15th OpenFOAM Workshop - June 22, 2020 - Technical Session I-D

A flexible and preCICE solver coupling ecosystem

Gerasimos Chourdakis, Technical University of Munich
chourdak@in.tum.de

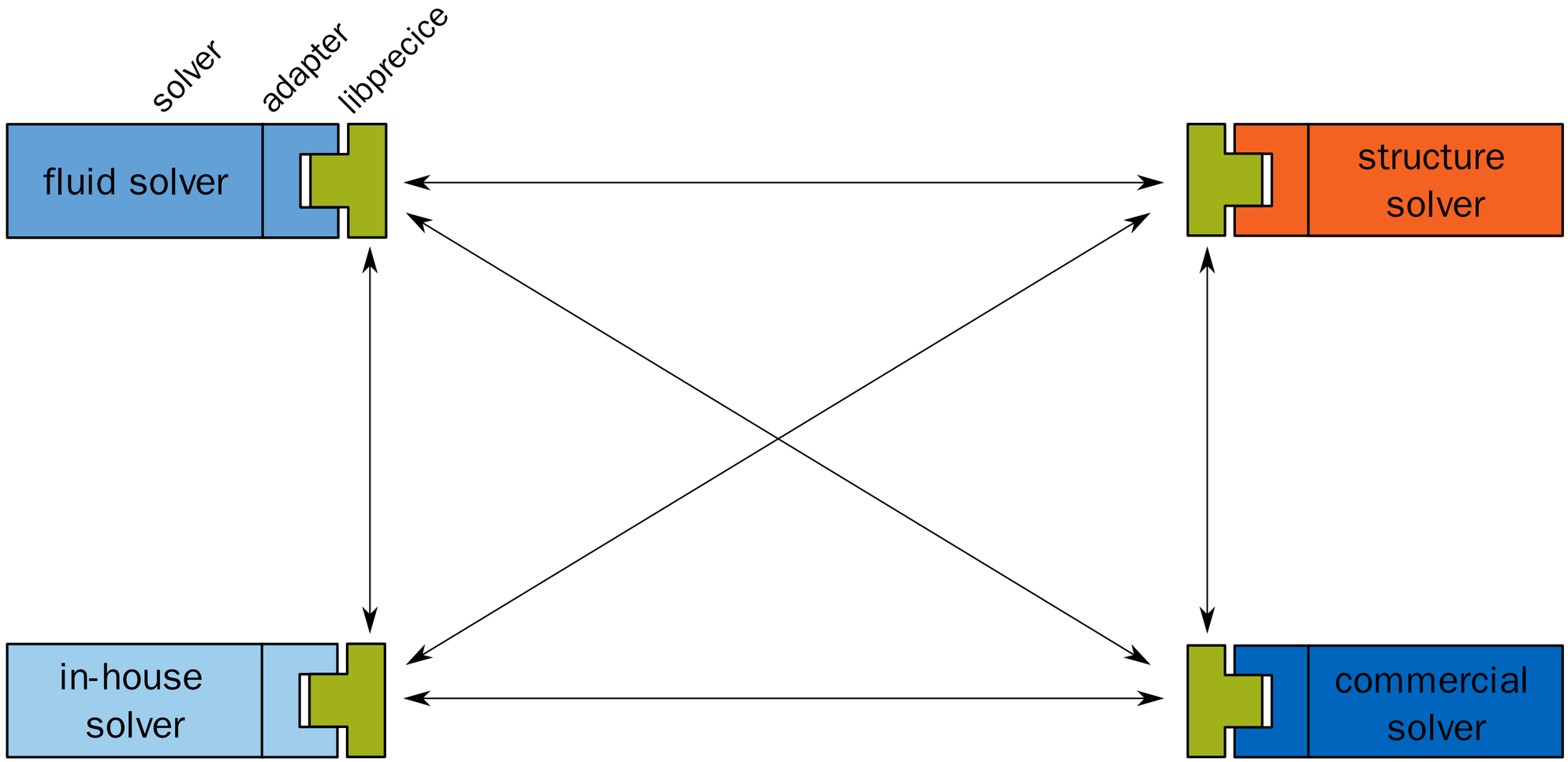
Benjamin Uekermann, Eindhoven University of Technology
b.w.uekermann@tue.nl

Find these slides on GitHub

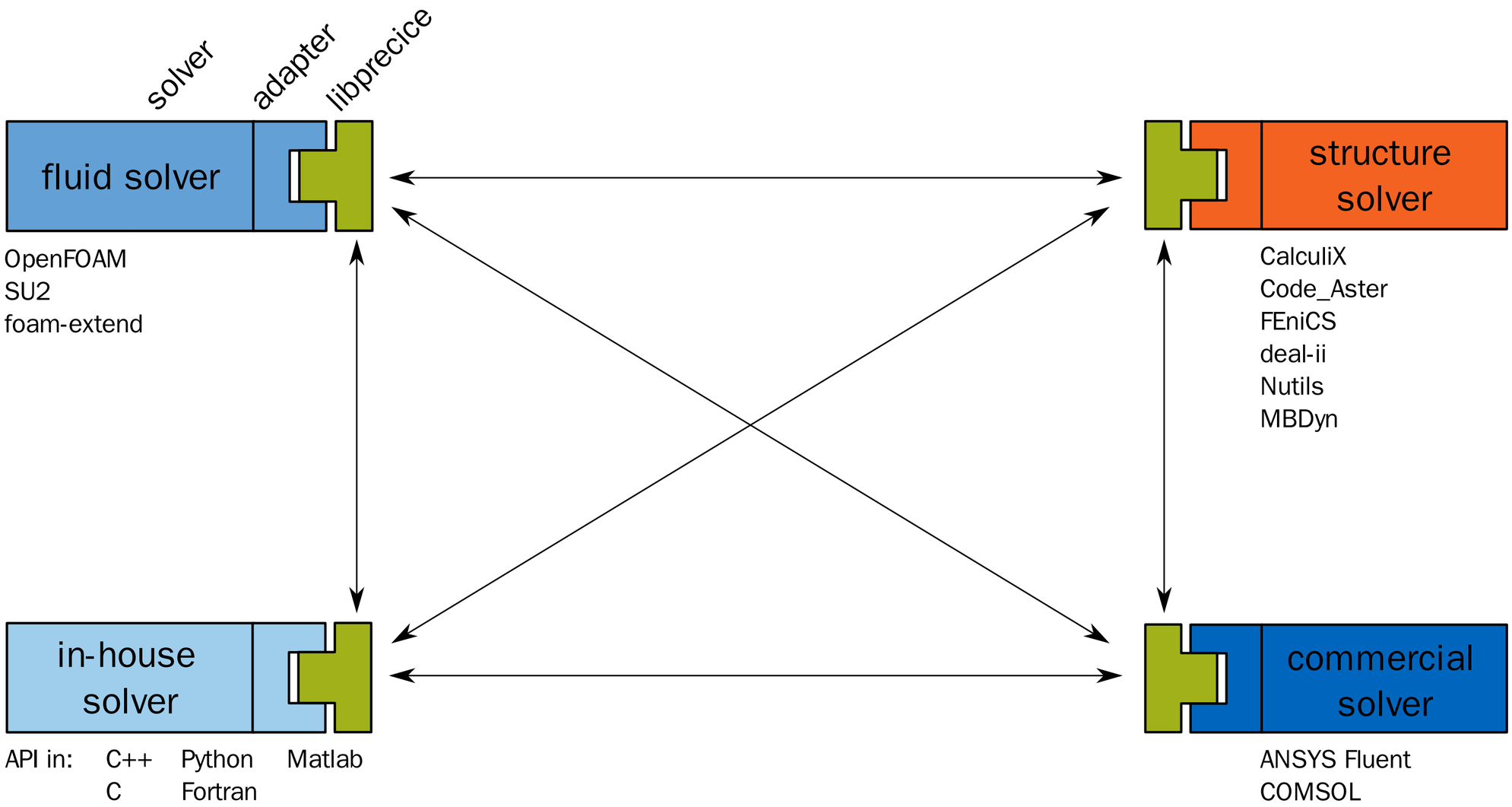
https://github.com/MakisH/ofw15-slides

The Big Picture

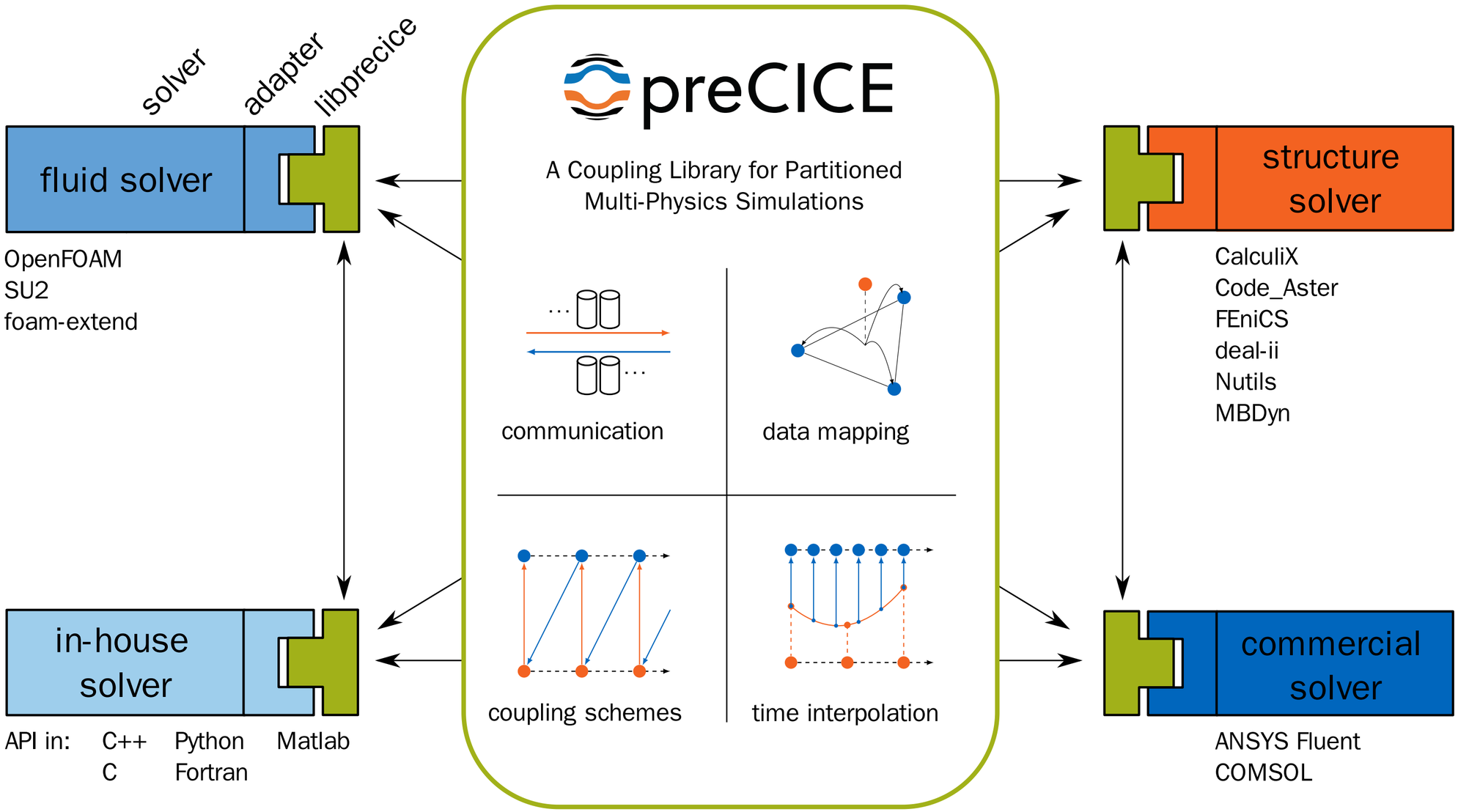
The Big Picture



The Big Picture



The Big Picture



News

preCICE v2.0

- Simplified config & API
- XML reference & visualizer
- Faster initialization
- Spack / Debian / AUR packages
- Better building & testing
- xSDK member
- Better Python bindings
- New Matlab bindings

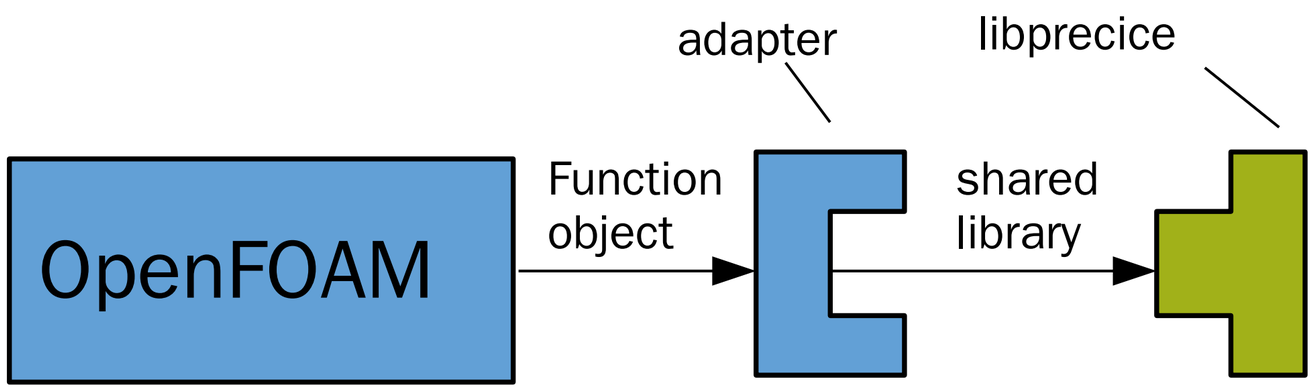
Upgrade guide in the wiki

Other news

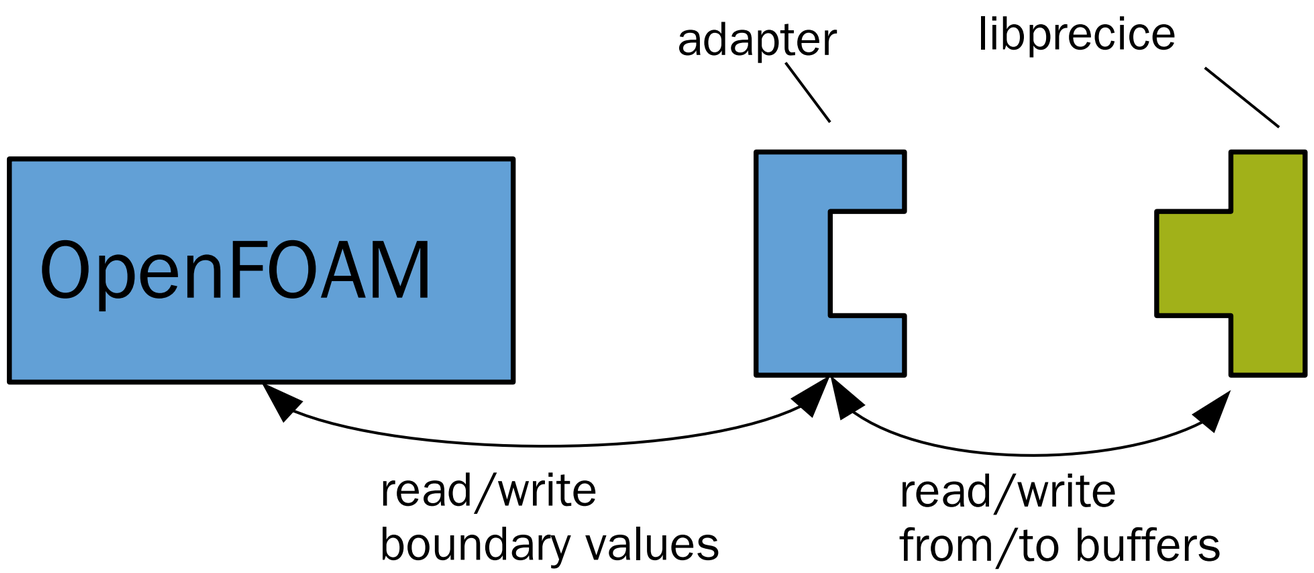
- deal.ii adapter
  - new non-linear example for FSI
- FEniCS adapter
  - new example for FSI
- code_aster adapter
  - revived for code_aster 14 and preCICE v2

The OpenFOAM adapter

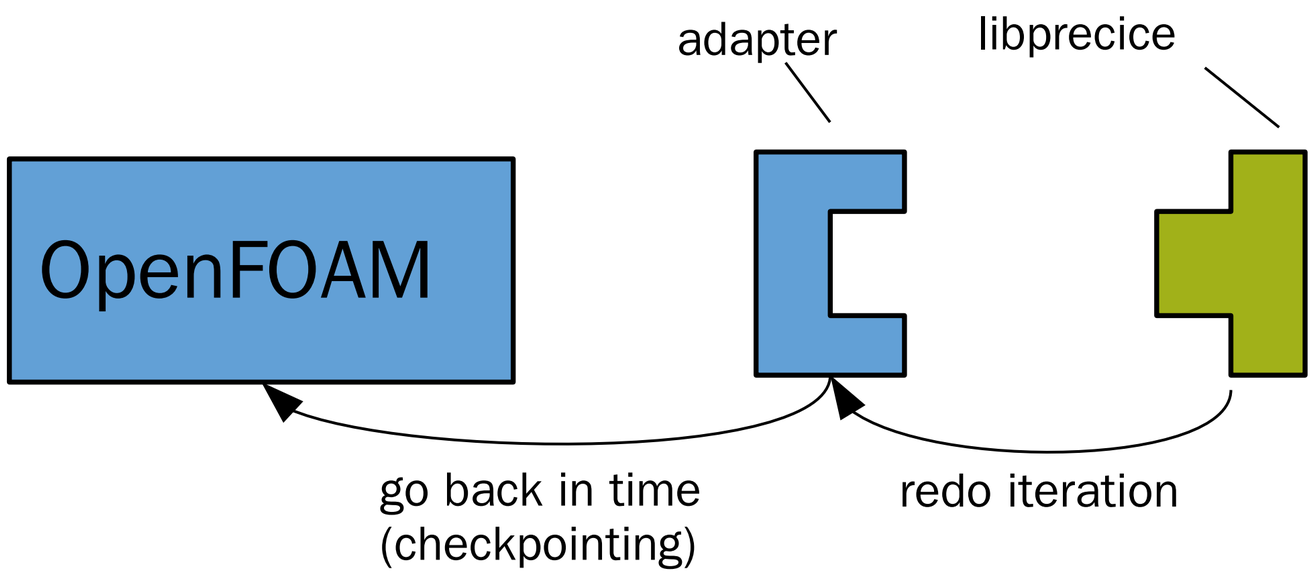
What does the adapter do?



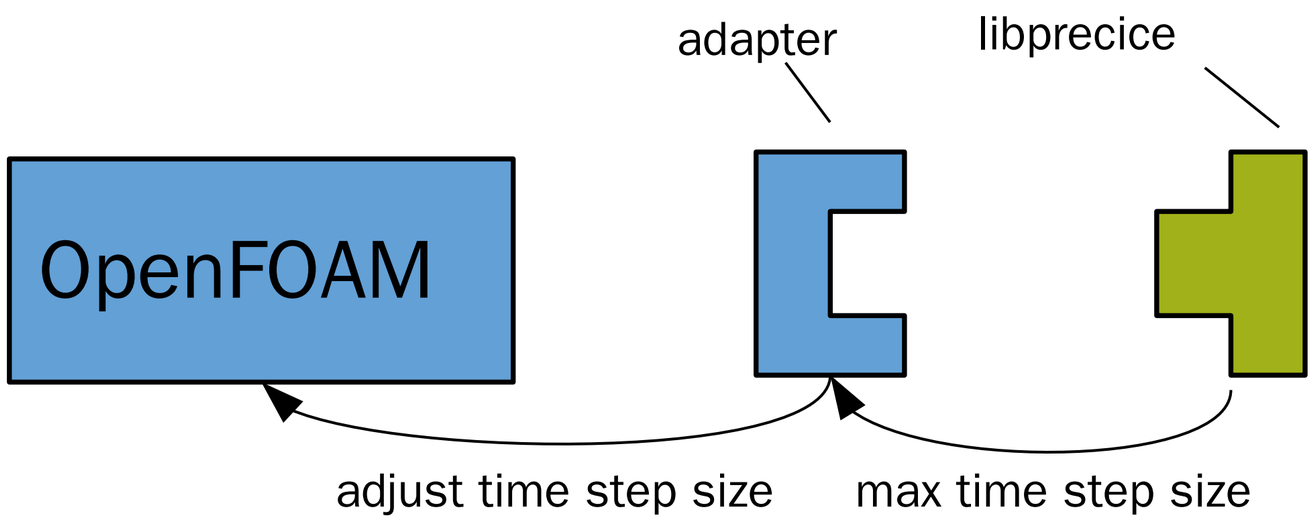
What does the adapter do?



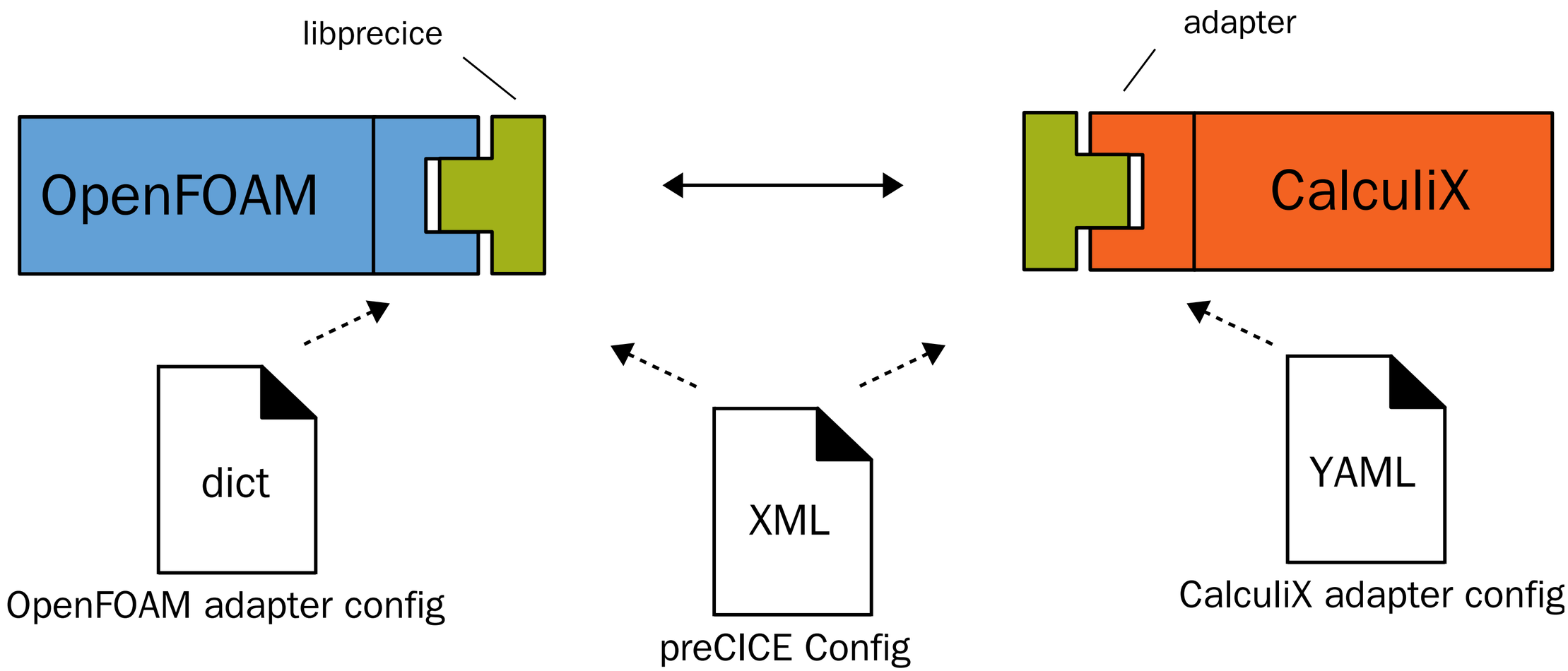
What does the adapter do?



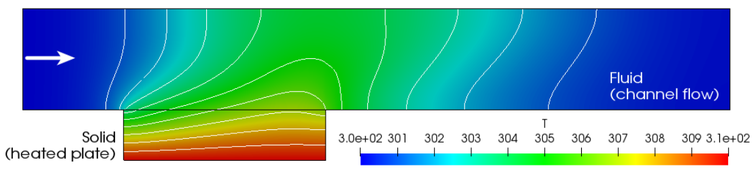
What does the adapter do?

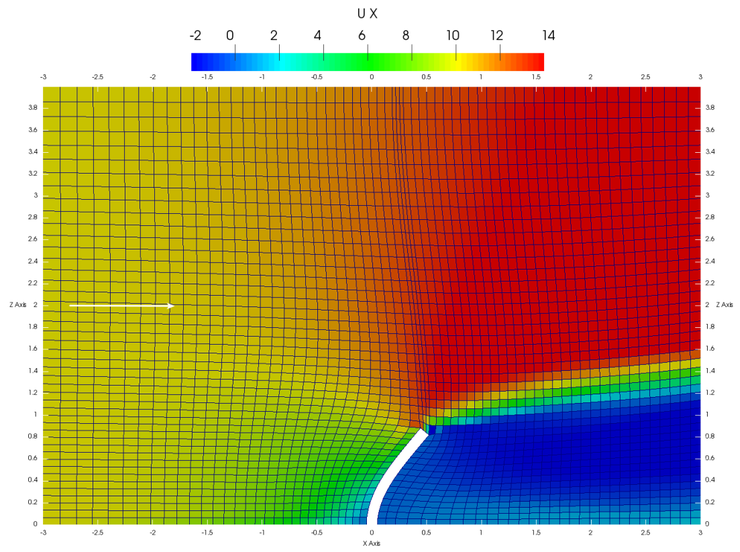


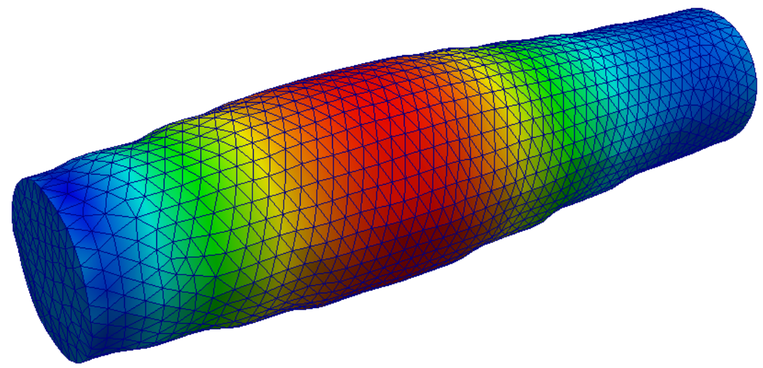
Configuration

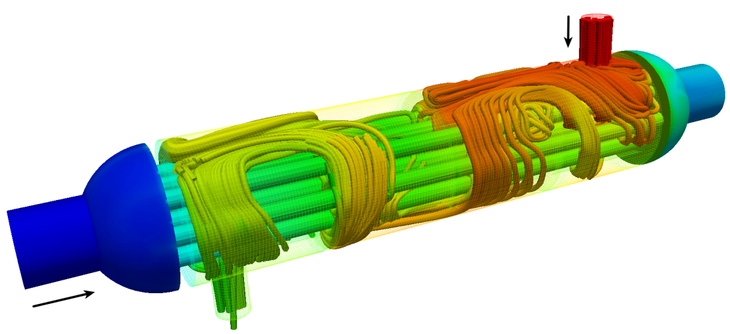


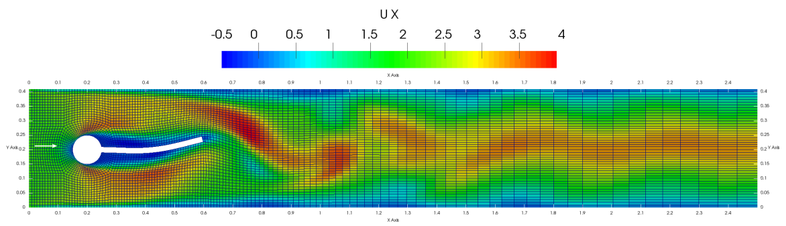
Tutorials

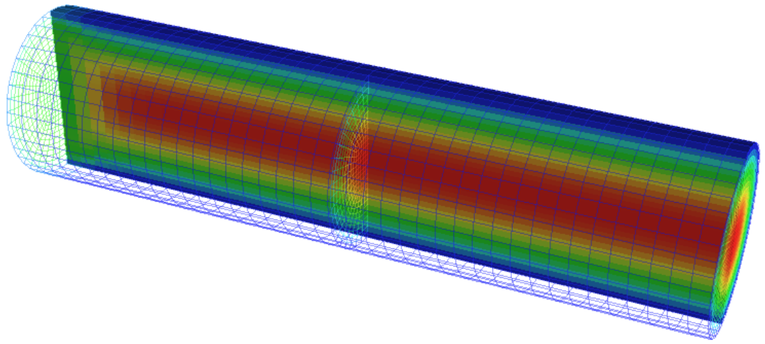












Tutorials using OpenFOAM in our wiki.

```
---[preciceAdapter] [DEBUG]     connectivity : 0
---[preciceAdapter] [DEBUG]     patches      :
---[preciceAdapter] [DEBUG]       - flap
---[preciceAdapter] [DEBUG]     writeData    :
---[preciceAdapter] [DEBUG]       - Forces0
---[preciceAdapter] [DEBUG]     readData     :
---[preciceAdapter] [DEBUG]   - mesh         : Fluid-Mesh-Nodes
---[preciceAdapter] [DEBUG]     locations    : faceNodes
---[preciceAdapter] [DEBUG]     connectivity : 0
---[preciceAdapter] [DEBUG]     patches      :
---[preciceAdapter] [DEBUG]       - flap
---[preciceAdapter] [DEBUG]     writeData    :
---[preciceAdapter] [DEBUG]     readData     :
---[preciceAdapter] [DEBUG]       - Displacements0
---[preciceAdapter] [DEBUG] Configuring the FSI module...
---[preciceAdapter] [DEBUG]     user-defined solver type :
---[preciceAdapter] [DEBUG]     pointDisplacement field name : pointDisplacem
ent
---[preciceAdapter] [DEBUG] Unknown solver type. Determining the solver type.
..
---[preciceAdapter] [DEBUG] Automatically determined solver type : incompress
ible
---[preciceAdapter] [DEBUG] Checking the timestep type (fixed vs adjustabl
..
---[preciceAdapter] [DEBUG]   Timestep type: fixed.
---[preciceAdapter] [DEBUG] Creating the preCICE solver interface...
---[preciceAdapter] [DEBUG]   Number of processes: 1
---[preciceAdapter] [DEBUG]   MPI rank: 0
---[precice]  This is preCICE version 2.0.2
---[precice]  Revision info: v2.0.2
---[precice]  Configuring preCICE with configuration: "precice-config.xml"
---[preciceAdapter] [DEBUG]   preCICE solver interface was created.
---[preciceAdapter] [DEBUG] Creating interfaces...
---[preciceAdapter] [DEBUG] Number of face centres: 33
---[preciceAdapter] [DEBUG] Interface created on mesh Fluid-Mesh-Faces
---[preciceAdapter] [DEBUG] Adding coupling data writers...
---[preciceAdapter] [DEBUG] Added writer: Force.
---[preciceAdapter] [DEBUG] Adding coupling data readers...
---[preciceAdapter] [DEBUG] Number of face nodes: 68
---[preciceAdapter] [DEBUG] Interface created on mesh Fluid-Mesh-Nodes
---[preciceAdapter] [DEBUG] Adding coupling data writers...
---[preciceAdapter] [DEBUG] Adding coupling data readers...
---[preciceAdapter] [DEBUG] Added reader: Displacement.
---[preciceAdapter] [DEBUG] Initalizing the preCICE solver interface...
---[precice]  Setting up master communication to coupling partner/s
```

```
(base) ~/git/hub/tutorials/FSI/flap_perp/OpenFOAM-CalculiX/Solid [master]$ ls
flap.12d  flap.inp  mesh.sh  post_flap.fbd  pre_flap.fbd
(base) ~/git/hub/tutorials/FSI/flap_perp/OpenFOAM-CalculiX/Solid [master]$ cg
x -bg pre_flap.fbd
on a Linux machine, nodename Magdalena, release 4.15.0-106-generic, version #
107~16.04.1-Ubuntu SMP Thu Jun 4 15:40:05 UTC 2020, machine x86_64
parameters:3 arguments:2

pre_flap.fbd opened
 reading file
 #!/bin/bash
 #For the beam elements
 please wait for 'ready'
 makeSurfaces
 getElemNormalen
 realloc_colNr
 add the faces
 updateDispLists
 delSet
 ready
 please wait for 'ready'
 write abaqus data
 file all.msh opened
 ready
 please wait for 'ready'
 write file: fix1.nam
 ready
 ready
 please wait for 'ready'
 write file: surface.nam
 ready
 ready
 # #rot z
 #rot r 50
 #rot u 50
 # Add thing here for the boundary condition
 # sys echo
 # quit
 # plus la all
 # plus pa all
 done

(base) ~/git/hub/tutorials/FSI/flap_perp/OpenFOAM-CalculiX/Solid [master]$ cd
 ..
(base) ~/git/hub/tutorials/FSI/flap_perp/OpenFOAM-CalculiX [master]$ ccx_preC
ICE -i Solid/flap -precice-participant Calculix
```

```
[10] 0:bash*                                                    "Magdalena" 09:22 22-Jun-20
```

New: **system/preciceDict**

```
preciceConfig "precice-config.xml";

participant Fluid;

modules (FSI);

interfaces
{
  Interface1
  {
    mesh                Fluid-Mesh-Faces;
    patches             (flap);
    locations           faceCenters;

    readData
    (
    );
```

No need for yaml-cpp anymore!

preciceDict - implementation

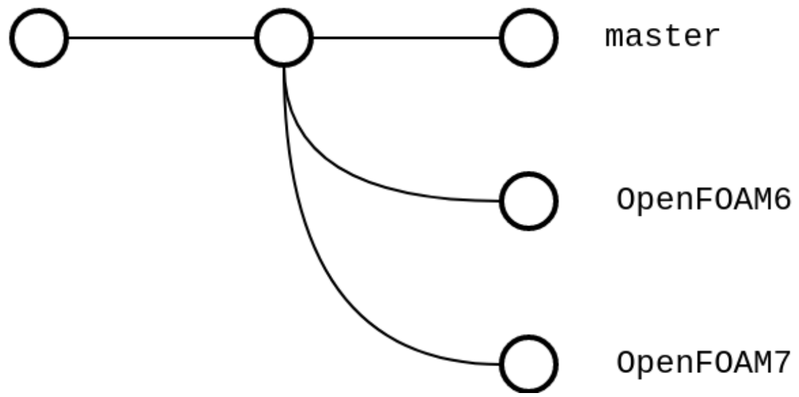
```
IOdictionary preciceDict
(
    IOobject
    (
        "preciceDict",
        runTime_.system(),
        mesh_,
        IOobject::MUST_READ_IF_MODIFIED,
        IOobject::NO_WRITE
    )
);

// lookupType<T>("name") is deprecated in openfoam.com since v1812,
// which recommends get<T>("name") instead.
preciceConfigFilename_ = preciceDict.lookupType<fileName>("preciceConfig'

// ...
```

Many thanks to Mark Olesen (ESI) for the hints

Before: Special cases in branches



Multiple Git branches, differing in small details

New: One source to rule them all

```
// Adapter.C

// Version-specific code with possible variants:
// - const_cast<Time&>(runTime_).setDeltaT(timestepSolver_, false);
// - const_cast<Time&>(runTime_).setDeltaTNoAdjust(timestepSolver_);
#include "version-specific/setDeltaT.H"
```

```
# Make/options

EXE_INC = \
    # ...
    -Ivariants/$(WM_PROJECT_VERSION) \
    # e.g. -Ivariants/v1912
```

New: One source to rule them all

```
variants/
├── 4.0
│   └── version-specific
│       └── setDeltaT.H
├── 4.1 -> 4.0
├── 5.0
│   └── version-specific
│       └── setDeltaT.H -> ../../4.0/version-specific/setDeltaT.H
├── 6
│   └── version-specific
│       └── setDeltaT.H
├── 7
│   └── version-specific
│       └── setDeltaT.H -> ../../6/version-specific/setDeltaT.H
├── v1712
│   └── version-specific
│       └── setDeltaT.H -> ../../4.0/version-specific/setDeltaT.H
```

New: One source to rule them all

```
variants/
├── 4.0
│   └── version-specific
│       ├── directory_type.H
│       ├── init.H
│       └── setDeltaT.H
├── 4.1 -> 4.0
├── 5.0
│   └── version-specific
│       ├── directory_type.H -> ../../4.0/version-specific/directory_type
│       ├── init.H
│       └── setDeltaT.H -> ../../4.0/version-specific/setDeltaT.H
├── 6
│   └── version-specific
│       ├── directory_type.H -> ../../5.0/version-specific/directory_type
│       ├── init.H
│       └── setDeltaT.H
├── 7
```

New: Pressure-based solver type selection

```
dimensionSet pressureDimensionsCompressible(1, -1, -2, 0, 0, 0, 0);
dimensionSet pressureDimensionsIncompressible(0, 2, -2, 0, 0, 0, 0);

if (mesh_.foundObject<volScalarField>("p"))
{
  volScalarField p_ = mesh_.lookupObject<volScalarField>("p");

  if (p_.dimensions() == pressureDimensionsCompressible)
    solverType = "compressible";
  else if (p_.dimensions() == pressureDimensionsIncompressible)
    solverType = "incompressible";
}
```

Thanks to David Schneider (TUM) for adding this.

New: Write stresses (FSI)

- Before: write forces, read displacements
- Now: write forces or stresses, read displacements
- No need for conservative mapping!

Thanks to David Schneider (TUM) for adding this.

Upcoming: Unit & integration tests

- Already CI with system tests
- Wish: Test specific parts of the adapter
  - Unit tests with Catch2
  - Integration tests with Google Test
  - Other ideas?

Prototype contributed by Qunsheng Huang (TUM).

A community

New: Forces for all

Forces calculation for compressible, multiphase and turbulence #64

Merged **MakisH** merged 14 commits into `precice:develop` from `JSeuffert:FSI` on Sep 18, 2019

Conversation 5 | Commits 14 | Checks 1 | Files changed 5

JSeuffert commented on Jan 22, 2019 — Contributor

Concerning issues #57 #42
Maybe in parts the same as pull request #51

Forces calculation is implemented as in the forces functionObject.

I tested the correct implementation with: pimpleDyMFoam, rhoPimpleDyMFoam, interDyMFoam, compressibleInterDyMFoam; laminar flow and RAS turbulence model
Tested with Openfoam 4 (https://github.com/OpenFOAM/OpenFOAM-4.x), precice 1.3.0, Calculix 2.13

Revie

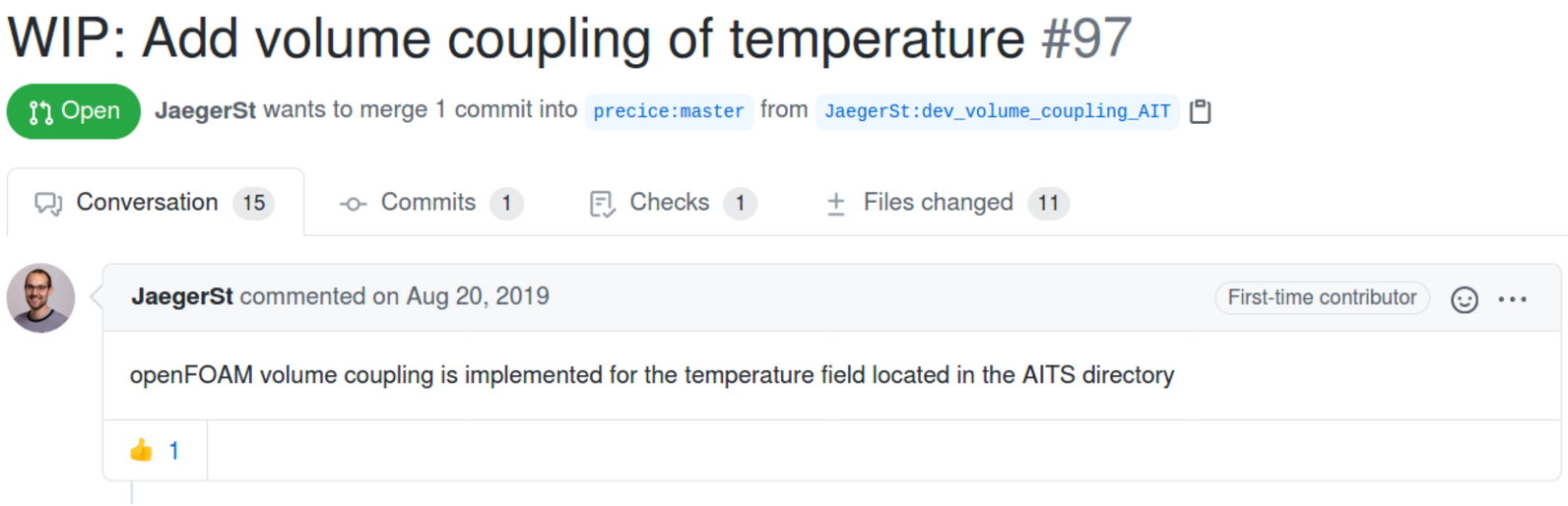
Assig

Labe

None

Contributed by Julian Seuffert (KIT).

WIP: Volume coupling



Contributed by @JaegerSt and @StefanScG (AIT).

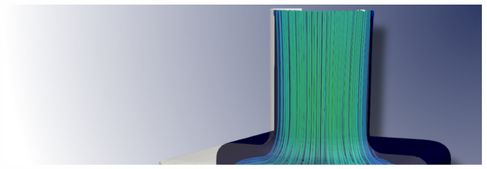
Testimonials

Do you use preCICE?

We are always happy to hear about interesting projects that use preCICE.

Tell us your story!

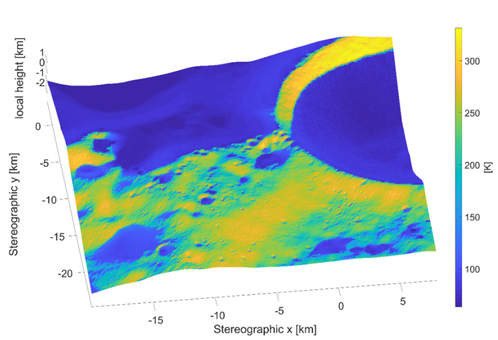
A GUI for OpenFOAM/CalculiX FSI Coupling with preCICE



DHCAE Tools provides services to support companies and universities for the implementation of open-source solver technology. Beside this, DHCAE Tools offers calculation services for flow applications, whereby fluid-structure applications are a long-standing field of activity with commercial solvers as well as with self-developed couplings. Due to preCICE's outstanding capabilities, the setup of a coupling between OpenFOAM and CalculiX is now supported by DHCAE Tools in the graphical interface CastNet. This simplifies the mesh generation, the case setup for both OpenFOAM and CalculiX, the definition of the coupling parameters and finally the monitoring of the simulation. Learn more

Ulrich Heck, DHCAE Tools GmbH, Krefeld, Germany

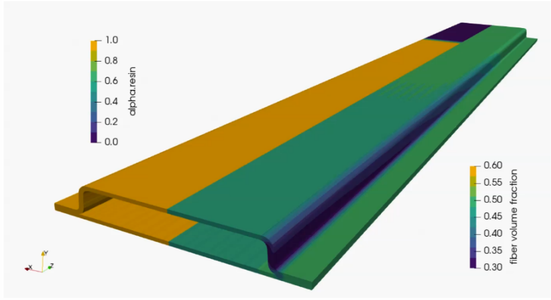
Simulation of Temperatures on the Moon with THerMoS



We are developing THerMoS, a tool for simulation of temperatures on the surface of the Moon including rovers and astronauts operating on the lunar surface. Ray tracing on a single or multiple GPUs with NVIDIA Optix calculates the heat transfer by radiation between surface elements, while a MATLAB routine solves the equation of heat diffusion. preCICE couples the two domains and handles the communication between MATLAB and NVIDIA Optix. Other solvers and simulation approaches (instead of MATLAB and NVIDIA Optix) are going to be tested in the future with the aid of the flexibility that preCICE offers. Learn more

Matthias Killian, Chair of Astronautics, Technical University of Munich (TUM), Germany

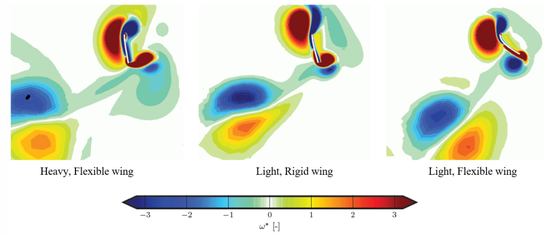
Fluid-Structure Interaction during Sandwich Manufacturing



Structural sandwich components made of continuous fiber reinforced plastics (CFRP) are increasingly demanded by the automotive industry. In the Resin Transfer Molding (RTM) manufacturing process, a polymer foam core is embedded between dry fibers. During manufacturing, a liquid polymer resin infiltrates the fibers with high injection pressure, which leads to a deformation of the foam core. We simulate the mold filling with OpenFOAM and the foam core deformation with CalculiX. By coupling OpenFOAM and CalculiX via preCICE, we can now predict foam core deformation and mold filling behavior correctly. This allows us to optimize the manufacturing of high performance lightweight CFRP sandwich components. Learn more

Julian Seuffert, Lightweight Technology, Institute of Vehicle System Technology (FAST), Karlsruhe Institute of Technology (KIT), Germany

Fluid-Structure Interaction on Flapping Wings

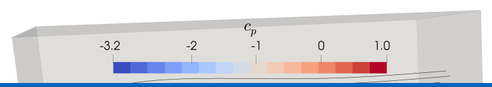


The flow around flapping wings allows them to create high lift using various of unsteady flow phenomena. Adding flexibility to the wing can help to reduce drag and increase performance. At TU Delft, we are investigating flapping wings and the implication of flexibility in these wings. The high, nonlinear deformations of these wings require a strongly coupled simulation to find a solution. For this means a framework is set up using CalculiX and OpenFOAM, coupled with preCICE. For this work, the OpenFOAM adapter was extended to support force and displacement coupling in FSI simulations. The large number of coupling functionalities in preCICE gives the user the opportunity to build advanced and scalable simulations with ease. Learn more

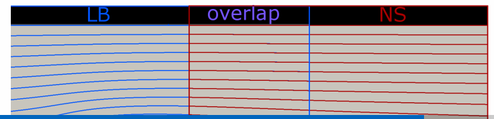
Derek Risseeuw, Aerodynamics, Faculty of Aerospace Engineering, TU Delft, The Netherlands

Coupled Simulation of the Continuous Casting Process

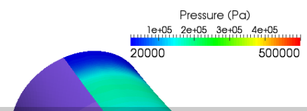
Fluid-Structure Interaction of Inflatable Wing Sections



Hybrid simulation methods for wind modelling in urban areas



FSI Simulations of High Impact Loads on Structures



Resources

Start here: precice.org

Welcome to preCICE

News: preCICE release v2.0.2 available since Apr 3, 2020

Now also on YouTube: Subscribe to our new YouTube channel, where you can also find the talks from the preCICE Workshop 2020.

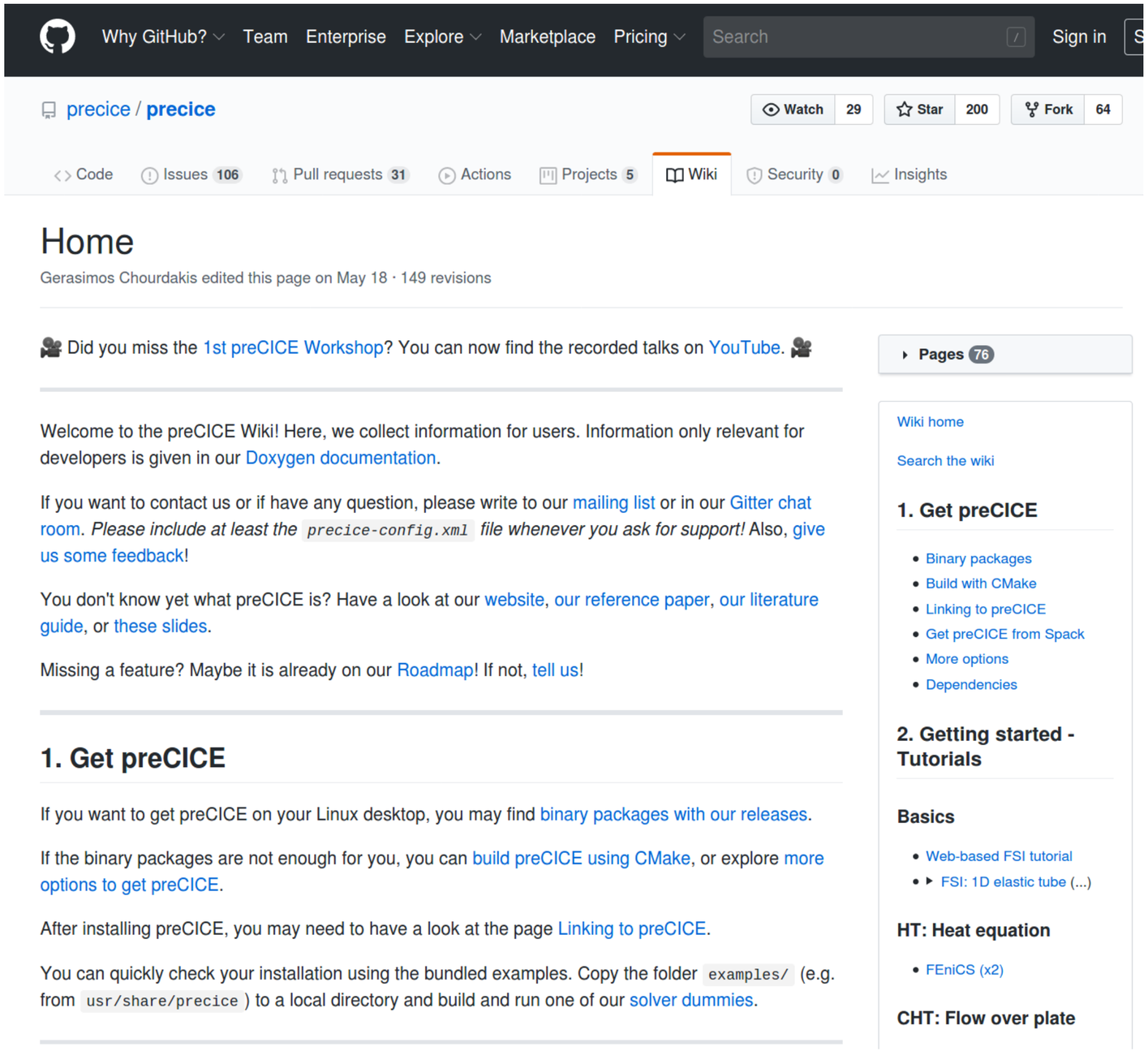
preCICE in xSDK: preCICE is now part of the xSDK framework. Read the compatibility policies…

preCICE (Precise Code Interaction Coupling Environment) is a coupling library for partitioned multi-physics simulations, including, but not restricted to fluid-structure interaction and conjugate heat transfer simulations. Partitioned means that preCICE couples existing programs (solvers) capable of simulating a subpart of the complete physics involved in a simulation. This allows for the high flexibility that is needed to keep a decent time-to-solution for complex multi-physics scenarios. preCICE runs efficiently on a wide spectrum of systems, from low-end workstations up to complete compute clusters and has proven scalability on 10000s of MPI Ranks.

The software offers methods for transient equation coupling, communication means, and data mapping schemes. preCICE is written in C++ and offers additional bindings for C, Fortran, Matlab, and Python. Ready-to-use adapters for well-known commercial and open-source solvers, such as OpenFOAM, deal.II, FEniCS, SU2, or CalculiX, are available. Due to the minimally-invasive approach of preCICE, adapters for in-house codes can be implemented and validated in only a few weeks.

preCICE is an open-source software under the LGPL3 license and available on GitHub.

Documentation



Discuss & get help (threaded)

preCICE

Home GitHub Twitter YouTube Sign Up Log In

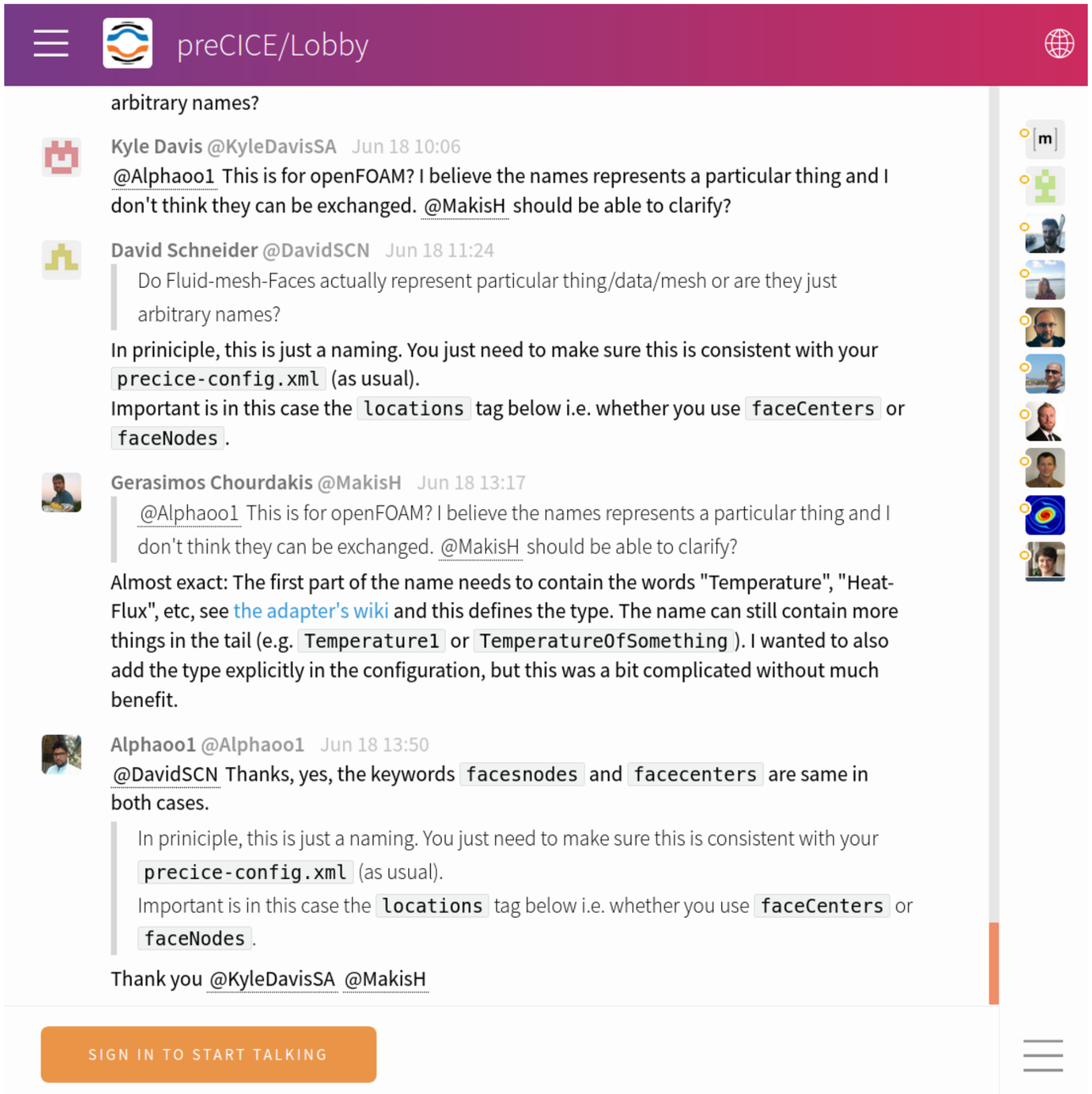
all categories all tags all Categories Latest Top

Category	Topics
News News, announcements, “blog”-like posts	4
Is preCICE for me? General questions regarding preCICE as a coupling solution.	11
Installing preCICE Any issues with getting the preCICE library installed	15
Using preCICE Using the preCICE API, configuring a new simulation	44
Official adapters and tutorials Installing, configuring, extending adapters, running or modifying the tutorials	33
Site Feedback Discussion about this site, its organization, how it works, and how we can improve it.	2
Uncategorized Topics that don't need a category, or don't fit into any other existing category.	1

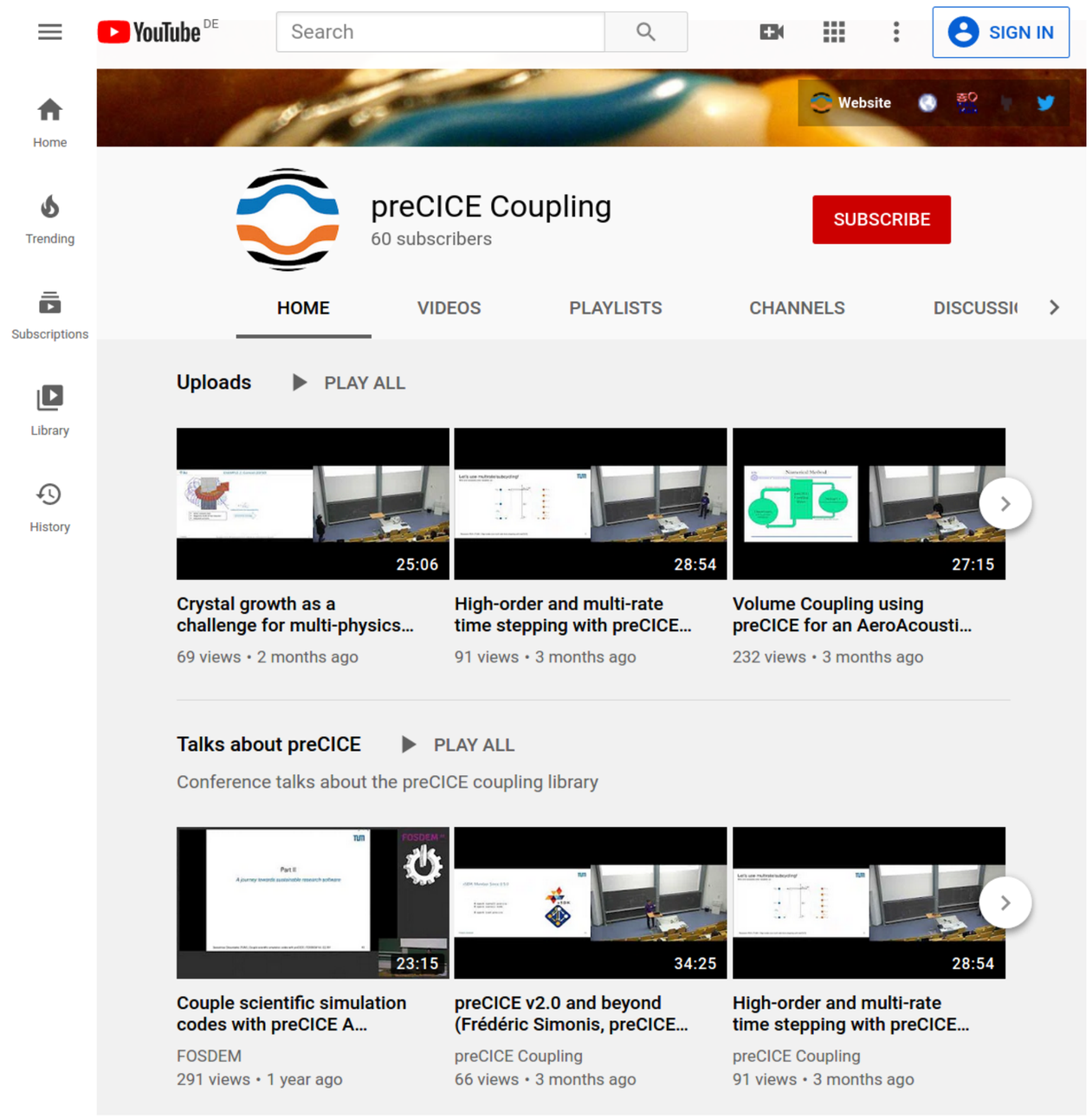
Latest

	Welcome to the preCICE Forum on Discourse Site Feedback	2 Nov '19
B	Status of Quads on the Interface Is preCICE for me?	2 3h
J	Energy conservation through two independent consistent mappings Using preCICE configuration	4 9d
	IQN fails for quickly changing forces during wetting in a free-surface FSI problem Using preCICE	3 9d
	Assertion failed on ResidualSumPreconditioner Using preCICE	3 9d
	RBF mapping vs. nearest projection mapping Using preCICE openfoam data-mapping calculix fsi	3 14d

Discuss & get help (quick)



Learn: YouTube



Get news: Twitter

preCICE

@preCICE_org

A free/open-source coupling library for partitioned multi-physics simulations, including fluid-structure interaction, conjugate heat transfer, and more.

Germany precice.org Joined April 2018

54 Following **174** Followers

Follow

Get news: Mailing list

preCICE -- preCICE Parallel Coupling Environment

About preCICE	English (USA)

To see the collection of prior postings to the list, visit the preCICE Archives.

Using preCICE

To post a message to all the list members, send email to precice@mailman.informatik.uni-stuttgart.de.

You can subscribe to the list, or change your existing subscription, in the sections below.

Subscribing to preCICE

Subscribe to preCICE by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a private list, which means that the list of members is not available to non-members.

Your email address:	
Your name (optional):	

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

Advertisements

preCICE Workshop 2021

preCICE Workshop 2020 @ TUM

See also

Training III-E: Multiphysics Modeling with the preCICE Coupling Library: Coupling OpenFOAM with CalculiX/deal.II

Wed. Jun 24, 2020 8:30 AM - 10:00 AM

Gerasimos Chourdakis

Technical University of Munich, Germ…

Send Message

Training II-E: OpenFOAM for Multiphysics Applications

Tue. Jun 23, 2020 8:30 AM - 10:00 AM

Holger Marschall

Technical University Darmstadt, Germ…

Send Message

preCICE is free because of

H2020 grant 754462

and the code/issues/testing/documentation contributions of people like you (thank you!).

Summary

Easily exchange your coupled solvers and explore advanced & performant numerical algorithms

Gerasimos Chourdakis (TUM)
Benjamin Uekermann (TU/e)
+ many more (see precice.org/about)

Write me at chourdak@in.tum.de
or start a thread on our forum

Get this presentation from GitHub.