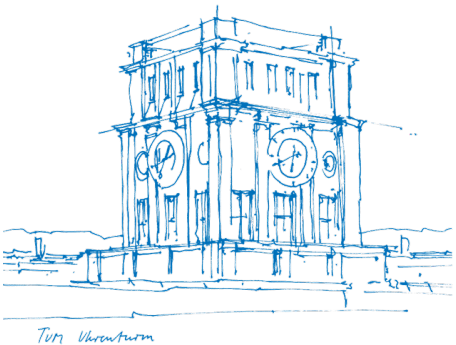
An introduction to the preCICE coupling library

ECCOMAS 2022

Frédéric Simonis
Technical University of Munich

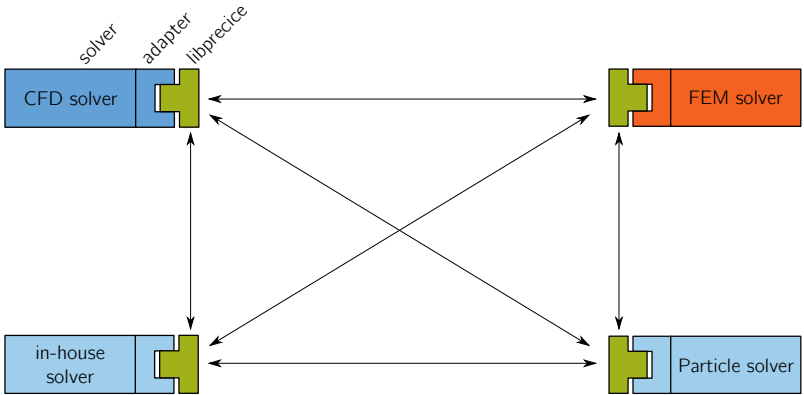
9th June 2022



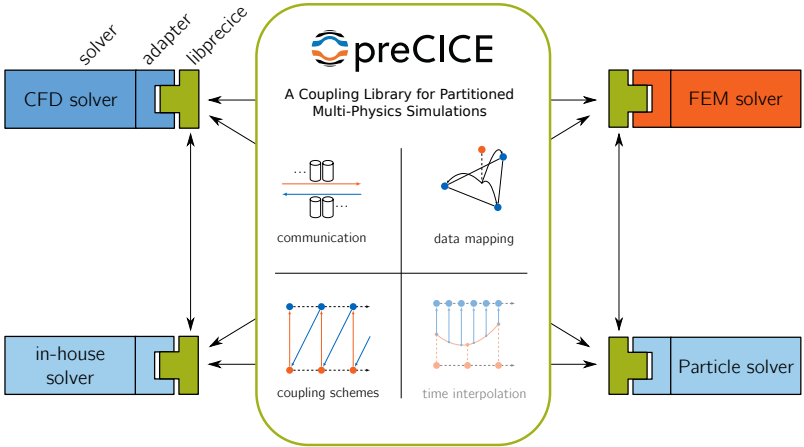
A brief introduction to preCICE

A brief introduction to preCICE

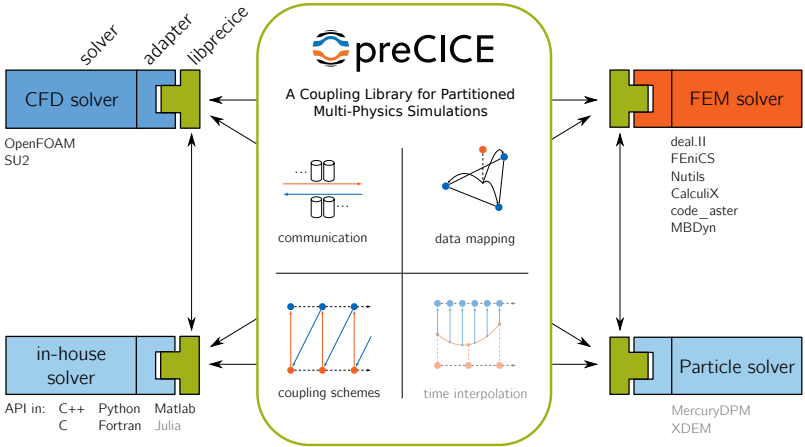
A brief introduction to preCICE



A brief introduction to preCICE



A brief introduction to preCICE



How to couple my own solver?

```
initializeSolver(); //e.g. setup and partition mesh



while (not simulationDone()){ // time loop

  dt = beginTimeStep(); // e.g. compute adaptive dt

  solveTimeStep(dt);


}
```

How to couple my own solver?

```
initializeSolver(); //e.g. setup and partition mesh
precice::SolverInterface precice("FluidSolver","precice-config.xml",rank,size);

p_dt = precice.initialize();
while (not simulationDone()){ // time loop

  dt = beginTimeStep(); // e.g. compute adaptive dt

  solveTimeStep(dt);


}
```

How to couple my own solver?

```
initializeSolver(); //e.g. setup and partition mesh
precice::SolverInterface precice("FluidSolver","precice-config.xml",rank,size);

p_dt = precice.initialize();
while (not simulationDone()){ // time loop

  s_dt = beginTimeStep(); // e.g. compute adaptive dt
  dt = min(p_dt, s_dt);
  solveTimeStep(dt);


}
```

How to couple my own solver?

```
1  initializeSolver(); //e.g. setup and partition mesh
2  precice::SolverInterface precice("FluidSolver","precice-config.xml",rank,size);
3
4  p_dt = precice.initialize();
5  while (not simulationDone()){ // time loop
6
7    s_dt = beginTimeStep(); // e.g. compute adaptive dt
8    dt = min(p_dt, s_dt);
9    solveTimeStep(dt);
10
11   p_dt = precice.advance(dt);
12 }
```

How to couple my own solver?

```
1  initializeSolver(); //e.g. setup and partition mesh
2  precice::SolverInterface precice("FluidSolver","precice-config.xml",rank,size);
3
4  p_dt = precice.initialize();
5  while (precice.isCouplingOngoing()){ // time loop
6
7    s_dt = beginTimeStep(); // e.g. compute adaptive dt
8    dt = min(p_dt, s_dt);
9    solveTimeStep(dt);
10
11   p_dt = precice.advance(dt);
12 }
```

How to couple my own solver?

```
initializeSolver(); //e.g. setup and partition mesh
precice::SolverInterface precice("FluidSolver","precice-config.xml",rank,size);
precice.setMeshVertices();
p_dt = precice.initialize();
while (precice.isCouplingOngoing()){ // time loop

  s_dt = beginTimeStep(); // e.g. compute adaptive dt
  dt = min(p_dt, s_dt);
  solveTimeStep(dt);

  p_dt = precice.advance(dt);
}
```

How to couple my own solver?

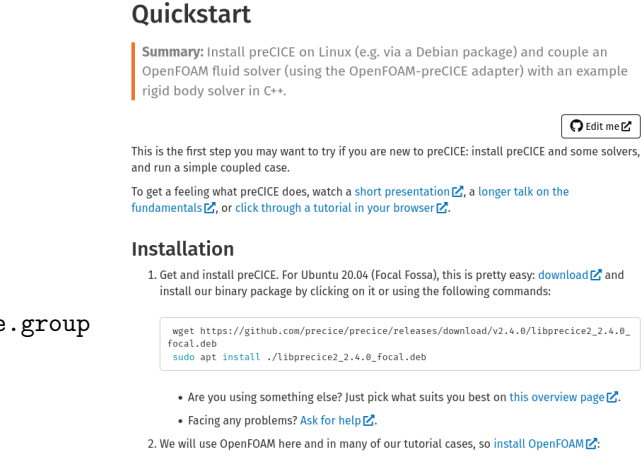
```
initializeSolver(); //e.g. setup and partition mesh
precice::SolverInterface precice("FluidSolver","precice-config.xml",rank,size);
precice.setMeshVertices();
p_dt = precice.initialize();
while (precice.isCouplingOngoing()){ // time loop
  precice.readData();
  s_dt = beginTimeStep(); // e.g. compute adaptive dt
  dt = min(p_dt, s_dt);
  solveTimeStep(dt);
  precice.writeData();
  p_dt = precice.advance(dt);
}
```

Selling points of preCICE

1. Multiple ready-to-use solver adapters (OpenFOAM, FEniCS, CalculiX, SU2, deal.ii)
2. Minimally-invasive coupling
3. Coupling of arbitrarily many components
4. Baked-in scalability
5. Robust quasi-Newton coupling
6. Free open-source software
7. Extensive high-quality documentation
8. Growing user community

How to get started?

- Quickstart: `precice.org/quickstart`
- Tutorials: `precice.org/tutorials`
- User documentation on the website: `precice.org/docs`
- Open source: `github.com/precice`
- Need help?
  - Discourse forum: `precice.discourse.group`
  - Gitter chatroom: `gitter.im/precice`



Quickstart

Summary: Install preCICE on Linux (e.g. via a Debian package) and couple an OpenFOAM fluid solver (using the OpenFOAM-preCICE adapter) with an example rigid body solver in C++.

Edit me

This is the first step you may want to try if you are new to preCICE: install preCICE and some solvers, and run a simple coupled case.

To get a feeling what preCICE does, watch a short presentation, a longer talk on the fundamentals, or click through a tutorial in your browser.

Installation

1. Get and install preCICE. For Ubuntu 20.04 (Focal Fossa), this is pretty easy: download and install our binary package by clicking on it or using the following commands:

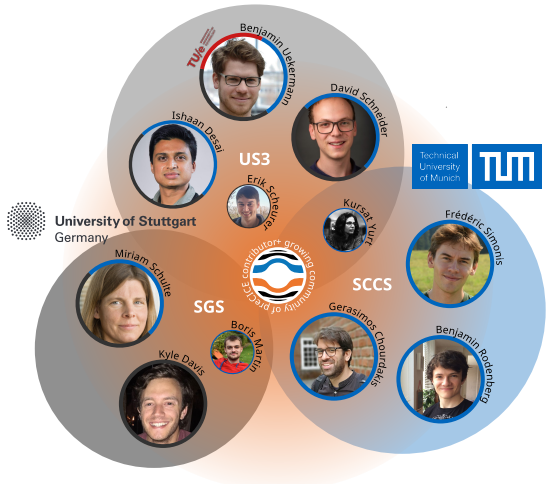
```
wget https://github.com/precice/precice/releases/download/v2.4.0/libprecice2_2.4.0_focal.deb
sudo apt install ./libprecice2_2.4.0_focal.deb
```

   - Are you using something else? Just pick what suits you best on this overview page.
   - Facing any problems? Ask for help.

2. We will use OpenFOAM here and in many of our tutorial cases, so install OpenFOAM:

Core Team 2022

- **Technical University of Munich**, SCCS (since $<$ 2008)
  - Hans-Joachim Bungartz
  - Gerasimos Chourdakis
  - Benjamin Rodenberg
  - Frédéric Simonis
- **University of Stuttgart**, SGS (since 2013)
  - Miriam Schulte
  - Kyle Davis
- **University of Stuttgart**, US3 (since 2021)
  - Benjamin Uekermann
  - Ishaan Desai
  - David Schneider



preCICE News

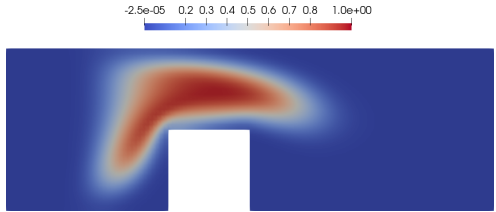
Tutorials New cases and solvers

Bindings Julia `github.com/precice/PreCICE.jl`

Adapters Elmer `github.com/precice/elmer-adapter`

FeniCSX `github.com/precice/fenicsx-adapter`

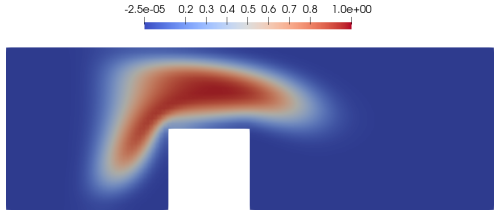
Two-scale coupling manager `github.com/precice/micro-manager`



New tutorial: channel transport

preCICE News

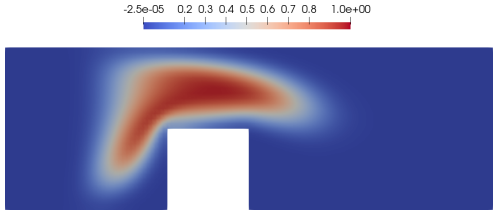
Tutorials New cases and solvers
Bindings Julia `github.com/precice/PreCICE.jl`
Adapters Elmer `github.com/precice/elmer-adapter`
FeniCSX `github.com/precice/fenicsx-adapter`
Two-scale coupling manager `github.com/precice/micro-manager`



New tutorial: channel transport

preCICE News

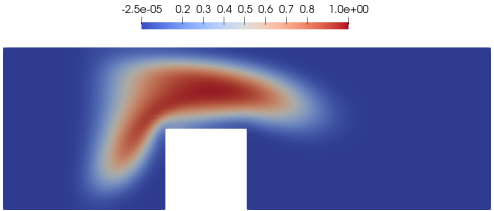
Tutorials New cases and solvers
Bindings Julia `github.com/precice/PreCICE.jl`
Adapters Elmer `github.com/precice/elmer-adapter`
FeniCSX `github.com/precice/fenicsx-adapter`
Two-scale coupling manager `github.com/precice/micro-manager`



New tutorial: channel transport

preCICE News

Tutorials New cases and solvers
Bindings Julia `github.com/precice/PreCICE.jl`
Adapters Elmer `github.com/precice/elmer-adapter`
FeniCSX `github.com/precice/fenicsx-adapter`
Two-scale coupling manager `github.com/precice/micro-manager`



New tutorial: channel transport

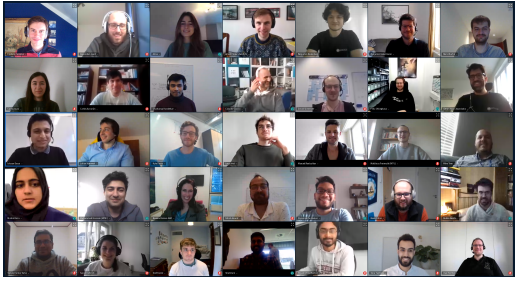
Major news!

Workshop #4 2023

preCICE

Technical University of Munich, Germany

February 13 - 16



preCICE workshop 2022

Major news!

preCICE v2: A sustainable and user-friendly coupling library[version 1; peer review: awaiting peer review]

Open Research Europe 2022

doi.org/10.12688/openreseurope.14445.1

Roadmap

Gerasimos Chourdakis	Extendable and modular system tests **Geometric multi-scale data mapping**
Kyle Davis	More robust and efficient quasi-Newton acceleration
Ishaan Desai	**Adaptive and flexible macro-micro coupling software**
Boris Martin	Cell-based linear interpolation for volumetric coupling
Benjamin Rodenberg	Waveform relaxation for multi-rate coupling and higher-order time stepping
David Schneider	Solver-based data mapping to take advantage of higher-order shape functions **Partition-of-Unity RBF data mapping for very large problems**
Frédéric Simonis	Adaptive-dynamic coupling meshes and run-time remeshing Support multiple SolverInterface instances simultaneously

Summary

Flexible & easy: Couple your own solver with any other, by adding a few lines to your code

Ready: Out-of-the-box support for many solvers

Fast: Fully-parallel, peer-to-peer, designed for HPC

Stable: Implicit coupling, accelerated with quasi-Newton

Multi-coupling: Couple more than two solvers

Free: LGPL3, source on GitHub

+: documented, tested, supported by a growing community

`precice.org`

`github.com/precice`

`@preCICE_org`

`precice.discourse.group` & `gitter.im/precice`