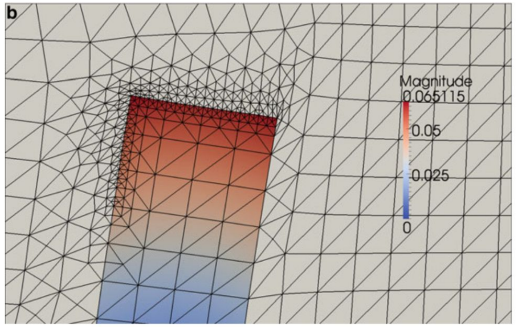
Enabling FSI: Ratel Adapter for preCICE

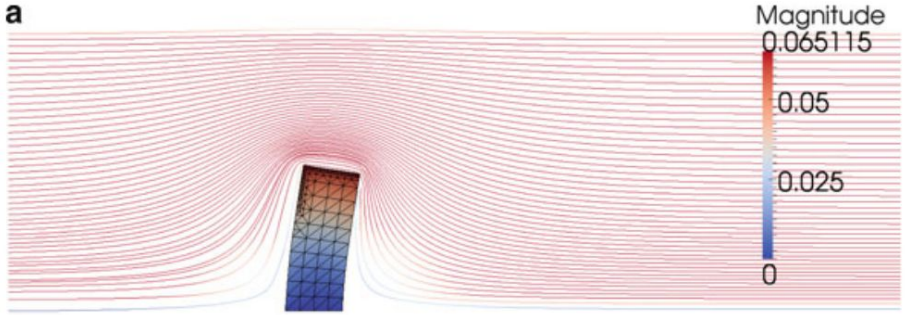
Stefano Fochesatto
University of Colorado Boulder

Outline

- Primer on Fluid-Structure Interaction
- The preCICE Library
- Coupling Schemes and Data Mapping
- Adapter Software Requirements
- 3D Tube Demo

A very quick and not at all comprehensive description of an FSI problem





FSI occurs when a fluid interacts with a solid structure, exerting traction forces that cause deformation and, thus altering the fluid flow itself.

Two Frameworks

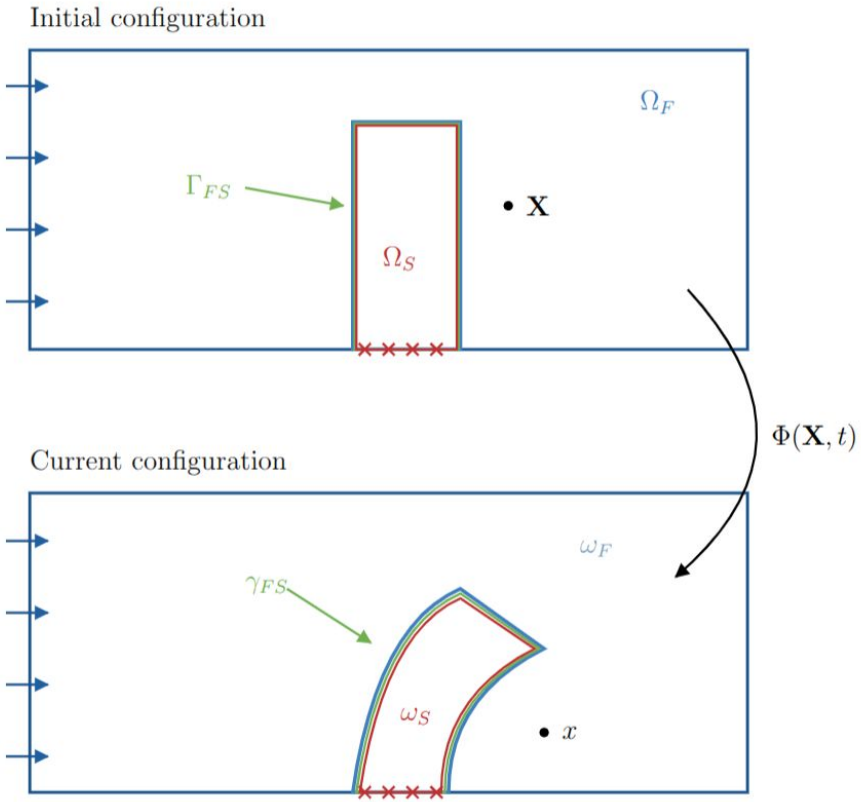
The Interaction must still obey balance of linear momentum and conservation of mass.

- Tractions along and FSI Boundary need to be equal and opposite
- Velocity of the solid and fluid along the boundary must be equal .

Two Frameworks

- In structural mechanics displacement is generally posed in a Lagrangian frame (the motion of a body is related to a fixed material point)

- In fluid mechanics, fluid velocity and pressure are generally posed in a Eulerian frame (the motion of the body is a function of a fixed point in space)



An example fully coupled FSI problem

Chapter 29: The FEniCS Book

$$
\begin{aligned}
(f): \quad & \rho_F(\dot{u}_F + \operatorname{grad} u_F \cdot u_F) - \operatorname{div} \sigma_F(u_F, p_F) = b_F && \text{in } \omega_F(t), \\
& \operatorname{div} u_F = 0 && \text{in } \omega_F(t), \\
(S): \quad & \rho_S \ddot{U}_S - \operatorname{Div} \Sigma_S(U_S) = B_S && \text{in } \Omega_S \times (0,T], \\
(M): \quad & \dot{U}_M - \operatorname{Div} \Sigma_M(U_M) = 0 && \text{in } \Omega_F \times (0,T],
\end{aligned}
$$

$$
\begin{aligned}
(J_M\ (\sigma_F \circ \Phi_M) \cdot F_M^{-\top}) \cdot N_F &= -\Sigma_S \cdot N_S && \text{on } \Gamma_{FS}, \\
u_F \circ \Phi_M &= \dot{U}_S && \text{on } \Gamma_{FS}.
\end{aligned}
$$

Partitioned vs Monolithic

The (arbitrary lagrangian eulerian) ALE formulation of Navier-Stokes allows the moving fluid problem to be solved in the initial configuration.

Monolithic Approach:

- Matching spatial discretization on FSI boundary.
- Matching FE spaces for Fluid velocity, Solid displacement, Mesh displacement.

https://redbkit.github.io/redbKIT/math/FSIsolver/

Partitioned vs Monolithic

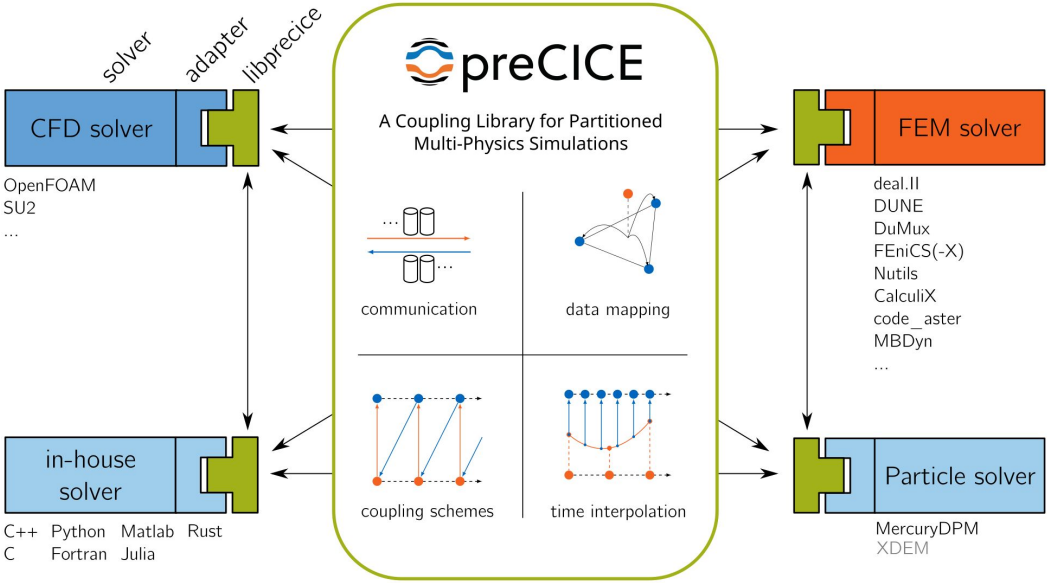
Alternatively we keep the solvers partitioned. To recover some of the benefits of the monolithic approach we need:

- Conservative and Consistent mappings across FSI boundary between discretizations.
- Methods for conserving physics at the boundary.

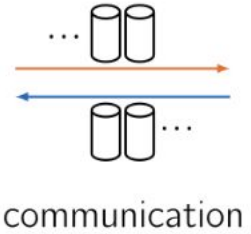
preCICE provides methods for both of these and more.

The preCICE Library

preCICE is a general software library for partitioned simulations.



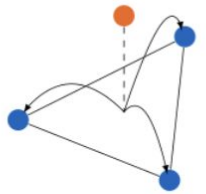
The preCICE Library



Communications

- **Independent Executables:** Participants maintain separate MPI communicators and can run across heterogeneous clusters (e.g., coupling CPU nodes with GPU nodes).
- **Fully Parallel & Asynchronous:** Data transfer between solvers is parallel and non-blocking.
- **Targeted Exchange:** Communication is strictly limited to the specific MPI ranks that need to exchange interface data.

The preCICE Library

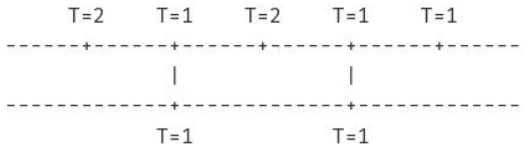


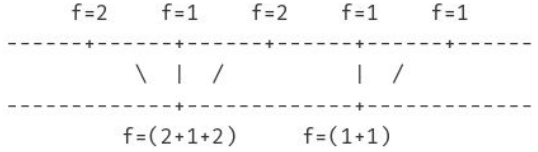
data mapping

Data Mapping

- **Non-Conforming Mesh Support:** preCICE handles all data transfer between non-matching and non-conforming coupling interfaces.

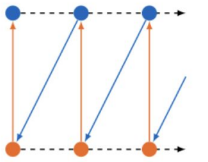
- **Consistent or Conservative Mappings**





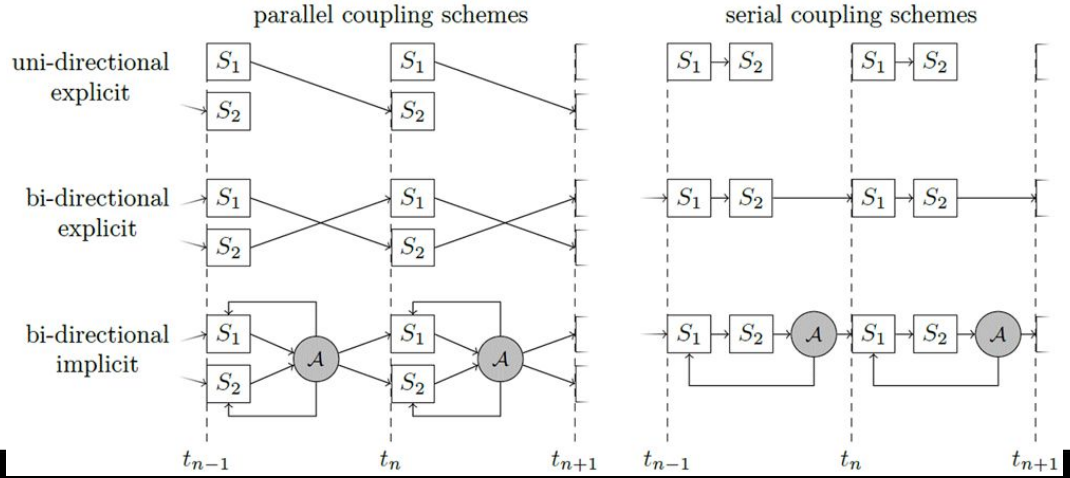
- **Projection or Kernel Methods:** preCICE offers projection based methods like nearest neighbor and Radial-basis function interpolation.

The preCICE Library



Coupling Schemes

- **Configurable Coupling Flow:** Define data dependencies at runtime, choosing between uni/bi-directional, explicit/implicit, and serial/parallel execution.



Coupling Schemes

Coupling Schemes

- **Explicit (Weak) Coupling:** Boundary conditions are passed between solvers once per time step. Is susceptible to added-mass instability (for a given time step, the inertia from the fluid in front of the flap is ignored)

- **Implicit (Strong) Coupling:** Solvers iterate multiple times per timestep until boundary field quantities converge.

Implicit Coupling Sc

- Implicit coupling requires finding the root of a fixed-point represents the combined execution of the coupled solvers:

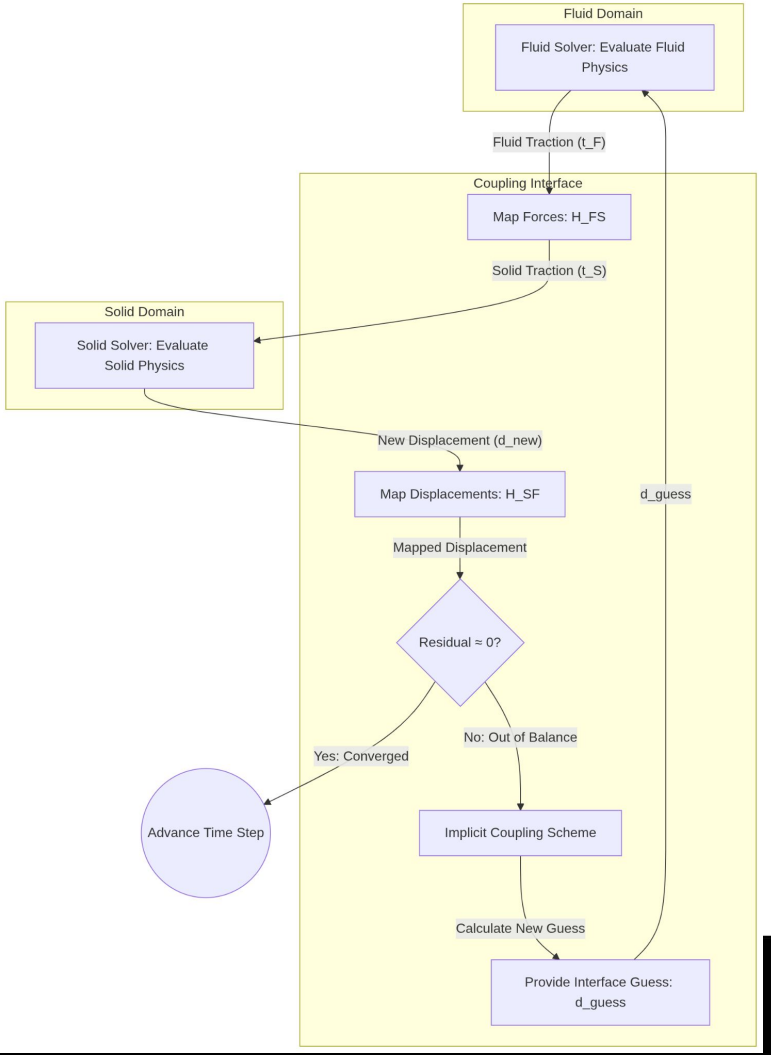
$$H(x) = x$$

Example (Serial Coupling): $S \circ F(x_1) = x_1$

- preCICE approximates the inverse Jacobian of the residual using multi-secant equations. The generalized Quasi-Newt

$$x_{k+1} = \tilde{x}_k + (W_k - J_{prev} V_k)\alpha_k - J_{prev} I$$

- $\tilde{x}_k = H(x_k)$ is the current solver output.
- V_k and W_k are observation matrices storing the history of differences.
- J_{prev} reuses the inverse Jacobian approximation from previ



Implicit Coupling Schemes

Ratel Adapter Implementation

- Implicit schemes require checkpointing!
  - Implemented with a TSPostStep callback and TSRollBack

```
// Check if we need to reload checkpoint
PetscCall(RatelAdapterReloadCheckpointIfRequired(ctx->adapter, ctx->U, ctx->V, &time, &step, &reloaded));
if (reloaded) {
  // Tell TS that the step failed and it must roll back its internal state
  PetscCall(TSRollBack(ts));
```

Data Mapping

- Black-Box Spatial Mapping
  - preCICE treats interface meshes as point clouds ($M_{S_1} \rightarrow M_{S_2}$)
  - Data transfer is executed via a linear mapping matrix $\mathbf{M}$:

$$\mathbf{M}\vec{v}_{S_1} = \vec{v}_{S_2}$$

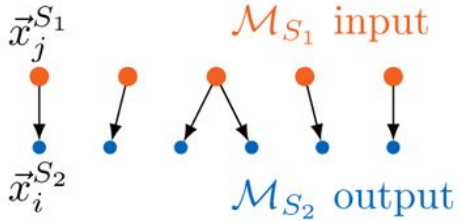
  - **Methods:** Nearest-neighbor, nearest-projection, and Radial Basis Function (RBF) interpolation.

- Consistent and Conservative Definitions
  - **Consistent Mapping**: *Row sums of* $\mathbf{M}$ *equal 1.*
  - **Conservative Mapping**: *Column sums of* $\mathbf{M}$ *equal 1 (uses* $\mathbf{M}^T$*).*

Projection Methods

- Nearest-Neighbor
- For an output vertex $x_i^{S_2}$ and an input vertex $x_j^{S_1}$, the mapping matrix entries are:

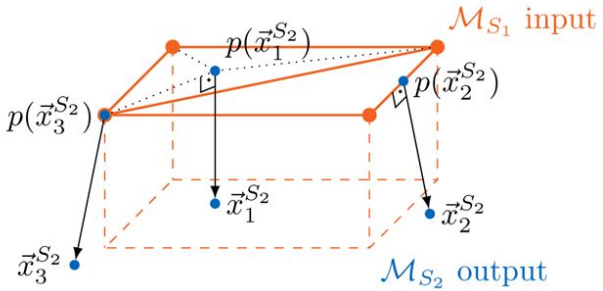
$$M_{ij} = \begin{cases} 1, & \text{if } i = j \text{ (the nearest neighbor)} \\ 0, & \text{otherwise} \end{cases}$$



Projection Methods

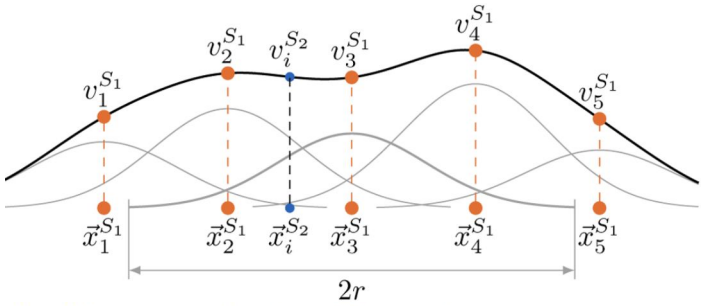
- Nearest-Projection
- Projects the output vertex $x_i^{S_2}$ onto the geometric connectivity of the input mesh (Triangle $\rightarrow$ Edge $\rightarrow$ Vertex) to interpolate values.
- If output vertex $x_i^{S_2}$ projects onto a triangle defined by input vertices $(x_{j_1}^{S_1}, x_{j_2}^{S_1}, x_{j_3}^{S_1})$ with calculated barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$:

$$M_{ij} = \begin{cases} \lambda_1, & \text{if } j = j_1 \\ \lambda_2, & \text{if } j = j_2 \\ \lambda_3, & \text{if } j = j_3 \\ 0, & \text{otherwise} \end{cases}$$



(Uses linear weights w_1, w_2 if projected onto an edge).

Kernel Methods



- Radial Basis Functions
- Instead of local geometric projections, RBF builds a global interpolation function $s(\vec{x})$ evaluated at the output vertices $x_i^{S_2}$:

$$s(\vec{x}) = \sum_{j=1}^{n_1} \lambda_j \phi(||\vec{x} - x_j^{S_1}||_2) + q(\vec{x})$$

  - ϕ: Radially symmetric basis function centered at input nodes.
  - $q(\vec{x})$: A first-order polynomial ensuring exact interpolation of constant/linear fields (guarantees consistency).

Data Mapping

Ratel Adapter Implementation

- Requires registering mesh vertices/FE dofs/Quadrature Points with preCICE.
  - Nearest Projection requires mesh connectivity. (not sure how it works for HO Elements)
  - Ghost nodes need handling.
  - Currently registering only locally-owned mesh vertices

- Data mapping from preCICE Format to Ratel Format

Adapter Software Requirements

Streamed Traction Boundary Conditions

- Since we are only registering mesh vertices, values are accumulated on a global vec and subtracted from the RHS.

```
// Wrapper for the TS I2Function to include preCICE forces
static PetscErrorCode ApplyTractionI2Function(TS ts, PetscReal t, Vec U, Vec U_t, Vec U_tt, Vec Y, void *ctx) {
  AdapterCtx *my_ctx = (AdapterCtx *)ctx;
  PetscFunctionBeginUser;

  // Evaluate Ratel's native residual (Internal Forces - Ratel Tractions)
  PetscCall(my_ctx->ratel_i2function(ts, t, U, U_t, U_tt, Y, my_ctx->ratel_i2ctx));
  // Explicitly subtract preCICE nodal forces from the total residual Y
  PetscCall(VecAXPY(Y, -1.0, my_ctx->F));
  PetscFunctionReturn(PETSC_SUCCESS);
}
```

Elastic 3D Tube (preCICE tutorial)

Physical Model & Material Properties

- 3D Hollow Cylinder (Inner Radius: 5mm, Outer Radius: 6mm, Length: 50mm)
- Material Model: Linear Elasticity (Infinitesimal Strain)
- Young's Modulus: 0.3 MPa
- Poisson's Ratio: 0.3
- Solid Density: 1200 kg/m³
- Boundary Conditions:
  - Solid: Both ends are rigidly clamped (Dirichlet), the inner surface is FSI boundary.
  - Fluid: A pressure pulse at the inlet triggers a traveling wave that deforms the tube.

Elastic 3D Tube (preCICE tutorial)

Numerical Discretization

- Structural Time Integrator: PETSc TSALPHA2
- Integration Parameters: \alpha_m=1.0, \alpha_f=1.0 (fully implicit Newmark-beta method)
- Time Step: .0001 seconds
- Solid Mesh (Gmsh): 80 layers (lengthwise), 6 elements/quadrant (circumferential), 2 elements (radial).