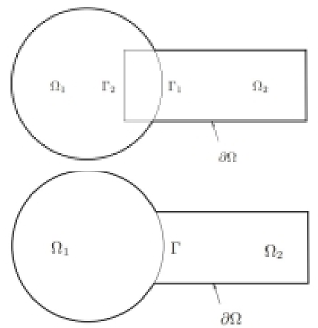
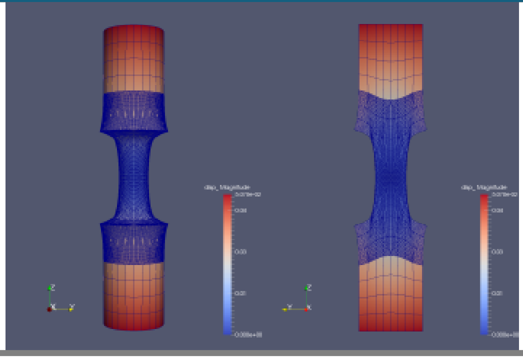
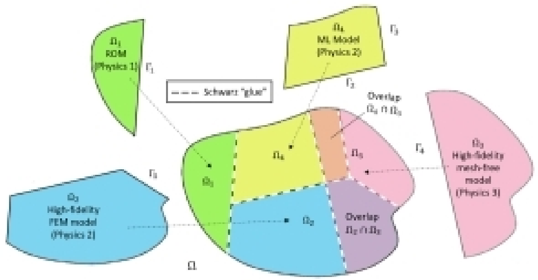
SAND2024-08453C

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







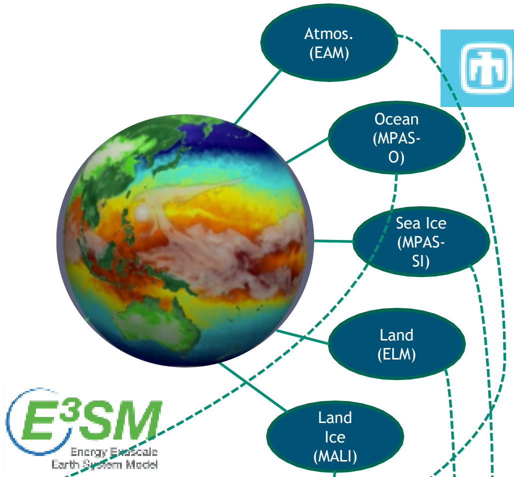
Irina Tezaur[1], Chris Wentland[1], Francesco Rizzi[2], Joshua Barnett[3], Alejandro Mota[1]

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

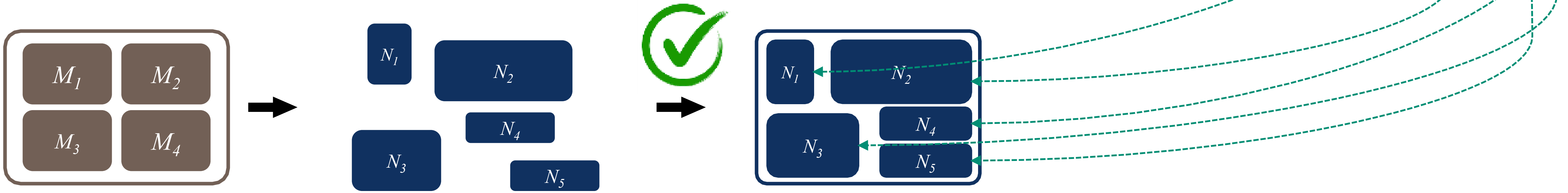
2nd AMS-UNI International Joint Meeting
Palermo, Italy. July 23-26, 2024

SAND2024-XXXX

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, …

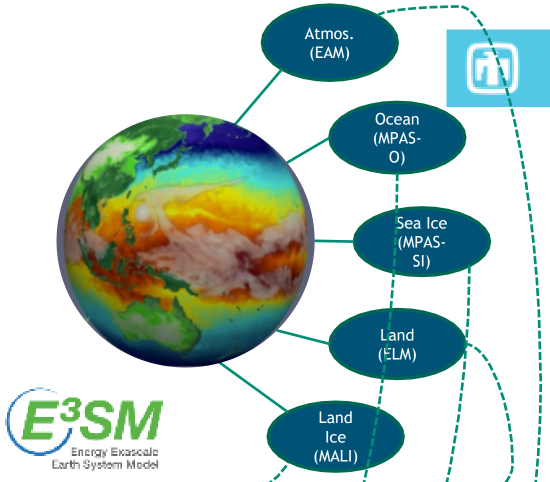
Traditional Methods

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

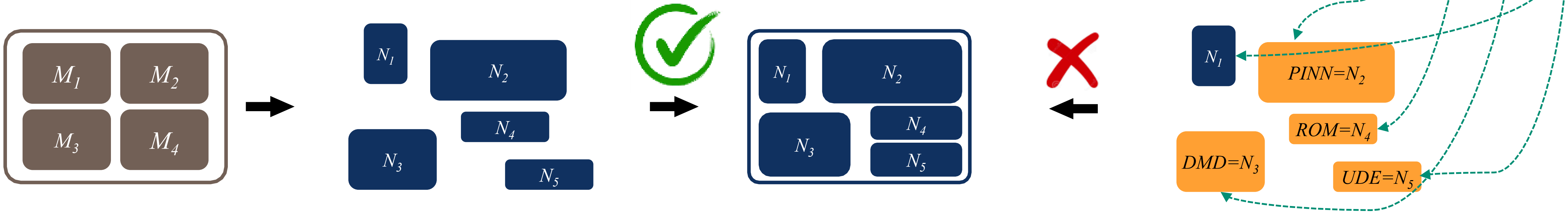
Coupled Numerical Model

- 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**!

Flexible Heterogeneous Numerical Methods (fHNM) Project

Principal research objective:

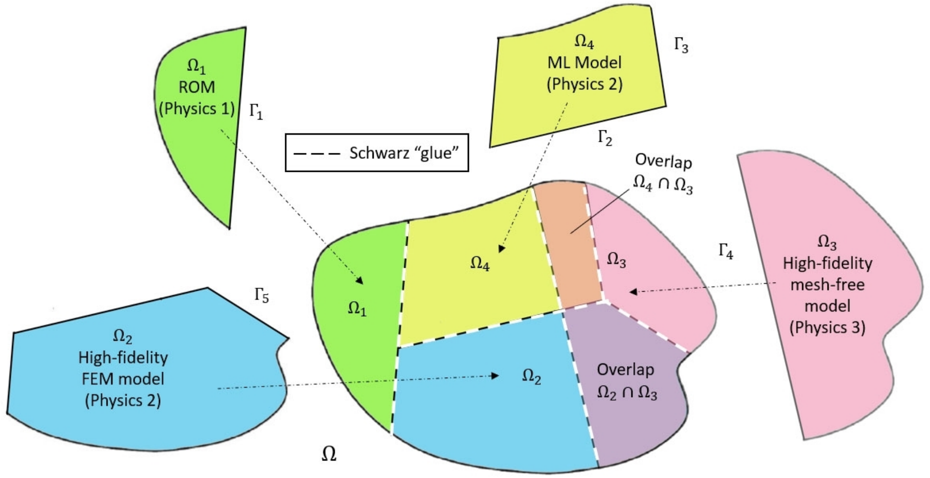
- **Discover mathematical principles** guiding the assembly of **standard** and **data-driven** numerical models in stable, accurate and physically consistent ways.

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.

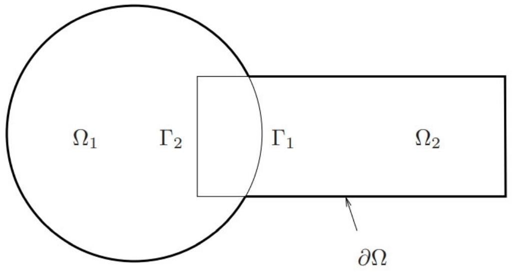
Three coupling methods:

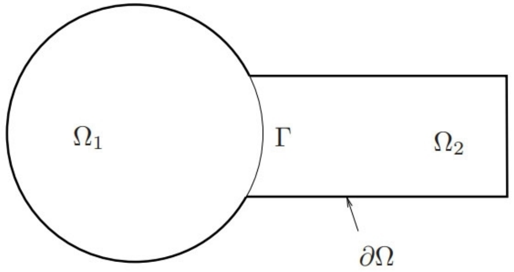
- Alternating Schwarz-based coupling ***This talk.***
- Optimization-based coupling
- Coupling via generalized mortar methods

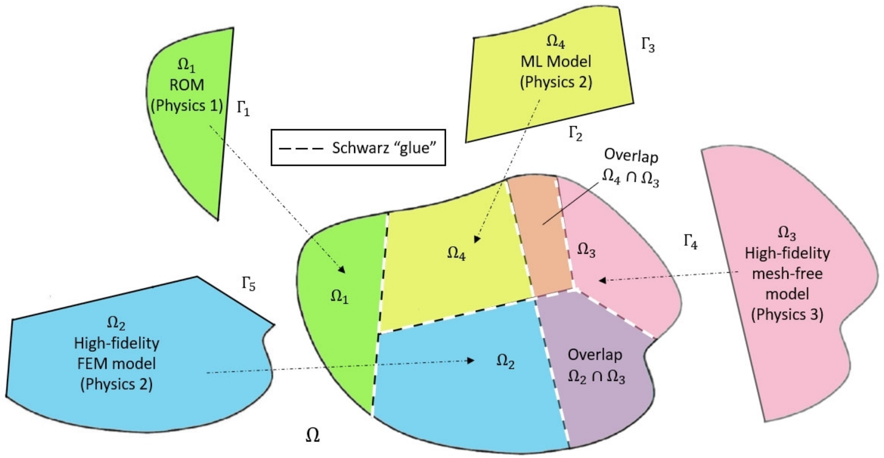


Outline

- 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







*Full-Order Model. #Reduced Order Model.

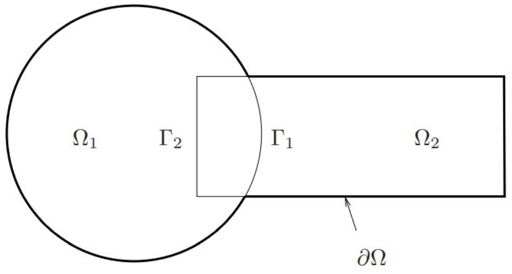
Outline

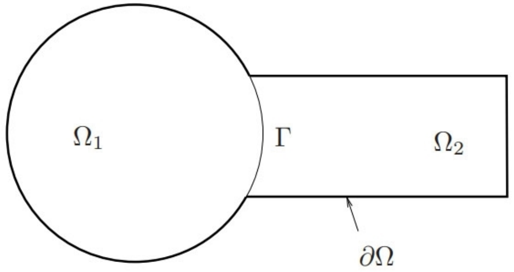
- **The Schwarz Alternating Method for Domain Decomposition-Based Coupling**

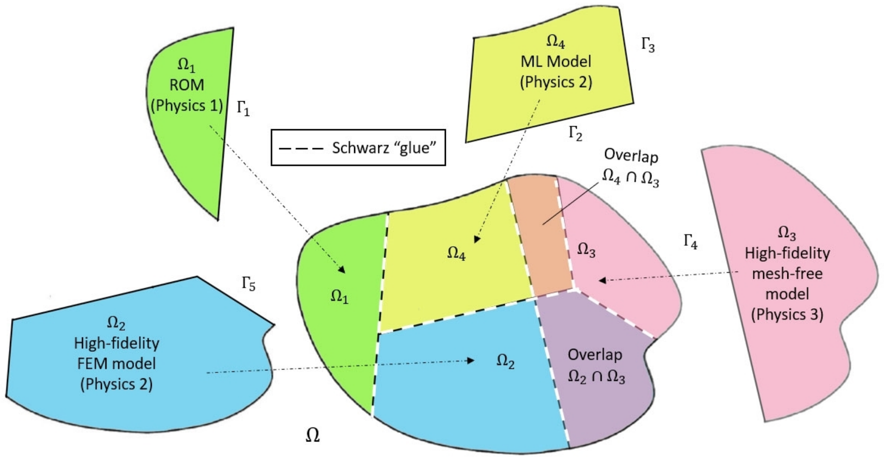
- Extension to FOM*-ROM# and ROM-ROM Coupling

- Numerical Examples
  - ➢ 2D Burgers Equation
  - ➢ 2D Shallow Water Equations
  - ➢ Teaser: 2D Euler Equations Riemann Problem

- Summary & Future Work

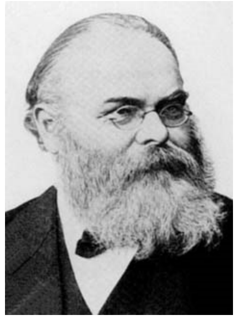






*Full-Order Model. #Reduced Order Model.

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

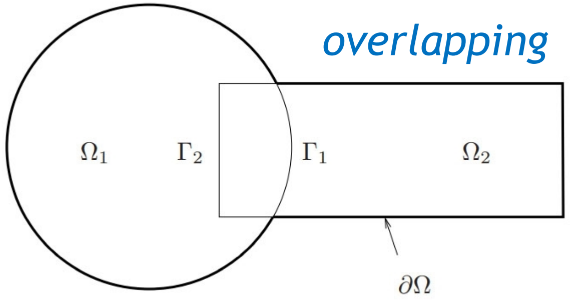
Initialize:

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

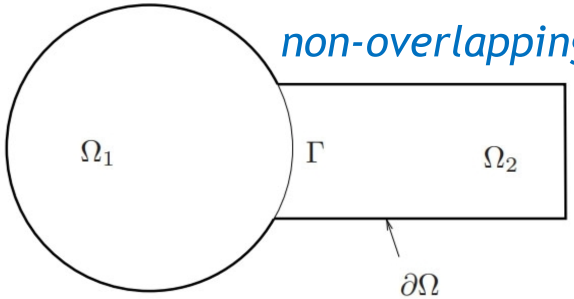
Iterate until convergence:

- 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.

overlapping



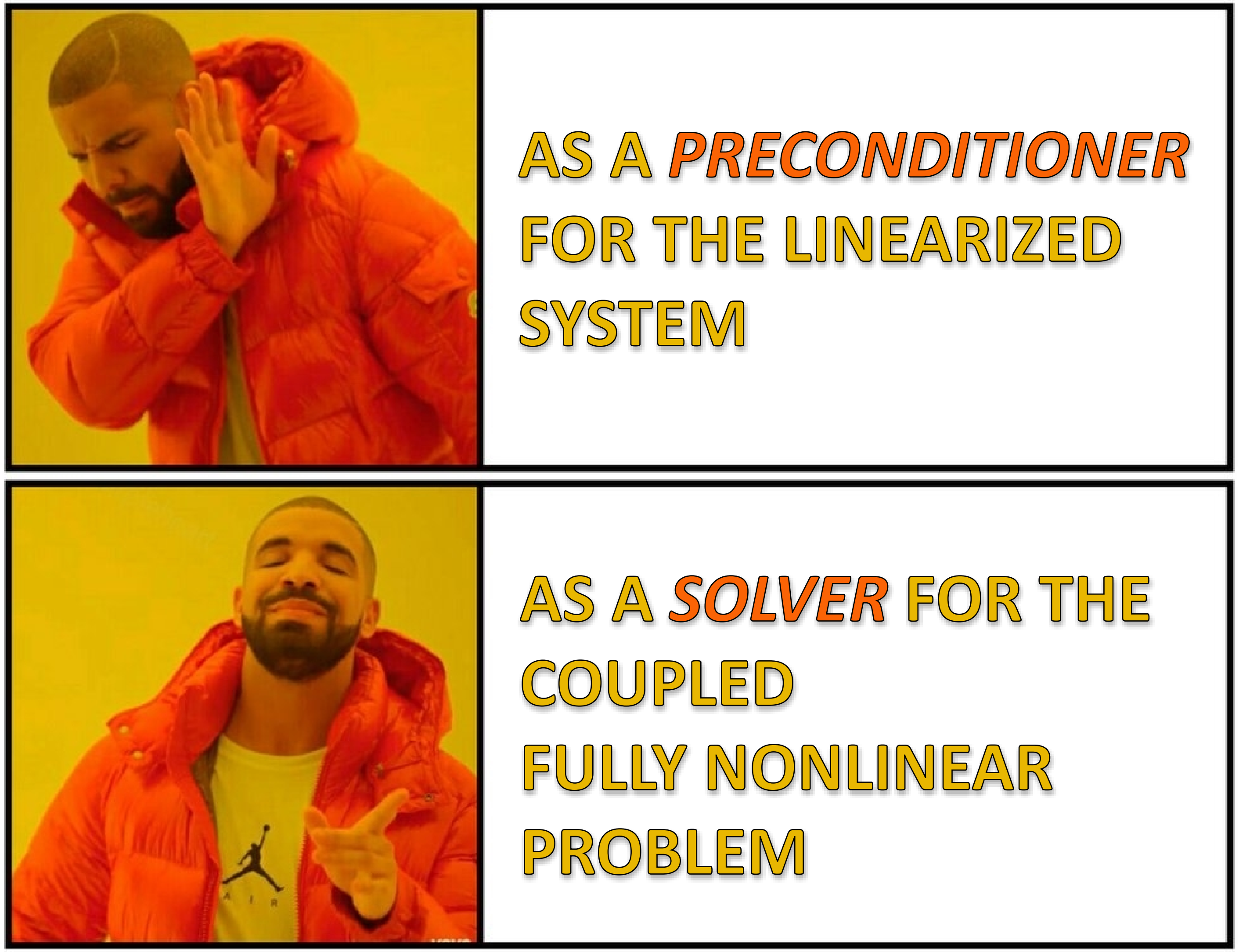
non-overlapping



- 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).

How We Use the Schwarz Alternating Method

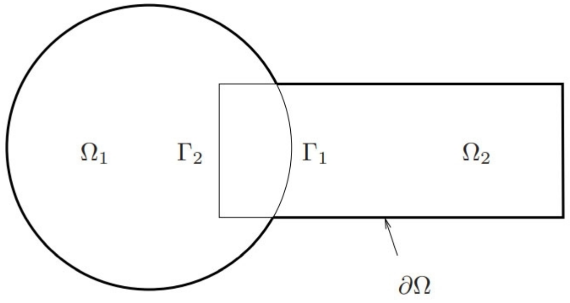


Spatial Coupling via (Multiplicative) Alternating Schwarz

Overlapping Domain Decomposition

$$\begin{cases} N\left(\boldsymbol{u}_1^{(n+1)}\right) = f \text{ , in } \Omega_1 \\ \boldsymbol{u}_1^{(n+1)} = \boldsymbol{g}, \text{ on } \partial\Omega_1 \backslash \Gamma_1 \\ \boldsymbol{u}_1^{(n+1)} = \boldsymbol{u}_2^{(n)} \quad \text{on } \Gamma_1 \end{cases}$$

$$\begin{cases} N\left(\boldsymbol{u}_2^{(n+1)}\right) = f \text{ , in } \Omega_2 \\ \boldsymbol{u}_2^{(n+1)} = \boldsymbol{g}, \text{ on } \partial\Omega_2 \backslash \Gamma_2 \\ \boldsymbol{u}_2^{(n+1)} = \boldsymbol{u}_1^{(n+1)} \quad \text{on } \Gamma_2 \end{cases}$$

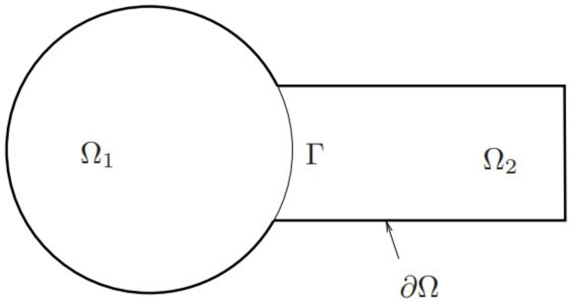


Model PDE:

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

Non-overlapping Domain Decomposition

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



- 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)

Additional Parallelism via Additive Schwarz

Multiplicative Overlapping Schwarz

$$\begin{cases} N\left(\boldsymbol{u}_1^{(n+1)}\right) = f \text{ , in } \Omega_1 \\ \boldsymbol{u}_1^{(n+1)} = \boldsymbol{g}, \text{ on } \partial\Omega_1 \backslash \Gamma_1 \\ \boldsymbol{u}_1^{(n+1)} = \boldsymbol{u}_2^{(n)} \quad \text{ on } \Gamma_1 \end{cases}$$

$$\begin{cases} N\left(\boldsymbol{u}_2^{(n+1)}\right) = f \text{ , in } \Omega_2 \\ \boldsymbol{u}_2^{(n+1)} = \boldsymbol{g}, \text{ on } \partial\Omega_2 \backslash \Gamma_2 \\ \boldsymbol{u}_2^{(n+1)} = \boldsymbol{u}_1^{(n+1)} \quad \text{ on } \Gamma_2 \end{cases}$$

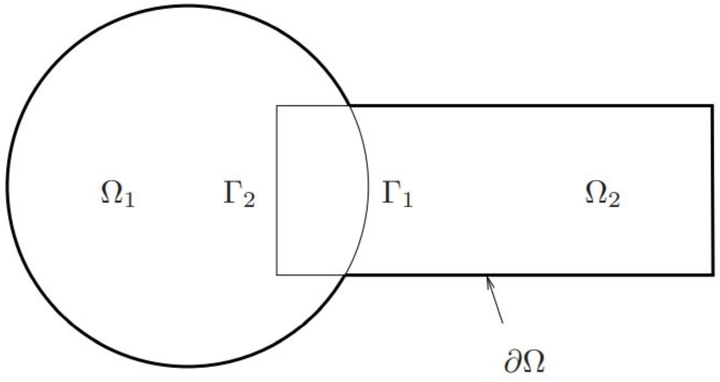
Additive Overlapping Schwarz

$$\begin{cases} N\left(\boldsymbol{u}_1^{(n+1)}\right) = f \text{ , in } \Omega_1 \\ \boldsymbol{u}_1^{(n+1)} = \boldsymbol{g}, \text{ on } \partial\Omega_1 \backslash \Gamma_1 \\ \boldsymbol{u}_1^{(n+1)} = \boldsymbol{u}_2^{(n)} \quad \text{ on } \Gamma_1 \end{cases}$$

$$\begin{cases} N\left(\boldsymbol{u}_2^{(n+1)}\right) = f \text{ , in } \Omega_2 \\ \boldsymbol{u}_2^{(n+1)} = \boldsymbol{g}, \text{ on } \partial\Omega_2 \backslash \Gamma_2 \\ \boldsymbol{u}_2^{(n+1)} = \boldsymbol{u}_1^{(n)} \quad \text{ on } \Gamma_2 \end{cases}$$

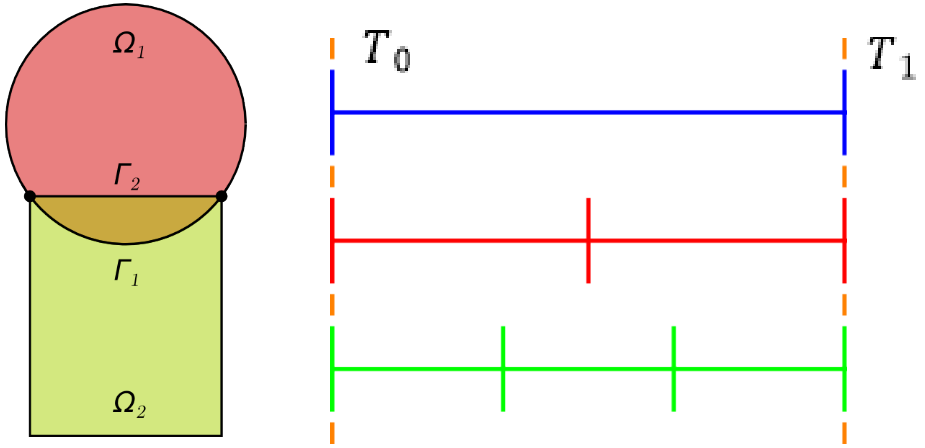
Model PDE:

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



- **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

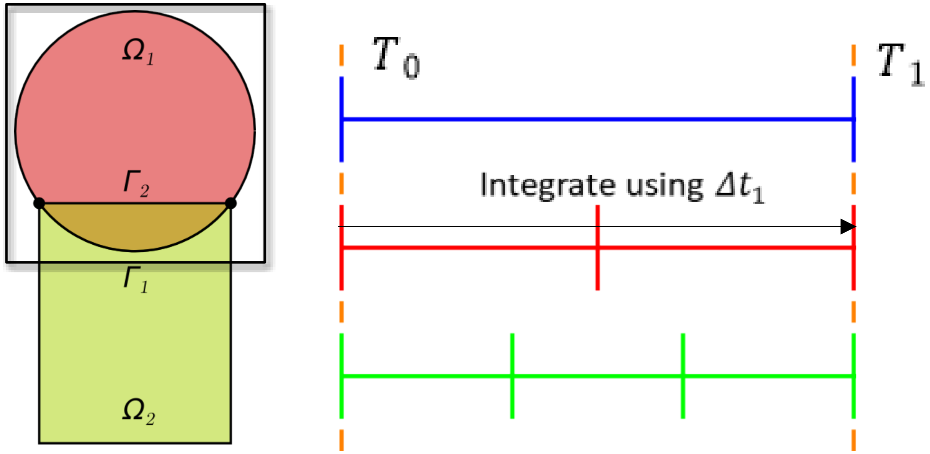
Time integrator for Ω_2

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

Model PDE:

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

Time-Advancement Within the Schwarz Framework



Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

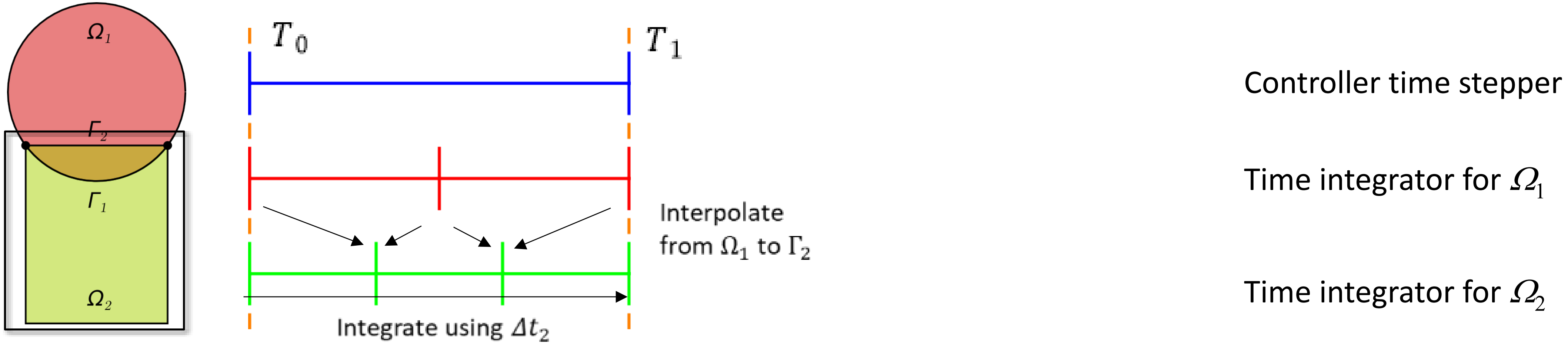
Step 0: Initialize $i = 0$ (controller time index).

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 $T_i + n\Delta t_1$.

Model PDE:

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

Time-Advancement Within the Schwarz Framework



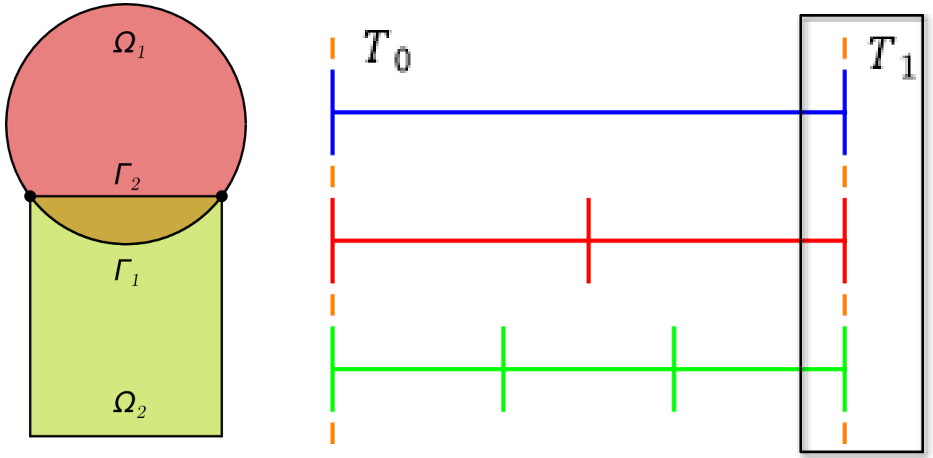
Step 0: Initialize $i = 0$ (controller time index).

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

Time-Advancement Within the Schwarz Framework



Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

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

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

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

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

Model PDE:

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

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 $T_i + n\Delta t_2$.

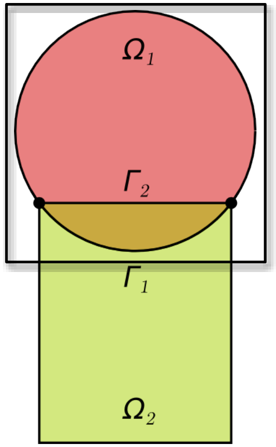
Step 3: Check for convergence at time T_{i+1}.

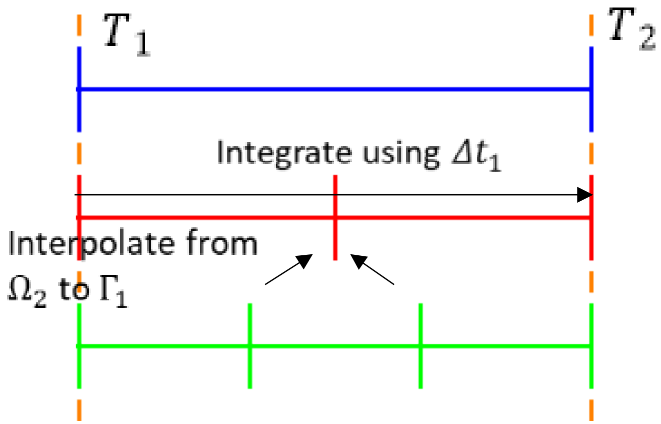
- If unconverged, return to Step 1.

Model PDE:

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

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 $T_i + n\Delta t_2$.

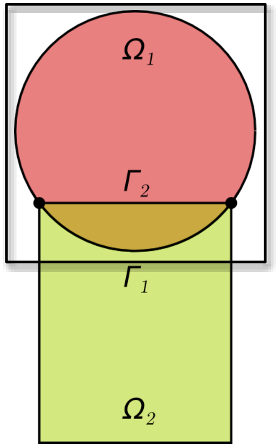
Step 3: Check for convergence at time T_{i+1}.

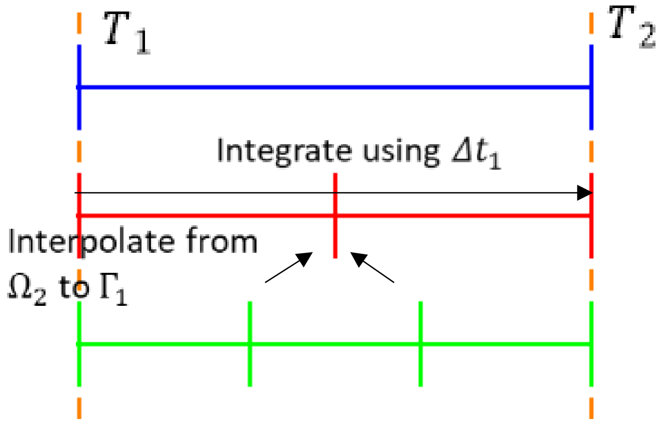
- If unconverged, return to Step 1.
- If converged, set $i = i + 1$ and return to Step 1.

Model PDE:

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

Time-Advancement Within the Schwarz Framework





Controller time stepper

Time integrator for Ω_1

Time integrator for Ω_2

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:

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

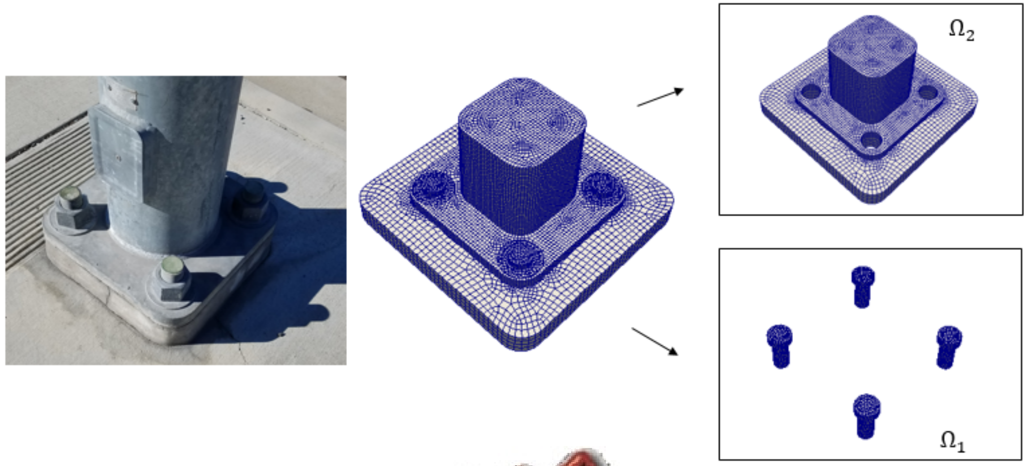
Schwarz for Multiscale FOM-FOM Coupling in Solid Mechanics[1]

Model Solid Mechanics PDEs:

Quasistatic: $\mathrm{Div}\,\boldsymbol{P} + \rho_0 \boldsymbol{B} = \boldsymbol{0} \quad \text{in} \quad \Omega$

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

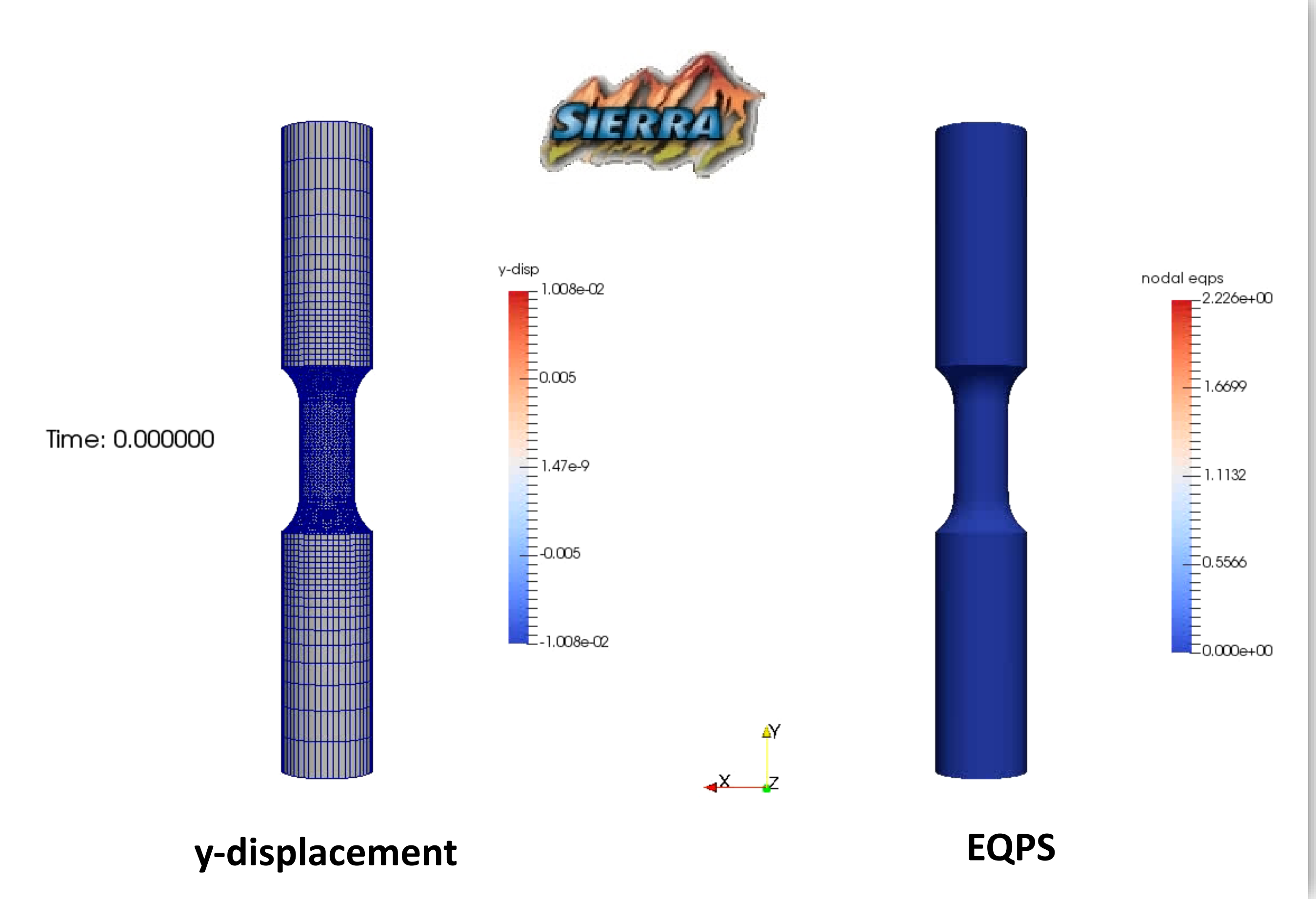
- Coupling is ***concurrent*** (two-way).
- ***Ease of implementation*** into existing massively-parallel HPC codes.
- ***Scalable***, ***fast***, ***robust*** (we target ***real*** engineering problems, e.g., analyses involving failure of bolted components!).
- Coupling does not introduce ***nonphysical 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.



Albany-LCM [2]   SIERRA

[1] Mota *et al*. 2017; Mota *et al*. 2022. [2] https://github.com/sandialabs/LCM.

Schwarz for Multiscale FOM-FOM Coupling in Solid Mechanics[1]



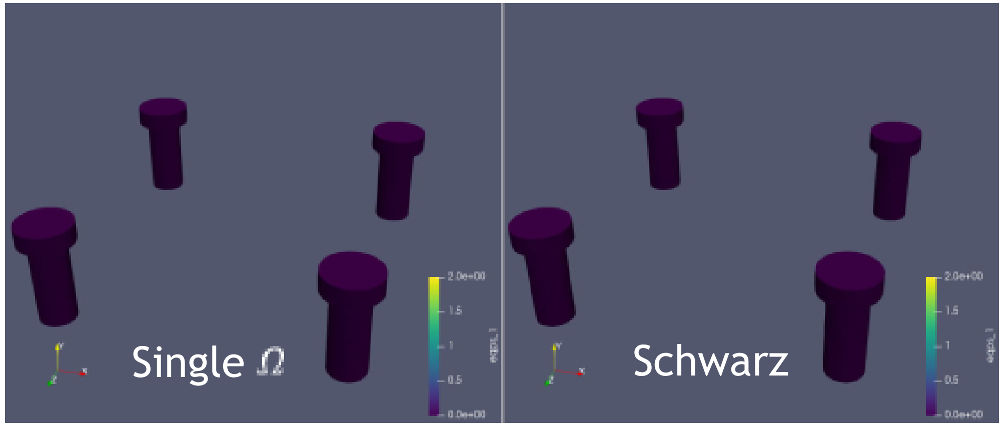


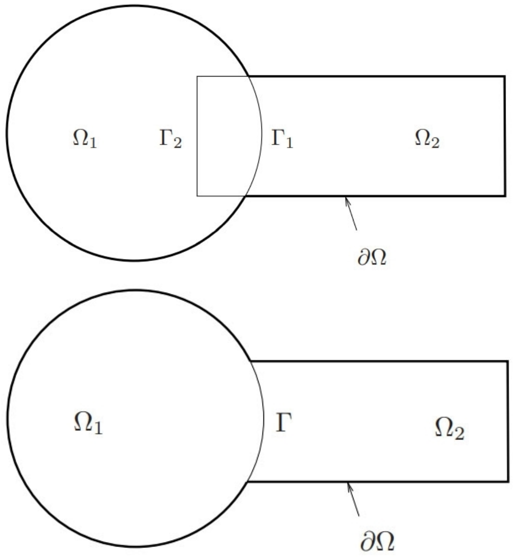
Figure above: tension specimen simulation coupling composite TET10 elements with HEX elements in Sierra/SM.

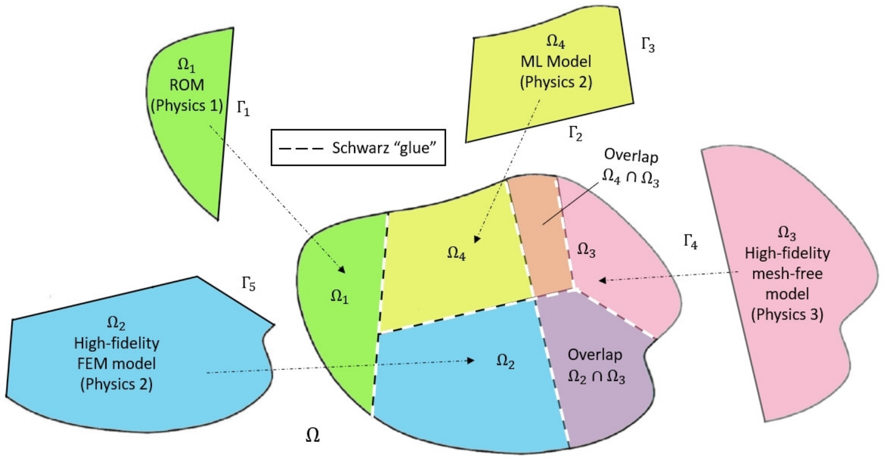
Figures right: bolted joint simulation coupling composite TET10 elements with HEX elements in Sierra/SM.

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

Outline

- 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





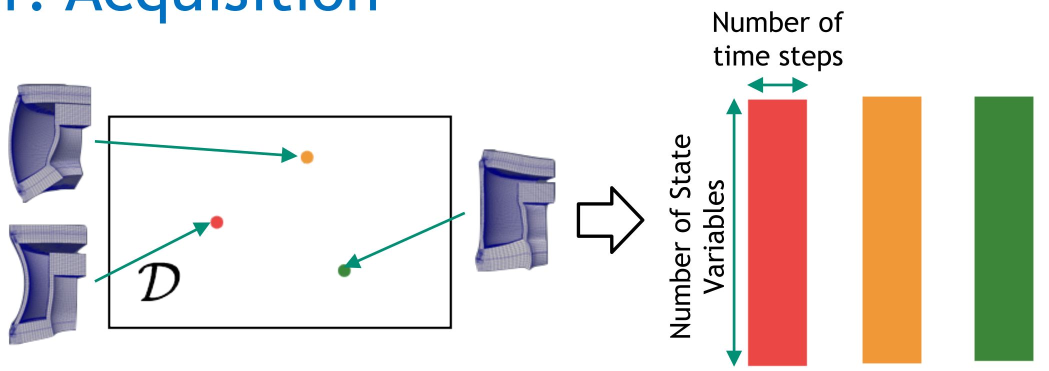
*Full-Order Model. #Reduced Order Model.

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

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

* Least-Squares Petrov-Galerkin

1. Acquisition

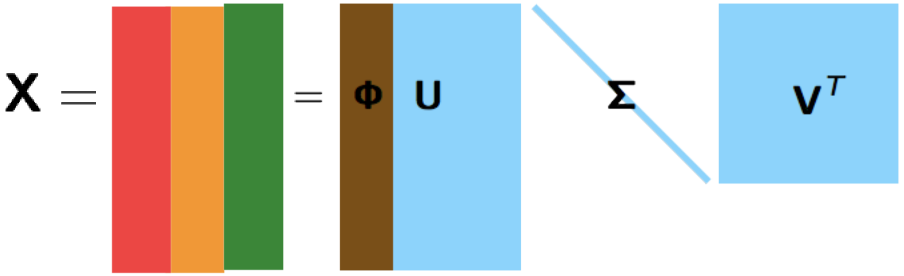


Solve ODE at different design points

Save solution data

2. Learning

Proper Orthogonal Decomposition (POD):



3. Projection-Based Reduction

Choose ODE temporal discretization

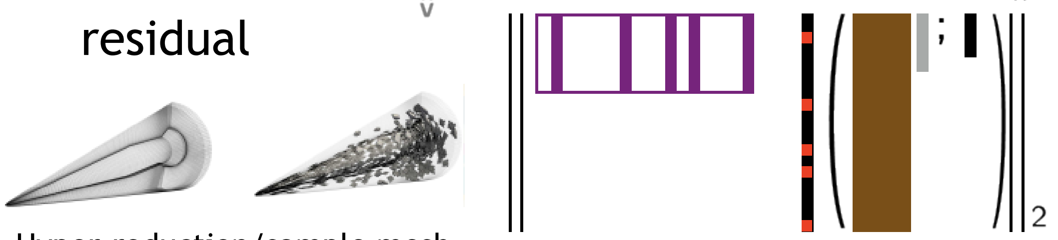
$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}; t, \boldsymbol{\mu})$$
$$\Downarrow$$
$$\boldsymbol{r}^n(\boldsymbol{u}^n; \boldsymbol{\mu}) = 0, \; n = 1, \dots, T$$

Reduce the number of unknowns

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

Minimize residual

$$\underset{\hat{\boldsymbol{v}}}{\text{minimize}} \left\| \boldsymbol{A} \, \boldsymbol{r}^n(\boldsymbol{\Phi}\hat{\boldsymbol{v}}; \boldsymbol{\mu}) \right\|_2$$



Hyper-reduction/sample mesh

ROM = projection-based Reduced Order Model

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

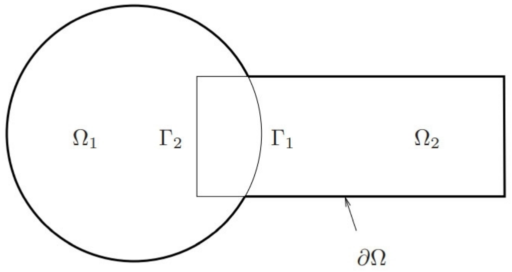
- **Strong** vs. **weak** BC enforcement?
  - ➢ Strong BC enforcement difficult for some models (e.g., cell-centered finite volume, PINNs)
- **Optimizing parameters** in Schwarz BCs for non-overlapping Schwarz?

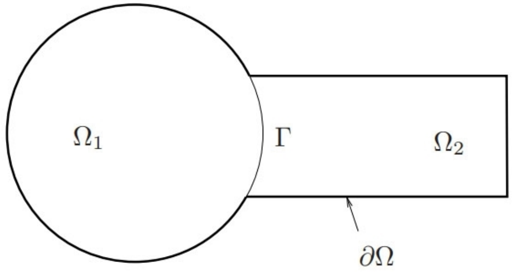
Choice of hyper-reduction

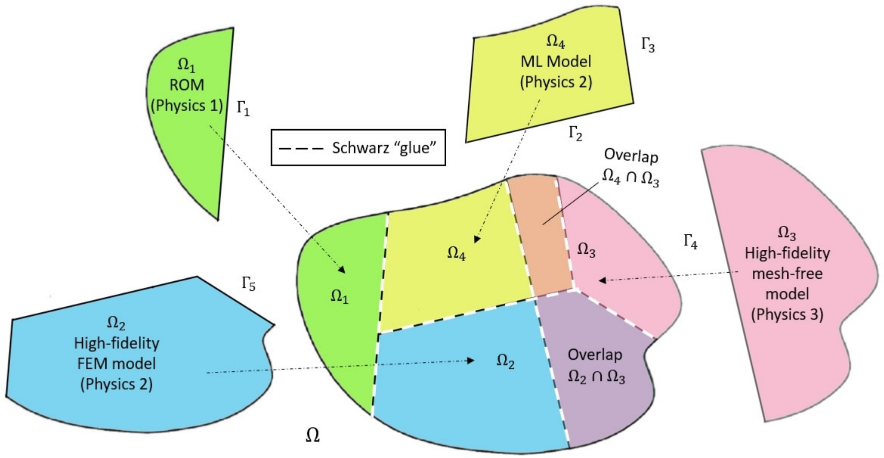
- What **hyper-reduction** method to use?
  - ➢ Application may require particular method (e.g., ECSW for solid mechanics problems)
- How to **sample Schwarz boundaries** in applying hyper-reduction?
  - ➢ Need to have enough sample mesh points at Schwarz boundaries to apply Schwarz

Outline

- 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







*Full-Order Model. #Reduced Order Model.

2D Inviscid Burgers Equation

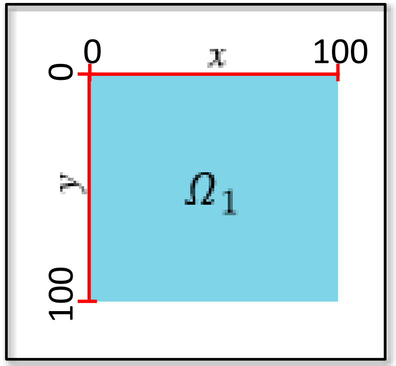
Popular analog for fluid problems where **shocks** are possible, and particularly **difficult** for conventional projection-based ROMs

$$\frac{\partial u}{\partial t} + \frac{1}{2}\left(\frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y}\right) = 0.02\exp\left(\mu_2 x\right)$$

$$\frac{\partial v}{\partial t} + \frac{1}{2}\left(\frac{\partial (vu)}{\partial x} + \frac{\partial (v^2)}{\partial y}\right) = 0$$

$$u(0, y, t; \boldsymbol{\mu}) = \mu_1$$

$$u(x, y, 0) = v(x, y, 0) = 1$$



Problem setup:

- $\Omega = (0,100)^2$, $t \in [0, 25]$
- Two **parameters** $\boldsymbol{\mu} = (\mu_1, \mu_2)$. **Training:** uniform sampling of $= [4.25, 5.50]\times[0.015, 0.03]$ 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 $\Delta t = 0.05$

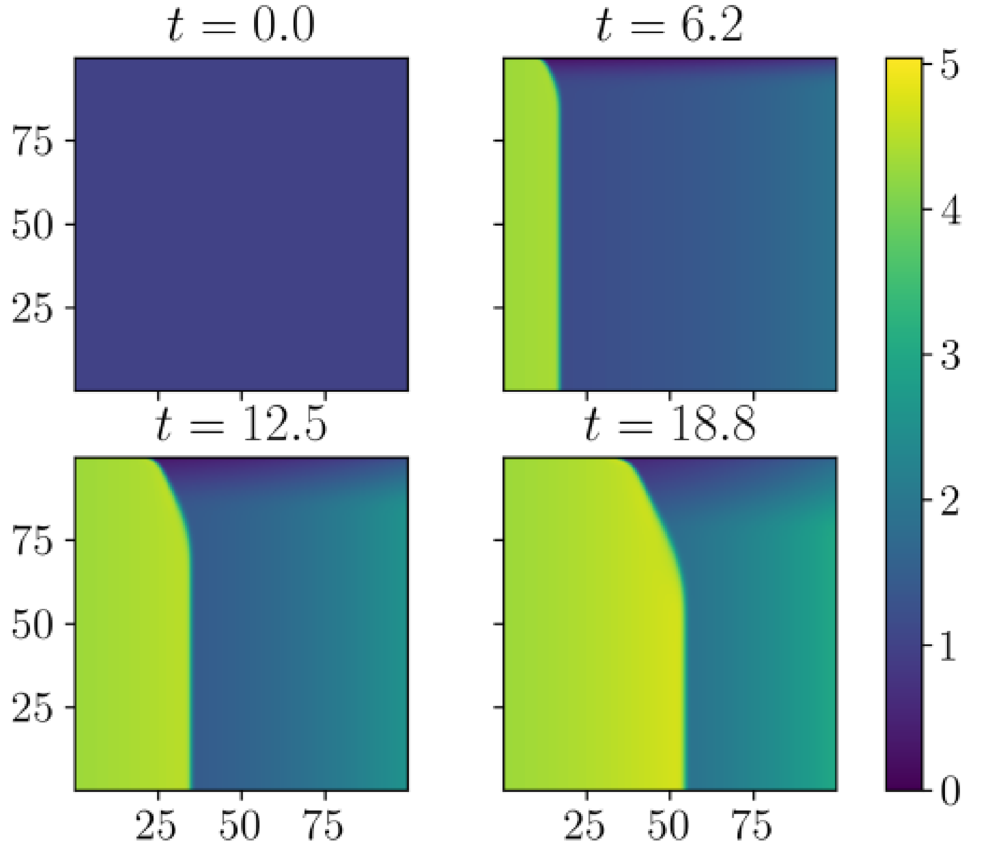
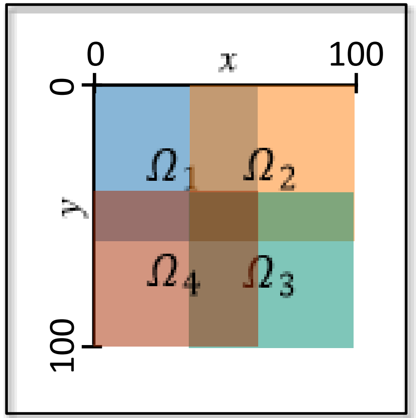
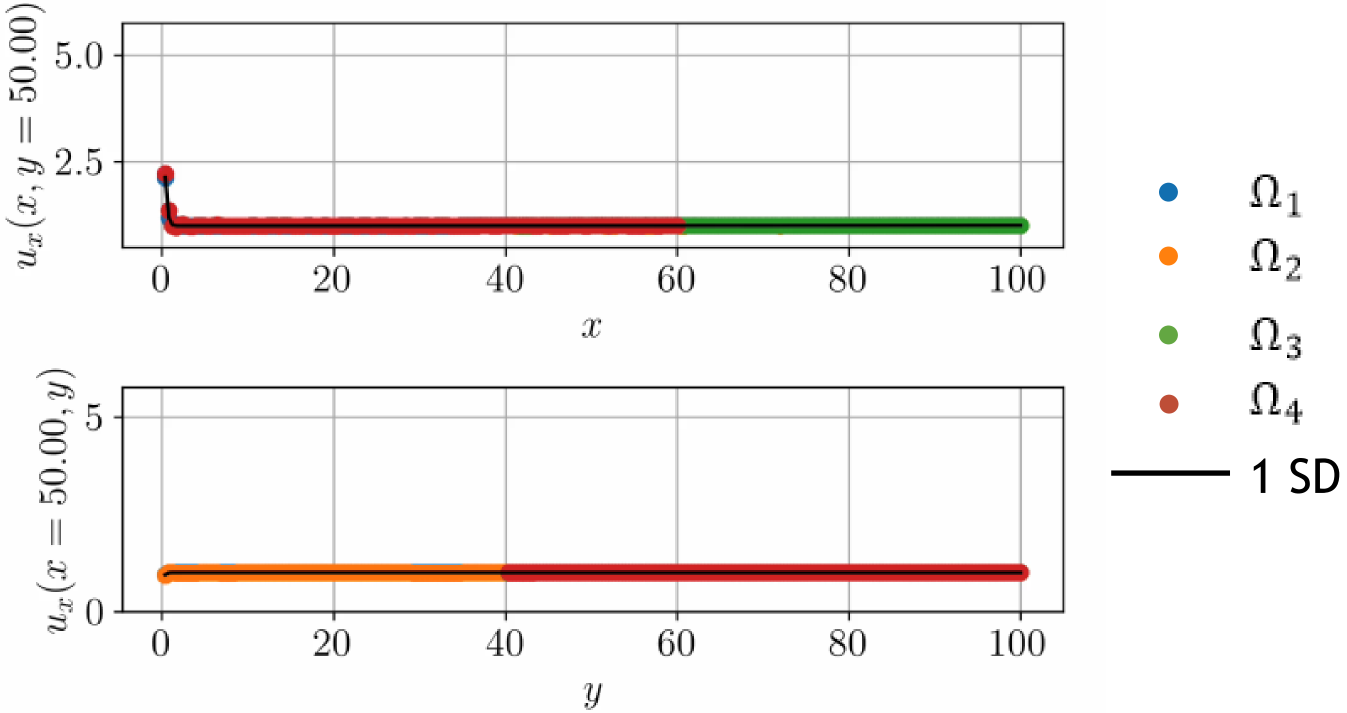


Figure above: solution of u component at various times

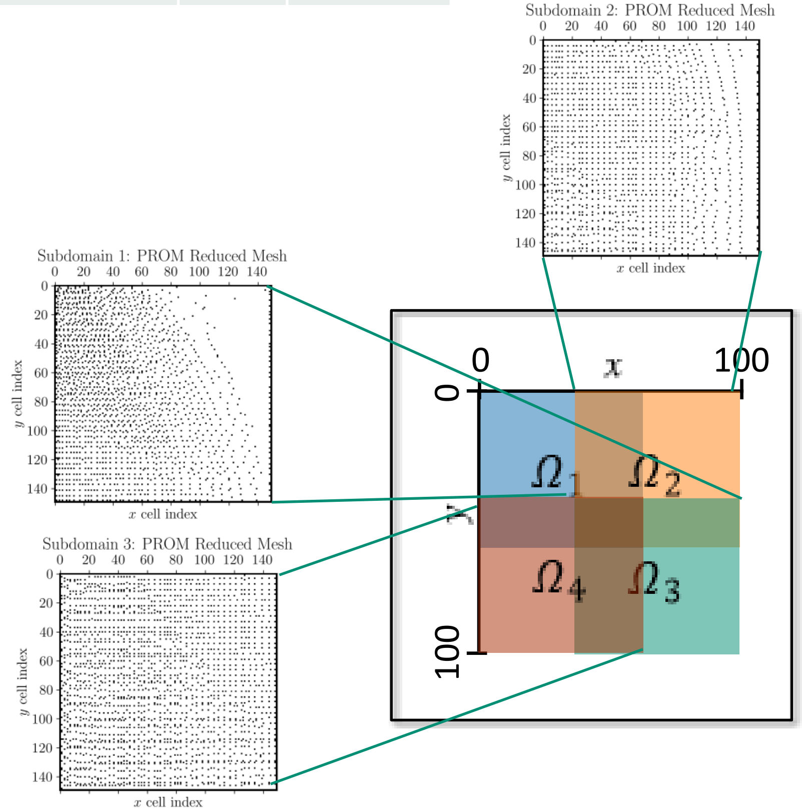
Schwarz Coupling Details



FOM-HROM-HROM-HROM Coupling



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**



- **FOM** in Ω_1 as this is **"hardest" subdomain** for ROM
- **HROMs** in Ω_2, Ω_3, Ω_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× speedup** achieved over all-FOM coupling

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

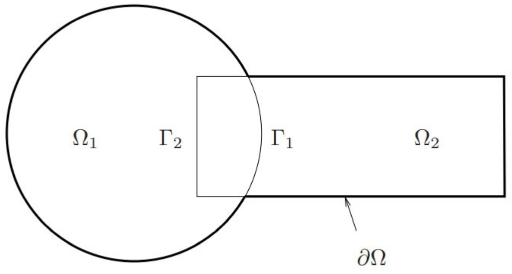
Outline

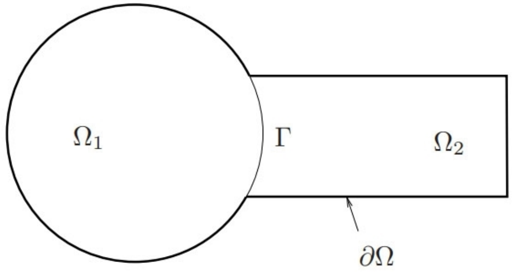
- The Schwarz Alternating Method for Domain Decomposition-Based Coupling

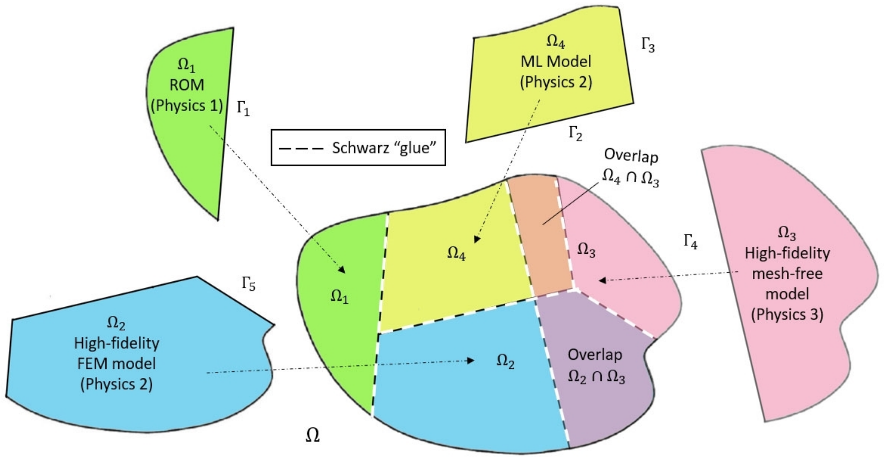
- Extension to FOM*-ROM# and ROM-ROM Coupling

- **Numerical Examples**
  - ➢ 2D Burgers Equation
  - ➢ **2D Shallow Water Equations**
  - ➢ Teaser: 2D Euler Equations Riemann Problem

- Summary & Future Work







*Full-Order Model. #Reduced Order Model.

2D Shallow Water Equations (SWE)

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

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) = -\mu v$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = \mu u$$

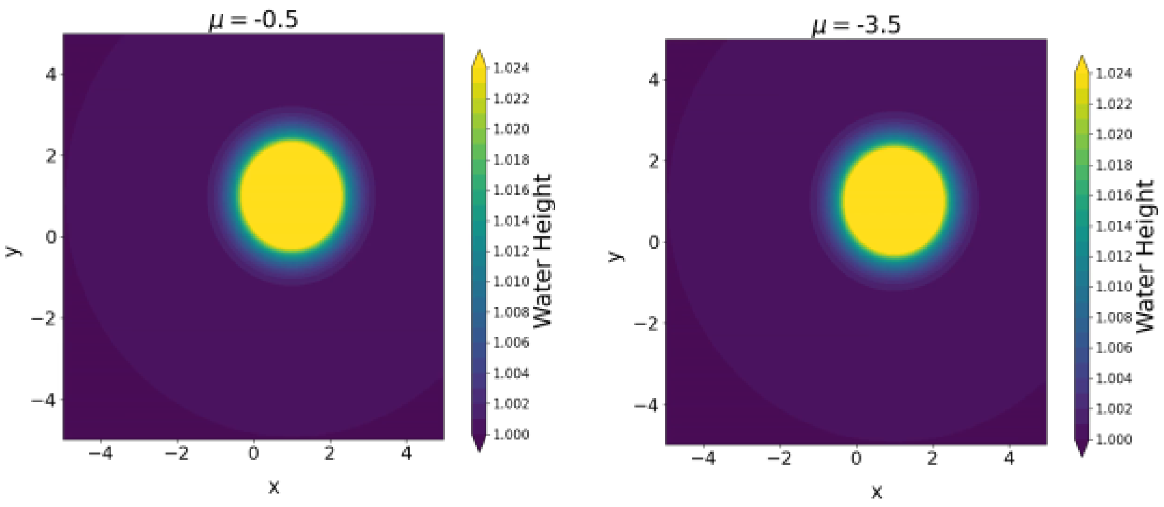


Figure above: FOM solutions to SWE for $\mu = -0.5$ (left) and $\mu = -3.5$ (right).

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)

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

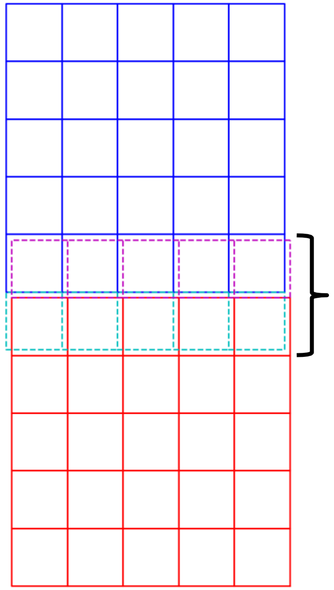
Enforcement of boundary conditions (BCs) in ROM at Schwarz boundaries

- 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

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



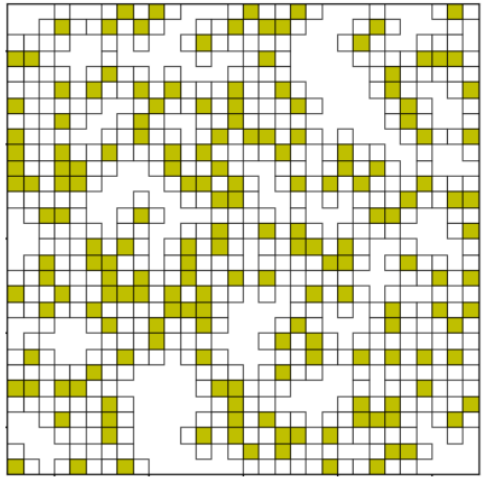
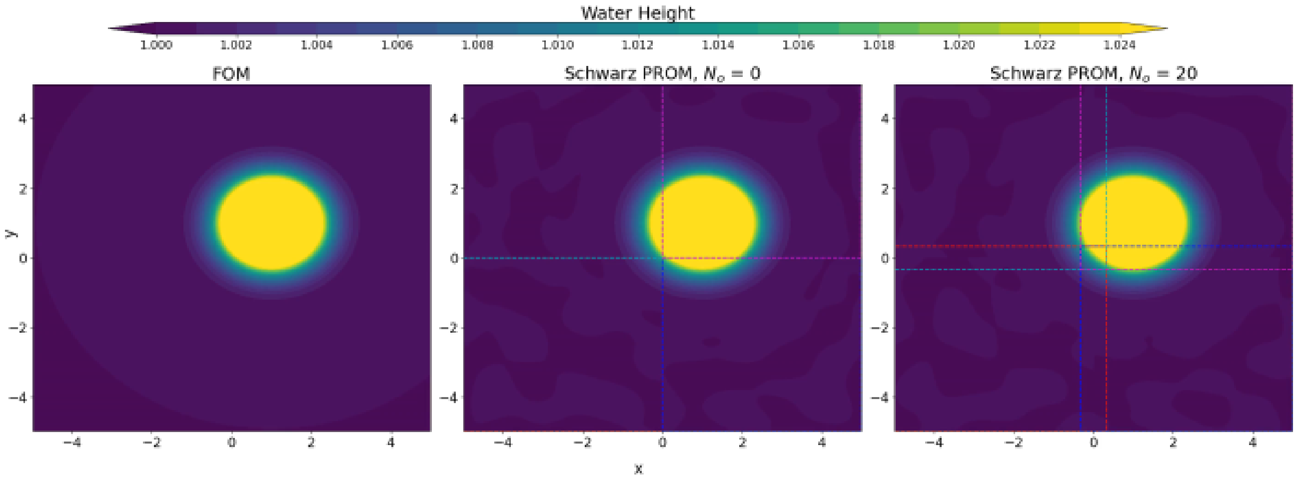


Figure above: sample mesh (yellow) and stencil (white) cells

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

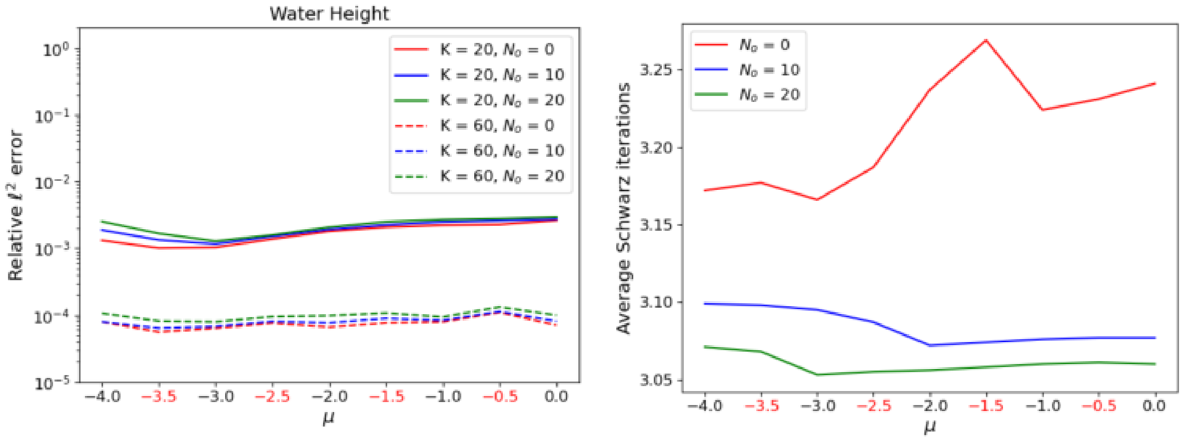
Schwarz All-ROM Domain Overlap Study

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.

- 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)



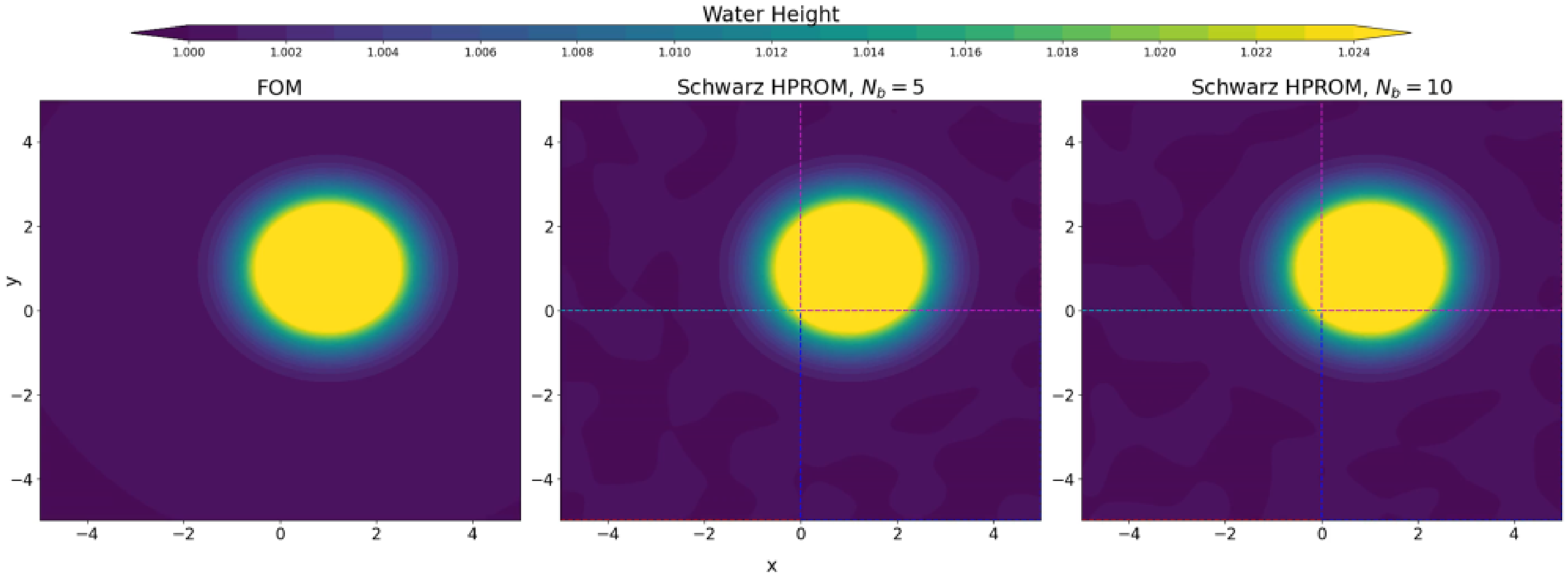
Figures above: relative error and average # Schwarz iterations as a function of μ and N_o. Black μ: training, red μ: testing.

- **Schwarz 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

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.

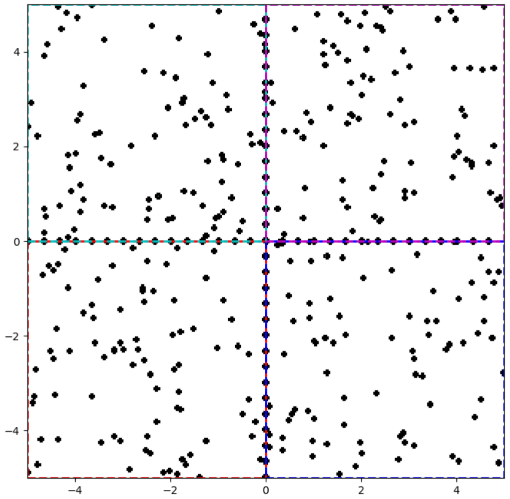
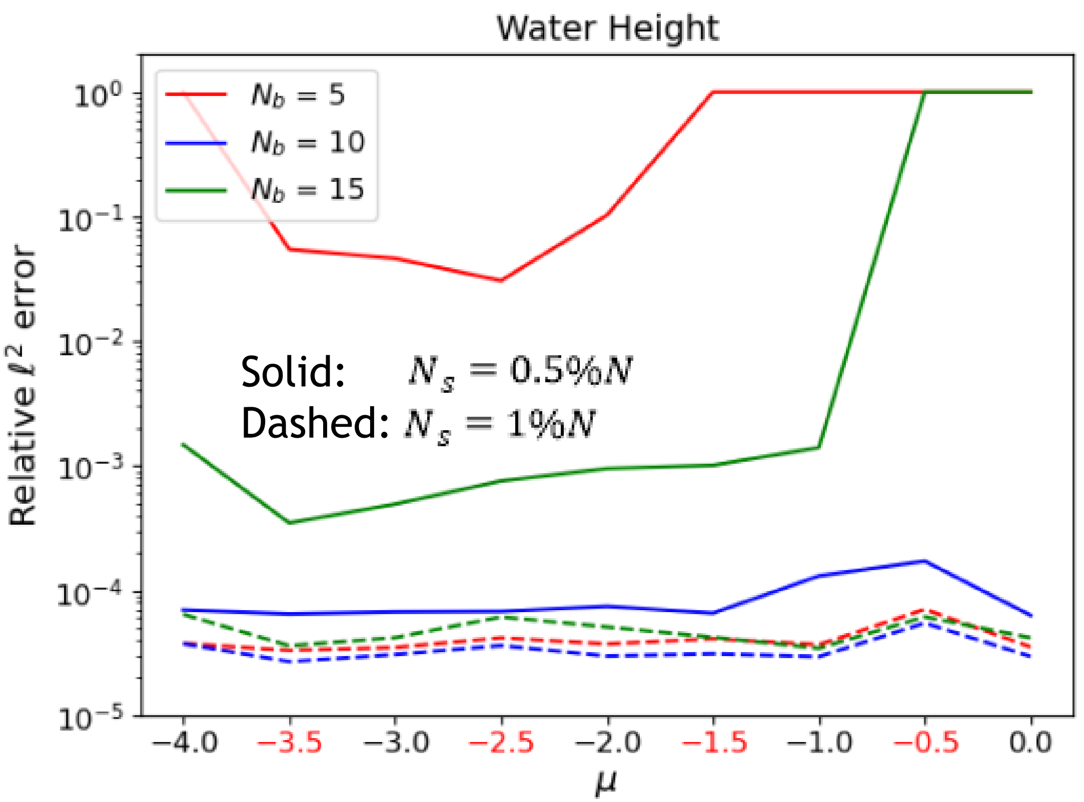
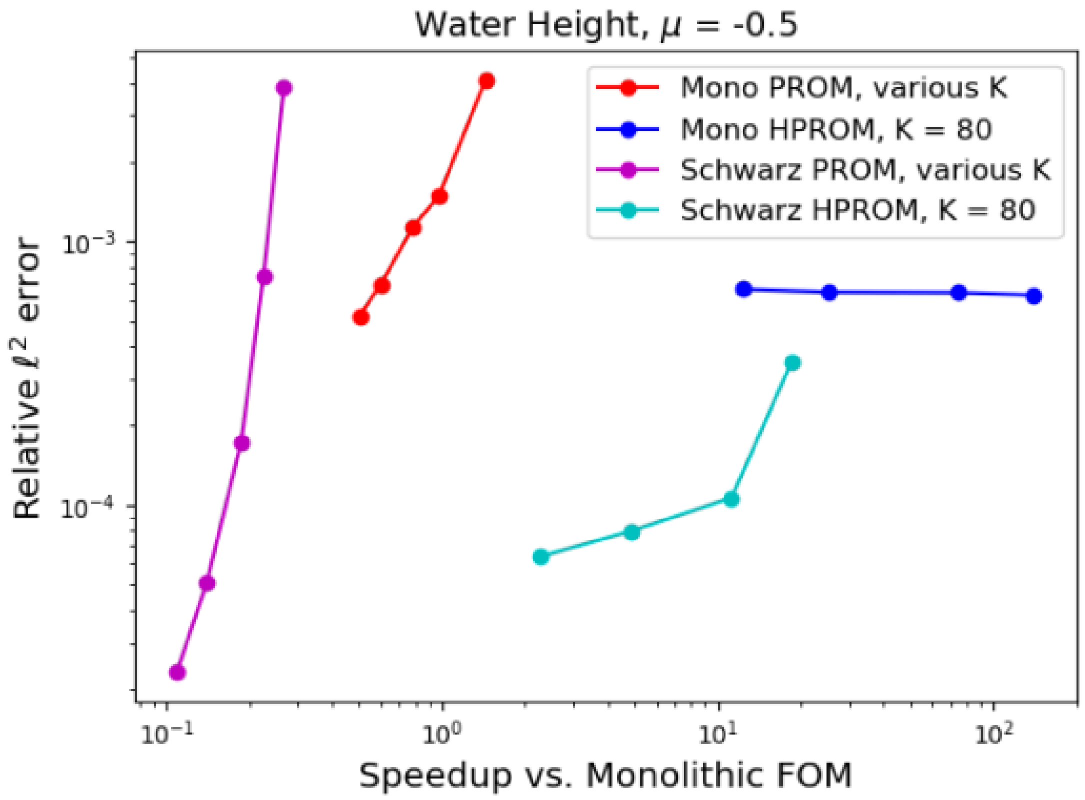


Figure above: example sample mesh with sampling rate $N_b = 10\%$

- 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 ~**10% boundary sampling** (movie above, right-most frame)

Coupled HROM Performance

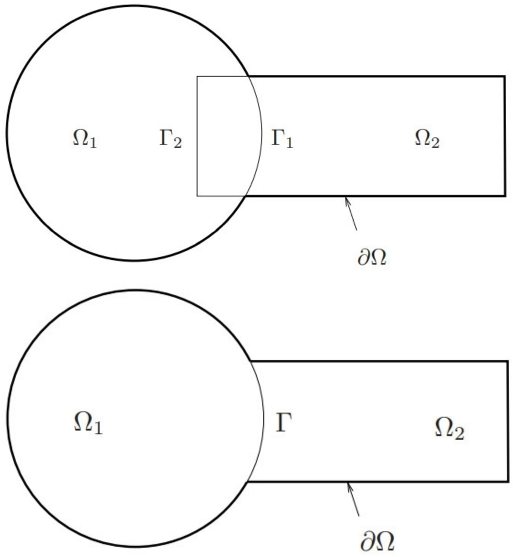


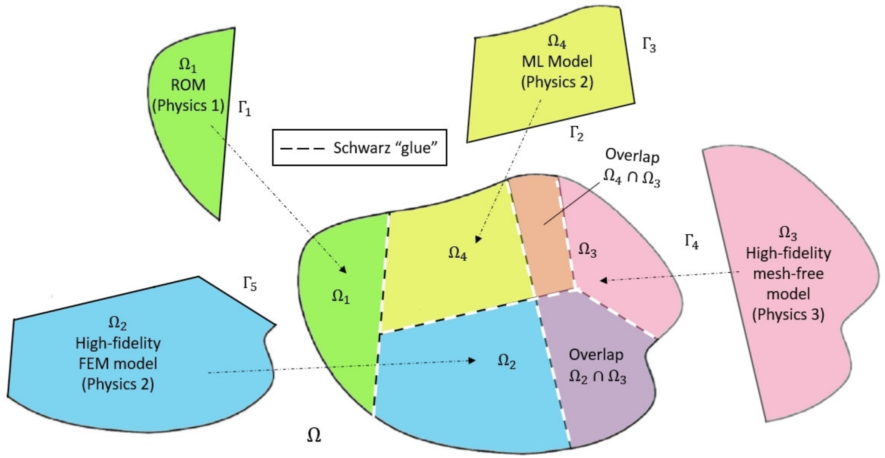


- For a fixed ROM dimension, Schwarz delivers **lower error** and **comparable cost!**
- There are noticeable **cost savings** relative to **monolithic FOM!**
- Accuracy similar for **predictive** μ (red) and **non-predictive** μ (black) cases.

Outline

- 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 Probler**
- Summary & Future Work





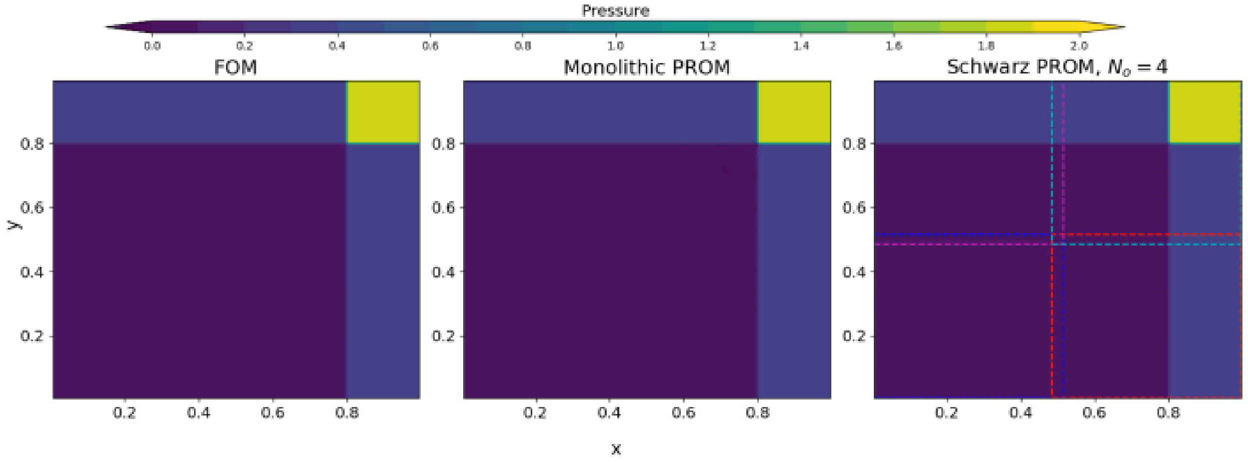
*Full-Order Model. #Reduced Order Model.

Teaser: 2D Euler Equations Riemann Problem

Pressio

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

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



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: $p_1 \in [1.125, 1.375, 1.625, 1.875]$

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**

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.

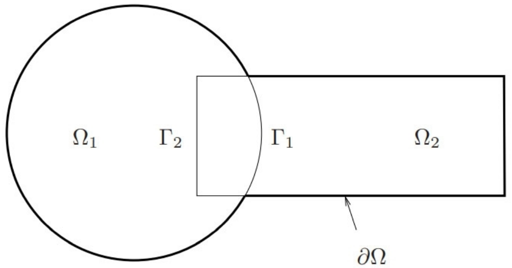
Outline

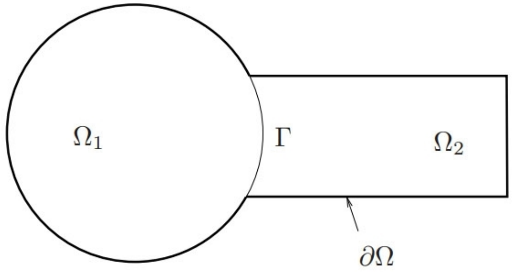
- The Schwarz Alternating Method for Domain Decomposition-Based Coupling

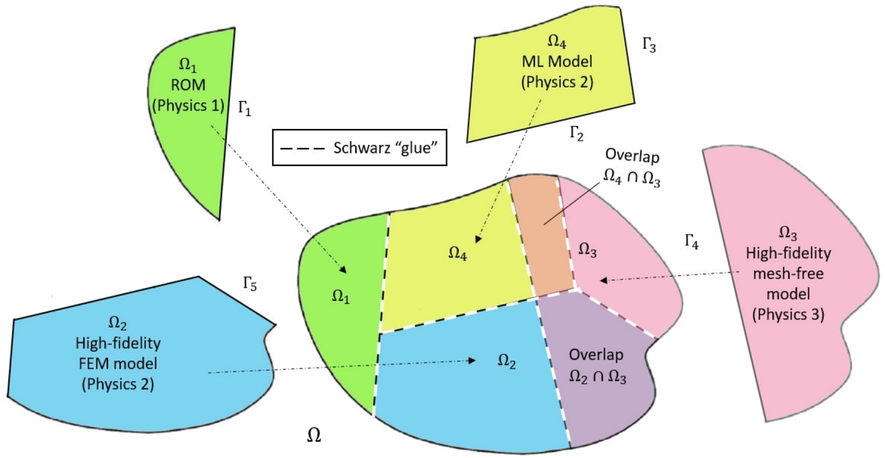
- Extension to FOM*-ROM# and ROM-ROM Coupling

- Numerical Examples
  - ➢ 2D Burgers Equation
  - ➢ 2D Shallow Water Equations
  - ➢ Teaser: 2D Euler Equations Riemann Problem

- **Summary & Future Work**







*Full-Order Model. #Reduced Order Model.

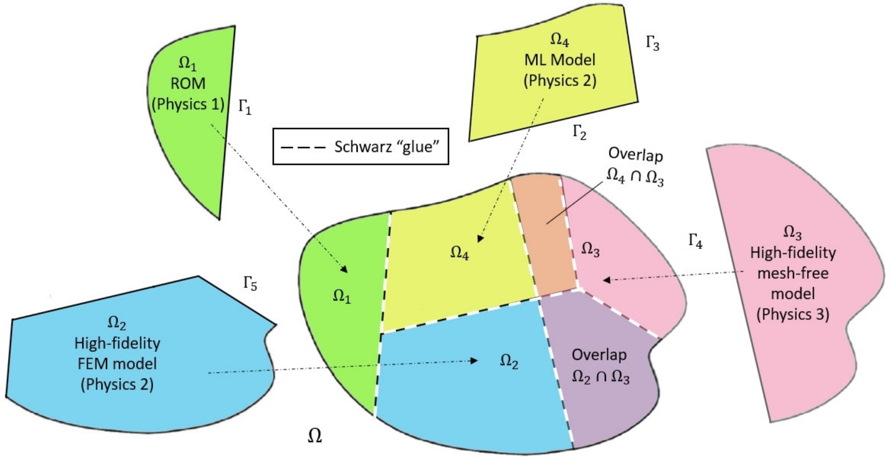
Summary and Future Work

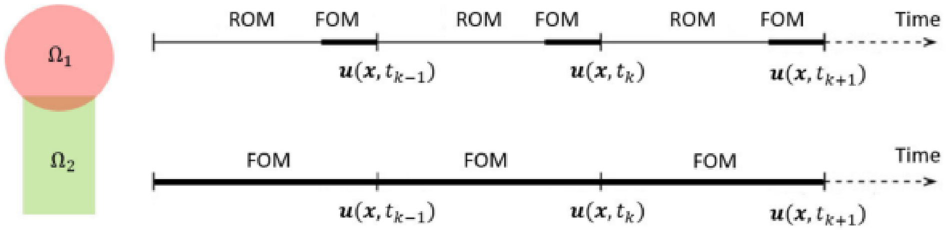
Summary:

- 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





* https://pressio.github.io

Team & Acknowledgments


Irina Tezaur


Chris Wentland


Francesco Rizzi


Joshua Barnett


Alejandro Mota

Sandia National Laboratories


Will Snyder


Ian Moore

LDRD
LABORATORY DIRECTED RESEARCH & DEVELOPMENT

References

[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

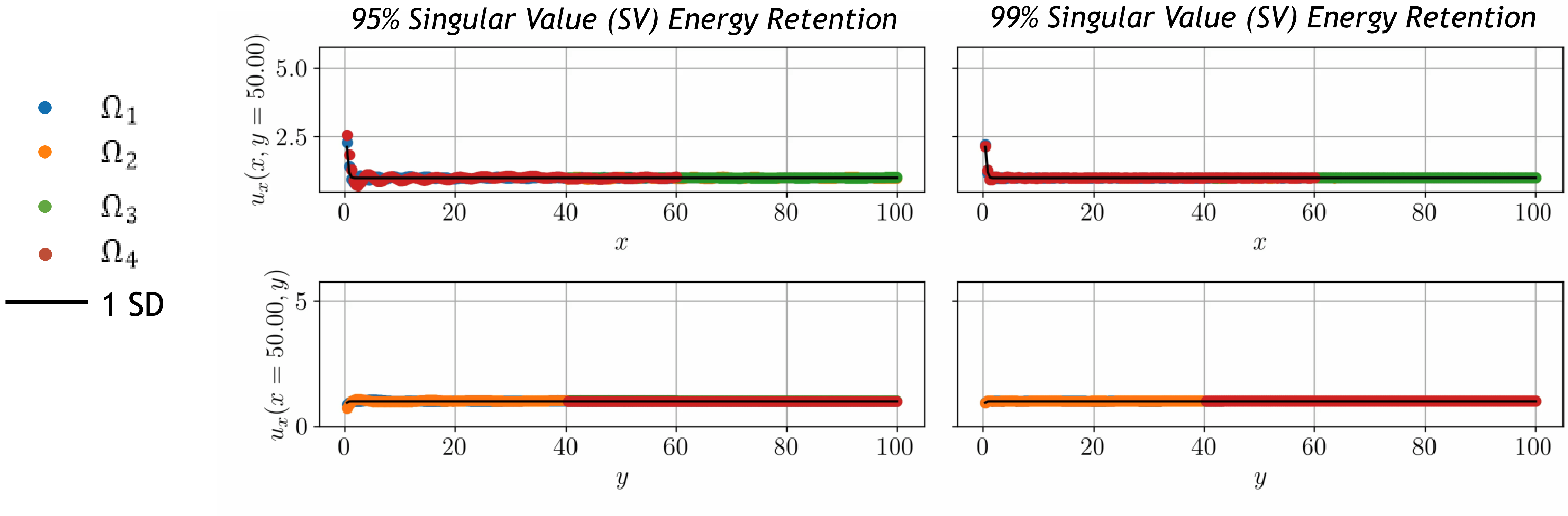
[4] W. Snyder, **I. Tezaur**, C. Wentland. "Domain decomposition-based coupling of physics-informed neural networks via the Schwarz alternating method", ArXiv pre-print, 2023. https://arxiv.org/abs/2311.00224

[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

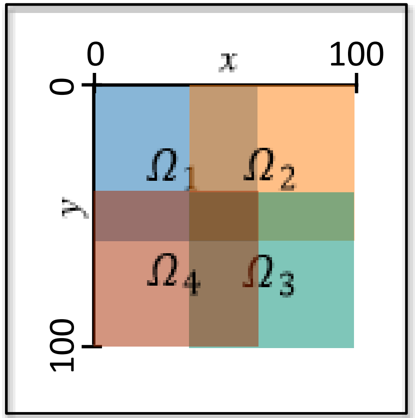
Start of Backup Slides

All-ROM Coupling



- Method converges in **only 3 Schwarz iterations** per controller time-step
- **Errors** $O(1\%)$ or less
- **1.47× 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**



Summary

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!

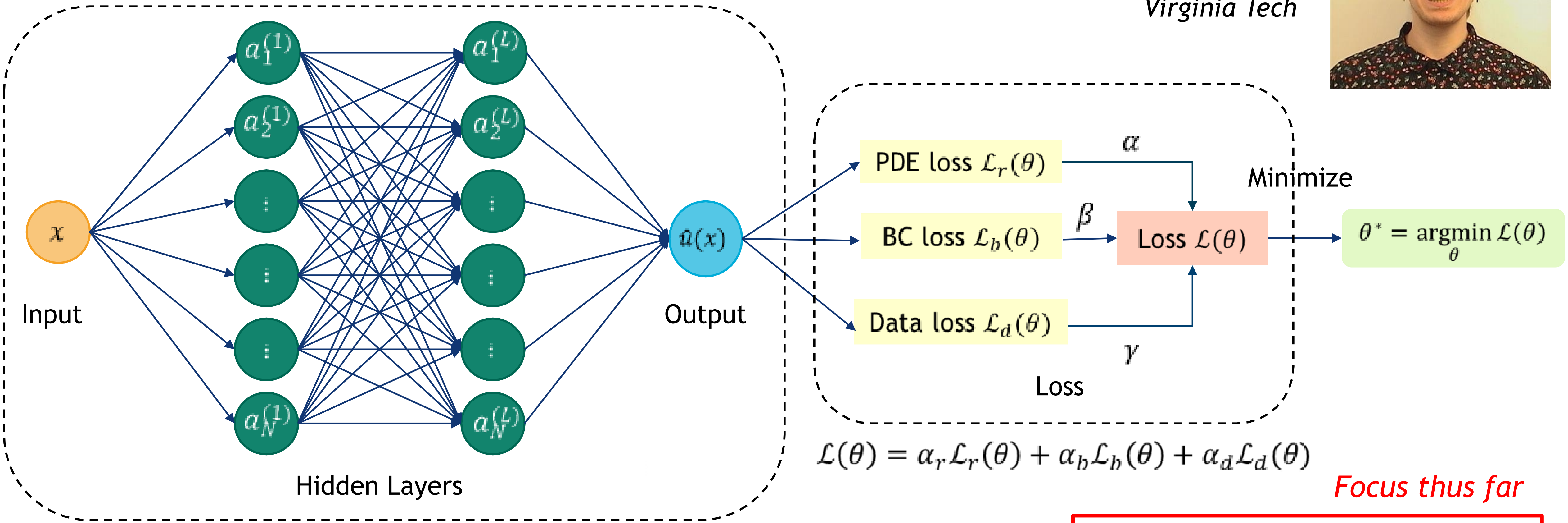
- ☺ Ability to couple regions with ***different non-conformal meshes***, ***different element types*** and ***different levels of refinement***.
- ☺ Ability to use ***different solvers (including ROM/FOM)*** and ***time-integrators*** in different regions.

☺ ***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

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

Focus thus far

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

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

Related work: Li et al., 2019, Li et al., 2020, Wang et al., 2022.

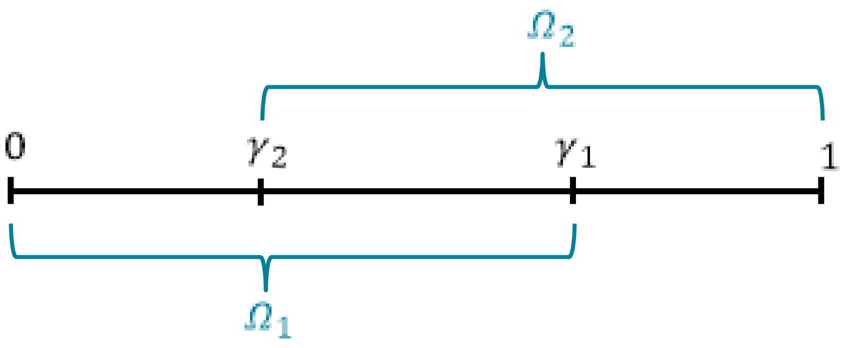
Bonus: PINN-PINN coupling

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 $\partial\Omega = \{0,1\}$

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

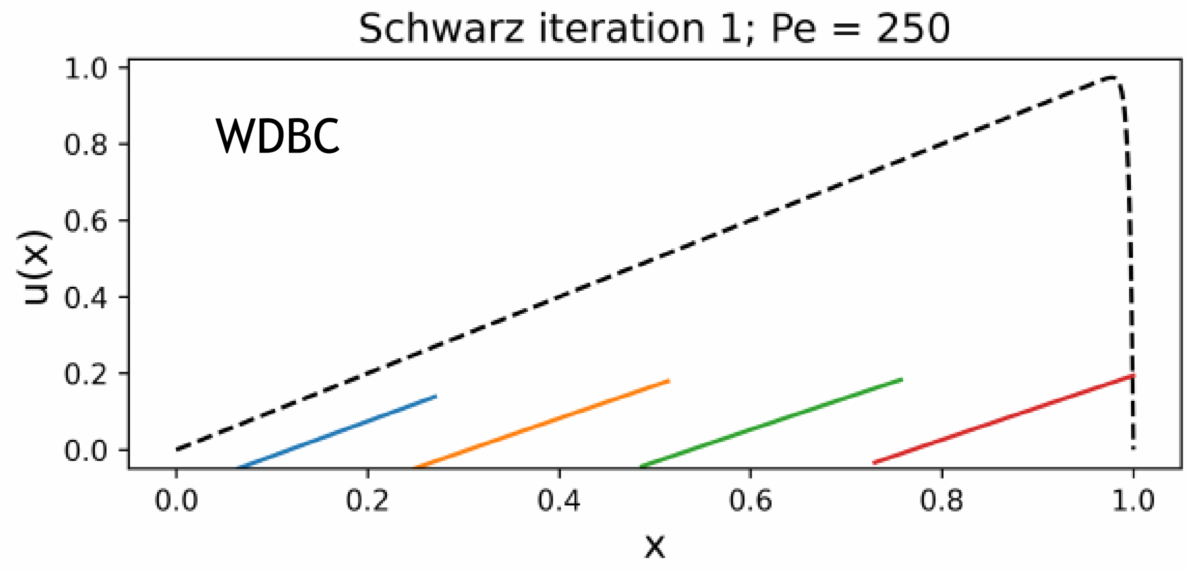
$$\mathcal{L}_{b,i}(\theta) = MSE\left(NN_{\Omega_i}(\partial\Omega,\theta)\right) + MSE\left(NN_{\Omega_i}(\gamma_i,\theta) - NN_{\Omega_j}(\gamma_i,\theta)\right)$$

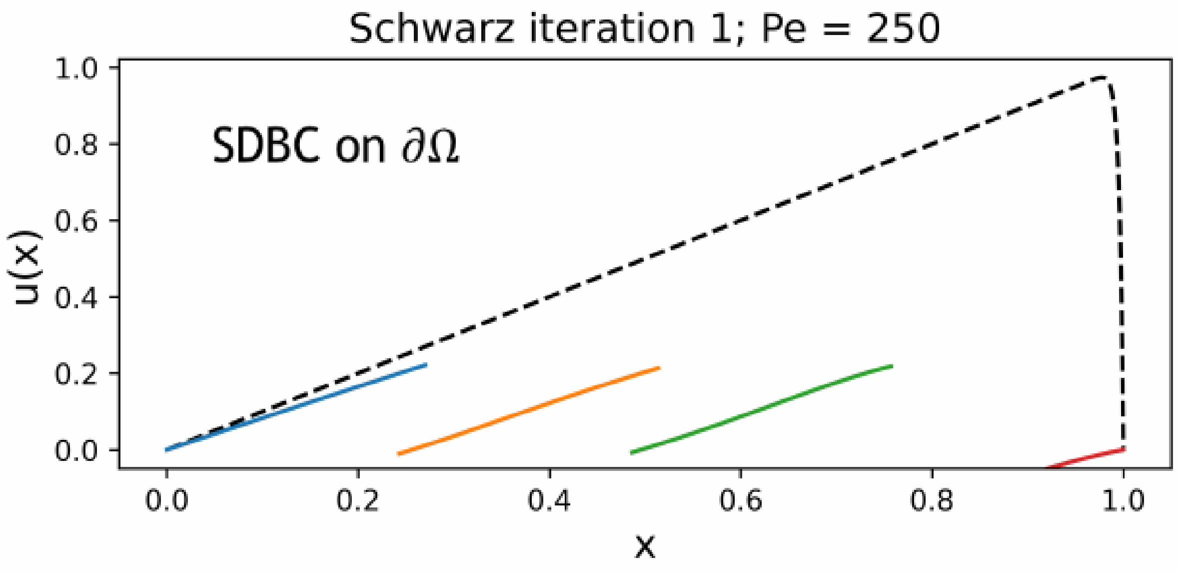
Schwarz PINN training algorithm:

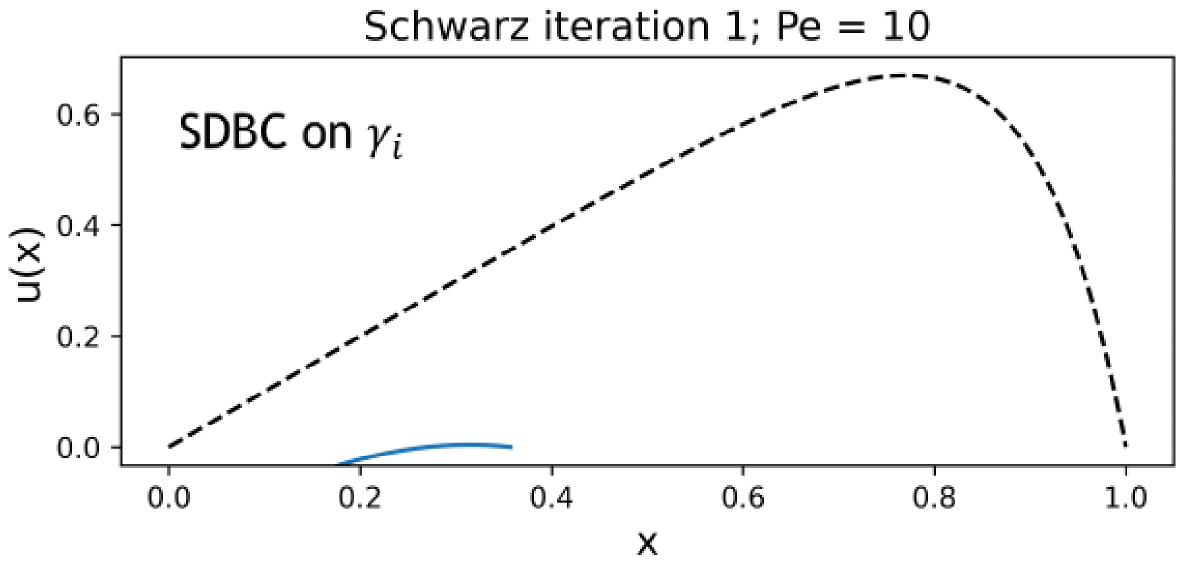
Loop over subdomains Ω_i until convergence of Schwarz method

- **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. $v(\gamma_i) = 1$

Bonus: PINN-PINN coupling

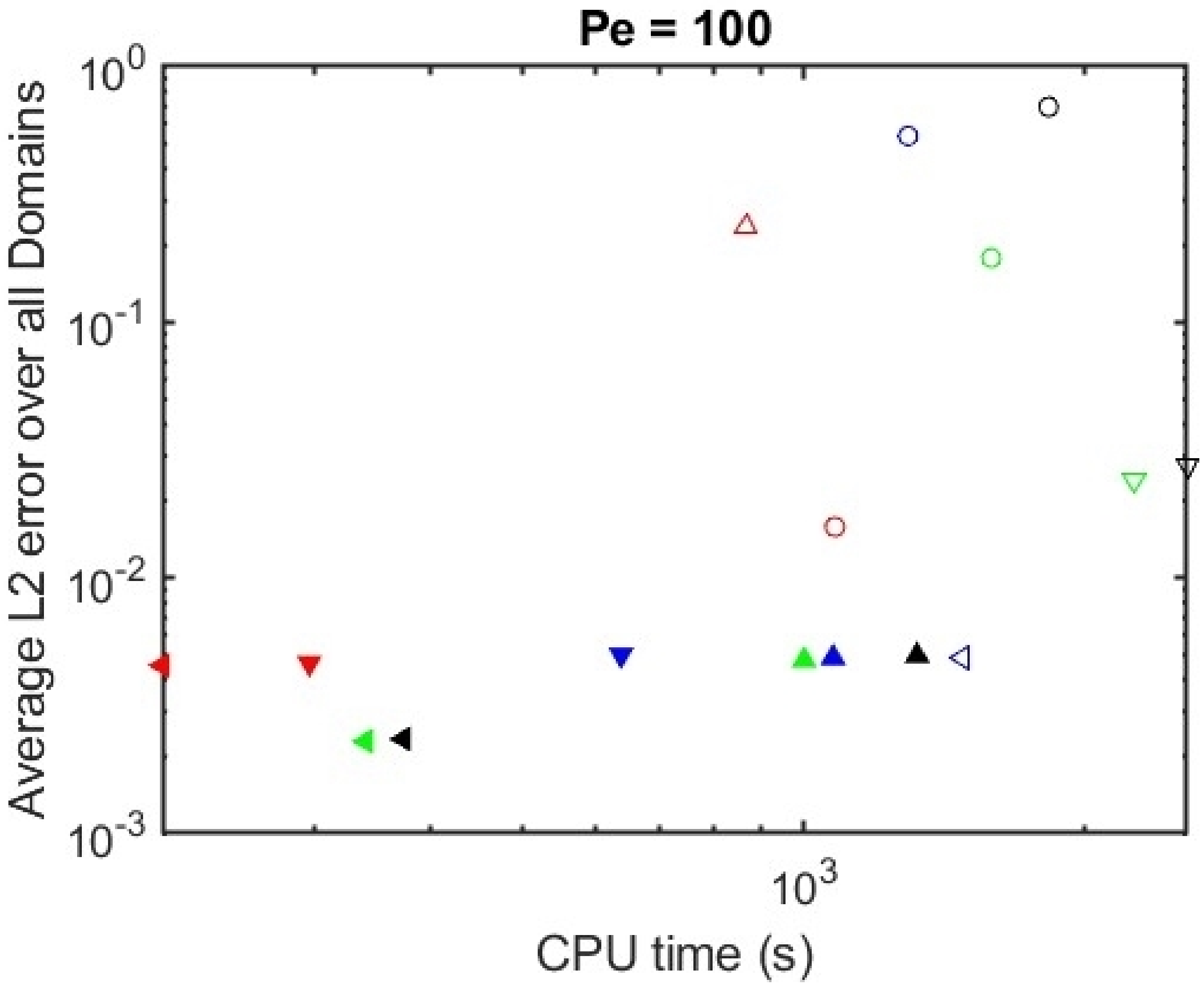


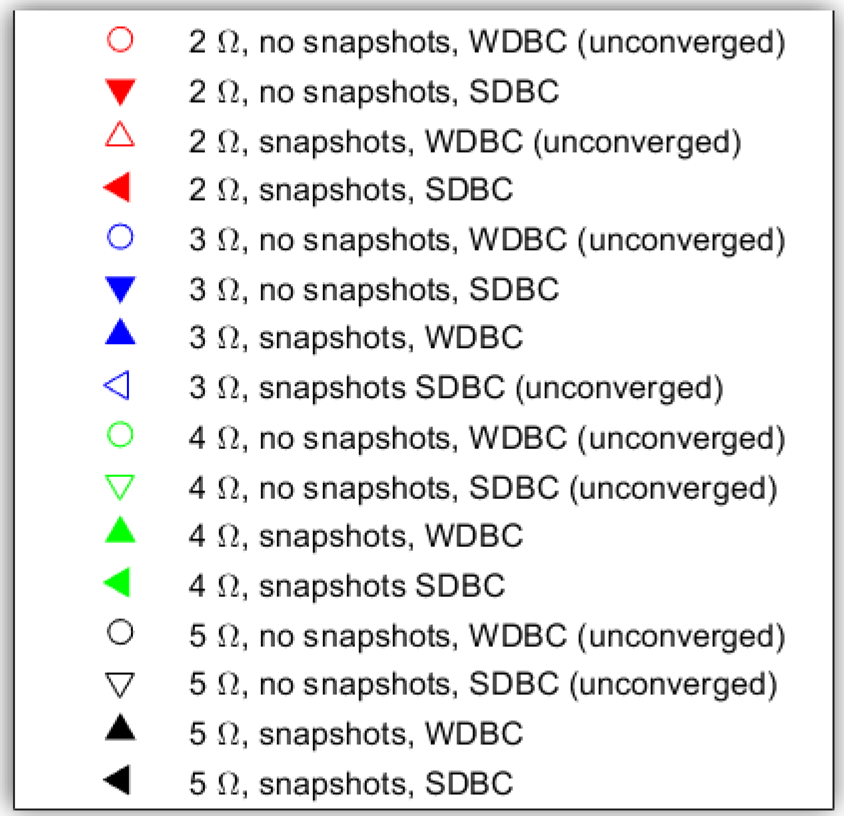




- How **Dirichlet boundary conditions** are handled has a large impact on PINN convergence
- Convergence not improved in general with **increasing overlap**
- Increasing **# subdomains** in general will increase CPU time

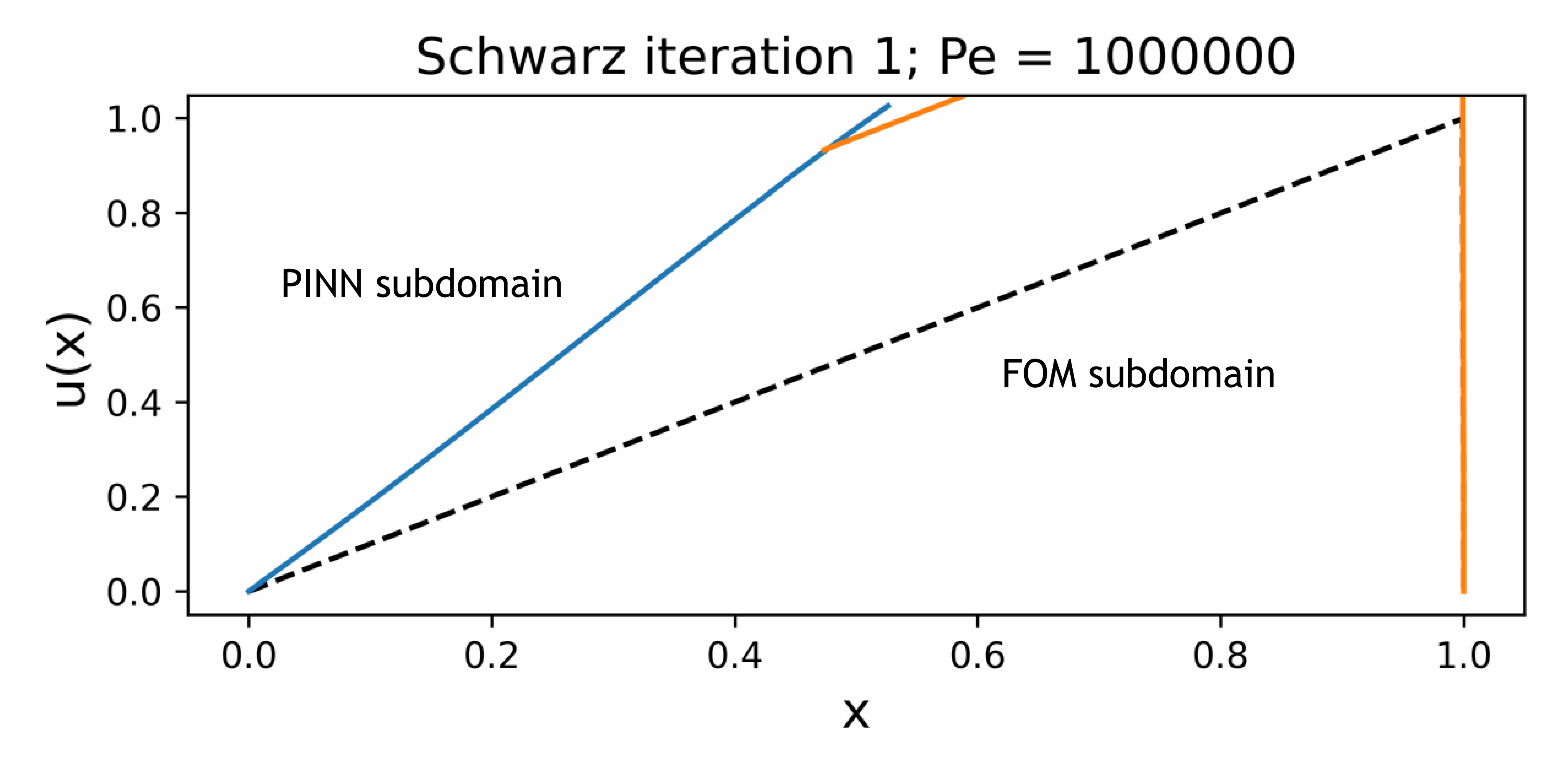
Bonus: PINN-PINN coupling





- Using **SDBCs** 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

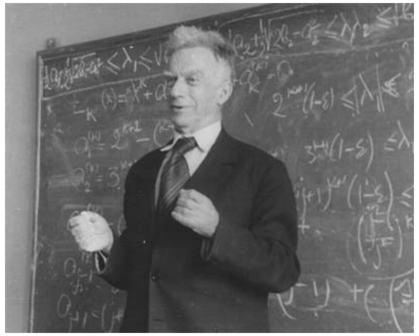
Theoretical Foundation

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

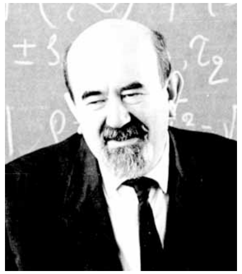
- S.L. Sobolev (1936): posed Schwarz method for ***linear elasticity*** in variational form and ***proved method's convergence*** by proposing a convergent sequence of energy functionals.
- S.G. Mikhlin (1951): ***proved convergence*** of Schwarz method for general linear elliptic PDEs.
- P.-L. Lions (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 $\boldsymbol{\Phi}[\boldsymbol{\varphi}]$) with a ***geometric convergence rate.***

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

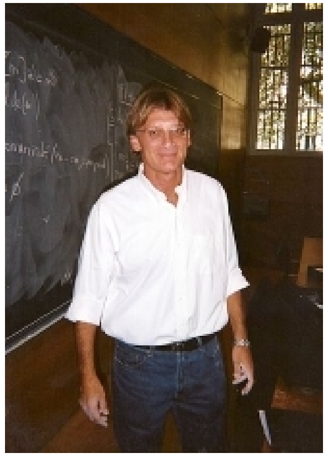
$$\nabla\cdot\boldsymbol{P} + \boldsymbol{B} = \boldsymbol{0}$$



S.L. Sobolev (1908 – 1989)



S.G. Mikhlin (1908 – 1990)



P.- L. Lions (1956-)

A. Mota, I. Tezaur, C. Alleman

Convergence Proof*

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

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

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

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

Schwarz Alternating Method for Dynamic Multiscale Coupling: Theory

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

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

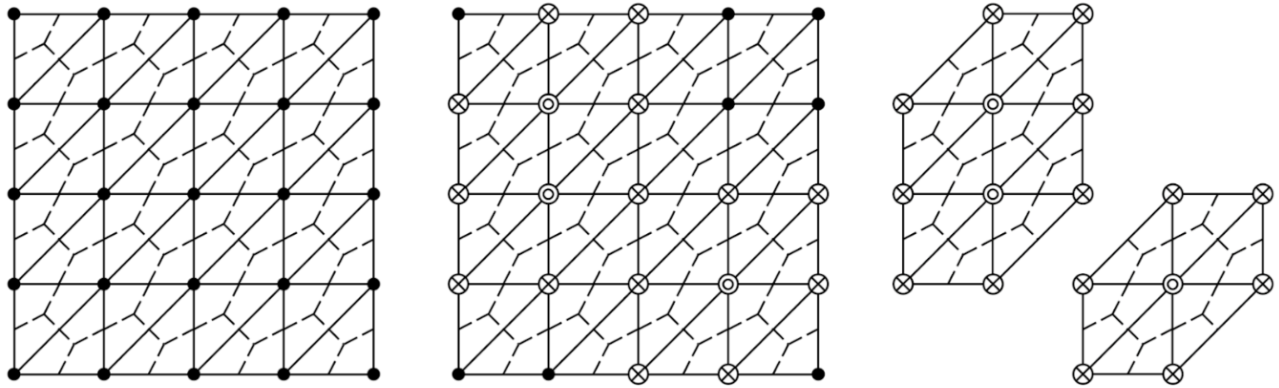
  - ➢ $\delta^2 S[\boldsymbol{\varphi}_h]$ can always be made positive by choosing a ***sufficiently small*** Δt
  - ➢ Numerical experiments reveal that Δt requirements for ***stability/accuracy*** typically lead to automatic satisfaction of this bound.

Energy-Conserving Sampling and Weighting (ECSW)

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

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

- $L_e \in \{0,1\}^{d_e \times N}$ 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: ⊚ represents a selected node attached to a selected element; and ⊗ 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 $u_i, i \in 1, \ldots, n_h$ used to generate the **state basis** V, we can train the reduced mesh
- Snapshots are first **projected** onto their associated basis and then **reconstructed**

$$\begin{aligned} c_{se} &= W^T L_e^T r_e\left(L_{e^+}\left(u_{ref} + V\, V^T\left(u_s - u_{ref}\right)\right), t\right) \in \mathbb{R}^n \\ d_s &= r_k(\tilde{u}, t) \in \mathbb{R}^n, \qquad s = 1, \ldots, n_h \end{aligned}$$

- We can then form the **system**

$$\boldsymbol{C} = \begin{pmatrix} c_{11} & \cdots & c_{1N_e} \\ \vdots & \ddots & \vdots \\ c_{n_h 1} & \cdots & c_{n_h N_e} \end{pmatrix}, \qquad \boldsymbol{d} = \begin{pmatrix} d_1 \\ \vdots \\ d_{n_h} \end{pmatrix}$$

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

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

- Solve the above optimization problem using a **non-negative least squares solver** with an **early termination condition** to **promote sparsity** of the vector $\boldsymbol{\xi}$