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:

Part II:
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)

outer fluid — preCICE — solid — preCICE — inner fluid

buoyantSimpleFoam — CalciX — buoyantSimpleFoam
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

- Call preCICE methods
- Store checkpoints
- Manipulate time step size
- Manipulate interface data
- Perform problem-specific operations
Part IIa: previous approach

precICE + 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

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

Kevin Rave, 2017 (Univ. Siegen)
CHT, foam-extend

Lucia Cheung Yau, 2016 (TUM)
CHT, OpenFOAM

All these adapters are **bound to specific solvers!**
Duplicated development effort

OpenFOAM (and family) adapters for preCICE

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

Kevin Rave, 2017 (Univ. Siegen)  
CHT, foam-extend

Lucia Cheung Yau, 2016 (TUM)  
CHT, OpenFOAM

All these adapters are **bound to specific solvers**!  
→ We need an official, general adapter!
Example of an adapted solver (previous)

```cpp
/* 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 in buffers */
    adapter.writeCouplingData();

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

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

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

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

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

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

Gerasimos Chourdakis (TUM) | Couple OpenFOAM with any other solver using preCICE | Feb 21, 2018 | GOFUN2
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**

2x OpenFOAM solvers
1x OpenFOAM adapter
1x preCICE

The human-like figure is a property of ikea.com.
Part IIb: a new, official adapter
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...

Gerasimos Chourdakis (TUM) | Couple OpenFOAM with any other solver using preCICE | Feb 21, 2018 | GOFUN2
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

```plaintext
// 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
```

Gerasimos Chourdakis (TUM) | Couple OpenFOAM with any other solver using preCICE | Feb 21, 2018 | GOFUN2
OpenFOAM configuration

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

Set the appropriate boundary condition types:

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

Properties for incompressible solvers:

```plaintext
// 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:

```plaintext
// constant/transportProperties
k k [ 1 1 -3 -1 0 0 0 ] 100;
```
To run the simulation, just execute the solvers as usual.
## Fluid-Structure Interaction

<table>
<thead>
<tr>
<th>1D FSI Example</th>
<th>FSI with SU2 and CalculiX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow through a deformable tube</td>
<td>Flow in a channel with an elastic flap</td>
</tr>
</tbody>
</table>

## Conjugate Heat Transfer

<table>
<thead>
<tr>
<th>CHT with OpenFOAM</th>
<th>CHT with OpenFOAM and CalculiX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow above a heated plate</td>
<td>Shell-and-tube heat exchanger</td>
</tr>
</tbody>
</table>
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: ≤ 3.0
- ESI - OpenCFD: ≤ v1606+
- foam-extend: any version

Coming soon:
- Support for older versions
- Code improvements and tests
- Fluid-Structure Interaction Module
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
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
Additional slide: Time step sizes

Time step sizes and checkpointing

Subcycling

Solver time step

Coupling step $n$

Coupling step $n + 1$

Read checkpoint (implicit coupling only)
Additional slide: Function Objects

 Callbacks in OpenFOAM function objects

**Solver**

- while (runTime.loop())
- or
- while (runTime.run())

**solvers with adjustable timestep:**

- `#include "setDeltaT.H"`

**dynamic mesh solvers:**

- `mesh.update();`
- `mesh.movePoints();`

**Foam::Time**

- `loop()`
- `run()`

**Foam::functionObjectList**

- `read()`
- `start()`
- `execute()`
- `write()`
- `end()`

**Foam::functionObject**

- `read()`
- `execute()`
- `write()`
- `end()`
- `adjustTimeStep()`
- `updateMesh()`
- `movePoints()`

Gerasimos Chourdakis (TUM) | Couple OpenFOAM with any other solver using preCICE | Feb 21, 2018 | GOFUN2
The Conjugate Heat Transfer module

namespace CHT

CouplingDataUser
+ write()
+ read()

Adapter

interface

CTH
+ determineSolverType()
+ readConfig( YAML::Node )
+ configure( YAML::Node )
+ addWriters( Interface )
+ addReaders( Interface )

Temperature

Sink Temperature

HeatFlux

HeatTransferCoefficient

HeatFlux_Compressible
kappaEff_Compressible
HeatTransferCoefficient_Compressible

HeatFlux_Incompressible
kappaEff_Incompressible
HeatTransferCoefficient_Incompressible

HeatFlux_Basic
kappaEff_Basic
HeatTransferCoefficient_Basic

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