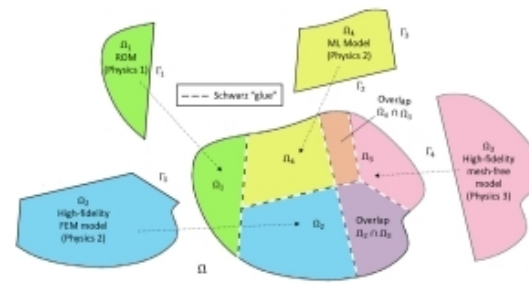
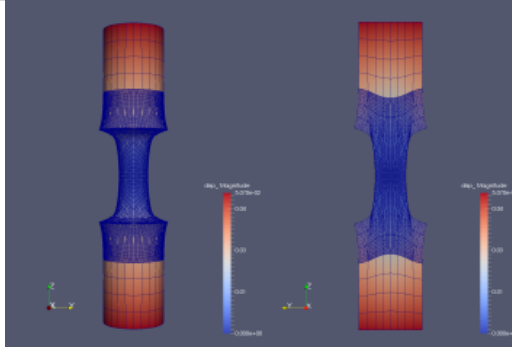
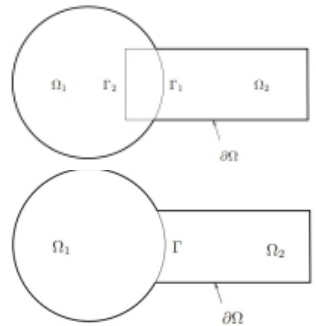


Flexible domain decomposition-based couplings of conventional and data-driven models via the Schwarz alternating method



Irina Tezaur¹, Chris Wentland¹, Francesco Rizzi², Joshua Barnett³,
Alejandro Mota¹

¹Sandia National Laboratories, ²NexGen Analytics, ³Cadence Design Systems

2nd AMS-UNI International Joint Meeting
Palermo, Italy. July 23-26, 2024

SAND2024-XXXX

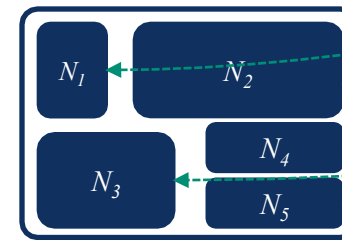
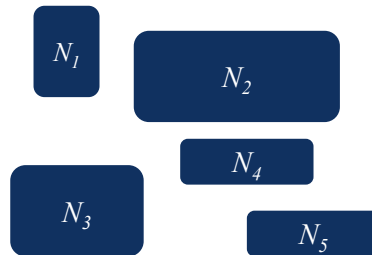
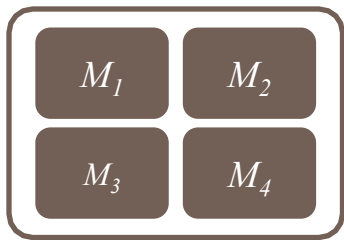


Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation: multi-scale & multi-physics coupling



There exist established **rigorous mathematical theories** for **coupling** multi-scale and multi-physics components based on **traditional discretization methods** (“Full Order Models” or FOMs).



Complex System Model

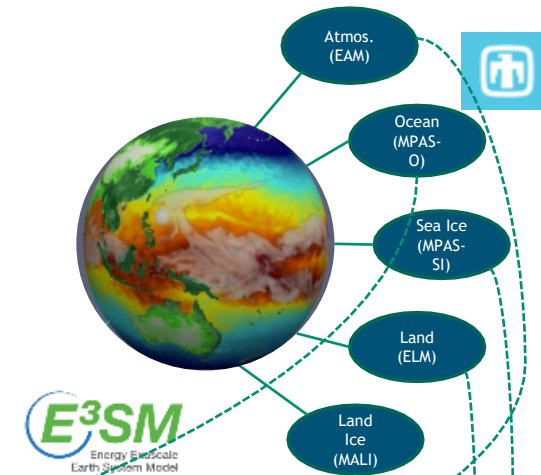
- PDEs, ODEs
- Nonlocal integral
- Classical DFT
- Atomistic, ...

Traditional Methods

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

Coupled Numerical Model

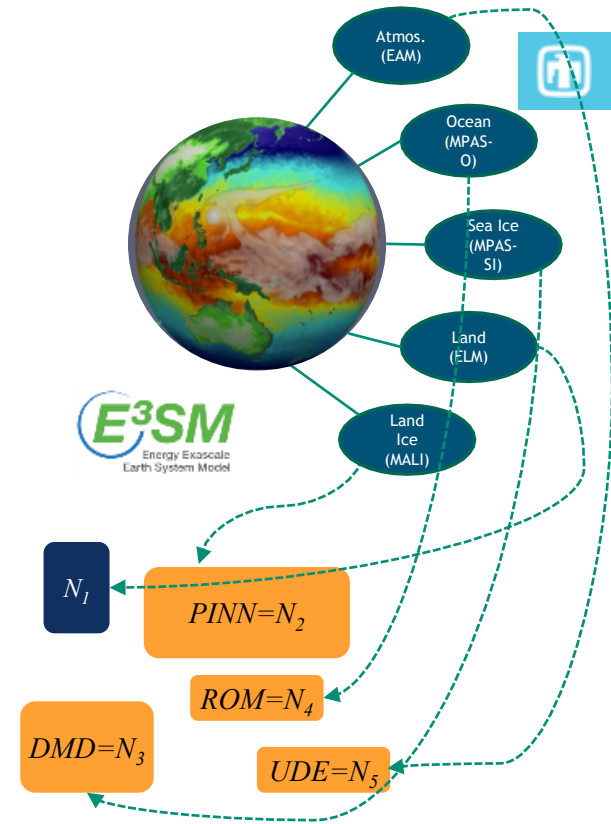
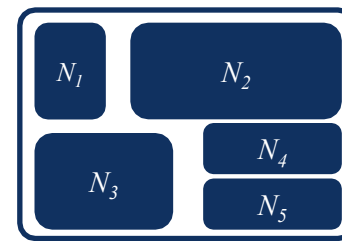
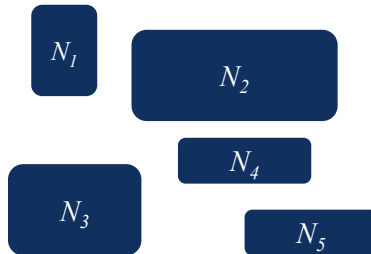
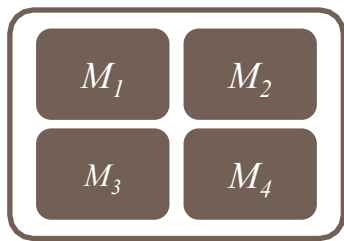
- Monolithic (Lagrange multipliers)
- Partitioned (loose) coupling
- Iterative (Schwarz, optimization)



Motivation: multi-scale & multi-physics coupling



There exist established **rigorous mathematical theories** for **coupling** multi-scale and multi-physics components based on **traditional discretization methods** (“Full Order Models” or FOMs).



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, ...

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



Principal research objective:

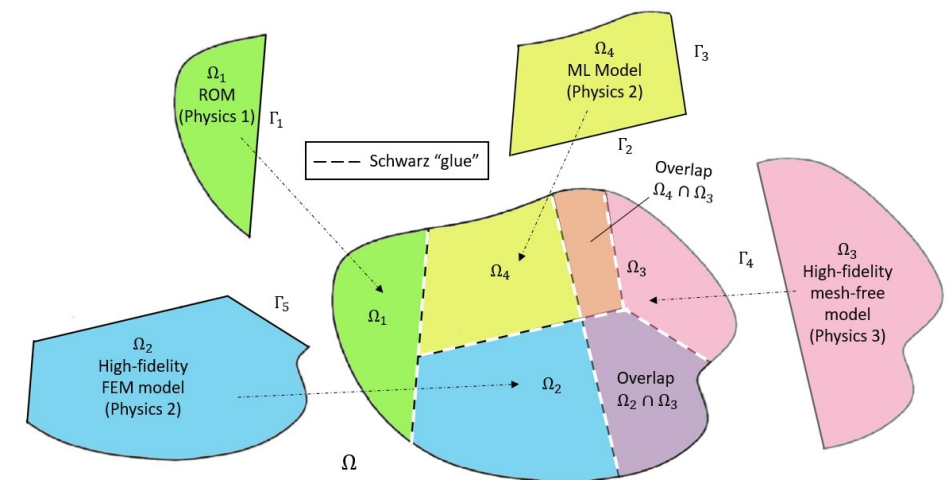
- Discover mathematical principles guiding the assembly of **standard** and **data-driven** numerical models in stable, accurate and physically consistent ways.

Principal research goals:

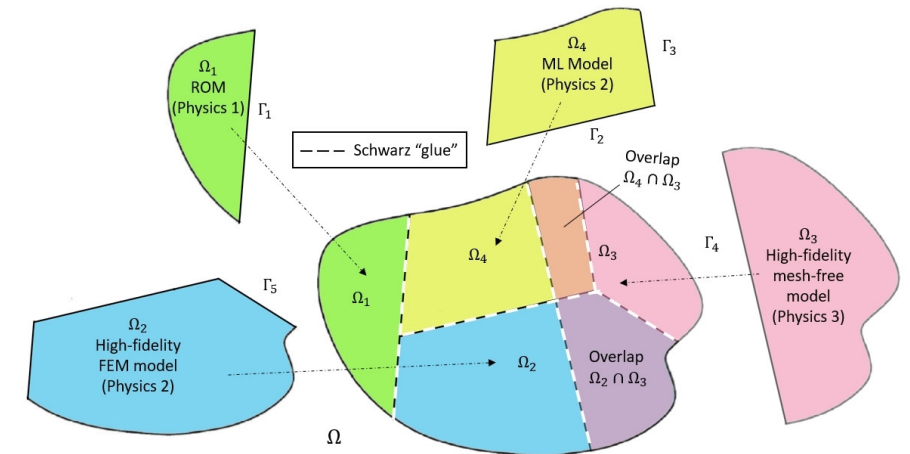
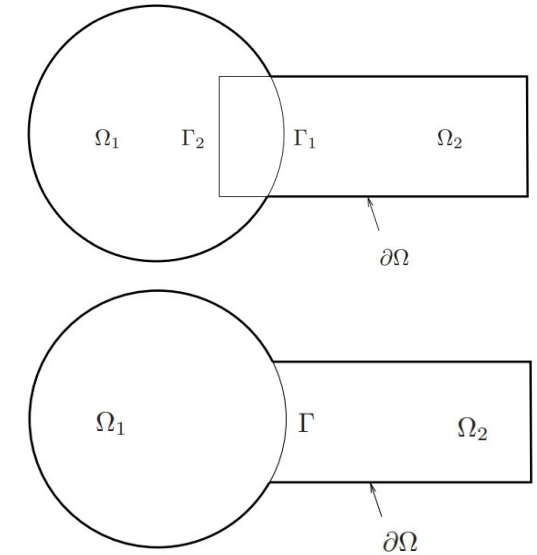
- “Mix-and-match” standard and data-driven models from three-classes
 - **Class A:** projection-based reduced order models (ROMs) *This talk.*
 - **Class B:** machine-learned models, i.e., Physics-Informed Neural Networks (PINNs)
 - **Class C:** flow map approximation models, i.e., dynamic model decomposition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

Three coupling methods:

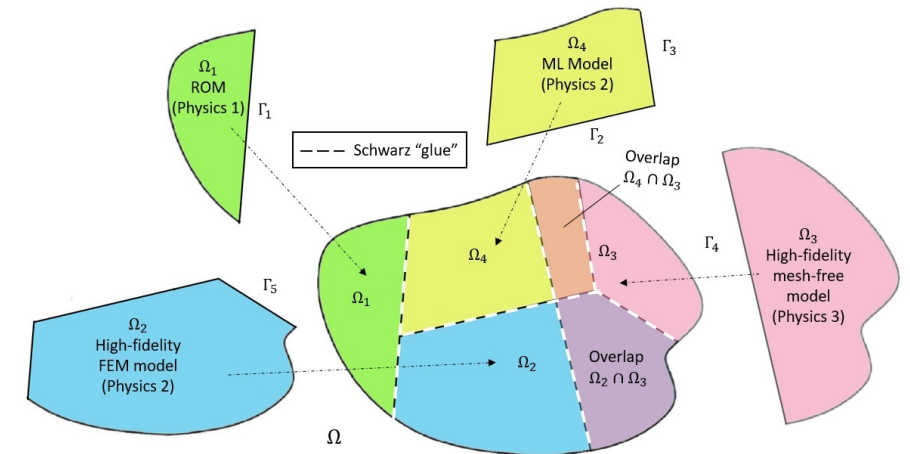
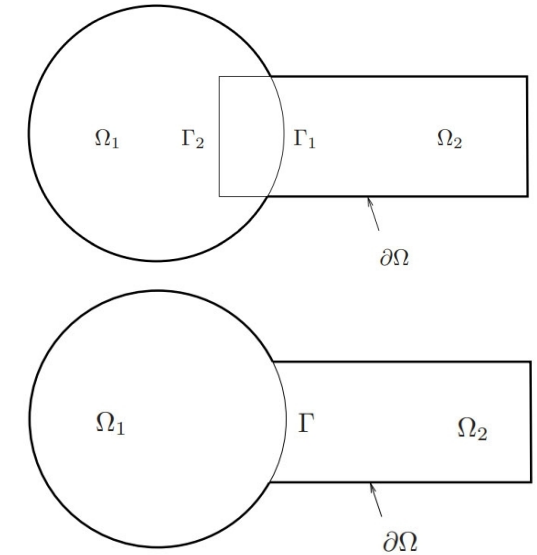
- Alternating Schwarz-based coupling *This talk.*
- Optimization-based coupling
- Coupling via generalized mortar methods



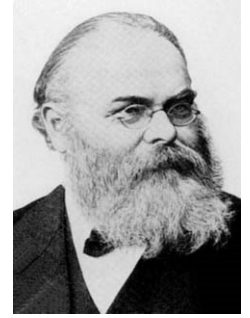
- The Schwarz Alternating Method for Domain Decomposition-Based Coupling
- Extension to FOM*-ROM[#] and ROM-ROM Coupling
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work



- **The Schwarz Alternating Method for Domain Decomposition-Based Coupling**
- Extension to FOM*-ROM[#] and ROM-ROM Coupling
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work



Schwarz Alternating Method for Domain Decomposition



H. Schwarz (1843-1921)



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

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

Basic Schwarz Algorithm

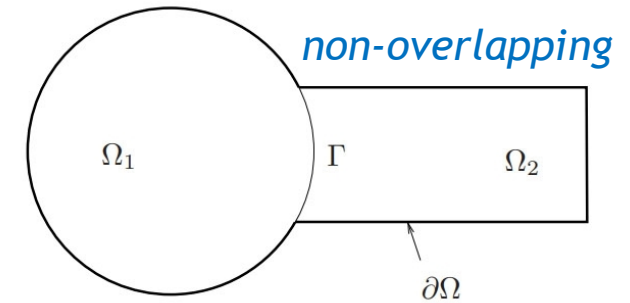
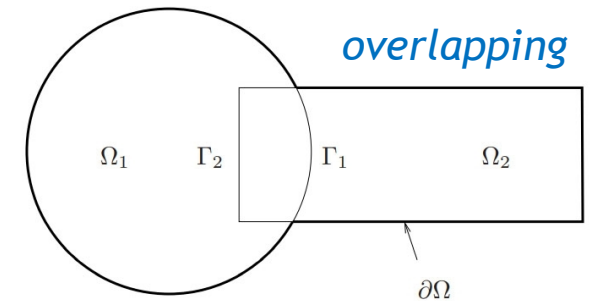
Initialize:

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

Iterate until convergence:

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

- Schwarz alternating method most commonly used as a **preconditioner** for Krylov iterative methods to solve linear algebraic equations.



Idea behind this work: using the Schwarz alternating method as a **discretization method** for solving multi-scale or multi-physics partial differential equations (PDEs).



AS A *PRECONDITIONER*
FOR THE LINEARIZED
SYSTEM

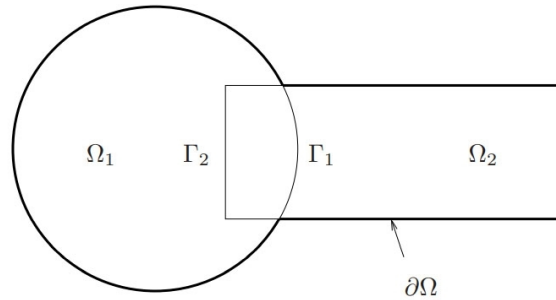


AS A *SOLVER* FOR THE
COUPLED
FULLY NONLINEAR
PROBLEM

Overlapping Domain Decomposition

$$\begin{cases} N(\mathbf{u}_1^{(n+1)}) = f, & \text{in } \Omega_1 \\ \mathbf{u}_1^{(n+1)} = \mathbf{g}, & \text{on } \partial\Omega_1 \setminus \Gamma_1 \\ \mathbf{u}_1^{(n+1)} = \mathbf{u}_2^{(n)} & \text{on } \Gamma_1 \end{cases}$$

$$\begin{cases} N(\mathbf{u}_2^{(n+1)}) = f, & \text{in } \Omega_2 \\ \mathbf{u}_2^{(n+1)} = \mathbf{g}, & \text{on } \partial\Omega_2 \setminus \Gamma_2 \\ \mathbf{u}_2^{(n+1)} = \mathbf{u}_1^{(n+1)} & \text{on } \Gamma_2 \end{cases}$$



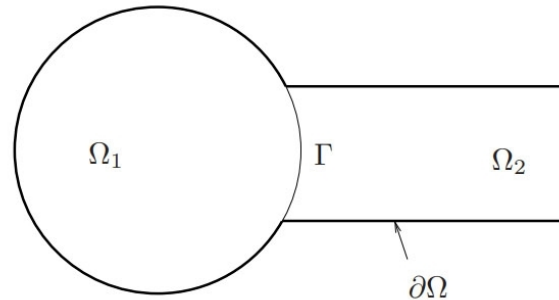
Model PDE: **No Image**

- Dirichlet-Dirichlet transmission BCs [Schwarz 1870; Lions 1988; Mota *et al.* 2017; Mota *et al.* 2022]

Non-overlapping Domain Decomposition

No Image

No Image



$$\lambda_{n+1} = \theta \varphi_2^{(n)} + (1 - \theta) \lambda_n, \text{ on } \Gamma, \text{ for } n \geq 1$$

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



Multiplicative Overlapping Schwarz

$$\begin{cases} N(\mathbf{u}_1^{(n+1)}) = f, & \text{in } \Omega_1 \\ \mathbf{u}_1^{(n+1)} = \mathbf{g}, & \text{on } \partial\Omega_1 \setminus \Gamma_1 \\ \mathbf{u}_1^{(n+1)} = \mathbf{u}_2^{(n)} & \text{on } \Gamma_1 \end{cases}$$

$$\begin{cases} N(\mathbf{u}_2^{(n+1)}) = f, & \text{in } \Omega_2 \\ \mathbf{u}_2^{(n+1)} = \mathbf{g}, & \text{on } \partial\Omega_2 \setminus \Gamma_2 \\ \mathbf{u}_2^{(n+1)} = \mathbf{u}_1^{(n+1)} & \text{on } \Gamma_2 \end{cases}$$

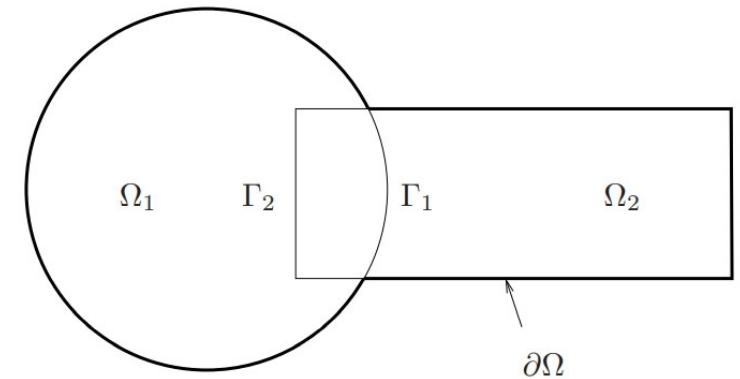
Additive Overlapping Schwarz

$$\begin{cases} N(\mathbf{u}_1^{(n+1)}) = f, & \text{in } \Omega_1 \\ \mathbf{u}_1^{(n+1)} = \mathbf{g}, & \text{on } \partial\Omega_1 \setminus \Gamma_1 \\ \mathbf{u}_1^{(n+1)} = \mathbf{u}_2^{(n)} & \text{on } \Gamma_1 \end{cases}$$

$$\begin{cases} N(\mathbf{u}_2^{(n+1)}) = f, & \text{in } \Omega_2 \\ \mathbf{u}_2^{(n+1)} = \mathbf{g}, & \text{on } \partial\Omega_2 \setminus \Gamma_2 \\ \mathbf{u}_2^{(n+1)} = \mathbf{u}_1^{(n+1)} & \text{on } \Gamma_2 \end{cases}$$

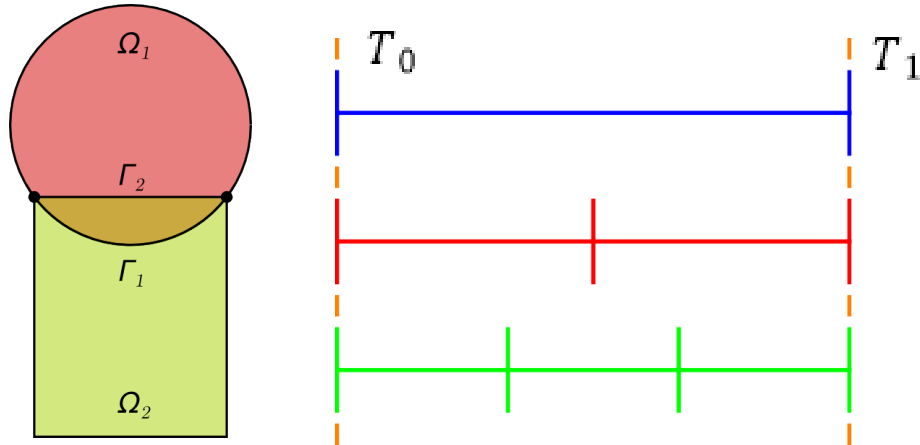
Model PDE:

$$\begin{cases} N(\mathbf{u}) = f, & \text{in } \Omega \\ \mathbf{u} = \mathbf{g}, & \text{on } \partial\Omega \end{cases}$$



- **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

Time-Advancement Within the Schwarz Framework



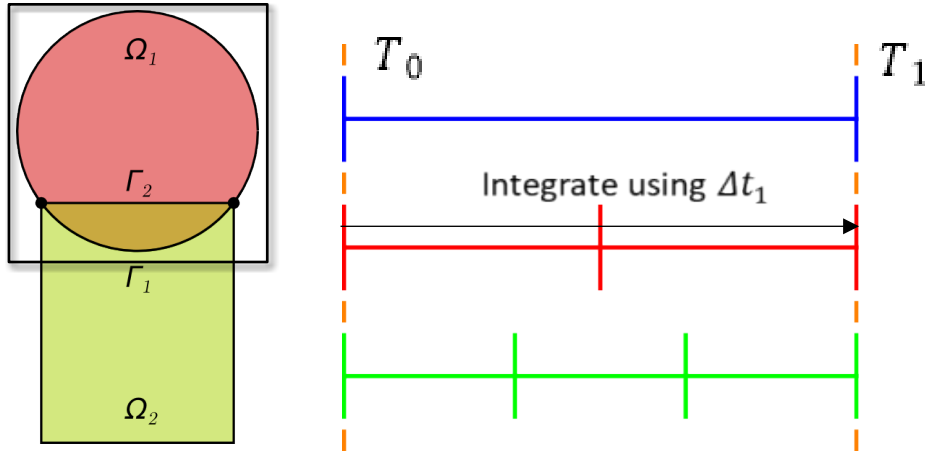
Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

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

$$\text{Model PDE: } \begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$



Controller time stepper

Time integrator for Ω_1

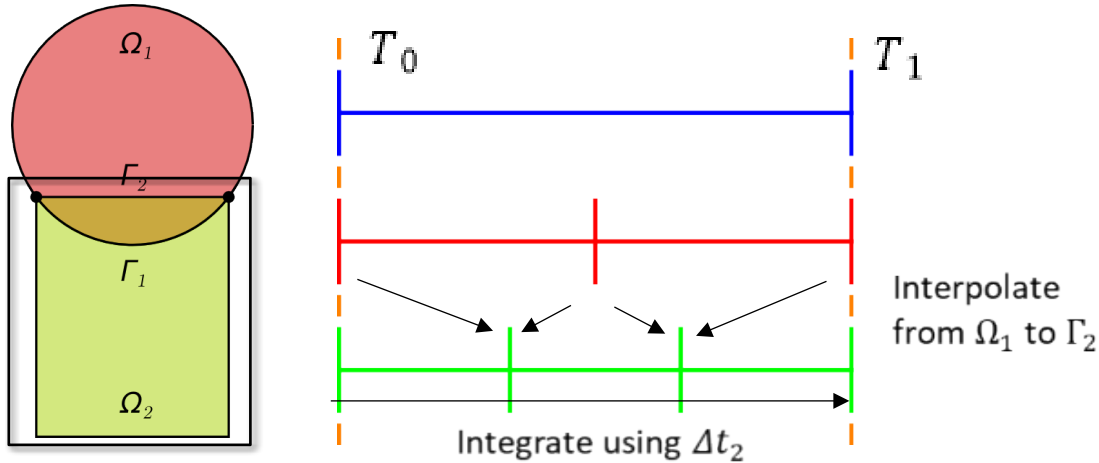
Time integrator for Ω_2

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 $T_i + n\Delta t_1$.

$$\text{Model PDE: } \begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$

Time-Advancement Within the Schwarz Framework



Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

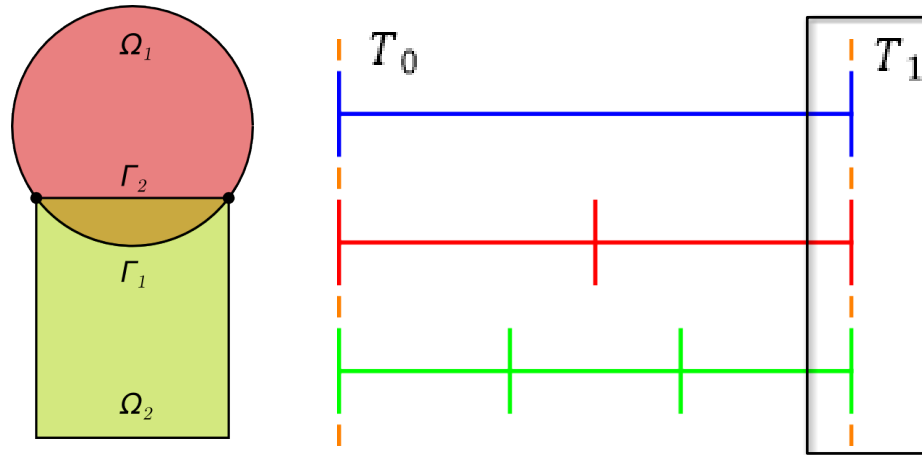
Step 0: Initialize $i = 0$ (controller time index).

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 $T_i + n\Delta t_2$.

$$\text{Model PDE: } \begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$

Time-Advancement Within the Schwarz Framework



Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

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

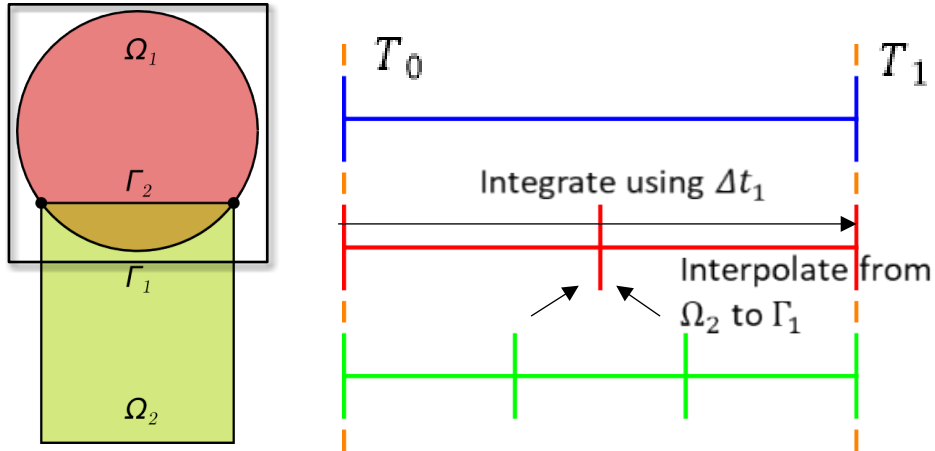
Step 1: Advance Ω_1 solution from time T_i to time T_{i+1} using time-stepper in Ω_1 with time-step Δt_1 , using solution in Ω_2 interpolated to Γ_1 at times $T_i + n\Delta t_1$.

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} .

$$\text{Model PDE: } \begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$

Time-Advancement Within the Schwarz Framework



Controller time stepper

Time integrator for Ω_1

Time integrator for 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 $T_i + n\Delta t_2$.

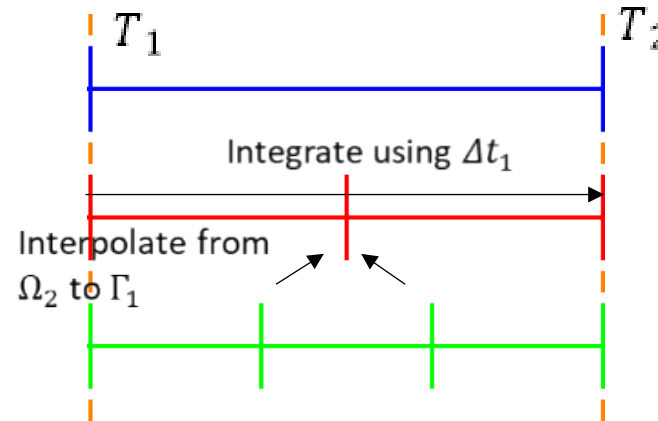
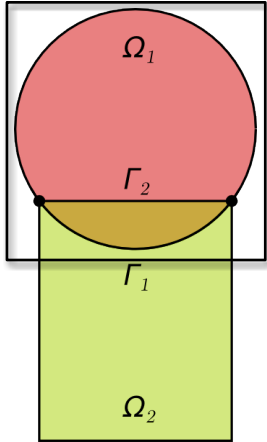
Step 3: Check for convergence at time T_{i+1} .

➤ If unconverged, return to Step 1.

Model PDE:

$$\begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$

Time-Advancement Within the Schwarz Framework



Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

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

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 $T_i + n\Delta t_1$.

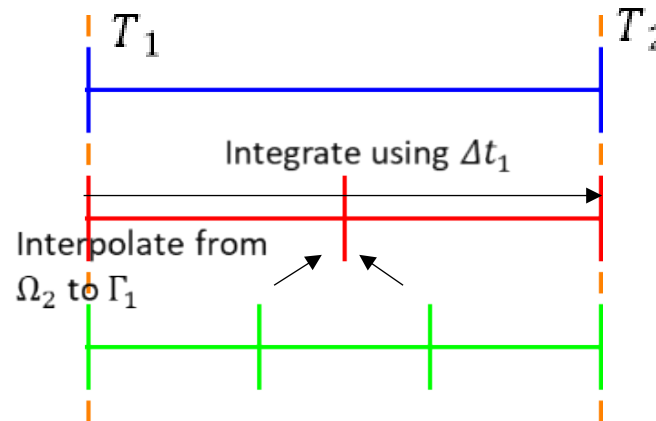
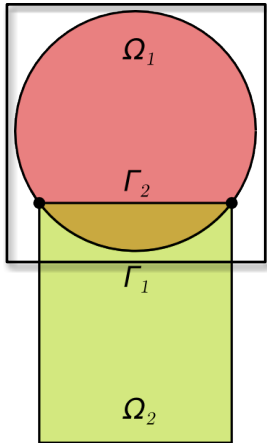
Step 2: Advance Ω_2 solution from time T_i to time T_{i+1} using time-stepper in Ω_2 with time-step Δt_2 , using solution in Ω_1 interpolated to Γ_2 at times $T_i + n\Delta t_2$.

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:

$$\begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$



Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

Time-stepping procedure is **equivalent** to doing Schwarz on **space-time domain** [Mota *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:

$$\begin{cases} \dot{\mathbf{u}} + N(\mathbf{u}) = \mathbf{f}, & \text{in } \Omega \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(t), & \text{on } \partial\Omega \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, & \text{in } \Omega \end{cases}$$

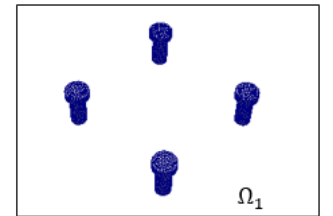
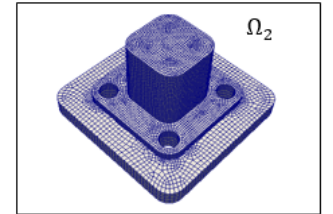
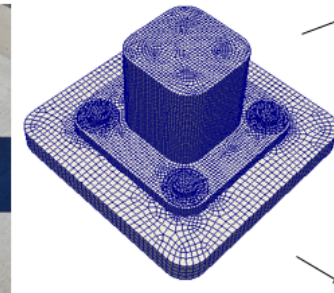


Model Solid Mechanics PDEs:

$$\text{Quasistatic: } \operatorname{Div} \mathbf{P} + \rho_0 \mathbf{B} = \mathbf{0} \quad \text{in } \Omega$$

$$\text{Dynamic: } \operatorname{Div} \mathbf{P} + \rho_0 \mathbf{B} = \rho_0 \ddot{\boldsymbol{\varphi}} \quad \text{in } \Omega \times I$$

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



¹ Mota et al. 2017; Mota et al. 2022. ² <https://github.com/sandialabs/LCM>.

Schwarz for Multiscale FOM-FOM Coupling in Solid Mechanics¹

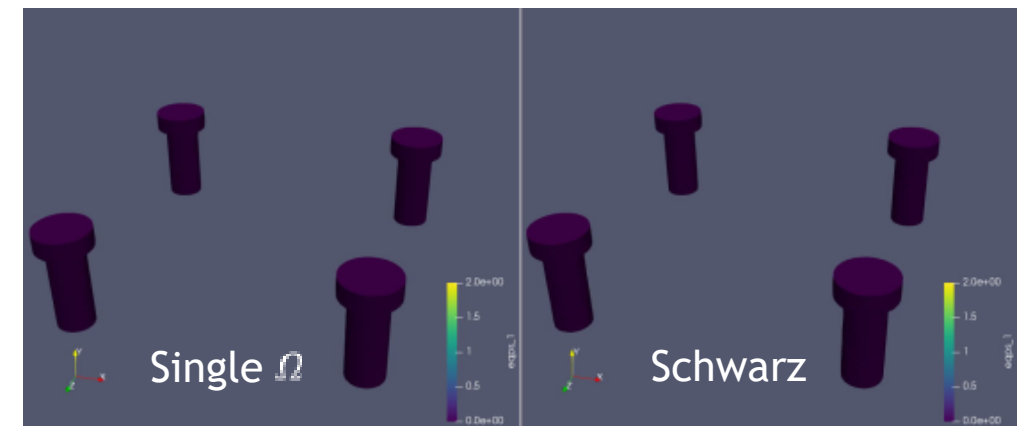
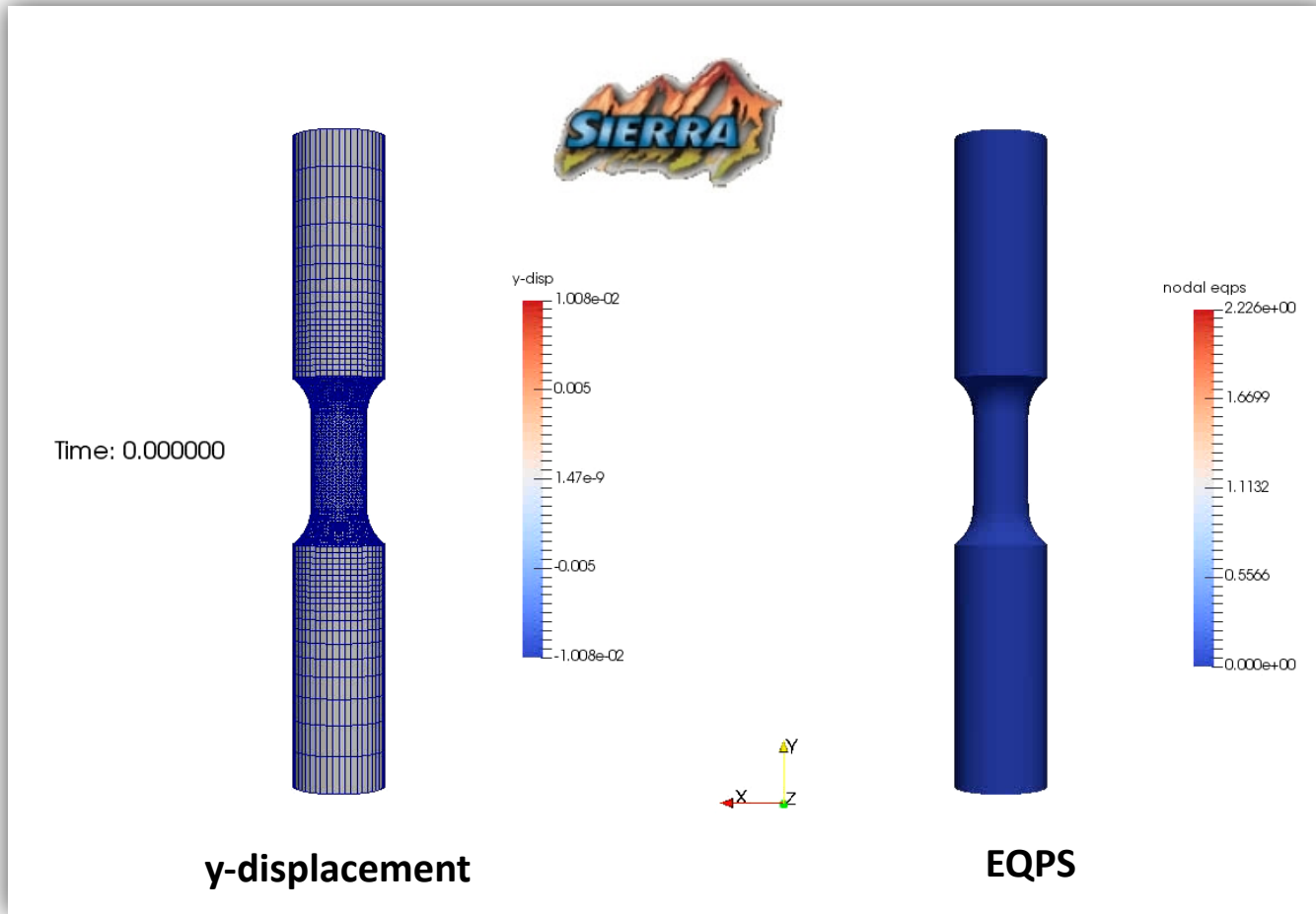
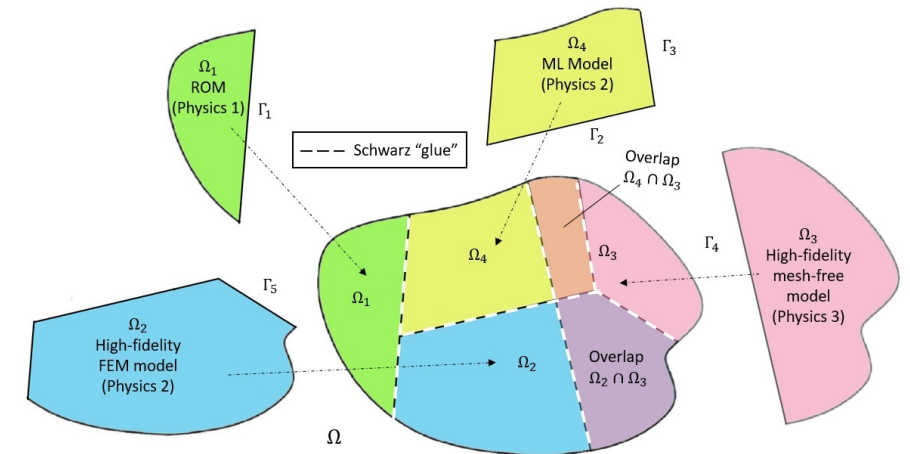
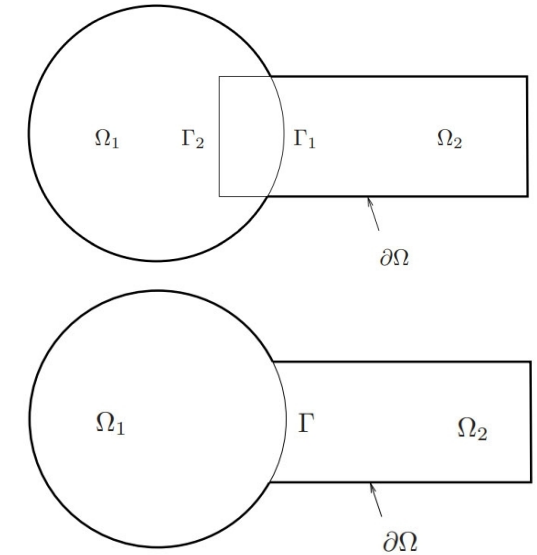


Figure above: tension specimen simulation coupling composite TET10 elements with HEX elements in Sierra/SM.

Figures right: bolted joint simulation coupling composite TET10 elements with HEX elements in Sierra/SM.

¹ Mota *et al.* 2017; Mota *et al.* 2022.

- The Schwarz Alternating Method for Domain Decomposition-Based Coupling
- **Extension to FOM*-ROM# and ROM-ROM Coupling**
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work

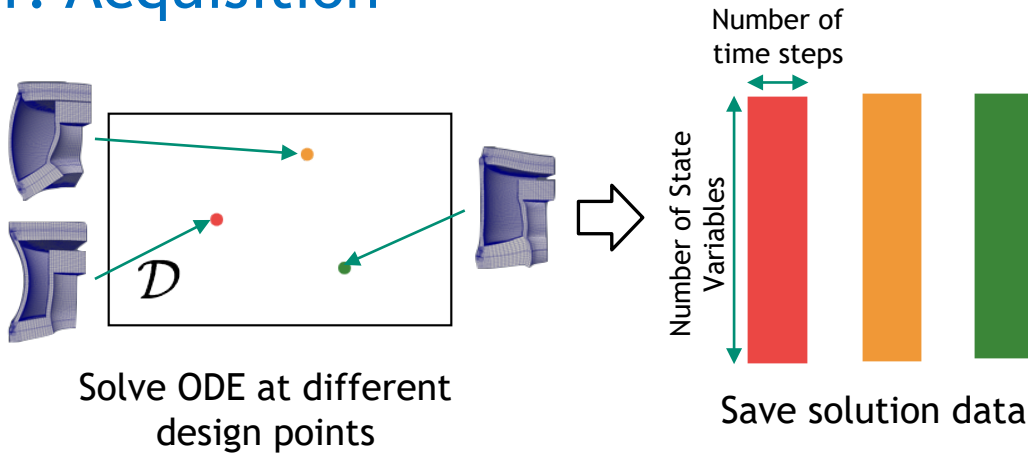


Projection-Based Model Order Reduction via the POD/LSPG* Method

Full Order Model (FOM): $\frac{du}{dt} = f(u; t, \mu)$

* Least-Squares Petrov-Galerkin

1. Acquisition



2. Learning

Proper Orthogonal Decomposition (POD):

$$\mathbf{X} = \begin{bmatrix} \text{red bar} & \text{orange bar} & \text{green bar} \end{bmatrix} = \begin{bmatrix} \text{brown bar} & \text{blue bar} \end{bmatrix} \begin{bmatrix} \Sigma & \mathbf{v}^T \end{bmatrix}$$

Diagram illustrating the Proper Orthogonal Decomposition (POD) process. The matrix \mathbf{X} (represented by three vertical bars: red, orange, green) is decomposed into the product of a matrix Φ (represented by a brown and a blue vertical bar) and a matrix \mathbf{U} (represented by a light blue square). The matrix \mathbf{U} is further decomposed into Σ (represented by a diagonal line) and \mathbf{v}^T (represented by a light blue square).

ROM = projection-based Reduced Order Model

3. Projection-Based Reduction

Choose ODE temporal discretization

$$\frac{du}{dt} = f(u; t, \mu) \downarrow r^n(u^n; \mu) = 0, n = 1, \dots, T$$

Reduce the number of unknowns

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

Minimize residual

$$\text{minimize}_{\hat{v}} \left\| \begin{bmatrix} A & r^n(\Phi \hat{v}; \mu) \end{bmatrix} \right\|_2$$

Hyper-reduction/sample mesh

HROM = Hyper-reduced ROM

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

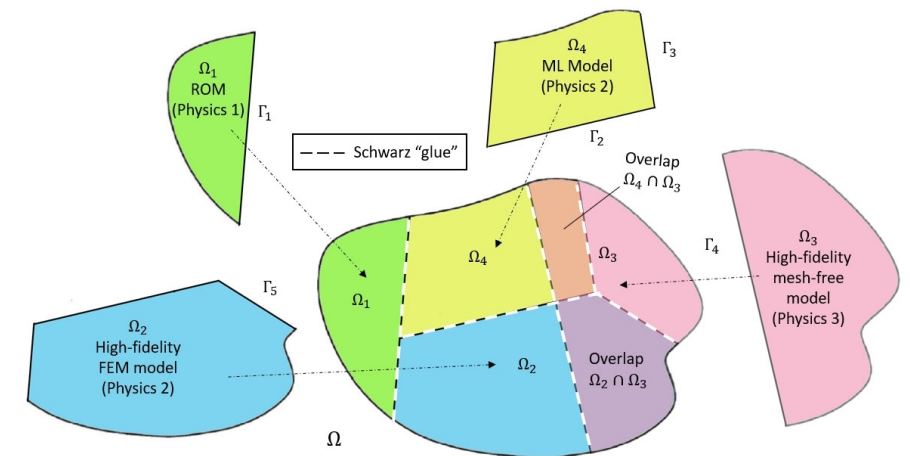
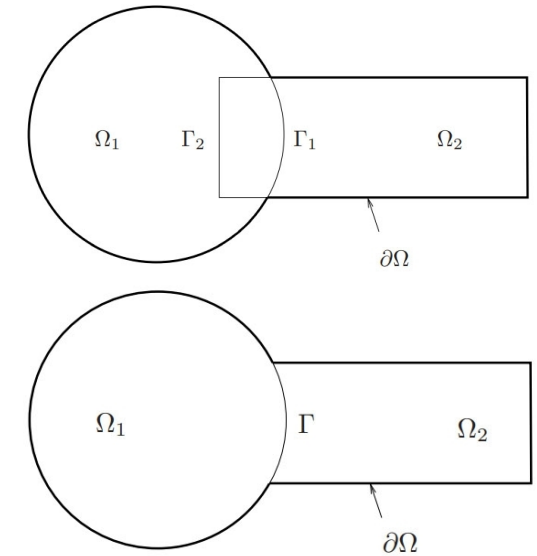
- **Strong** vs. **weak** BC enforcement?
 - Strong BC enforcement difficult for some models (e.g., cell-centered finite volume, PINNs)
- **Optimizing parameters** in Schwarz BCs for non-overlapping Schwarz?

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



- The Schwarz Alternating Method for Domain Decomposition-Based Coupling
- Extension to FOM*-ROM[#] and ROM-ROM Coupling
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work

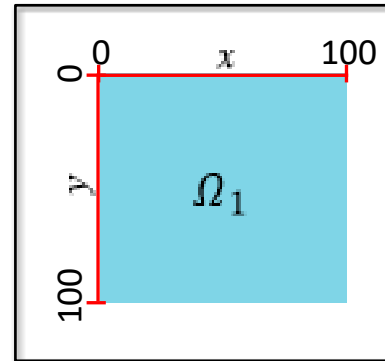


2D Inviscid Burgers Equation



Popular analog for fluid problems where **shocks** are possible, and particularly **difficult** for conventional projection-based ROMs

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{1}{2} \left(\frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} \right) &= 0.02 \exp(\mu_2 x) \\ \frac{\partial v}{\partial t} + \frac{1}{2} \left(\frac{\partial (vu)}{\partial x} + \frac{\partial (v^2)}{\partial y} \right) &= 0 \\ u(0, y, t; \boldsymbol{\mu}) &= \mu_1 \\ u(x, y, 0) &= v(x, y, 0) = 1 \end{aligned}$$



Problem setup:

- $\Omega = (0, 100)^2$, $t \in [0, 25]$
- Two parameters $\boldsymbol{\mu} = (\mu_1, \mu_2)$. **Training:** uniform sampling of $\boldsymbol{\mu} \in [4.25, 5.50] \times [0.015, 0.03]$ by a 3×3 grid. **Testing:** query unsampled point $\boldsymbol{\mu} = [4.75, 0.02]$

FOM discretization:

- Spatial discretization given by a **Godunov-type scheme** with $N = 250$ elements in each dimension
- Implicit **trapezoidal method** with fixed $\Delta t = 0.05$

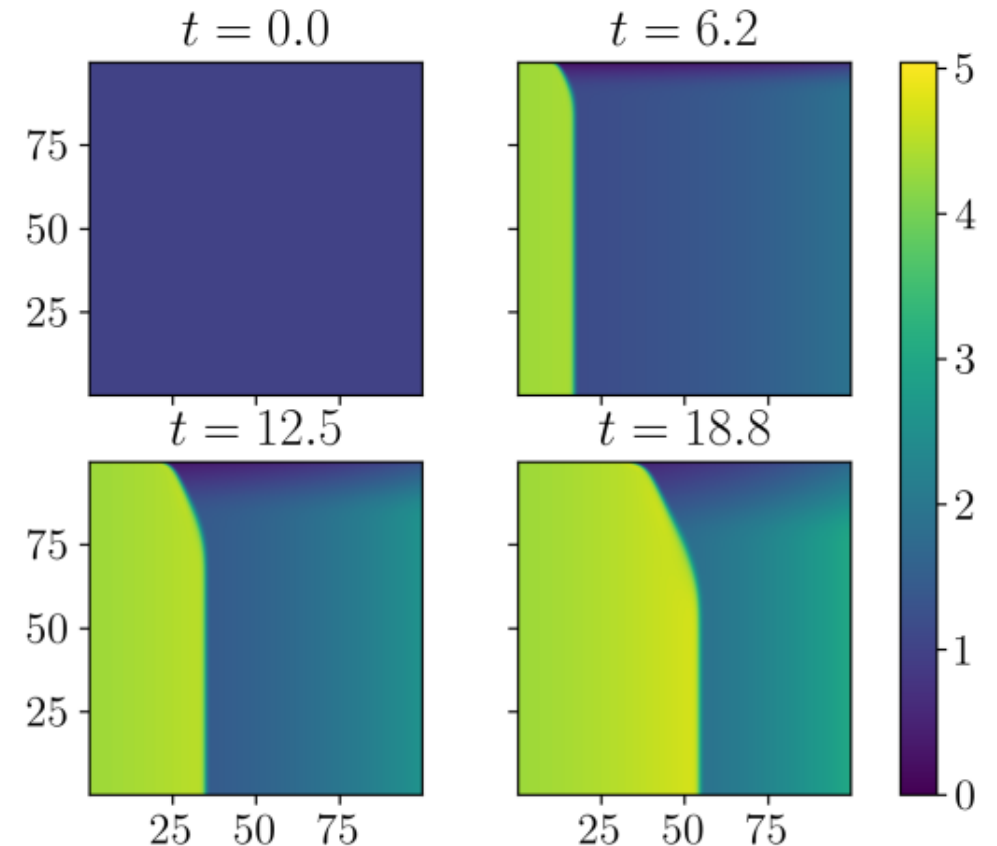
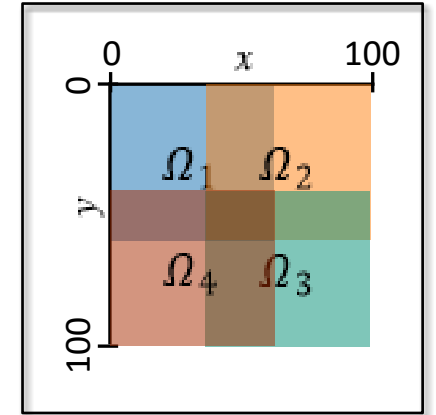
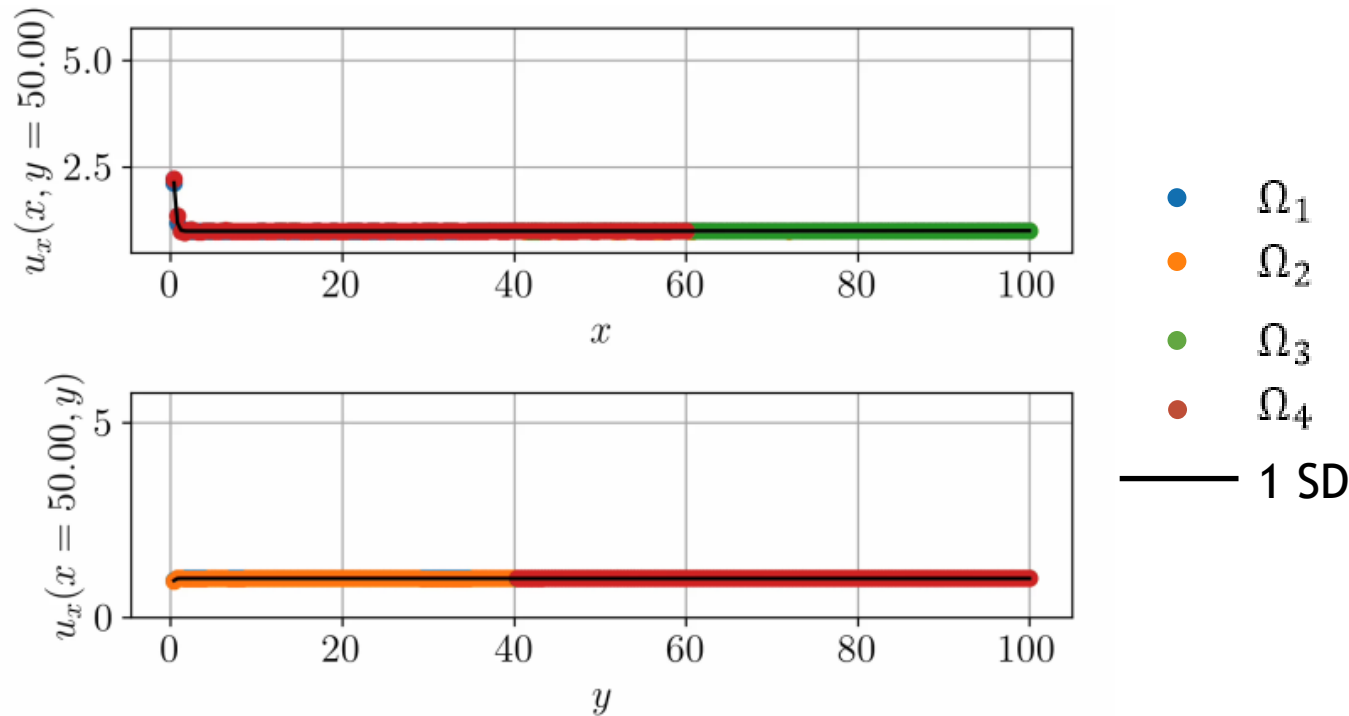


Figure above: solution of u component at various times



No
Image

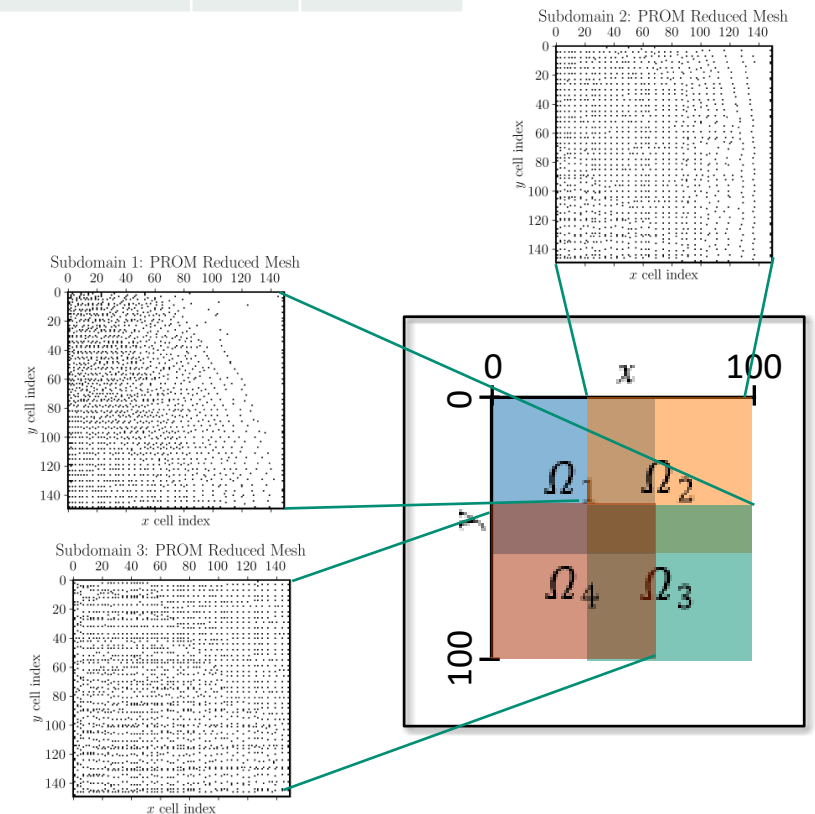
FOM-HROM-HROM-HROM Coupling



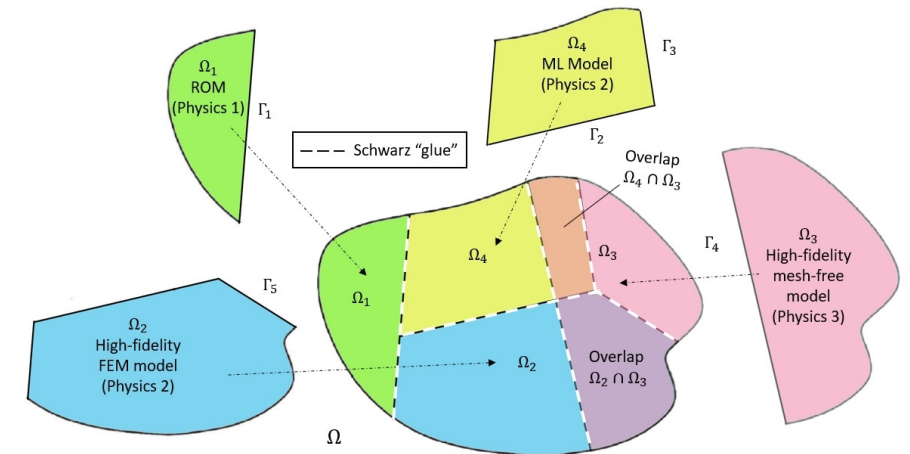
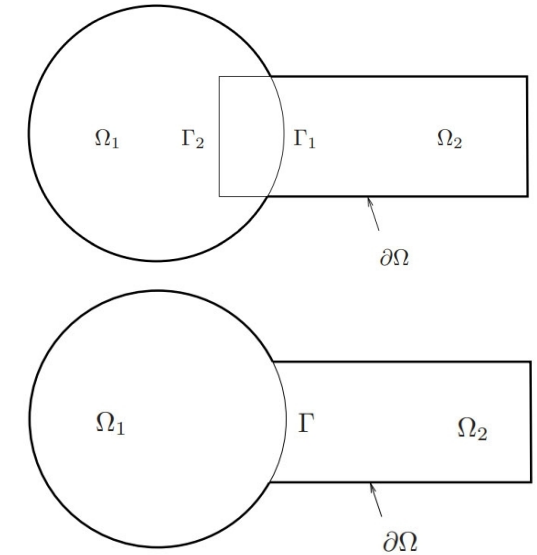
- FOM in Ω_1 as this is “hardest” subdomain for ROM
- HROMs in $\Omega_2, \Omega_3, \Omega_4$ capture 99% snapshot energy
- Method converges in 3 Schwarz iterations per controller time-step
- Errors $O(0.1\%)$ with 0 error in Ω_1
- 2.26 \times speedup achieved over all-FOM coupling

Further speedups possible via code optimizations, additive Schwarz and reduction of # sample mesh points.

Subdomains	99% SV Energy		
	M	MSE (%)	CPU time (s)
Ω_1	—	0.0	95
Ω_2	120	0.26	26
Ω_3	60	0.43	17
Ω_4	66	0.34	21
Total			159



- The Schwarz Alternating Method for Domain Decomposition-Based Coupling
- Extension to FOM*-ROM[#] and ROM-ROM Coupling
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work



2D Shallow Water Equations (SWE)



Hyperbolic PDEs modeling **wave propagation** below a pressure surface in a fluid (e.g., atmosphere, ocean).

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} &= 0 \\ \frac{\partial(hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial(huv)}{\partial y} &= -\mu v \\ \frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) &= \mu u \end{aligned}$$

Problem setup:

- $\Omega = (-5, 5)^2$, $t \in [0, 10]$, Gaussian initial condition
- **Coriolis parameter** $\mu \in \{-4, -3, -2, -1, 0\}$ for training, and $\mu \in \{-3.5, -2.5, -1.5, -0.5\}$ for testing

FOM discretization:

- Spatial discretization given by a first-order **cell-centered finite volume** discretization with $N = 300$ elements in each dimension
- Implicit first order temporal discretization: **backward Euler** with fixed $\Delta t = 0.01$
- Implemented in **Pressio-demoapps** (<https://github.com/Pressio/pressio-demoapps>)

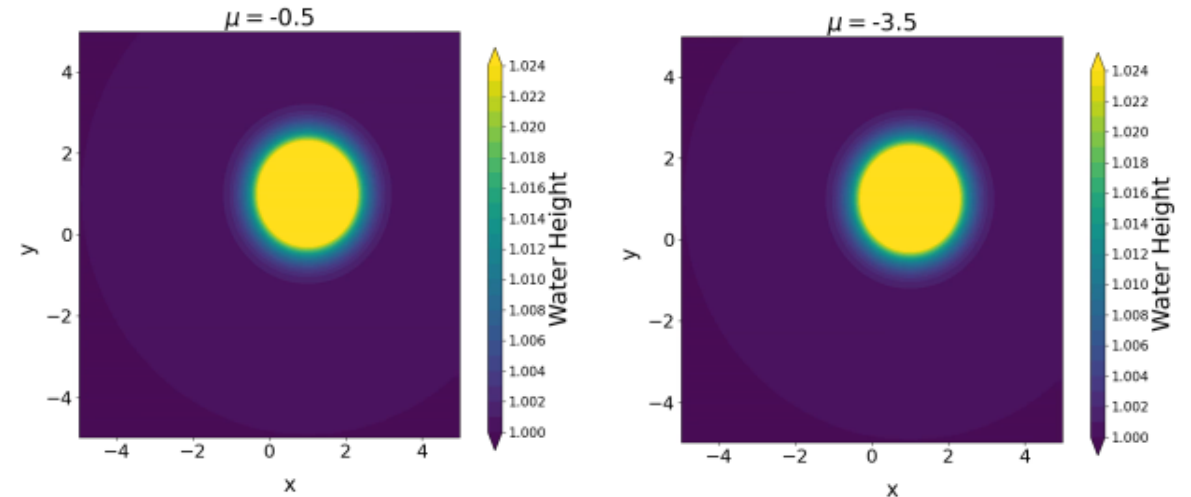


Figure above: FOM solutions to SWE for $\mu = -0.5$ (left) and $\mu = -3.5$ (right).

Schwarz Coupling Details

Green: different from Burgers' problem

Choice of domain decomposition

- **Non-overlapping** DD of Ω into 4 subdomains coupled via **additive Schwarz**
 - **OpenMP parallelism** with 1 thread/subdomain
- **All-ROM** or **All-HROM** coupling via Pressio*

Snapshot collection and reduced basis construction

- **Single-domain FOM** on Ω used to generate snapshots/POD modes

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

- BCs are imposed **approximately** by fictitious ghost cell states
 - Implementing Neumann and Robin BCs is **challenging**
- **Ghost cells** introduce some overlap even with non-overlapping DD
 - \Rightarrow **Dirichlet-Dirichlet non-overlapping Schwarz** is stable/convergent!

Choice of hyper-reduction

- **Collocation** for hyper-reduction: min residual at small subset DOFs
- Assume **fixed budget of sample mesh points** at Schwarz boundaries

*<https://github.com/Pressio/pressio-demoapps>

Figure right: non-overlapping DD w/ ghost cells creating overlap

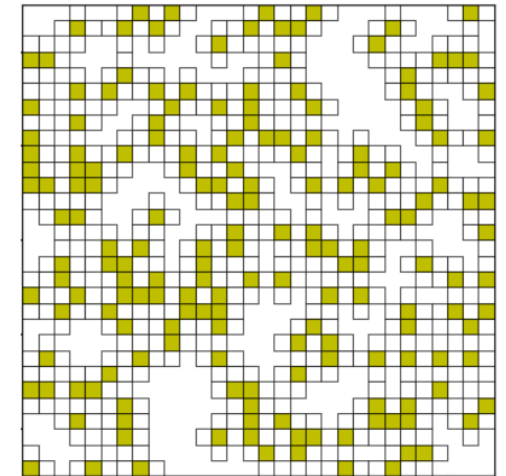
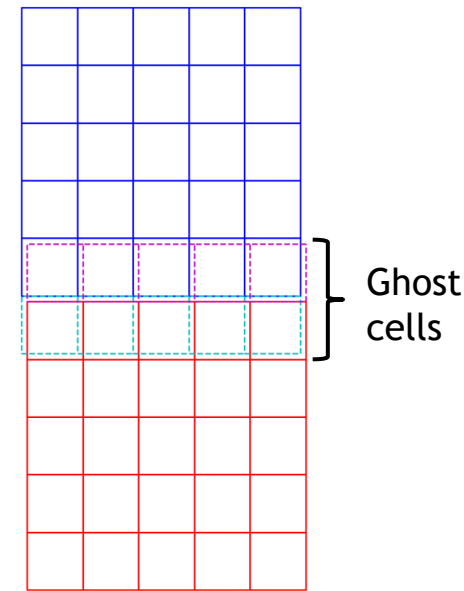
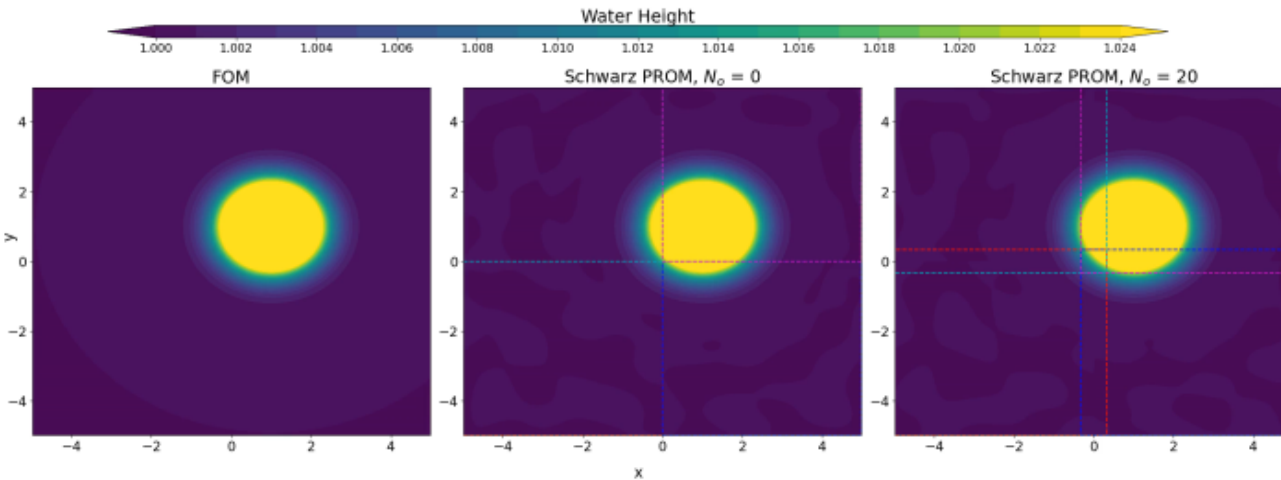


Figure above: sample mesh (yellow) and stencil (white) cells

Schwarz All-ROM Domain Overlap Study



Study of Schwarz convergence for all-ROM coupling as a function of $N_o :=$ cell width of overlap region (not including ghost cells).

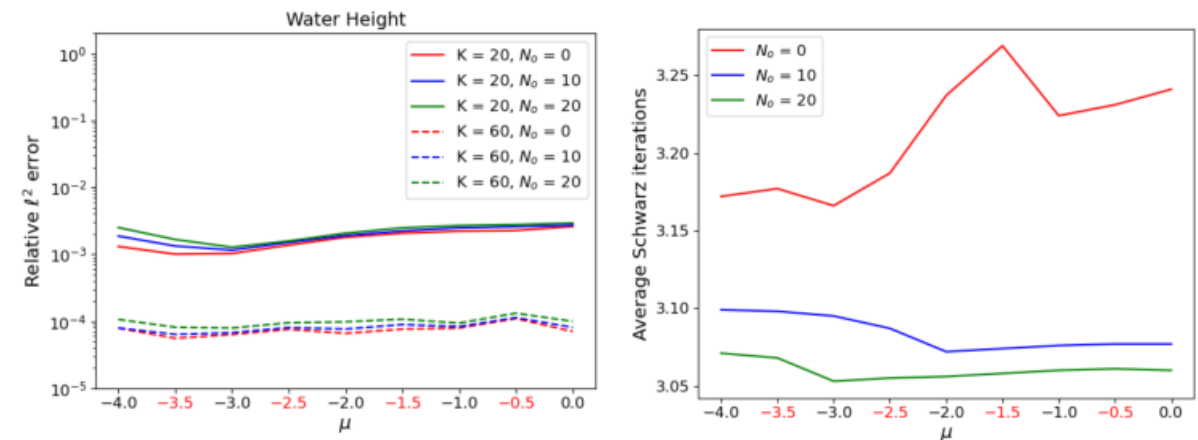


Movie above: FOM (left), 4 subdomain ROM coupled via non-overlapping Schwarz (middle), and 4 subdomain ROM coupled via overlapping Schwarz (right) for predictive SWE problem with $\mu = -0.5$. All ROMs have $K = 80$ POD modes.

- Schwarz iterations decrease (very roughly) with $N_o^{0.25}$ (figure, right) whereas evaluating $r(q)$ scales with N_o^2

➤ \Rightarrow there is no reason not to do non-overlapping coupling for this problem

- Dirichlet-Dirichlet coupling with no-overlap ($N_o = 0$) performs well with no convergence issues (movie, left) and errors comparable to Dirichlet-Dirichlet coupling with overlap (figure below, left)



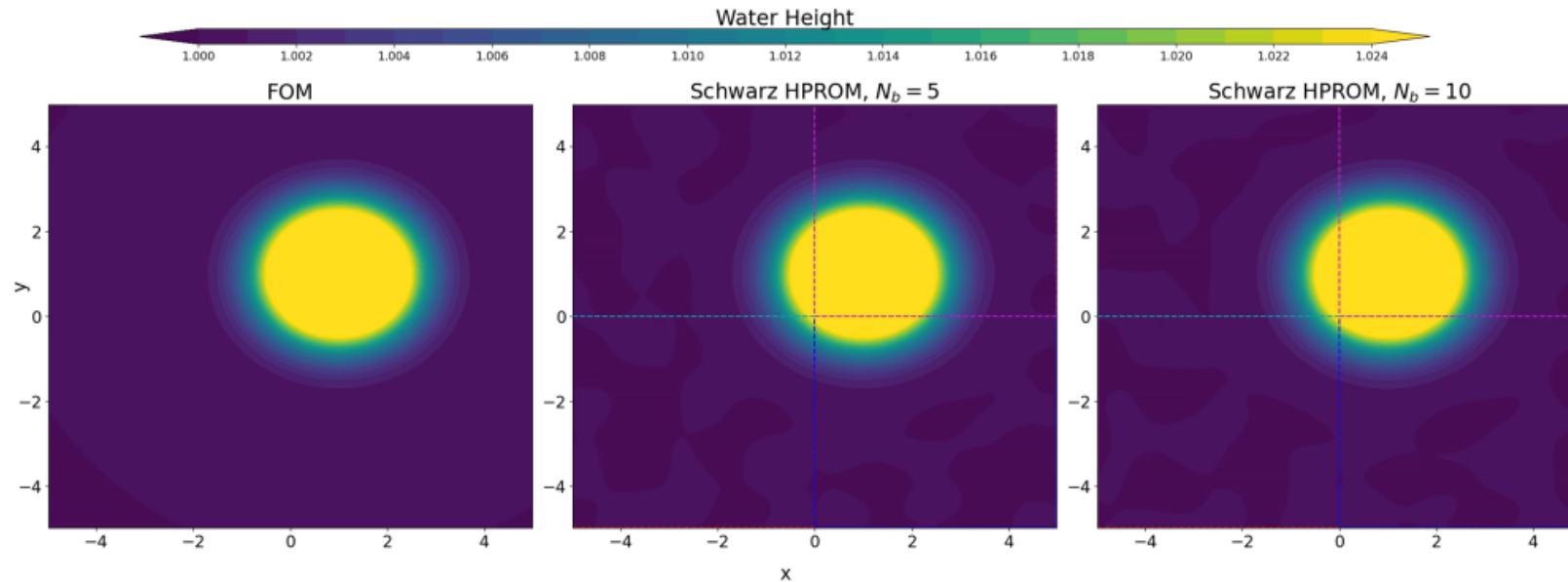
Figures above: relative error and average # Schwarz iterations as a function of μ and N_o . Black μ : training, red μ : testing.

Schwarz Boundary Sampling for All-HROM Coupling



Key question: how many Schwarz boundary points need to be included in **sample mesh** when performing HROM coupling?

- Naïve/sparsely-sampled Schwarz boundary results in **failure** to transmit coupling information during Schwarz



Movie above: FOM (left), all HROM with $N_b = 5\%$ (middle) and all HROM with $N_b = 10\%$ (left). ROMs have $K = 100$ modes and $N_s = 0.5\%N$ sample mesh points.

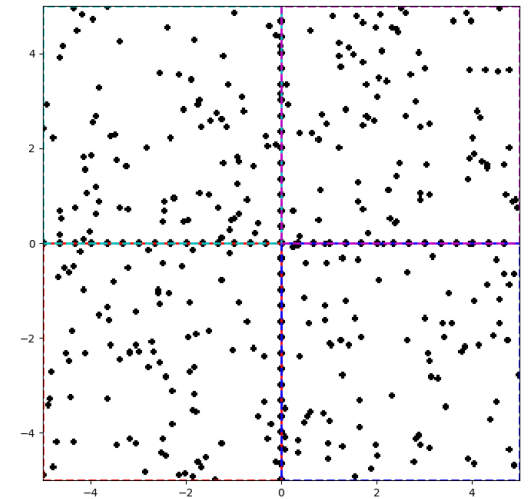
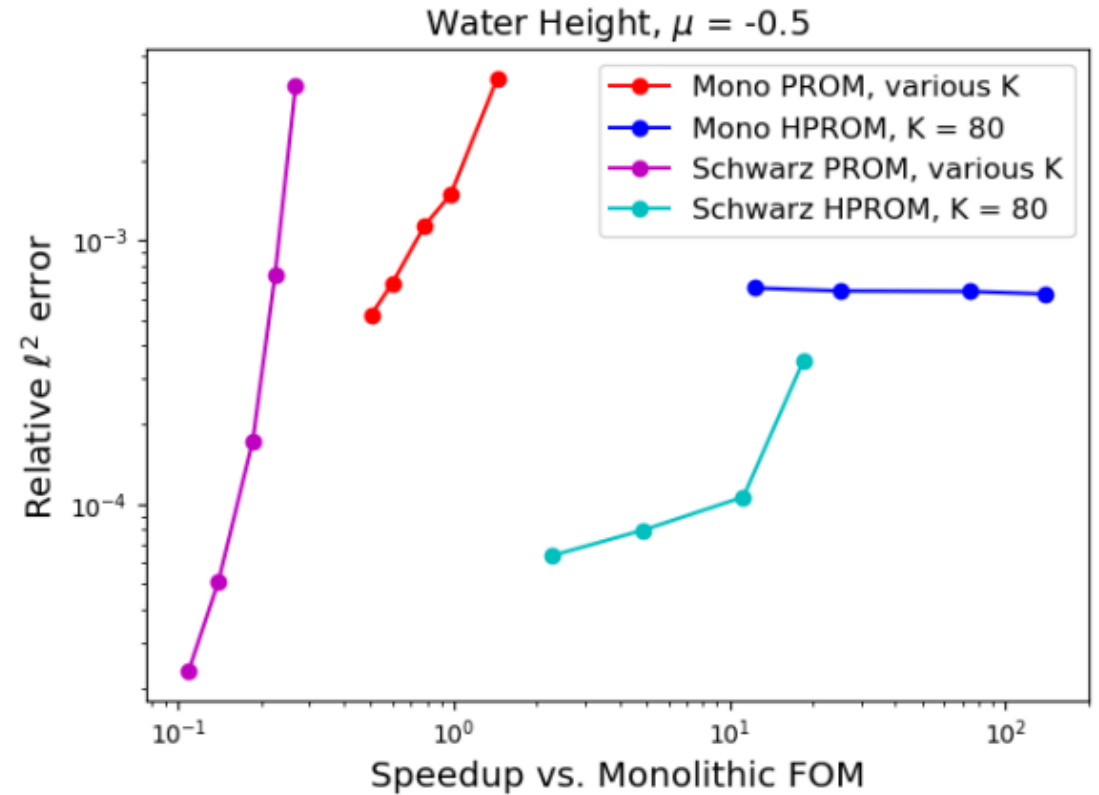
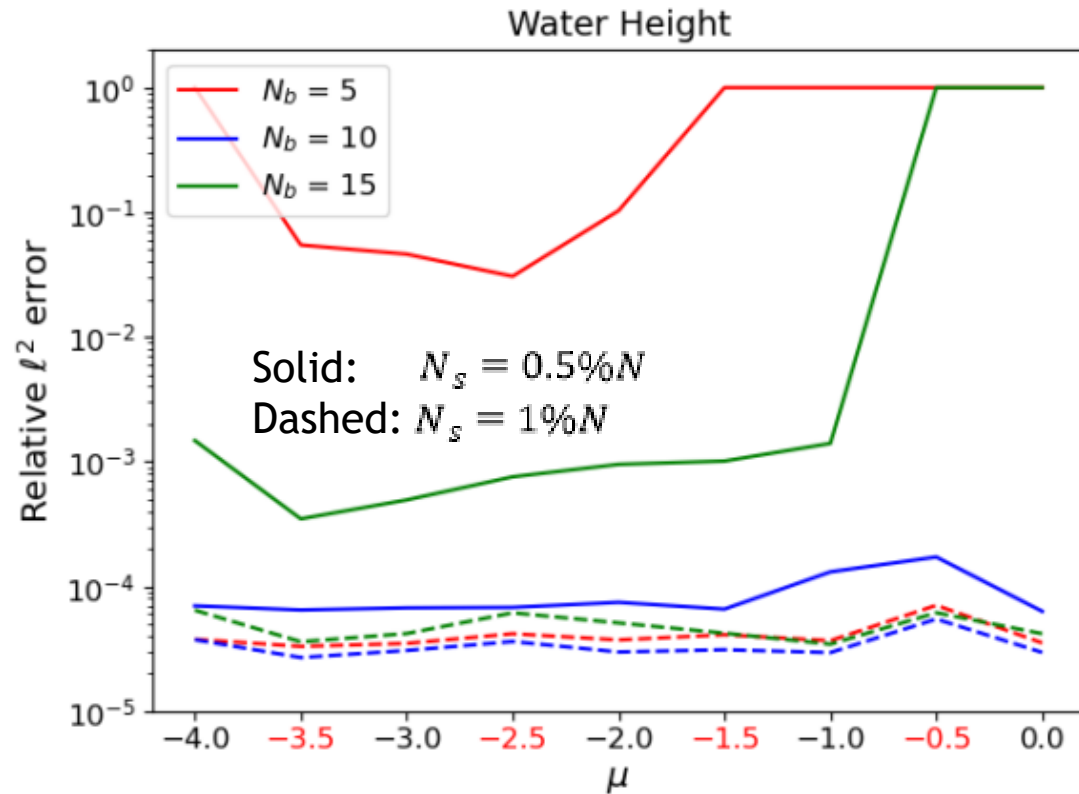


Figure above: example sample mesh with sampling rate $N_b = 10\%$

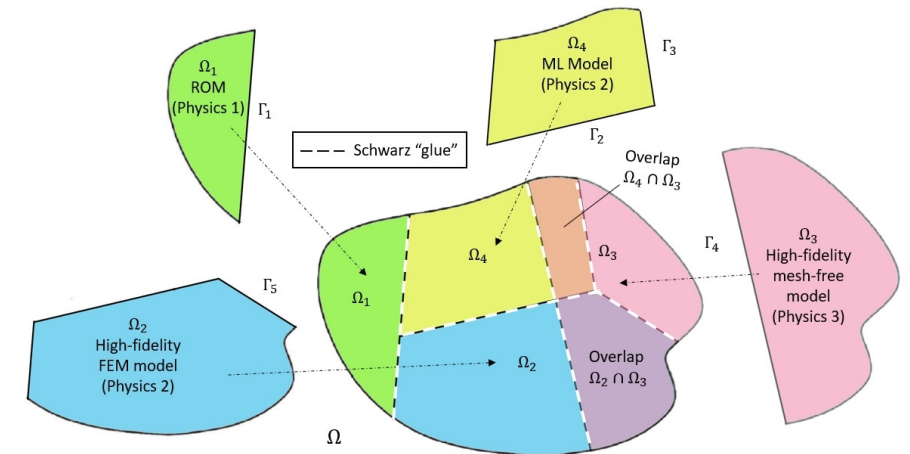
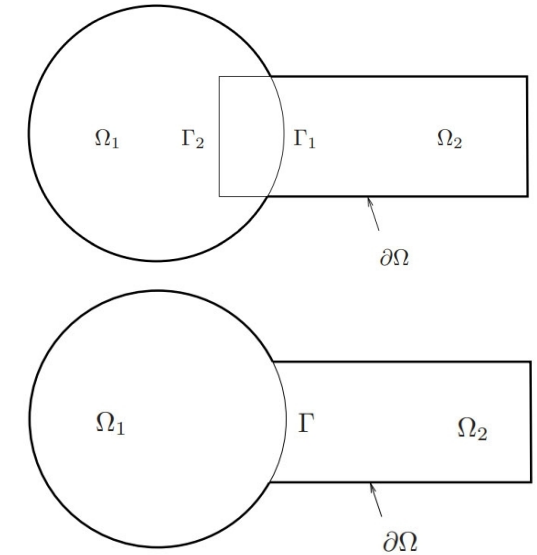
- Including too many Schwarz boundary points (N_b) in sample mesh given fixed budget of N_s sample mesh points may lead to too few sample mesh points in interior
- For SWE problem, we can get away with $\sim 10\%$ boundary sampling (movie above, right-most frame)

Coupled HRROM Performance



- For a fixed ROM dimension, Schwarz delivers **lower error** and **comparable cost**!
- There are noticeable **cost savings** relative to **monolithic FOM**!
- Accuracy similar for **predictive μ** (red) and **non-predictive μ** (black) cases.

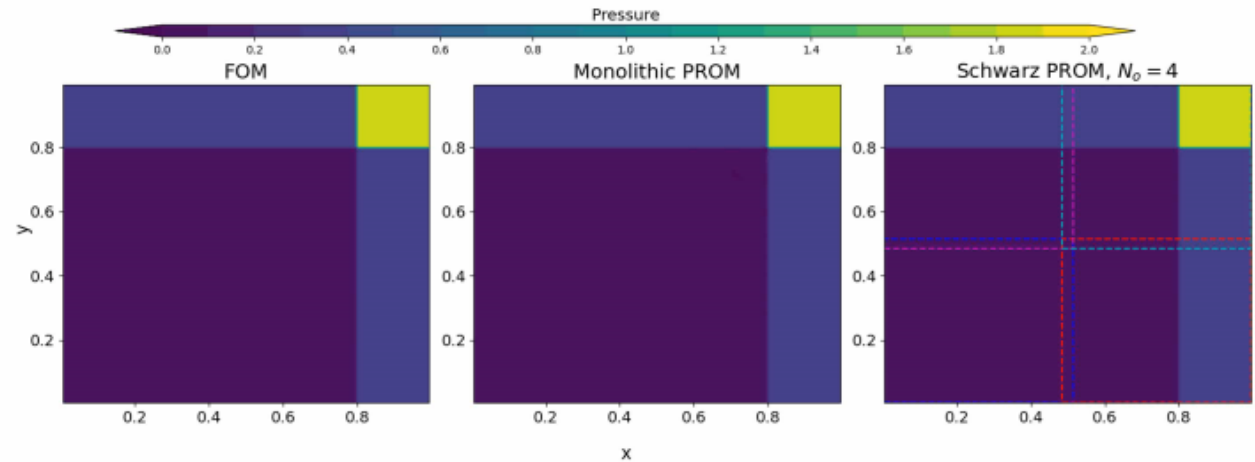
- The Schwarz Alternating Method for Domain Decomposition-Based Coupling
- Extension to FOM*-ROM[#] and ROM-ROM Coupling
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work



Teaser: 2D Euler Equations Riemann Problem

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{pmatrix} = \mathbf{0}$$

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \rho (u^2 + v^2) \right)$$



Problem setup:

- $\Omega = (0,1)^2$, $t \in [0, 0.8]$, homogeneous Neumann BCs
- Fix $\rho_1 = 1.5$, $u_1 = v_1 = 0$, $p_3 = 0.029$
- Vary p_1 ; IC from compatibility conditions*
 - Training: $p_1 \in [1.0, 1.25, 1.5, 1.75, 2.0]$
 - Testing: $p_1 \in [1.125, 1.375, 1.625, 1.875]$

FOM discretization:

- Spatial discretization given by a first-order **cell-centered finite volume** discretization with $N = 300$ or $N = N = 100$ elements in each dimension
- Implicit first order temporal discretization: **backward Euler** with fixed $\Delta t = 0.005$
- Implemented in **Pressio-demoapps** (<https://github.com/Pressio/pressio-demoapps>)

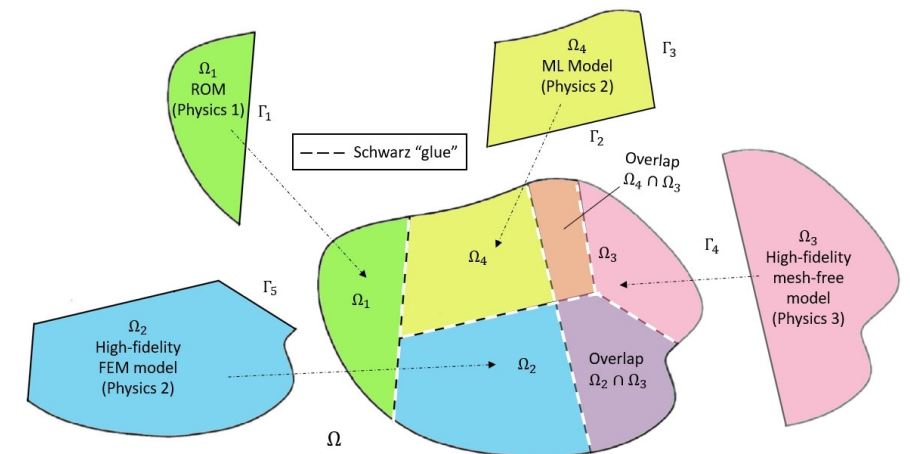
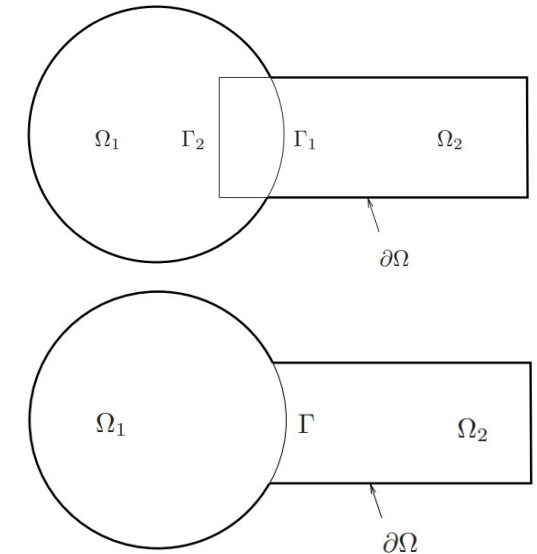
*Schulz-Rinne, 1993.

Preliminary results:

- Schwarz can **stabilize** unstable monolithic ROM for fixed dimension K (above)
- Since shock traverses all parts of domain, achieving **speedups** with Schwarz is **more difficult**



- The Schwarz Alternating Method for Domain Decomposition-Based Coupling
- Extension to FOM*-ROM[#] and ROM-ROM Coupling
- Numerical Examples
 - 2D Burgers Equation
 - 2D Shallow Water Equations
 - Teaser: 2D Euler Equations Riemann Problem
- Summary & Future Work

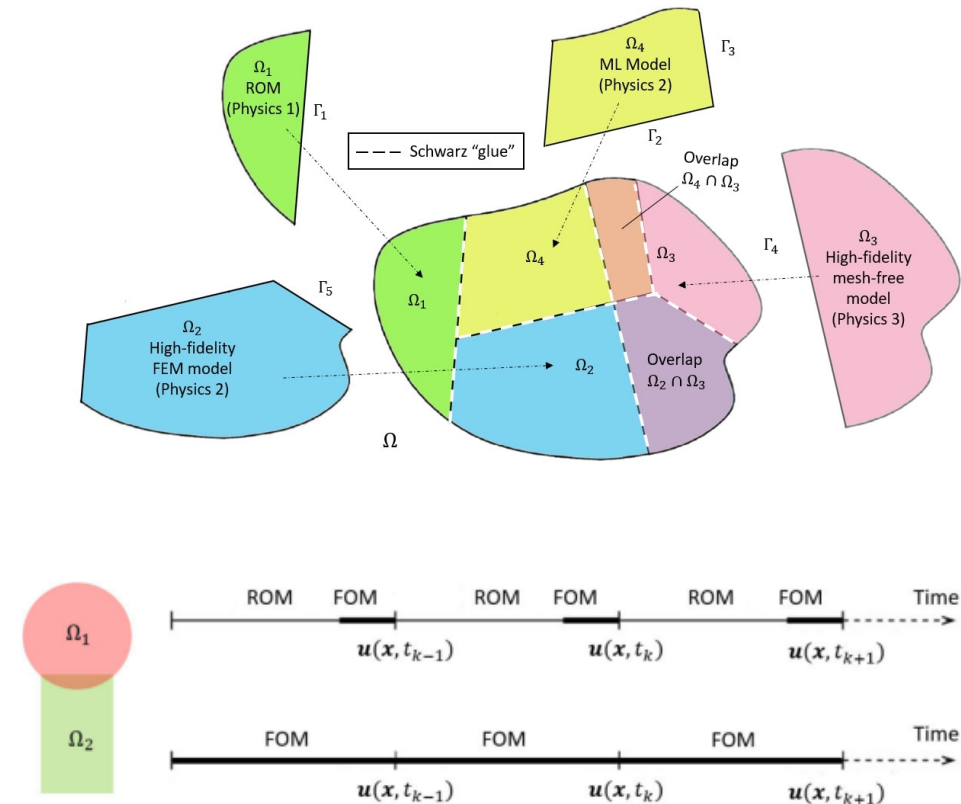


Summary:

- Schwarz has been **demonstrated** for **coupling** of FOMs and (H)ROMs
- **Computational gains** can be achieved by coupling (H)ROMs and using the additive Schwarz variant

Ongoing & future work:

- Extension to **other applications** (fasteners, laser welds)
- **Rigorous analysis** of why Dirichlet-Dirichlet BC “work” when employing non-overlapping Schwarz with discretizations that employ ghost cells
- **Learning** of “optimal” transmission conditions to ensure **structure preservation**
- Extension of Schwarz to enabling coupling of **non-intrusive ROMs** (e.g., DMD, OpInf, Neural Networks)
- Development of **automated criteria** to determine appropriate use of less refined or reduced-order models without sacrificing accuracy, enabling **real-time transitions** between different model fidelities



Team & Acknowledgments



Irina Tezaur



Chris Wentland



Francesco Rizzi



Joshua Barnett



Alejandro Mota



Will Snyder



Ian Moore



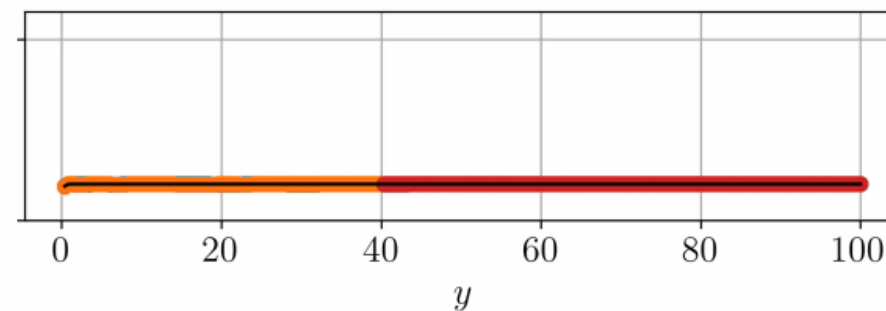
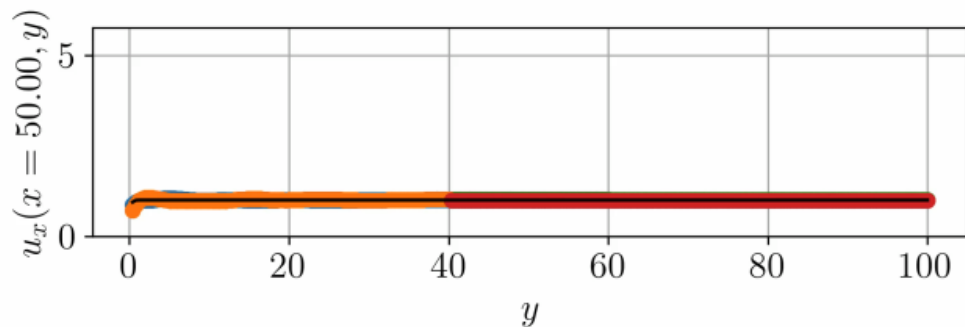
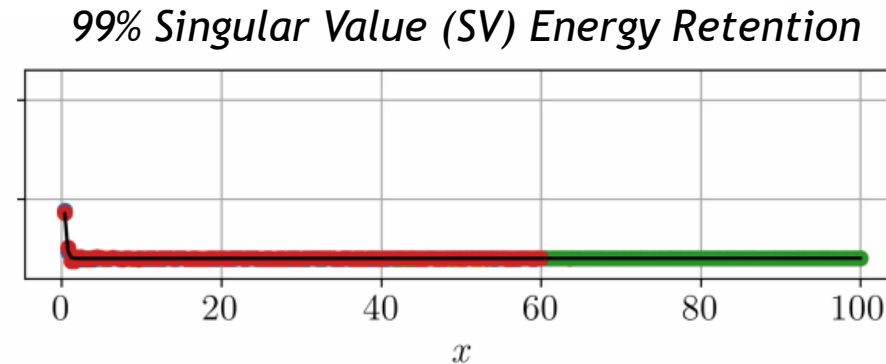
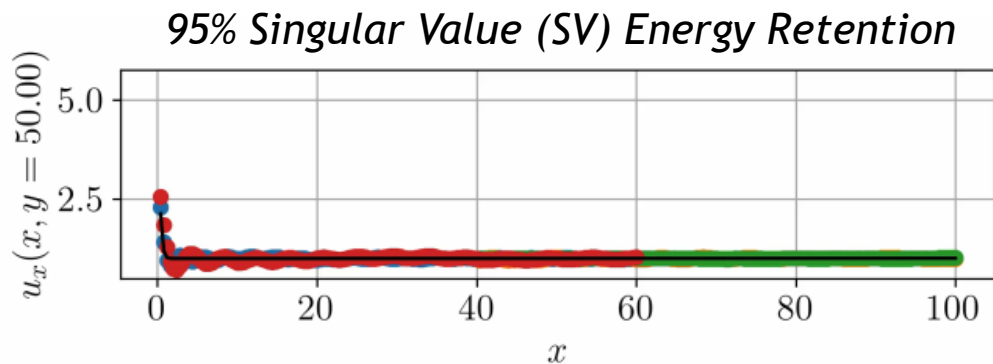
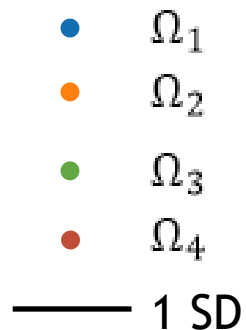


- [1] A. Mota, I. Tezaur, C. Alleman. “The Schwarz Alternating Method in Solid Mechanics”, *Comput. Meth. Appl. Mech. Engng.* 319 (2017), 19-51.
- [2] A. Mota, I. Tezaur, G. Phlipot. “The Schwarz Alternating Method for Dynamic Solid Mechanics”, *Comput. Meth. Appl. Mech. Engng.* 121 (21) (2022) 5036-5071.
- [3] J. Barnett, I. Tezaur, A. Mota. “The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models”, ArXiv pre-print, 2022.
<https://arxiv.org/abs/2210.12551>
- [4] W. Snyder, I. Tezaur, C. Wentland. “Domain decomposition-based coupling of physics-informed neural networks via the Schwarz alternating method”, ArXiv pre-print, 2023.
<https://arxiv.org/abs/2311.00224>
- [5] A. Mota, D. Koliesnikova, I. Tezaur. “A Fundamentally New Coupled Approach to Contact Mechanics via the Dirichlet-Neumann Schwarz Alternating Method”, ArXiv pre-print, 2023.
<https://arxiv.org/abs/2311.05643>

Email: ikalash@sandia.gov
URL: www.sandia.gov/~ikalash

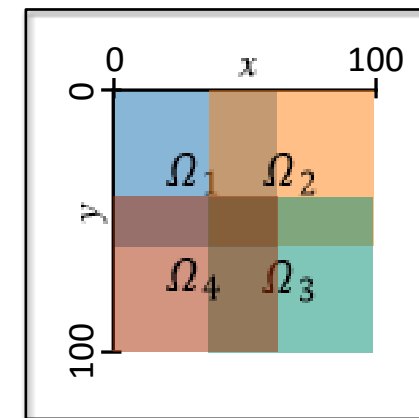
Start of Backup Slides

All-ROM Coupling



- Method converges in **only 3 Schwarz iterations** per controller time-step
- **Errors $O(1\%)$ or less**
- **$1.47\times$ speedup** over all-FOM coupling for 95% SV energy retention case

Subdomains	95% SV Energy			99% SV Energy		
	M	MSE (%)	CPU time (s)	M	MSE (%)	CPU time (s)
Ω_1	57	1.1	85	146	0.18	295
Ω_2	44	1.2	56	120	0.18	216
Ω_3	24	1.4	43	60	0.16	89
Ω_4	32	1.9	61	66	0.25	100
Total			245			700





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

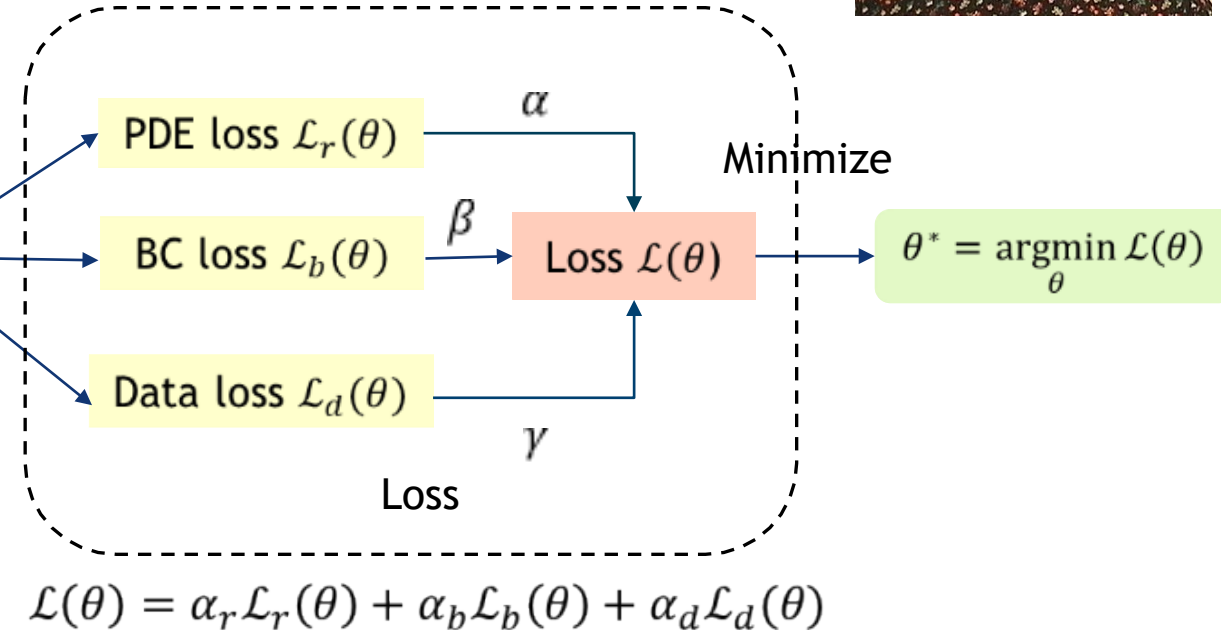
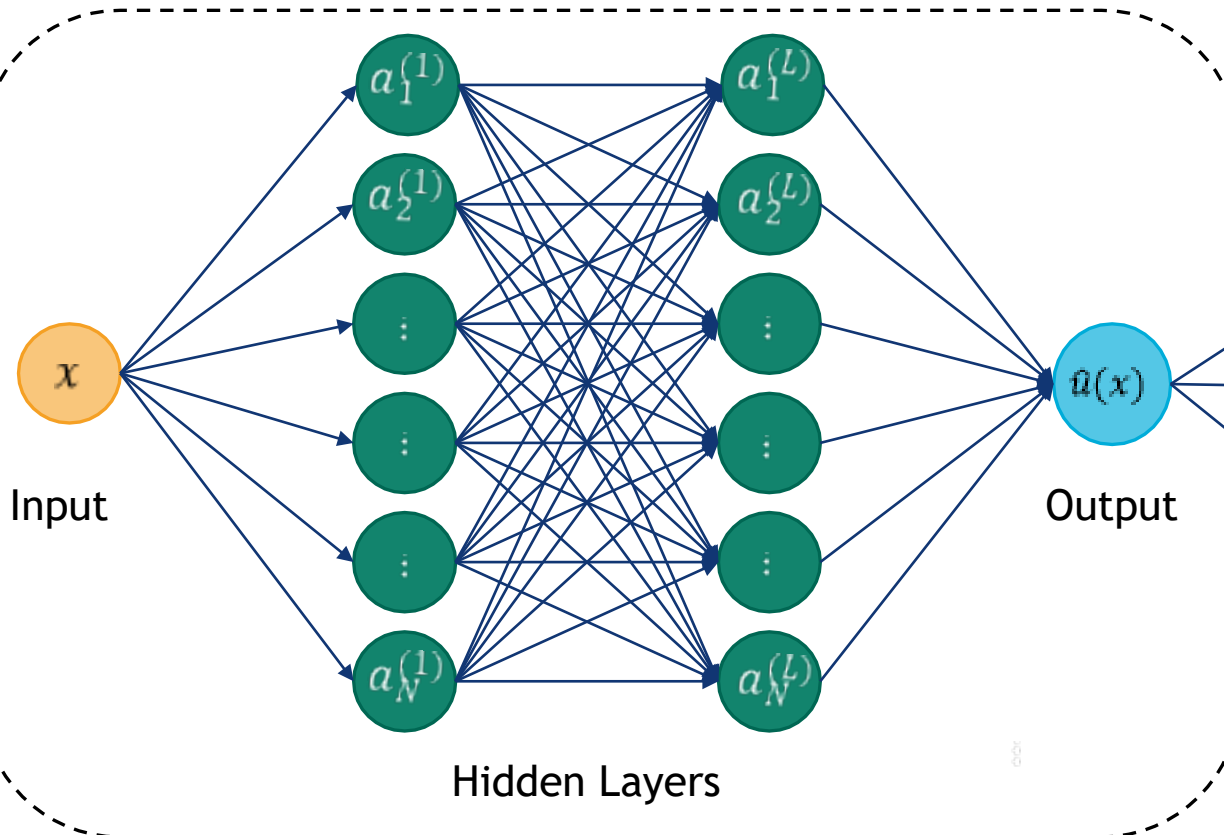
- ☺ Coupling is *concurrent* (two-way).
- ☺ *Ease of implementation* into existing massively-parallel HPC codes.
- ☺ “*Plug-and-play*” *framework*: simplifies task of meshing complex geometries!
 - ☺ Ability to couple regions with *different non-conformal meshes*, *different element types* and *different 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.

Bonus: PINN-PINN and PINN-FOM coupling

Will Snyder
Summer Intern
Virginia Tech



Neural Network



Focus thus far

Goal: investigate the use of the Schwarz alternating method as a means to couple **Physics-Informed Neural Networks (PINNs)**

Scenario 1: use Schwarz to train subdomain PINNs (offline)

Scenario 2: use Schwarz to couple pre-trained subdomain PINNs/NNs (online)

Bonus: PINN-PINN coupling

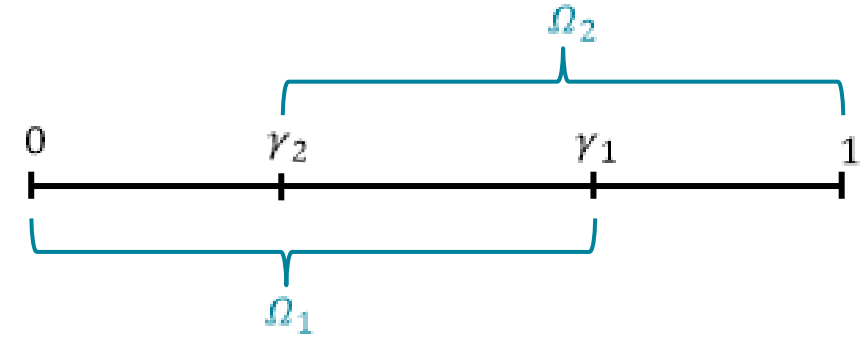


1D steady **advection-diffusion** equation on $\Omega = [0,1]$:

$$u_x - \nu u_{xx} = 1, \quad u(0) = u(1) = 0$$

PINNs are notoriously difficult to train for higher Peclet numbers!

→ *Can Schwarz help?*



Overlapping DD: $\Omega = \Omega_1 \cup \Omega_2$ with boundary $\partial\Omega = \{0,1\}$

Schwarz PINN training algorithm:

Loop over subdomains Ω_i until convergence of Schwarz method

Train PINN in Ω_i with loss $\mathcal{L}_i(\theta) = \alpha \mathcal{L}_{r,i}(\theta) + \beta \mathcal{L}_{b,i}(\theta) + \gamma \mathcal{L}_{d,i}(\theta)$

Communicate Dirichlet data between neighboring subdomains

Update boundary data on γ_i from neighboring subdomains

If **strong enforcement of Dirichlet BC (SDBC)**, set $\hat{u}_{\Omega_i}(x, \theta) = NN_{\Omega_i}(x, \theta)$

If **weak enforcement of Dirichlet BC (WDBC)**, set $\beta = 0$ and $\hat{u}_{\Omega_i}(x, \theta) = v(x)NN_{\Omega_i}(x, \theta) + \psi(x)\hat{u}_{\Omega_j}(\gamma_j, \theta)$ where $v(x)$ is chosen s.t. $v(0) = v(\gamma_i) = v(1) = 0$ and $\psi(x)$ is chosen s.t. $\psi(\gamma_i) = 1$

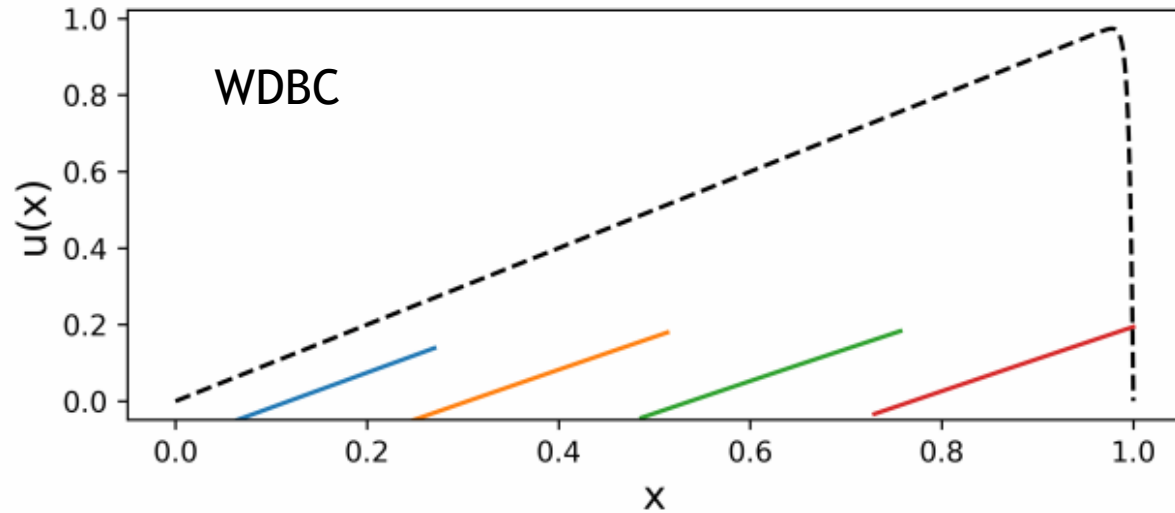
$$\mathcal{L}_{r,i}(\theta) = \text{MSE}\left(-\nu \nabla_x^2 NN_{\Omega_i}(x, \theta) + \nabla_x NN_{\Omega_i}(x, \theta) - 1\right)$$

$$\mathcal{L}_{b,i}(\theta) = \text{MSE}\left(NN_{\Omega_i}(\gamma_i, \theta)\right) + \text{MSE}\left(NN_{\Omega_i}(\gamma_i, \theta) - NN_{\Omega_j}(\gamma_i, \theta)\right)$$

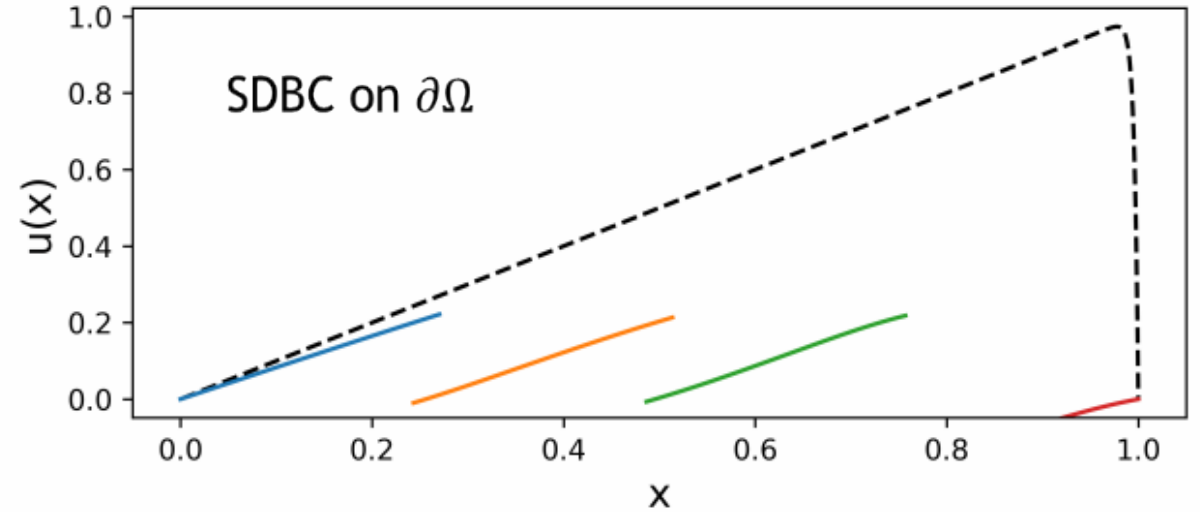
Bonus: PINN-PINN coupling



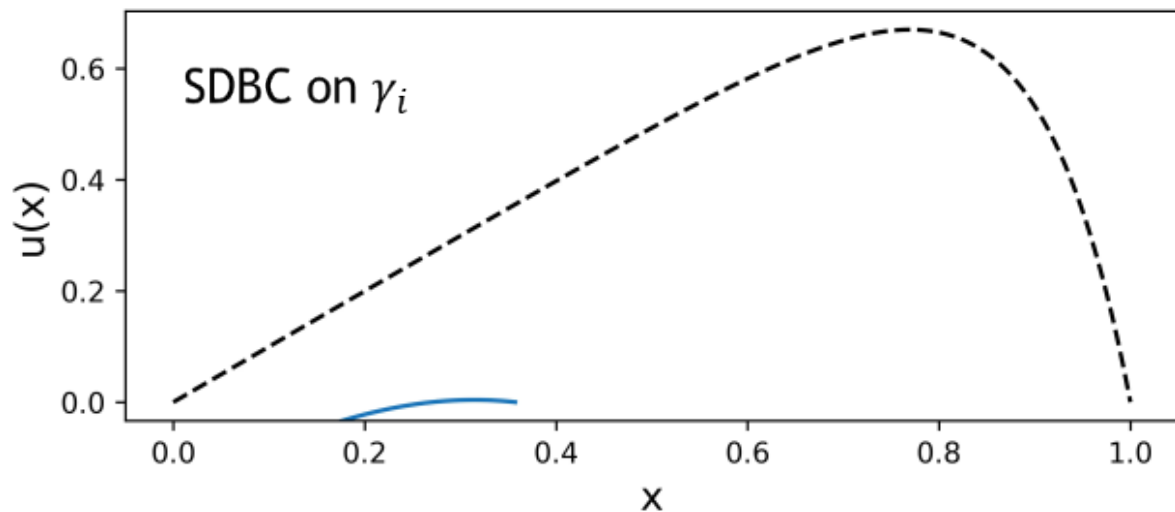
Schwarz iteration 1; $Pe = 250$



Schwarz iteration 1; $Pe = 250$

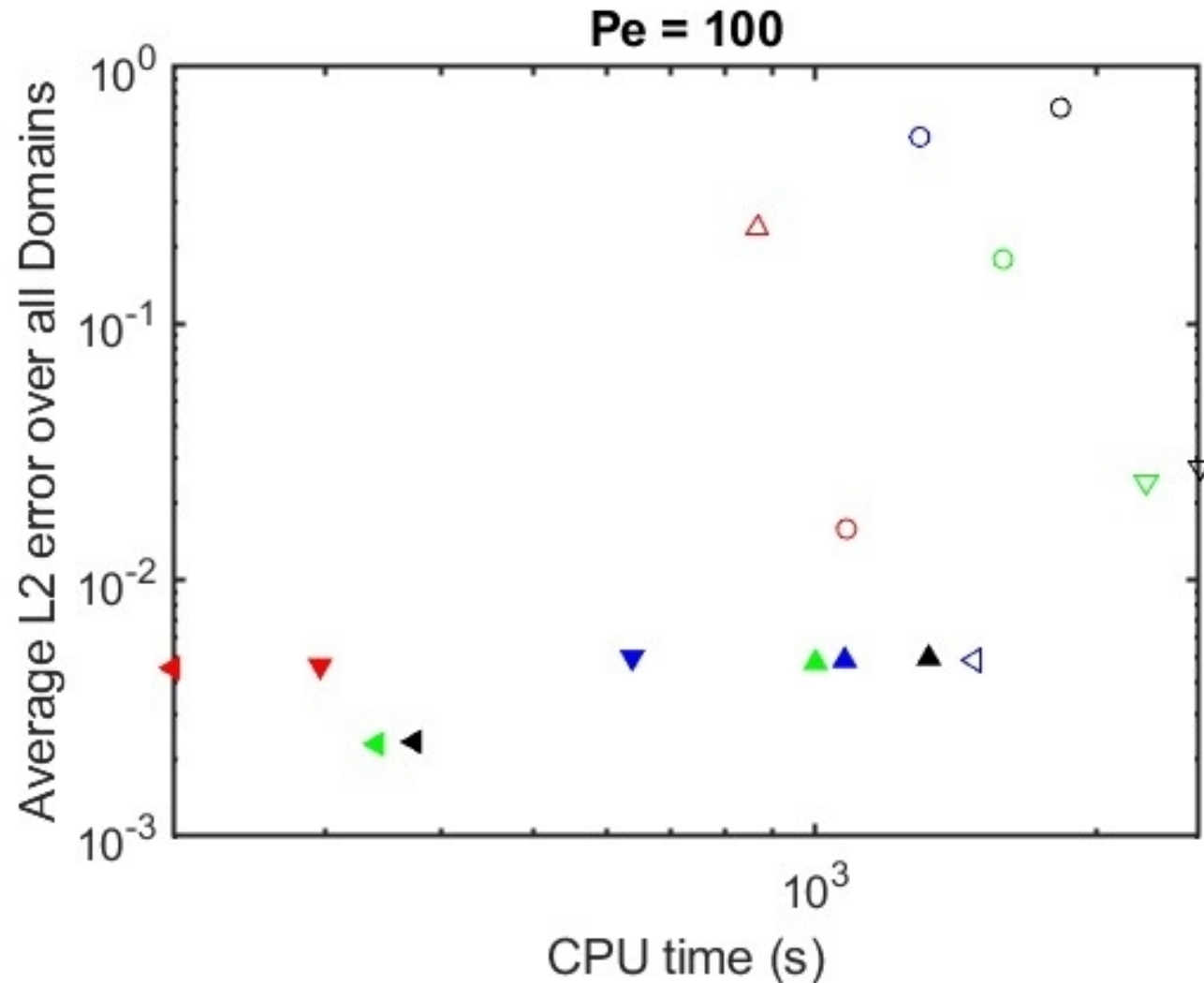


Schwarz iteration 1; $Pe = 10$



- How **Dirichlet boundary conditions** are handled has a large impact on PINN convergence
- Convergence not improved in general with **increasing overlap**
- Increasing # **subdomains** in general will increase CPU time

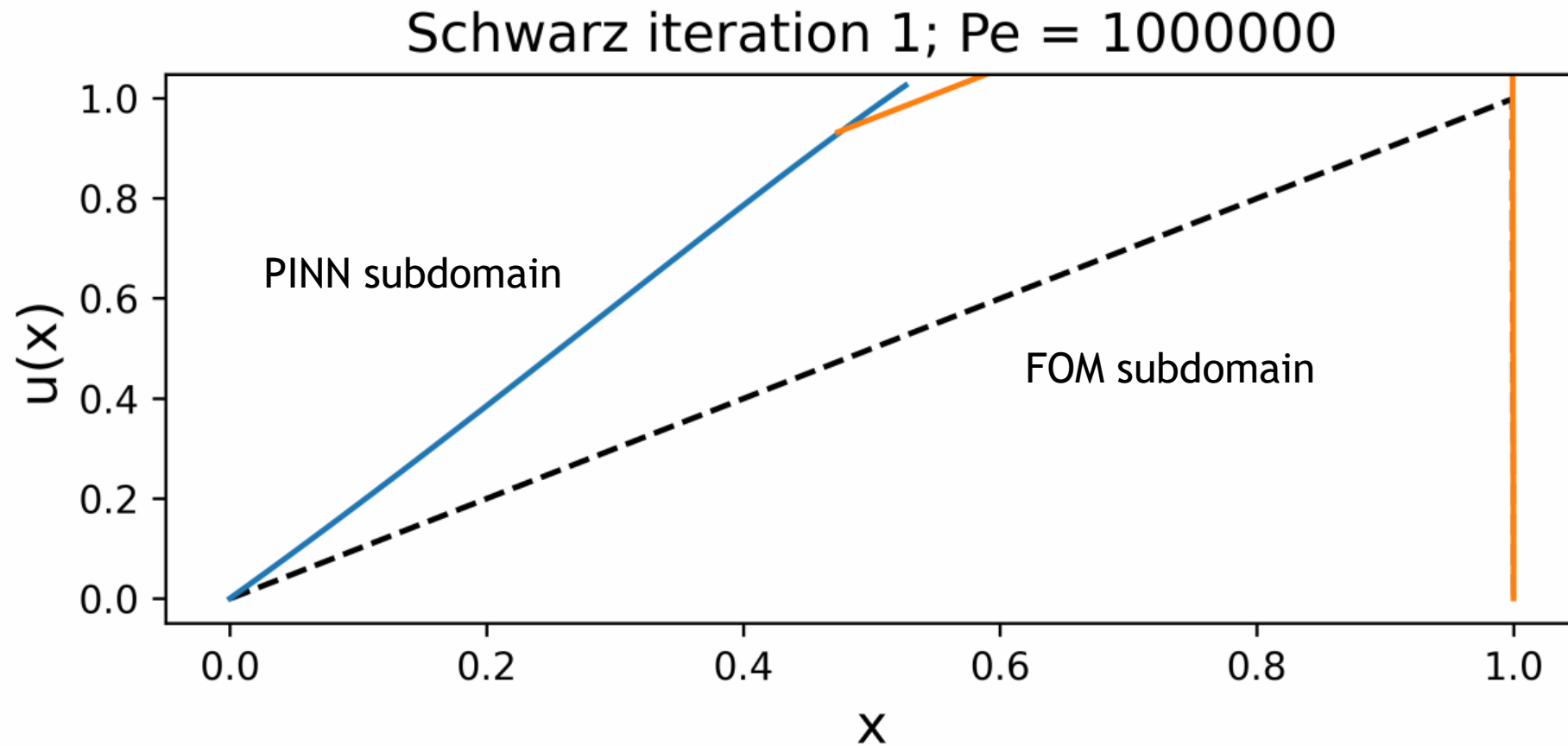
Bonus: PINN-PINN coupling



- 2 Ω , no snapshots, WDBC (unconverged)
- ▼ 2 Ω , no snapshots, SDBC
- △ 2 Ω , snapshots, WDBC (unconverged)
- ◀ 2 Ω , snapshots, SDBC
- 3 Ω , no snapshots, WDBC (unconverged)
- ▼ 3 Ω , no snapshots, SDBC
- ▲ 3 Ω , snapshots, WDBC
- ◁ 3 Ω , snapshots SDBC (unconverged)
- 4 Ω , no snapshots, WDBC (unconverged)
- ▽ 4 Ω , no snapshots, SDBC (unconverged)
- ▲ 4 Ω , snapshots, WDBC
- ◀ 4 Ω , snapshots SDBC
- 5 Ω , no snapshots, WDBC (unconverged)
- ▽ 5 Ω , no snapshots, SDBC (unconverged)
- ▲ 5 Ω , snapshots, WDBC
- ◀ 5 Ω , snapshots, SDBC

- Using **SDBC**s and **data loss** helps with PINN/NN convergence and accuracy

Bonus: PINN-FOM coupling



- PINN-FOM coupling gives **rapid PINN convergence** for **arbitrarily high Peclet numbers**
- PINN-FOM couplings works with **both WDBC and SDBC** configurations

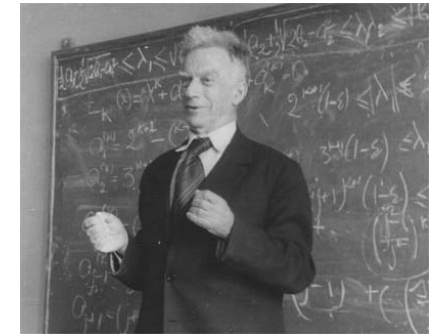
Theoretical Foundation

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

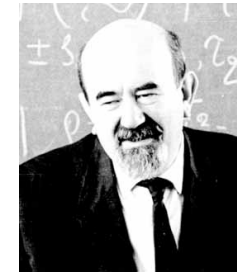
- **S.L. Sobolev (1936)**: posed Schwarz method for **linear elasticity** in variational form and **proved method's convergence** by proposing a convergent sequence of energy functionals.
- **S.G. Mikhlin (1951)**: **proved convergence** of Schwarz method for general linear elliptic PDEs.
- **P.-L. Lions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

$$\Phi[\varphi] = \int_B A(F, Z) dV - \int_B B \cdot \varphi dV$$

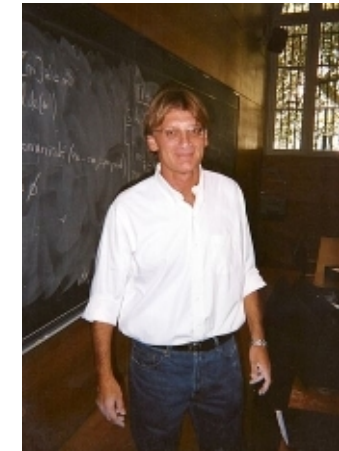
$$\nabla \cdot P + B = 0$$



S.L. Sobolev (1908 – 1989)



S.G. Mikhlin
(1908 – 1990)



P.-L. Lions (1956-)



A. Mota, I. Tezaur, C. Alleman

Convergence Proof*



2 Formulation of the Schwarz Alternating Method

We start by defining the standard three-dimensional variational formulation to establish notation before presenting the formulation of the coupling method.

2.1 Variational Formulation on a Single Domain

Consider a body as the open set $\Omega \subset \mathbb{R}^3$ satisfying a source described by the mapping $\mathbf{u} = \mathbf{u}(\mathbf{X}) : \Omega \rightarrow \mathbb{R}^3$, $\mathbf{X} \in \Omega$. Assume that the boundary of the body is $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ with unit normal \mathbf{N} , where $\partial\Omega_D$ is a displacement boundary, $\partial\Omega_N$ is a traction boundary, and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. The prescribed boundary displacements or Dirichlet boundary conditions are $\mathbf{u}|_{\partial\Omega_D} = \mathbf{g}$. The prescribed boundary tractions or Neumann boundary conditions are $\mathbf{T} \cdot \mathbf{N}|_{\partial\Omega_N} = \mathbf{h}$, $\mathbf{T} = \mathbf{C} \cdot \nabla \mathbf{u}$. Consider the deformation gradient. Let also $\partial\Omega = \Omega \cup \partial\Omega$ be the body locus, with $\partial\Omega$ its cross-section in the reference configuration. Furthermore, introduce the energy functional

$$\Phi[\mathbf{u}] = \int_{\Omega} (F, \mathbf{E}) dV - \int_{\partial\Omega_D} \varphi dV - \int_{\partial\Omega_N} \mathbf{T} \cdot \mathbf{u} dV, \quad (1)$$

in which $\mathbf{A}, \mathbf{F}, \mathbf{E} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the Helmholtz free-energy density and \mathbf{E} is a collection of internal variables. The weak form of the problem is obtained by minimizing the energy functional Φ over the Sobolev space $H^1(\Omega)$ that is comprised of all functions that are square-integrable and have square-integrable first derivatives. Define

$$\mathcal{S} = \{\mathbf{u} \in H^1(\Omega) : \mathbf{u} = \mathbf{g} \text{ on } \partial\Omega_D\} \quad (2)$$

and

$$\mathcal{V} = \{\mathbf{u} \in H^1(\Omega) : \mathbf{E} = 0 \text{ on } \partial\Omega_N\} \quad (3)$$

where $\mathbf{E} \in \mathcal{V}$ is a test function. The potential energy is minimized if and only if $\Phi[\mathbf{u}] \leq \Phi[\mathbf{v}] + \langle \mathbf{h}, \mathbf{v} - \mathbf{u} \rangle$ for all $\mathbf{v} \in \mathcal{V}$ and $\mathbf{u} \in \mathcal{S}$. It is straightforward to show that the minimum of $\Phi[\mathbf{u}]$ in the mapping \mathcal{S} is \mathbf{u} that satisfies

$$(\nabla \Phi[\mathbf{u}])|_{\Omega} = \int_{\Omega} \mathbf{P}^T \cdot \nabla \mathbf{u} dV = \int_{\partial\Omega_D} \mathbf{Q} dV = \int_{\partial\Omega_N} \mathbf{Q} \cdot \mathbf{u} dV = 0, \quad (4)$$

where $\mathbf{P}^T = \mathbf{A}(\mathbf{F}, \mathbf{E})$ denotes the first Piola-Kirchhoff stress. The Euler-Lagrange equation corresponding to the variational statement (1) is

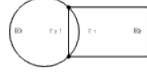


Figure 1: Two subdomains Ω_1 and Ω_2 and their corresponding interfaces Γ_1 and Γ_2 used by the Schwarz alternating method.

Let $\mathbf{u} = \mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^3$ with $\mathbf{x} \in \Omega$, and $\mathbf{x} = \mathbf{x}(\mathbf{X}) : \Omega \rightarrow \mathbb{R}^3$ with $\mathbf{X} \in \Omega$. Introduce the following definition for each subdomain Ω_i :

- Closure: $\bar{\Omega}_i = \Omega_i \cup \partial\Omega_i$
- Discrete boundary: $\partial\Omega_i = \partial\Omega_i \cup \partial\Omega_i$
- Measure boundary: $\partial\Omega_i = \partial\Omega_i \cup \partial\Omega_i$
- Interface boundary: $\Gamma_i = \partial\Omega_i \cap \partial\Omega_j$

Now that with these definitions we generate that $\partial\Omega_i \cap \partial\Omega_j = \emptyset$, $\partial\Omega_i \cap \Gamma_j = \emptyset$ and $\partial\Omega_i \cap \Gamma_j = \emptyset$. Now add to the system

$$\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i \quad (5)$$

and

$$\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i \quad (6)$$

where the operator $P_{\Omega_i}(\cdot)$ denotes the projection from the subdomain Ω_i onto the discrete boundary Γ_i . This projection operator plays a central role in the Schwarz alternating method. In fact, the projection operator is used to project a field \mathbf{u} from one subdomain to the Schwarz boundary of the other subdomains.

The Schwarz alternating method can be seen as a sequence of problems on Ω_1 and Ω_2 . The solution $\mathbf{u}^{(n)}$ for the

$$\begin{aligned} 1. & \mathbf{u}^{(0)} = \mathbf{u}^{(0)}|_{\Omega_1}, \mathbf{u}^{(0)} = \mathbf{u}^{(0)}|_{\Omega_2} && \text{is unknown for } \Omega_1 \\ 2. & \mathbf{u}^{(1)} = \mathbf{u}^{(1)}|_{\Omega_1}, \mathbf{u}^{(1)} = \mathbf{u}^{(1)}|_{\Omega_2} && \text{is unknown for } \Omega_2 \\ 3. & \mathbf{u}^{(2)} = \mathbf{u}^{(2)}|_{\Omega_1}, \mathbf{u}^{(2)} = \mathbf{u}^{(2)}|_{\Omega_2} && \text{is known for } \Omega_1 \\ 4. & \mathbf{u}^{(3)} = \mathbf{u}^{(3)}|_{\Omega_1}, \mathbf{u}^{(3)} = \mathbf{u}^{(3)}|_{\Omega_2} && \text{is known for } \Omega_2 \\ 5. & \mathbf{u}^{(4)} = \mathbf{u}^{(4)}|_{\Omega_1}, \mathbf{u}^{(4)} = \mathbf{u}^{(4)}|_{\Omega_2} && \text{is known for } \Omega_1 \\ 6. & \mathbf{u}^{(5)} = \mathbf{u}^{(5)}|_{\Omega_1}, \mathbf{u}^{(5)} = \mathbf{u}^{(5)}|_{\Omega_2} && \text{is known for } \Omega_2 \\ 7. & \mathbf{u}^{(6)} = \mathbf{u}^{(6)}|_{\Omega_1}, \mathbf{u}^{(6)} = \mathbf{u}^{(6)}|_{\Omega_2} && \text{is known for } \Omega_1 \end{aligned}$$

sequence $\{\mathbf{u}^{(n)}\}$ converges to the minimizer \mathbf{u}^* of $\Phi[\mathbf{u}]$ in \mathcal{S} .

15, 16, 17. Although we do not provide here formal convergence proofs for the existing variants of the Schwarz method, we offer some numerical results illustrating their convergence in Section 3.

Consider the energy functional $\Phi[\mathbf{u}]$ defined in (1). We will denote by \mathbf{u}^* the weak L^2 inner product over Ω that is

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} dV. \quad (7)$$

for $\mathbf{u}, \mathbf{v} \in H^1(\Omega)$, with corresponding norm $\|\cdot\|$. The proof of the convergence of the Schwarz alternating method requires that the functional $\Phi[\mathbf{u}]$ satisfy the following properties over the space \mathcal{S} defined in (2):

1. $\Phi[\mathbf{u}]$ is convex.
2. $\Phi[\mathbf{u}]$ is finite differentiable, with $\Phi'[\mathbf{u}]$ denoting its Fréchet derivative.
3. $\Phi[\mathbf{u}]$ is strictly convex.
4. $\Phi[\mathbf{u}]$ is lower semi-continuous.
5. $\Phi[\mathbf{u}]$ is uniformly continuous on \mathcal{S} , where

$$\mathcal{S}_\delta = \{\mathbf{u} \in \mathcal{S} : \|\mathbf{u} - \mathbf{u}^*\| \leq \delta, \delta > 0, \delta < \infty\}. \quad (8)$$

It can be shown that the energy functional $\Phi[\mathbf{u}]$ defined in (1) satisfies convexity, 2, 3, 4, 5, and 6. The Schwarz alternating method can be seen as a sequence of problems on Ω_1 and Ω_2 . The solution $\mathbf{u}^{(n)}$ for the

$$\text{Remark 1.} \quad \mathbf{u}^{(n)} = \mathbf{u}^{(n)}|_{\Omega_1}, \mathbf{u}^{(n)} = \mathbf{u}^{(n)}|_{\Omega_2} \quad (9)$$

18, 19, 20. Consider the energy functional $\Phi[\mathbf{u}]$ 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 \Phi[\tilde{\varphi}^{(2)}] \geq \Phi[\tilde{\varphi}^{(3)}] \geq \Phi[\tilde{\varphi}^{(4)}] \geq \Phi[\tilde{\varphi}^{(5)}] \geq \Phi[\tilde{\varphi}^{(6)}] \geq \Phi[\tilde{\varphi}^{(7)}] \geq \Phi[\tilde{\varphi}^{(8)}] \geq \Phi[\tilde{\varphi}^{(9)}] \geq \Phi[\tilde{\varphi}^{(10)}] \geq \Phi[\tilde{\varphi}^{(11)}] \geq \Phi[\tilde{\varphi}^{(12)}] \geq \Phi[\tilde{\varphi}^{(13)}] \geq \Phi[\tilde{\varphi}^{(14)}] \geq \Phi[\tilde{\varphi}^{(15)}] \geq \Phi[\tilde{\varphi}^{(16)}] \geq \Phi[\tilde{\varphi}^{(17)}] \geq \Phi[\tilde{\varphi}^{(18)}] \geq \Phi[\tilde{\varphi}^{(19)}] \geq \Phi[\tilde{\varphi}^{(20)}] \geq \Phi[\tilde{\varphi}^{(21)}] \geq \Phi[\tilde{\varphi}^{(22)}] \geq \Phi[\tilde{\varphi}^{(23)}] \geq \Phi[\tilde{\varphi}^{(24)}] \geq \Phi[\tilde{\varphi}^{(25)}] \geq \Phi[\tilde{\varphi}^{(26)}] \geq \Phi[\tilde{\varphi}^{(27)}] \geq \Phi[\tilde{\varphi}^{(28)}] \geq \Phi[\tilde{\varphi}^{(29)}] \geq \Phi[\tilde{\varphi}^{(30)}] \geq \Phi[\tilde{\varphi}^{(31)}] \geq \Phi[\tilde{\varphi}^{(32)}] \geq \Phi[\tilde{\varphi}^{(33)}] \geq \Phi[\tilde{\varphi}^{(34)}] \geq \Phi[\tilde{\varphi}^{(35)}] \geq \Phi[\tilde{\varphi}^{(36)}] \geq \Phi[\tilde{\varphi}^{(37)}] \geq \Phi[\tilde{\varphi}^{(38)}] \geq \Phi[\tilde{\varphi}^{(39)}] \geq \Phi[\tilde{\varphi}^{(40)}] \geq \Phi[\tilde{\varphi}^{(41)}] \geq \Phi[\tilde{\varphi}^{(42)}] \geq \Phi[\tilde{\varphi}^{(43)}] \geq \Phi[\tilde{\varphi}^{(44)}] \geq \Phi[\tilde{\varphi}^{(45)}] \geq \Phi[\tilde{\varphi}^{(46)}] \geq \Phi[\tilde{\varphi}^{(47)}] \geq \Phi[\tilde{\varphi}^{(48)}] \geq \Phi[\tilde{\varphi}^{(49)}] \geq \Phi[\tilde{\varphi}^{(50)}] \geq \Phi[\tilde{\varphi}^{(51)}] \geq \Phi[\tilde{\varphi}^{(52)}] \geq \Phi[\tilde{\varphi}^{(53)}] \geq \Phi[\tilde{\varphi}^{(54)}] \geq \Phi[\tilde{\varphi}^{(55)}] \geq \Phi[\tilde{\varphi}^{(56)}] \geq \Phi[\tilde{\varphi}^{(57)}] \geq \Phi[\tilde{\varphi}^{(58)}] \geq \Phi[\tilde{\varphi}^{(59)}] \geq \Phi[\tilde{\varphi}^{(60)}] \geq \Phi[\tilde{\varphi}^{(61)}] \geq \Phi[\tilde{\varphi}^{(62)}] \geq \Phi[\tilde{\varphi}^{(63)}] \geq \Phi[\tilde{\varphi}^{(64)}] \geq \Phi[\tilde{\varphi}^{(65)}] \geq \Phi[\tilde{\varphi}^{(66)}] \geq \Phi[\tilde{\varphi}^{(67)}] \geq \Phi[\tilde{\varphi}^{(68)}] \geq \Phi[\tilde{\varphi}^{(69)}] \geq \Phi[\tilde{\varphi}^{(70)}] \geq \Phi[\tilde{\varphi}^{(71)}] \geq \Phi[\tilde{\varphi}^{(72)}] \geq \Phi[\tilde{\varphi}^{(73)}] \geq \Phi[\tilde{\varphi}^{(74)}] \geq \Phi[\tilde{\varphi}^{(75)}] \geq \Phi[\tilde{\varphi}^{(76)}] \geq \Phi[\tilde{\varphi}^{(77)}] \geq \Phi[\tilde{\varphi}^{(78)}] \geq \Phi[\tilde{\varphi}^{(79)}] \geq \Phi[\tilde{\varphi}^{(80)}] \geq \Phi[\tilde{\varphi}^{(81)}] \geq \Phi[\tilde{\varphi}^{(82)}] \geq \Phi[\tilde{\varphi}^{(83)}] \geq \Phi[\tilde{\varphi}^{(84)}] \geq \Phi[\tilde{\varphi}^{(85)}] \geq \Phi[\tilde{\varphi}^{(86)}] \geq \Phi[\tilde{\varphi}^{(87)}] \geq \Phi[\tilde{\varphi}^{(88)}] \geq \Phi[\tilde{\varphi}^{(89)}] \geq \Phi[\tilde{\varphi}^{(90)}] \geq \Phi[\tilde{\varphi}^{(91)}] \geq \Phi[\tilde{\varphi}^{(92)}] \geq \Phi[\tilde{\varphi}^{(93)}] \geq \Phi[\tilde{\varphi}^{(94)}] \geq \Phi[\tilde{\varphi}^{(95)}] \geq \Phi[\tilde{\varphi}^{(96)}] \geq \Phi[\tilde{\varphi}^{(97)}] \geq \Phi[\tilde{\varphi}^{(98)}] \geq \Phi[\tilde{\varphi}^{(99)}] \geq \Phi[\tilde{\varphi}^{(100)}] \geq \Phi[\tilde{\varphi}^{(101)}] \geq \Phi[\tilde{\varphi}^{(102)}] \geq \Phi[\tilde{\varphi}^{(103)}] \geq \Phi[\tilde{\varphi}^{(104)}] \geq \Phi[\tilde{\varphi}^{(105)}] \geq \Phi[\tilde{\varphi}^{(106)}] \geq \Phi[\tilde{\varphi}^{(107)}] \geq \Phi[\tilde{\varphi}^{(108)}] \geq \Phi[\tilde{\varphi}^{(109)}] \geq \Phi[\tilde{\varphi}^{(110)}] \geq \Phi[\tilde{\varphi}^{(111)}] \geq \Phi[\tilde{\varphi}^{(112)}] \geq \Phi[\tilde{\varphi}^{(113)}] \geq \Phi[\tilde{\varphi}^{(114)}] \geq \Phi[\tilde{\varphi}^{(115)}] \geq \Phi[\tilde{\varphi}^{(116)}] \geq \Phi[\tilde{\varphi}^{(117)}] \geq \Phi[\tilde{\varphi}^{(118)}] \geq \Phi[\tilde{\varphi}^{(119)}] \geq \Phi[\tilde{\varphi}^{(120)}] \geq \Phi[\tilde{\varphi}^{(121)}] \geq \Phi[\tilde{\varphi}^{(122)}] \geq \Phi[\tilde{\varphi}^{(123)}] \geq \Phi[\tilde{\varphi}^{(124)}] \geq \Phi[\tilde{\varphi}^{(125)}] \geq \Phi[\tilde{\varphi}^{(126)}] \geq \Phi[\tilde{\varphi}^{(127)}] \geq \Phi[\tilde{\varphi}^{(128)}] \geq \Phi[\tilde{\varphi}^{(129)}] \geq \Phi[\tilde{\varphi}^{(130)}] \geq \Phi[\tilde{\varphi}^{(131)}] \geq \Phi[\tilde{\varphi}^{(132)}] \geq \Phi[\tilde{\varphi}^{(133)}] \geq \Phi[\tilde{\varphi}^{(134)}] \geq \Phi[\tilde{\varphi}^{(135)}] \geq \Phi[\tilde{\varphi}^{(136)}] \geq \Phi[\tilde{\varphi}^{(137)}] \geq \Phi[\tilde{\varphi}^{(138)}] \geq \Phi[\tilde{\varphi}^{(139)}] \geq \Phi[\tilde{\varphi}^{(140)}] \geq \Phi[\tilde{\varphi}^{(141)}] \geq \Phi[\tilde{\varphi}^{(142)}] \geq \Phi[\tilde{\varphi}^{(143)}] \geq \Phi[\tilde{\varphi}^{(144)}] \geq \Phi[\tilde{\varphi}^{(145)}] \geq \Phi[\tilde{\varphi}^{(146)}] \geq \Phi[\tilde{\varphi}^{(147)}] \geq \Phi[\tilde{\varphi}^{(148)}] \geq \Phi[\tilde{\varphi}^{(149)}] \geq \Phi[\tilde{\varphi}^{(150)}] \geq \Phi[\tilde{\varphi}^{(151)}] \geq \Phi[\tilde{\varphi}^{(152)}] \geq \Phi[\tilde{\varphi}^{(153)}] \geq \Phi[\tilde{\varphi}^{(154)}] \geq \Phi[\tilde{\varphi}^{(155)}] \geq \Phi[\tilde{\varphi}^{(156)}] \geq \Phi[\tilde{\varphi}^{(157)}] \geq \Phi[\tilde{\varphi}^{(158)}] \geq \Phi[\tilde{\varphi}^{(159)}] \geq \Phi[\tilde{\varphi}^{(160)}] \geq \Phi[\tilde{\varphi}^{(161)}] \geq \Phi[\tilde{\varphi}^{(162)}] \geq \Phi[\tilde{\varphi}^{(163)}] \geq \Phi[\tilde{\varphi}^{(164)}] \geq \Phi[\tilde{\varphi}^{(165)}] \geq \Phi[\tilde{\varphi}^{(166)}] \geq \Phi[\tilde{\varphi}^{(167)}] \geq \Phi[\tilde{\varphi}^{(168)}] \geq \Phi[\tilde{\varphi}^{(169)}] \geq \Phi[\tilde{\varphi}^{(170)}] \geq \Phi[\tilde{\varphi}^{(171)}] \geq \Phi[\tilde{\varphi}^{(172)}] \geq \Phi[\tilde{\varphi}^{(173)}] \geq \Phi[\tilde{\varphi}^{(174)}] \geq \Phi[\tilde{\varphi}^{(175)}] \geq \Phi[\tilde{\varphi}^{(176)}] \geq \Phi[\tilde{\varphi}^{(177)}] \geq \Phi[\tilde{\varphi}^{(178)}] \geq \Phi[\tilde{\varphi}^{(179)}] \geq \Phi[\tilde{\varphi}^{(180)}] \geq \Phi[\tilde{\varphi}^{(181)}] \geq \Phi[\tilde{\varphi}^{(182)}] \geq \Phi[\tilde{\varphi}^{(183)}] \geq \Phi[\tilde{\varphi}^{(184)}] \geq \Phi[\tilde{\varphi}^{(185)}] \geq \Phi[\tilde{\varphi}^{(186)}] \geq \Phi[\tilde{\varphi}^{(187)}] \geq \Phi[\tilde{\varphi}^{(188)}] \geq \Phi[\tilde{\varphi}^{(189)}] \geq \Phi[\tilde{\varphi}^{(190)}] \geq \Phi[\tilde{\varphi}^{(191)}] \geq \Phi[\tilde{\varphi}^{(192)}] \geq \Phi[\tilde{\varphi}^{(193)}] \geq \Phi[\tilde{\varphi}^{(194)}] \geq \Phi[\tilde{\varphi}^{(195)}] \geq \Phi[\tilde{\varphi}^{(196)}] \geq \Phi[\tilde{\varphi}^{(197)}] \geq \Phi[\tilde{\varphi}^{(198)}] \geq \Phi[\tilde{\varphi}^{(199)}] \geq \Phi[\tilde{\varphi}^{(200)}] \geq \Phi[\tilde{\varphi}^{(201)}] \geq \Phi[\tilde{\varphi}^{(202)}] \geq \Phi[\tilde{\varphi}^{(203)}] \geq \Phi[\tilde{\varphi}^{(204)}] \geq \Phi[\tilde{\varphi}^{(205)}] \geq \Phi[\tilde{\varphi}^{(206)}] \geq \Phi[\tilde{\varphi}^{(207)}] \geq \Phi[\tilde{\varphi}^{(208)}] \geq \Phi[\tilde{\varphi}^{(209)}] \geq \Phi[\tilde{\varphi}^{(210)}] \geq \Phi[\tilde{\varphi}^{(211)}] \geq \Phi[\tilde{\varphi}^{(212)}] \geq \Phi[\tilde{\varphi}^{(213)}] \geq \Phi[\tilde{\varphi}^{(214)}] \geq \Phi[\tilde{\varphi}^{(215)}] \geq \Phi[\tilde{\varphi}^{(216)}] \geq \Phi[\tilde{\varphi}^{(217)}] \geq \Phi[\tilde{\varphi}^{(218)}] \geq \Phi[\tilde{\varphi}^{(219)}] \geq \Phi[\tilde{\varphi}^{(220)}] \geq \Phi[\tilde{\varphi}^{(221)}] \geq \Phi[\tilde{\varphi}^{(222)}] \geq \Phi[\tilde{\varphi}^{(223)}] \geq \Phi[\tilde{\varphi}^{(224)}] \geq \Phi[\tilde{\varphi}^{(225)}] \geq \Phi[\tilde{\varphi}^{(226)}] \geq \Phi[\tilde{\varphi}^{(227)}] \geq \Phi[\tilde{\varphi}^{(228)}] \geq \Phi[\tilde{\varphi}^{(229)}] \geq \Phi[\tilde{\varphi}^{(230)}] \geq \Phi[\tilde{\varphi}^{(231)}] \geq \Phi[\tilde{\varphi}^{(232)}] \geq \Phi[\tilde{\varphi}^{(233)}] \geq \Phi[\tilde{\varphi}^{(234)}] \geq \Phi[\tilde{\varphi}^{(235)}] \geq \Phi[\tilde{\varphi}^{(236)}] \geq \Phi[\tilde{\varphi}^{(237)}] \geq \Phi[\tilde{\varphi}^{(238)}] \geq \Phi[\tilde{\varphi}^{(239)}] \geq \Phi[\tilde{\varphi}^{(240)}] \geq \Phi[\tilde{\varphi}^{(241)}] \geq \Phi[\tilde{\varphi}^{(242)}] \geq \Phi[\tilde{\varphi}^{(243)}] \geq \Phi[\tilde{\varphi}^{(244)}] \geq \Phi[\tilde{\varphi}^{(245)}] \geq \Phi[\tilde{\varphi}^{(246)}] \geq \Phi[\tilde{\varphi}^{(247)}] \geq \Phi[\tilde{\varphi}^{(248)}] \geq \Phi[\tilde{\varphi}^{(249)}] \geq \Phi[\tilde{\varphi}^{(250)}] \geq \Phi[\tilde{\varphi}^{(251)}] \geq \Phi[\tilde{\varphi}^{(252)}] \geq \Phi[\tilde{\varphi}^{(253)}] \geq \Phi[\tilde{\varphi}^{(254)}] \geq \Phi[\tilde{\varphi}^{(255)}] \geq \Phi[\tilde{\varphi}^{(256)}] \geq \Phi[\tilde{\varphi}^{(257)}] \geq \Phi[\tilde{\varphi}^{(258)}] \geq \Phi[\tilde{\varphi}^{(259)}] \geq \Phi[\tilde{\varphi}^{(260)}] \geq \Phi[\tilde{\varphi}^{(261)}] \geq \Phi[\tilde{\varphi}^{(262)}] \geq \Phi[\tilde{\varphi}^{(263)}] \geq \Phi[\tilde{\varphi}^{(264)}] \geq \Phi[\tilde{\varphi}^{(265)}] \geq \Phi[\tilde{\varphi}^{(266)}] \geq \Phi[\tilde{\varphi}^{(267)}] \geq \Phi[\tilde{\varphi}^{(268)}] \geq \Phi[\tilde{\varphi}^{(269)}] \geq \Phi[\tilde{\varphi}^{(270)}] \geq \Phi[\tilde{\varphi}^{(271)}] \geq \Phi[\tilde{\varphi}^{(272)}] \geq \Phi[\tilde{\varphi}^{(273)}] \geq \Phi[\tilde{\varphi}^{(274)}] \geq \Phi[\tilde{\varphi}^{(275)}] \geq \Phi[\tilde{\varphi}^{(276)}] \geq \Phi[\tilde{\varphi}^{(277)}] \geq \Phi[\tilde{\varphi}^{(278)}] \geq \Phi[\tilde{\varphi}^{(279)}] \geq \Phi[\tilde{\varphi}^{(280)}] \geq \Phi[\tilde{\varphi}^{(281)}] \geq \Phi[\tilde{\varphi}^{(282)}] \geq \Phi[\tilde{\varphi}^{(283)}] \geq \Phi[\tilde{\varphi}^{(284)}] \geq \Phi[\tilde{\varphi}^{(285)}] \geq \Phi[\tilde{\varphi}^{(286)}] \geq \Phi[\tilde{\varphi}^{(287)}] \geq \Phi[\tilde{\varphi}^{(288)}] \geq \Phi[\tilde{\varphi}^{(289)}] \geq \Phi[\tilde{\varphi}^{(290)}] \geq \Phi[\tilde{\varphi}^{(291)}] \geq \Phi[\tilde{\varphi}^{(292)}] \geq \Phi[\tilde{\varphi}^{(293)}] \geq \Phi[\tilde{\varphi}^{(294)}] \geq \Phi[\tilde{\varphi}^{(295)}] \geq \Phi[\tilde{\varphi}^{(296)}] \geq \Phi[\tilde{\varphi}^{(297)}] \geq \Phi[\tilde{\varphi}^{(298)}] \geq \Phi[\tilde{\varphi}^{(299)}] \geq \Phi[\tilde{\varphi}^{(300)}] \geq \Phi[\tilde{\varphi}^{(301)}] \geq \Phi[\tilde{\varphi}^{(302)}] \geq \Phi[\tilde{\varphi}^{(303)}] \geq \Phi[\tilde{\varphi}^{(304)}] \geq \Phi[\tilde{\varphi}^{(305)}] \geq \Phi[\tilde{\varphi}^{(306)}] \geq \Phi[\tilde{\varphi}^{(307)}] \geq \Phi[\tilde{\varphi}^{(308)}] \geq \Phi[\tilde{\varphi}^{(309)}] \geq \Phi[\tilde{\varphi}^{(310)}] \geq \Phi[\tilde{\varphi}^{(311)}] \geq \Phi[\tilde{\varphi}^{(312)}] \geq \Phi[\tilde{\varphi}^{(313)}] \geq \Phi[\tilde{\varphi}^{(314)}] \geq \Phi[\tilde{\varphi}^{(315)}] \geq \Phi[\tilde{\varphi}^{(316)}] \geq \Phi[\tilde{\varphi}^{(317)}] \geq \Phi[\tilde{\varphi}^{(318)}] \geq \Phi[\tilde{\varphi}^{(319)}] \geq \Phi[\tilde{\varphi}^{(320)}] \geq \Phi[\tilde{\varphi}^{(321)}] \geq \Phi[\tilde{\varphi}^{(322)}] \geq \Phi[\tilde{\varphi}^{(323)}] \geq \Phi[\tilde{\varphi}^{(324)}] \geq \Phi[\tilde{\varphi}^{(325)}] \geq \Phi[\tilde{\varphi}^{(326)}] \geq \Phi[\tilde{\varphi}^{(327)}] \geq \Phi[\tilde{\varphi}^{(328)}] \geq \Phi[\tilde{\varphi}^{(329)}] \geq \Phi[\tilde{\varphi}^{(330)}] \geq \Phi[\tilde{\varphi}^{(331)}] \geq \Phi[\tilde{\varphi}^{(332)}] \geq \Phi[\tilde{\varphi}^{(333)}] \geq \Phi[\tilde{\varphi}^{(334)}] \geq \Phi[\tilde{\varphi}^{(335)}] \geq \Phi[\tilde{\varphi}^{(336)}] \geq \Phi[\tilde{\varphi}^{(337)}] \geq \Phi[\tilde{\varphi}^{(338)}] \geq \Phi[\tilde{\varphi}^{(339)}] \geq \Phi[\tilde{\varphi}^{(340)}] \geq \Phi[\tilde{\varphi}^{(341)}] \geq \Phi[\tilde{\varphi}^{(342)}] \geq \Phi[\tilde{\varphi}^{(343)}] \geq \Phi[\tilde{\varphi}^{(344)}] \geq \Phi[\tilde{\varphi}^{(345)}] \geq \Phi[\tilde{\varphi}^{(346)}] \geq \Phi[\tilde{\varphi}^{(347)}] \geq \Phi[\tilde{\varphi}^{(348)}] \geq \Phi[\tilde{\varphi}^{(349)}] \geq \Phi[\tilde{\varphi}^{(350)}] \geq \Phi[\tilde{\varphi}^{(351)}] \geq \Phi[\tilde{\varphi}^{(352)}] \geq \Phi[\tilde{\varphi}^{(353)}] \geq \Phi[\tilde{\varphi}^{(354)}] \geq \Phi[\tilde{\varphi}^{(355)}] \geq \Phi[\tilde{\varphi}^{(356)}] \geq \Phi[\tilde{\varphi}^{(357)}] \geq \Phi[\tilde{\varphi}^{(358)}] \geq \Phi[\tilde{\varphi}^{(359)}] \geq \Phi[\tilde{\varphi}^{(360)}] \geq \Phi[\tilde{\varphi}^{(361)}] \geq \Phi[\tilde{\varphi}^{(362)}] \geq \Phi[\tilde{\varphi}^{(363)}] \geq \Phi[\tilde{\varphi}^{(364)}] \geq \Phi[\tilde{\varphi}^{(365)}] \geq \Phi[\tilde{\varphi}^{(366)}] \geq \Phi[\tilde{\varphi}^{(367)}] \geq \Phi[\tilde{\varphi}^{(368)}] \geq \Phi[\tilde{\varphi}^{(369)}] \geq \Phi[\tilde{\varphi}^{(370)}] \geq \Phi[\tilde{\varphi}^{(371)}] \geq \Phi[\tilde{\varphi}^{(372)}] \geq \Phi[\tilde{\varphi}^{(373)}] \geq \Phi[\tilde{\varphi}^{(374)}] \geq \Phi[\tilde{\varphi}^{(375)}] \geq \Phi[\tilde{\varphi}^{(376)}] \geq \Phi[\tilde{\varphi}^{(377)}] \geq \Phi[\tilde{\varphi}^{(378)}] \geq \Phi[\tilde{\varphi}^{(379)}] \geq \Phi[\tilde{\varphi}^{(380)}] \geq \Phi[\tilde{\varphi}^{(381)}] \geq \Phi[\tilde{\varphi}^{(382)}] \geq \Phi[\tilde{\varphi}^{(383)}] \geq \Phi[\tilde{\varphi}^{(384)}] \geq \Phi[\tilde{\varphi}^{(385)}] \geq \Phi[\tilde{\varphi}^{(386)}] \geq \Phi[\tilde{\varphi}^{(387)}] \geq \Phi[\tilde{\varphi}^{(388)}] \geq \Phi[\tilde{\varphi}^{(389)}] \geq \Phi[\tilde{\varphi}^{(390)}] \geq \Phi[\tilde{\varphi}^{(391)}] \geq \Phi[\tilde{\varphi}^{(392)}] \geq \Phi[\tilde{\varphi}^{(393)}] \geq \Phi[\tilde{\varphi}^{(394)}] \geq \Phi[\tilde{\varphi}^{(395)}] \geq \Phi[\tilde{\varphi}^{(396)}] \geq \Phi[\tilde{\varphi}^{(397)}] \geq \Phi[\tilde{\varphi}^{(398)}] \geq \Phi[\tilde{\varphi}^{(399)}] \geq \Phi[\tilde{\varphi}^{(400)}] \geq \Phi[\tilde{\varphi}^{(401)}] \geq \Phi[\tilde{\varphi}^{(402)}] \geq \Phi[\tilde{\varphi}^{(403)}] \geq \Phi[\tilde{\varphi}^{(404)}] \geq \Phi[\tilde{\varphi}^{(405)}] \geq \Phi[\tilde{\varphi}^{(406)}] \geq \Phi[\tilde{\var$



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

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

- $\delta^2 S[\boldsymbol{\varphi}_h]$ 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.

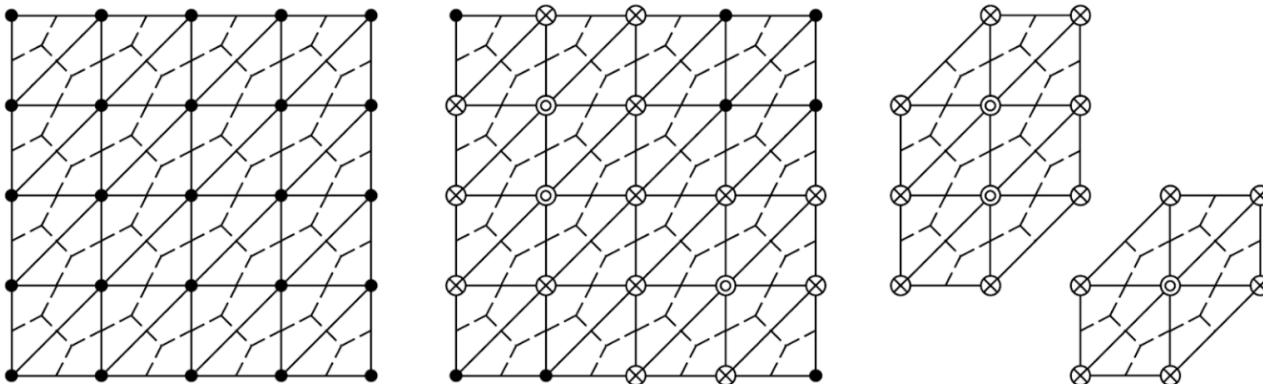
Energy-Conserving Sampling and Weighting (ECSW)



- **Project-then-approximate** paradigm (as opposed to approximate-then-project)

$$\begin{aligned} r_k(q_k, t) &= W^T r(\tilde{u}, t) \\ &= \sum_{e \in \mathcal{E}} W^T L_e^T r_e(L_{e+} \tilde{u}, t) \end{aligned}$$

- $L_e \in \{0,1\}^{d_e \times N}$ where d_e is the **number of degrees of freedom** associated with each mesh element (this is in the context of meshes used in first-order hyperbolic problems where there are N_e mesh elements)
- $L_{e+} \in \{0,1\}^{d_e \times N}$ selects degrees of freedom necessary for **flux reconstruction**
- Equality can be **relaxed**



Augmented reduced mesh: \odot represents a selected node attached to a selected element; and \otimes represents an added node to enable the full representation of the computational stencil at the selected node/element

ECSW: Generating the Reduced Mesh and Weights



- Using a subset of the same snapshots $u_i, i \in 1, \dots, n_h$ used to generate the **state basis** V , we can train the reduced mesh
- Snapshots are first **projected** onto their associated basis and then **reconstructed**

$$c_{se} = W^T L_e^T r_e \left(L_e + \left(u_{ref} + V V^T (u_s - u_{ref}) \right), t \right) \in \mathbb{R}^n$$

$$d_s = r_k(\tilde{u}, t) \in \mathbb{R}^n, \quad s = 1, \dots, n_h$$

- We can then form the **system**

$$\mathbf{C} = \begin{pmatrix} c_{11} & \dots & c_{1N_e} \\ \vdots & \ddots & \vdots \\ c_{n_h 1} & \dots & c_{n_h N_e} \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} d_1 \\ \vdots \\ d_{n_h} \end{pmatrix}$$

- Where $\mathbf{C}\xi = \mathbf{d}, \xi \in \mathbb{R}^{N_e}, \xi = \mathbf{1}$ must be the solution
- Further relax the equality to yield **non-negative least-squares problem**:

$$\xi = \arg \min_{x \in \mathbb{R}^n} \|\mathbf{C}x - \mathbf{d}\|_2 \text{ subject to } x \geq 0$$

- Solve the above optimization problem using a **non-negative least squares solver** with an **early termination condition** to **promote sparsity** of the vector ξ