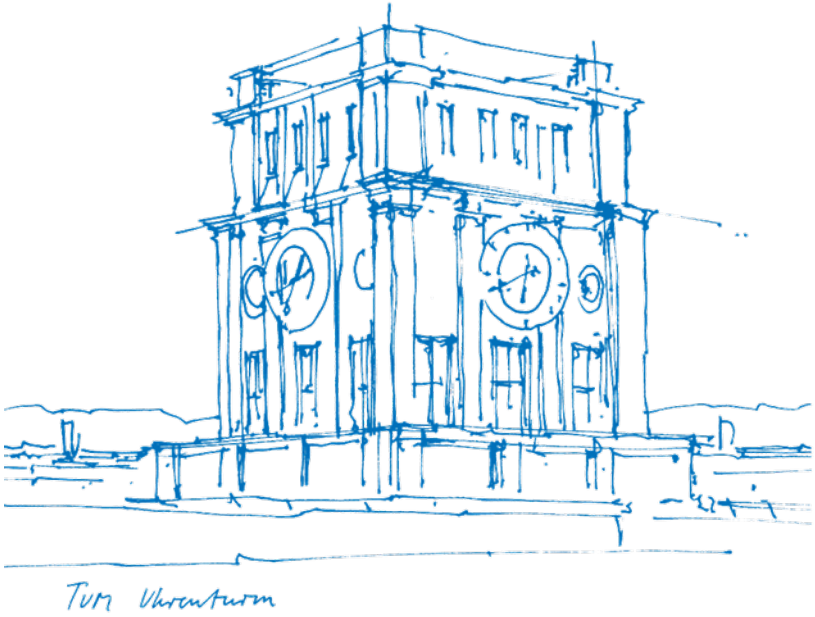
Couple OpenFOAM with any other solver using preCICE

Gerasimos Chourdakis et al.

Technical University of Munich
Department of Informatics
Chair of Scientific Computing in Computer Science

2nd German OpenFoam User meetiNg
TU Braunschweig
February 21, 2018



Agenda

Part I:

preCICE

Agenda

Part I:

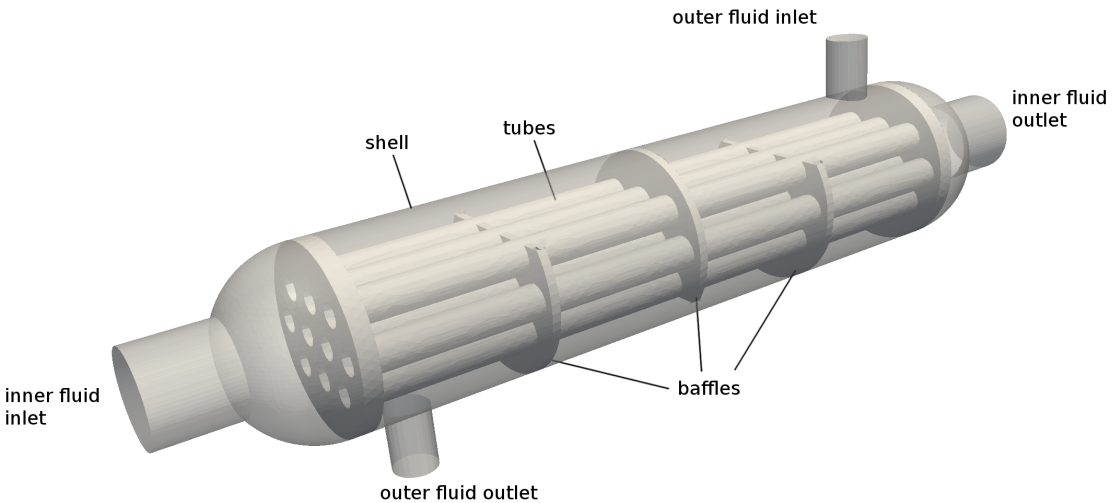
preCICE

Part II:

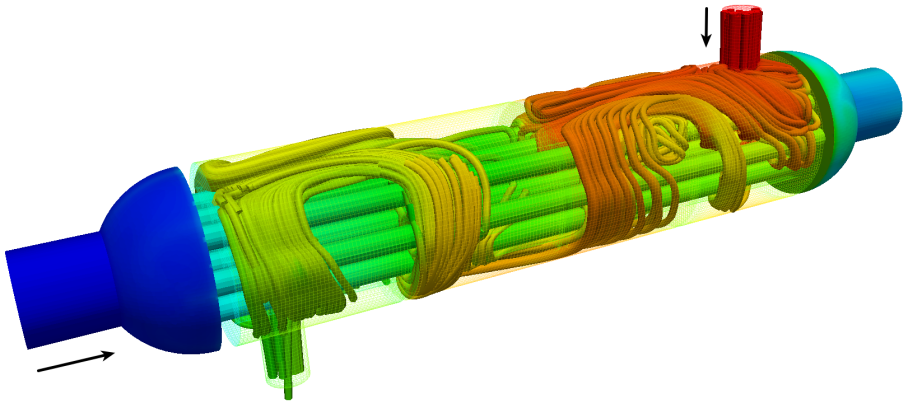
preCICE + OpenFOAM The OpenFOAM Foundation

OpenFOAM

How to simulate this heat exchanger?

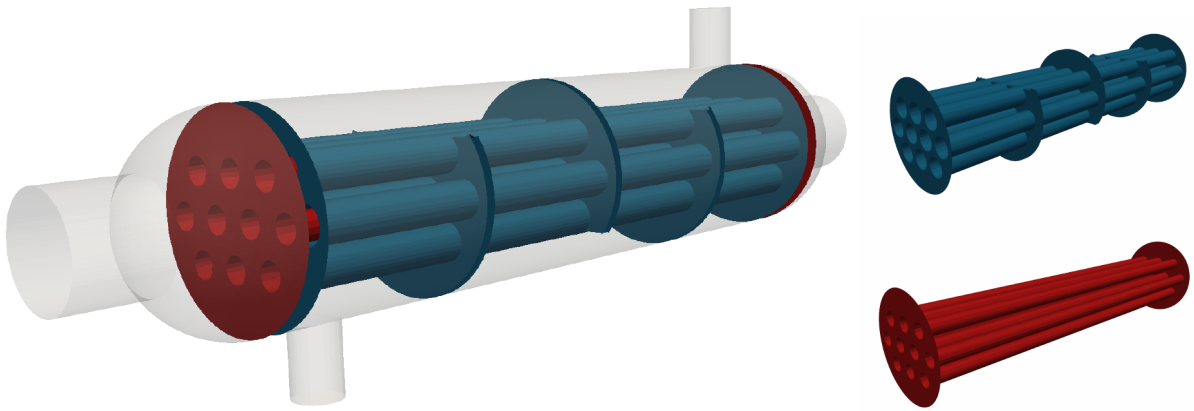


Geometry of a shell-and-tube heat exchanger (Image by L. Cheung Yau, 2016)

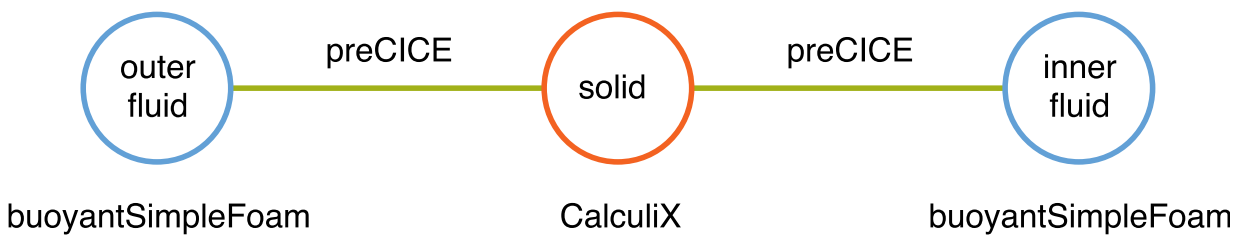


Surface plot and streamlines of the two fluids colored by temperature. Solid not shown.

A shell-and-tube heat exchanger with preCICE

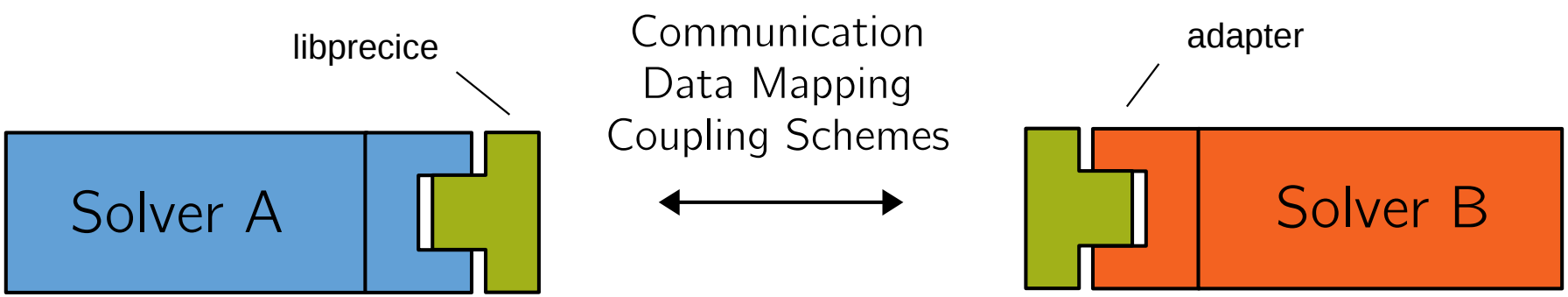


Coupling interfaces (Image by L. Cheung Yau, 2016)



preCICE

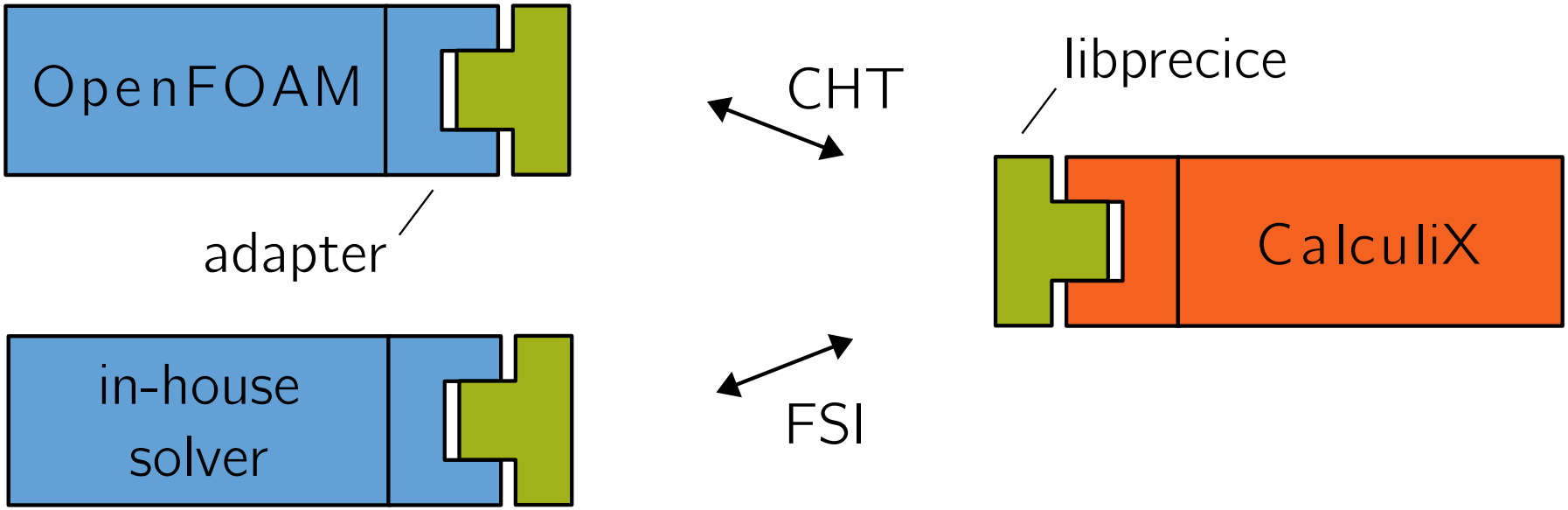
precise Code Interaction Coupling Environment



- **Free (GNU LGPL)**, developed at TU Munich & Univ. of Stuttgart.
- **Version 1.0** in November 2017 (10+ years, 3 PhD generations).
- **Official adapters** for CalculiX, Code_Aster, COMSOL, Fluent, **OpenFOAM**, SU2

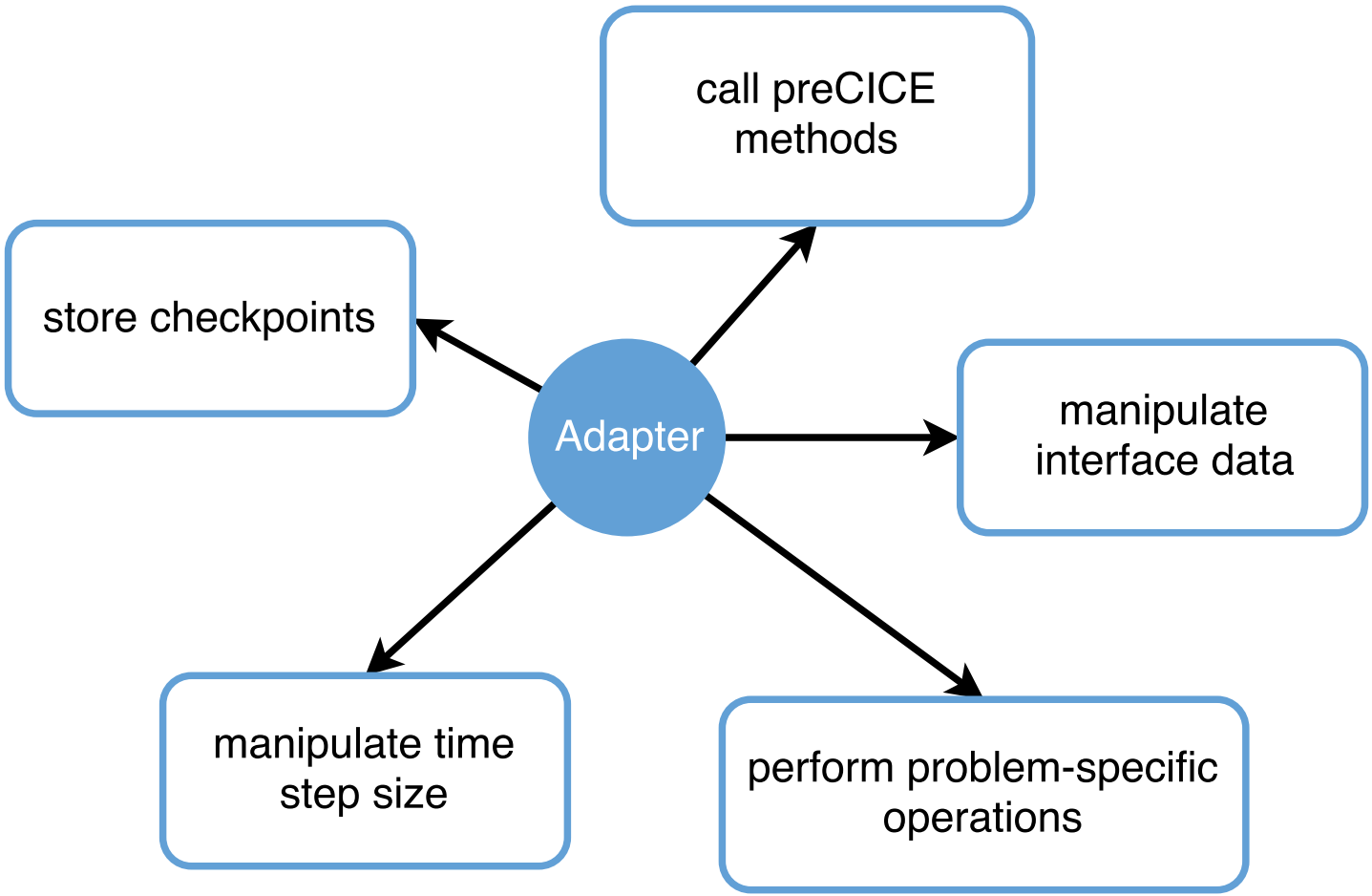
- **Third-party adapters** for Ateles, Alya, Carat++, FASTEST, FEAP, **foam-extend**, ...
- **API** in C, **C++**, Fortran, Python

But why preCICE?



- Pure **library** approach → flexibility
- Fully parallel, **peer-to-peer** concept → scalable and efficient communication
- Sophisticated and robust **quasi-Newton** coupling algorithms
- **Multi-coupling**

The roles of an adapter



Part IIa: previous approach

preCICE + OpenFOAM The OpenFOAM Foundation

OpenFOAM

Duplicated development effort

OpenFOAM (and family) adapters for preCICE

David Blom,
2015-17 (TU Delft)
FSI, foam-extend

Duplicated development effort

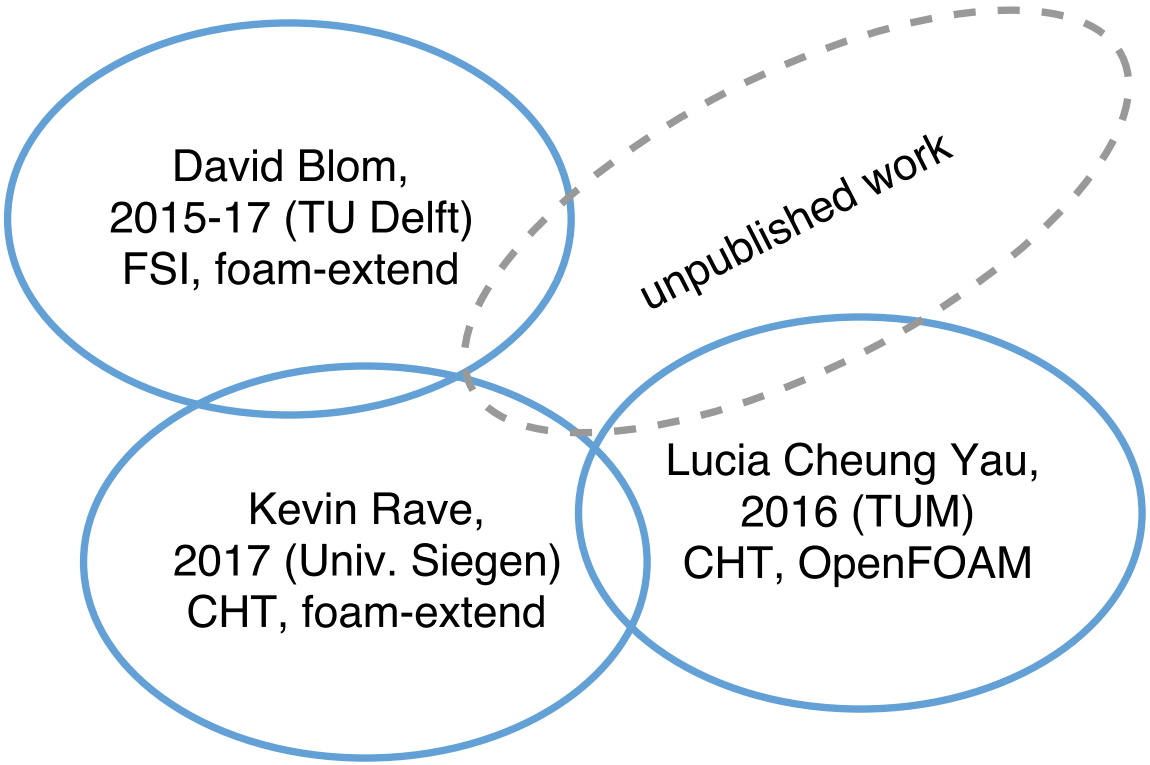
OpenFOAM (and family) adapters for preCICE

David Blom,
2015-17 (TU Delft)
FSI, foam-extend

Lucia Cheung Yau,
2016 (TUM)
CHT, OpenFOAM

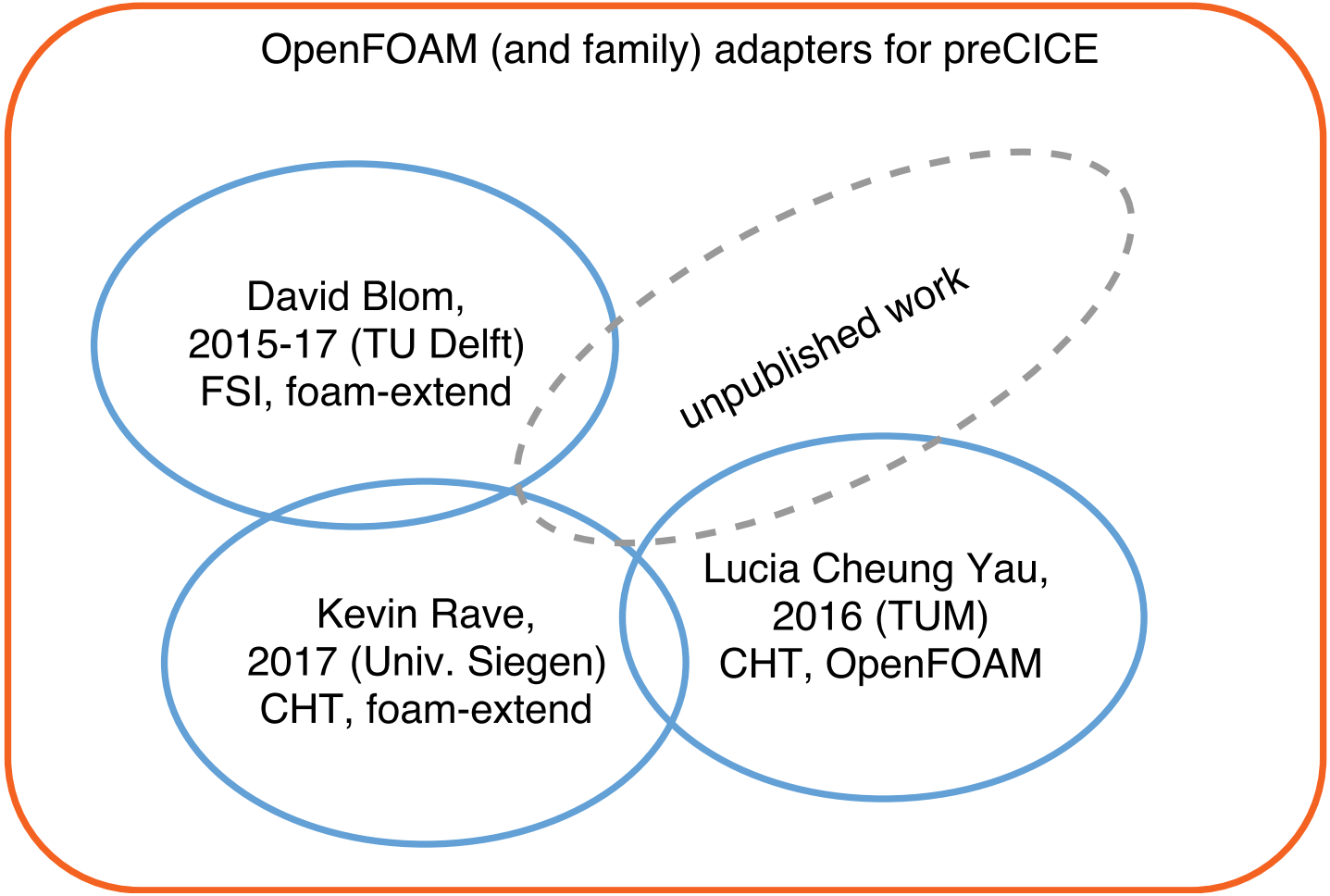
Duplicated development effort

OpenFOAM (and family) adapters for preCICE



All these adapters are **bound to specific solvers**!

Duplicated development effort



All these adapters are **bound to specific solvers**!
→ We need an official, general adapter!

Example of an adapted solver (previous)

```
/* Adapter: Initialize coupling */
adapter.initialize();

Info<< "\nStarting time loop\n" << endl;

while (adapter.isCouplingOngoing()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  /* Adapter: Adjust solver time */
  adapter.adjustSolverTimeStep();

  /* Adapter: Write checkpoint */
  if(adapter.isWriteCheckptRequired())
    adapter.writeCheckpoint();

  runTime++;

  /* Adapter: Receive coupling data */
  adapter.readCouplingData();

  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

  /* Adapter: Write in buffers */
  adapter.writeCouplingData();

  /* Adapter: advance the coupling */
  adapter.advance();

  /* Adapter: Read checkpoint */
  if(adapter.isReadCheckptRequired())
      adapter.readCheckpoint();

  if(adapter.isCouplTimeStepComplete())
    runTime.write();

}
```

Before: Working and validated prototypes

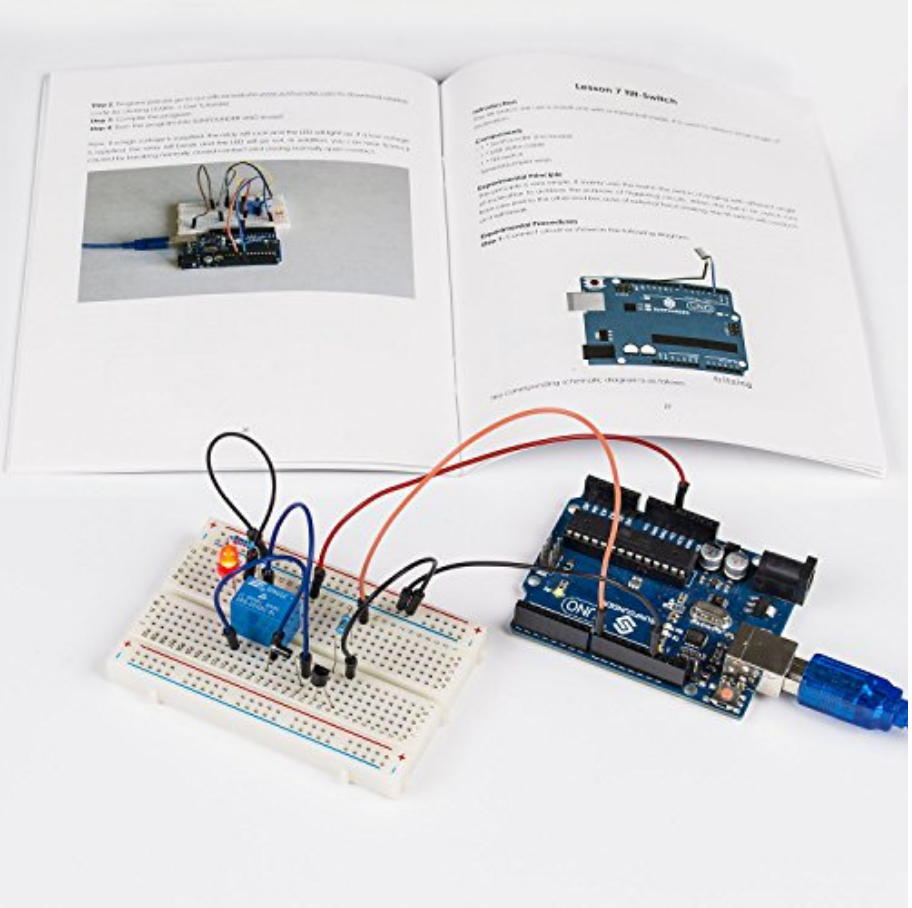


Image from desertcart.ae.

Before: Working and validated prototypes

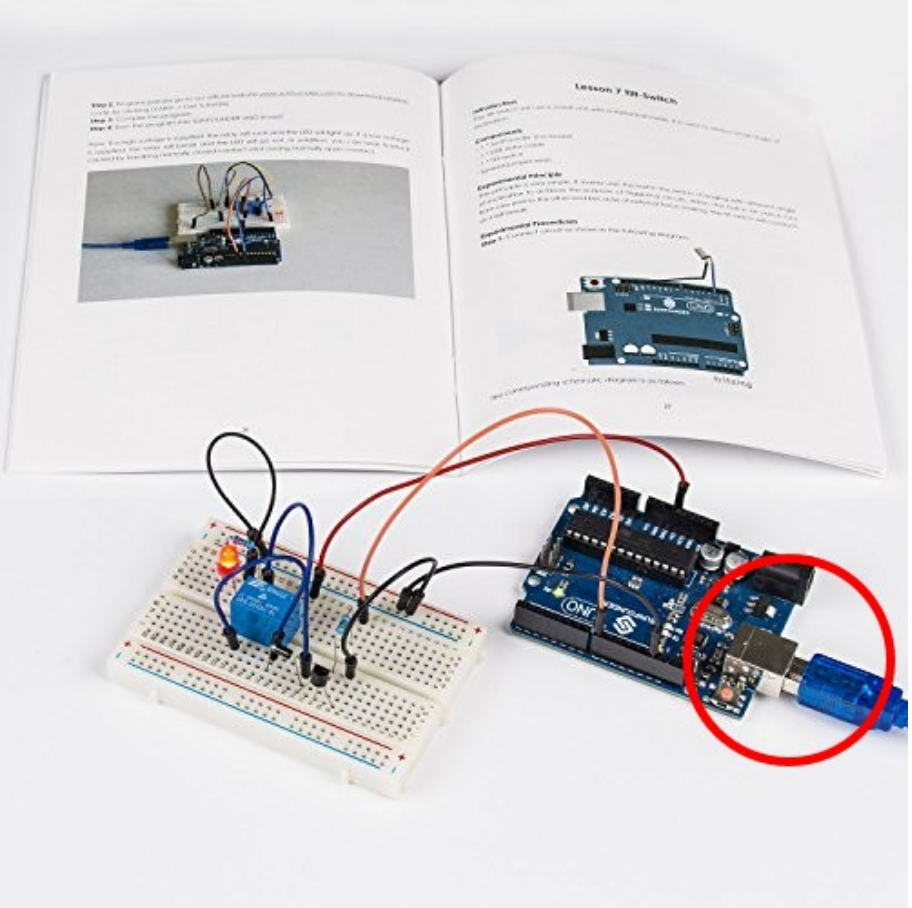
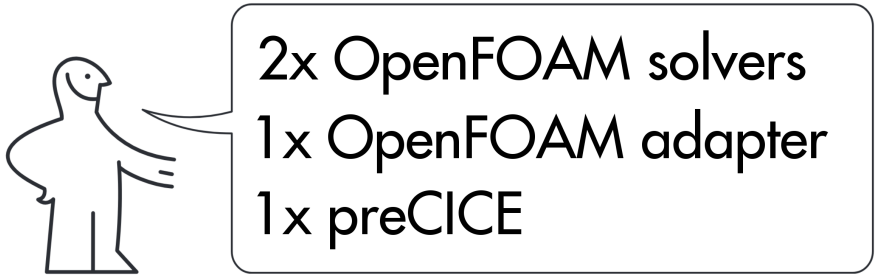


Image from desertcart.ae.

Now: A user-friendly, plug-and-play adapter

KOPPLAD



The human-like figure is a property of ikea.com.

Part IIb: a new, official adapter

preCICE + OpenFOAM The OpenFOAM Foundation

OpenFOAM

Making this a function object

Several **challenges**:

- No changes in the source allowed
  - Cannot use variables directly
  - Ask the objects' registry
- One adapter for all the solvers and problem types
  - Some parameters are not available
- Only one call to `execute()` at the end
  - We may need to reload a checkpoint at the last timestep...
  - Set the `endTime` to `GREAT` and exit when ready.
- Collaboration with other function objects
  - At the end, call any other `end()` methods explicitly.
- Error handling
  - `read()` degrades errors to warnings
  - Catch them and throw them in `execute`
- One adapter for all the OpenFOAM flavors and versions?
  - E.g. `boundaryField()` and `boundaryFieldRef()`
  - E.g. missing `adjustTimeStep()`
  - How to distribute? Branches/Tags? Preprocessor `ifdef`?
- ...

Making this a function object

Several **challenges**:

- No changes in the source allowed
  - Cannot use variables directly
  - Ask the objects' registry
- One adapter for all the solvers and problem types
  - Some parameters are not available
- Only one call to `execute()` at the end
  - We may need to reload a checkpoint at the last timestep...
  - Set the `endTime` to `GREAT` and exit when ready.
- Collaboration with other function objects
  - At the end, call any other `end()` methods explicitly.
- Error handling
  - `read()` degrades errors to warnings
  - Catch them and throw them in `execute`
- One adapter for all the OpenFOAM flavors and versions?
  - E.g. `boundaryField()` and `boundaryFieldRef()`
  - E.g. missing `adjustTimeStep()`
  - How to distribute? Branches/Tags? Preprocessor `ifdef`?
- ...

Several **advantages**:

- No source code changes
- Load at runtime
- (mostly) Solver agnostic

Making this a function object

Several **challenges**:

- No changes in the source allowed
  - Cannot use variables directly
  - Ask the objects' registry
- One adapter for all the solvers and problem types
  - Some parameters are not available
- Only one call to `execute()` at the end
  - We may need to reload a checkpoint at the last timestep...
  - Set the `endTime` to `GREAT` and exit when ready.
- Collaboration with other function objects
  - At the end, call any other `end()` methods explicitly.
- Error handling
  - `read()` degrades errors to warnings
  - Catch them and throw them in `execute`
- One adapter for all the OpenFOAM flavors and versions?
  - E.g. `boundaryField()` and `boundaryFieldRef()`
  - E.g. missing `adjustTimeStep()`
  - How to distribute? Branches/Tags? Preprocessor `ifdef`?
- ...

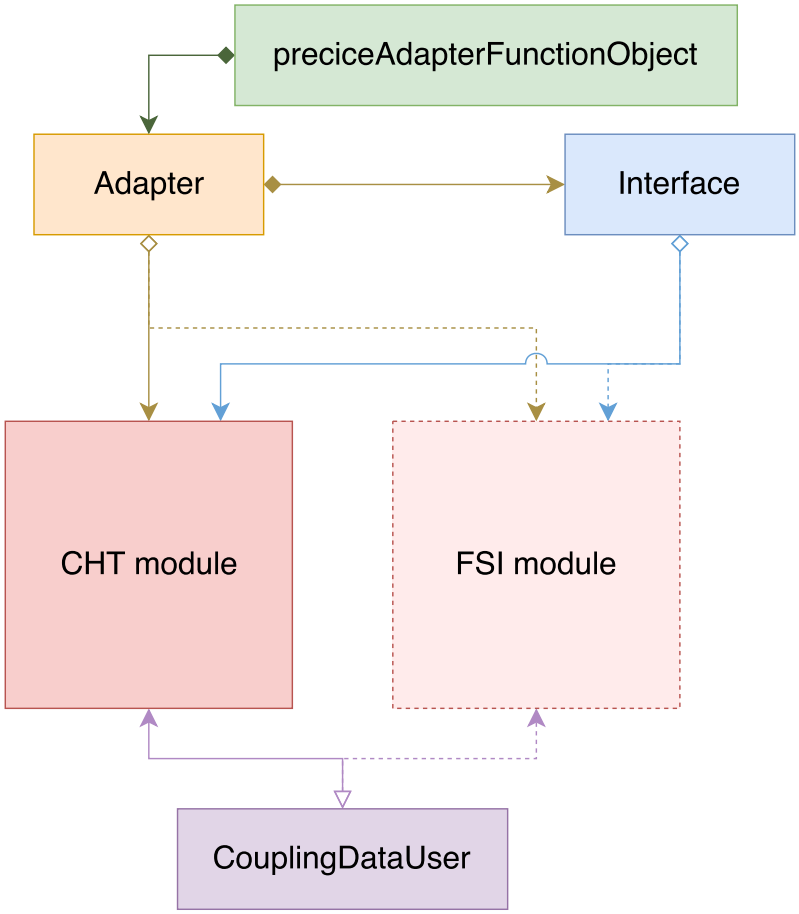
Several **advantages**:

- No source code changes
- Load at runtime
- (mostly) Solver agnostic

However:

- Still ready-to-run only for CHT
- but...

An extensible adapter



OK! I want to use it!

OpenFOAM configuration

```
// system/controlDict
functions
{
    preCICE_Adapter
    {
        type preciceAdapterFunctionObject;
        libs ("libpreciceAdapterFunctionObject.so");
    }
}
```

Set the appropriate boundary condition types:

```
// 0/T
interface
{
    type            fixedValue;
    value           uniform 300;
}

// other types: fixedGradient, mixed
```

OpenFOAM configuration

```
// system/controlDict
functions
{
    preCICE_Adapter
    {
        type preciceAdapterFunctionObject;
        libs ("libpreciceAdapterFunctionObject.so");
    }
}
```

Set the appropriate boundary condition types:

```
// 0/T
interface
{
    type            fixedValue;
    value           uniform 300;
}

// other types: fixedGradient, mixed
```

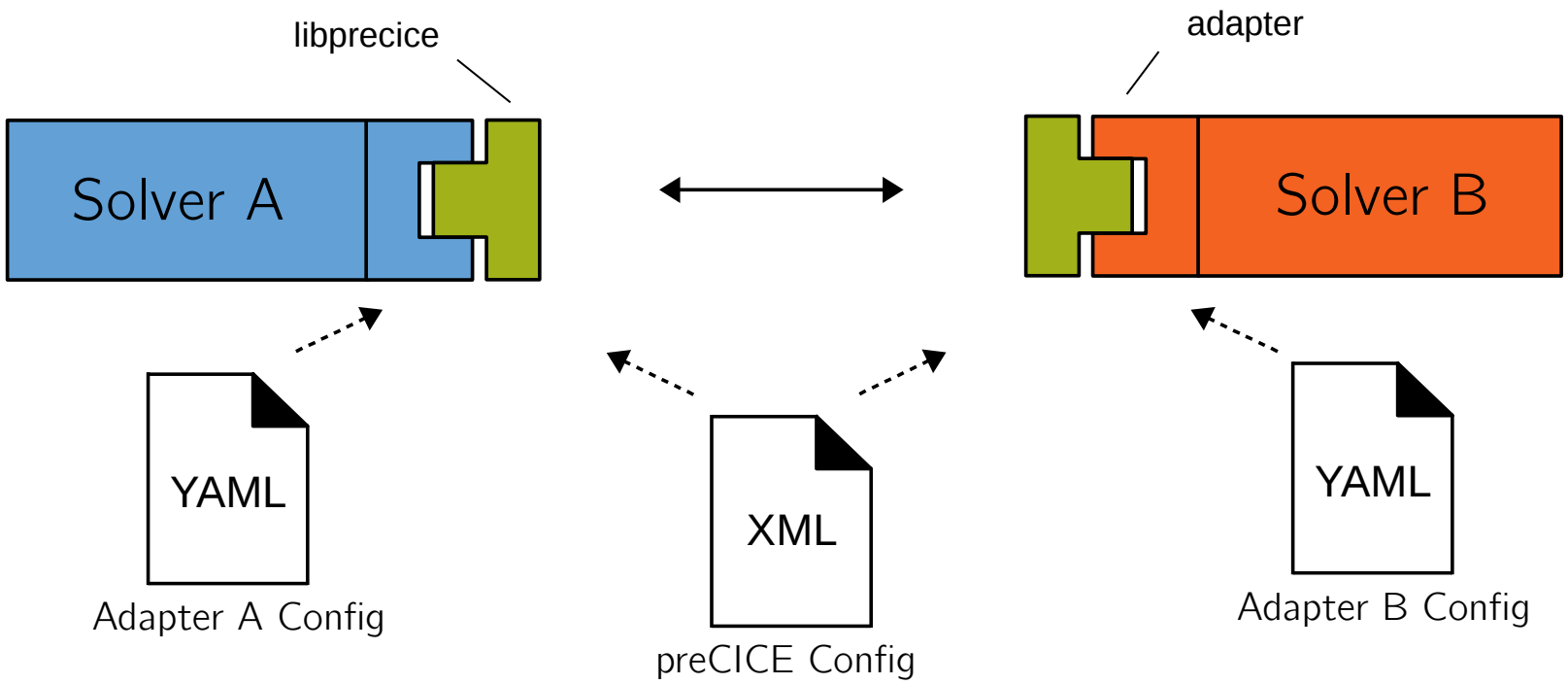
Properties for incompressible solvers:

```
// constant/transportProperties
rho     rho [ 1 -3  0  0 0 0 0 ] 1;
Cp      Cp  [ 0  2 -2 -1 0 0 0 ] 5000;
```

Properties for basic solvers:

```
// constant/transportProperties
k       k   [ 1  1 -3 -1 0 0 0 ] 100;
```

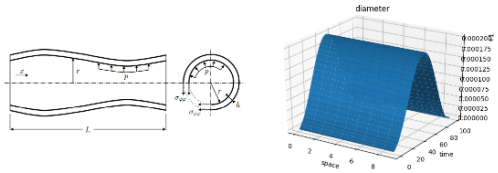
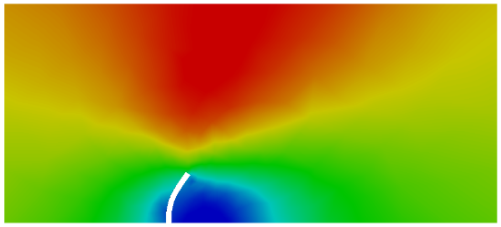
preCICE & adapter configuration



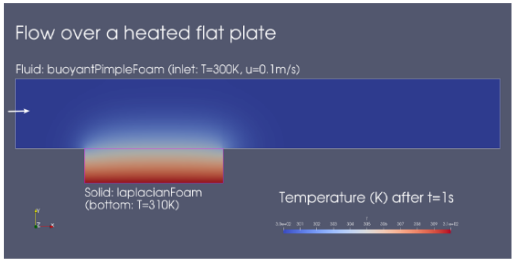
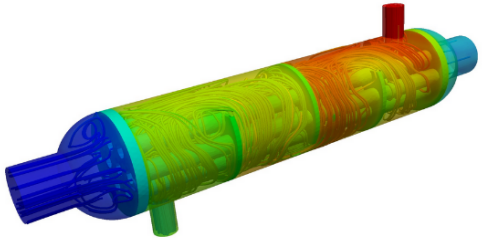
To run the simulation, just execute the solvers as usual.

Tutorials

Fluid-Structure Interaction

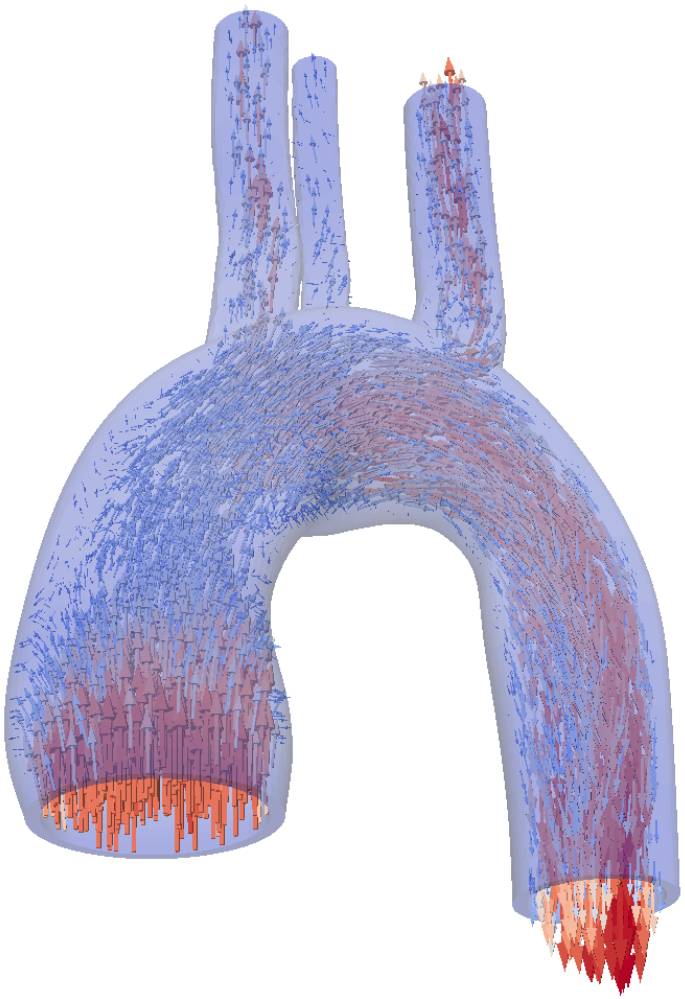
1D FSI Example	FSI with SU2 and CalculiX
Flow through a deformable tube	Flow in a channel with an elastic flap
	

Conjugate Heat Transfer

CHT with OpenFOAM	CHT with OpenFOAM and CalculiX
Flow above a heated plate	Shell-and-tube heat exchanger
	

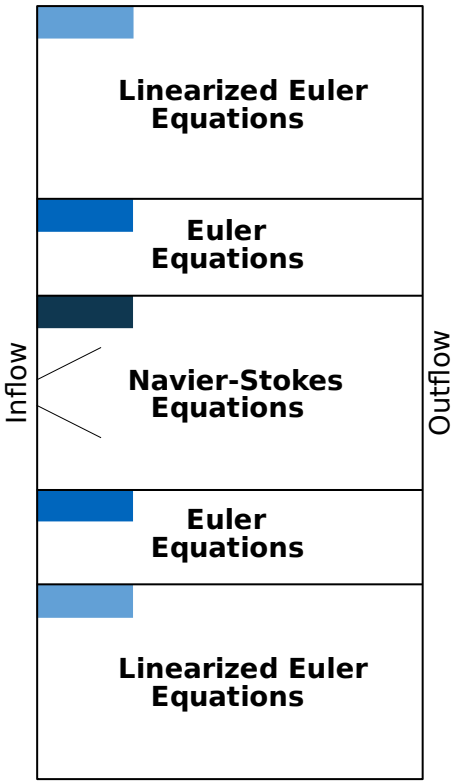
Example: Biomedical applications

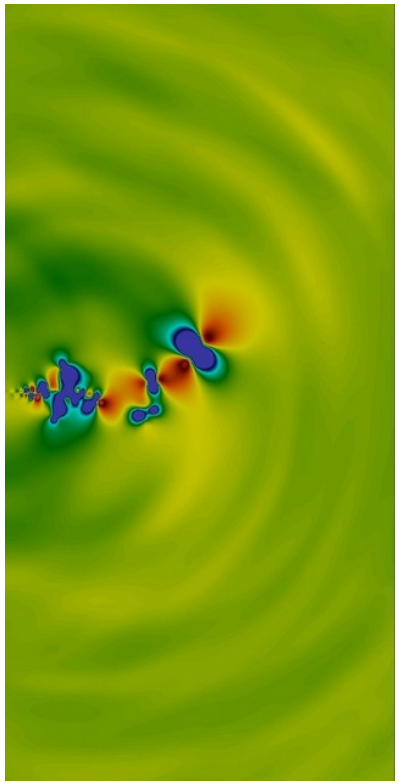
- FSI simulation of an aortic bloodflow
- Joint work with the Barcelona Supercomputing Center



Example: Multi-fluid coupling

- Besides FSI, many other possible applications of preCICE
- Simulation of a subsonic jet
- Explicit, parallel coupling between three fluid solvers
- Joint work with the University of Siegen





Does it work with “chocolate” OpenFOAM?

Known to work with:

The OpenFOAM Foundation: 4.0 – dev

ESI - OpenCFD: v1706

Currently does not work with:

The OpenFOAM Foundation: $\leq$ 3.0

ESI - OpenCFD: $\leq$ v1606+

foam-extend: any version

Coming soon:

- Support for older versions
- Code improvements and tests
- Fluid-Structure Interaction Module

Support OpenFOAM 3.x #8

Open **MakisH** opened this issue on Nov 28, 2017 · 0 comments

MakisH commented on Nov 28, 2017 • edited by floli Collaborator

While openfoam.org versions 4.x have no changes that affect the adapter in comparison to versions 5.x, there are some differences in older versions.

Versions 3.x are different from 4.x in at least the following points:

- The way that the `boundaryField` and `internalField` are accessed. In the OpenFOAM 4.0 release notes, it is stated:

> Robust data handling: new convention for const and non-const reference functions of fields where the non-const function uses ...Ref(); for example, where boundaryField() provides the const reference to the boundary field, boundaryFieldRef() provides a non-const reference. for tmp objects, non-const access uses a ref() function rather than the () dereferencing operator.

See also the OpenFOAM-dev commit a4e2afa and this issue.

- The class hierarchy for turbulence models, as version 4.0 introduced the templated class `TurbulenceModels`. See the OpenFOAM-4.x commit 93732c8.
- Small changes in the function objects. See the OpenFOAM 4.x commit 91aba2d. You may find a function objects code template for your version in the `$FOAM_ETC/` directory. The files preciceAdapterFunctionObject.H and preciceAdapterFunctionObject.C need to be adjusted.

Contribute on GitHub!

Questions?

Website: `precice.org`
Source/Wiki: `github.com/precice` ☆

Mailing list: `precice.org/resources`
My e-mail: `gerasimos.chourdakis@tum.de`

preCICE

Homework:

- Follow a tutorial
- Join our mailing list
- Star on GitHub
- Send us feedback
- Ask me for stickers

Questions?

Website: `precice.org`
Source/Wiki: `github.com/precice` ☆

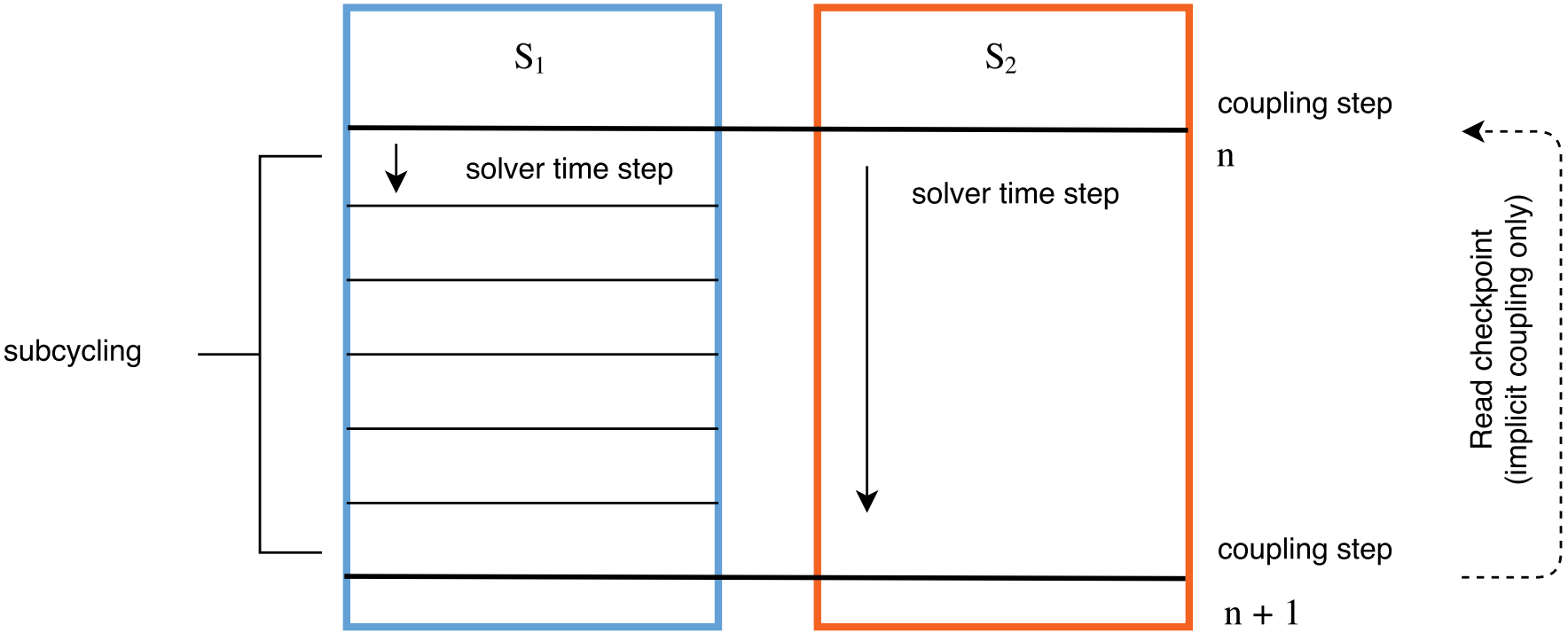
Mailing list: `precice.org/resources`
My e-mail: `gerasimos.chourdakis@tum.de`

Homework:

- Follow a tutorial
- Join our mailing list
- Star on GitHub
- Send us feedback
- Ask me for stickers

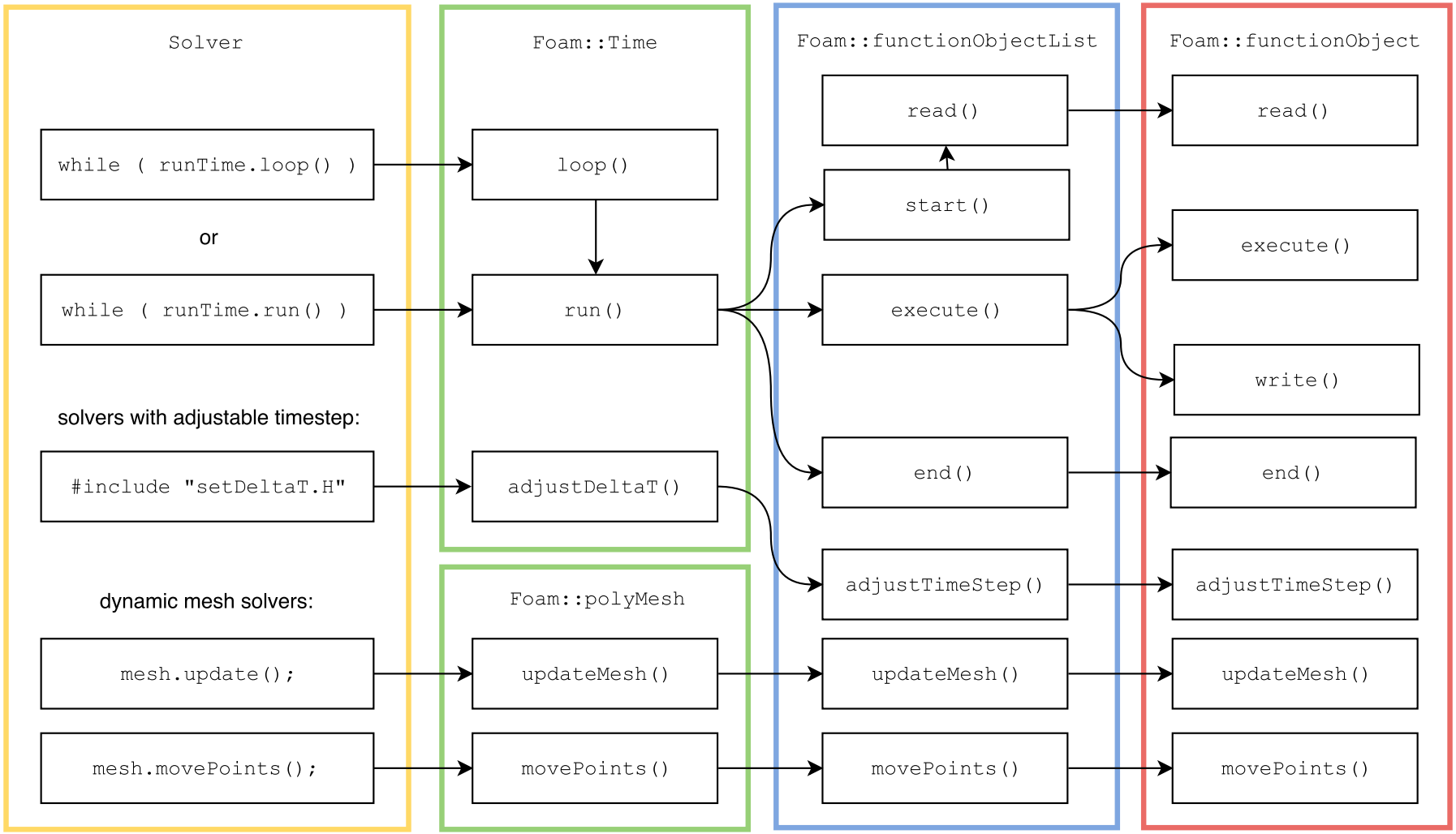
preCICE

Additional slide: Time step sizes



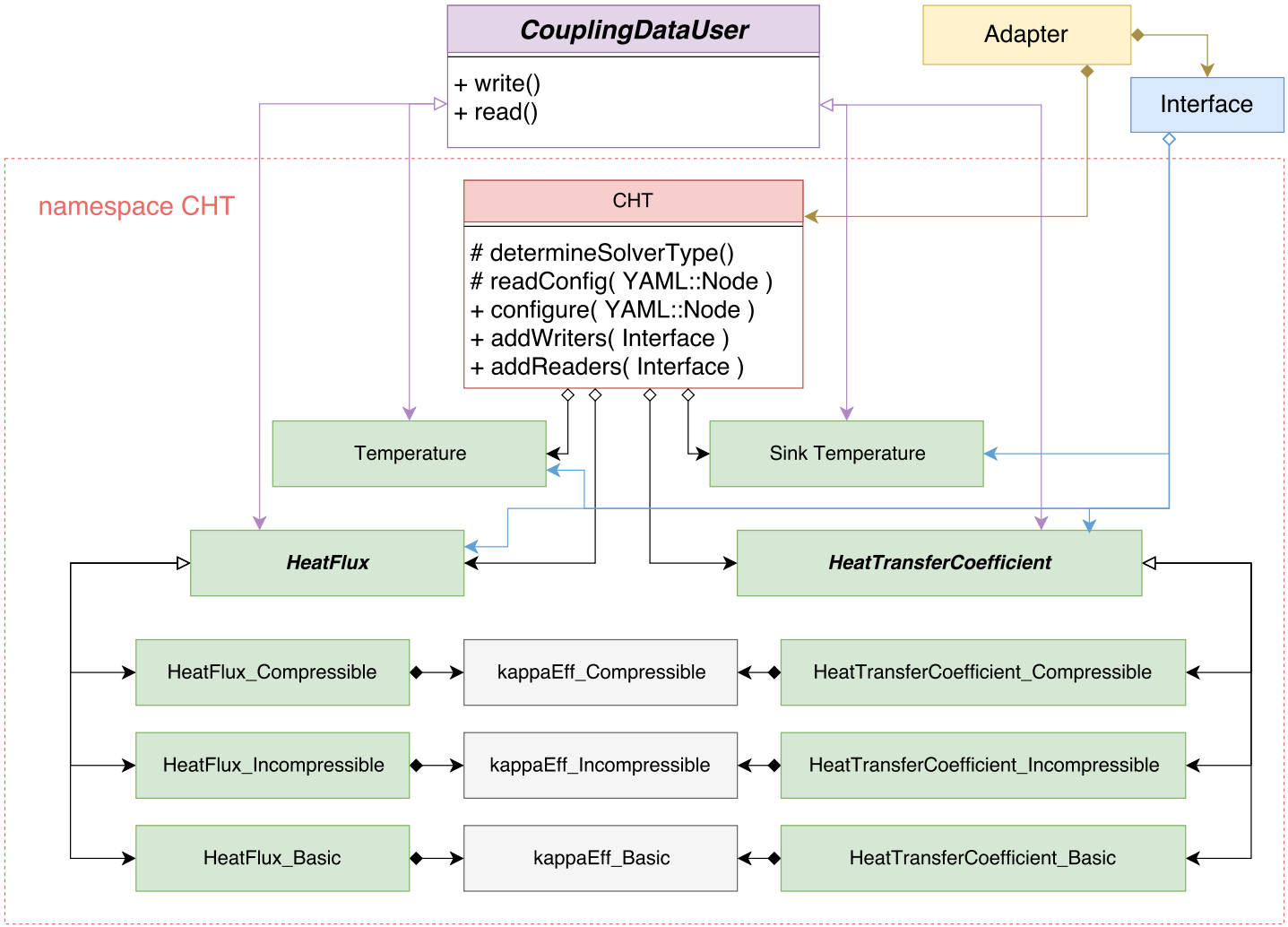
Time step sizes and checkpointing

Additional slide: Function Objects



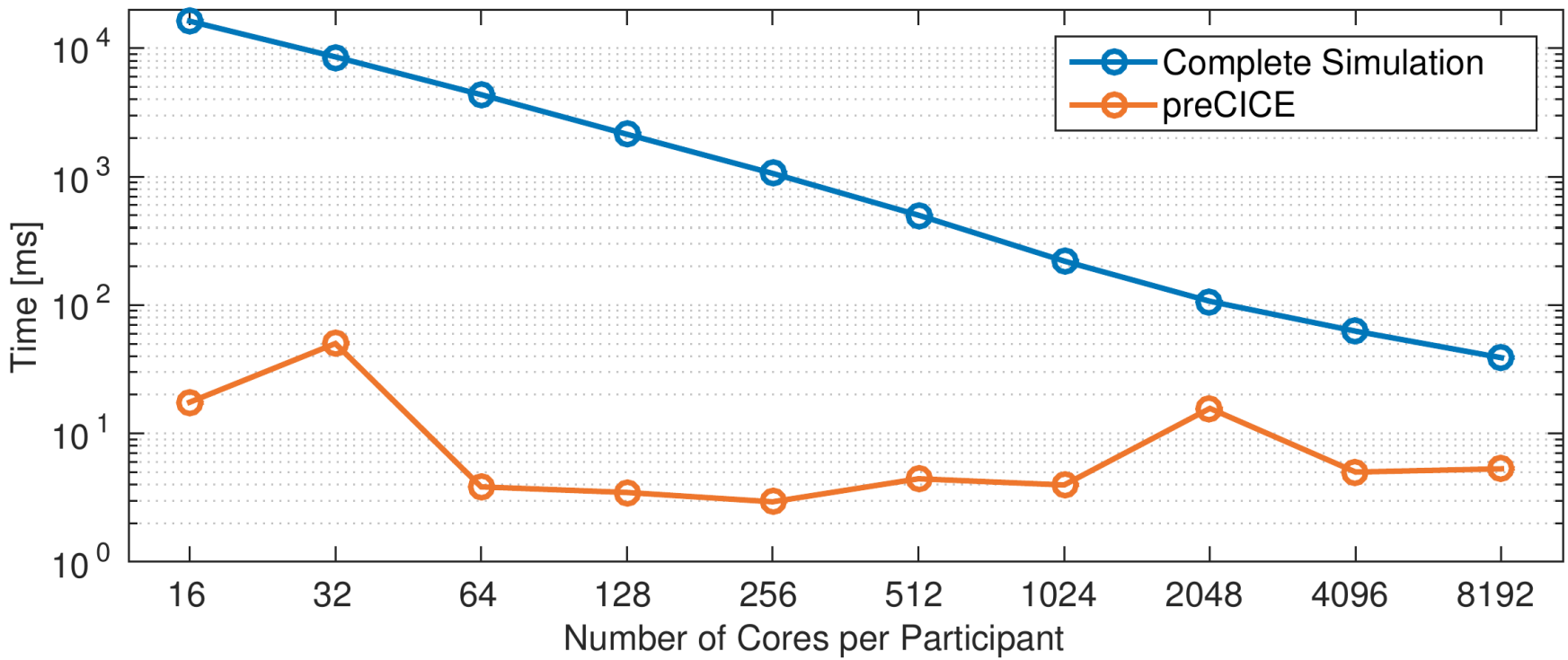
Callbacks in OpenFOAM function objects

Additional slide: The CHT Module



The Conjugate Heat Transfer module

Additional slide: preCICE scaling



Strong scaling of a coupled simulation with two Ateles participants and $5.7 \cdot 10^7$ *dofs*