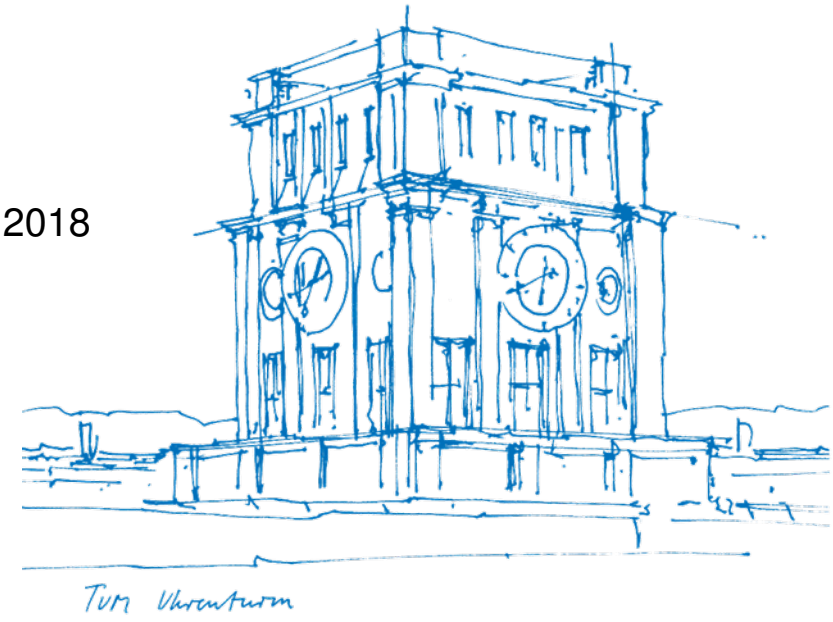
Couple OpenFOAM with other solvers for Multi-Physics simulations using preCICE

Gerasimos Chourdakis et al.

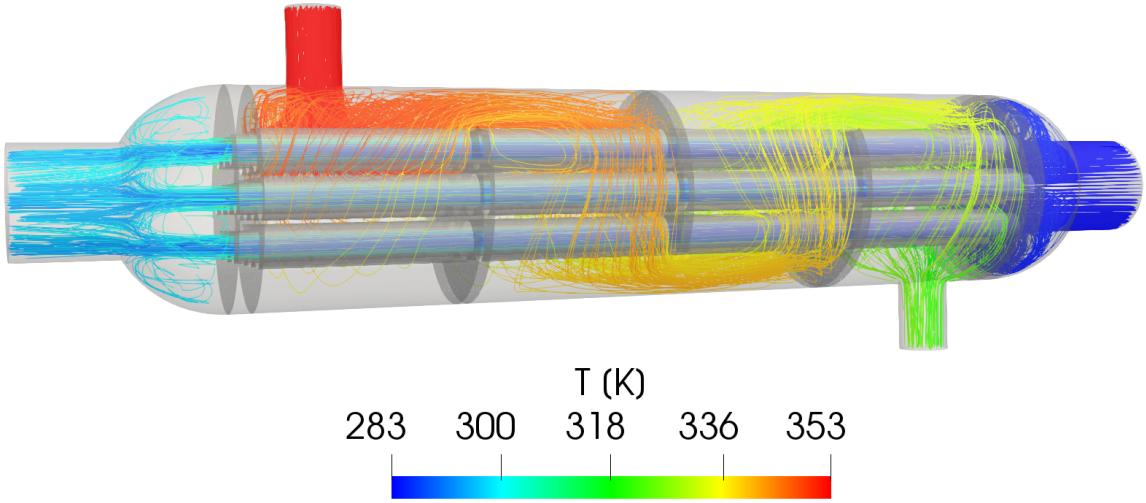
Technical University of Munich
Department of Informatics
Chair of Scientific Computing in Computer Science
October 24, 2018

ESI OpenFOAM conference in Germany | Oct. 23-25, 2018

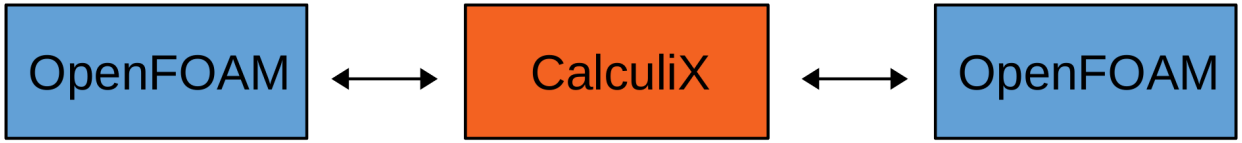


Multi-physics simulations

Conjugate Heat Transfer:

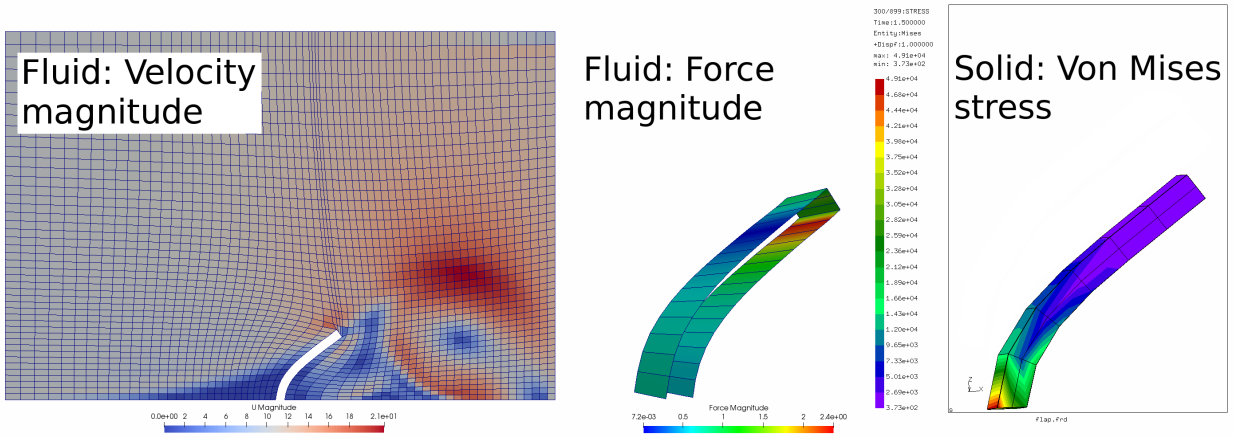


Three (coupled) simulations:

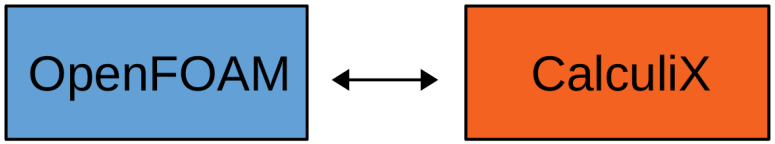


Multi-physics simulations

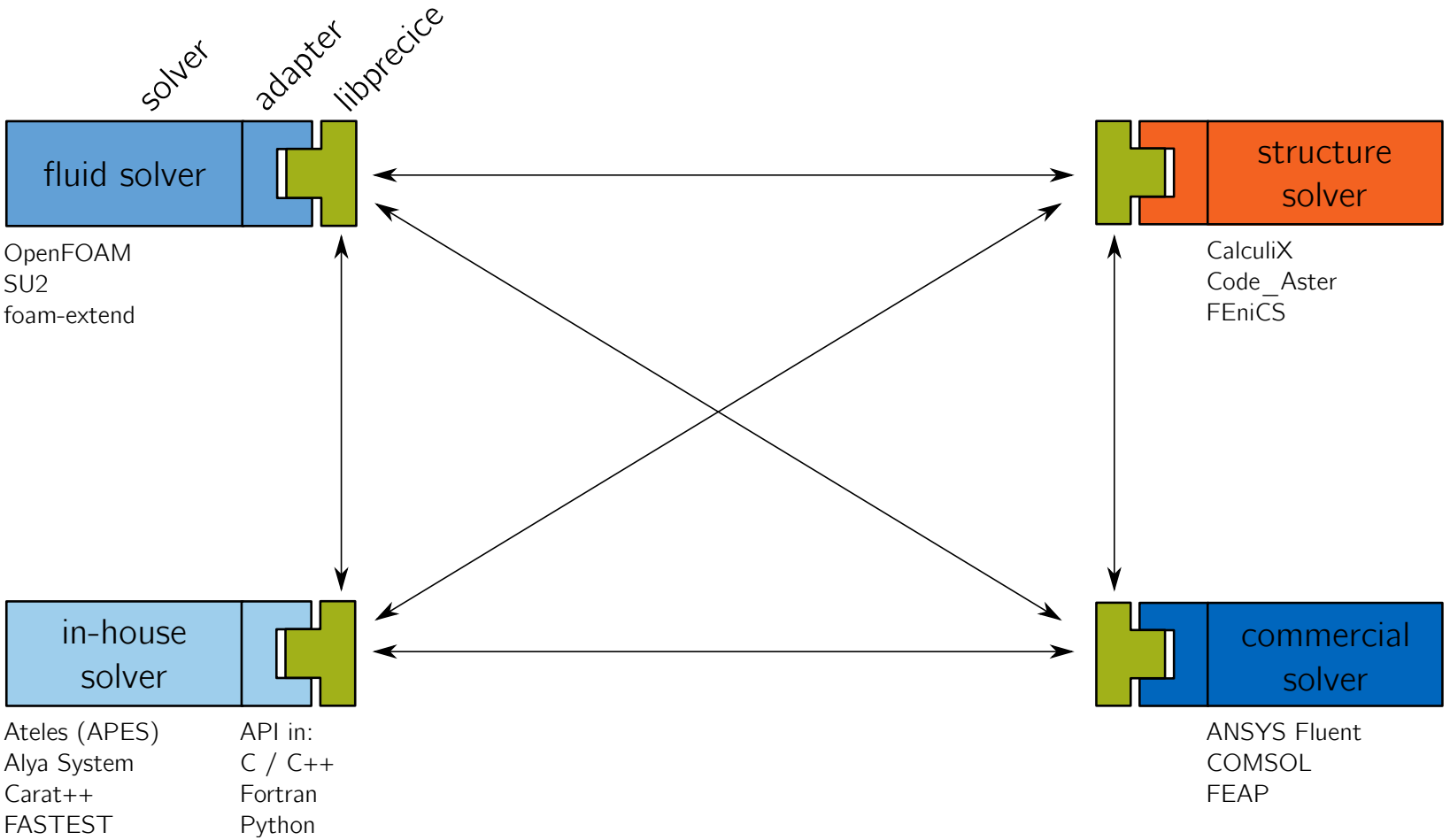
Fluid-Structure Interaction:



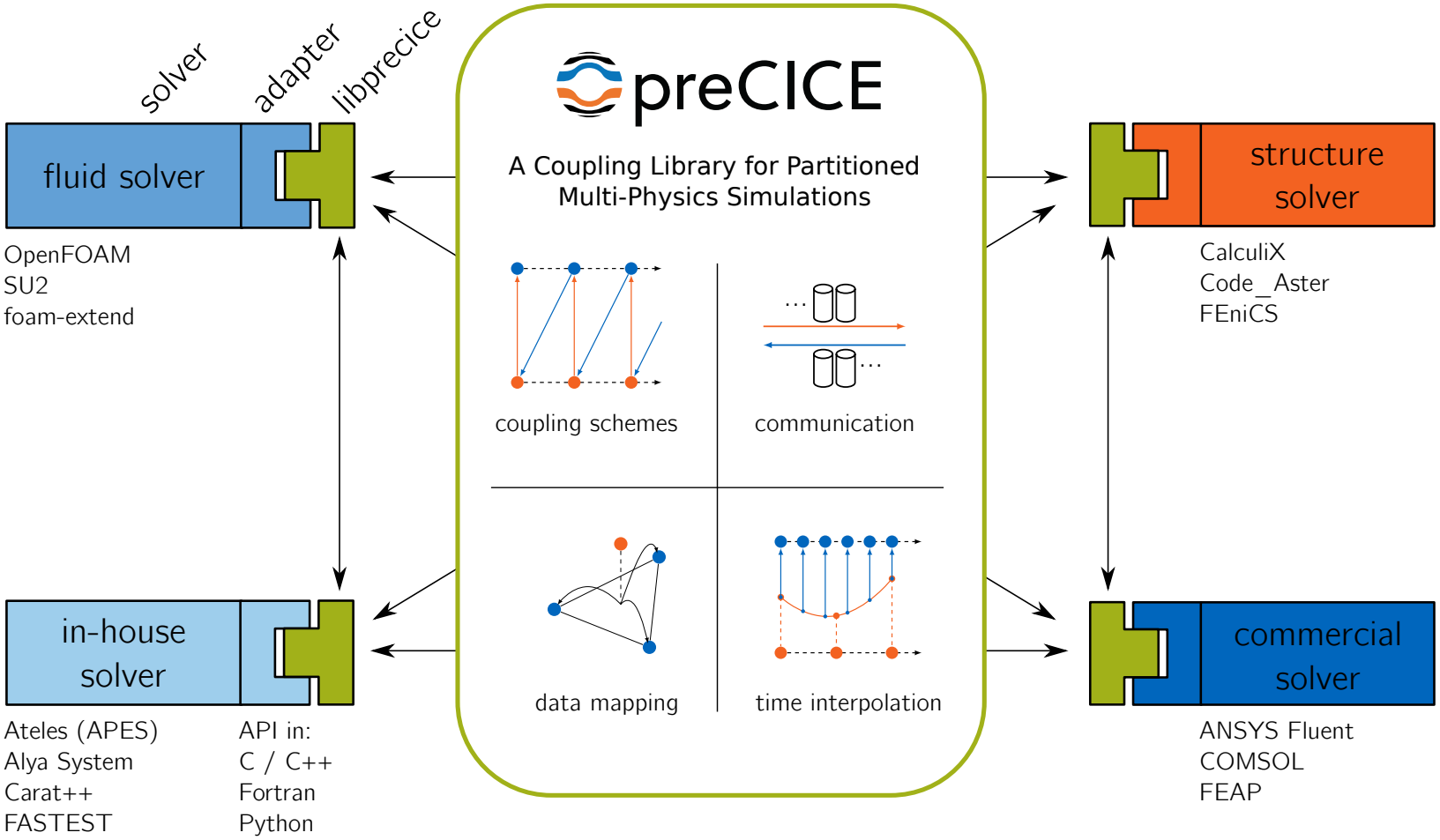
Two (coupled) simulations:



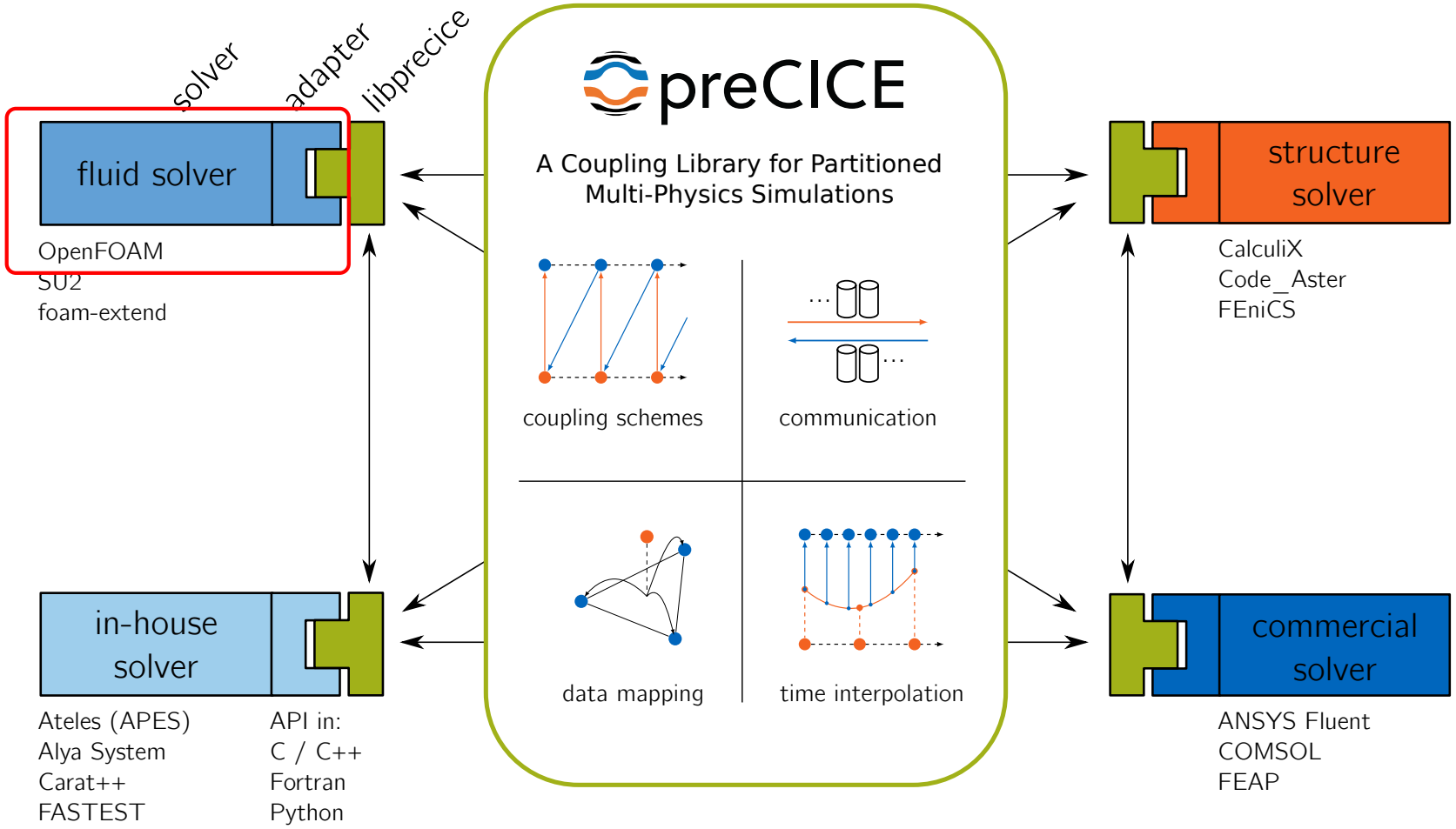
Overview


solver
adapter
libprecice
fluid solver
OpenFOAM
SU2
foam-extend
structure solver
CalculiX
Code_Aster
FEniCS
in-house solver
Ateles (APES)
Alya System
Carat++
FASTEST
API in:
C / C++
Fortran
Python
commercial solver
ANSYS Fluent
COMSOL
FEAP

Overview


solver
adapter
libprecice
fluid solver
OpenFOAM
SU2
foam-extend
in-house solver
Ateles (APES)
Alya System
Carat++
FASTEST
API in:
C / C++
Fortran
Python
preCICE
A Coupling Library for Partitioned Multi-Physics Simulations
coupling schemes
communication
data mapping
time interpolation
structure solver
CalculiX
Code_Aster
FEniCS
commercial solver
ANSYS Fluent
COMSOL
FEAP

Overview


solver
adapter
libprecice
fluid solver
OpenFOAM
SU2
foam-extend
preCICE
A Coupling Library for Partitioned Multi-Physics Simulations
coupling schemes
communication
data mapping
time interpolation
structure solver
CalculiX
Code_Aster
FEniCS
in-house solver
Ateles (APES)
Alya System
Carat++
FASTEST
API in:
C / C++
Fortran
Python
commercial solver
ANSYS Fluent
COMSOL
FEAP

How to couple my own solver?

```
precice::SolverInterface precice("FluidSolver",rank,size);
precice.configure("precice-config.xml");
precice.setMeshVertices();
precice.initialize();

while (precice.isCouplingOngoing()) { // main time loop
    solve();

    precice.writeBlockVectorData();
    precice.advance();
    precice.readBlockVectorData();

    endTimeStep(); // e.g. write results, increase time
}

precice.finalize();
```

Timesteps, most arguments and less important methods omitted. Full example in the wiki.

Adapting an OpenFOAM solver

```
/* Start the solver */


Info<< "\nStarting time loop\n" << endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"


  runTime++;


  /* continue --> */
```

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


  runTime.write();
}


/* Finalize */
```

Adapting an OpenFOAM solver

```
/* Start the solver */
/* Adapter: Initialize coupling
      calls precice->initialize() */
adapter.initialize();

Info<< "\nStarting time loop\n" << endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"




  runTime++;


  /* continue --> */
```

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




  runTime.write();
}


/* Finalize */
```

Adapting an OpenFOAM solver

```
/* Start the solver */
/* Adapter: Initialize coupling
      calls precice->initialize() */
adapter.initialize();

Info<< "\nStarting time loop\n" << endl;
while (adapter.isCouplingOngoing()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"




  runTime++;



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









  runTime.write();
}


/* Finalize */
```

Adapting an OpenFOAM solver

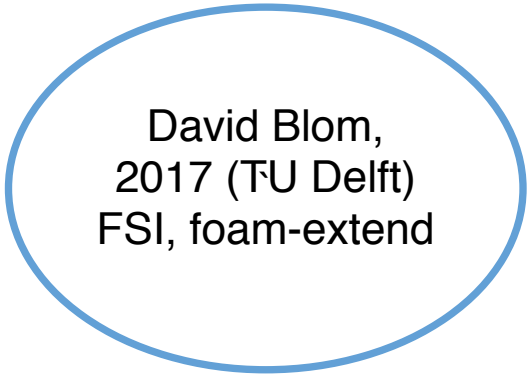
```
/* Start the solver */
/* Adapter: Initialize coupling
      calls precice->initialize() */
adapter.initialize();

Info<< "\nStarting time loop\n" << endl;
while (adapter.isCouplingOngoing()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"




  runTime++;

  /* Adapter: Receive coupling data */
  adapter.readCouplingData();
  /* continue --> */
  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

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

  /* Adapter: advance the coupling
        calls precice->advnace()  */
  adapter.advance();

  runTime.write();
}


/* Finalize */
```

An adapted OpenFOAM solver

```
/* Start the solver */
/* Adapter: Initialize coupling
      calls precice->initialize() */
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();

  runTime++;

  /* Adapter: Receive coupling data */
  adapter.readCouplingData();
  /* continue --> */
  /* solve the equations */
  #include "rhoEqn.H"
  while (pimple.loop())
  {
      ...
  }

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

  /* Adapter: advance the coupling
        calls precice->advnace()  */
  adapter.advance();

  runTime.write();
}


/* Finalize */
```

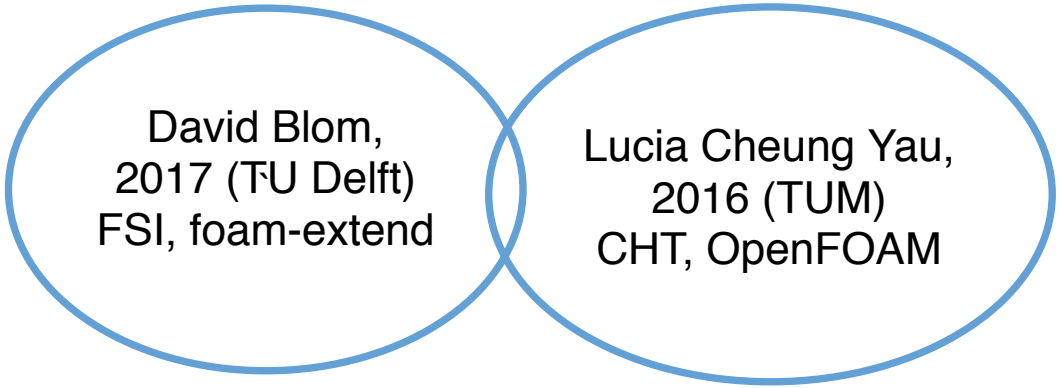
Duplicated development effort

OpenFOAM (and family) adapters for preCICE

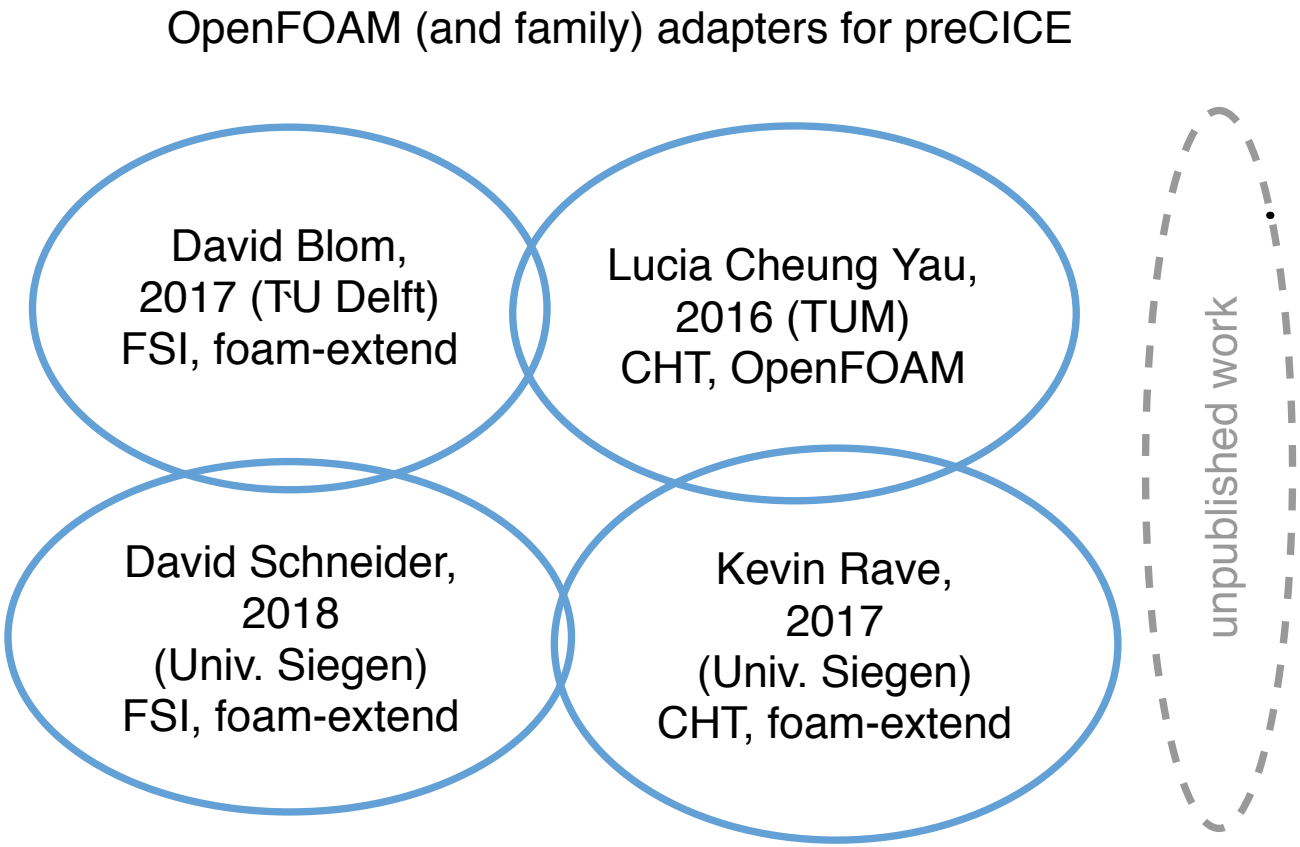


Duplicated development effort

OpenFOAM (and family) adapters for preCICE

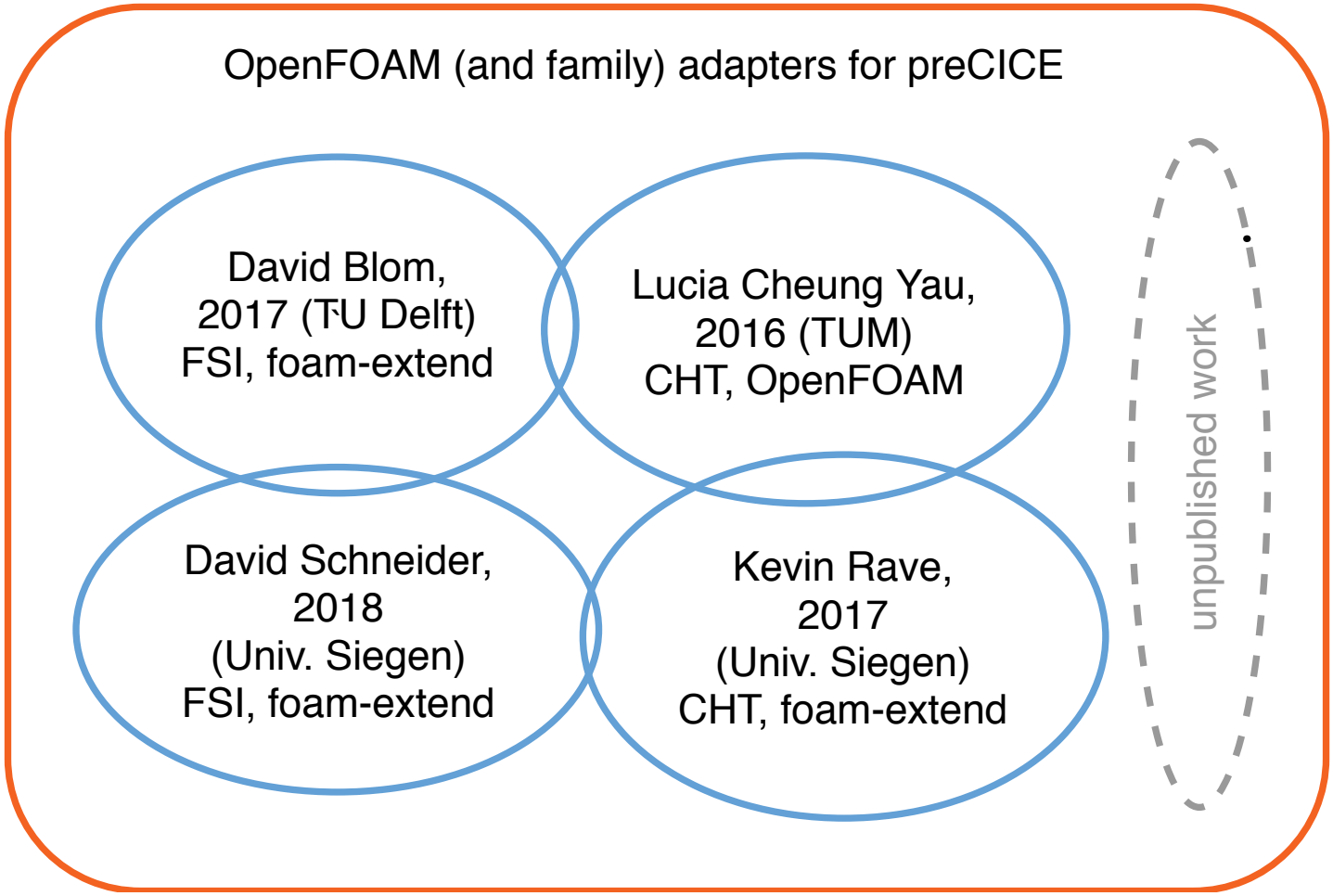


Duplicated development effort



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!

Before: Working and validated prototypes

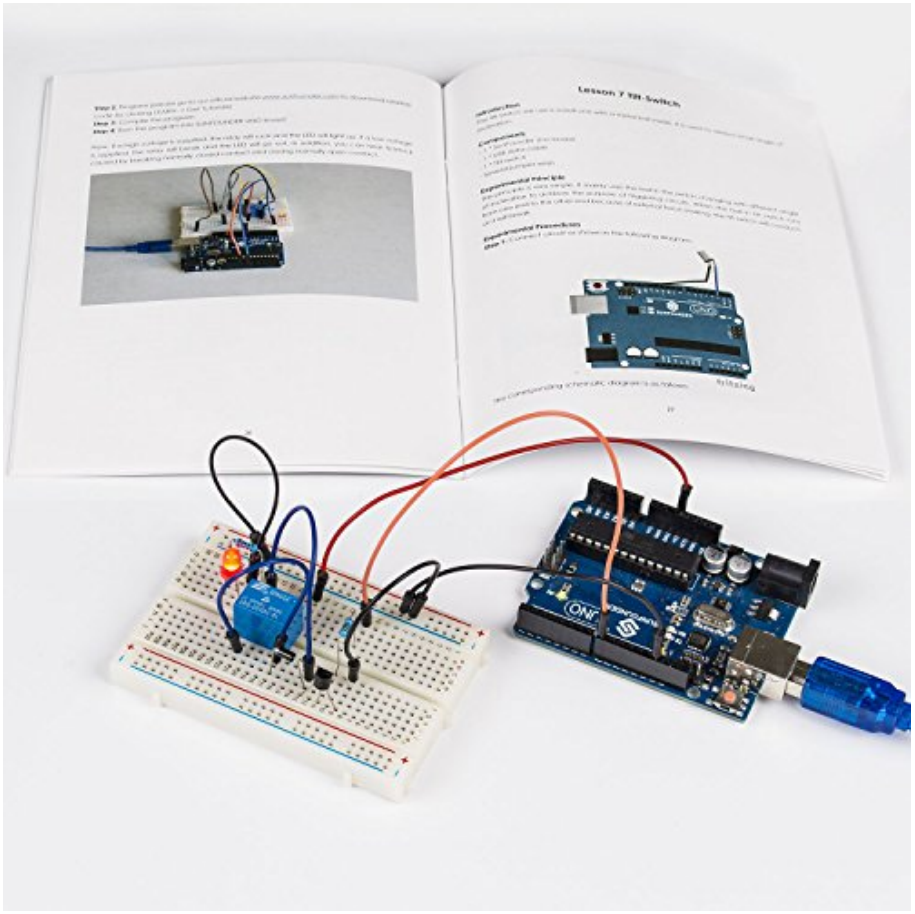


Image from desertcart.ae.

Before: Working and validated prototypes

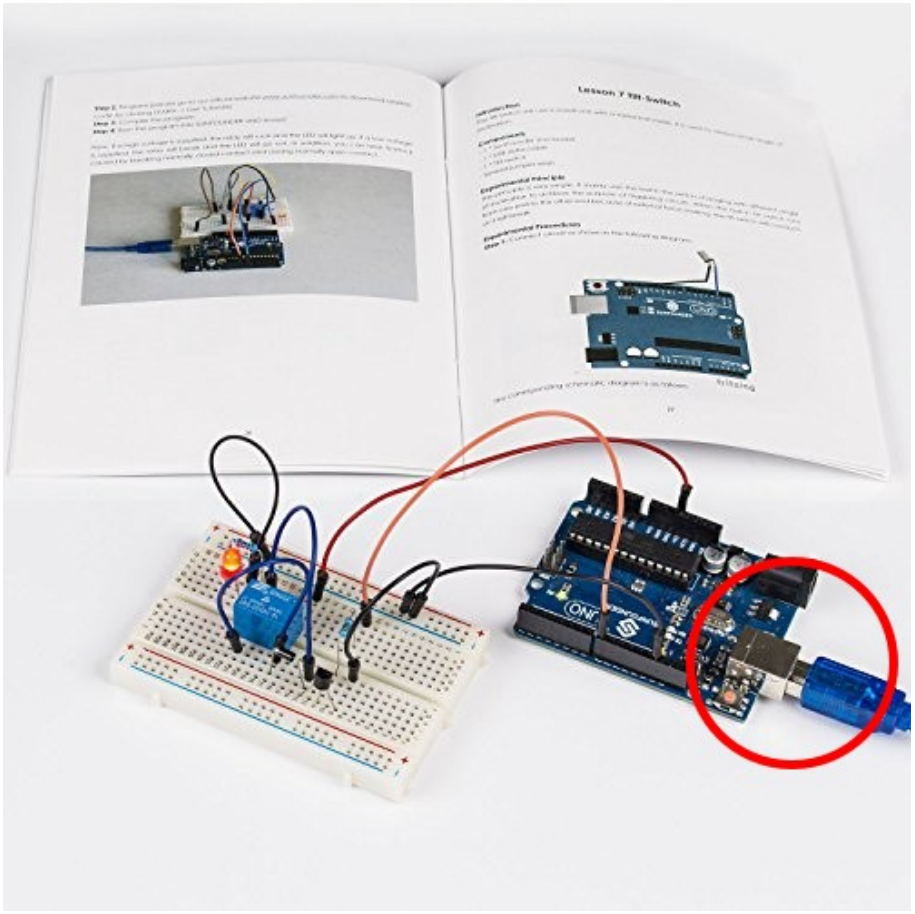
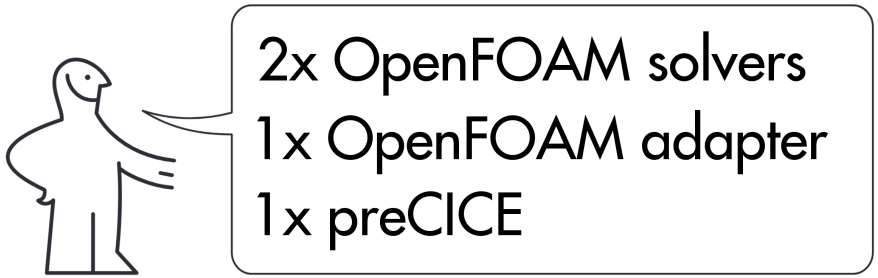


Image from desertcart.ae.

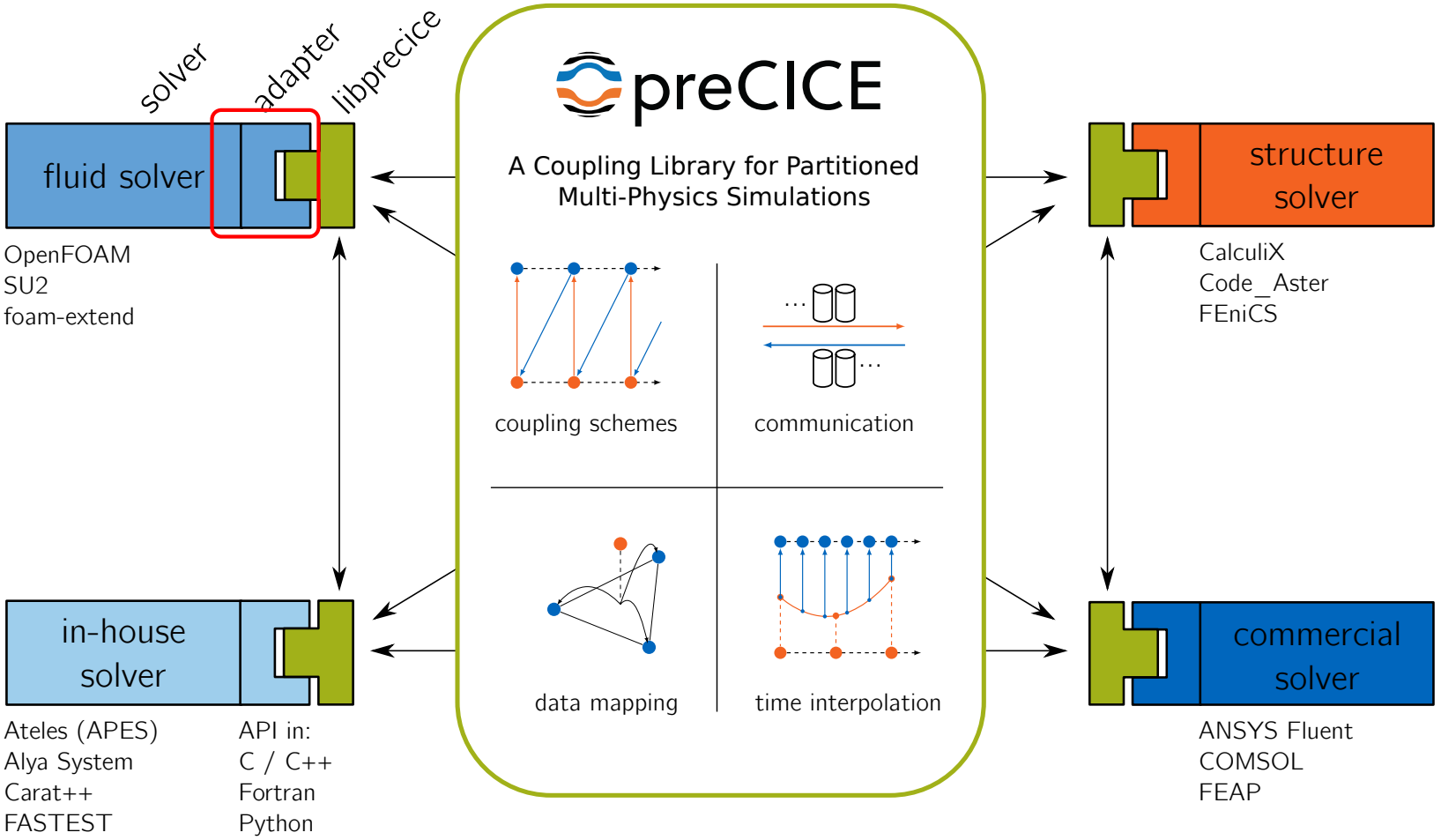
Now: A user-friendly, plug-and-play adapter

KOPPLAD



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

Overview


solver
adapter
libprecice
fluid solver
OpenFOAM
SU2
foam-extend
preCICE
A Coupling Library for Partitioned
Multi-Physics Simulations
coupling schemes
communication
data mapping
time interpolation
structure
solver
CalculiX
Code_Aster
FEniCS
in-house
solver
Ateles (APES)
Alya System
Carat++
FASTEST
API in:
C / C++
Fortran
Python
commercial
solver
ANSYS Fluent
COMSOL
FEAP

Isolating the adapter: Function Objects

```
/* Start the solver */

Info<<"\nStarting time loop\n"<< endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  runTime++;

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

  runTime.write();
}

/* Finalize */
```

Isolating the adapter: Function Objects

```
/* Start the solver */

Info<<"\nStarting time loop\n"<< endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  runTime++;

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

  runTime.write();
}

/* Finalize */
```

```
// system/controlDict OpenFOAM config file
functions
{
   preCICE_Adapter
   {
     type preciceAdapterFunctionObject;
     libs ("libpreciceAdapterFunctObj.so");
   }
}
```

Isolating the adapter: Function Objects

```
/* Start the solver */

Info<<"\nStarting time loop\n"<< endl;
while (runTime.run()) {
  #include "readTimeControls.H"
  #include "compressibleCourantNo.H"
  #include "setDeltaT.H"

  runTime++;

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

  runTime.write();
}

/* Finalize */
```

```
// system/controlDict OpenFOAM config file
functions
{
   preCICE_Adapter
   {
     type preciceAdapterFunctionObject;
     libs ("libpreciceAdapterFunctObj.so");
   }
}
```

```
// 0/T OpenFOAM config file
interface
{
    type            fixedValue;
    value           uniform 300;
}
// other types: fixedGradient, mixed
```

Coupling boundary patches, problem & solver type: `precice-adapter-config.yml`

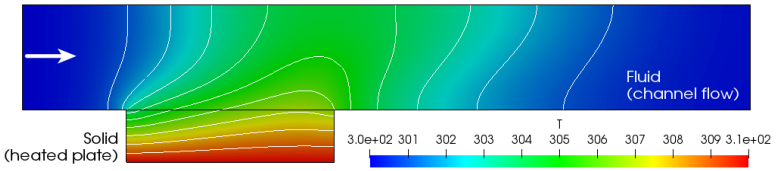
Todo: Convert to an OpenFOAM dictionary.

Tutorials

On `www.precice.org/resources` (step-by-step):

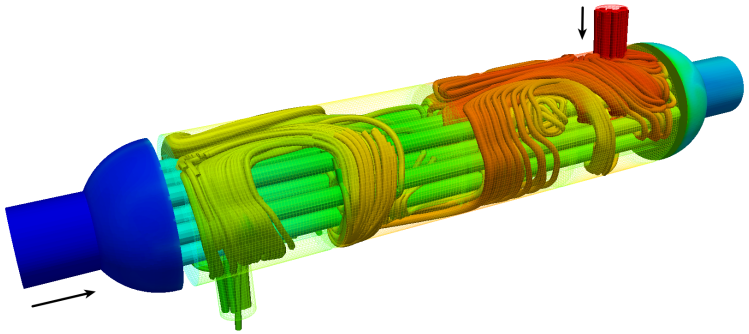
Flow above a heated plate

- Demo case, bundled with the adapter
- buoyantPimpleFoam + laplacianFoam
- Learn how to use the OpenFOAM adapter



Shell-and-Tube Heat Exchanger

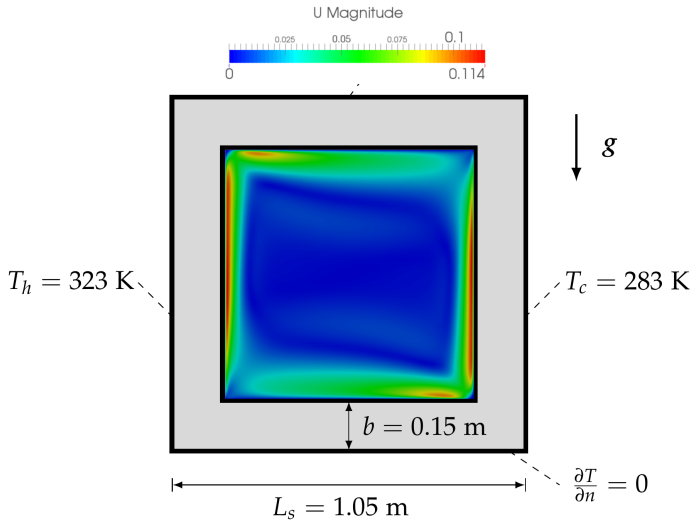
- Larger case, in `precice/tutorials`
- buoyantSimpleFoam (x2) + CalculiX
- Learn how to do multi-coupling

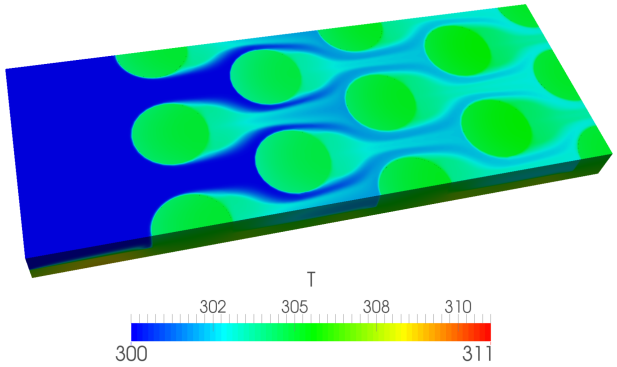


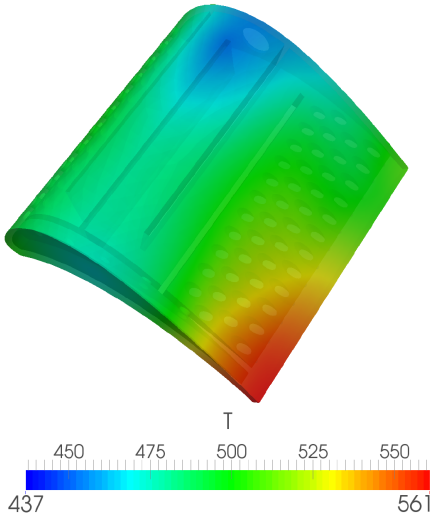
More CHT examples with OpenFOAM

- Natural convection cavity
  - OpenFOAM + CalculiX
  - Transient
  - Robin-Robin serial-implicit coupling, IQN-ILS
- Pin-Fin cooling system
  - OpenFOAM + CalculiX
  - Steady-state
  - Robin-Robin parallel-implicit coupling, IQN-ILS
- Cooling of a turbine blade
  - OpenFOAM + CalculiX (or Code_Aster)
  - Steady-state
  - Robin-Robin parallel-explicit coupling

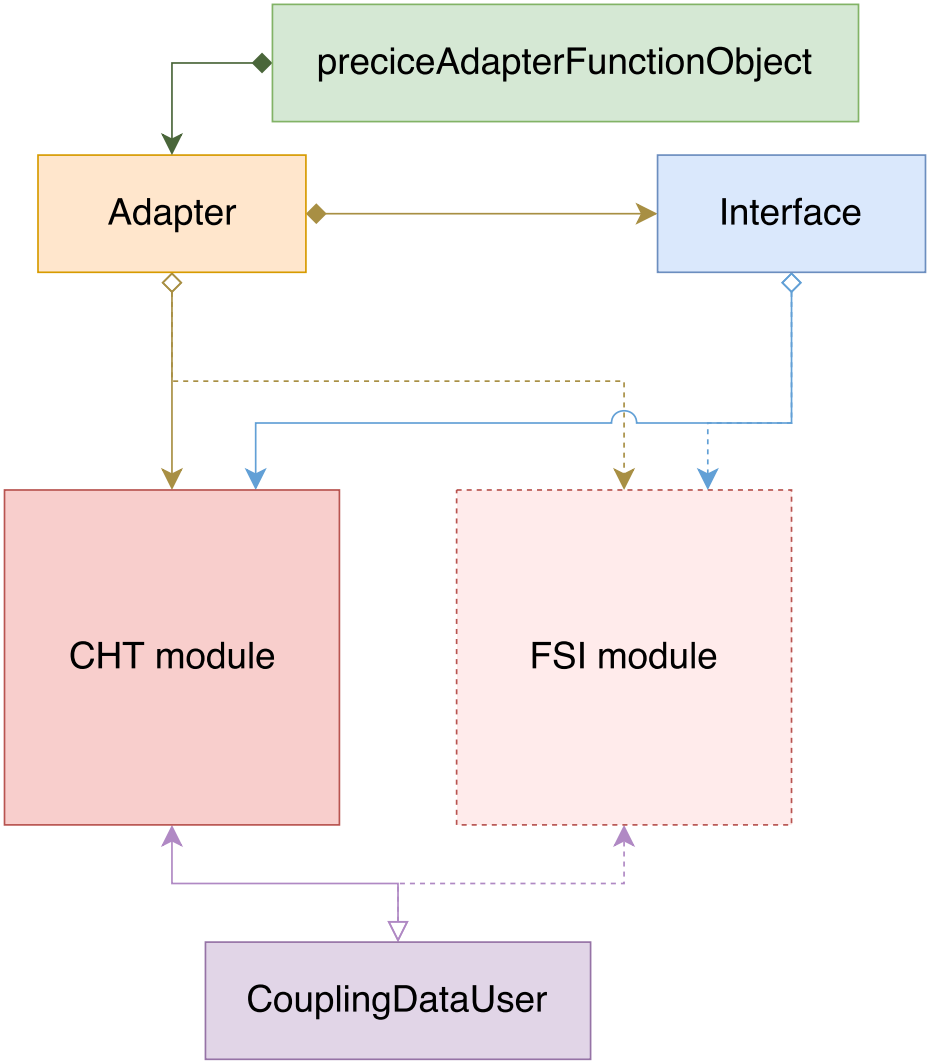
(simulations by L. Cheung Yau, 2016)





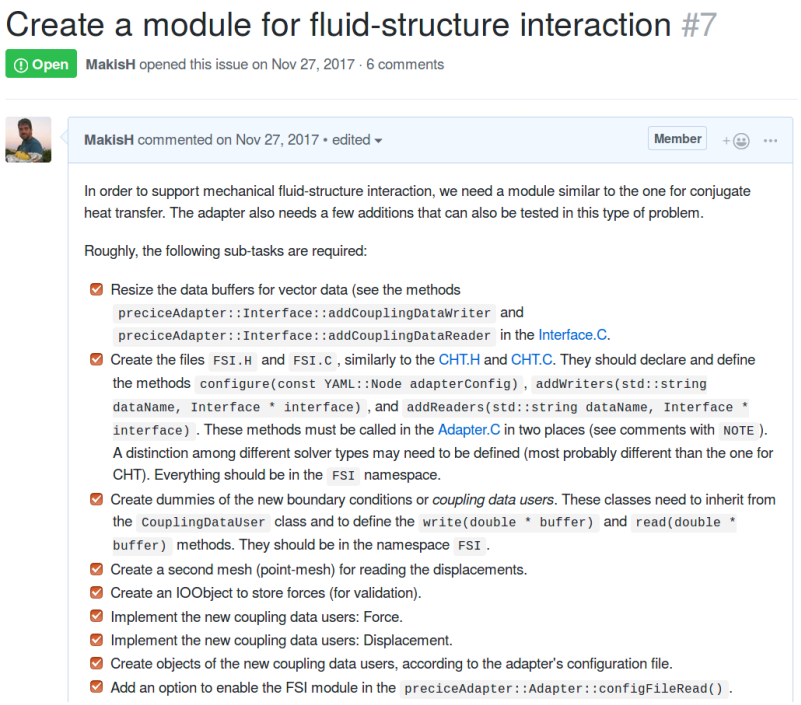


A modular design



Adding new features: Fluid-Structure Interaction

```
|openfoam-adapter/
|- preciceAdapterFunctionObject.C
|- Adapter.C
|- CouplingDataUser.C
|- CHT/
|   |- CHT.C
|   |- HeatFlux.C
|   |- HeatTransferCoefficient.C
|   |- KappaEffective.C
|   |- SinkTemperature.C
|   |- Temperature.C
|   |- ...
|- FSI/
|   |- Displacement.C
|   |- Force.C
|   |- FSI.C
|   |- ...
|- Interface.C
|- Utilities.C
...
```



Create a module for fluid-structure interaction #7

Open **MakisH** opened this issue on Nov 27, 2017 · 6 comments

MakisH commented on Nov 27, 2017 • edited — Member

In order to support mechanical fluid-structure interaction, we need a module similar to the one for conjugate heat transfer. The adapter also needs a few additions that can also be tested in this type of problem.

Roughly, the following sub-tasks are required:

- [x] Resize the data buffers for vector data (see the methods `preciceAdapter::Interface::addCouplingDataWriter` and `preciceAdapter::Interface::addCouplingDataReader` in the Interface.C.
- [x] Create the files `FSI.H` and `FSI.C`, similarly to the CHT.H and CHT.C. They should declare and define the methods `configure(const YAML::Node adapterConfig)`, `addWriters(std::string dataName, Interface * interface)`, and `addReaders(std::string dataName, Interface * interface)`. These methods must be called in the Adapter.C in two places (see comments with `NOTE`). A distinction among different solver types may need to be defined (most probably different than the one for CHT). Everything should be in the `FSI` namespace.
- [x] Create dummies of the new boundary conditions or *coupling data users*. These classes need to inherit from the `CouplingDataUser` class and to define the `write(double * buffer)` and `read(double * buffer)` methods. They should be in the namespace `FSI`.
- [x] Create a second mesh (point-mesh) for reading the displacements.
- [x] Create an IOObject to store forces (for validation).
- [x] Implement the new coupling data users: Force.
- [x] Implement the new coupling data users: Displacement.
- [x] Create objects of the new coupling data users, according to the adapter's configuration file.
- [x] Add an option to enable the FSI module in the `preciceAdapter::Adapter::configFileRead()`.

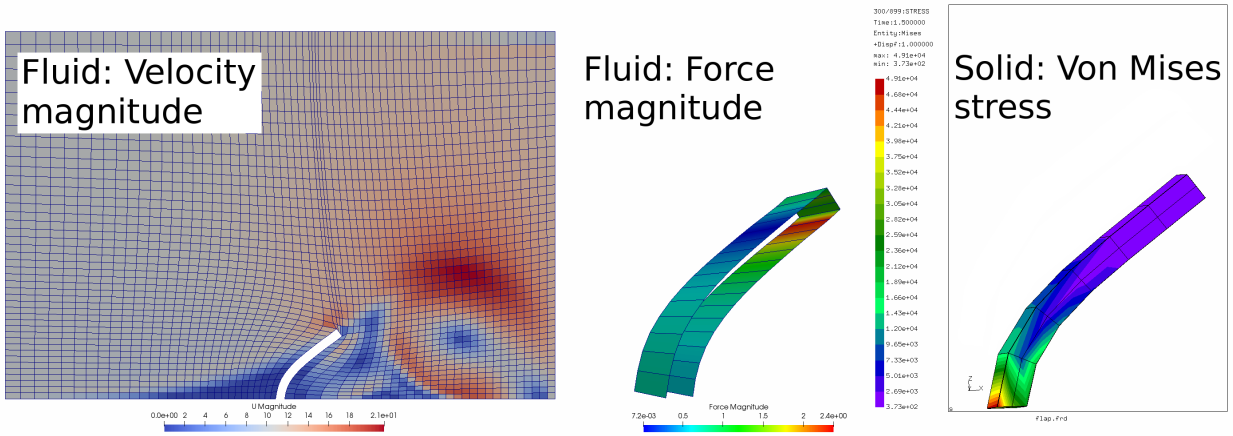
You can contribute! Comment, open issues, submit pull requests.

Fluid-Structure Interaction: first results

Simulations developed in the context of the M.Sc. thesis of Derek Risseeuw, TU Delft.

Flow in a channel with a perpendicular flap

- pimpleDyMFoam + CalculiX
- Will add as a tutorial in our wiki (already for compressible SU2 & CalculiX)

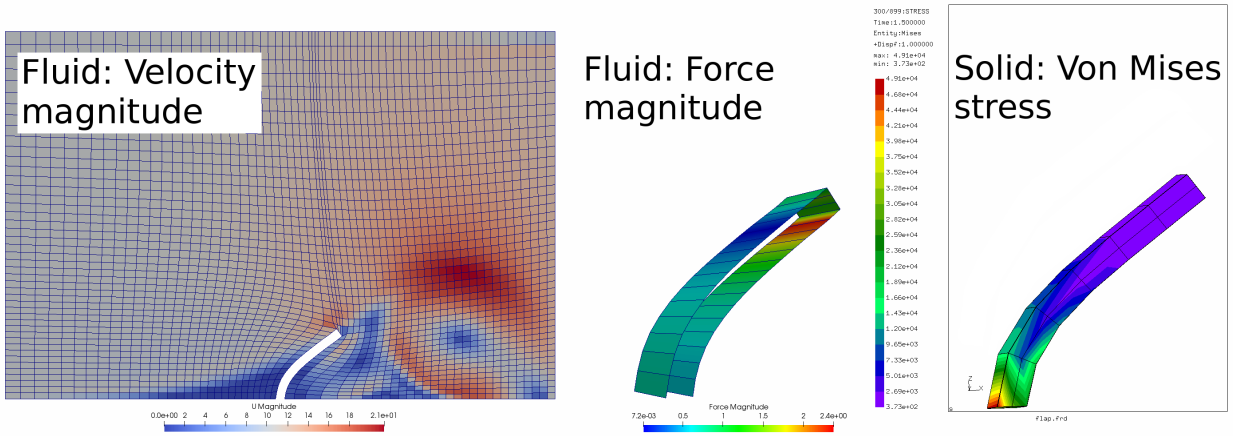


Fluid-Structure Interaction: first results

Simulations developed in the context of the M.Sc. thesis of Derek Risseeuw, TU Delft.

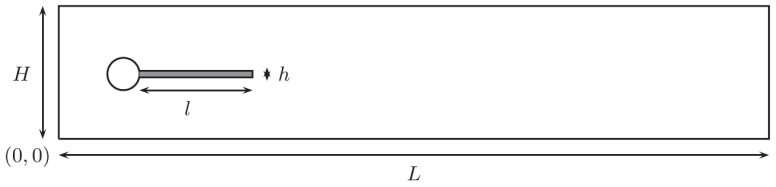
Flow in a channel with a perpendicular flap

- pimpleDyMFoam + CalculiX
- Will add as a tutorial in our wiki (already for compressible SU2 & CalculiX)



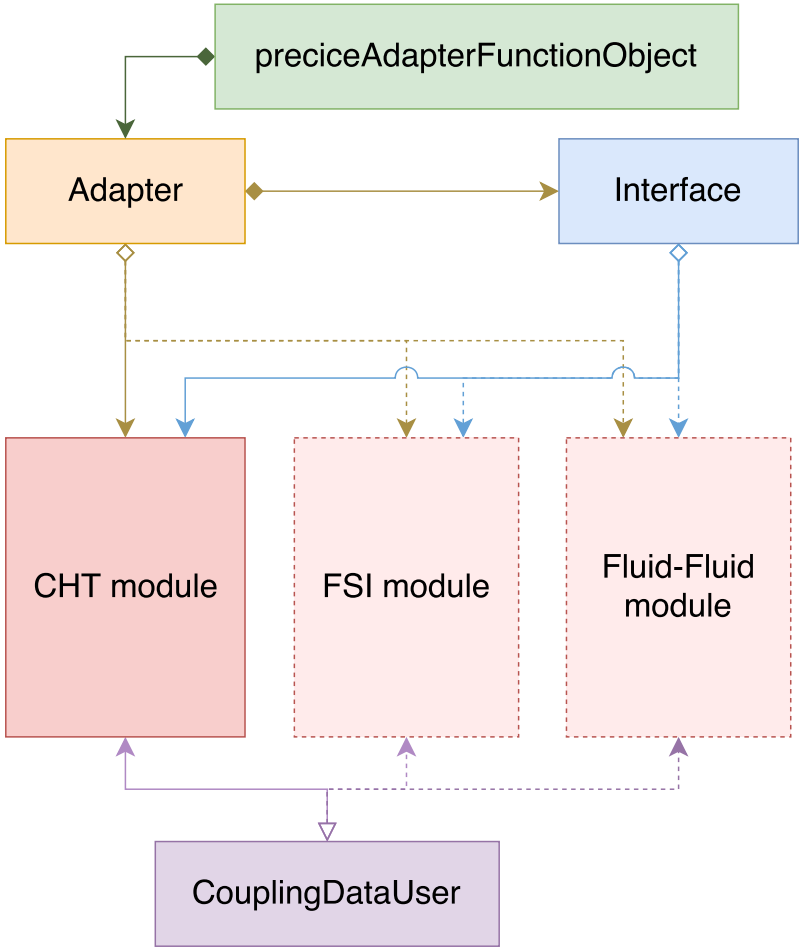
Current work: Turek-Hron FSI3 Benchmark

- Standard case for incompressible FSI
- pimpleDyMFoam + CalculiX



From S. Turek and J. Hron (2006): Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow.

Planned: Fluid-Fluid coupling



Long-term plan: couple 1D – 3D or 2D – 3D solvers.

Does it work with “chocolate” OpenFOAM?

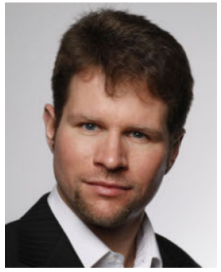
Known to work with:
ESI - OpenCFD: v1706 – v1806
The OpenFOAM Foundation: 4.0 – dev

Currently does not work with:
ESI - OpenCFD: $\leq$ v1606+
The OpenFOAM Foundation: $\leq$ 3.0
FOAM-extend: any version

Question: How to support multiple flavors and versions with the same code in the long term?
(please comment on issue #32)

Contributors

Miriam Mehl
U Stuttgart



Florian Lindner
U Stuttgart

Amin Totounferoush
U Stuttgart

Alexander Rusch
ETH Zürich

Hans Bungartz
TUM

Benjamin Rüth
TUM

Gerasimos Chourdakis
TUM

Frédéric Simonis
TUM

Benjamin Uekermann
TUM

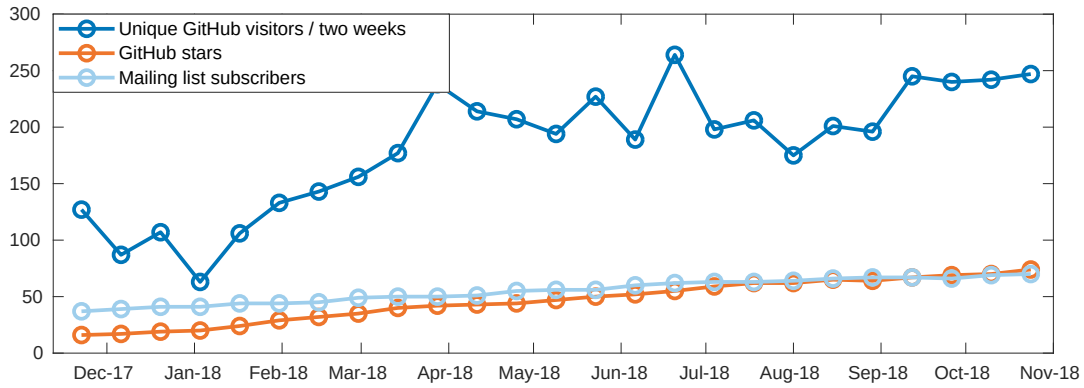
Previous and additional contributors: Bernhard Gatzhammer, Klaudius Scheufele, Lucia Cheung, Alexander Shukaev, Peter Vollmer, Georg Abrams, Alex Trujillo, Dmytro Sashko, David Sommer, David Schneider, Derek Risseeuw, ...

Users: A growing community

- LSM & STS, U Siegen, Germany
- SC & FNB, TU Darmstadt, Germany
- SCpA, CIRA, Italy
- Cardiothoracic Surgery, University of the Free State, South Africa
- A*STAR, Singapore
- CASA, TU Eindhoven, The Netherlands
- Nuclear Research and Consultancy Group, Petten, The Netherlands
- Aerospace Engineering, TU Delft, NL
- Mechanical and Aeronautical Eng., University of Manchester, UK
- University of Strathclyde, Glasgow, UK

Upcoming:

- Global Research for Safety (GRS), Garching, Germany
- MTU Aero Engines, Munich, Germany
- TU Braunschweig, Inst. of Aircraft Design and Lightweight Structures, Germany
- FAST, Karlsruhe Institute of Technology, Germany
- Numerical Analysis, Lund, Sweden
- Austrian Inst. of Techn., Vienna, Austria
- IAG, University of Stuttgart, Germany
- Polytechnique Montreal, Canada



Users: A growing community

preCICE is even where we don't expect it to be!

держивает программную модель, близкую к MPI и позволяет работать с неструктурированны-
ми сетками и планировщиками коммуникаций, обеспечивающие параллельные обмены между
матрицами приложений MxN.

FEStudio [2] – это российская разработка для решения задач взаимодействия конструкции с жидкостью или газом. Программная модель представляет собой клиент-серверную архитектуру, использующую программное обеспечение ICE (https://zeroc.com). Сопрягаемые приложения могут работать на несогласованных сетках. Поддерживает итерационную процедуру согласования данных в пределах одного временного шага.

preCICE [3] – это программное обеспечение для решения задач взаимодействия твердого тела с жидкостью или газом. Может работать только на структурированных сетках. Для коммуникаций использует MPI.

---

* Работа поддержана грантом РНФ 14-31-00024.

566

Take-away

Flexible: Couple your own solver with any other, for example
OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...

Take-away

Flexible: Couple your own solver with any other, for example OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...

Easy: Adapt your solver with around 30 extra lines of code

Take-away

Flexible: Couple your own solver with any other, for example OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...

Easy: Adapt your solver with around 30 extra lines of code

Ready: Out-of-the box support for OpenFOAM, CalculiX, SU2, ...

Take-away

Flexible:	Couple your own solver with any other, for example OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...
Easy:	Adapt your solver with around 30 extra lines of code
Ready:	Out-of-the box support for OpenFOAM, CalculiX, SU2, ...
Fast:	Fully parallel, peer-to-peer, designed for HPC (SPPEXA)

Take-away

Flexible:	Couple your own solver with any other, for example OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...
Easy:	Adapt your solver with around 30 extra lines of code
Ready:	Out-of-the box support for OpenFOAM, CalculiX, SU2, ...
Fast:	Fully parallel, peer-to-peer, designed for HPC (SPPEXA)
Stable:	Implicit coupling, accelerated with Quasi-Newton

Take-away

Flexible:	Couple your own solver with any other, for example OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...
Easy:	Adapt your solver with around 30 extra lines of code
Ready:	Out-of-the box support for OpenFOAM, CalculiX, SU2, ...
Fast:	Fully parallel, peer-to-peer, designed for HPC (SPPEXA)
Stable:	Implicit coupling, accelerated with Quasi-Newton
Multi-coupling:	Couple more than two solvers

Take-away

Flexible:	Couple your own solver with any other, for example OpenFOAM – ABAQUS, OpenFOAM – FOAM-extend, ...
Easy:	Adapt your solver with around 30 extra lines of code
Ready:	Out-of-the box support for OpenFOAM, CalculiX, SU2, ...
Fast:	Fully parallel, peer-to-peer, designed for HPC (SPPEXA)
Stable:	Implicit coupling, accelerated with Quasi-Newton
Multi-coupling:	Couple more than two solvers
Free:	LGPL3, source on GitHub

preCICE

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

Contact us: `precice.org/resources`
My e-mail: `chourdak@in.tum.de`

See also: Mailing-list, Gitter, Twitter, ResearchGate, Literature guide in our wiki

References

preCICE **preCICE – A Fully Parallel Library for Multi-Physics Surface Coupling**
H.-J. Bungartz, B. Gatzhammer, F. Lindner, M. Mehl, K. Scheufele, A. Shukaev, B. Uekermann, 2016
In Computers and Fluids, Volume 141, p. 250—258. Elsevier.

Adapters **Official preCICE Adapters for Standard Open-Source Solvers**
B. Uekermann, H.-J. Bungartz, L. Cheung Yau, G. Chourdakis, A. Rusch, 2017
7th GACM Colloquium on Computational Mechanics for Young Scientists from Academia

Thesis 2 **A general OpenFOAM adapter for the coupling library preCICE**
G. Chourdakis, 2017
Master's thesis, Institut für Informatik, Technische Universität München

Thesis 1 **Conjugate Heat Transfer with the Multiphysics Coupling Library preCICE**
L. Cheung Yau, 2016
Master's thesis, Institut für Informatik, Technische Universität München

FOAM-FSI **Efficient numerical methods for partitioned fluid-structure interaction simulations**
D. Blom, 2017
Dissertation, Delft University of Technology

FOAM-extend + deal.II **Simulation von Fluid-Struktur-Interaktion mit der Kopplungsbibliothek preCICE**
D. Schneider, 2018
Bachelor's thesis, Lehrstuhl für Strömungsmechanik, Universität Siegen