1 rel errs	displacement	–	2.67e-3	4.32e-2
	velocity	–	3.56e-2	1.49e-1
Ω_2 rel errs	displacement	–	1.13e-3	2.44e-2
	velocity	–	1.12e-2	9.52e-2
CPU time	–	39m 11.8s	1m 40.2s	1m 39.5s
Mean/max # Schwarz iters	–	3.0/3	2.0/2	2.0/2

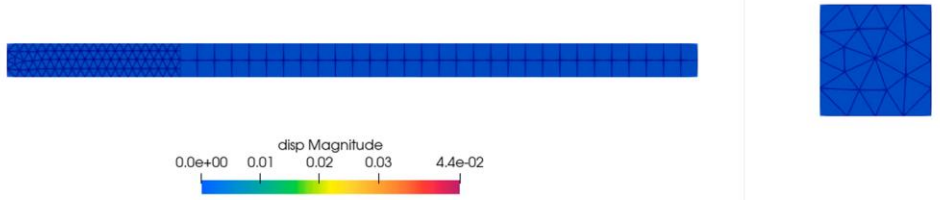
Relative errors are of $\boldsymbol{O}$**(1e-3)-**$\boldsymbol{O}$**(1e-2)** for displacement and velocity!

~**23.5× speedups** are achieved via **QOpInf-FOM** couplings

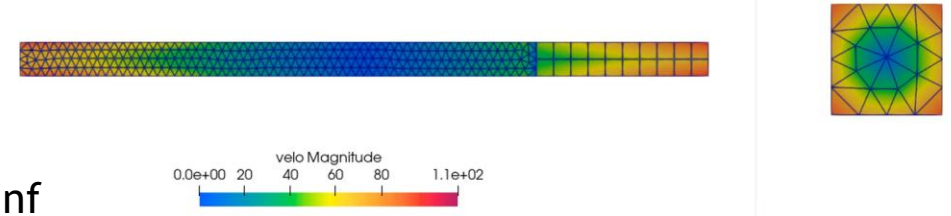


FOM-FOM





QOpInf-FOM



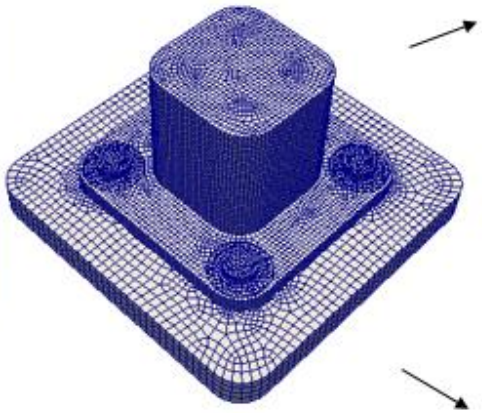
QOpInf = Quadratic OpInf

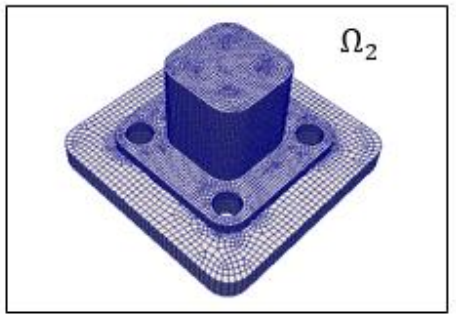
NORMA.JL

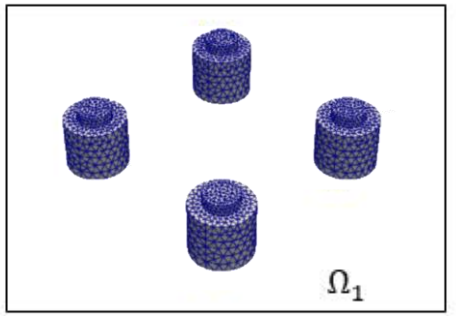
Bolted Joint (Overlapping SAM, Reproductive)

- **Hyper-elastic** (Saint-Venant Kirchhoff) problem with **TET10 - HEX8 meshes**
- **Loading** is applied to top part of bolted join
- **Cubic OpInf (COpInf)** coupled models
- Hybrid couplings require **fewer Schwarz iterations** to converge
- **Convergence** w.r.t. basis size is observed for reproductive FOM-QOpInf couplings (right)









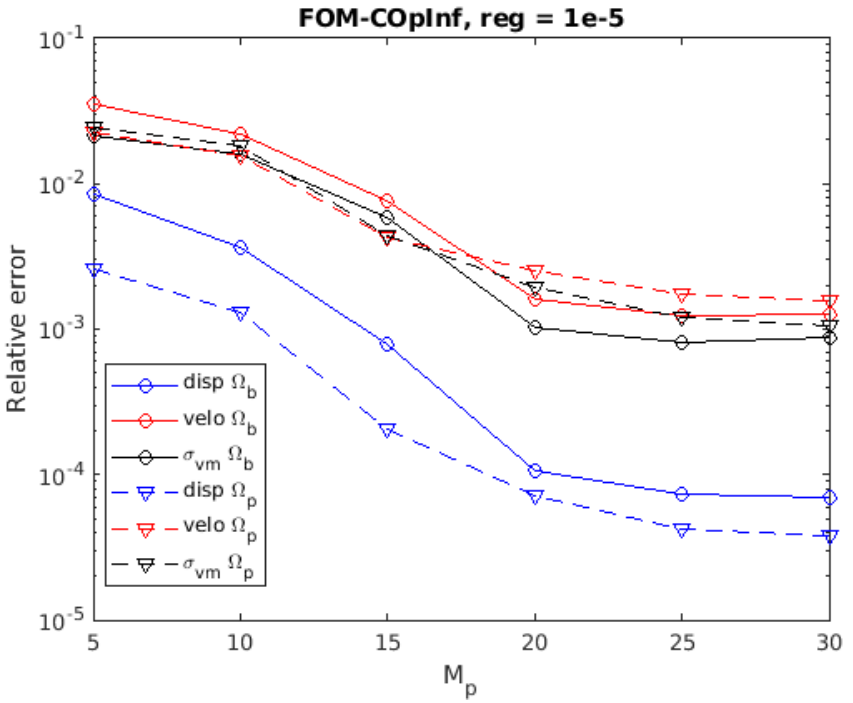
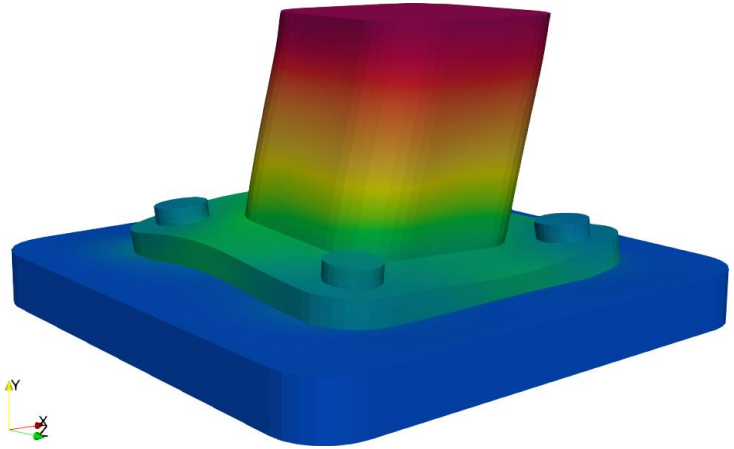


Figure above: convergence w.r.t. basis size for reproductive FOM-COpInf coupling

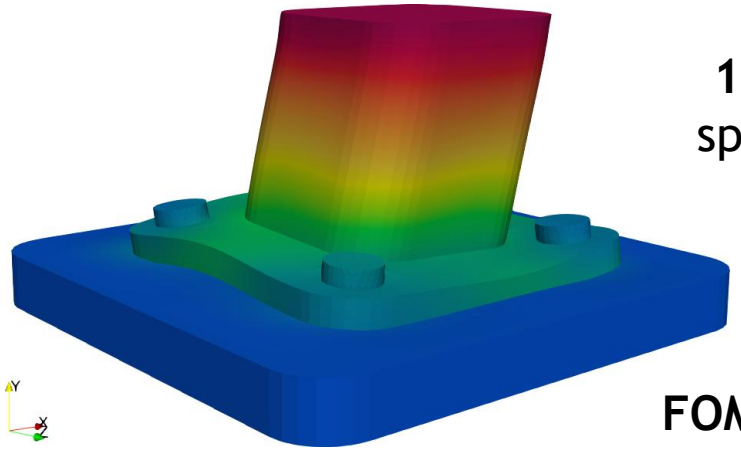
NORMA.JL

Bolted Joint (Overlapping SAM, Predictive): Displacements

COpInf = Cubic OpInf



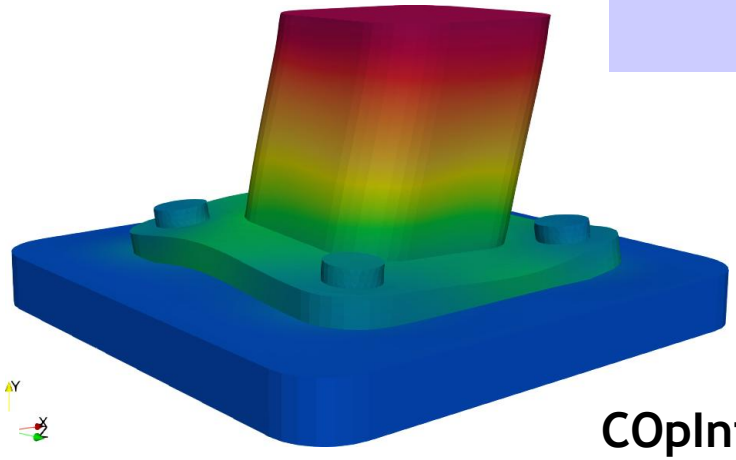
FOM-FOM



1.65×
speedup

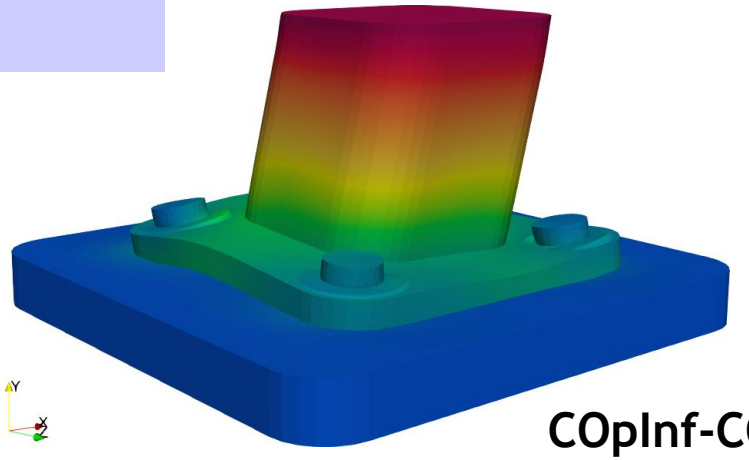
FOM-COpInf
0.69% rel err (parts)
3.51% rel err (bolts)

Models generally achieve **relative errors** of $\boldsymbol{O(1\%)}$ in **displacement.**



6.12×
speedup

COpInf-COpInf (best)
0.98% rel err (parts)
4.41% rel err (bolts)

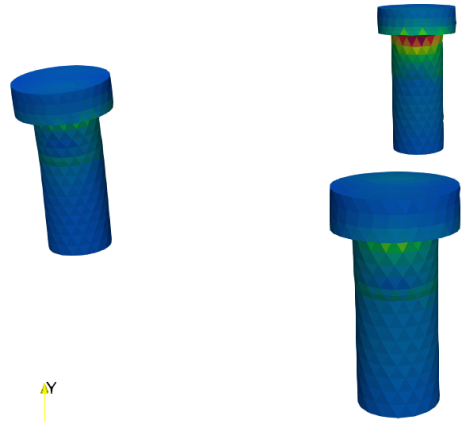


COpInf-COpInf (worst)
12.9% rel err (parts)
12.5% rel err (bolts)

NORMA.JL

Bolted Joint (Overlapping SAM, Predictive): von Mises Stresses

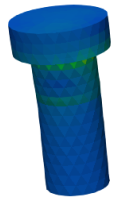
COpInf = Cubic OpInf

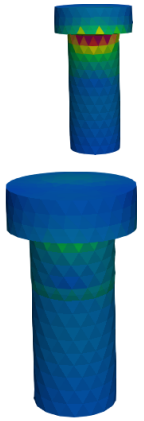


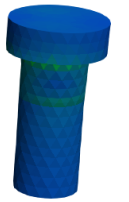
FOM-FOM

1.65× speedup



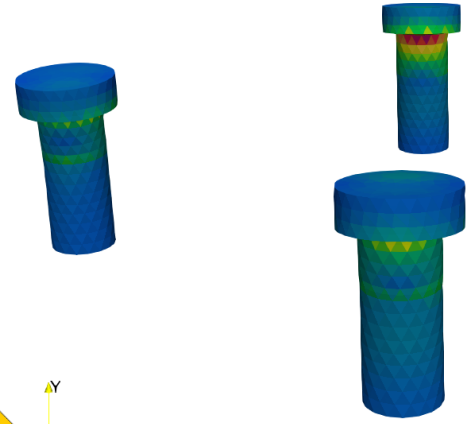






FOM-COpInf
7.74% rel err

All models **correctly predict locations** of **maximum von Mises stress** (failure).

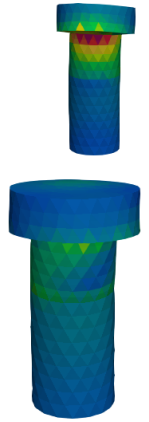


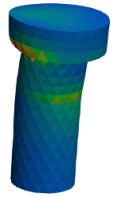
COpInf-COpInf (best)
12.4% rel error

6.12× speedup









COpInf-COpInf (worst)
30.9% rel error

High errors do not necessarily mean result cannot be informative.

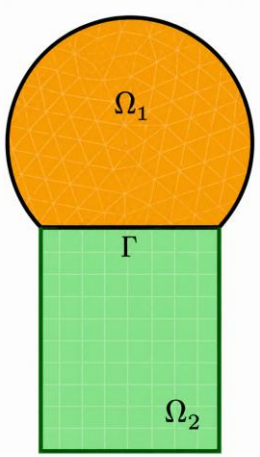
NORMA.JL

Non-overlapping SAM-based Coupling for Non-Intrusive Polynomial OpInf ROMs

Alternating Dirichlet-Neumann non-overlapping SAM (dynamic linear elastic problem)

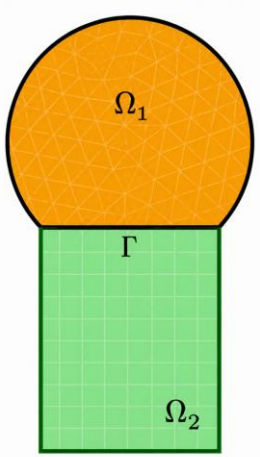
$$\begin{cases} \text{Div}\, \boldsymbol{P}_1^{(n+1)} = \rho \ddot{\boldsymbol{\varphi}}_1^{(n+1)}, & \text{in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\varphi}_2^{(n)}, & \text{on } \Gamma \end{cases}$$

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



Non-overlapping SAM-based Coupling for Non-Intrusive Polynomial OpInf ROMs

Alternating Dirichlet-Neumann non-overlapping SAM (dynamic linear elastic problem)



$$\begin{cases} \text{Div}\, \boldsymbol{P}_1^{(n+1)} = \rho \ddot{\boldsymbol{\varphi}}_1^{(n+1)}, & \text{in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\varphi}_2^{(n)}, & \text{on } \Gamma \end{cases}$$

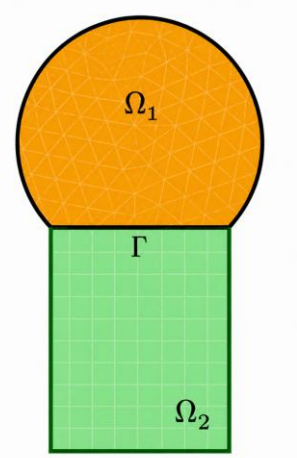
$$\begin{cases} \text{Div}\, \boldsymbol{P}_2^{(n+1)} = \rho \ddot{\boldsymbol{\varphi}}_2^{(n+1)}, & \text{in } \Omega_2 \\ \boldsymbol{\varphi}_2^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_2 \backslash \Gamma \\ \boldsymbol{P}_2^{(n+1)} \boldsymbol{n} = \boldsymbol{P}_1^{(n+1)} \boldsymbol{n},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

$$\int_I \left[\int_{\Omega_1} \left(\rho \ddot{\boldsymbol{\varphi}}_1^{(n+1)} \cdot \boldsymbol{\xi} + \boldsymbol{P}_1^{(n+1)} : \nabla \boldsymbol{\xi} \right) d\Omega \right] dt = 0$$

$$\int_I \left[\int_{\Omega_2} \left(\rho \ddot{\boldsymbol{\varphi}}_2^{(n+1)} \cdot \boldsymbol{\xi} + \boldsymbol{P}_2^{(n+1)} : \nabla \boldsymbol{\xi} \right) d\Omega - \int_{\Gamma} \left(\boldsymbol{P}_1^{(n+1)} \boldsymbol{n} \right) \cdot \boldsymbol{\xi} dS \right] dt = 0$$

Non-overlapping SAM-based Coupling for Non-Intrusive Polynomial OpInf ROMs

Alternating Dirichlet-Neumann non-overlapping SAM (dynamic linear elastic problem)



$$\begin{cases} \text{Div}\, \boldsymbol{P}_1^{(n+1)} = \rho \ddot{\boldsymbol{\varphi}}_1^{(n+1)}, & \text{in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, & \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\varphi}_2^{(n)}, & \text{on } \Gamma \end{cases}$$

$$\int_I \left[\int_{\Omega_1} \left(\rho \ddot{\boldsymbol{\varphi}}_1^{(n+1)} \cdot \boldsymbol{\xi} + \boldsymbol{P}_1^{(n+1)} : \nabla \boldsymbol{\xi} \right) d\Omega \right] dt = 0$$

$$\widehat{\boldsymbol{M}}_1 \ddot{\widehat{\boldsymbol{q}}}_1 + \widehat{\boldsymbol{K}}_1 \widehat{\boldsymbol{q}}_1 + \widetilde{\boldsymbol{B}}_1 \boldsymbol{g}_1 = \boldsymbol{0}$$

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

$$\int_I \left[\int_{\Omega_2} \left(\rho \ddot{\boldsymbol{\varphi}}_2^{(n+1)} \cdot \boldsymbol{\xi} + \boldsymbol{P}_2^{(n+1)} : \nabla \boldsymbol{\xi} \right) d\Omega - \int_\Gamma \left(\boldsymbol{P}_1^{(n+1)} \boldsymbol{n} \right) \cdot \boldsymbol{\xi} dS \right] dt = 0$$

$$\widehat{\boldsymbol{M}}_2 \ddot{\widehat{\boldsymbol{q}}}_2 + \widehat{\boldsymbol{K}}_2 \widehat{\boldsymbol{q}}_2 + \widetilde{\boldsymbol{B}}_2 \boldsymbol{g}_2 + \widetilde{\boldsymbol{H}}_2 \boldsymbol{t}_2 = \boldsymbol{0}$$

OpInf ROMs

Dirichlet problem: learn $\widehat{\boldsymbol{M}}_1$, $\widehat{\boldsymbol{K}}_1$, $\widetilde{\boldsymbol{B}}_1$

- Use snapshots of displacement, acceleration and Dirichlet data to train ROM

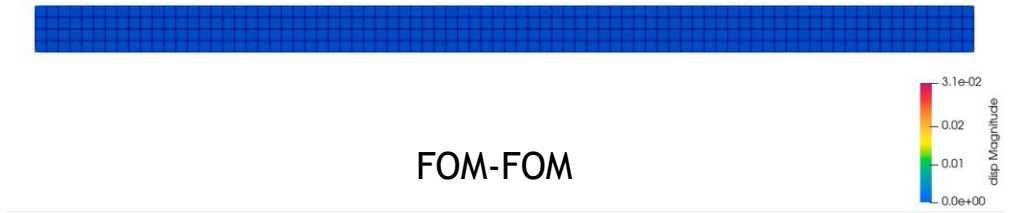
Neumann problem: learn $\widehat{\boldsymbol{M}}_2$, $\widehat{\boldsymbol{K}}_2$, $\widetilde{\boldsymbol{B}}_2$, $\widetilde{\boldsymbol{H}}_2$

- Use snapshots of displacement, acceleration, Dirichlet data and tractions to train ROM

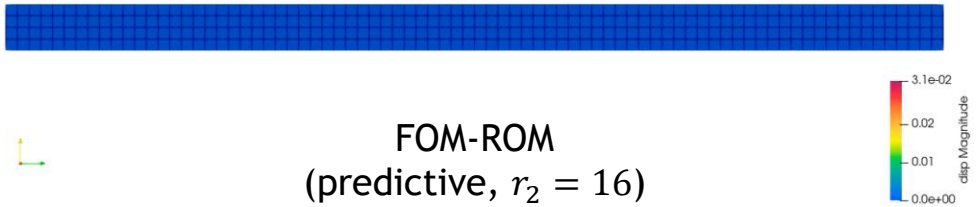
Torsion problem (Non-Overlapping SAM, Predictive & Reproductive)

- 3D Neohookean ***torsion problem*** w/ parameterized initial velocity
- Test non-overlapping SAM w/ ***2 subdomains*** discretized with ***conformal HEX8 meshes***, ***implicit-explicit*** Newmark with ***different Δt***
- Requires ***per-operator regularization parameters***:

$$\min_{\hat{A},\hat{H},\hat{B}} \left\| \dot{\hat{Q}}^T + \hat{Q}^T \hat{A}^T + (\hat{Q} \otimes \hat{Q})^T \hat{H}^T + \hat{U}^T \hat{B}^T \right\|_F^2 + \gamma_A \|\hat{A}\|_F + \gamma_H \|\hat{H}\|_F + \gamma_B \|\hat{B}\|_F$$



FOM-FOM



FOM-ROM
(predictive, $r_2 = 16$)

Good accuracy is achieved for QOpInf-QOpInf and QOpInf-FOM couplings in both the **reproductive** and **predictive regime** with **speedup** up to **2.16×**.

Reproductive

Model	r_1/r_2	Domain 1 disp. error	Domain 2 disp. error	Max Iterations	CPU time
FOM-FOM	-/-	-	-	2	10m 30.99s
QOpInf-FOM QOpInf-QOpInf	5/- 5/5	4.6867e-2 6.9680e-2	5.7994e-2 1.7522e-1	2 2	9m 21.32s 7m 2.13s
QOpInf-FOM QOpInf-QOpInf	15/- 15/16	4.7315e-2 4.6309e-2	4.4132e-2 8.4510e-2	2 1	9m 20.10s 6m 19.62s

Predictive

Model	r_1/r_2	Domain 1 disp. error	Domain 2 disp. error	Max Iterations	CPU time
FOM-FOM	-/-	-	-	2	10m 30.99s
QOpInf-FOM QOpInf-QOpInf	5/- 5/5	1.8201e-1 1.3647e-1	9.8893e-2 5.4770e-2	2 1	5m 57.66s 4m 43.67s
QOpInf-FOM QOpInf-QOpInf	15/- 15/16	1.3262e-1 8.5193e-2	7.7476e-2 1.4832e-2	2 1	6m 1.60s 4m 52.07s

Non-Intrusive NN-based OpInf (NN-OpInf) ROMs


E. Parish


A. Gruber


P. Blonigan

- **Key Polynomial OpInf Deficiency**: functional form is limited to **low-order polynomial** and has **no structure preservation**
- **Mitigation:** develop a **nonlinear NN-parameterized OpInf ROM**, equipped with **structure preservation**
- **Key Idea of NN-OpInf:** parameterize reduced velocity (for solid mechanics) with a **sum of NN operators**, each constrained to satisfy prescribed structure

$$\dot{\hat{\boldsymbol{x}}}(t,\mu) = \sum_{r=1}^{M} N_r(\hat{\boldsymbol{x}},\mu;\theta_r)$$

- **NN-OpInf Regression Problem:**

$$\min_{\{\theta_r\}} \sum_{i,j} \left\| \sum_{r=1}^{M} N_r\left(\hat{\boldsymbol{x}}_j^{(i)},\mu_i;\theta_r\right) - \dot{\hat{\boldsymbol{x}}}_j^{(i)} \right\|_2^2 + R(\{\theta_r\})$$

Example Parameterizations

- **Matrix:** $N_r(\hat{\boldsymbol{x}},\mu;\theta_r) = \boldsymbol{A}(\hat{\boldsymbol{x}},\mu;\theta)\hat{\boldsymbol{x}}$
- **SPSD:** $N_r(\hat{\boldsymbol{x}},\mu;\theta_r) = \boldsymbol{L}(\hat{\boldsymbol{x}},\mu;\theta)\boldsymbol{L}(\hat{\boldsymbol{x}},\mu;\theta)^T\hat{\boldsymbol{x}}$
- **Skew:** $N_r(\hat{\boldsymbol{x}},\mu;\theta_r) = (\boldsymbol{S}(\hat{\boldsymbol{x}},\mu;\theta) - \boldsymbol{S}(\hat{\boldsymbol{x}},\mu;\theta)^T)\hat{\boldsymbol{x}}$
- **Potential:** $N_r(\hat{\boldsymbol{x}},\mu;\theta_r) = \nabla_{\hat{\boldsymbol{x}}} L_\theta(\hat{\boldsymbol{x}},\mu)$

Mitigates training and **robustness issues** of standard NN operator inference through the use of **composable, algebraic structure-preserving operators**

NN-OpInf

NN-OpInf + Overlapping Schwarz for Solid Dynamics

- **Subdomain-Local Models:**

$$\overline{\boldsymbol{M}}_i \ddot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{f}}_i(\hat{\boldsymbol{x}}_i, \mu) + \overline{\boldsymbol{B}}_i \boldsymbol{g}_{\Gamma,i}, \quad \text{for } i = 1,2$$

where $\boldsymbol{g}_{\Gamma,i}$ is interface Dirichlet data received from adjacent subdomain

- **Composable NN-OpInf Formulation:**

$$\hat{\boldsymbol{f}}_i(\hat{\boldsymbol{x}}_i, \mu) = \hat{\boldsymbol{f}}_i^{\text{pot}}(\hat{\boldsymbol{x}}_i, \mu), \qquad \overline{\boldsymbol{B}}_i = \widehat{\boldsymbol{B}}_i^{\text{mat}}(\mu)$$

$$\hat{\boldsymbol{f}}_i^{\text{pot}} = \nabla_{\hat{\boldsymbol{x}}_i}\left(\hat{\boldsymbol{x}}_i^{\mathsf{T}} \boldsymbol{L}_{\theta,i}(\hat{\boldsymbol{x}}_i, \mu) \boldsymbol{L}_{\theta,i}(\hat{\boldsymbol{x}}_i, \mu)^{\mathsf{T}} \hat{\boldsymbol{x}}_i\right)$$

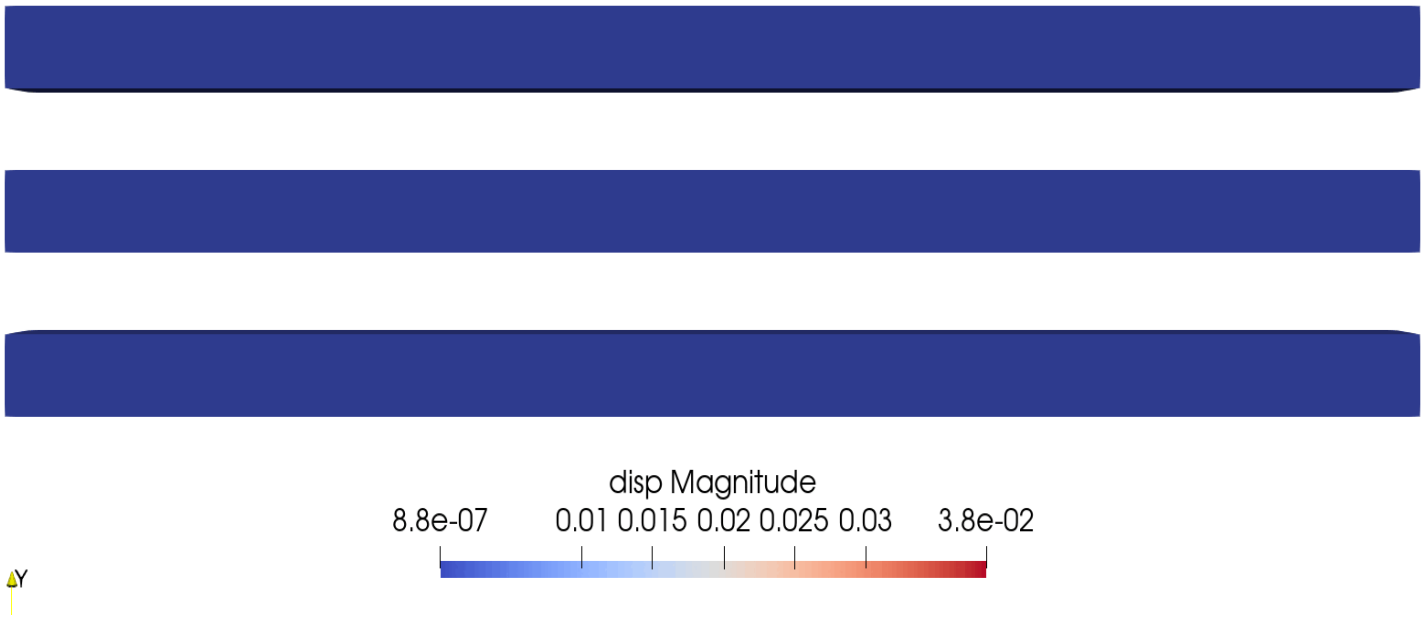
  - **Potential operator** enforces **Lagrangian structure** on primary dynamics
  - **Matrix BC operator** $\widehat{\boldsymbol{B}}_i^{\text{mat}}(\mu)$ isolates and learns **boundary-to-interior coupling**
  - Result is a **composable surrogate**: physics-structured dynamics + explicit interface forcing

NN-OpInf

Fully-differentiable open-source Python software implementation in nnopinf library:

https://sandialabs.github.io/nnopinf/

Torsion Problem (NN-OpInf + O-SAM, Prediction in Time)



NN-OpInf

NORMA.JL

- **3D hyper-elastic torsion problem**, FOM-ROM vs. FOM-FOM coupling, **predictive in time**
- Coupling involving **NN-OpInf** is **accurate** and **stable** past the training horizon (top)
- **Quadratic OpInf**-based coupling quickly goes **unstable** (bottom)

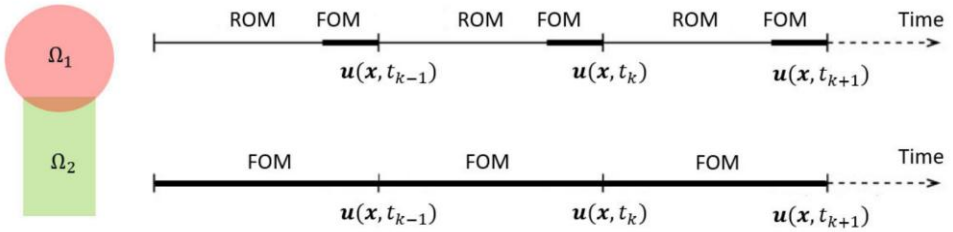
Towards Adaptive SAM-based Coupling: RL formulation



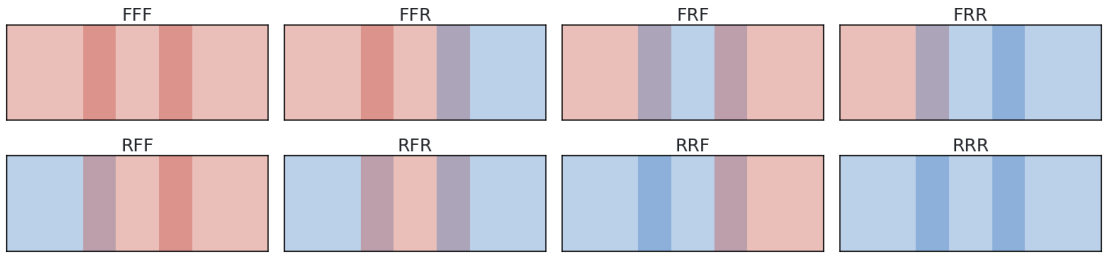
T. Mondal

Goal: on-the-fly **model switching/mesh adaptation** if inaccuracies are detected during the **SAM workflow.**

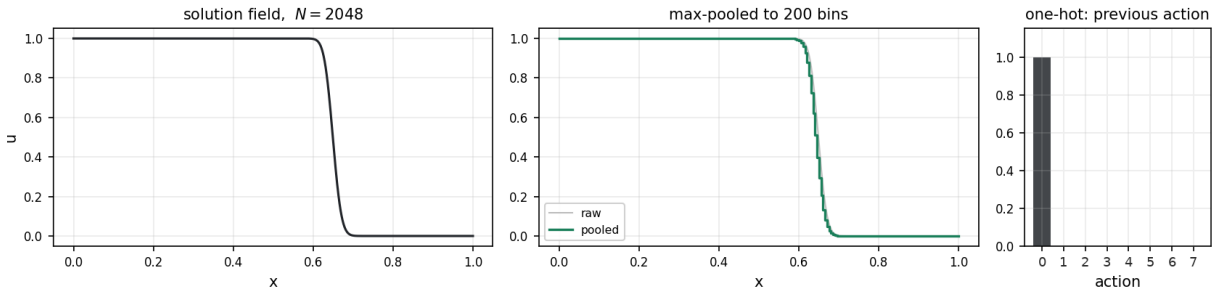
- **Challenge:** FOM/ROM assignment is usually fixed *a priori*, but, in transient problems, localized features (shocks, waves) may **propagate** across the domain
- **Key idea:** Formulate a **sequential decision problem** solved with **reinforcement learning (RL),** where SAM solves each subdomain independently, so that its model can be **switched online**
  - **Domain decomposition** fixed **offline**
  - OpInf ROMs are **pre-trained offline** and stored
  - **RL agent** learns ***which*** **combination** to **activate online**
- **State:** **solution field max-pooled** to 200 bins + **previous action**
- **Action:** one **FOM/ROM label per subdomain** (2^N combos for N subdomains), held for a decision window
- **Reward:** $r_k = -\text{cost}_k - \lambda e_k - \mu n_{\text{flip}}$
  - Balances **cost** vs. **relative L^2 error** w.r.t. FOM
  - **Switching penalty** μ prevents chattering
- **Agent:** **Deep Q-Network (DQN)** trained **offline** over full rollouts with randomized ICs and deployed **online** on unseen ICs



Model switching in time for two overlapping subdomains



3 subdomains → 8 actions (red = FOM, blue = ROM)



State: raw field (N = 2048) → 200 max-pooled values + previous action. Previous action 0 corresponds to FFF.

Adaptive SAM-based Coupling: 3D Elastic Wave (Norma.jl)

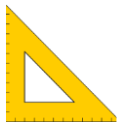
- **Problem:** 3D **linear elastic clamped beam** with Gaussian initial condition for z-displacement

$$u_z(\boldsymbol{x}, 0) = a\exp\left(-\frac{(z - z_0)^2}{2s^2}\right)$$

  - Solved in **Norma.jl**
  - Dynamics are effectively **1D**

- **Domain decomposition:** **3 overlapping subdomains,** $\Omega_1 = (-0.5, 0.1)$, $\Omega_2 = (-0.2, 0.2), \Omega_3 = (0.1, 0.5) \rightarrow$ 8 actions

- **Reward:**
  - **Error** measured in u, v, a vs. all-FOM
  - **Cost** = measured wall time



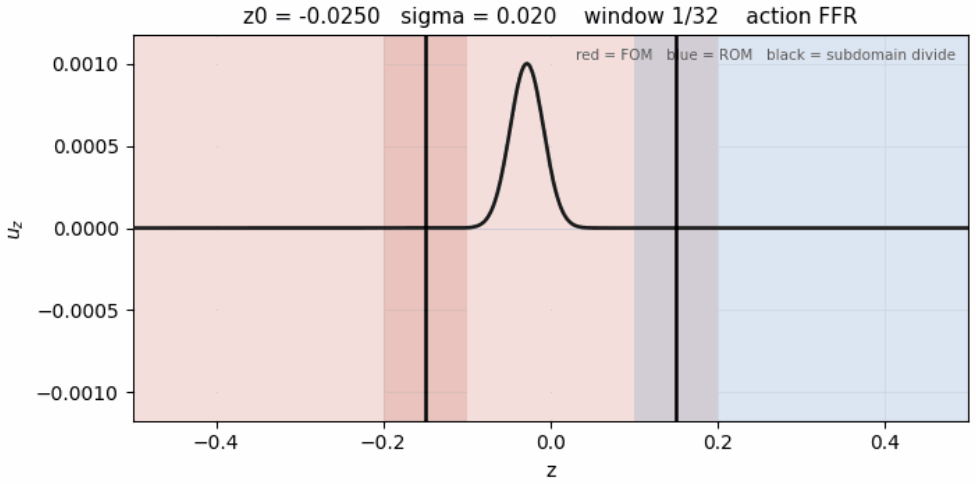
NORMA.JL

- **Training:**
  - **21 problem instances:** 7 pulse positions $z_0 \in [-0.05, 0.1]$ and 3 widths $s \in \{0.02, 0.03, 0.04\}$)
  - **Train** on 11 problem instances, **test** on 10
  - 5000 DQN **episodes**

- **Result (unseen test case: $z_0 = -0.025, s = 0.02$):**
  - Every **fixed hybrid assignment** has **40-220× larger error** ($\geq 1.5 \times 10^{-1}$), including the cheaper RRR / RRF couplings

Learned policy vs. all 8 fixed FOM/ROM assignments

Policy	Relative L^2 error	Wall time [s]
RL (8 actions)	**3.90e-3**	**54.04**
FFF	0	62.63
FFR	1.54e-1	60.03
FRF	2.23e-1	67.53
RRF	2.58e-1	42.42
FRR	2.58e-1	55.72
RRR	2.94e-1	42.10
RFF	4.05e-1	61.39
RFR	8.48e-1	50.22

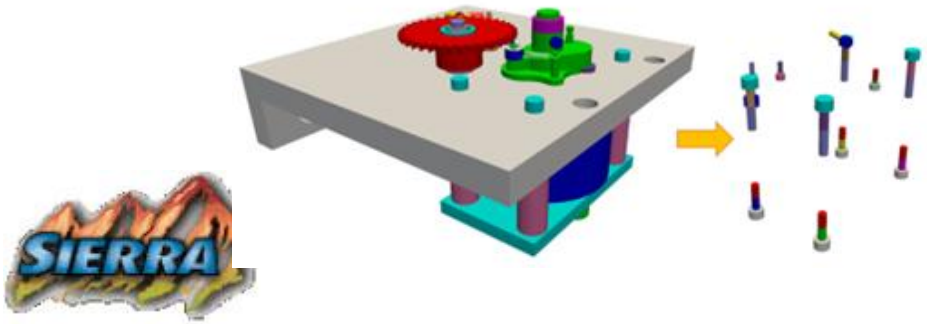


Same test case: FOM (red) follows the pulse across Ω_1 – Ω_3, ROM (blue) elsewhere

Ongoing & Future Work

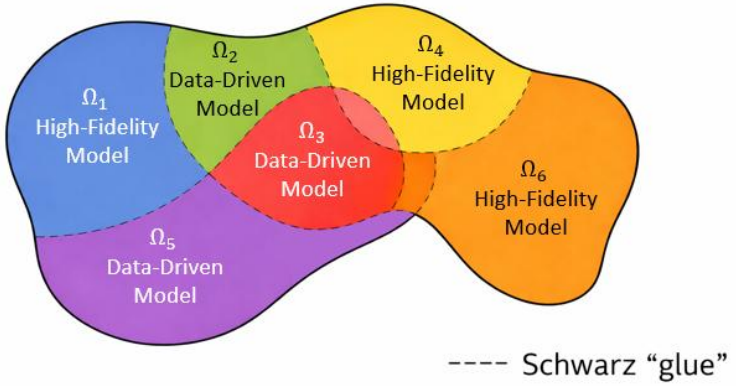
Performance Improvements

- Acceleration of Schwarz via optimized Robin-Robin TCs
- Acceleration of Schwarz via Aitken and Anderson Acceleration



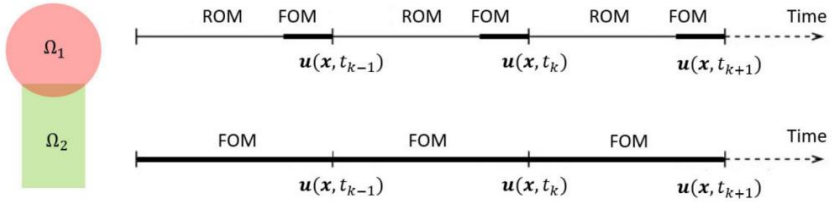
Automated & Adaptive Schwarz

- Automated criteria to determine appropriate use of less refined or reduced-order models w/o sacrificing accuracy
- Online OpInf ROM updates
- On-the-fly switching between ROMs and FOMs
- Collaboration with K. Willcox & L. Gkimisis [UT Austin] and N. Aretz [RPI]



Schwarz for ROM/Data-Driven Model Coupling

- Schwarz + kernel-based ROMs (w/ I. Farcas, Virginia Tech)
- Schwarz + numerical-informed NNs (w/ B. Riviere, Virginia Tech)
- Implementation in Sandia's production code, SIERRA/SM
- Schwarz + multiphysics coupling



References on SAM for FOM-ROM and ROM-ROM Coupling

Schwarz + Polynomial OpInf ROM:

- **I. Tezaur,** E. Parish, A. Gruber, I. Moore, C. Wentland, A. Mota. "Hybrid coupling with operator inference and the overlapping Schwarz alternating method". ArXiv pre-print, 2025. https://arxiv.org/abs/2511.20687
  - **Implementation:** Norma.jl Julia code (https://github.com/sandialabs/Norma.jl)
- C. Rodriguez, **I. Tezaur**, A. Mota, A. Gruber, E. Parish, C. Wentland. "Transmission Conditions for the Non-Overlapping Schwarz Coupling of Full Order and Operator Inference Models", CSRI Summer Proceedings 2025, SNL. https://arxiv.org/abs/2509.12228

Schwarz + Structure-Preserving NN-OpInf:

- E. Parish, A. Gruber, P. Blonigan, **I. Tezaur.** "NN-OpInf: an operator inference approach using structure-preserving composable neural networks", *J. Comput. Phys.*, 569 115505, 2027.
  - **Implementation:** nnopinf Python code (https://sandialabs.github.io/nnopinf/)

Adaptive Schwarz via RL:

- T. Mondal, **I. Tezaur**, A.Gruber. Adaptive hybrid coupling with operator inference, the overlapping Schwarz alternating method and reinforcement learning", CSRI Summer Proceedings 2026, SNL. https://arxiv.org/abs/2609.17837


I. Tezaur


A. Mota


I. Moore


E. Parish


C. Wentland


A. Gruber

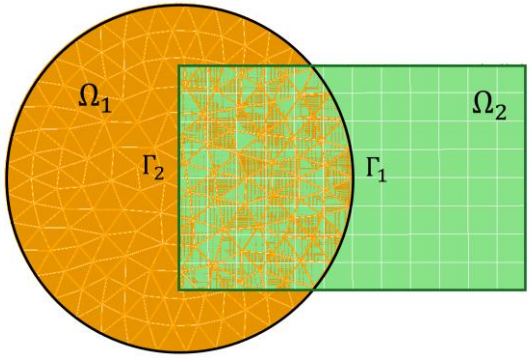

C. Rodriguez


P. Blonigan

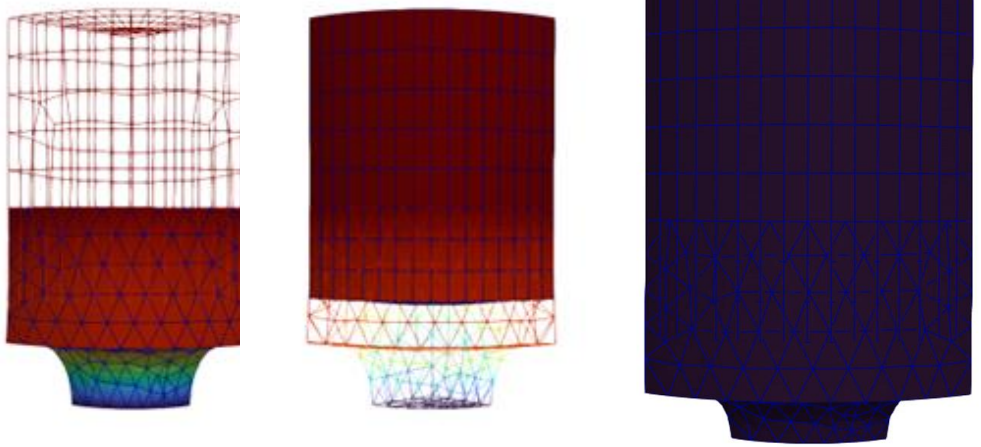

A. Diaz

Contact

NORMA.JL



SIERRA



Ω_1 Ω_2

ikalash@sandia.gov
www.sandia.gov/~ikalash

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT
ASC
U.S. DEPARTMENT OF ENERGY
Office of Science

Call for Papers: CiSE Special Issue on Hybrid Modeling

Computing in Science & Engineering (CiSE) Special Issue

Hybrid Scientific Simulation: Rigorous Integration of Data-Driven and Physics-Based Models

Data-driven models as components of larger predictive simulations, not standalone surrogates

IEEE Computing in Science & Engineering

Topics of interest include:

- **Hybrid domain-decomposition** methods
- **Adaptive simulation** & model switching
- **Learned closures** & constitutive models
- **Multifidelity & multiscale** (DA, inverse, UQ)
- **Correction & discrepancy** modeling
- **Digital twins** & experiment-simulation
- **Stability & structure preservation**
- **V&V, robustness** & benchmarks
- **Production-scale** algorithms & software
- **Applications & case studies**

Key dates

March 24, 2027
Submission deadline

Q4 2027
Publication

Types of papers welcome:

1. **Research articles**

   New hybrid coupling **methods, algorithms, software** & **applications**

   Focus on key innovations & computational insights

2. **Perspective / position articles**

   Opportunities, limitations & **open challenges** of hybrid modeling

   e.g., When is hybrid preferable? How do we trust its predictions? What are gaps?

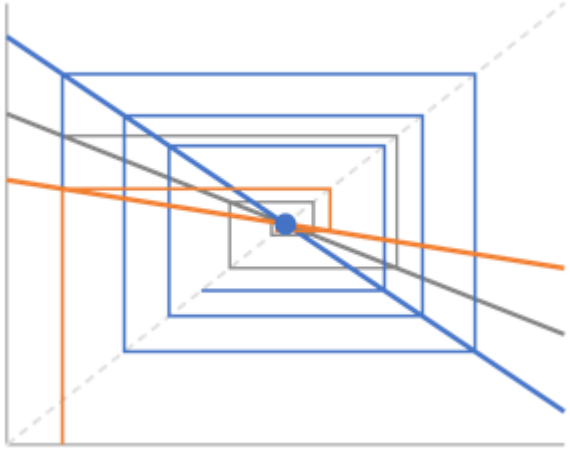
CiSE articles are written for a **broad CSE audience** and emphasize key computational ideas, insights & lessons learned over specialized detail.

Questions? ikalash@sandia.gov

Conjecture: Why OpInf Couplings Need Fewer Iterations

Analysis Approach

- Schwarz = **fixed-point map** on interface data: $\boldsymbol{s}^{(k+1)} = G(\boldsymbol{s}^k)$
- **Value** of G: where iteration lands (accuracy)
- **Slope** of G: how many iterations needed to converge (speed)



Same fixed point, different slopes

True slope: FOM-FOM

Too steep: intrusive projection-based ROM[1], neural operator[2] → more iterations

Too flat: OpInf (conjecture) → fewer iterations

[1]C. Wentland et al. "The role of interface boundary conditions and sampling strategies for Schwarz-based coupling of projection-based reduced order models", *JCAM*, 2025.
[2]Y. Wu. "FE-NO Schwarz coupling: iteration count reduction, *unpublished*, 2026.

Conjecture: OpInf ROM learns right value but wrong slope

- Schwarz BCs enter the OpInf ROM through the learned operator $\widetilde{\boldsymbol{B}}_i$, which sets the ROM's **sensitivity** to its neighbor, i.e., the slope of G.
- However, $\widetilde{\boldsymbol{B}}_i$ is **not learned exactly**: training data come from the FOM-FOM fixed point, so they cannot fully pin down how the ROM should respond to *other* Schwarz BCs.
- **Regularization** penalizes large entries in $\widetilde{\boldsymbol{B}}_i$ → learned $\widetilde{\boldsymbol{B}}_i$ is **too weak** → **flatter slope** → **fewer iterations**

Intrusive ROMs keep the physical coupling ($\widetilde{\boldsymbol{B}}_i$ comes from projection) but model error tends to steepen the slope → more iterations.

Key Insight: OpInf gets the fixed point but an inexact, **too-weak** $\widetilde{\boldsymbol{B}}_i$, so its Schwarz slope is **flatter** than the FOMs, leading to **less iterations.**

Other References on SAM for FOM-ROM and ROM-ROM Coupling

Schwarz Couplings Involving Intrusive Projection-Based ROMs:

- J. Barnett, **I. Tezaur**, A. Mota. "The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models", CSRI Summer Proceedings 2023, Sandia National Laboratories. https://arxiv.org/abs/2210.12551
- C. Wentland, F. Rizzi, J. Barnett, **I. Tezaur**. "The role of interface boundary conditions and sampling strategies for Schwarz-based coupling of projection-based reduced order models", *J. Comput. Appl. Math.*, 465 116584, 2025

Schwarz Couplings Involving Physics-Informed Neural Networks (PINNs):

- W. Snyder, **I. Tezaur**, C. Wentland. "Domain decomposition-based coupling of PINNs via the Schwarz alternating method", CSRI Summer Proceedings 2023, Sandia National Laboratories. https://arxiv.org/abs/2311.00224


I. Tezaur


J. Barnett


I. Moore


E. Parish


C. Wentland


A. Gruber


C. Rodriguez


W. Snyder


G. Sambataro

Start of Backup Slides

Requirements for Multiscale Coupling Method

- Coupling is ***concurrent*** (two-way)
- ***"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
  - Ability to mix-and-match ***conventional*** and ***data-driven models*** in different regions
- ***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!)
- Coupling does not introduce ***nonphysical artifacts***
- ***Theoretical*** convergence properties/ guarantees

*A great method theoretically may not make it if it is **too difficult** to implement in **production codes**.*

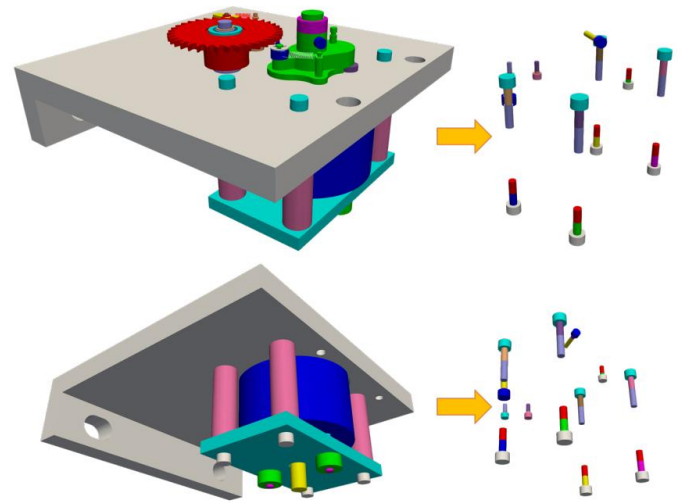
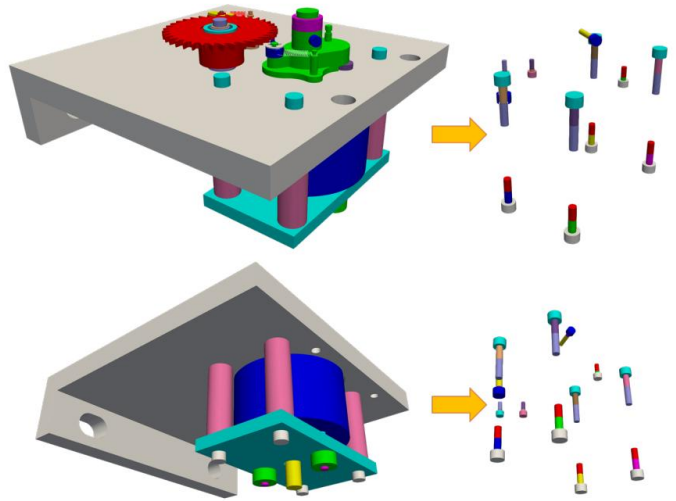


Figure above: schematic of difficult-to-mesh ratcheting mechanism with multiple threaded fasteners. From Parish *et al.*, 2024.

Requirements for Multiscale Coupling Method

- Coupling is ***concurrent*** (two-way)
- ***"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
  - Ability to mix-and-match ***conventional*** and ***data-driven models*** in different regions
- ***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!)
- Coupling does not introduce ***nonphysical artifacts***
- ***Theoretical*** convergence properties/ guarantees

*Creating a **high-quality mesh** can take **weeks** or longer, making it "the **single biggest bottleneck** in [mod/sim] analyses" [Sandia Lab News, 2020]*



Above: schematic of difficult-to-mesh ratcheting mechanism with multiple threaded fasteners. From Parish *et al.*, 2024.

Projects on Coupling for Predictive Hybrid Models

Three projects:

- **FHNM**: Flexible Heterogeneous Numerical Methods [LDRD, FY22-FY24]
- **M2dt**: Multi-faceted Mathematics for Predictive Digital Twins [ASCR, FY23-FY27]
- **AHEAD**: Adaptive Hybrid modEls via domAin Decomposition [LDRD, FY25-FY27]

Principal research objective:

- Develop rigorous methods to enable the **"plug-and-play" coupling** of **standard** and **data-driven models** from the following classes
  - *Class A*: intrusive projection-based ROMs
  - *Class B*: machine-learned models
  - *Class C*: flow map approximation models, i.e., dynamic model decomposition (DMD)
  - *Class D*: non-intrusive operator inference (OpInf) ROMs

Three classes of coupling methods:

- Alternating **Schwarz**-based coupling [FHNM, M2dt, AHEAD]
- **Optimization**-based coupling [FHNM, M2dt]
- Coupling via **generalized mortar methods** [FHNM, M2dt]

Projects on Coupling for Predictive Hybrid Models

Three projects:

- **FHNM**: Flexible Heterogeneous Numerical Methods [LDRD, FY22-FY24]
- **M2dt**: Multi-faceted Mathematics for Predictive Digital Twins [ASCR, FY23-FY27]
- **AHEAD**: Adaptive Hybrid modEls via domAin Decomposition [LDRD, FY25-FY27]

Principal research objective:

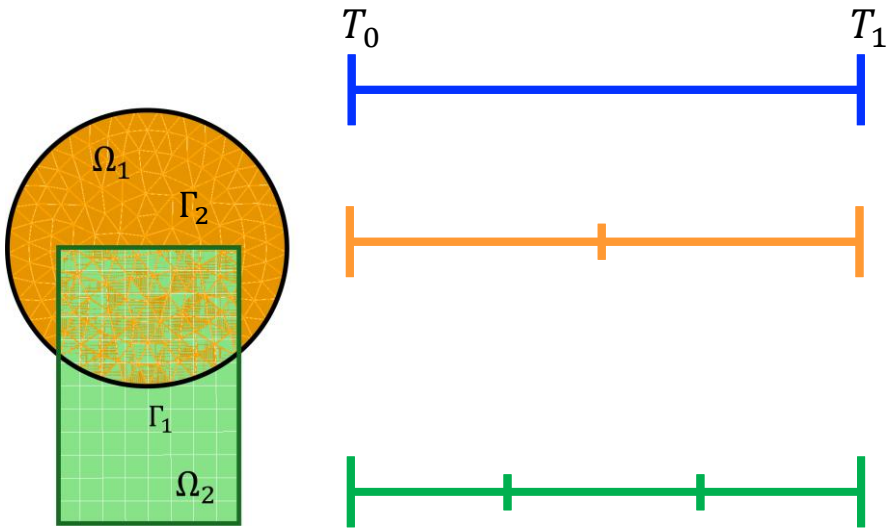
- Develop rigorous methods to enable the **"plug-and-play" coupling** of **standard** and **data-driven models** from the following classes
  - *Class A*: intrusive projection-based ROMs
  - *Class B*: machine-learned models
  - *Class C*: flow map approximation models, i.e., dynamic model decomposition (DMD)
  - *Class D*: non-intrusive operator inference (OpInf) ROMs → this talk

Three classes of coupling methods:

- Alternating **Schwarz**-based coupling [FHNM, M2dt, AHEAD] → this talk
- **Optimization**-based coupling [FHNM, M2dt]
- Coupling via **generalized mortar methods** [FHNM, M2dt]

Why? **Less intrusive** to integrate into **production codes.**

Time-Advancement within the Schwarz Framework



Controller time stepper

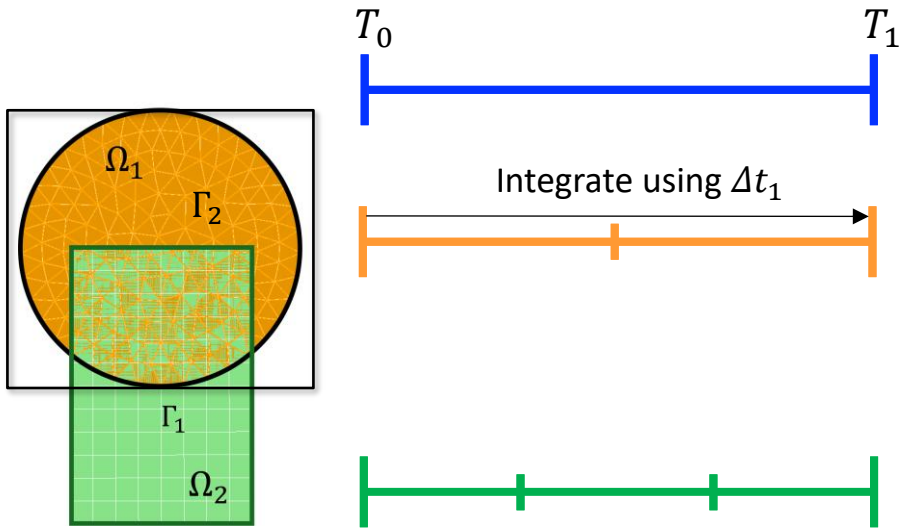
Time integrator in Ω_1

Time integrator in Ω_2

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

Model PDE: $\begin{cases} \boldsymbol{M\ddot{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}} \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \end{cases}$

Time-Advancement within the Schwarz Framework



Controller time stepper

Time integrator in Ω_1

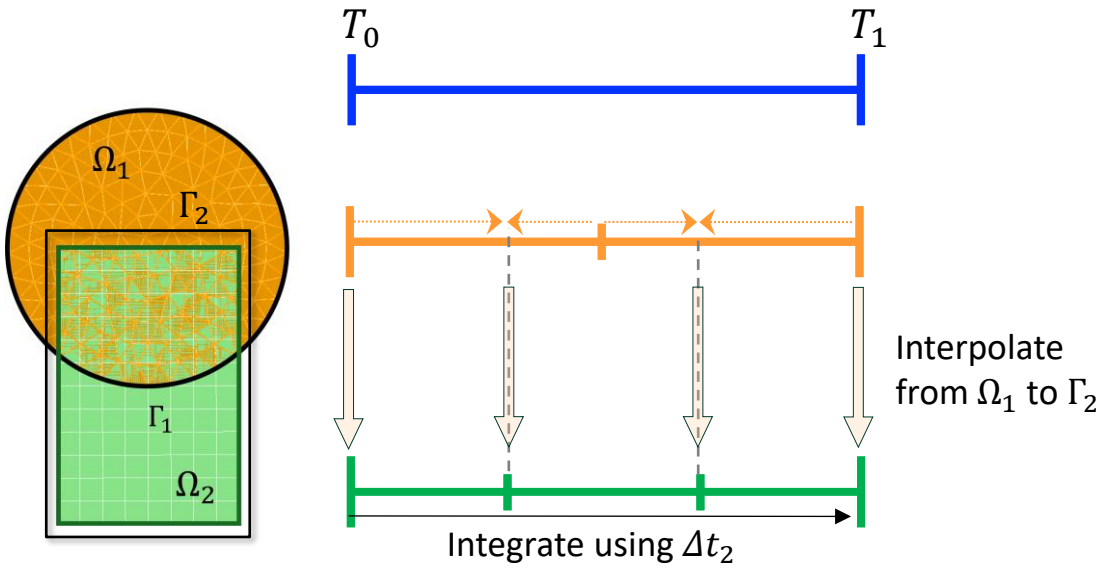
Time integrator in Ω_2

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 $T_i + n\Delta t_1$.

Model PDE: $\begin{cases} \boldsymbol{M\ddot{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}} \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \end{cases}$

Time-Advancement within the Schwarz Framework



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

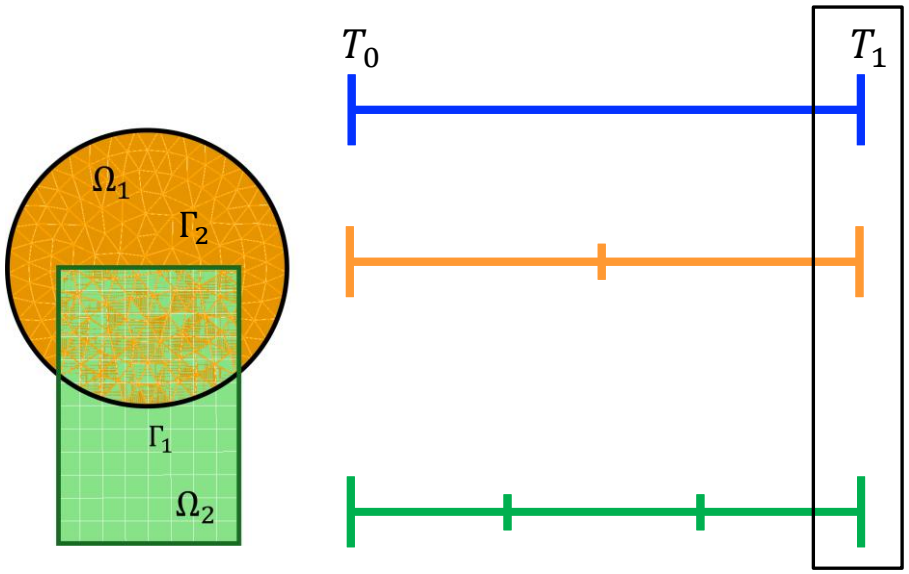
Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

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 $T_i + n\Delta t_2$.

.

Model PDE: $\begin{cases} \boldsymbol{M\ddot{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}} \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \end{cases}$

Time-Advancement within the Schwarz Framework



Controller time stepper

Time integrator in Ω_1

Time integrator in Ω_2

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

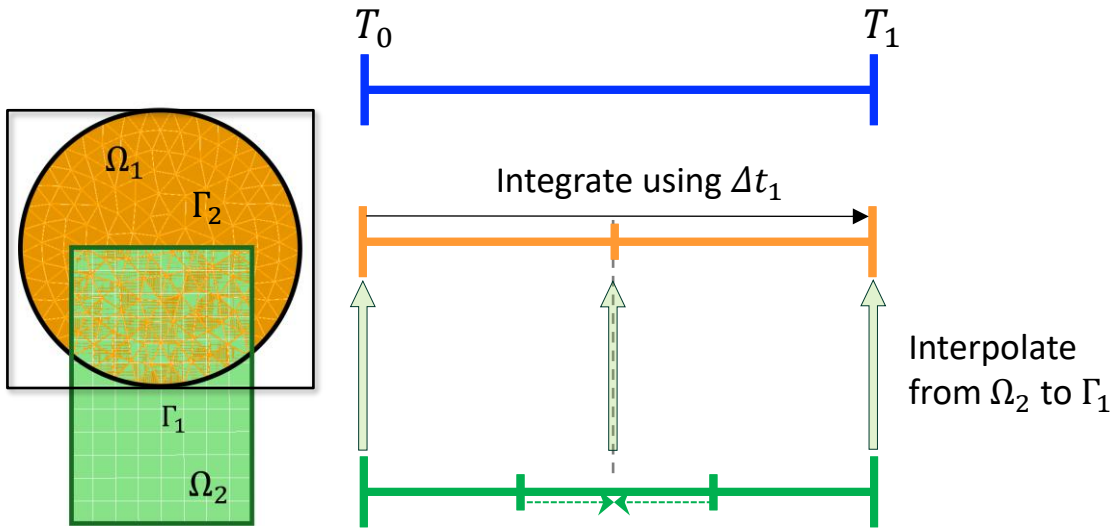
Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1, using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

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}.

Model PDE: $\begin{cases} \boldsymbol{M\ddot{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}} \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \end{cases}$

Time-Advancement within the Schwarz Framework



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 $T_i + n\Delta t_1$.

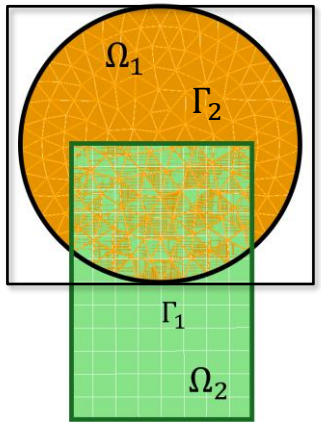
Step 2: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2, using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

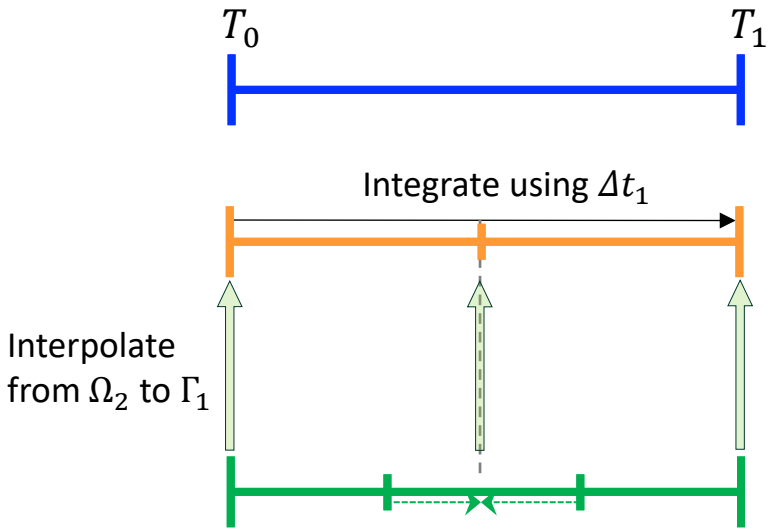
Step 3: Check for convergence at time T_{i+1}.

- If unconverged, return to Step 1.

Model PDE: $\begin{cases} \boldsymbol{M\ddot{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}} \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \end{cases}$

Time-Advancement within the Schwarz Framework





Controller time stepper

Time integrator in Ω_1

Time integrator in Ω_2

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

Time-stepping procedure is ***equivalent*** to doing Schwarz on ***space-time domain*** [Mota, **IT**, *et al.* 2022].

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}.

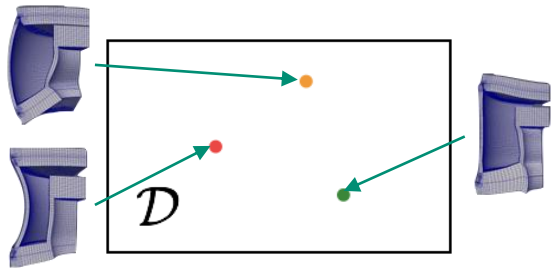
- If unconverged, return to Step 1.
- If converged, set $i = i + 1$ and return to Step 1.

Model PDE: $\begin{cases} \boldsymbol{M\ddot{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}} \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \end{cases}$

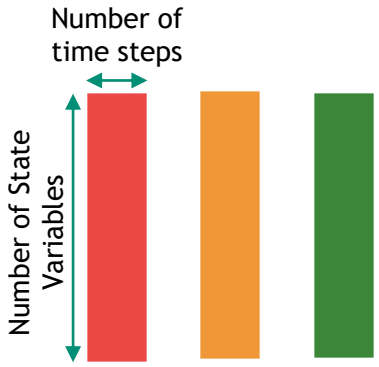
Projection-Based Model Order Reduction via the POD/Galerkin Method

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

1. Acquisition



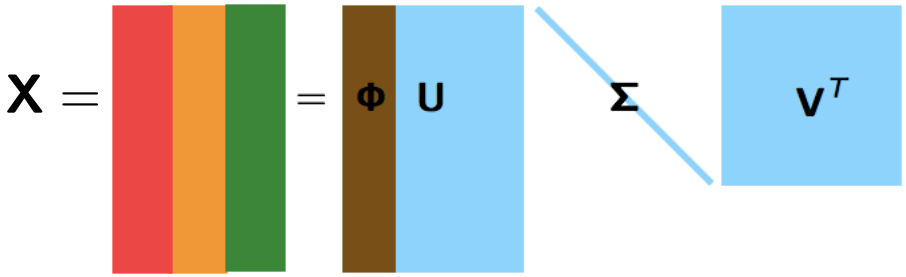
Solve ODE at different design points



Save solution data

2. Learning

Proper Orthogonal Decomposition (POD):



3. Projection-Based Reduction

Reduce the number of unknowns

$$\boldsymbol{u}(t) \approx \tilde{\boldsymbol{u}}(t) = \boldsymbol{\Phi}\hat{\boldsymbol{u}}(t)$$



Perform Galerkin projection

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

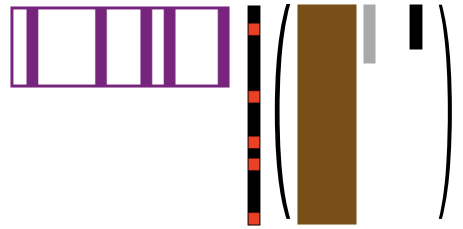
***Disadvantage:** intrusive!*

Hyper-reduce nonlinear terms

$$\boldsymbol{f}_{\text{int}}(\boldsymbol{\Phi}\hat{\boldsymbol{u}}) \approx \boldsymbol{A}\,\boldsymbol{f}_{\text{int}}(\boldsymbol{\Phi}\hat{\boldsymbol{u}})$$



Hyper-reduction/sample mesh

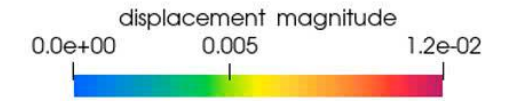


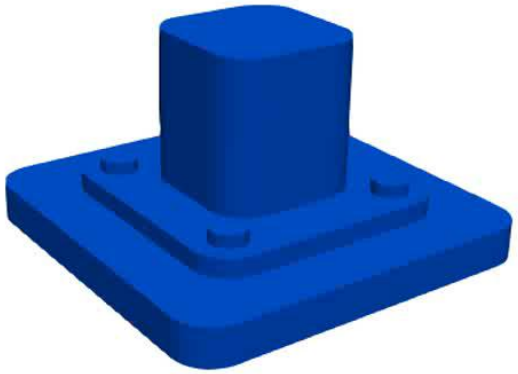
ROM = projection-based Reduced Order Model

HROM = Hyper-reduced ROM

Bolted Joint (Overlapping SAM, Predictive): Animations

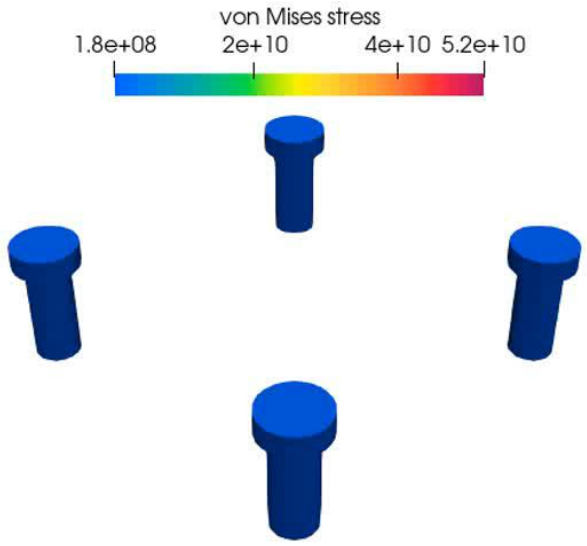
COpInf = Cubic OpInf





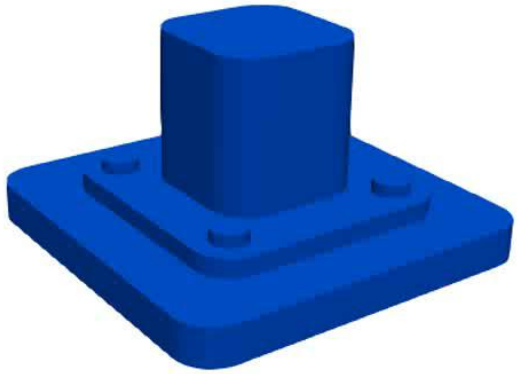


FOM-FOM
~53m



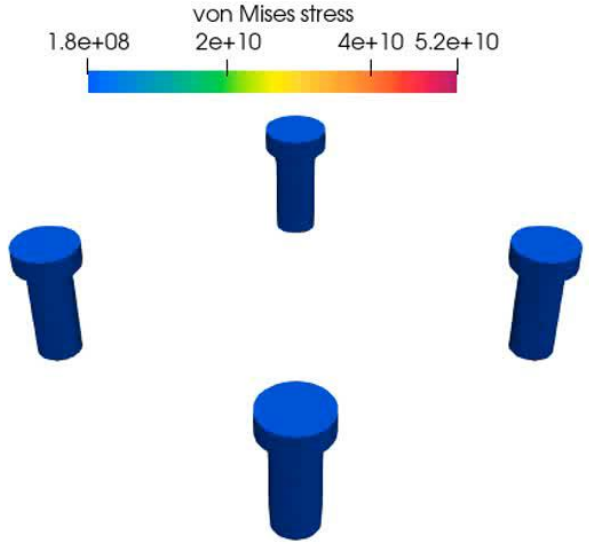








COpInf-COpInf
~8.5m

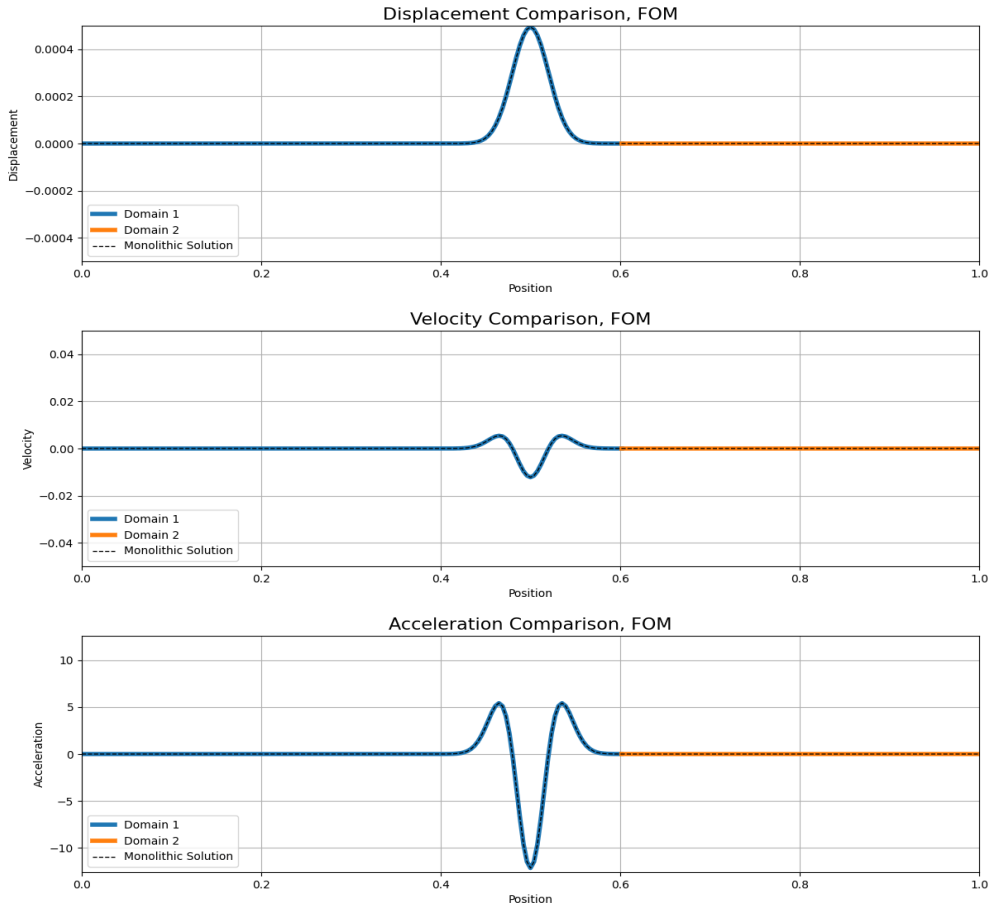




Linear Elastic Wave Propagation (Non-Overlapping SAM, Reproductive)

- 1D linear elastic ***clamped beam*** with Gaussian initial condition for the displacement.
- Simple problem with analytical exact solution but very ***stringent test*** for discretization methods.
- Test non-overlapping SAM w/ ***2 subdomains***: $\Omega_1 = (0,0.6), \Omega_2 = (0.6,1)$.

Transmission Type	Model	M_1 / M_2	Error	Iterations	CPU time (s)
Dirichlet-Neumann	FOM-FOM	– / –	3.43×10^{-4}	3.73	52
Dirichlet-Neumann	OpInf-FOM	20 / –	7.37×10^{-4}	6.08	64
Dirichlet-Neumann	FOM-OpInf	– / 17	7.31×10^{-4}	4.10	47
Dirichlet-Neumann	OpInf-OpInf	20 / 17	9.63×10^{-4}	4.12	37
Dirichlet-Neumann	OpInf-FOM	34 / –	1.29×10^{-4}	5.45	60
Dirichlet-Neumann	FOM-OpInf	– / 29	1.53×10^{-4}	4.19	51
Dirichlet-Neumann	OpInf-OpInf	34 / 29	2.03×10^{-4}	4.10	39



Above: representative OpInf-FOM coupling result

Good accuracy and ***moderate speedups*** over FOM-FOM coupling are possible. Larger speedups expected in ***multi-D***.

Work in Progress: Accelerating Non-Overlapping SAM via Transmission Condition Design



C. Rodriguez
(Columbia U)

- **Risk:** non-overlapping Schwarz can be **slow to converge** w/ **Dirichlet-Neumann TCs**.
- **Mitigation:** explore optimized **Robin-Robin** transmission conditions (TCs).

$$
\begin{cases}
M_1\ddot{u}_1^{(s+1)} + K_1 u_1^{(s+1)} &= F_1^{(s+1)} & \text{in } \Omega_1 \\
u_1^{(s+1)} &= u_D & \text{on } \partial_\Psi \Omega_1 \backslash \Gamma \\
\alpha_{12} T_1^{(s+1)} + \beta_{12} u_1^{(s+1)} &= \lambda_1^{(s+1)} & \text{on } \Gamma
\end{cases}
$$

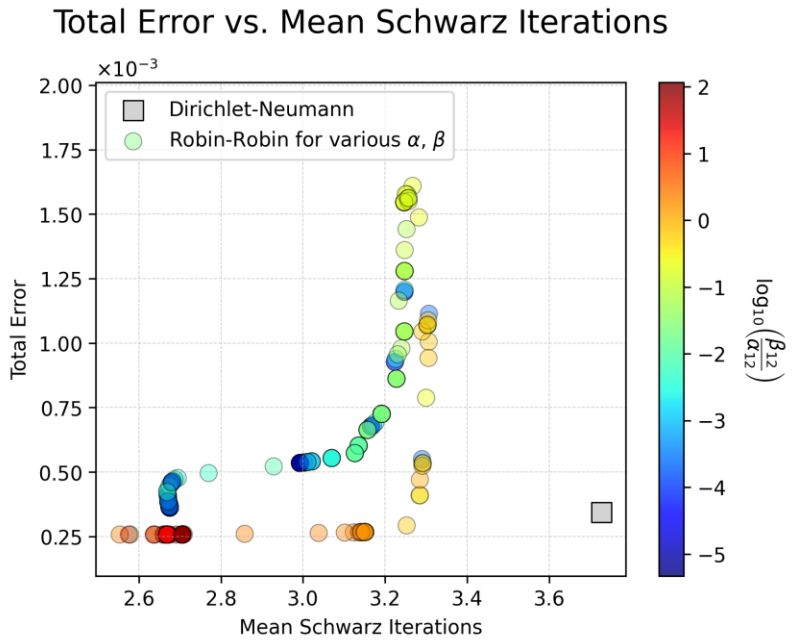
$$
\begin{cases}
M_2\ddot{u}_2^{(s+1)} + K_2 u_2^{(s+1)} &= F_2^{(s+1)} & \text{in } \Omega_2 \\
u_2^{(s+1)} &= u_D & \text{on } \partial_\Psi \Omega_2 \backslash \Gamma \\
\alpha_{21} T_2^{(s+1)} + \beta_{21} u_2^{(s+1)} &= \lambda_2^{(s+1)} & \text{on } \Gamma
\end{cases}
$$

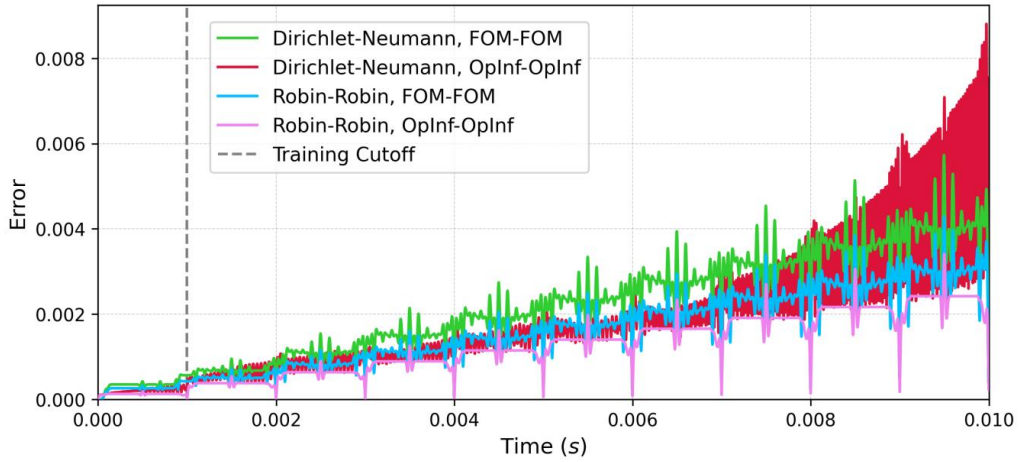
$$\lambda_1^{(s+1)} = \alpha_{12} T_2^{(s)} + \beta_{12} u_2^{(s)}$$

$$\lambda_2^{(s+1)} = \alpha_{21} T_1^{(s+1)} + \beta_{21} u_1^{(s+1)}$$

α_{ij}, β_{ij}: tuning parameters

Automatic tuning of α_{ij}, β_{ij} possible using ***Bayesian optimization*** via ***GPTune***.





Non-overlapping SAM with **optimized Robin-Robin TCs** can give to **faster convergence (29% less Schwarz iterations), lower errors (by up to 25%),** and **more stable/accurate time-predictive ROMs** than non-overlapping SAM with Dirichlet-Neumann TCs.

References on SAM for FOM-ROM and ROM-ROM Coupling (cont'd)

Schwarz Couplings Involving Intrusive Projection-Based ROMs:

- J. Barnett, **I. Tezaur**, A. Mota. "The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models", CSRI Summer Proceedings 2023, Sandia National Laboratories. https://arxiv.org/abs/2210.12551
- C. Wentland, F. Rizzi, J. Barnett, **I. Tezaur**. "The role of interface boundary conditions and sampling strategies for Schwarz-based coupling of projection-based reduced order models", *J. Comput. Appl. Math.*, 465 116584, 2025

Schwarz Couplings Involving Physics-Informed Neural Networks (PINNs):

- W. Snyder, **I. Tezaur**, C. Wentland. "Domain decomposition-based coupling of PINNs via the Schwarz alternating method", CSRI Summer Proceedings 2023, Sandia National Laboratories. https://arxiv.org/abs/2311.00224


I. Tezaur


J. Barnett


I. Moore


E. Parish


C. Wentland


A. Gruber


C. Rodriguez

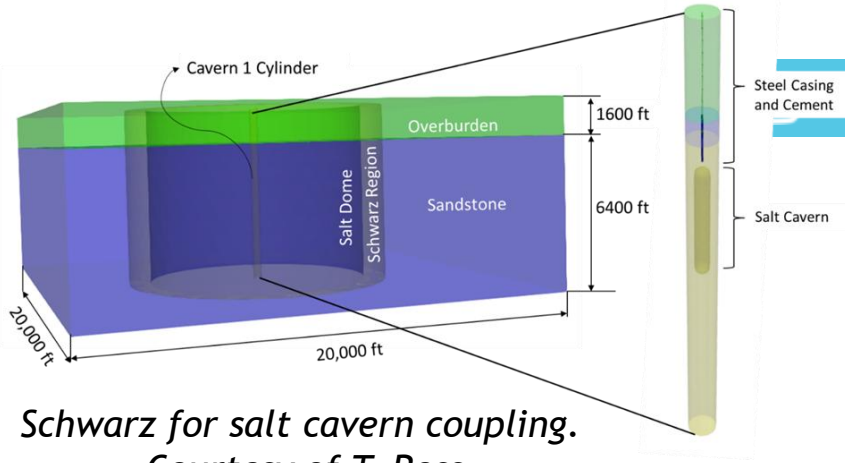

W. Snyder


G. Sambataro

Current Research Directions

Production Applications

- 4-way multiscale coupling in salt caverns for Strategic Petroleum Reserve (SPR)
- Schwarz coupling for J-integral around crack front on pressure vessel
- Multiscale coupling for stronglinks
- Schwarz multiphysics coupling for electronics package survivability



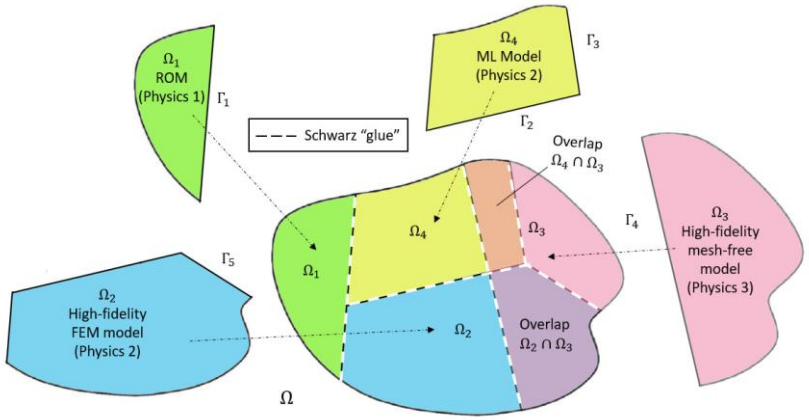
Schwarz for salt cavern coupling. Courtesy of T. Ross.

Performance Improvements

- Acceleration of Schwarz
- Asynchronous additive Schwarz on GPUs

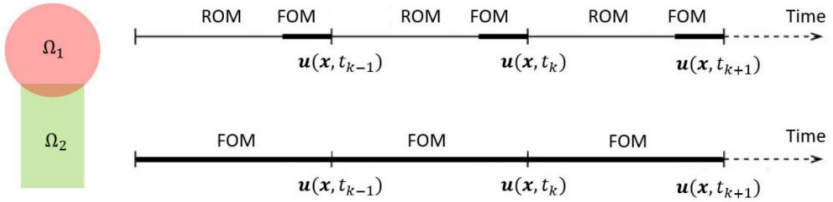
Automated & Adaptive Schwarz

- Automated optimization of meshes/DD with multiple constraints
- Automated criteria to determine appropriate use of less refined or reduced-order models w/o sacrificing accuracy

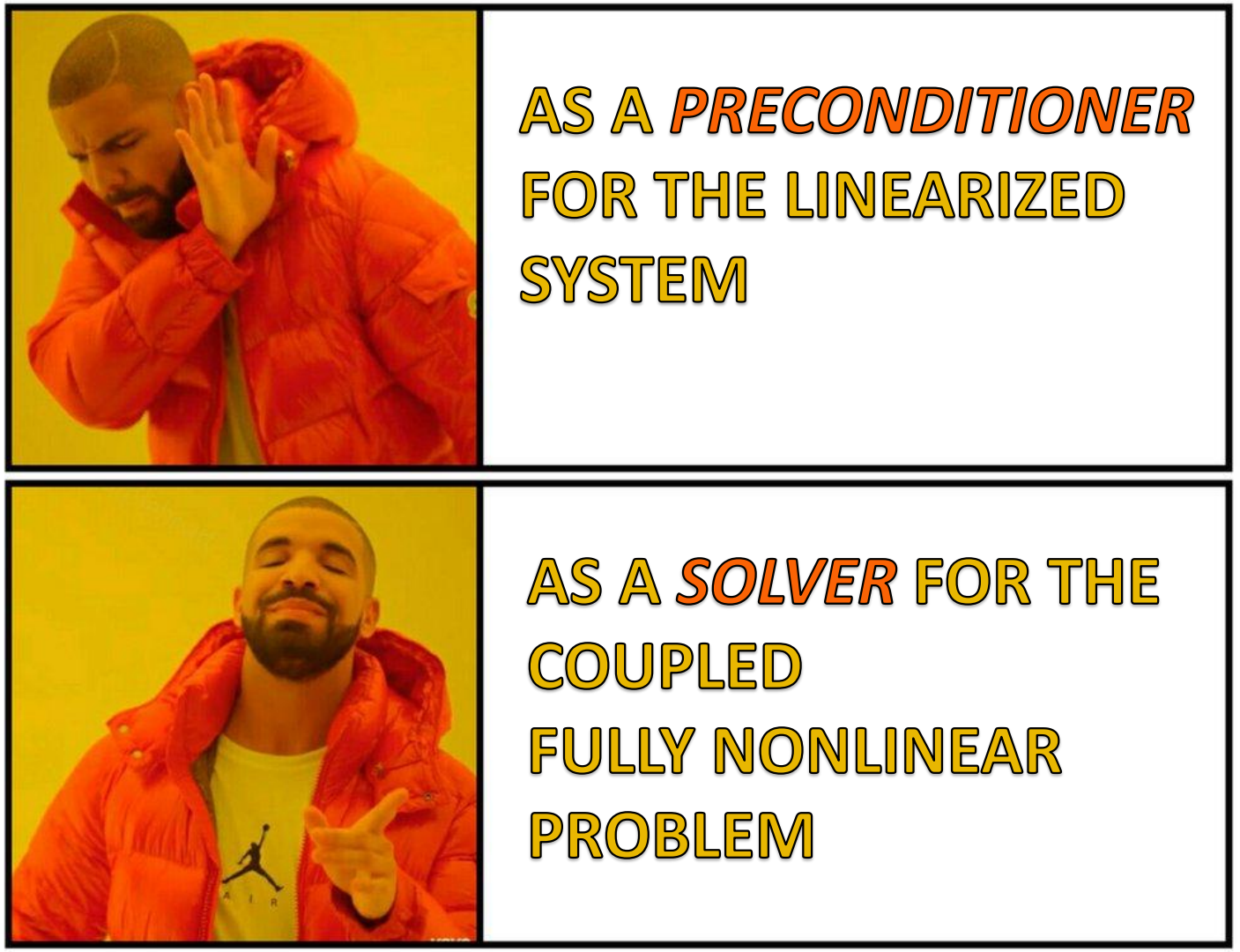


Schwarz for ROM/Data-Driven Model Coupling

- Non-overlapping Schwarz + OpInf
- Schwarz + Deep NN-based ROMs
- Schwarz + kernel-based ROMs
- Fully non-intrusive ROM-FOM coupling (w/ K. Willcox & N. Aretz, UT Austin)
- On-the-fly switching between ROMs and FOMs
- Implementation in SIERRA/SM

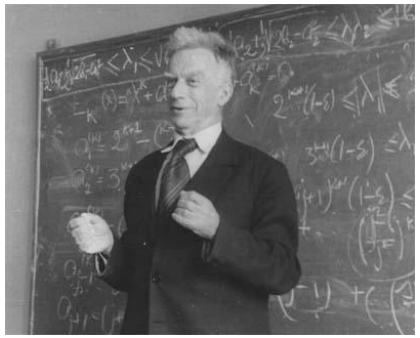


How We Use the Schwarz Alternating Method



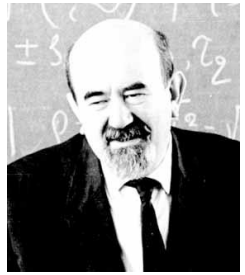
Theoretical Foundation

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

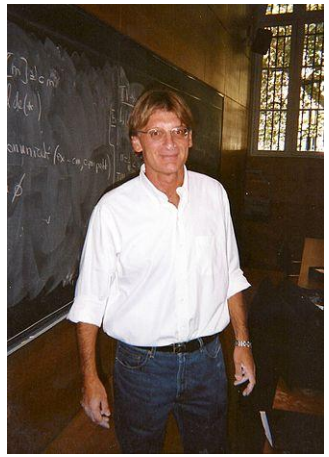


S.L. Sobolev (1908 – 1989)

- 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***.



S.G. Mikhlin
(1908 – 1990)



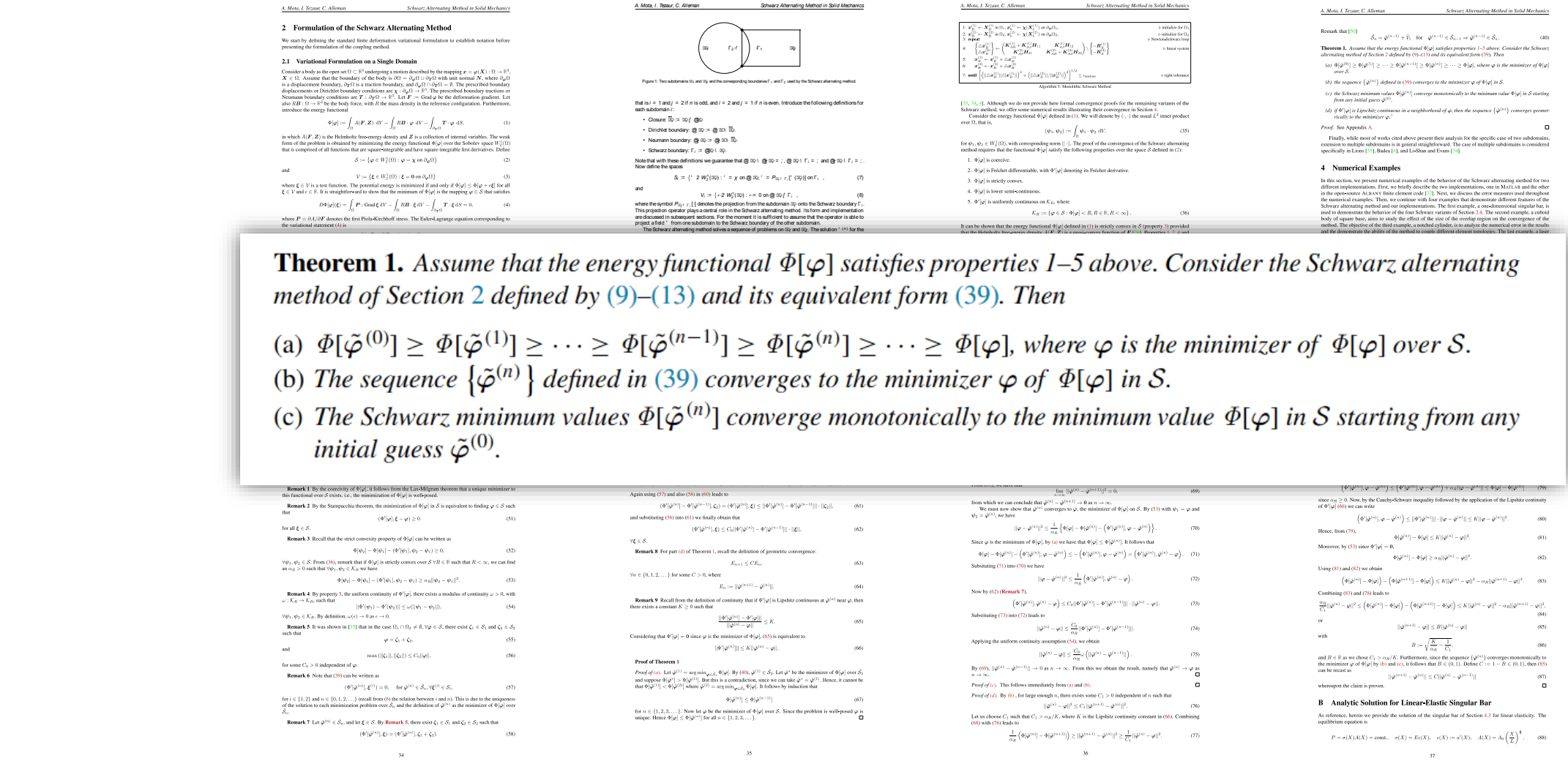
P.- L. Lions (1956-)

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



A. Mota, I. Tezaur, C. Alleman

Theory: Overlapping SAM for Quasistatic Multiscale Coupling*



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 $\{\tilde{\varphi}^{(n)}\}$ 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 $\tilde{\varphi}^{(0)}$.*



* Mota, IT, *et al.*, 2017.

Theory: Overlapping SAM for Dynamic Multiscale Coupling*

- 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 $S[\boldsymbol{\varphi}] \coloneqq \int_I \int_\Omega L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) dV dt$ be ***strictly convex*** or ***strictly concave***, where $L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) \coloneqq T(\dot{\boldsymbol{\varphi}}) + V(\boldsymbol{\varphi})$ 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.

* Mota, **IT**, *et al.*, 2022.

Additional Parallelism via Additive Schwarz

This talk

Multiplicative Overlapping Schwarz

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

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

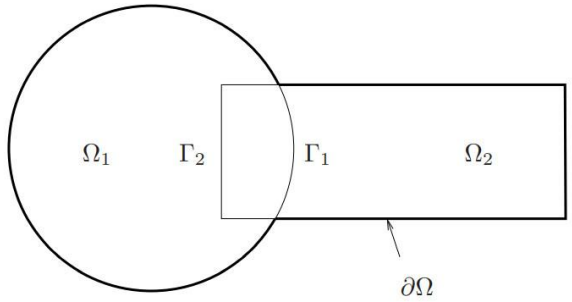
Additive Overlapping Schwarz

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

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

Model PDE:

$$\begin{cases} \text{Div } \boldsymbol{P} + \rho \boldsymbol{B} = \boldsymbol{0}\ , \text{ in } \Omega \\ \boldsymbol{\varphi} = \boldsymbol{\chi}, \qquad \text{on } \partial\Omega \end{cases}$$



- **Multiplicative Schwarz:** solves subdomain problems **sequentially** (in serial)
- **Additive Schwarz:** advance subdomains in **parallel**, communicate boundary condition data later
  - Typically requires a few more **Schwarz iterations**, but does not degrade **accuracy**
  - **Parallelism** helps balance additional **cost** due to Schwarz iterations
  - Applicable to both **overlapping** and **non-overlapping** Schwarz

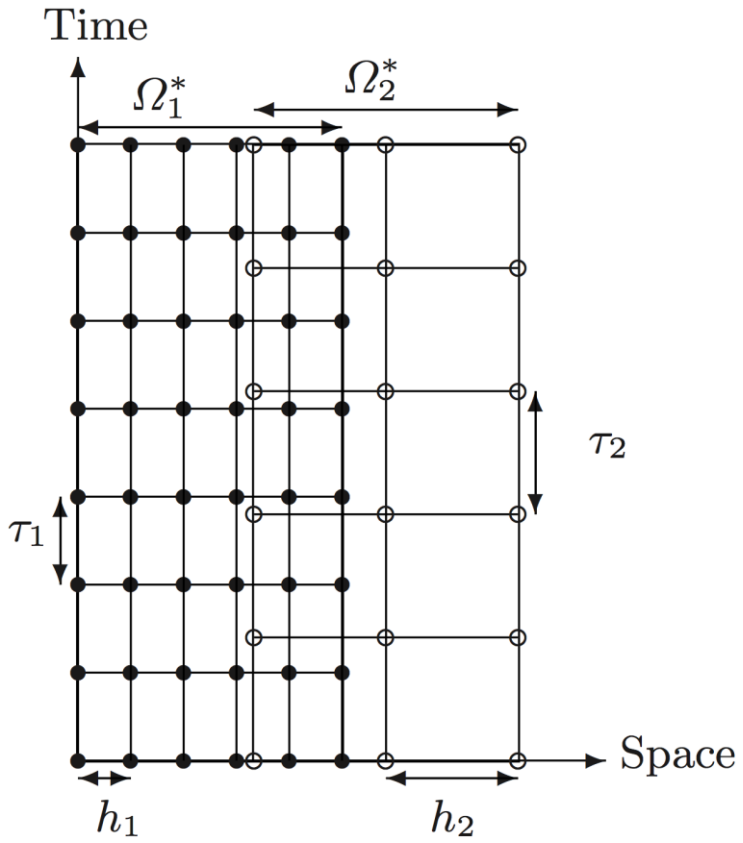
How to Apply Schwarz to Dynamics?

- 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).

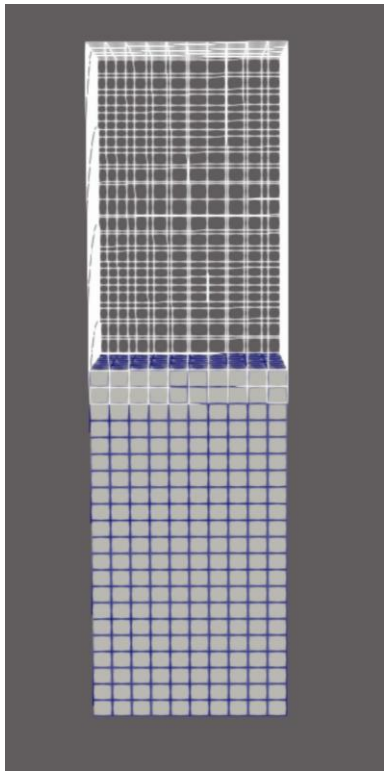
Con ☹: ***Unfeasible*** given the design of our current codes and size of simulations.

Ease of implementation is a very important requirement!

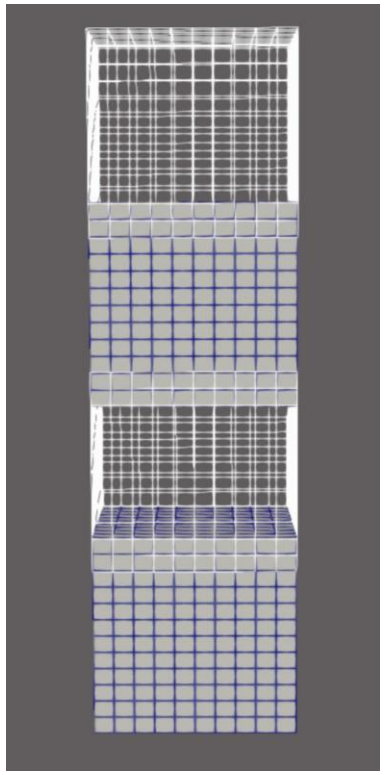


Overlapping non-matching meshes and time steps in dynamics.

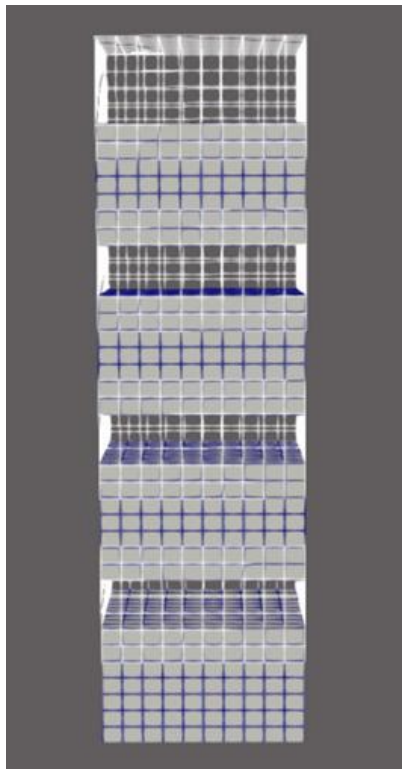
Scaling with Respect to the Number of Subdomains



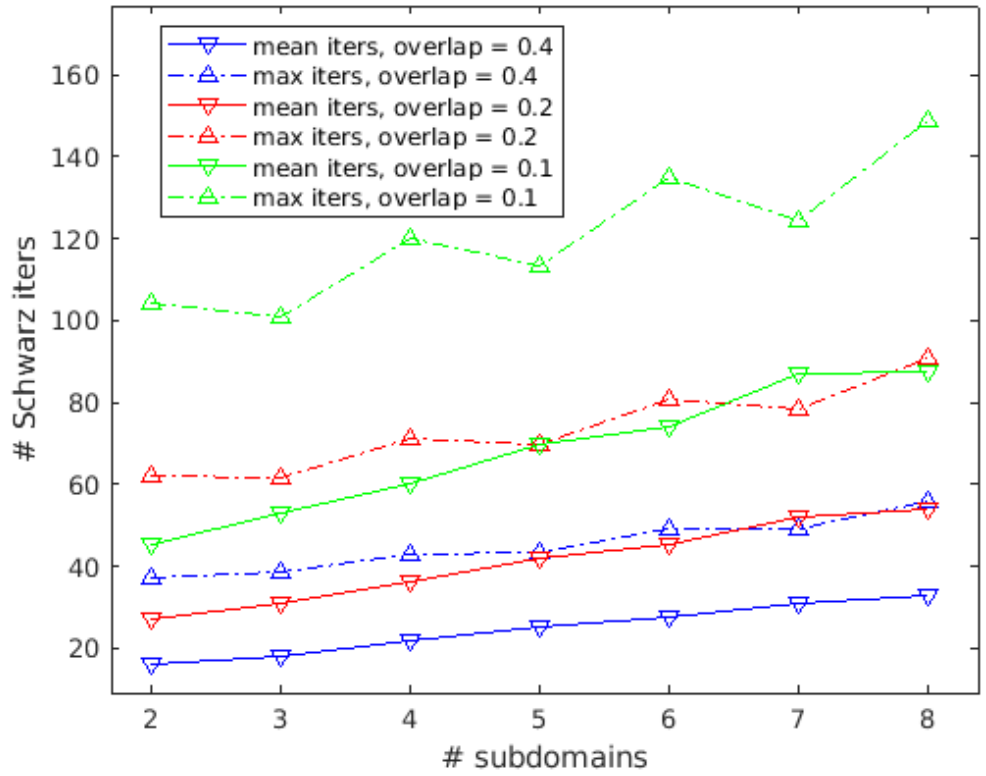
2 subdomains



4 subdomains



8 subdomains



- Overlapping SAM scaling study from 2 to 8 subdomains
- Linear elastic cuboid problem, pulled from top at rate of $0.5 - 0.5\cos(\pi t)$
- Overlap size varied: 0.4, 0.2 (above) and 0.1
- Same mesh resolution of 0.1 in all subdomains
- Linear convergence observed w.r.t. # subdomains

Generally, we do not target cases with >5-6 subdomains

Schwarz Extensions to FOM-ROM and ROM-ROM Couplings



Choice of domain decomposition

- **Overlapping** vs. **non-overlapping** domain decomposition?
  - Non-overlapping more flexible but typically requires more Schwarz iterations
- **FOM** vs. **ROM** subdomain assignment?
  - Do not assign ROM to subdomains where they have no hope of approximating solution

Snapshot collection and reduced basis construction

- Are subdomains **simulated independently** in each subdomains or together?

Enforcement of boundary conditions (BCs) in ROM at Schwarz boundaries

- **Strong** vs. **weak** BC enforcement?
  - Strong BC enforcement difficult for some models (e.g., cell-centered finite volume, PINNs)
- **Optimizing parameters** in Schwarz BCs for non-overlapping Schwarz?

Choice of hyper-reduction

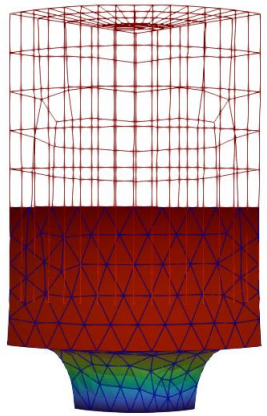
- What **hyper-reduction** method to use?
  - Application may require particular method (e.g., ECSW for solid mechanics problems)
- How to **sample Schwarz boundaries** in applying hyper-reduction?
  - Need to have enough sample mesh points at Schwarz boundaries to apply Schwarz

3D Linear Elastic Notched Cylinder Problem

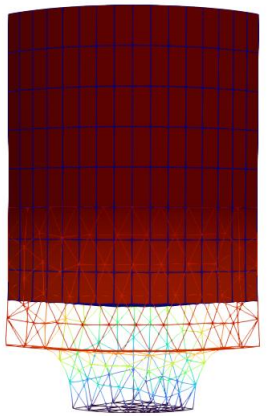
- Geometry is ***linear elastic notched cylinder*** pulled from top ***dynamically*** to time $T_{max} = 1.5$ at rate of $0.0064t$ with $\Delta t = 0.005$
- Demonstration of SAM's ability to ***couple disparate meshes***, ***element types*** and ***models***: TET10 + FOM (notched region) and HEX8 + Linear OpInf with $M = 30$ modes (top region)
- ***Linear OpInf*** trained on 301 snapshots in time
- ***Reproductive*** problem for now

Key result: **regularization parameter** γ influences **accuracy & convergence.** Coupled models are remarkably **accurate!** Schwarz is ***not*** introducing **coupling error/artifacts.**

	Mean/max # Schwarz iters	Max z-disp rel error Ω_1	Max z-disp rel error Ω_2
FOM-FOM	5.83/9	–	–
FOM-OpInf ($\gamma = 1 \times 10^{-6}$)	5.09/8	2.9e-3	4.2e-3
FOM-OpInf ($\gamma = 1 \times 10^{-7}$)	5.48/9	3.8e-4	4.3e-4
FOM-OpInf ($\gamma = 1 \times 10^{-8}$)	5.54/9	1.3e-4	2.2e-4
FOM-OpInf ($\gamma = 1 \times 10^{-9}$)	5.52/9	3.1e-5	3.6e-5



Ω_1 (TET10 + FOM)



Ω_2 (HEX8 + OpInf)



FOM-FOM

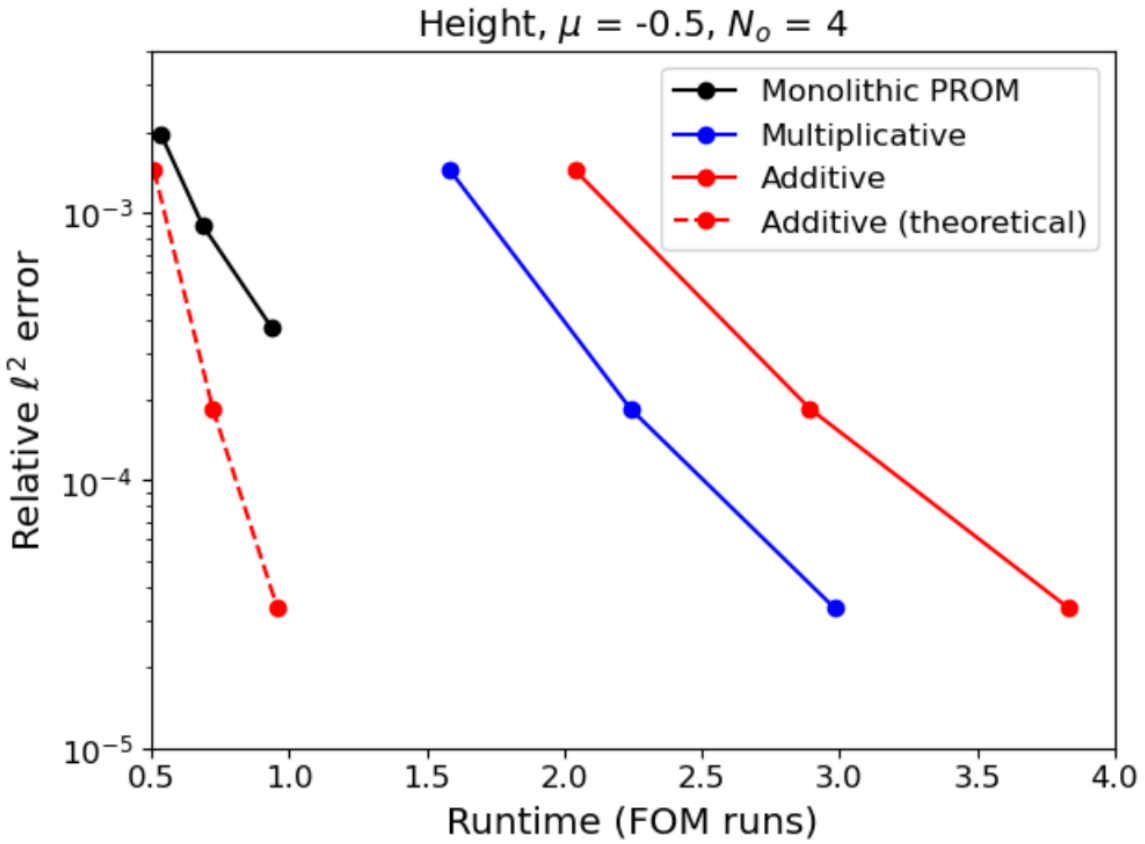


FOM-OpInf

Movies above: z-displacement solutions for FOM-FOM and FOM-OpInf ($M = 30$, $\gamma = 1 \times 10^{-6}$)

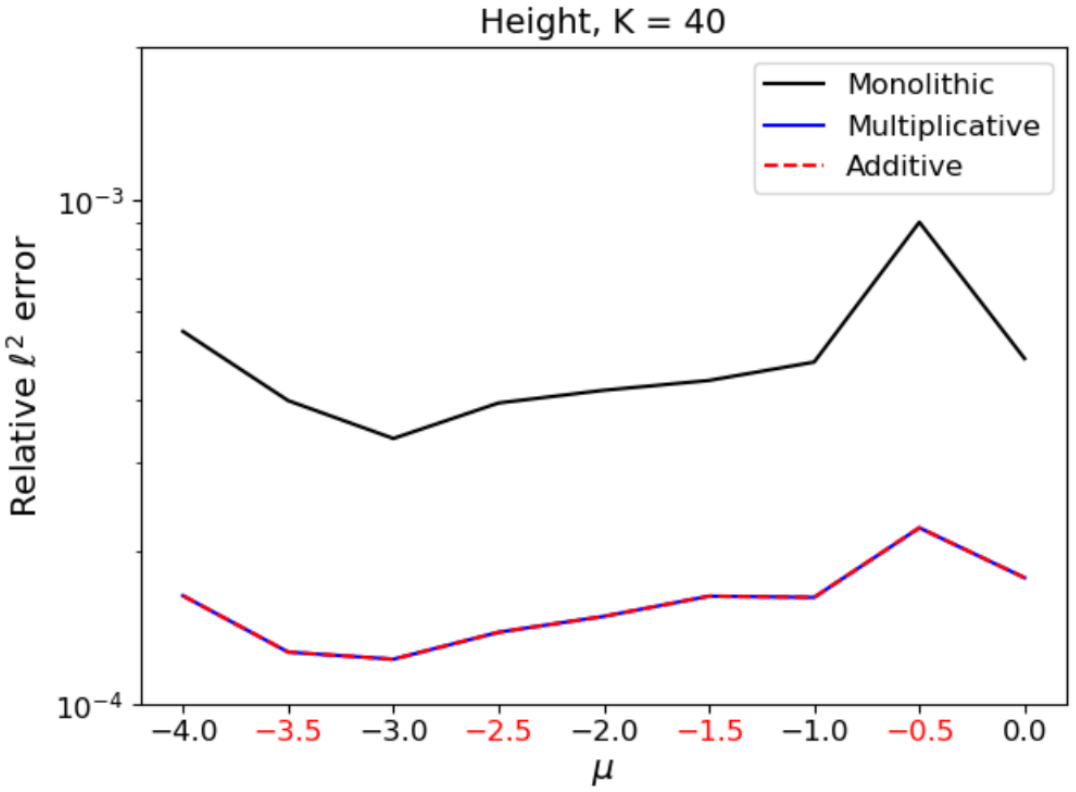
2D SWE: Takeaways

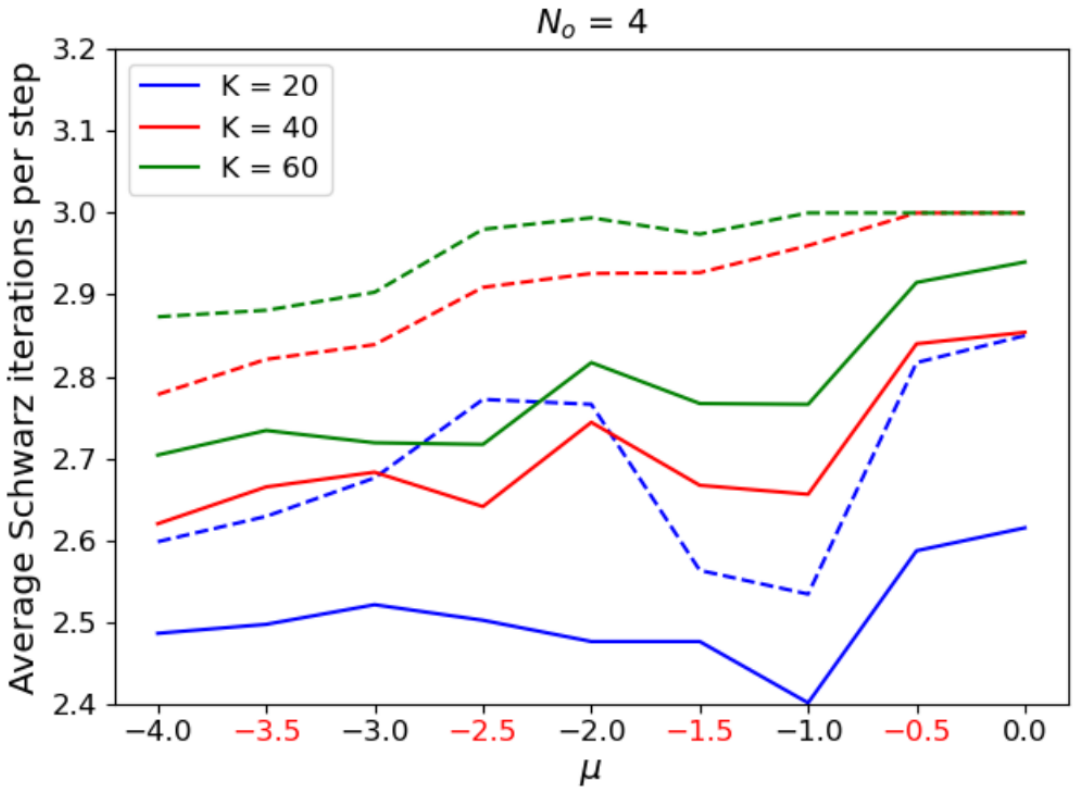
- Resolving overlap region discrepancy is critical
- Additive Schwarz exacerbates PROM dimension costs
- Available parallel cost savings
- Unexpected overlap error requires further investigation



2D SWE: PROMs, additive Schwarz

- Additive does not affect accuracy, converging to same result
- As expected, additive incurs additional Schwarz iterations
- Increased PROM dimension also degrades convergence

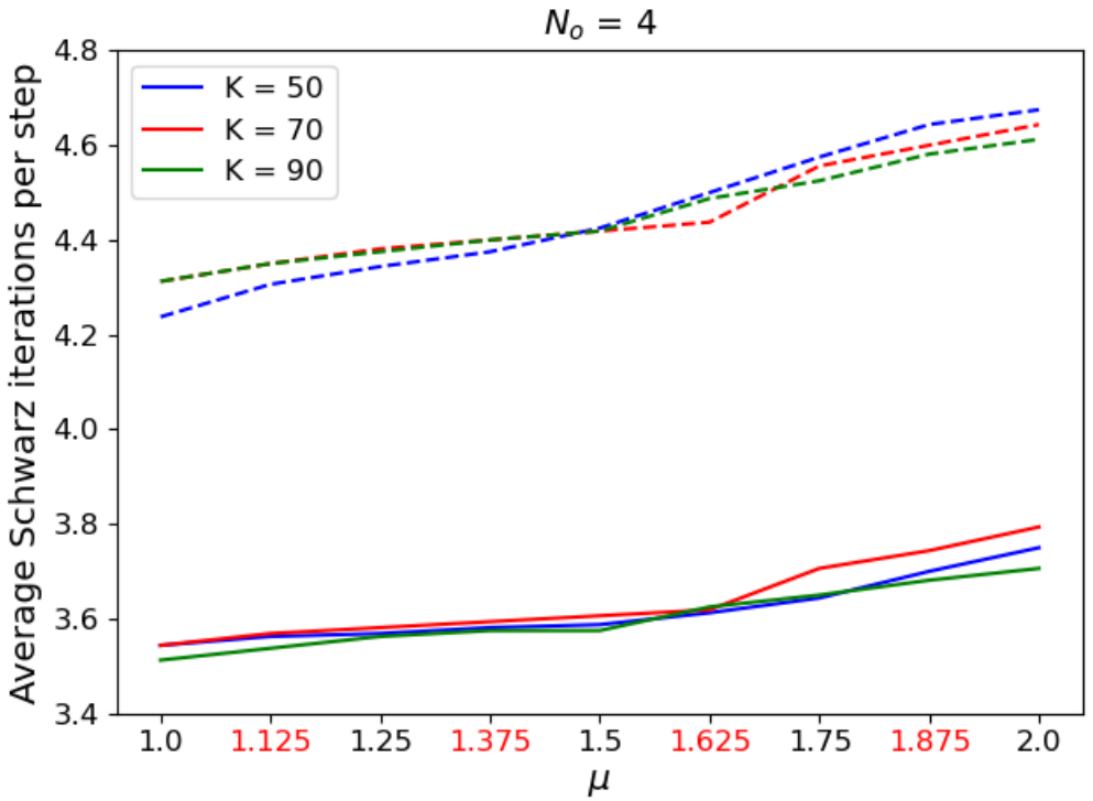


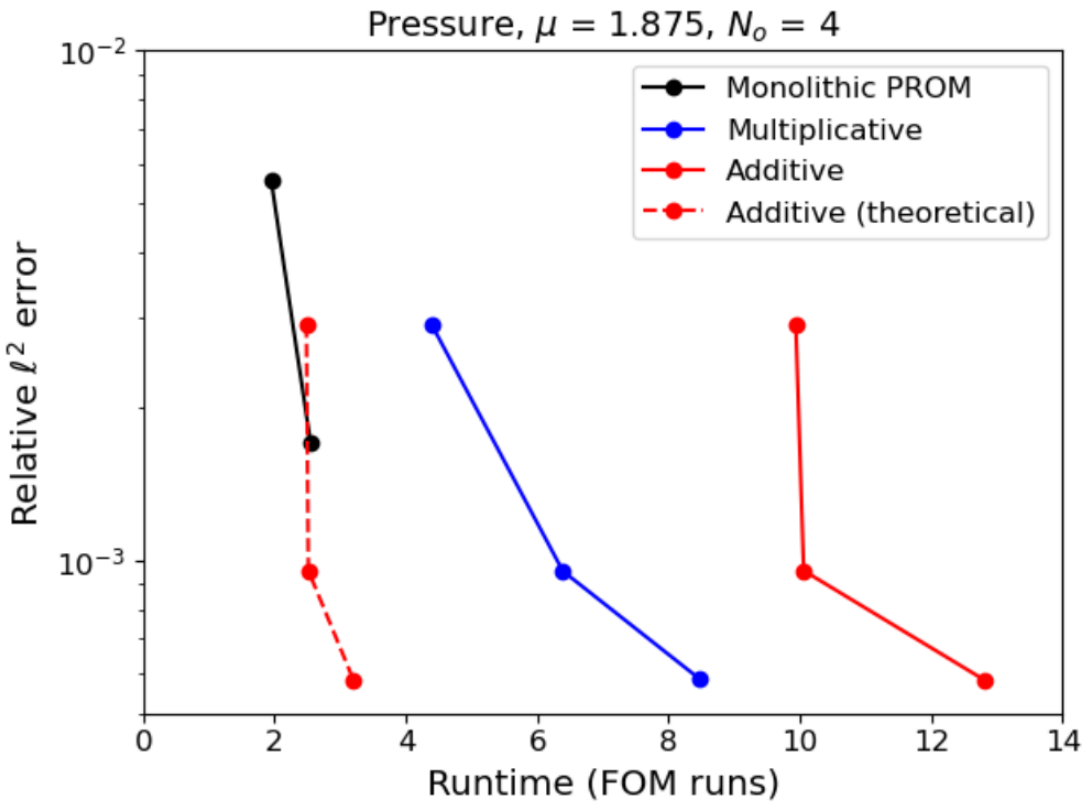


Solid: multiplicative, Dashed: additive

2D Euler: Schwarz PROMs

- PROM dimension has **less** effect on convergence, additive has **more**
- True cost savings unachievable without hyper-reduction





Solid: multiplicative, Dashed: additive

Work in Progress: Accelerating Non-Overlapping Schwarz



G. Sambataro

- **Risk:** non-overlapping Schwarz can be **slow to converge** w/ **Dirichlet-Neumann TCs.**
- **Mitigation:** explore accelerating Dirichlet-Neumann Schwarz via **Aitken** and **Anderson acceleration.**

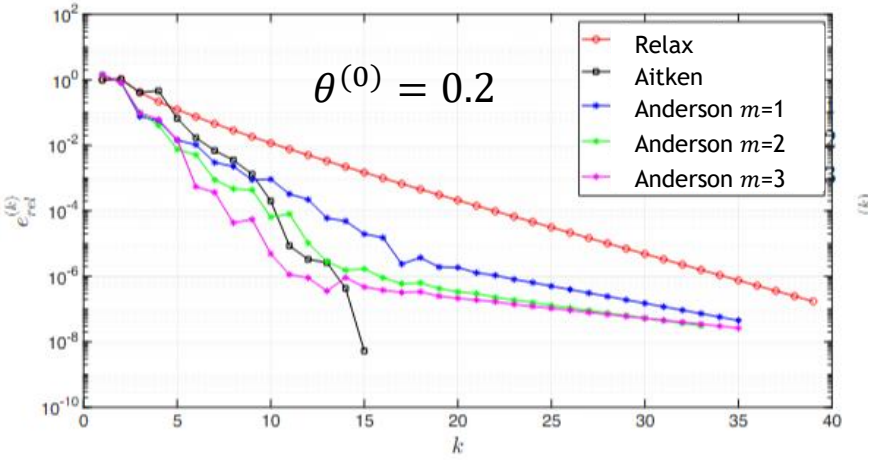
Non-overlapping Dirichlet-Neumann Schwarz

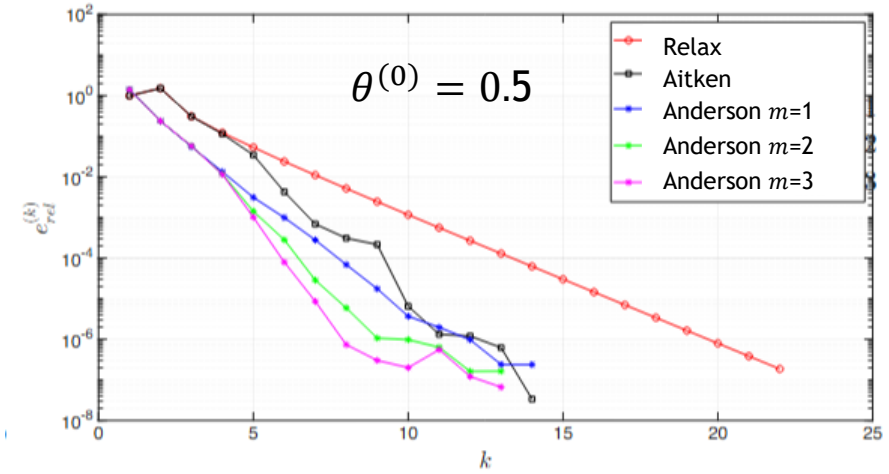
$$\begin{cases} \text{Div}\, \boldsymbol{P}_1^{(n+1)} + \rho \boldsymbol{B}(t_i) = \boldsymbol{0}\,, \text{ in } \Omega_1 \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\chi}, \qquad \text{on } \partial\Omega_1 \backslash \Gamma \\ \boldsymbol{\varphi}_1^{(n+1)} = \boldsymbol{\lambda}_{n+1} \qquad \text{on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

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

Aitken acceleration can **reduce** the # of Schwarz iterations by **3×** and has virtually **no tuning knobs.**

Minimal tuning knobs improves usability.





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

Aitken acceleration: $\boldsymbol{\lambda}_{n+1} = \boldsymbol{\varphi}_1^{(n)} + \theta^{(n-1)} \left(\boldsymbol{\varphi}_2^{(n-1)} - \boldsymbol{\varphi}_1^{(n-1)} \right)$

$$\theta^{(n)} = -\frac{\left(\boldsymbol{\varphi}_2^{(n)} - \boldsymbol{\varphi}_1^{(n)} - \boldsymbol{\varphi}_2^{(n-1)} - \boldsymbol{\varphi}_1^{(n-1)} \right) \cdot \left(\boldsymbol{\varphi}_1^{(n)} - \boldsymbol{\varphi}_1^{(n-1)} \right)}{\left\| \boldsymbol{\varphi}_2^{(n)} - \boldsymbol{\varphi}_1^{(n)} - \boldsymbol{\varphi}_2^{(n-1)} - \boldsymbol{\varphi}_1^{(n-1)} \right\|^2}$$

Anderson acceleration: $\boldsymbol{\lambda}_{n+1} = \sum_{j=0}^{m} \theta^{(n)} T\left(\boldsymbol{\lambda}_{k-m+j} \right),$

$\theta^{(n)}$ from optimization problem

Work in Progress: SAM-based Coupling for Non-Intrusive NN-based Models




E. Parish A. Gruber

Motivation: finite deformation mechanics ***does not*** admit a ***polynomial structure***

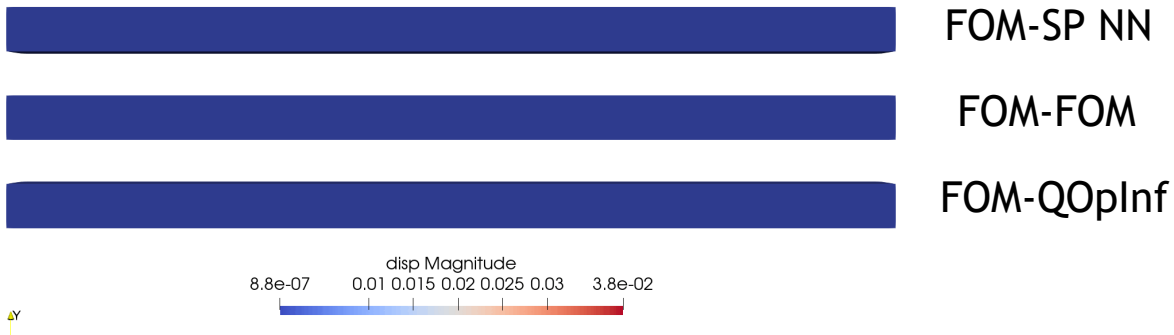
- Limits capacity of polynomial-based OpInf models

Approach: Develop a ***neural-network***-based ***OpInf*** model-reduction strategy

- Can represent ***general nonlinearities***
- Enforces ***structure*** by designing NN operators to parameterize a positive definite stiffness matrix
- ***Fully differentiable ML software*** enables standard and dynamics-constrained training

Pros: Enables accurate models for ***strong nonlinearities***

Cons: Higher ***offline cost***



NN modeling strategy

$$\ddot{\hat{\boldsymbol{q}}}^{NN} + \hat{\boldsymbol{K}}(\hat{\boldsymbol{q}}^{NN}; \boldsymbol{w}_K) - \hat{\boldsymbol{B}}(\boldsymbol{u}_i, \hat{\boldsymbol{q}}^{NN}; \boldsymbol{w}_B) = \boldsymbol{0}$$

$\hat{\boldsymbol{K}}$, $\hat{\boldsymbol{B}}$: NN models for stiffness and boundary forcings

$\boldsymbol{w}_K$, $\boldsymbol{w}_B$: Learnable parameters

Structure is enforced via SPD parameterization of stiffness:

$$\hat{\boldsymbol{K}} = \hat{\boldsymbol{L}}\hat{\boldsymbol{D}}\hat{\boldsymbol{L}}^T$$

Enforced structure significantly improves performance

Future work: Hamiltonian parameterizations

Training paradigms

Offline (traditional) training

$$min_{\boldsymbol{w}_K, \boldsymbol{w}_B} \sum_{i=1}^{N} (\ddot{\hat{\boldsymbol{q}}}_i + \hat{\boldsymbol{K}}(\hat{\boldsymbol{q}}_i; \boldsymbol{w}_K) - \hat{\boldsymbol{B}}(\boldsymbol{u}_i, \hat{\boldsymbol{q}}_i; \boldsymbol{w}_B))^2$$

Dynamics-constrained training (rollout)

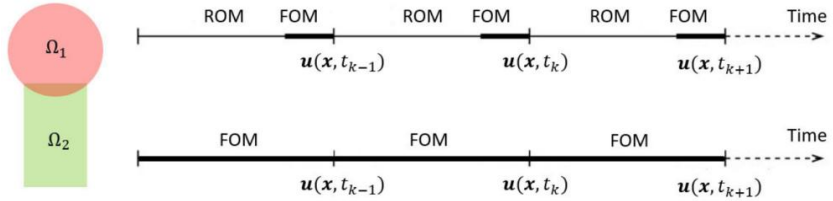
$$min_{\boldsymbol{w}_K, \boldsymbol{w}_B} \sum_{i=1}^{N} (\hat{\boldsymbol{q}}(t_i) - \hat{\boldsymbol{q}}^{NN}(t_i))^2$$

$$\text{s.t. } \ddot{\hat{\boldsymbol{q}}}^{NN} + \hat{\boldsymbol{K}}(\hat{\boldsymbol{q}}^{NN}; \boldsymbol{w}_K) - \hat{\boldsymbol{B}}(\boldsymbol{u}_i, \hat{\boldsymbol{q}}^{NN}; \boldsymbol{w}_B) = \boldsymbol{0}$$

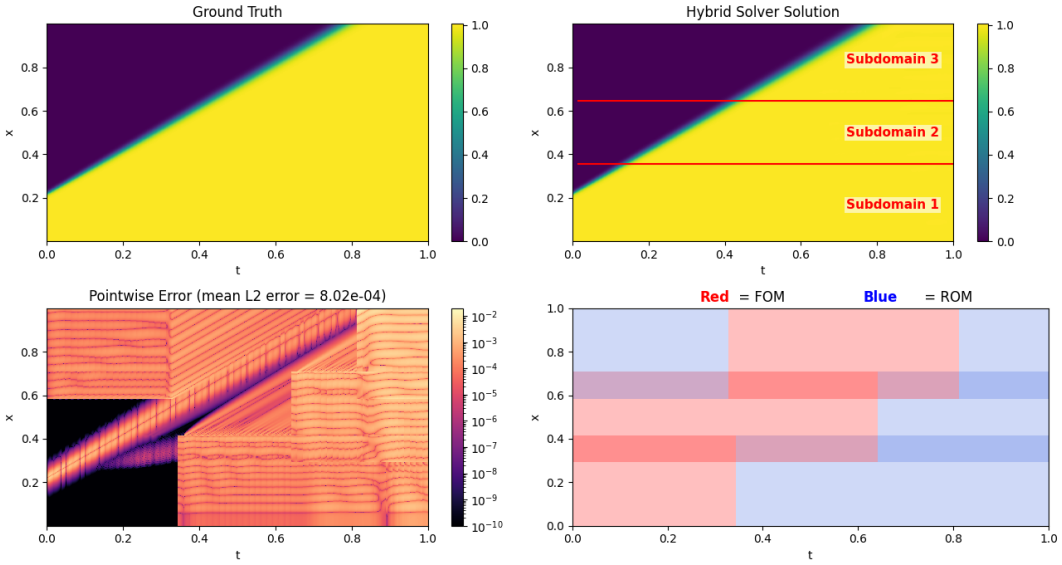
Towards adaptive SAM-based coupling

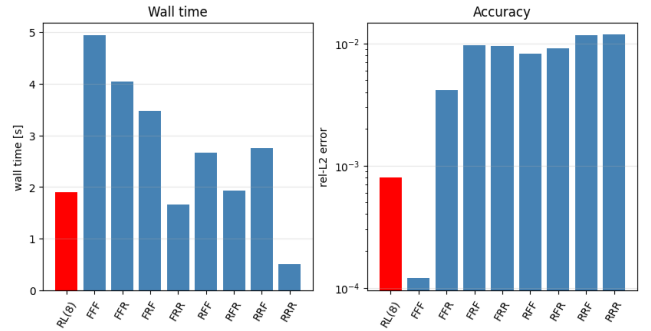
Goal: on-the-fly **model switching/mesh adaptation** if inaccuracies are detected within the **SAM workflow.**

Work in progress: reinforcement learning (RL) approach to model switching

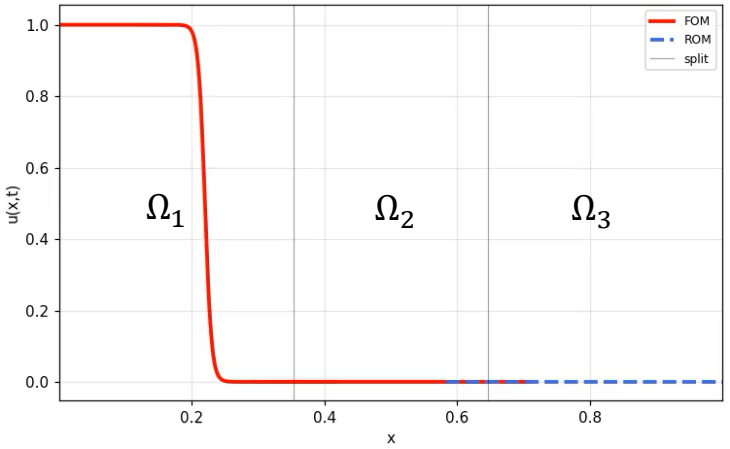


- **State:** down-sampled solution field (200 dimensions)
- **Action:** for 3 overlapping subdomains, we have a discrete action of size 8 that selects FOM/ROM combinations
  - ➢ Model combos: {FFF, FRF, FFR, …, RRR}
  - ➢ OpInf ROMs for all configurations pre-trained and stored in library
- **Reward:** $r_k = -\text{cost}_k - \lambda e_k - \mu \text{nflip}$
- **Episode structure:** full rollout over $[0, T]$, 32 Δts/episode with resets
- **Architecture:** Deep Q-Network
- **Train** on several initial conditions, **infer** on **unseen initial condition**





Result: RL algorithm applied to 1D advection-diffusion problem with shock



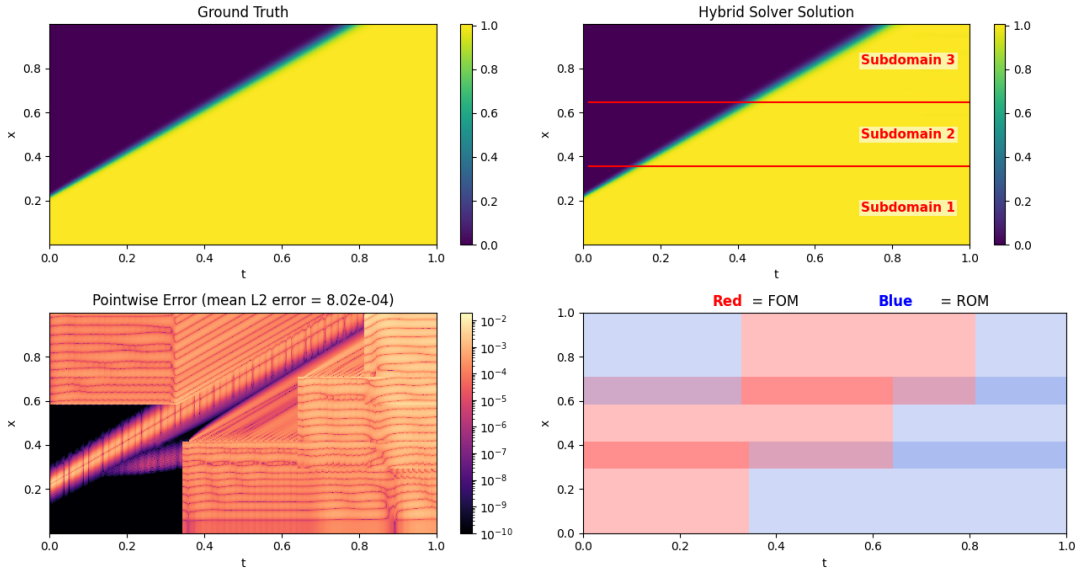
Adaptive SAM-based Coupling: 1D Advection-Diffusion

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \quad \text{on } (0,1) \text{ with } c = 1,\ \nu = 10^{-4}$$

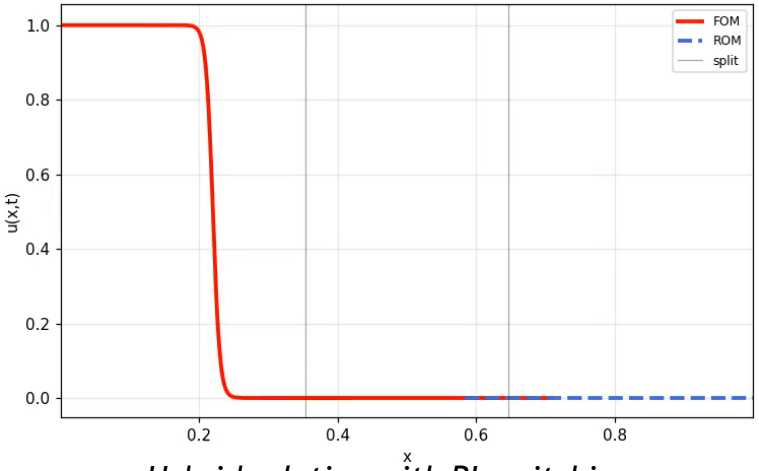
$$u(x,0) = \frac{1}{2}\left(1 - \tanh\left(100(x - x_0)\right)\right), x_0 \sim U(0.05, 0.5)$$

$N = 2048$, 3 subdomains, 10 mode ROMs, 8 actions, 64 windows

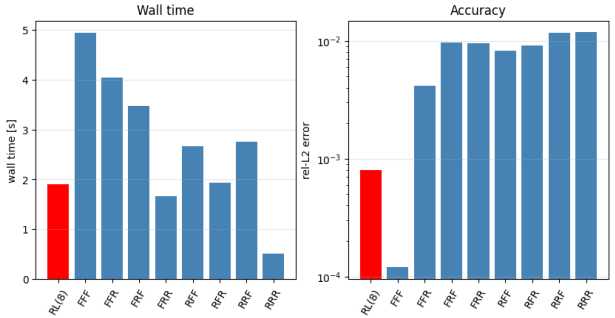
- Learned FOM band **tracks the moving front** whereas ROMs cover smooth regions
- Relative L^2 error $\approx \mathbf{8 \times 10^{-4}}$ vs. monolithic FOM on ***unseen*** **IC**
- **Beats every fixed FOM/ROM combo** on cost vs. accuracy
- Switching penalty μ is essential: **31 → 4 switches** (right)



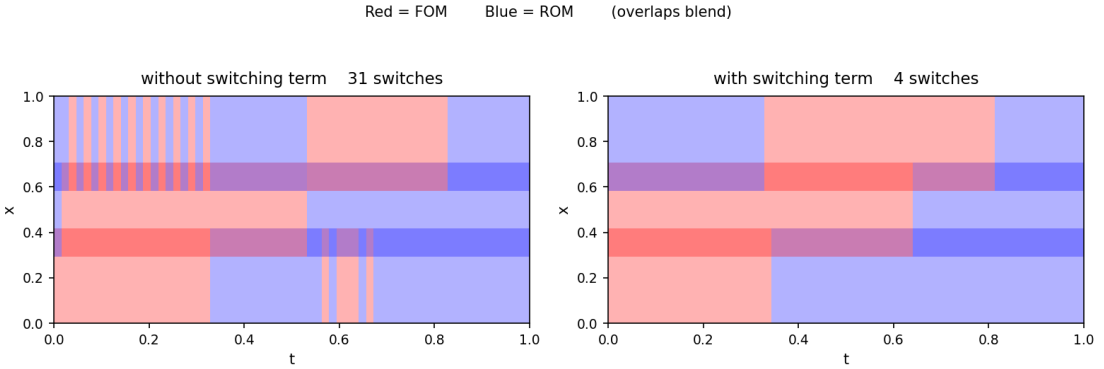
Ground truth, hybrid RL solution, pointwise error, learned policy (red = FOM, blue = ROM)



Hybrid solution with RL switching



RL(8) vs. 8 fixed combos: wall time & L^2 relative error



Learned decision maps: without (left) vs. with (right) switching penalty

Summary

The **Schwarz alternating method** has been developed for concurrent multiscale coupling of **conventional** and **data-driven models.**

- ☺ Coupling is ***concurrent*** (two-way).
- ☺ ***Ease of implementation*** into existing massively-parallel HPC codes.
- ☺ ***"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 (including ROM/FOM)*** and ***time-integrators*** in different regions.
- ☺ ***Scalable***, ***fast***, ***robust*** on ***real*** engineering problems
- ☺ Coupling does not introduce ***nonphysical artifacts***.
- ☺ ***Theoretical*** convergence properties/guarantees.