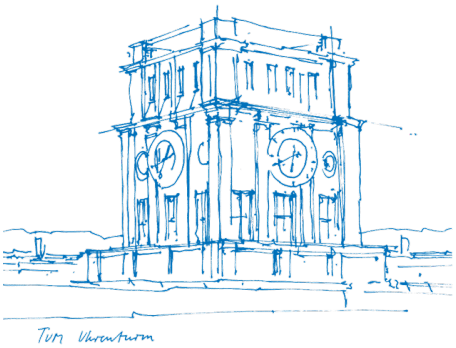
What is new in preCICE?

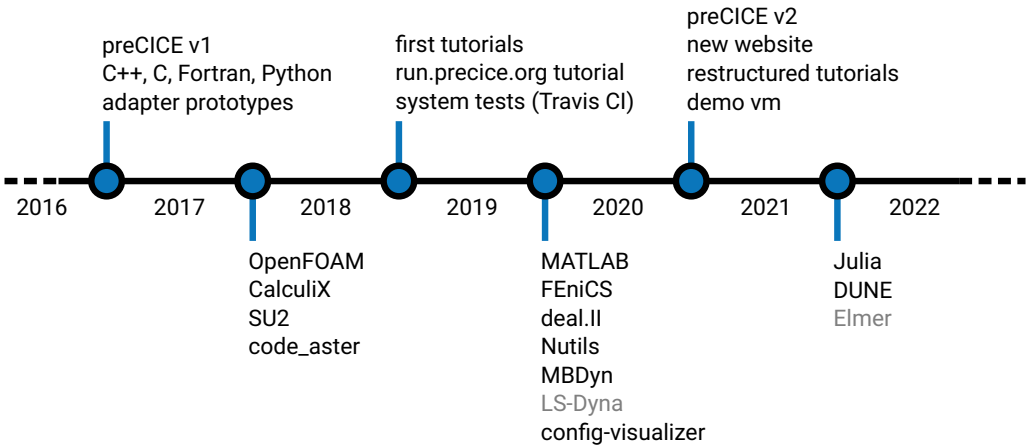
preCICE Workshop 2023

Gerasimos Chourdakis
Technical University of Munich

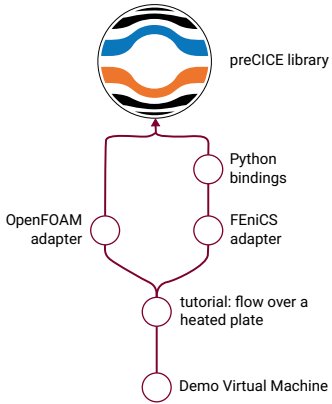
February 14, 2023



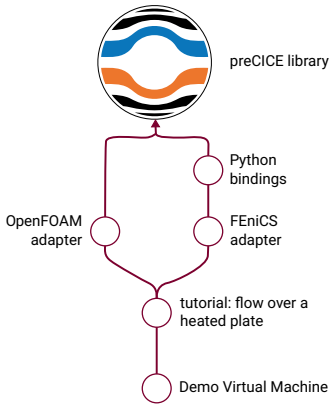
So much going on!

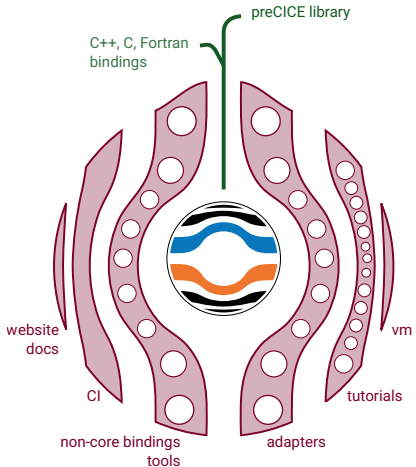


Multiple components, in multiple layers



Multiple components, in multiple layers





preCICE library

preCICE tools at your fingertip

`binprecice` has been renamed and is now useful (again):

- Check your configuration (dry-run):
  `precice-tools check precice-config.xml`
- Find out your preCICE version:
  `precice-tools version`

More to come – See Frédéric's talk tomorrow

Configuration checker

```
$ precice-tools check precice-config.xml
ERROR: Data with name "forces" used by mesh "Solid" is
not defined. Please define a data tag with name="forces".
```

- Checks XML structure: Typos in tags and attribute names
- Checks names of Participants, Meshes, Data
- Cannot check coupling logic
  Is data actually exchanged in a coupling scheme?

Exporting meshes is now cleaner and more versatile

- `<export:vtk />` – legacy VTK
- `<export:vtu />` – modern VTK, UnstructuredGrid
- `<export:vtp />` – modern VTK, PolyData
- `<export:csv />` – post-process in any tool

Debug messages in release builds

```
<log>
  <sink filter="%Severity% >= debug"
        format="---[precice] %ColorizedSeverity% %Message%" enabled="true" />
</log>
```

Debug messages in release builds

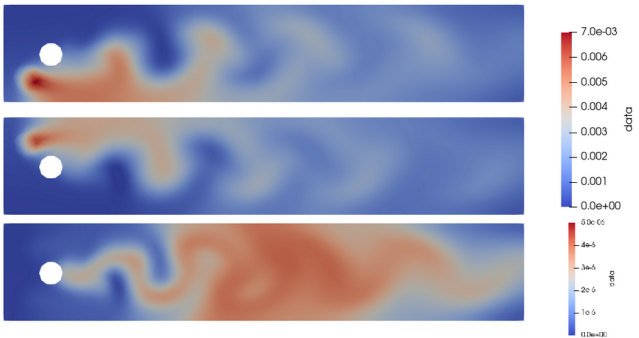
```
<log>
  <sink filter="%Severity% >= debug"
        format="---[precice] %ColorizedSeverity% %Message%" enabled="true" />
</log>
```

```
$ cmake -DCMAKE_BUILD_TYPE=Release \
        -DPRECICE_RELEASE_WITH_DEBUG_LOG=YES ..
```

Slower, but usable.

New mapping: Linear cell interpolation

Nearest projection mapping meets volume coupling



Master's Thesis of Boris Martin

New mapping: Linear cell interpolation

```
<participant name="Flow">
  <use-mesh name="Flow-Mesh" provide="yes" />
  <use-mesh name="Chemical-Mesh" from="Chemical" />
  <write-data name="Velocity" mesh="Flow-Mesh" />
</participant>

<participant name="Chemical">
  <use-mesh name="Chemical-Mesh" provide="yes" />
  <use-mesh name="Flow-Mesh" from="Flow" />
  <read-data name="Velocity" mesh="Chemical-Mesh" />
  <mapping:linear-cell-interpolation
    direction="read"
    from="Flow-Mesh"
    to="Chemical-Mesh"
    constraint="consistent" />
</participant>
```

Skip the mapping: Direct mesh access (since v2.3)

```
<participant name="Dirichlet">
  <use-mesh name="Dirichlet-Mesh" provide="yes" />
  <use-mesh name="Neumann-Mesh" from="Neumann" direct-access="true" />
  <write-data name="Heat-Flux" mesh="Neumann-Mesh" />
  <read-data name="Temperature" mesh="Dirichlet-Mesh" />
</participant>

<participant name="Neumann">
  <use-mesh name="Neumann-Mesh" provide="yes" />
  <use-mesh name="Dirichlet-Mesh" from="Dirichlet" direct-access="true" />
  <write-data name="Temperature" mesh="Dirichlet-Mesh" />
  <read-data name="Heat-Flux" mesh="Neumann-Mesh" />
</participant>
```

```
// Get relevant IDs. Note that "ReceivedMeshname" is not a name of a
// provided mesh, but a mesh defined by another participant. Accessing
// a received mesh directly is disabled in a usual preCICE configuration.
const int otherMeshID = precice.getMeshID("ReceivedMeshName");
const int writeDataID = precice.getDataID("WriteDataName", otherMeshID);

// Define region of interest, where we want to obtain the direct access.
// See also the API documentation of this function for further notes.
precice.setMeshAccessRegion(otherMeshID, boundingBox.data());
```

```
// Get relevant IDs. Note that "ReceivedMeshname" is not a name of a
// provided mesh, but a mesh defined by another participant. Accessing
// a received mesh directly is disabled in a usual preCICE configuration.
const int otherMeshID = precice.getMeshID("ReceivedMeshName");
const int writeDataID = precice.getDataID("WriteDataName", otherMeshID);

// Define region of interest, where we want to obtain the direct access.
// See also the API documentation of this function for further notes.
precice.setMeshAccessRegion(otherMeshID, boundingBox.data());

                                    ...

// Now finally get the data. First allocate memory for the IDs and the
// vertices
std::vector<double> otherSolverVertices(otherMeshSize * dim);
std::vector<int>    ids(otherMeshSize);
// ... and afterwards ask preCICE to fill the vectors
precice.getMeshVerticesAndIDs(otherMeshID,
                              otherMeshSize,
                              ids.data(),
                              otherSolverVertices.data());
```

Skip the mapping: Direct mesh access – now a tutorial!

Partitioned heat conduction (direct access setup)

> **Summary:** This tutorial is a modified version of the “partitioned heat conduction” tutorial showcasing direct mesh access.

Ask David for more details

New version macros

Make your adapter work with multiple preCICE versions:

```
#ifndef PRECICE_VERSION_GREATER_EQUAL
   // compatibility with older versions
   #define PRECICE_VERSION_GREATER_EQUAL(x,y,z) FALSE
#endif

#if PRECICE_VERSION_GREATER_EQUAL(2,5,0)
  setMeshTetrahedron(...)
#endif
```

preCICE ecosystem

preCICE Distribution v2211.0

Citable collection of packages working together.

(v2202.0, v2211.0 DOI pending)

New versions

As always, we keep updating components. In this distribution release, find:

- OpenFOAM adapter 3 v1.2.0, with support for more solvers and features (release notes).
- CalculiX adapter v2.20.0, with many bugfixes and updates for CalculiX 2.20 (release notes).
- FEniCS adapter v1.4.0, with more capablities for providing mesh connectivity (release notes).
- Further updates to the deal.II adapter 1 and to more components.

v2211.0

preCICE distribution v2211.0 (publishing to DaRUS pending)

This is the last release that works with preCICE v2.x.

It comprises the following components:

- preCICE: v2.5.0
- Tools:
  - ASTE: v3.0.0 (new)
  - config-visualizer: commit 60f2165 (new)
  - ci-images: commit f48c7e8 (new)
- Bindings:
  - Fortran module: commit 9e3f405
  - Julia bindings: v2.5.0 (new and experimental)
  - Matlab bindings: v2.5.0.0
  - Python bindings: v2.5.0.1
- Adapters:
  - CalculiX adapter: v2.20.0
  - code_aster adapter: commit ce995e0 (same as in v2202.0)
  - deal.II adapter: dbb25be
  - DUNE adapter: commit 5f2364d (same as in v2202.0, experimental)
  - FEniCS adapter: v1.4.0
  - OpenFOAM adapter: v1.2.0
  - SU2 adapter: ab84387 (same as in v2202.0)
- Tutorials: v202211.0
- vm: v202211.0.0
- Website and documentation: v202211.0.0, libprecice2_2.2.5_docs_v202211.0.0.pdf

New tool: ASTE

Artificial Solver Testing Environment (ASTE)

> **Summary:** ASTE is a lightweight wrapper around the preCICE API, which allows emulating participants to investigate simulation setups.

Command line interface

After the installation procedure, the following executables are available

- `precice-aste-run`: core module interfacing with preCICE
- `precice-aste-evaluate`: python tool to compute and store data on mesh files
- `precice-aste-partition`: python tool to partition a single mesh file into several ones for parallel runs
- `precice-aste-join`: python tool to join several mesh files into a single mesh file for parallel runs.

New tool: ASTE (in the course pt4)

Task 4: Replay Generator with ASTE

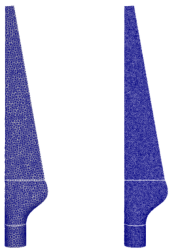
The second use case of ASTE is to replay single participants. This can be helpful to avoid re-computations of expensive parts of coupled simulations.

```
{
  "participant": "Generator",
  "startdt": "1",
  "meshes": [
    {
      "mesh": "Generator-Mesh",
      "meshfileprefix": "./precice-output/Generator-Mesh-Generator",
      "write-data": {
        "scalar": ["Data"]
      }
    }
  ],
  "precice-config": "../precice-config.xml"
}
```

New tool: ASTE (in the tutorials)

ASTE (artificial solver testing environment) wind turbine blade tutorial

> **Summary:** This tutorial is an example case for ASTE, where we investigate different preCICE mappings using ASTE.



Julia bindings getting in shape

PreCICE.jl

license LGPL-3.0

This package provides Julia language bindings for the C++ library preCICE. It is a Julia package that wraps the API of preCICE.

Note that the first two digits of the version number of the bindings indicate the preCICE major and minor version that the bindings support. The last digit represents the version of the bindings. Example: `v2.5.1` and `v2.5.2` of the bindings represent versions `1` and `2` of the bindings that are compatible with preCICE `v2.5.0`.

Contributors 4

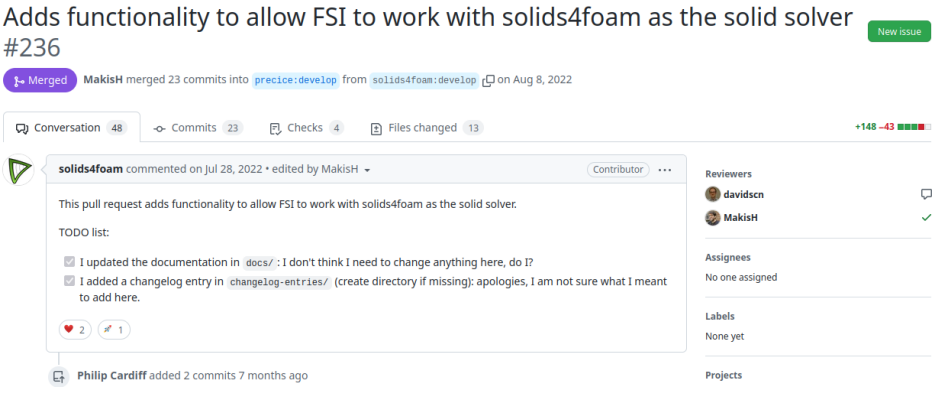

pavelkharitenko


erikscheurer


IshaanDesai Ishaan Desai


kursatyurt Kürşat Yurt

The OpenFOAM adapter now works with solids4foam

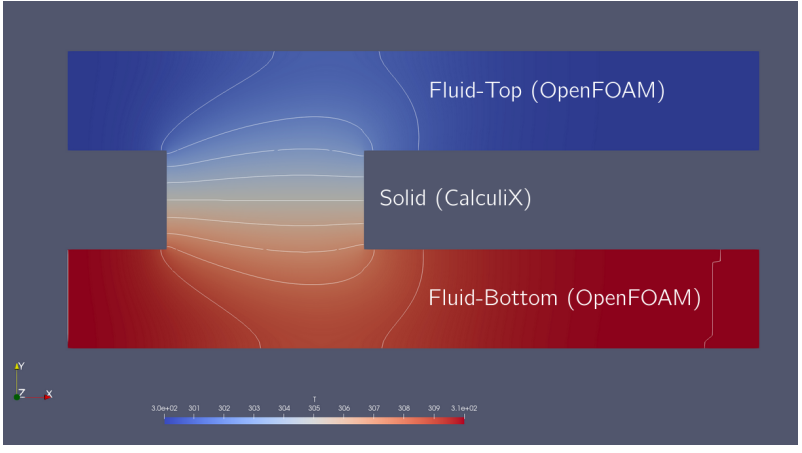


More on Philip Cardiff's talk later today

The OpenFOAM adapter now works with solids4foam

```
~/repos/precice/tutorials/perpendicular-flap [(v202211.0)]$ tree -L 1
.
├── clean-tutorial.sh -> ../tools/clean-tutorial-base.sh
├── fluid-nutils
├── fluid-openfoam
├── fluid-su2
├── images
├── plot-all-displacements.sh
├── plot-displacement.sh
├── precice-config.xml
├── precice-run
├── README.md
├── solid-calculix
├── solid-dealii
├── solid-dune
├── solid-fenics
├── solid-openfoam
└── solid-solids4foam
```

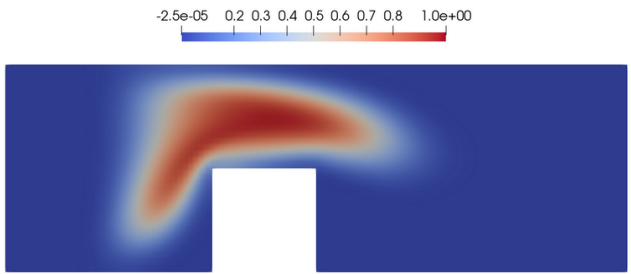
New tutorial: Simplified heat exchanger



New tutorial: Channel transport (volume coupling)

Channel transport

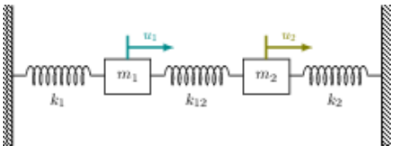
Summary: A CFD problem is coupled to a transport (of, e.g., a chemistry species) in a uni-directional way.



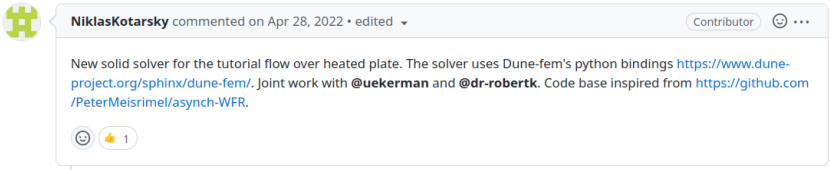
New tutorial: Oscillator

Setup

This tutorial solves a simple mass-spring oscillator with two masses and three springs. The system is cut at the middle spring and solved in a partitioned fashion:



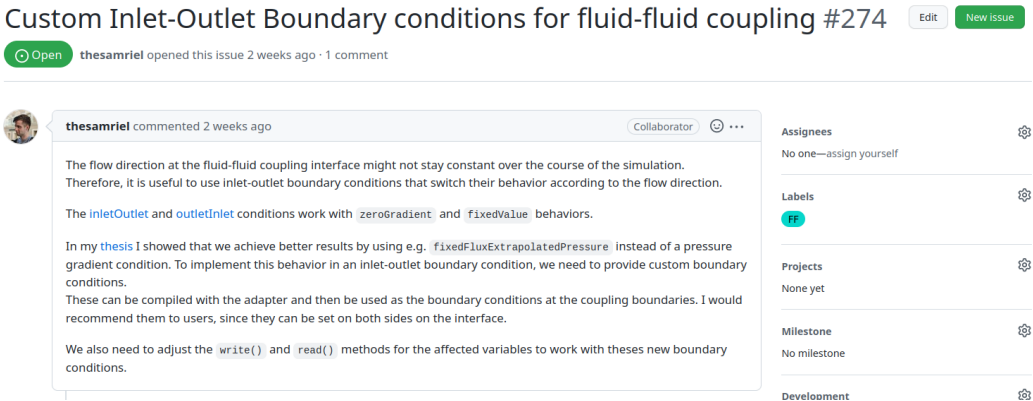
New tutorial case: DUNE-FEM



Current work

More on other talks today/tomorrow

The OpenFOAM adapter will soon offer better flow coupling



Custom Inlet-Outlet Boundary conditions for fluid-fluid coupling #274

Edit | New issue

Open **thesamriel** opened this issue 2 weeks ago · 1 comment

thesamriel commented 2 weeks ago — Collaborator

The flow direction at the fluid-fluid coupling interface might not stay constant over the course of the simulation.
Therefore, it is useful to use inlet-outlet boundary conditions that switch their behavior according to the flow direction.

The inletOutlet and outletInlet conditions work with `zeroGradient` and `fixedValue` behaviors.

In my thesis I showed that we achieve better results by using e.g. `fixedFluxExtrapolatedPressure` instead of a pressure gradient condition. To implement this behavior in an inlet-outlet boundary condition, we need to provide custom boundary conditions.
These can be compiled with the adapter and then be used as the boundary conditions at the coupling boundaries. I would recommend them to users, since they can be set on both sides on the interface.

We also need to adjust the `write()` and `read()` methods for the affected variables to work with theses new boundary conditions.

Assignees: No one—assign yourself

Labels: FF

Projects: None yet

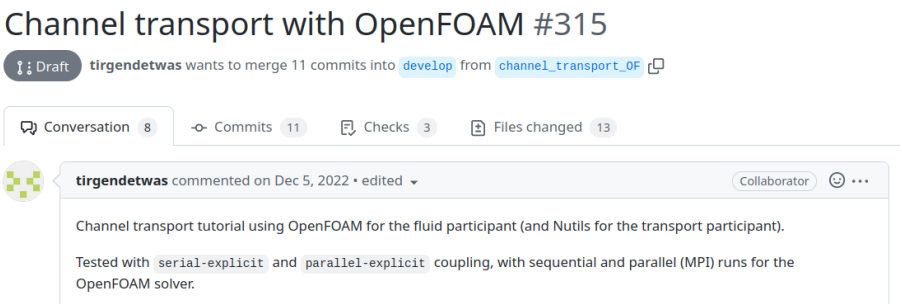
Milestone: No milestone

Development

Master's thesis of Markus Mühlhäußer – See talk tomorrow

The OpenFOAM adapter will soon support volume coupling

The OpenFOAM adapter will soon support volume coupling



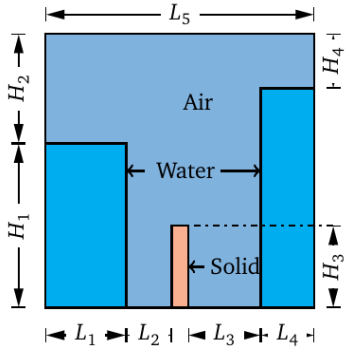
Student project (IDP) of Tina Vladimirova

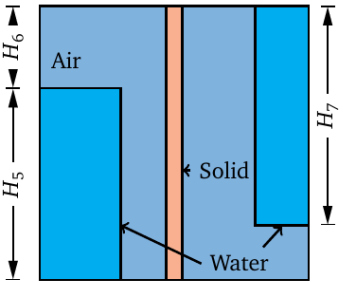
The SU2 adapter will soon be up-to-date

Community contribution (whoever you are, thank you!)

Upcoming tutorial cases: FSI with two-phase flows

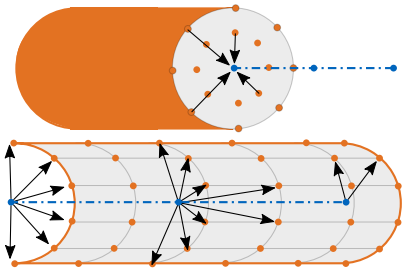
Upcoming tutorial cases: FSI with two-phase flows





Part of Kyle Davis' (defended!) PhD

Geometric multiscale coupling



RadialGeoMultiscale #1552

Draft **ezonta** wants to merge 6 commits into `precice:develop` from `ezonta:RadialGeoMultiscale`

Conversation 0 · Commits 6 · Checks 20 · Files changed 10

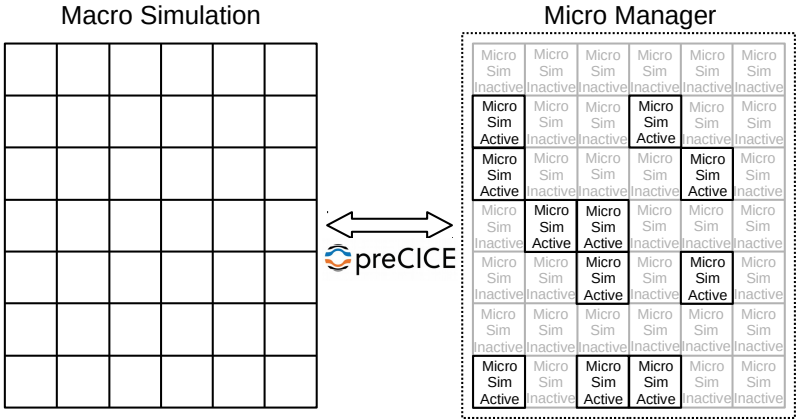
ezonta commented 3 weeks ago — First-time contributor

Main changes of this PR

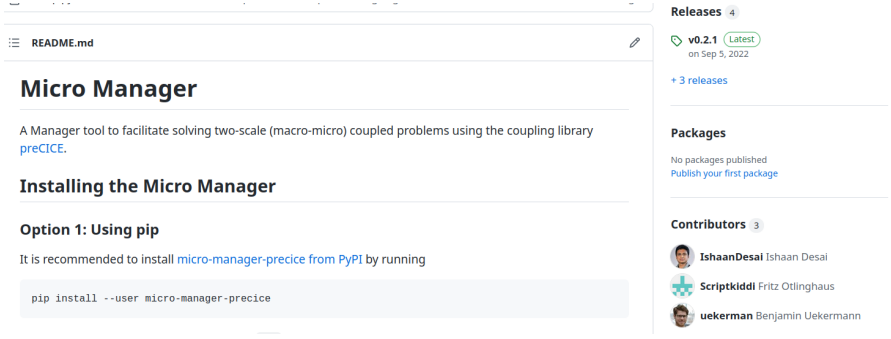
Adds radial geometric multiscale mapping to preCICE. Builds on top of AxialGeoMultiscaleMapping.

We are getting there (thanks, Elia Zonta!)

Coupling macroscopic and microscopic simulations

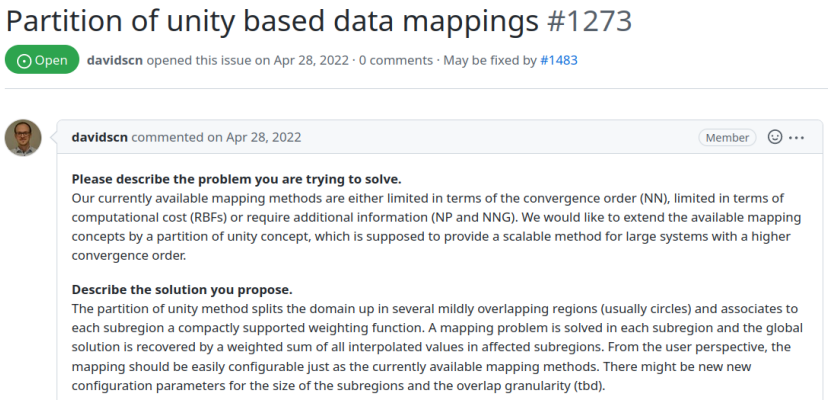


Coupling macroscopic and microscopic simulations



More on Ishaan's talk later today.

Much faster RBF mapping



Partition of unity based data mappings #1273

Open **davidscn** opened this issue on Apr 28, 2022 · 0 comments · May be fixed by #1483

davidscn commented on Apr 28, 2022 — Member

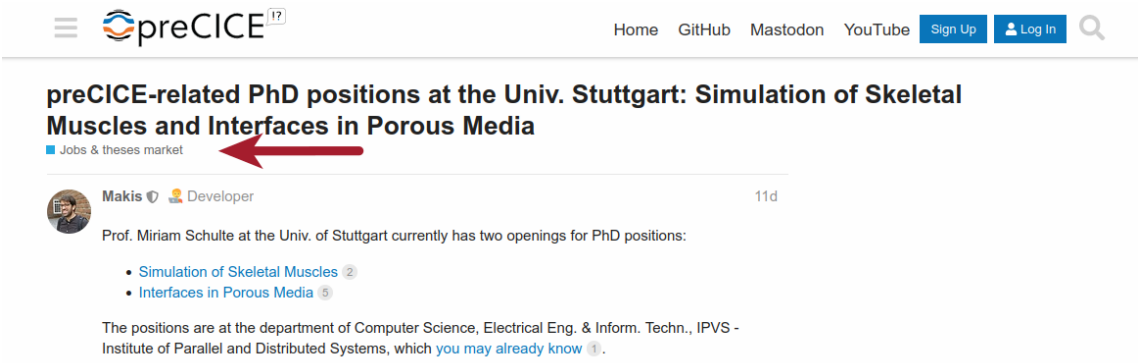
Please describe the problem you are trying to solve.
Our currently available mapping methods are either limited in terms of the convergence order (NN), limited in terms of computational cost (RBFs) or require additional information (NP and NNG). We would like to extend the available mapping concepts by a partition of unity concept, which is supposed to provide a scalable method for large systems with a higher convergence order.

Describe the solution you propose.
The partition of unity method splits the domain up in several mildly overlapping regions (usually circles) and associates to each subregion a compactly supported weighting function. A mapping problem is solved in each subregion and the global solution is recovered by a weighted sum of all interpolated values in affected subregions. From the user perspective, the mapping should be easily configurable just as the currently available mapping methods. There might be new new configuration parameters for the size of the subregions and the overlap granularity (tbd).

More on David's talk later today.

Community

Forum: new job / theses announcement board



Forum: community projects (reminder)

Community projects | all tags | Latest | Top

Topic		Replies	Views	Activity
About the Community projects category inactive Share your simulation cases for everybody to admire and try. Did you make a nice simulation case that you want to share with the rest of the community so that everybody can run it and build upon this? Please open a thre… read more		0	290	Jun '21
Turek-Hron FSI 2 Benchmark calculix fsi inactive openfoam	U D A	2	719	Oct '22
FSI coupling between OpenFOAM and MBDyn of a cycloidal rotor fsi mbdyn openfoam		2	343	Aug '22
Coupled Phase-Field Brittle Fracture Simulation fenics nutils		0	361	Apr '21

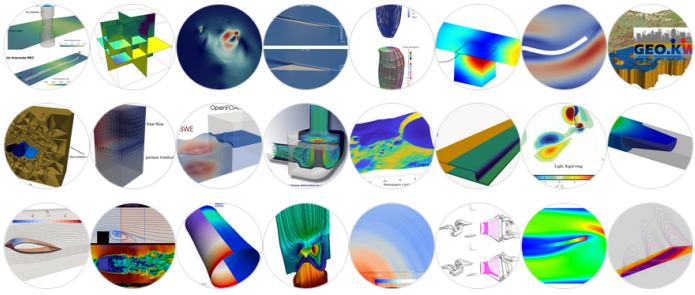
Website: community stories (yours?)

Community

- Overview & news
- preCICE Workshop 2023
- Coupled Problems 2023
- Support preCICE
- Stories
- Contributors
- Contribute to preCICE
- Community channels
- Past events

How our community is using preCICE

We have compiled a community showcase with several of the most impressive preCICE implementations. The range of application fields is diverse, including e.g. simulating the temperatures on the moon, the biomechanics of the heart valve, modelling wind in urban areas, and simluating high impact loads on structures. You can find many more use cases on the projects page.



Do you use preCICE? Tell us your story and will be happy to feature it here!

Help us maintain preCICE, a tool you rely upon

You support preCICE – preCICE supports you

As the developers of preCICE, we enjoy supporting our academic and industry users, but due to the growing demand, we are not able to service all support requests any more. We are introducing **preCICE support** as a way to contribute to sustainable open-source software development and to ensure preCICE is developed and maintained in the future. Supporting preCICE comes with several benefits:

- **Increased success rate** for your research proposal (include preCICE as a partner, software provider, or sub-contractor)
- **Priority support** and direct access to the preCICE developers
- Private, on-site support and bespoke training

We offer different levels of support for industry and academia.

Support preCICE - see options ›

preCICE v2 paper fully reviewed

preCICE v2 paper fully reviewed

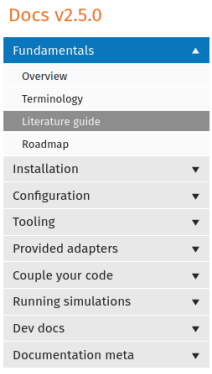
preCICE v2: A sustainable and user-friendly coupling library

Gerasimos Chourdakis, Kyle Davis, Benjamin Rodenberg, Miriam Schulte, Frédéric Simonis, Benjamin Uekermann et al., **2022**. In Open Research Europe, 2:51.

Publisher's Site Download BibTeX

Citations of preCICE v1 paper (222) Literature guide

Updated literature guide



Literature guide

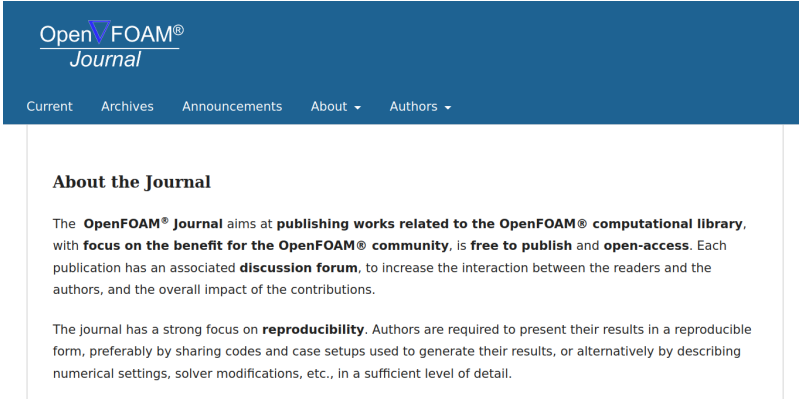
Summary: A guide to the main reference literature for each component and feature of preCICE

Table of Contents

Edit me

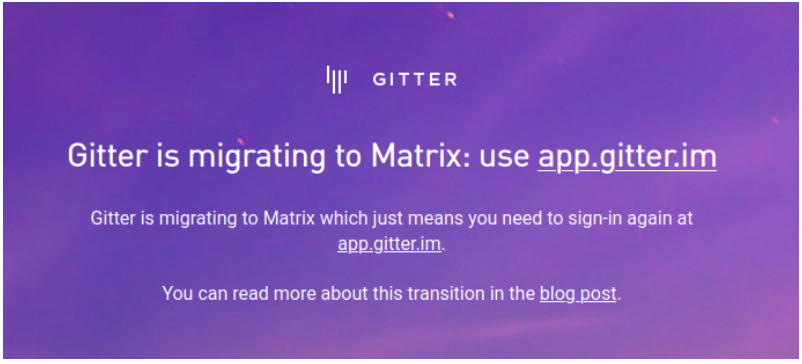
The literature one can read to understand different aspects of preCICE may feel a bit overwhelming. This page aims to give some starting points and citation guidelines.

Watch out for related research



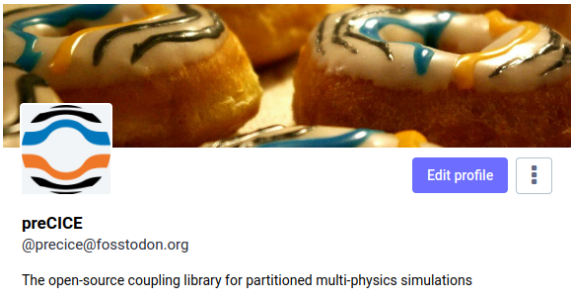
Our developers chatroom is changing

Gitter is merging with Matrix (and it will be confusing for a bit):



preCICE on social media

Now on Mastodon (`@precice@fosstodon.org`):



preCICE goes to Crete (finally!)

IS14 - Multi-Physics and Multi-Scale Simulations with the Coupling Library preCICE

Organized by: ***G. Chourdakis (Technische Universität München, Germany)**** *and B. Uekermann (University of Stuttgart, Germany)*

Abstract submission deadline extended to Feb 28,
optional paper submission Mar 31

Wrapping up

Don’t miss

Developer talks: Updates on:

- Macro-micro coupling – Ishaan Desai
- Data mapping – David Schneider
- Time interpolation – Benjamin Rodenberg
- preCICE v3 – Frédéric Simonis

World Café: Tell us where to go next

2023 resolution

be part of the “What’s new in preCICE” list
in the preCICE Workshop 2024.

(or simply manage to catch up on all these updates)