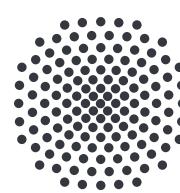


von Karman Institute for Fluid Dynamics: OpenFOAM seminar, January 24, 2025

Coupling OpenFOAM with external codes via preCICE

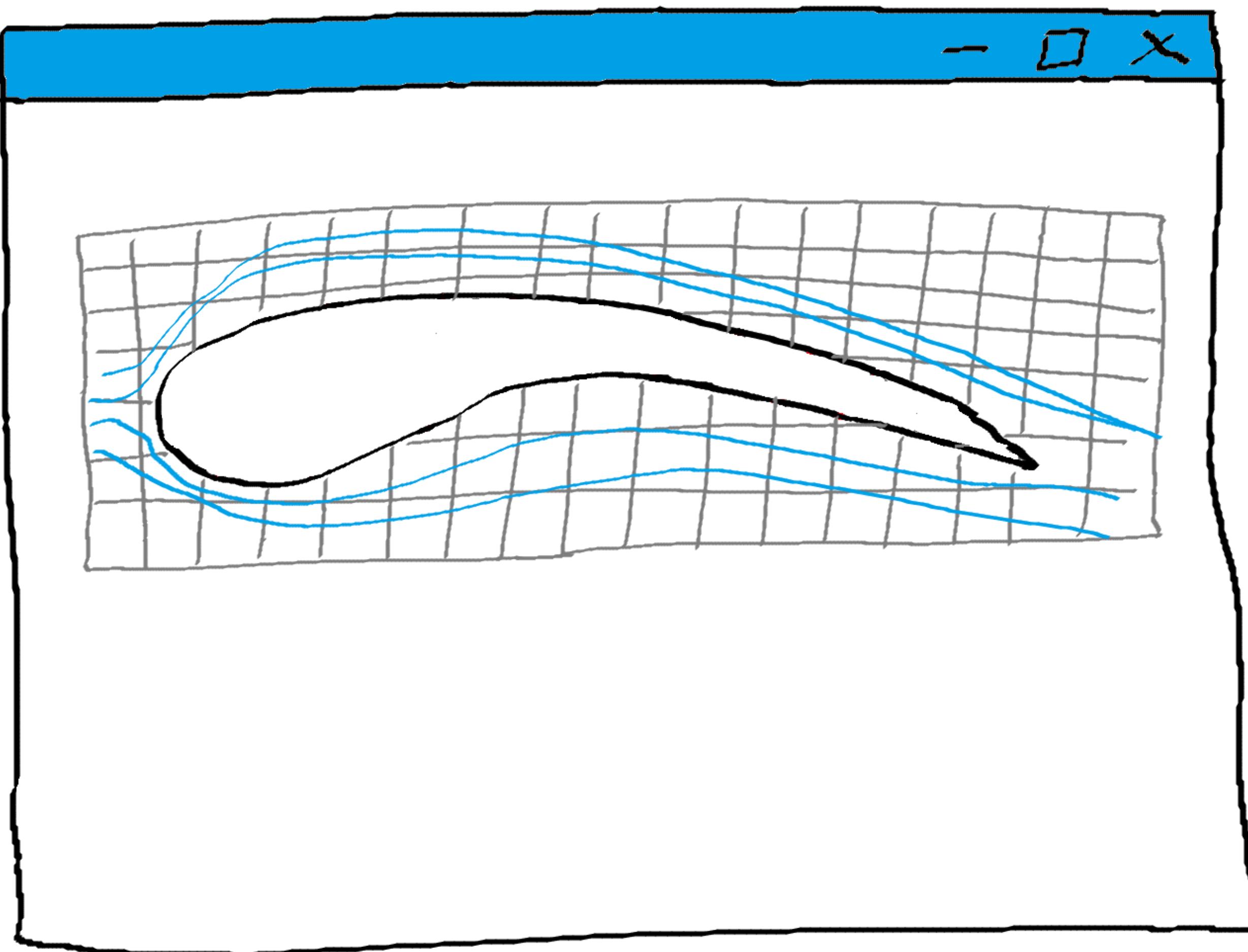
Gerasimos Chourdakis, University of Stuttgart
gerasimos.chourdakis@ipvs.uni-stuttgart.de

Jun Chen, University of Stuttgart

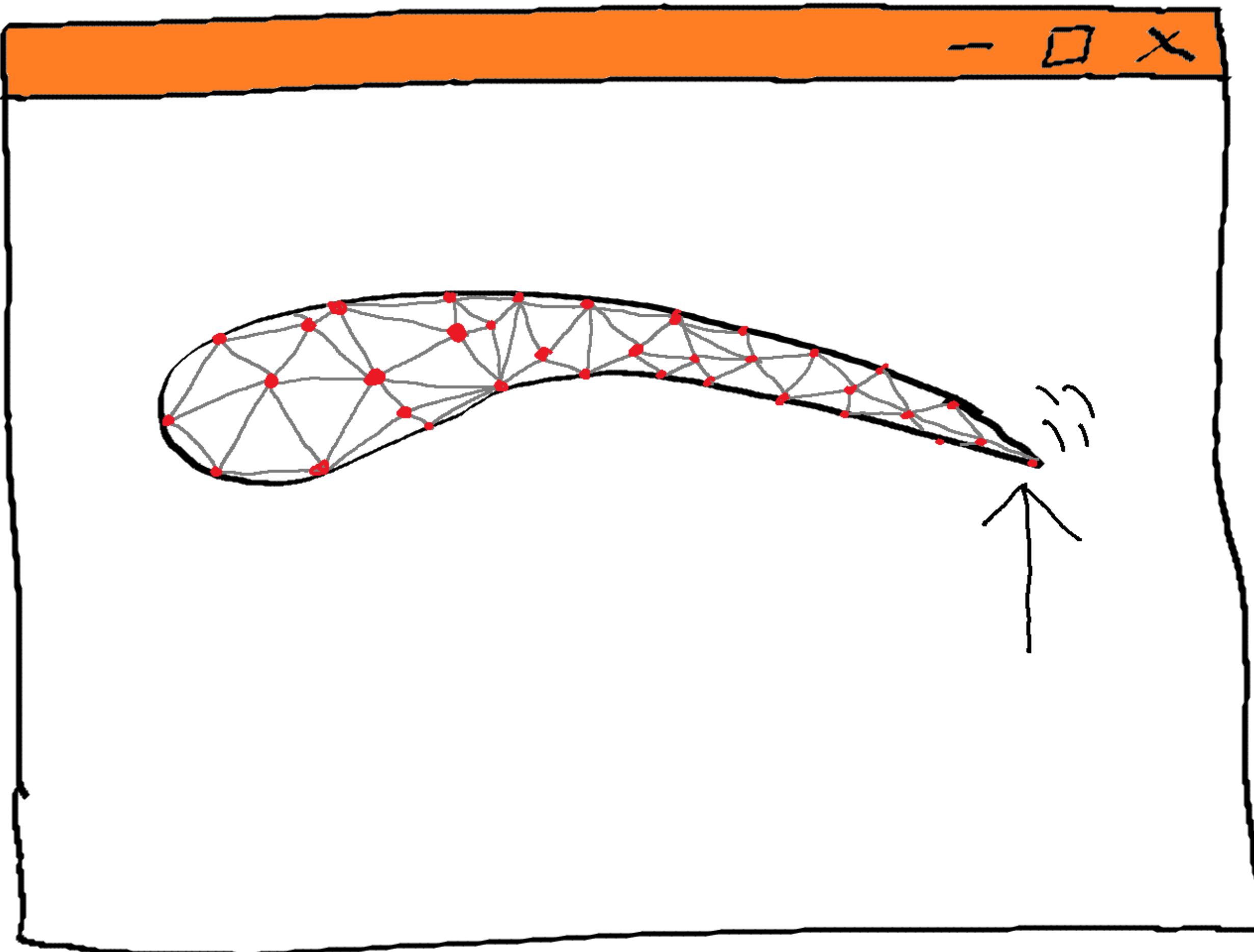


University of Stuttgart
Germany

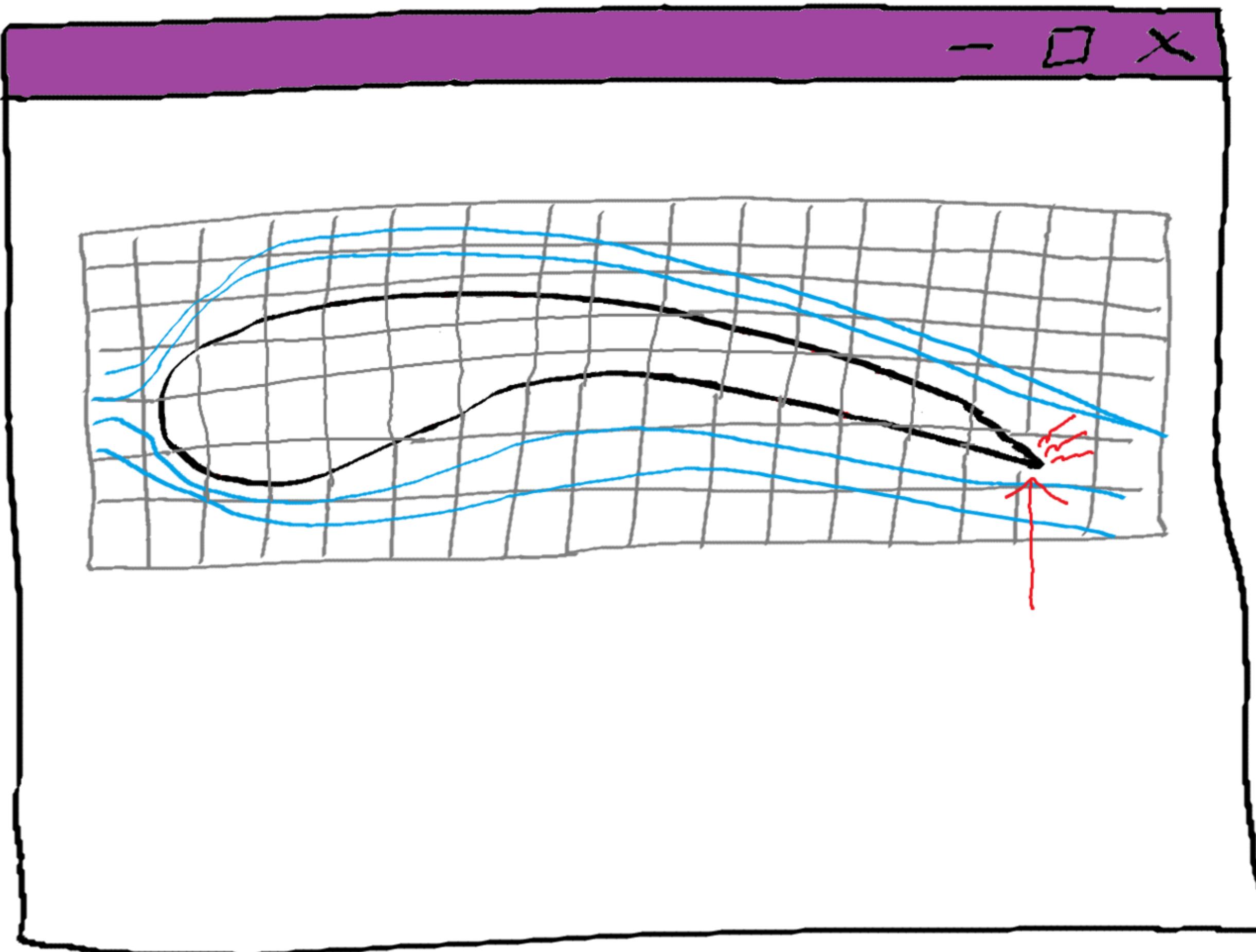
CFD: Simulating flow around a wing



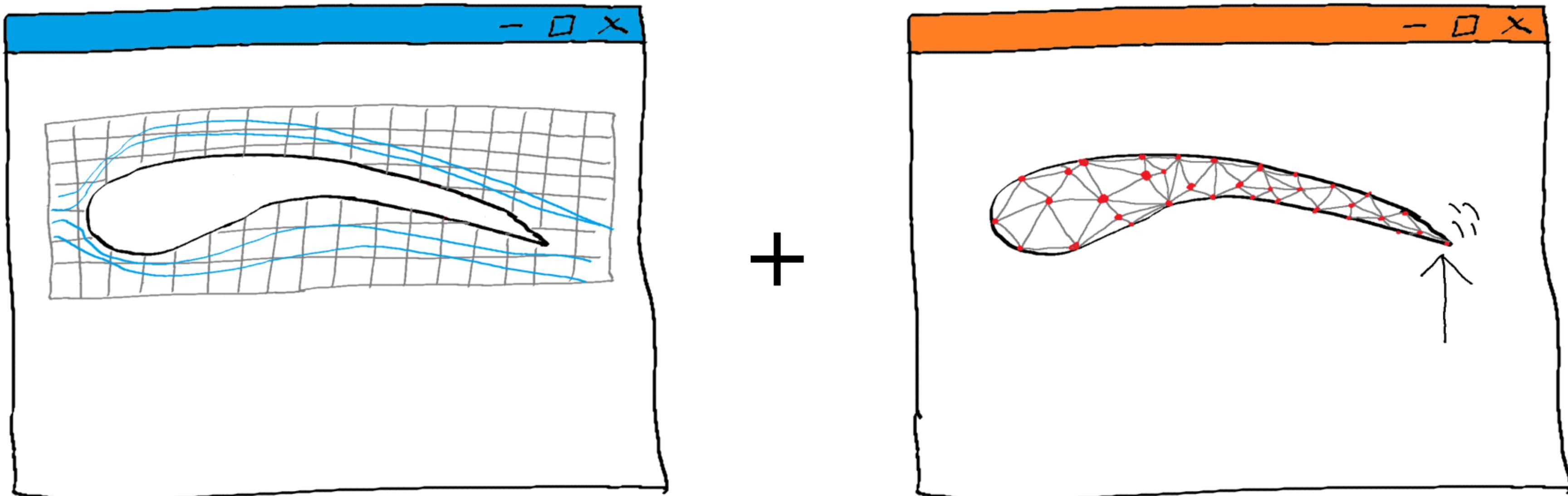
FEM: Simulating stresses on a wing



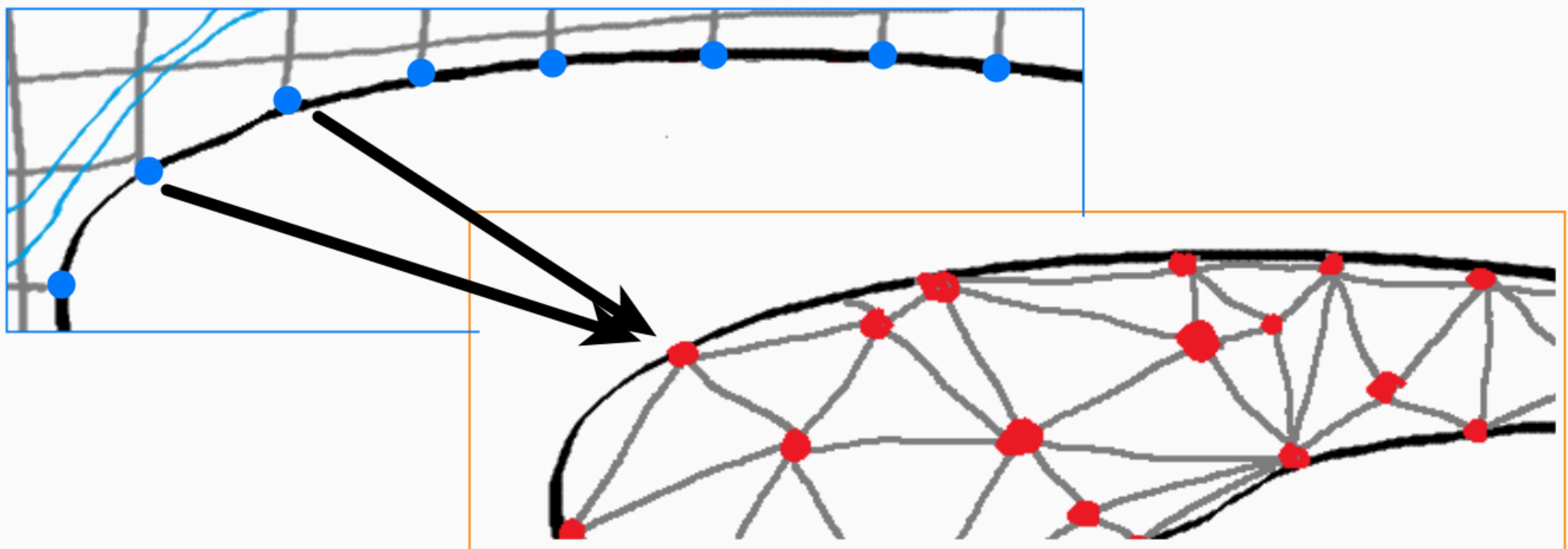
Can we simulate both at the same time?



...but reusing the CFD and FEM codes?



From the CFD world to the FEM world



Where preCICE helps



Communication

Options:

- MPI ports (fast)
- TCP sockets (robust)

Fully-parallel, peer-to-peer

Data mapping

Options:

- radial-basis functions
- projection-based
- conservative/consistent
- direct mesh access

Compute on any side

Coupling schemes

Options:

- serial / parallel
- explicit / implicit
- compositional, multi
- IQN, Aitken, ...

Same high-level API
Configurable at runtime

Time interpolation

Options:

- waveform iteration



What people do with preCICE

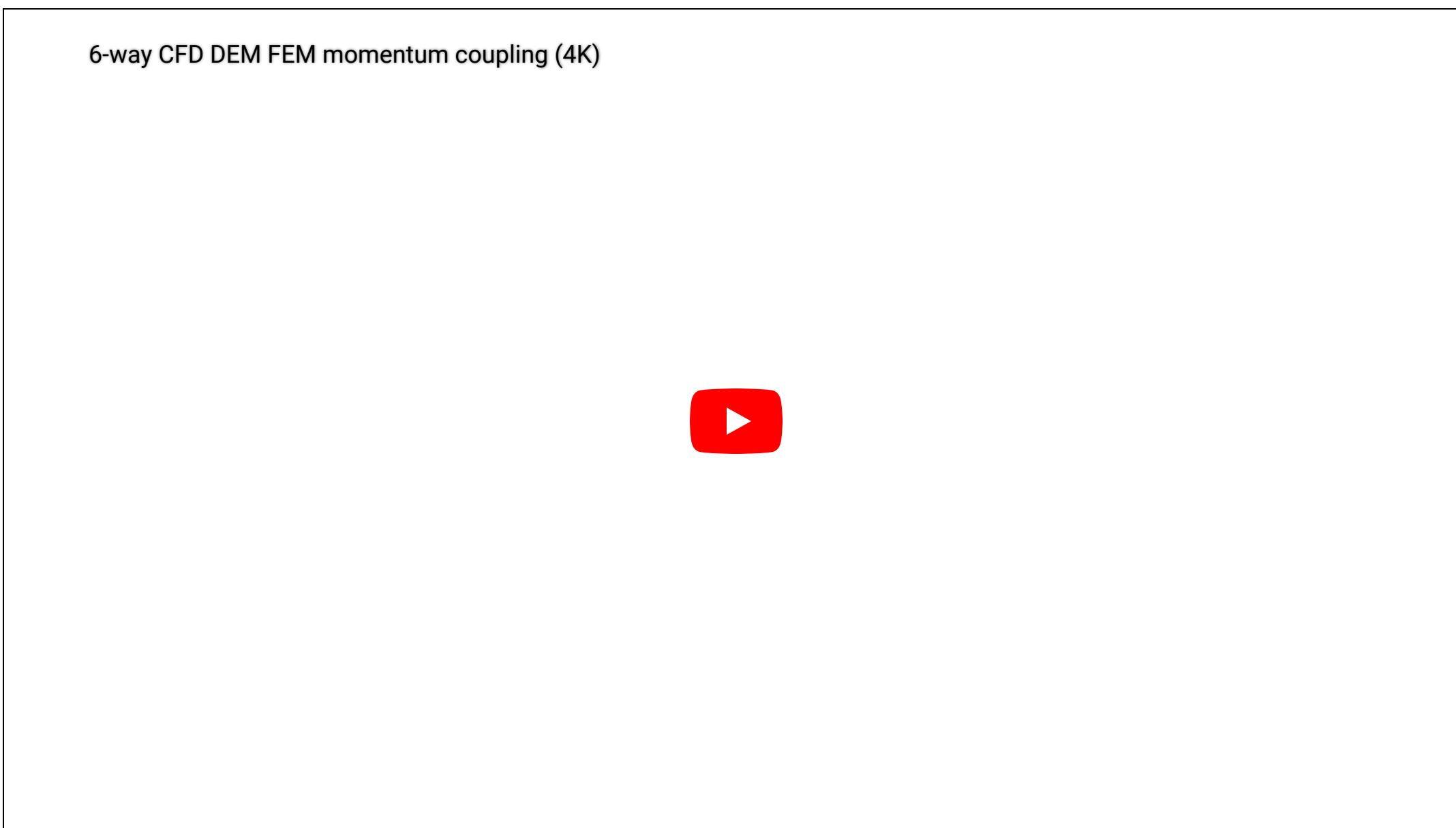
What people do with preCICE (1)

FSI coupling with OpenFOAM, CalculiX and preCICE for an undulation membrane tidal energy converter



Credit: Ulrich Heck, DHCAE Tools GmbH (Germany)

What people do with preCICE (2)



Credit: Prasad Adhav, Univ. of Luxembourg (Luxembourg)

What people do with preCICE (3)



Credit: ESLAB Simulations. SNU (South Korea)

What people do with preCICE (4)

[Home](#) > [Software for Exascale Computing - SPPEXA 2016-2019](#) > Conference paper

ExaFSA: Parallel Fluid-Structure-Acoustic Simulation

[Florian Lindner](#), [Amin Totounferoush](#), [Miriam Mehl](#)✉, [Benjamin Uekermann](#), [Neda Ebrahimi Pour](#), [Verena Krupp](#), [Sabine Roller](#), [Thorsten Reimann](#), [Dörte C. Sternal](#), [Ryusuke Egawa](#), [Hiroyuki Takizawa](#) & [Frédéric Simonis](#)

Conference paper | [Open Access](#) | [First Online: 31 July 2020](#)



Kolb, Elena (2022):

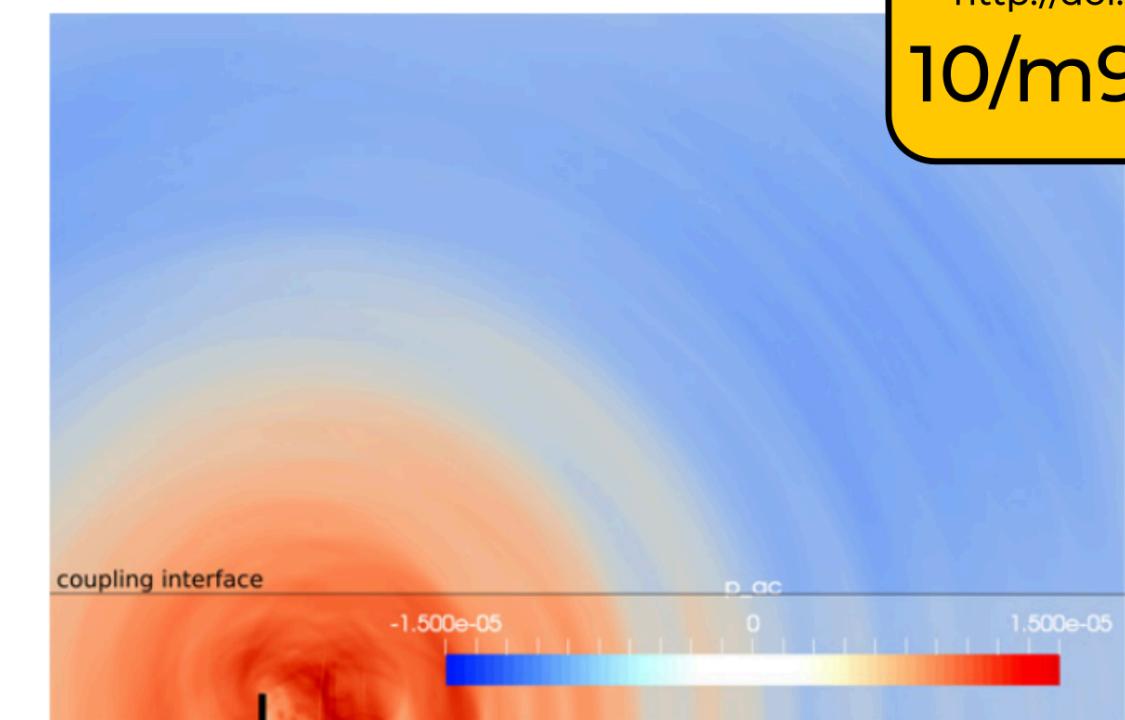
Aeroacoustic Simulation of Turbulent Fluid-Structure Interactions at Low Mach Numbers.

In: Ingenieurwissenschaften, Berlin, Verlag Dr. Hut, TU Darmstadt, ISBN 978-3-8439-5061-9, [Dissertation]

Mariño Salguero, Jessica Marcela (2021):

Numerical simulation of free surface flows interacting with flexible structures. (Verlagsversion)

Darmstadt, Technische Universität, → DOI: 10.26083/tuprints-00019193, → Offizielle URL, [Dissertation]



ExaFSA SPPEXA program, TU Darmstadt (Germany)

What people do with preCICE (5)

Massively Multiscale Modeling using NASA Multiscale Analysis Tool through Partitioned Task-Parallel Approach

Ibrahim Kaleel, Trenton M. Ricks, Peter A. Gustafson, Evan J. Pineda, Brett A. Bednarcyk and Steven M. Arnold

AIAA 2023-2027
Session: Multiscale Modeling II

Published Online: 19 Jan 2023 • <https://doi.org/10.2514/6.2023-2027>

 Read Now

Tools Share

Abstract:

View Video Presentation: <https://doi.org/10.2514/6.2023-2027.vid>

NASA Multiscale Analysis Tool (NASMAT) is a "plug and play" software package that allows users to conduct massively multiscale modeling of hierarchical and nonlinear materials. This work extends the scalability and improves the High Performance Computing friendliness of NASMAT by adopting a partitioned task-parallel approach. The interoperability of NASMAT with external software is enhanced through integration with preCICE, an open source library for multiphysics coupling in a partitioned manner. Enhancement through preCICE allows for easy integration of NASMAT to other macro solvers and dissociates the parallelization strategy adopted within NASMAT from the macro solver. A task-parallel framework based on the master-worker approach is implemented as the parallelization scheme. The scheme accounts for the hierarchy of multiple scales (task-dependence) and the heterogeneous nature (dynamic load balancing) of computations. The applicability and scalability of the framework are evaluated by analyzing multiscale problems with varying degrees of complexity and computational loads.

Figures References Related Details

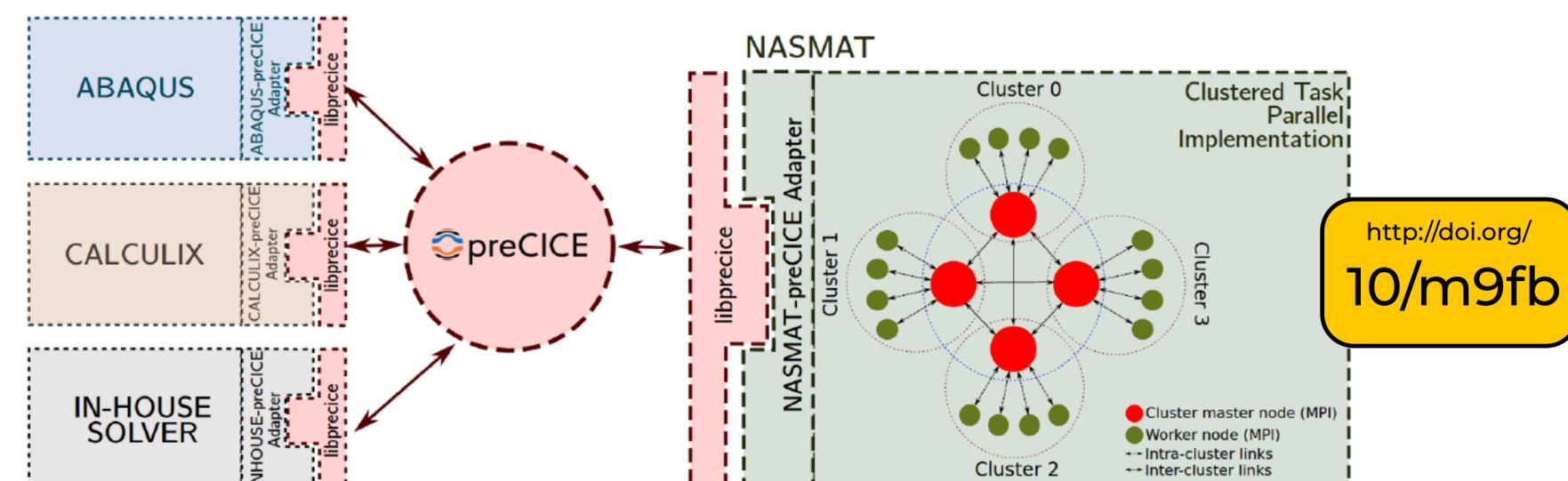
SCI TECH FORUM

AIAA SCITECH 2023 Forum
23-27 January 2023
National Harbor, MD & Online
<https://doi.org/10.2514/6.2023-2027>

Crossmark 

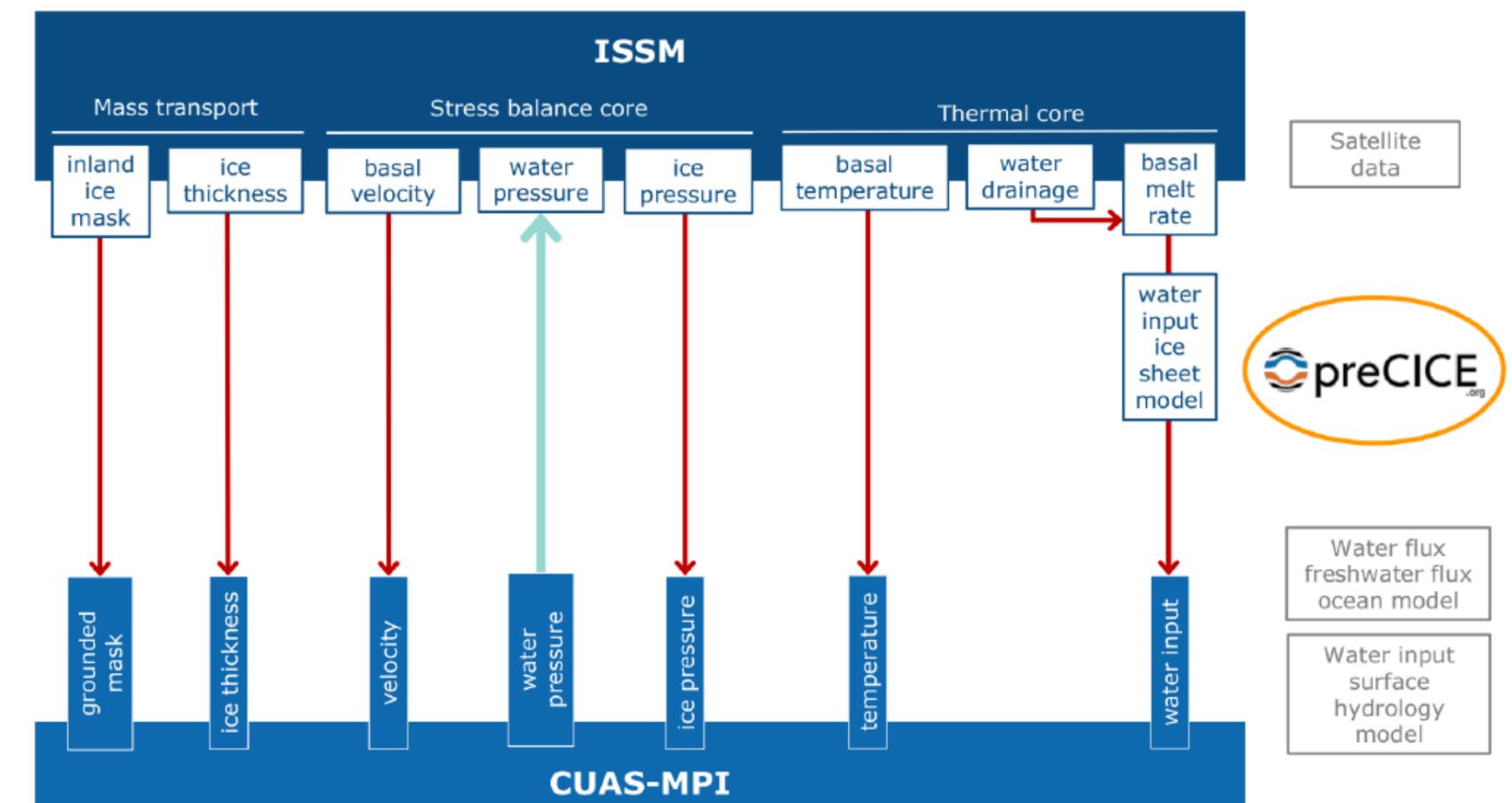
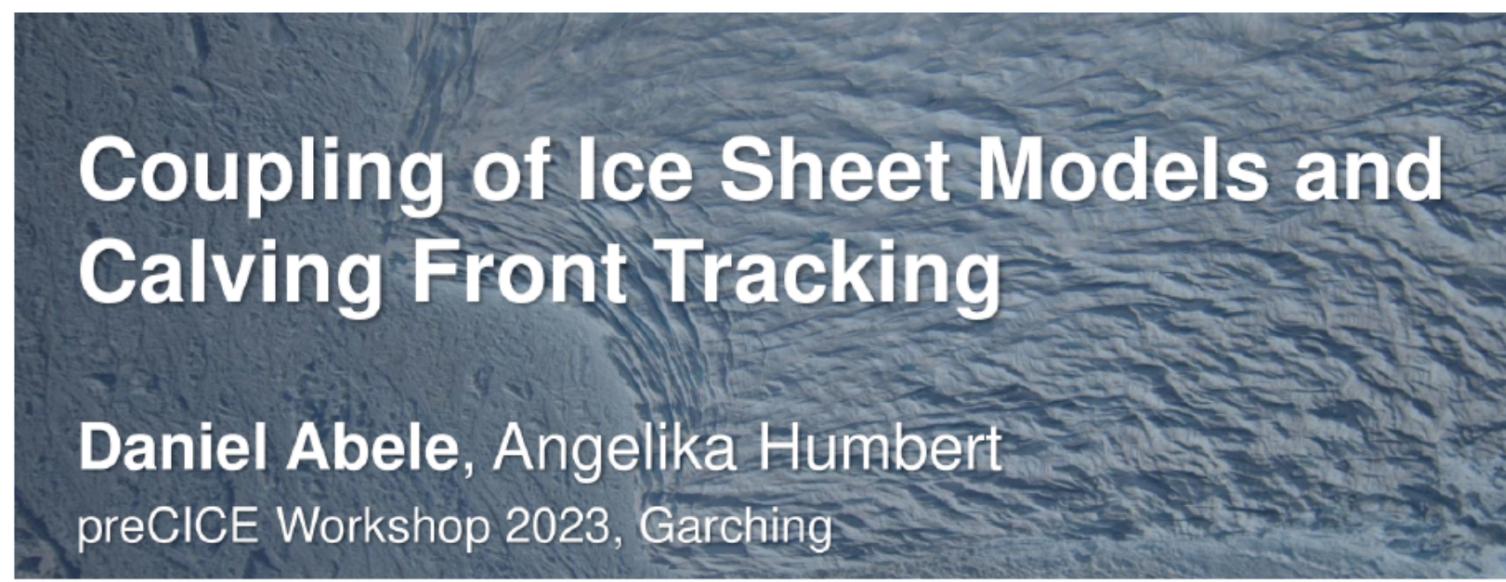
Information
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

PDF



A publication by NASA Glenn Research Center (USA)

What people do with preCICE (6)



Ice-sheet simulations by AWI (Germany)

What people do with preCICE (6)



Contents lists available at [ScienceDirect](#)

SoftwareX

journal homepage: www.elsevier.com/locate/softx



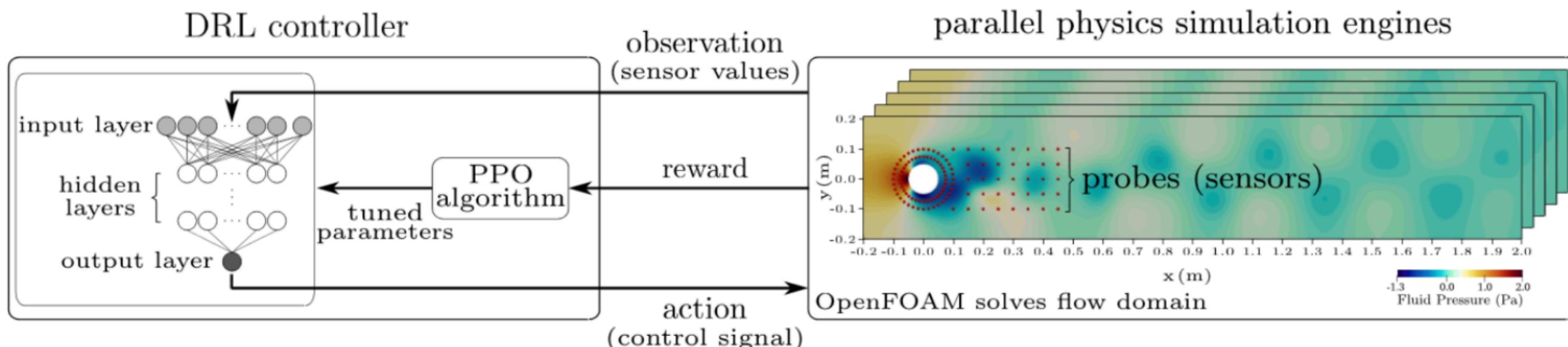
Original software publication

Gym-preCICE: Reinforcement learning environments for active flow control

Mosayeb Shams*, Ahmed H. Elsheikh

Heriot-Watt University, Edinburgh, United Kingdom

[http://doi.org/
10/n3rd](http://doi.org/10/n3rd)



Active flow control with ML, Heriot-Watt University (UK)

In this talk

- The preCICE library: Couple two toy Python solvers
- The preCICE ecosystem: Couple OpenFOAM + deal.II



Get the slides



github.com/MakisH/vki-training



Organizational notes

1. You are not expected to try things live.
2. Ask questions live, feel free to interrupt me.
3. Find all software installed in a demo virtual machine:
precice.org/installation-vm.html
4. Everything presented here is free software. preCICE and all the adapters are developed publicly on GitHub:
github.com/precice



Using the preCICE library

Coupling two toy Python solvers

Dependencies

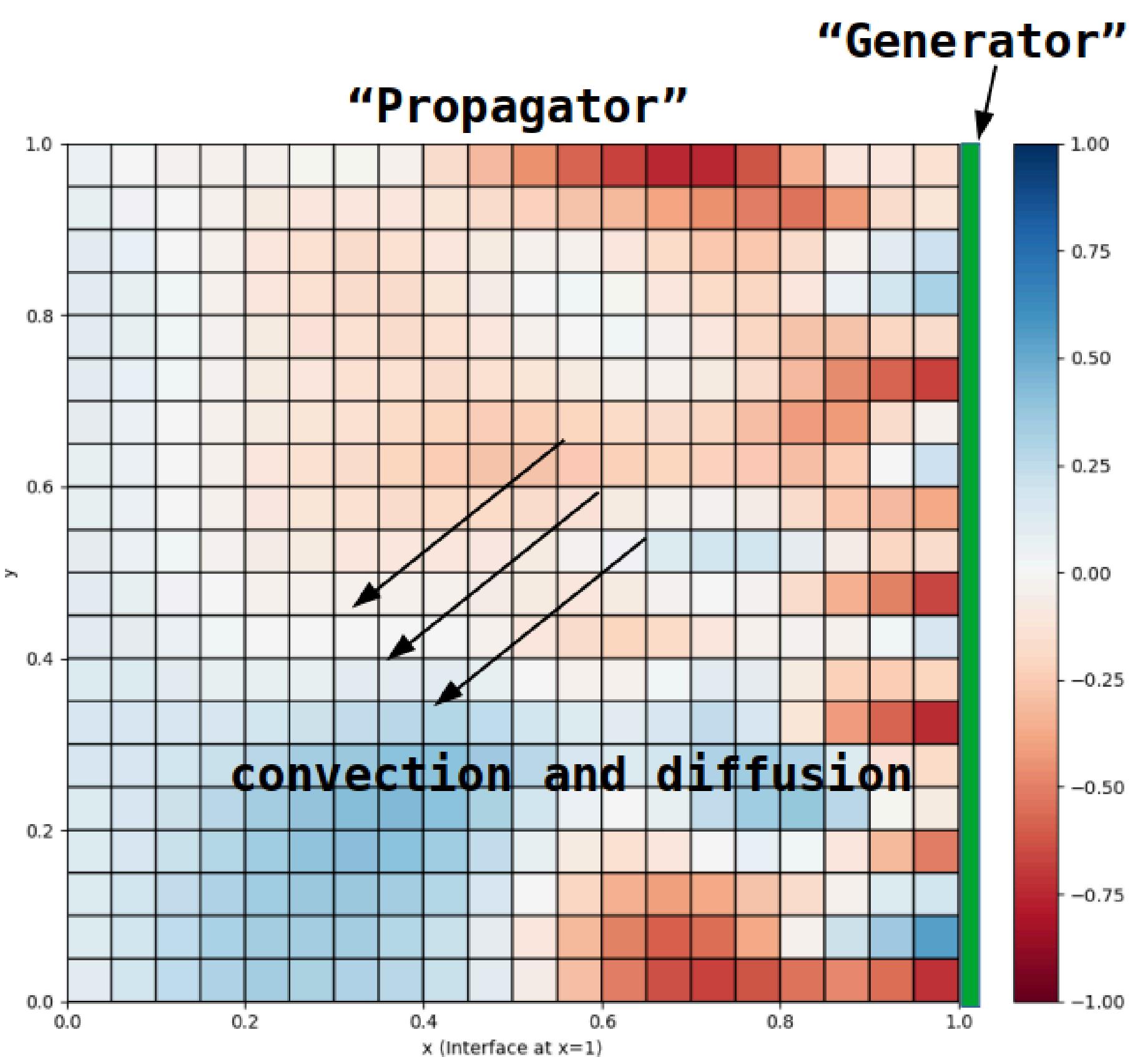
- preCICE v3 (e.g. packages for Ubuntu)
- Python 3
- Python packages numpy, matplotlib
- preCICE Python bindings:

```
pip3 install --upgrade pip
python3 -m venv .venv && . .venv/bin/activate
pip3 install pyprecice
```

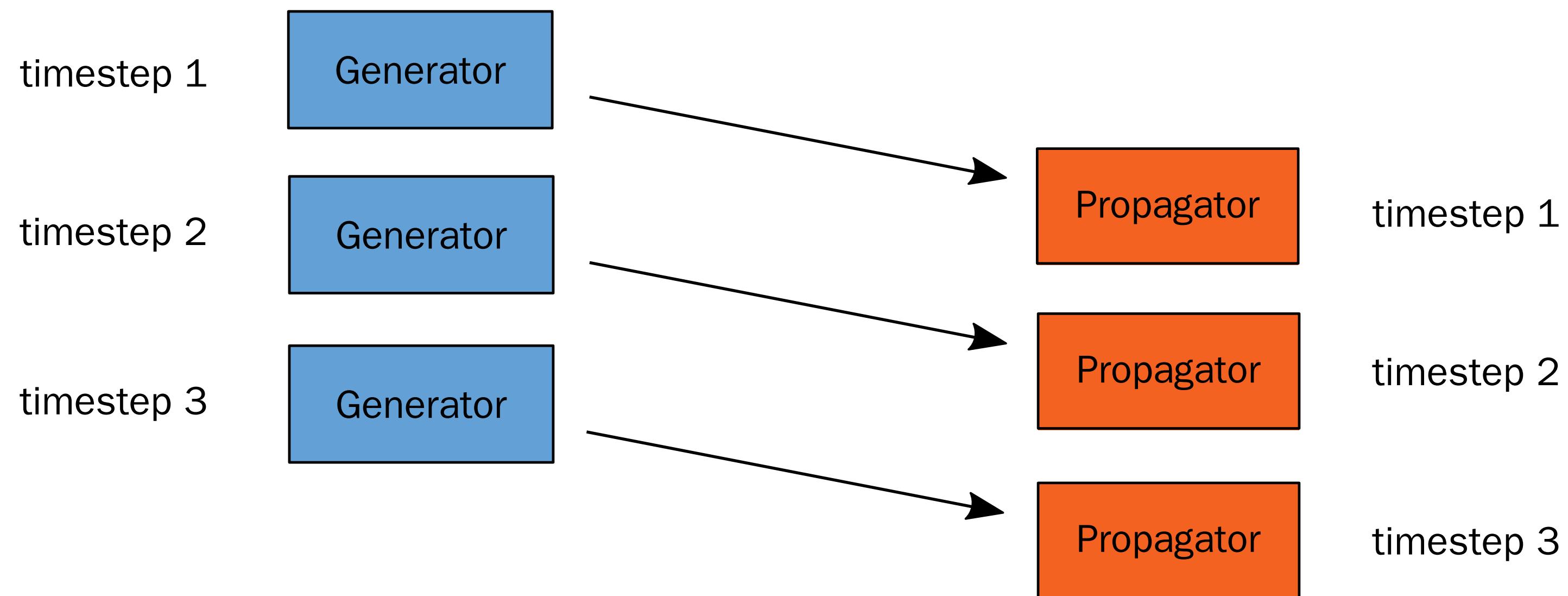


"Generator" and "Propagator"

- `generator.py`: generates random data in a 1D domain
- `propagator.py`: propagates boundary data over a 2D domain



Unidirectional coupling



generator.py

```
import numpy

# generate mesh
n = 20
y = numpy.linspace(0, 1, n + 1)

dt = 0.01
t = 0

while True:
    print("Generating data")
    u = 1 - 2 * numpy.random.rand(n)

    t = t + dt
    if(t > 0.1):
        break
```

propagator.py

```
# generate mesh
n = 20
x = numpy.linspace(0, 1, n+1)
y = numpy.linspace(0, 1, n+1)

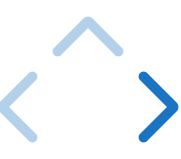
# initial data, associated to cell centers
u = numpy.zeros([n, n])

dt = 0.01
t = 0

# boundary condition for u (arbitrary)
u[:, -1] = y[:-1]

while True:

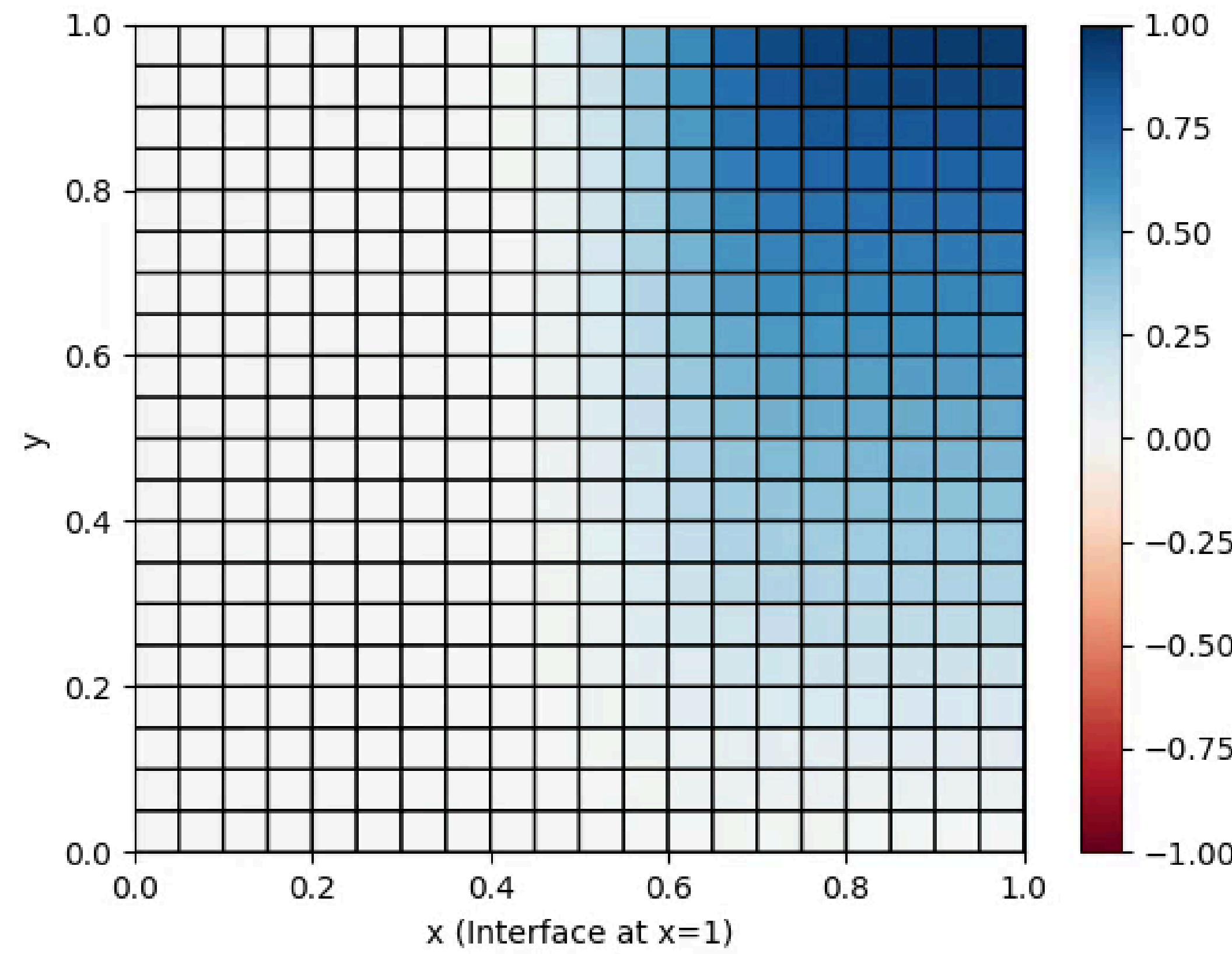
    print("Propagating data")
```



```
vagrant@precicevm: ~/Des...          vagrant@precicevm: ~/Desktop/skeleton
vagrant@precicevm: ~/Desktop/skeleton 162x44
vagrant@precicevm:~/Desktop/skeleton$ tree
.
+-- clean.sh
+-- generator
|   +-- generator.py
+-- precice-config.xml
+-- propagator
|   +-- propagator.py
+-- README.txt

2 directories, 5 files
vagrant@precicevm:~/Desktop/skeleton$
```

Uncoupled simulation



Import preCICE

```
1 import numpy
2
3
4 # generate mesh
5 n = 20
6 y = numpy.linspace(0, 1, n + 1)
7
8 dt = 0.01
9 t = 0
10
11 while True:
12     print("Generating data")
13     u = 1 - 2 * numpy.random.rand(n)
14
15     t = t + dt
16     if(t > 0.1):
17         break
```

Import preCICE

```
1 import numpy
2 import precice
3
4 # generate mesh
5 n = 20
6 y = numpy.linspace(0, 1, n + 1)
7
8 dt = 0.01
9 t = 0
10
11 while True:
12     print("Generating data")
13     u = 1 - 2 * numpy.random.rand(n)
14
15     t = t + dt
16     if(t > 0.1):
17         break
```

Configure preCICE

```
1 import numpy
2 import precice
3
4 # generate mesh
5 n = 20
6 y = numpy.linspace(0, 1, n + 1)
7
8 # preCICE setup
9 participant_name      = "Generator"
10 config_file_name     = "../precice-config.xml"
11 solver_process_index = 0
12 solver_process_size  = 1
13 participant =
14     precice.Participant(
15         participant_name,
16         config_file_name,
17         solver_process_index,
```

Configure preCICE

```
5 n = 20
6 y = numpy.linspace(0, 1, n + 1)
7
8 # preCICE setup
9 participant_name      = "Generator"
10 config_file_name     = "../precice-config.xml"
11 solver_process_index = 0
12 solver_process_size  = 1
13 participant =
14     precice.Participant(
15         participant_name,
16         config_file_name,
17         solver_process_index,
18         solver_process_size
19     )
20
21 dt = 0.01
22 t = 0
```



Define the coupling mesh

```
5 n = 20
6 y = numpy.linspace(0, 1, n + 1)
7
8 # preCICE setup
9 participant_name      = "Generator"
10 config_file_name     = "../precice-config.xml"
11 solver_process_index = 0
12 solver_process_size  = 1
13 participant =
14     precice.Participant(
15         participant_name,
16         config_file_name,
17         solver_process_index,
18         solver_process_size
19     )
20
21 # Define the coupling mesh name
22 mesh_name = "Generator-Mesh"
```

Define the coupling mesh

```
16             config_file_name,
17             solver_process_index,
18             solver_process_size
19         )
20
21 # Define the coupling mesh name
22 mesh_name = "Generator-Mesh"
23
24 # Define the coupling mesh
25 positions = [[1, y0] for y0 in y[:-1]]
26 vertex_ids = participant.set_mesh_vertices(mesh_name, positions)
27
28 dt = 0.01
29 t = 0
30
31 while True:
32     print("Generating data")
```

Initialize and finalize preCICE

```
16         config_file_name,
17         solver_process_index,
18         solver_process_size
19     )
20
21 # Define the coupling mesh name
22 mesh_name = "Generator-Mesh"
23
24 # Define the coupling mesh
25 positions = [[1, y0] for y0 in y[:-1]]
26 vertex_ids = participant.set_mesh_vertices(mesh_name, positions)
27
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
```

Initialize and finalize preCICE

```
20
21 # Define the coupling mesh name
22 mesh_name = "Generator-Mesh"
23
24 # Define the coupling mesh
25 positions = [[1, y0] for y0 in y[:-1]]
26 vertex_ids = participant.set_mesh_vertices(mesh_name, positions)
27
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
33 while True:
34     print("Generating data")
35     u = 1 - 2 * numpy.random.rand(n)
36
```

Advance the coupling

```
20
21 # Define the coupling mesh name
22 mesh_name = "Generator-Mesh"
23
24 # Define the coupling mesh
25 positions = [[1, y0] for y0 in y[:-1]]
26 vertex_ids = participant.set_mesh_vertices(mesh_name, positions)
27
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
33 while True:
34     print("Generating data")
35     u = 1 - 2 * numpy.random.rand(n)
36
```



Advance the coupling

```
27
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
33 while True:
34     print("Generating data")
35     u = 1 - 2 * numpy.random.rand(n)
36
37
38
39     t = t + dt
40     if(t > 0.1):
41         break
42
43 participant.finalize()
```

Advance the coupling

```
27
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
33 while True:
34     print("Generating data")
35     u = 1 - 2 * numpy.random.rand(n)
36
37     participant.advance(dt)
38
39     t = t + dt
40     if(t > 0.1):
41         break
42
43 participant.finalize()
```

Advance the coupling

```
27
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
33 while participant.is_coupling_ongoing():
34     print("Generating data")
35     u = 1 - 2 * numpy.random.rand(n)
36
37     participant.advance(dt)
38
39     t = t + dt
40
41
42
43 participant.finalize()
```

Advance the coupling

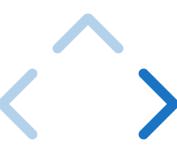
```
28 participant.initialize()
29
30 dt = 0.01
31 t = 0
32
33 while participant.is_coupling_ongoing():
34     print("Generating data")
35     u = 1 - 2 * numpy.random.rand(n)
36
37     participant.advance(dt)
38     precice_dt = participant.get_max_time_step_size()
39
40     t = t + dt
41
42
43
44 interface.finalize()
```

Advance the coupling

```
27
28 participant.initialize()
29 precice_dt = participant.get_max_time_step_size()
30
31 dt = 0.01
32 t = 0
33
34 while participant.is_coupling_ongoing():
35     print("Generating data")
36     u = 1 - 2 * numpy.random.rand(n)
37
38     participant.advance(dt)
39     precice_dt = participant.get_max_time_step_size()
40
41     t = t + dt
42
43 interface.finalize()
```

Advance the coupling

```
27
28 participant.initialize()
29 precice_dt = participant.get_max_time_step_size()
30
31 dt = 0.01
32 t = 0
33
34 while participant.is_coupling_ongoing():
35     print("Generating data")
36     u = 1 - 2 * numpy.random.rand(n)
37
38     participant.advance(dt)
39     precice_dt = participant.get_max_time_step_size()
40
41     t = t + dt
42
43 interface.finalize()
```



Not there yet, but let's run it
(similar changes in propagator.py - try it at home)

Nothing happening?

```
vagrant@precicevm:~/Desktop/solution/T4/generator$ tree
.
└── generator.py

0 directories, 1 file

vagrant@precicevm:~/Desktop/solution/T4/generator$ python3 generator.py
---[precice] This is preCICE version 3.1.1
---[precice] Revision info: v3.1.1
---[precice] Build type: Release + debug log
---[precice] Configuring preCICE with configuration "../precice-config.xml"
---[precice] I am participant "Generator"
---[precice] Setting up primary communication to coupling partner/s

vagrant@precicevm:~/Desktop/solution/T4/propagator$ tree
.
└── propagator.py

0 directories, 1 file

vagrant@precicevm:~/Desktop/solution/T4/propagator$ python3 propagator.py
```

Steering works. But did we forget something?



Write & read data

```
20
21 # Get the preCICE mesh name
22 mesh_name = "Generator-Mesh"
23
24 # Define the coupling mesh
25 positions = [[1, y0] for y0 in y[:-1]]
26 vertex_ids = participant.set_mesh_vertices(mesh_name, positions)
27
28
29
30
31
32 participant.initialize()
33 precice_dt = participant.get_max_time_step_size()
34
35 dt = 0.01
36 t = 0
```

Write & read data

```
20
21 # Get the preCICE mesh name
22 mesh_name = "Generator-Mesh"
23
24 # Define the coupling mesh
25 positions = [[1, y0] for y0 in y[:-1]]
26 vertex_ids = participant.set_mesh_vertices(mesh_id, positions)
27
28 # Get the exchanged data name
29 data_name = "Color"
30
31 participant.initialize()
32
33 dt = 0.01
34 t = 0
35
36 while participant.is_coupling_ongoing():
```



Write & read data

```
32
33 dt = 0.01
34 t = 0
35
36 while participant.is_coupling_ongoing():
37     dt = np.minimum(dt, precice_dt)
38
39     print("Generating data")
40     u = 1 - 2 * numpy.random.rand(n)
41
42     participant.write_data(mesh_name, data_name, vertex_ids, u)
43     precice_dt = participant.advance(dt)
44     # u[:, -1] = participant.read_data(mesh_name, data_name, vertex_ids)
45
46     t = t + dt
47
48 participant.finalize()
```

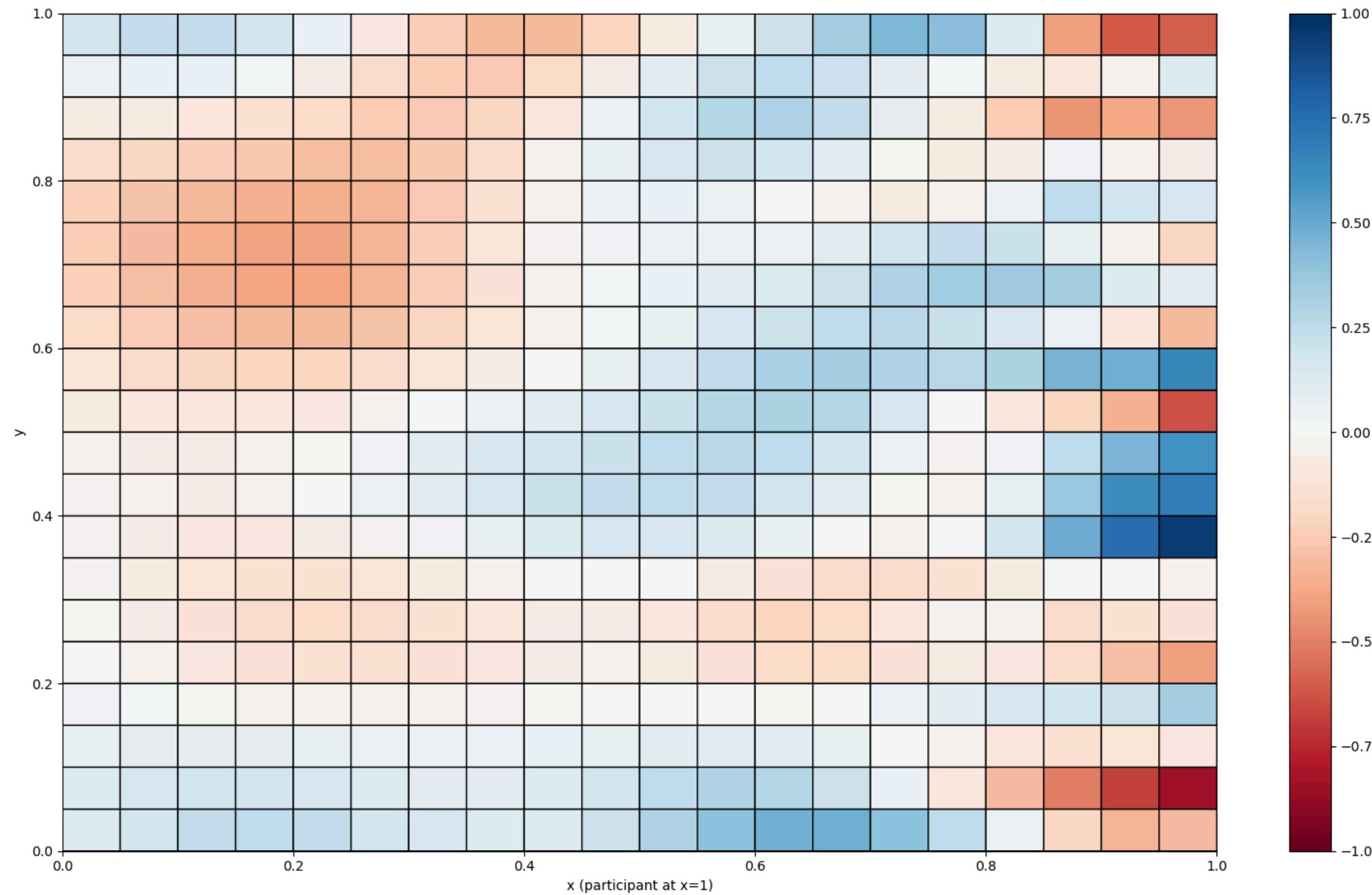


It should work now!

```
vagrant@precicevm:~/Desktop... └─ No Indicators ─ +  
vagrant@precicevm:~/Desktop/solution/T5/generator 8x44  
vagrant@precicevm:~/Desktop/solution/T5/generator$ tree  
└── generator.py  
  
0 directories, 1 file  
vagrant@precicevm:~/Desktop/solution/T5/generator$ python3 generator.py  
vagrant@precicevm:~/Desktop/solution/T5/generator$ vagrant@precicevm:~/Desktop/solution/T5/propagator 8x44  
vagrant@precicevm:~/Desktop/solution/T5/propagator$ tree  
└── propagator.py  
  
0 directories, 1 file  
vagrant@precicevm:~/Desktop/solution/T5/propagator$ python3 propagator.py  
---[precice] This is preCICE version 3.1.1  
---[precice] Revision info: v3.1.1  
---[precice] Build type: Release + debug log  
---[precice] Configuring preCICE with configuration "../precice-config.xml"  
---[precice] I am participant "Propagator"  
---[precice] Setting up primary communication to coupling partner/s
```



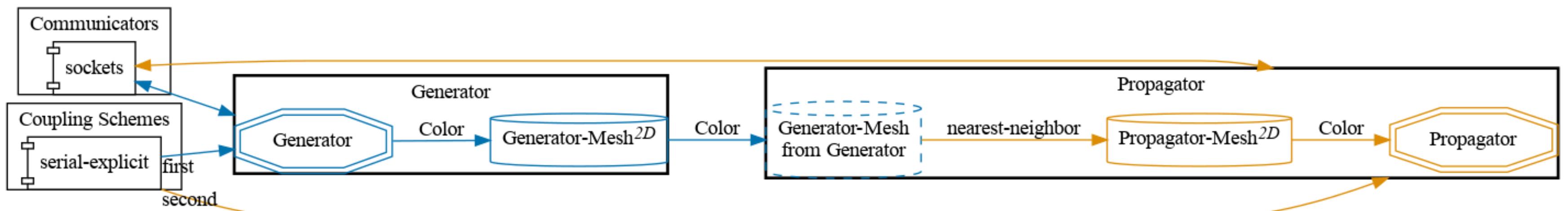
Data is being transferred!



Most basic case: uni-directional, serial-explicit, nearest-neighbor mapping, ...



preCICE configuration



Visual representation of `precice-config.xml` using the [config visualizer](#).

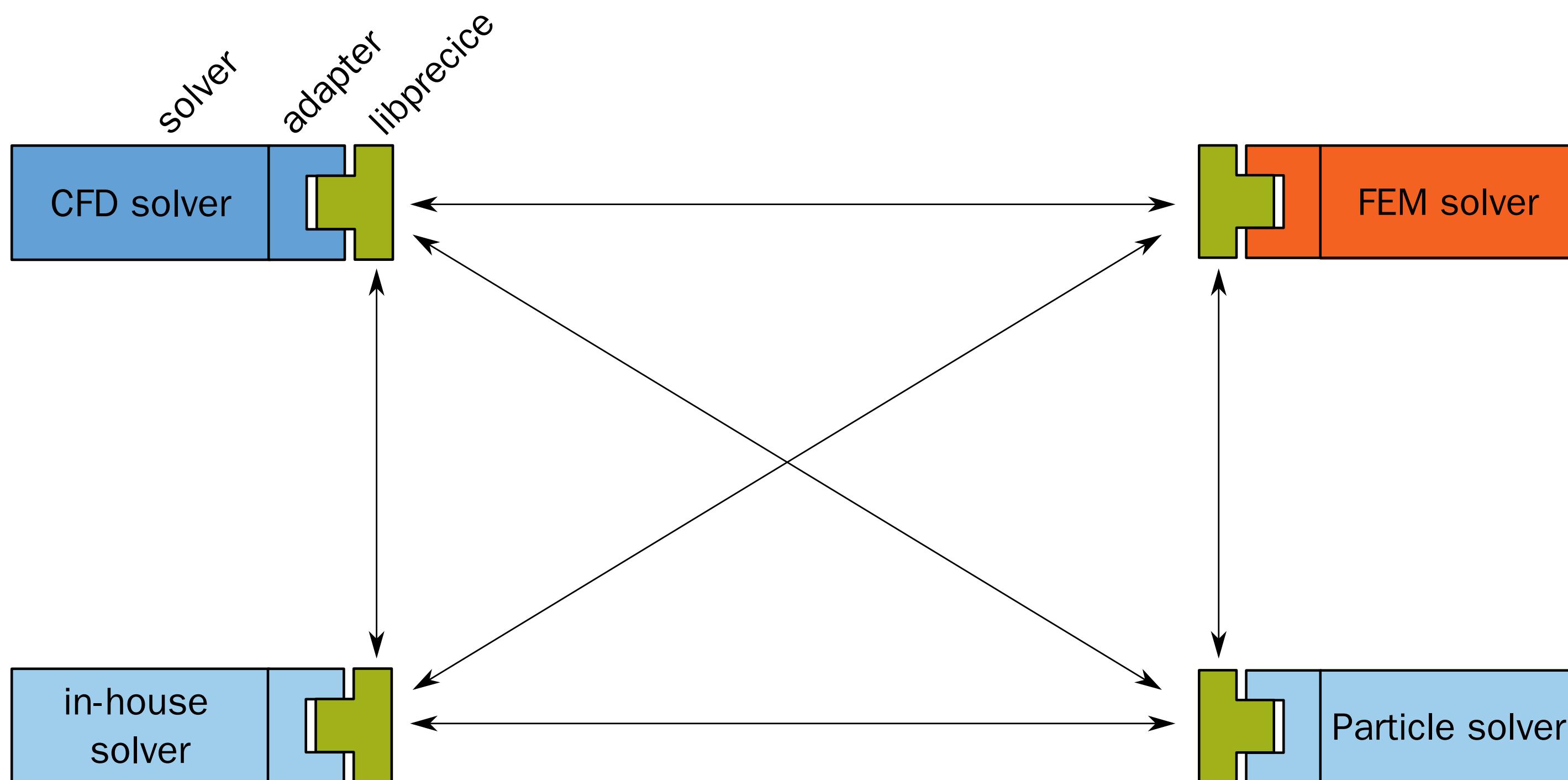
Using the preCICE ecosystem

Example: Coupling OpenFOAM and deal.II

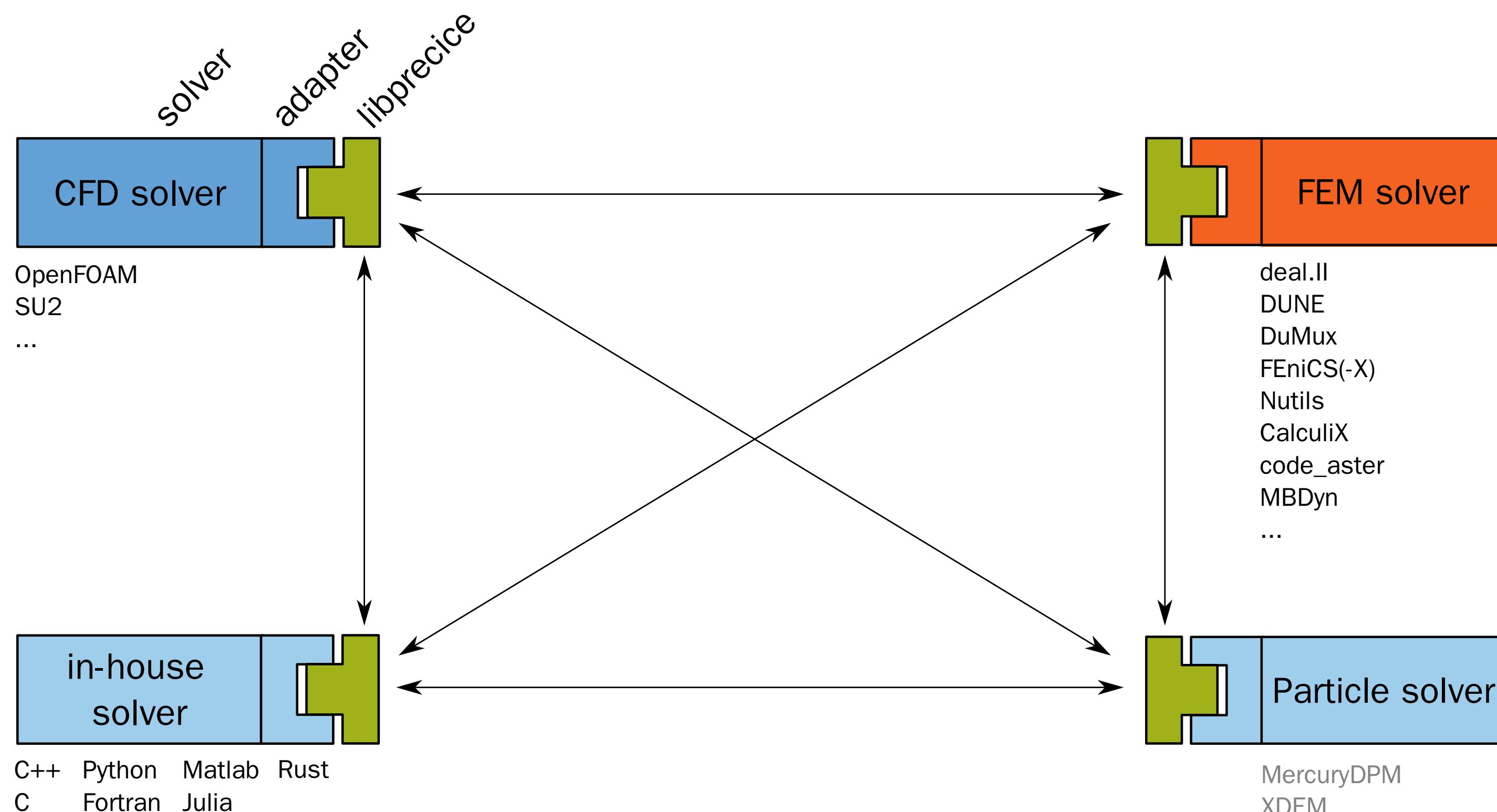
The big picture



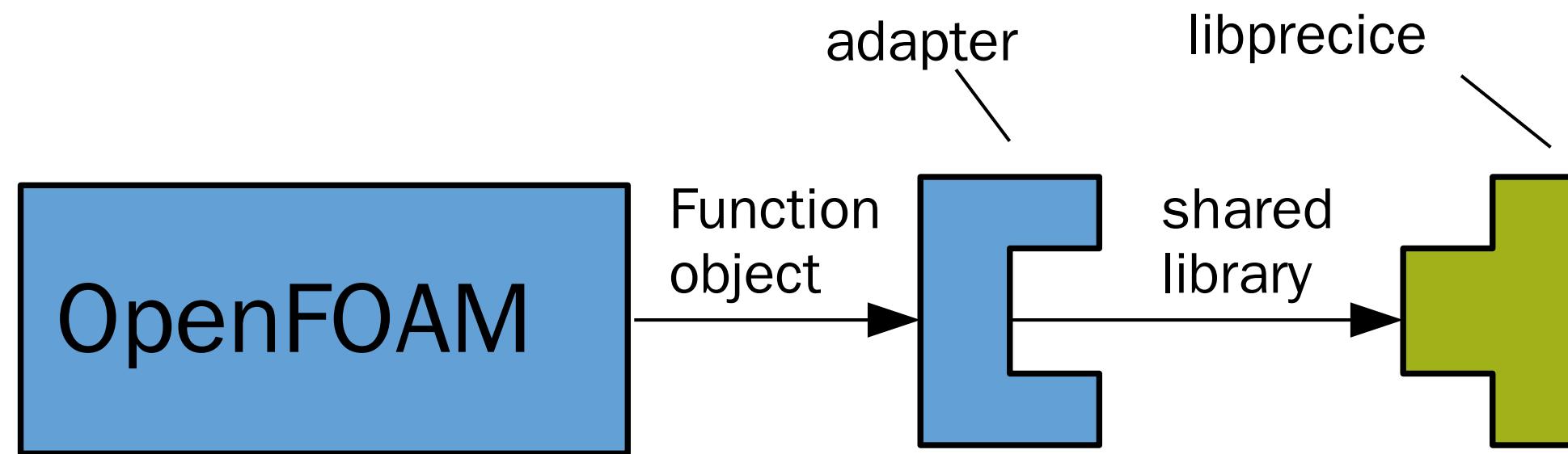
The big picture



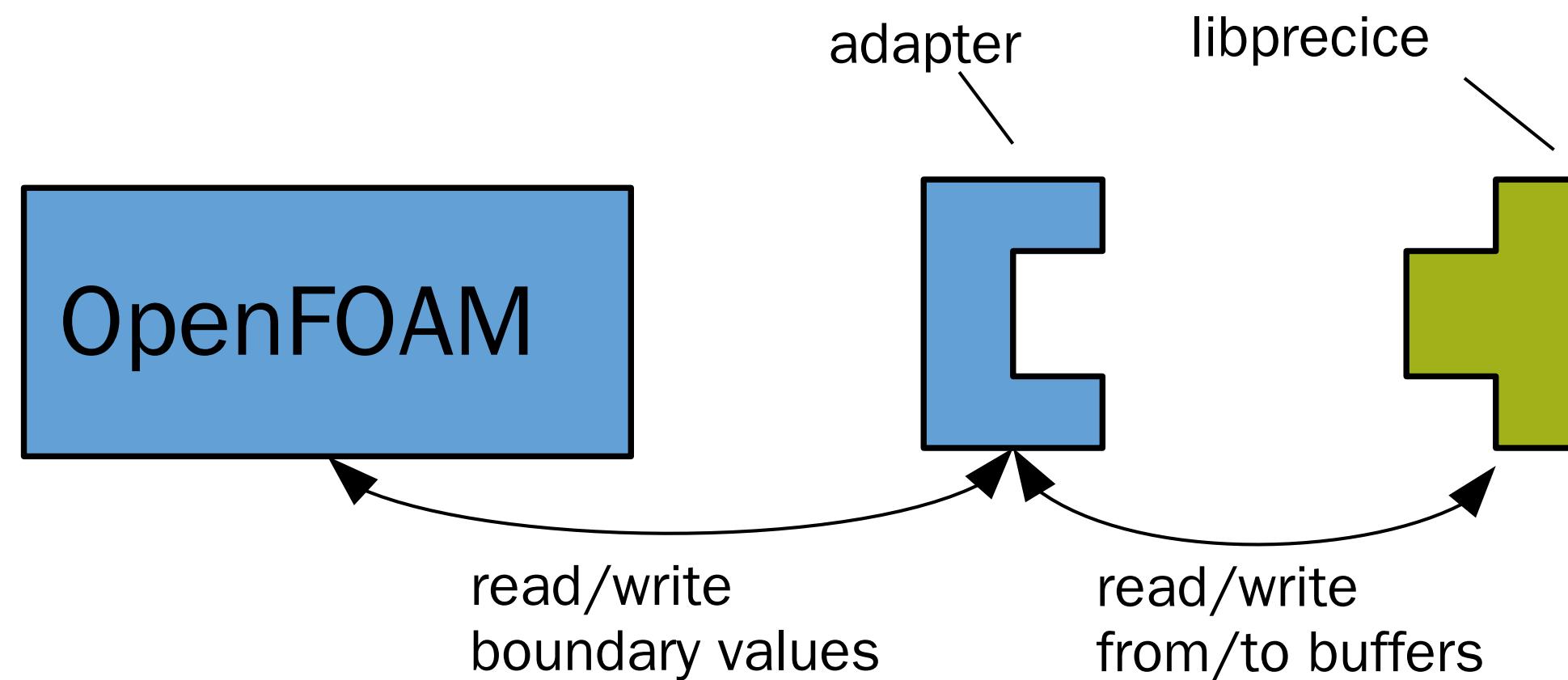
The big picture



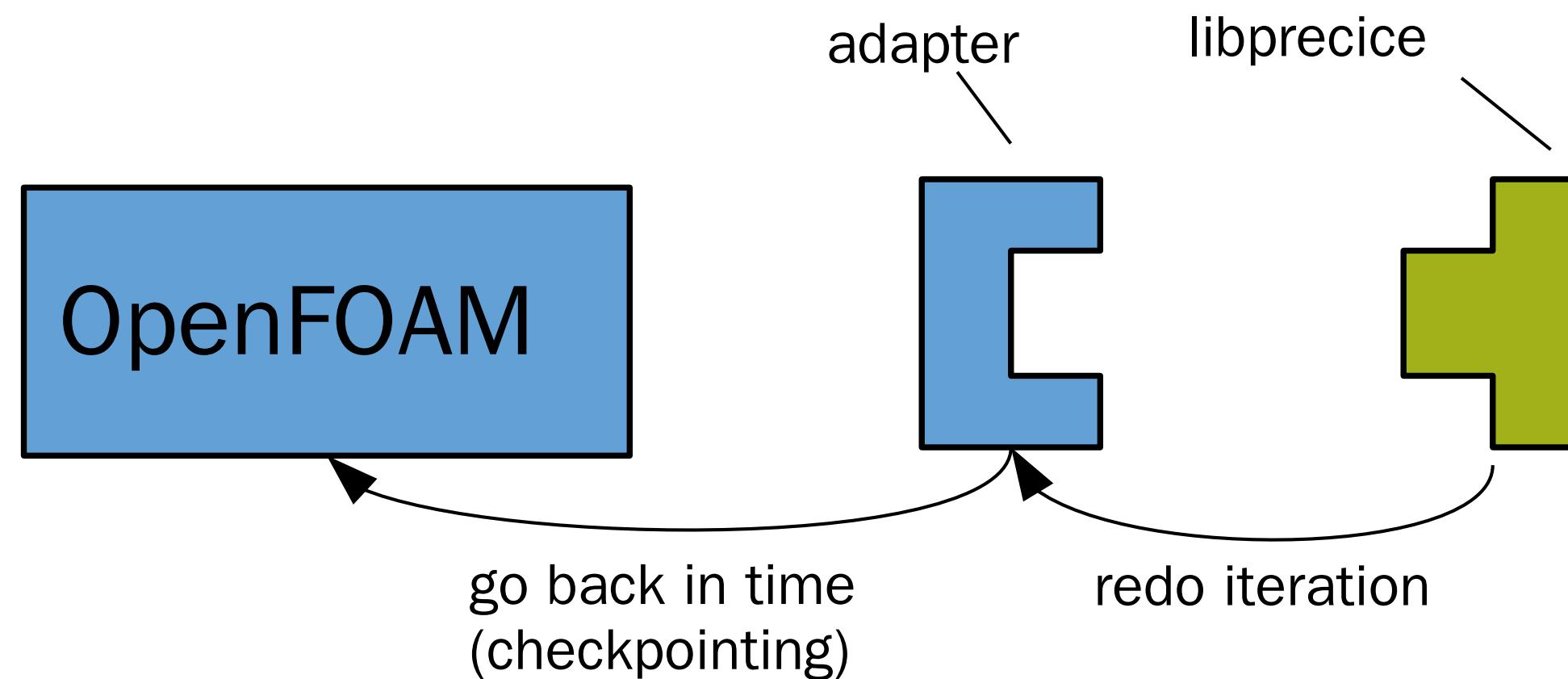
What does the adapter do?



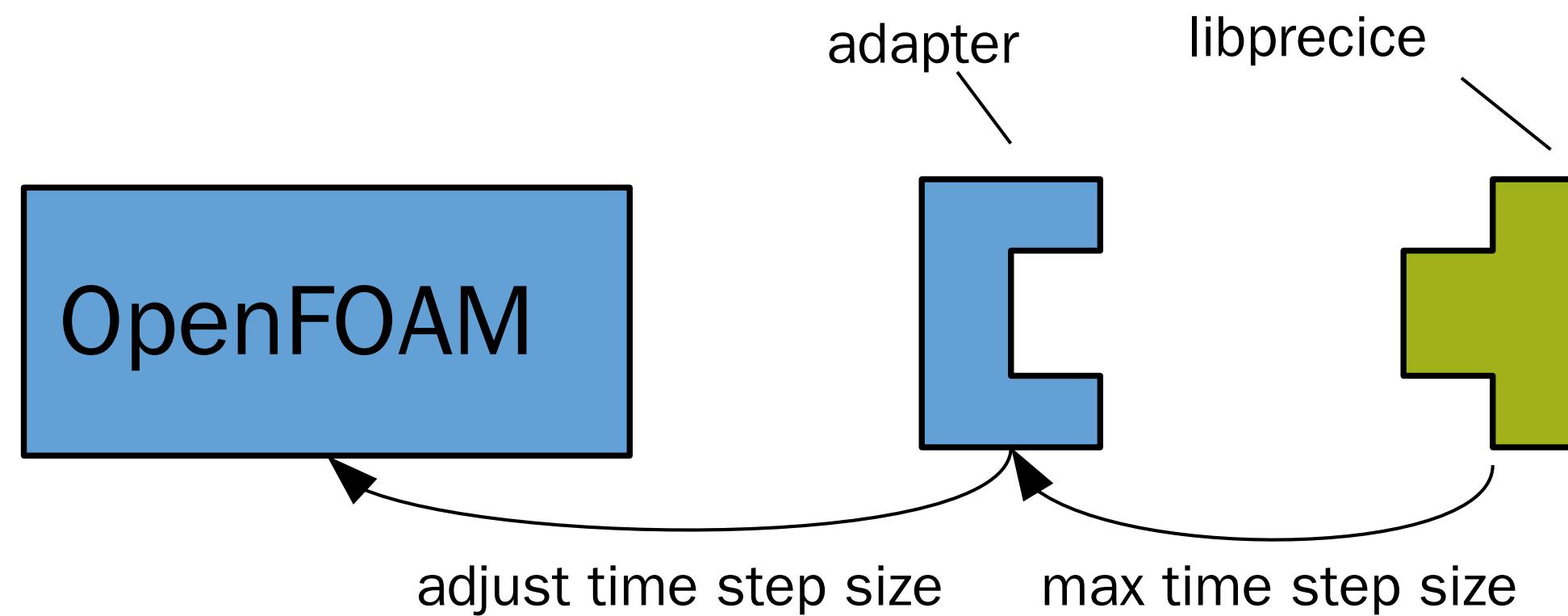
What does the adapter do?



What does the adapter do?



What does the adapter do?



Save the date: preCICE Workshop 2025, Hamburg, Sep 8-12

Quickstart

Summary: Install preCICE on Linux (e.g. via a Debian package) and couple an OpenFOAM fluid solver (using the OpenFOAM-preCICE adapter) with an example rigid body solver in C++.

[Edit me](#)

Updated 24 Jan 25

This is the first step you may want to try if you are new to preCICE: install preCICE and some solvers, and run a simple coupled case.

To get a feeling what preCICE does, watch a [short presentation](#) or a [longer talk on the fundamentals](#).

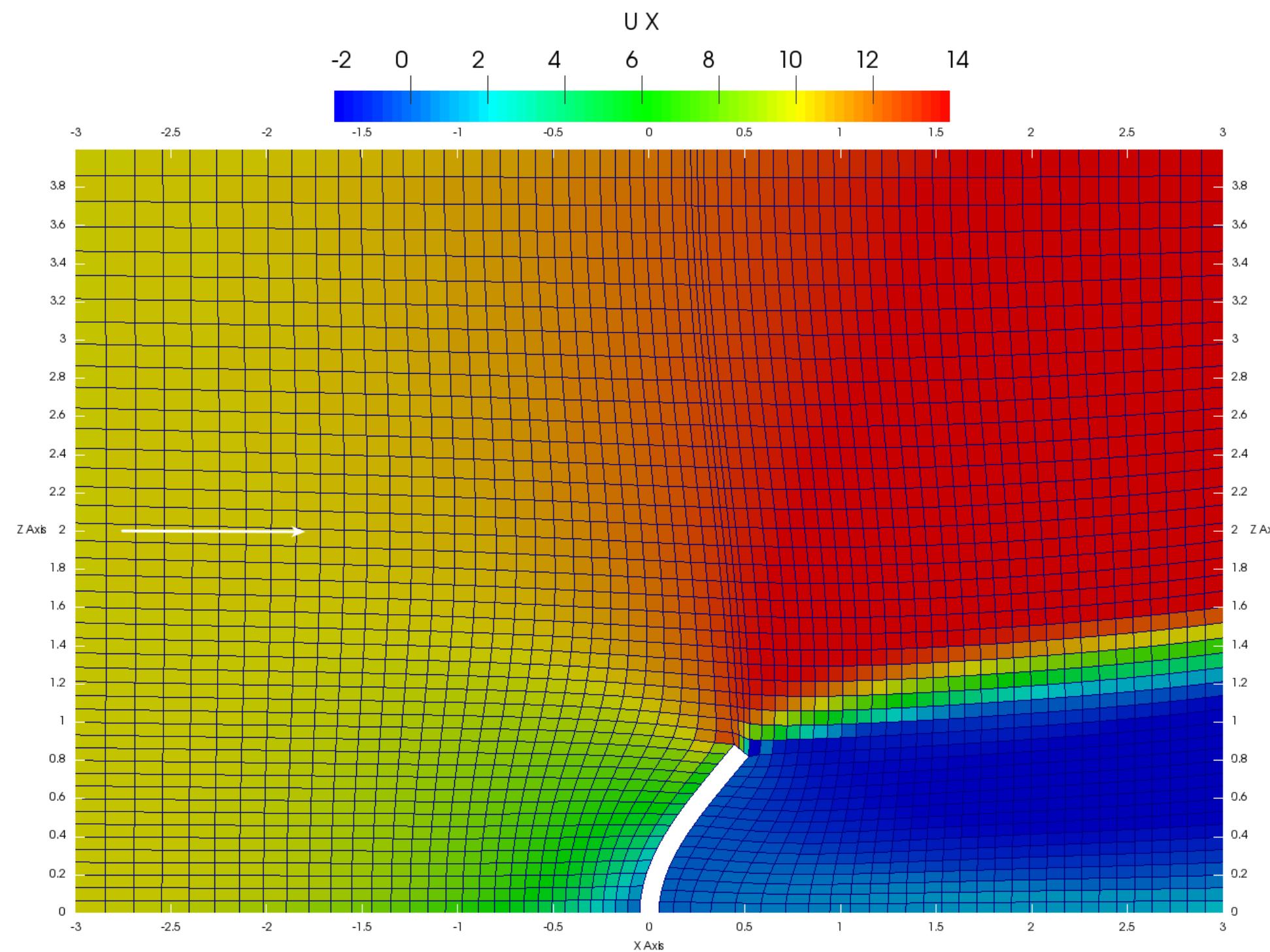
Installation

1. Get and install preCICE. For Ubuntu 24.04 (Noble Numbat), this is pretty easy: [download](#) and install our binary package by clicking on it or using the following commands:

```
wget https://github.com/precice/precice/releases/download/v3.1.2/libprecice3_3.1.2_noble.deb  
sudo apt install ./libprecice3_3.1.2_noble.deb
```

OS	Package
Ubuntu 20.04 Focal Fossa	libprecice3_3.1.2_focal.deb
Ubuntu 22.04 Jammy Jellyfish	libprecice3_3.1.2_jammy.deb
Ubuntu 24.04 Noble Numbat	libprecice3_3.1.2_noble.deb

Tutorial: Channel with a perpendicular flap



Find this tutorial on precice.org/tutorials-perpendicular-flap.html.

Arbitrary solver combinations

FLUID

SOLID

pimpleFoam

CalculiX

SU2

deal.II

Nutils

FEniCS

DUNE

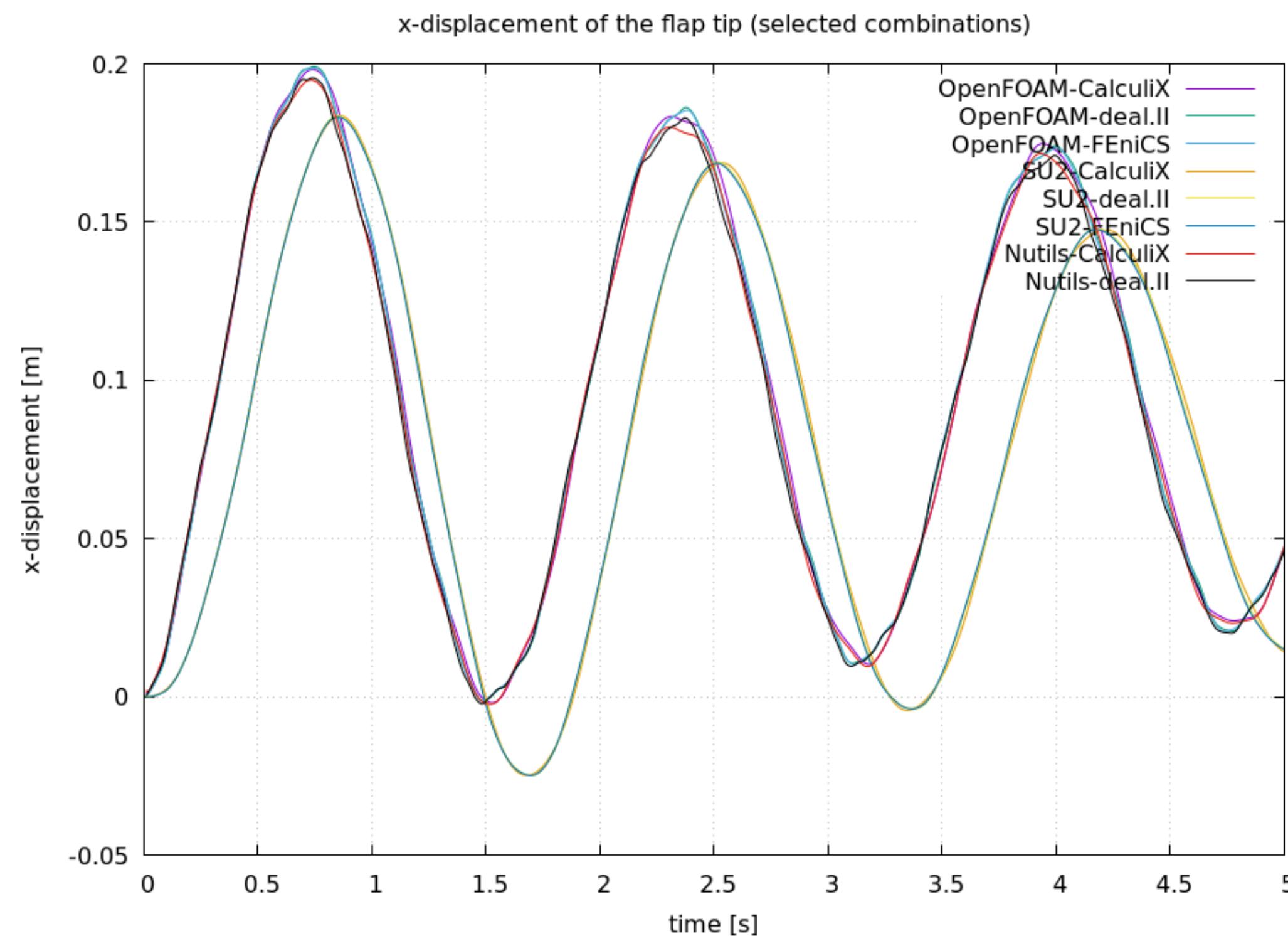
solidDisplacementFoam

solids4Foam

Nutils

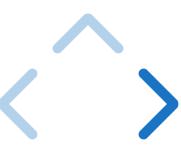


Arbitrary solver combinations



Dependencies

- preCICE v3 (e.g. packages for Ubuntu)
- Recent OpenFOAM (e.g., v2312)
- OpenFOAM-preCICE adapter v1.3.0
- deal.II 9.2 or greater
- deal.II-preCICE adapter 9.5.0



File structure

- precice-config.xml
- fluid-openfoam/
 - 0/U ...
 - constant/dynamicMeshDict ...
 - system/
 - controlDict
 - preciceDict
 - ...
- solid-dealii/
 - parameters.prm
 - ...

For demonstration purposes

- In precice-config.xml:
 - Reduce the max-time from 5s to 1.5s
 - Switch to a serial-explicit coupling scheme
- In solid-dealii/parameters.prm:
 - Lower the solid density from 3000 to 42



Configure OpenFOAM

```
1 // fluid-openfoam/0/U
2
3 flap
4 {
5     type          movingWallVelocity;
6     value         uniform (0 0 0);
7 }
```

```
1 // fluid-openfoam/0/pointDisplacement
2
3 flap
4 {
5     type          fixedValue;
6     value         $internalField;
7 }
```

```
1 // fluid-openfoam/constant/dynamicMeshDict
2
3 solver      displacementLaplacian;
```

Load the OpenFOAM adapter

```
1 // fluid-openfoam/system/controlDict
2 functions
3 {
4     precICE_Adapter
5     {
6         type preciceAdapterFunctionObject;
7         libs ("libpreciceAdapterFunctionObject.so");
8     }
9 }
10
11 // Don't forget to build the adapter with Allwmake
```



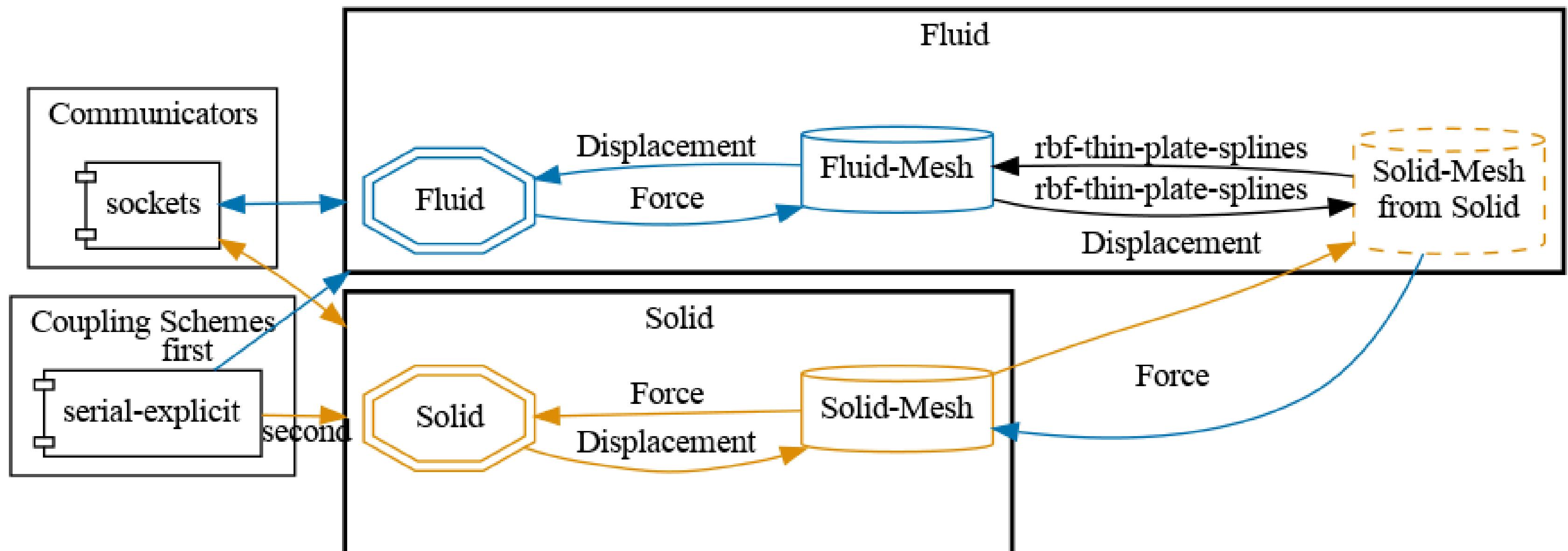
Configure the OpenFOAM adapter

```
1 // fluid-openfoam/system/preciceDict
2 preciceConfig "../../precice-config.xml";
3 participant Fluid;
4
5 modules (FSI);
6
7 interfaces {
8     Interface1 {
9         mesh Fluid-Mesh;
10        patches (flap);
11        locations faceCenters;
12
13        readData (Displacement);
14
15        writeData (Force);
16    };
17}
```

Configure the dealii adapter

```
1 // solid-dealii/parameters.prm
2
3 subsection precice configuration
4 # Cases: FSI3 or PF for perpendicular flap
5 set Scenario = PF
6
7 # Name of the precice configuration file
8 set precice config-file = ./precice-config.xml
9
10 # Name of the participant in the precice-config.xml file
11 set Participant name = Solid
12
13 # Name of the coupling mesh in the precice-config.xml file
14 set Mesh name = Solid-Mesh
15
16 # Name of the read data in the precice-config.xml file
17 set Read data name = Force
```

Configure preCICE



Configure preCICE

```
1 <data:vector name="Force" />
2 <data:vector name="Displacement" />
3
4 <mesh name="Fluid-Mesh" dimensions="2">
5   <use-data name="Force" />
6   <use-data name="Displacement" />
7 </mesh>
8
9 <mesh name="Solid-Mesh" dimensions="2">
10  <use-data name="Displacement" />
11  <use-data name="Force" />
12 </mesh>
13
14 <participant name="Fluid">
15   <export:vtk directory="results" />
16   <provide-mesh name="Fluid-Mesh" />
17   <receive-mesh name="Solid-Mesh" from="Solid" />
```

Notice the serial-explicit coupling scheme



Configure preCICE

```
1 <data:vector name="Force" />
2 <data:vector name="Displacement" />
3
4 <mesh name="Fluid-Mesh" dimensions="2">
5   <use-data name="Force" />
6   <use-data name="Displacement" />
7 </mesh>
8
9 <mesh name="Solid-Mesh" dimensions="2">
10  <use-data name="Displacement" />
11  <use-data name="Force" />
12 </mesh>
13
14 <participant name="Fluid">
15   <export:vtk directory="results" />
16   <provide-mesh name="Fluid-Mesh" />
17   <receive-mesh name="Solid-Mesh" from="Solid" />
```

Notice the serial-explicit coupling scheme



Configure preCICE

```
14 <participant name="Fluid">
15   <export:vtk directory="results" />
16   <provide-mesh name="Fluid-Mesh" />
17   <receive-mesh name="Solid-Mesh" from="Solid" />
18   <write-data name="Force" mesh="Fluid-Mesh" />
19   <read-data name="Displacement" mesh="Fluid-Mesh" />
20   <mapping:rbf direction="write"
21     from="Fluid-Mesh" to="Solid-Mesh"
22     constraint="conservative">
23     <basis-function:compact-polynomial-c6 support-radius="1." />
24   </mapping:rbf>
25   <mapping:rbf direction="read"
26     from="Solid-Mesh" to="Fluid-Mesh"
27     constraint="consistent">
28     <basis-function:compact-polynomial-c6 support-radius="1." />
29   </mapping:rbf>
30 </participant>
```

Notice the serial-explicit coupling scheme



Configure preCICE

```
14 <participant name="Fluid">
15   <export:vtk directory="results" />
16   <provide-mesh name="Fluid-Mesh" />
17   <receive-mesh name="Solid-Mesh" from="Solid" />
18   <write-data name="Force" mesh="Fluid-Mesh" />
19   <read-data name="Displacement" mesh="Fluid-Mesh" />
20   <mapping:rbf direction="write"
21     from="Fluid-Mesh" to="Solid-Mesh"
22     constraint="conservative">
23     <basis-function:compact-polynomial-c6 support-radius="1." />
24   </mapping:rbf>
25   <mapping:rbf direction="read"
26     from="Solid-Mesh" to="Fluid-Mesh"
27     constraint="consistent">
28     <basis-function:compact-polynomial-c6 support-radius="1." />
29   </mapping:rbf>
30 </participant>
```

Notice the serial-explicit coupling scheme



Configure preCICE

```
14 <participant name="Fluid">
15   <export:vtk directory="results" />
16   <provide-mesh name="Fluid-Mesh" />
17   <receive-mesh name="Solid-Mesh" from="Solid" />
18   <write-data name="Force" mesh="Fluid-Mesh" />
19   <read-data name="Displacement" mesh="Fluid-Mesh" />
20   <mapping:rbf direction="write"
21     from="Fluid-Mesh" to="Solid-Mesh"
22     constraint="conservative">
23     <basis-function:compact-polynomial-c6 support-radius="1." />
24   </mapping:rbf>
25   <mapping:rbf direction="read"
26     from="Solid-Mesh" to="Fluid-Mesh"
27     constraint="consistent">
28     <basis-function:compact-polynomial-c6 support-radius="1." />
29   </mapping:rbf>
30 </participant>
```

Notice the serial-explicit coupling scheme



Configure preCICE

```
31
32 <participant name="Solid">
33   <provide-mesh name="Solid-Mesh" />
34   <receive-data name="Displacement" mesh="Solid-Mesh" />
35   <read-data name="Force" mesh="Solid-Mesh" />
36   <watch-point mesh="Solid-Mesh" name="Flap-Tip" coordinate="0.0;1" /
37 </participant>
38
39 <m2n:sockets acceptor="Fluid" connector="Solid"
40           exchange-directory=".." /> <!-- initial handshake -->
41
42 <coupling-scheme:serial-explicit>
43   <participants first="Fluid" second="Solid" />
44   <max-time value="1.5" />
45   <time-window-size value="0.01" />
46   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
47   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
48 </coupling-scheme:serial-explicit>
```

Notice the serial-explicit coupling scheme



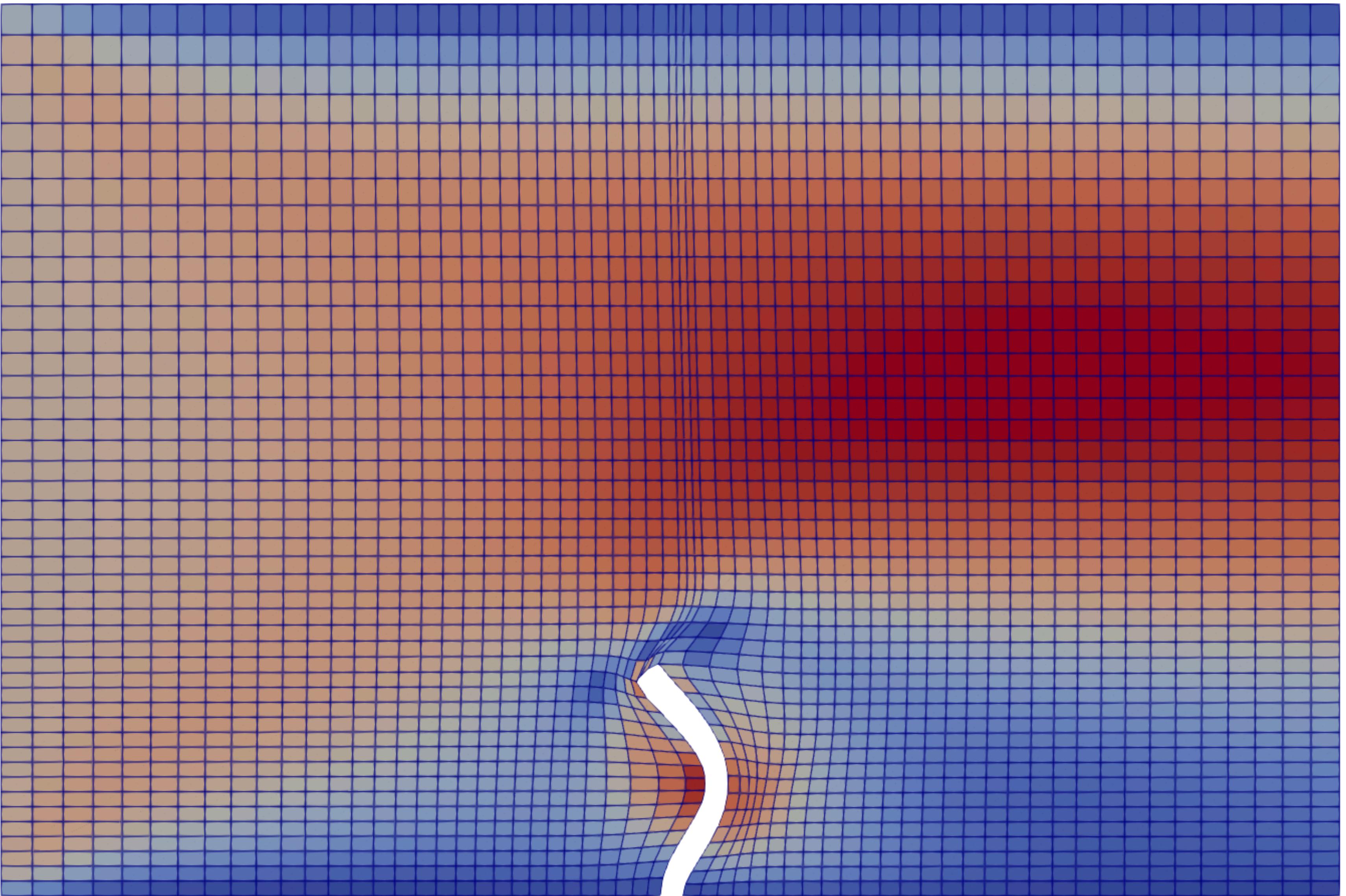
Configure preCICE

```
32 <participant name="Solid">
33   <provide-mesh name="Solid-Mesh" />
34   <receive-data name="Displacement" mesh="Solid-Mesh" />
35   <read-data name="Force" mesh="Solid-Mesh" />
36   <watch-point mesh="Solid-Mesh" name="Flap-Tip" coordinate="0.0;1" /
37 </participant>
38
39 <m2n:sockets acceptor="Fluid" connector="Solid"
40           exchange-directory=".." /> <!-- initial handshake -->
41
42 <coupling-scheme:serial-explicit>
43   <participants first="Fluid" second="Solid" />
44   <max-time value="1.5" />
45   <time-window-size value="0.01" />
46   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
47   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
48 </coupling-scheme:serial-explicit>
```

Notice the serial-explicit coupling scheme



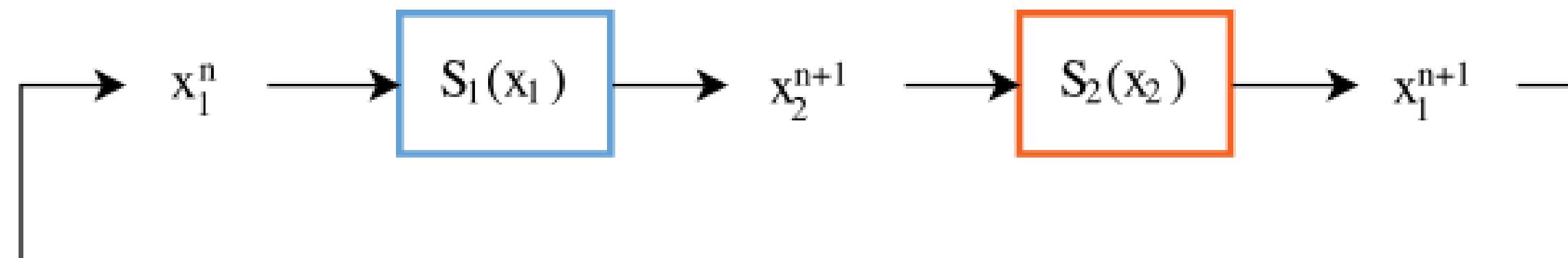
```
vagrant@precicevm:~/tutor... fsi-course - File Manager vagrant@precicevm: ~/tutorials/fsi-course/fluid-openfoam vagrant@precicevm: ~/tutorials/fsi-course 160x3  
vagrant@precicevm:~/tutorials/fsi-course$ ls clean-tutorial.sh fluid-openfoam plot-displacement.sh precice-config.xml solid-calculix  
vagrant@precicevm:~/tutorials/fsi-course$ vagrant@precicevm:~/tutorials/fsi-course/fluid-openfoam 79x44  
vagrant@precicevm:~/tutorials/fsi-course/fluid-openfoam$ tree .  
+-- 0  
|   +-- p  
|   +-- phi  
|   +-- pointDisplacement  
|   +-- U  
+-- clean.sh  
+-- constant  
|   +-- dynamicMeshDict  
|   +-- transportProperties  
|   +-- turbulenceProperties  
+-- run.sh  
+-- system  
|   +-- blockMeshDict  
|   +-- controlDict  
|   +-- decomposeParDict  
|   +-- fvSchemes  
|   +-- fvSolution  
|   +-- preciceDict  
  
3 directories, 15 files  
vagrant@precicevm:~/tutorials/fsi-course/fluid-openfoam$ ./run.sh  
vagrant@precicevm: ~/tutorials/fsi-course/fluid-openfoam  
vagrant@precicevm: ~/tutorials/fsi-course 160x3  
vagrant@precicevm:~/tutorials/fsi-course/solid-calculix 79x44  
vagrant@precicevm:~/tutorials/fsi-course/solid-calculix$ ls all.msh config.yml flap.inp frequency.inp run.sh  
clean.sh fix1_beam.nam flap_modal.inp interface_beam.nam  
vagrant@precicevm:~/tutorials/fsi-course/solid-calculix$ ./run.sh
```



When we decrease the solid density further to 1, the participants have stronger coupling, then:

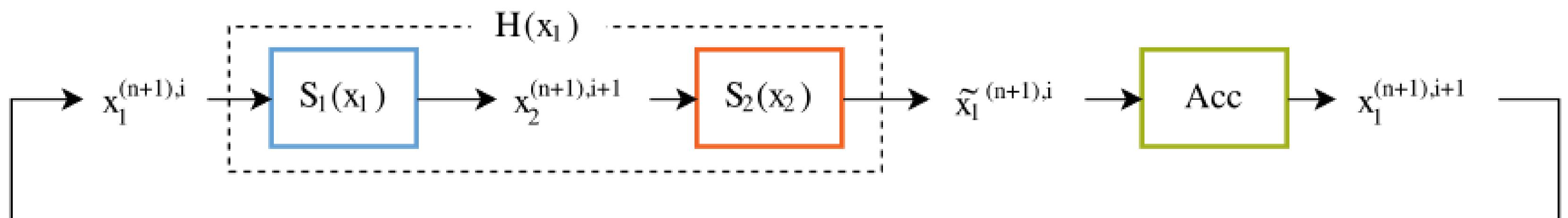
The image shows two terminal windows side-by-side. Both windows have a red header bar with the text "chenju@lapsgs05: ~/Desktop/perpendicular_flap/serial-explicit/fluid-openfoam 94x49". The left window's header also includes "[10:07:12]". The right window's header also includes "[10:07:14]". The main body of both windows is dark gray and mostly empty, indicating that the simulation is running or has just started.

What we did so far: serial-explicit scheme



$$n \leftarrow n + 1$$

Let's try an implicit coupling scheme



$$i \leftarrow i + 1, \quad S_1 \leftarrow S_1^{(n)}, \quad S_2 \leftarrow S_2^{(n)}$$

Configure the coupling scheme

```
1 <coupling-scheme:serial-implicit>
2   <participants first="Fluid" second="Solid" />
3   <max-time value="1.5" />
4   <time-window-size value="0.01" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <relative-convergence-measure limit="5e-3" data="Displacement" mesh
8   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
9   <max-iterations value="50" />
10 </coupling-scheme:serial-implicit>
```

Only change from explicit to implicit coupling

Configure the coupling scheme

```
1 <coupling-scheme:serial-implicit>
2   <participants first="Fluid" second="Solid" />
3   <max-time value="1.5" />
4   <time-window-size value="0.01" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <relative-convergence-measure limit="5e-3" data="Displacement" mesh
8   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
9   <max-iterations value="50" />
10 </coupling-scheme:serial-implicit>
```

Only change from explicit to implicit coupling

Configure the coupling scheme

```
1 <coupling-scheme:serial-implicit>
2   <participants first="Fluid" second="Solid" />
3   <max-time value="1.5" />
4   <time-window-size value="0.01" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <relative-convergence-measure limit="5e-3" data="Displacement" mesh=
8   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
9   <max-iterations value="50" />
10 </coupling-scheme:serial-implicit>
```

Only change from explicit to implicit coupling

Configure the coupling scheme

```
1 <coupling-scheme:serial-implicit>
2   <participants first="Fluid" second="Solid" />
3   <max-time value="1.5" />
4   <time-window-size value="0.01" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <relative-convergence-measure limit="5e-3" data="Displacement" mesh=
8   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
9   <max-iterations value="50" />
10 </coupling-scheme:serial-implicit>
```

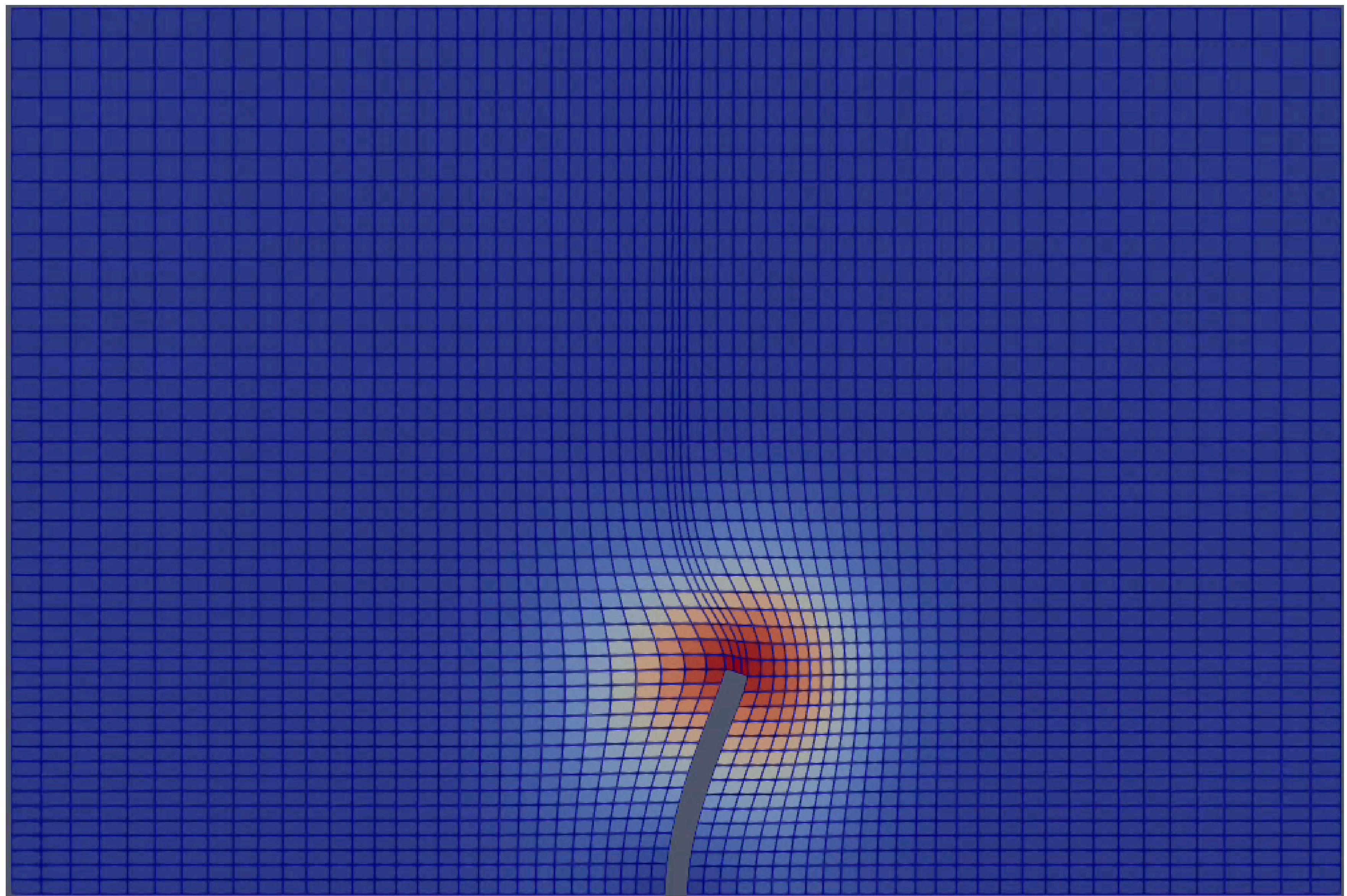
Only change from explicit to implicit coupling

Configure the coupling scheme

```
1 <coupling-scheme:serial-implicit>
2   <participants first="Fluid" second="Solid" />
3   <max-time value="1.5" />
4   <time-window-size value="0.01" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <relative-convergence-measure limit="5e-3" data="Displacement" mesh
8   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid
9   <max-iterations value="50" />
10 </coupling-scheme:serial-implicit>
```

Only change from explicit to implicit coupling

With solid density equal to 42:



With solid density equal to 1: breaks again!

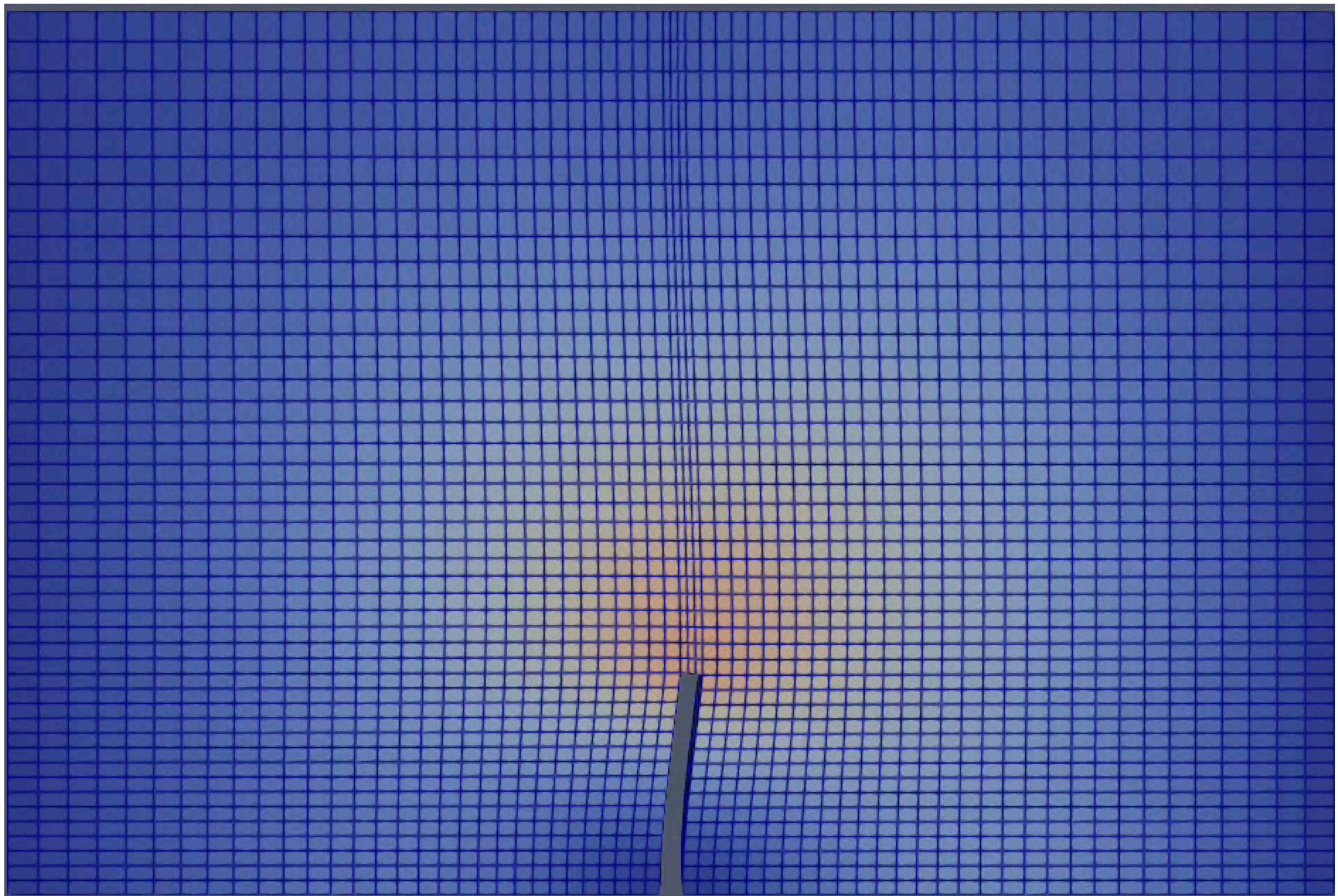
Still not enough stability for strong coupling!

Configure the coupling scheme

```
1 <coupling-scheme:serial-implicit>
2   <participants first="Fluid" second="Solid" />
3   <max-time value="1.5" />
4   <time-window-size value="0.01" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <relative-convergence-measure limit="5e-3" data="Displacement" mesh=
8   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
9   <max-iterations value="50" />
10  <acceleration:constant>
11    <relaxation value="0.5" />
12  </acceleration:constant>
13 </coupling-scheme:serial-implicit>
```

Simplest acceleration: constant under-relaxation

With solid density equal to 1:



Too slow! Can we do better?

Improvement 1: Aitken under-relaxation

```
<acceleration:aitken>
  <data name="Displacement" mesh="Solid-Mesh" />
  <initial-relaxation value="0.5" />
</acceleration:aitken>
```

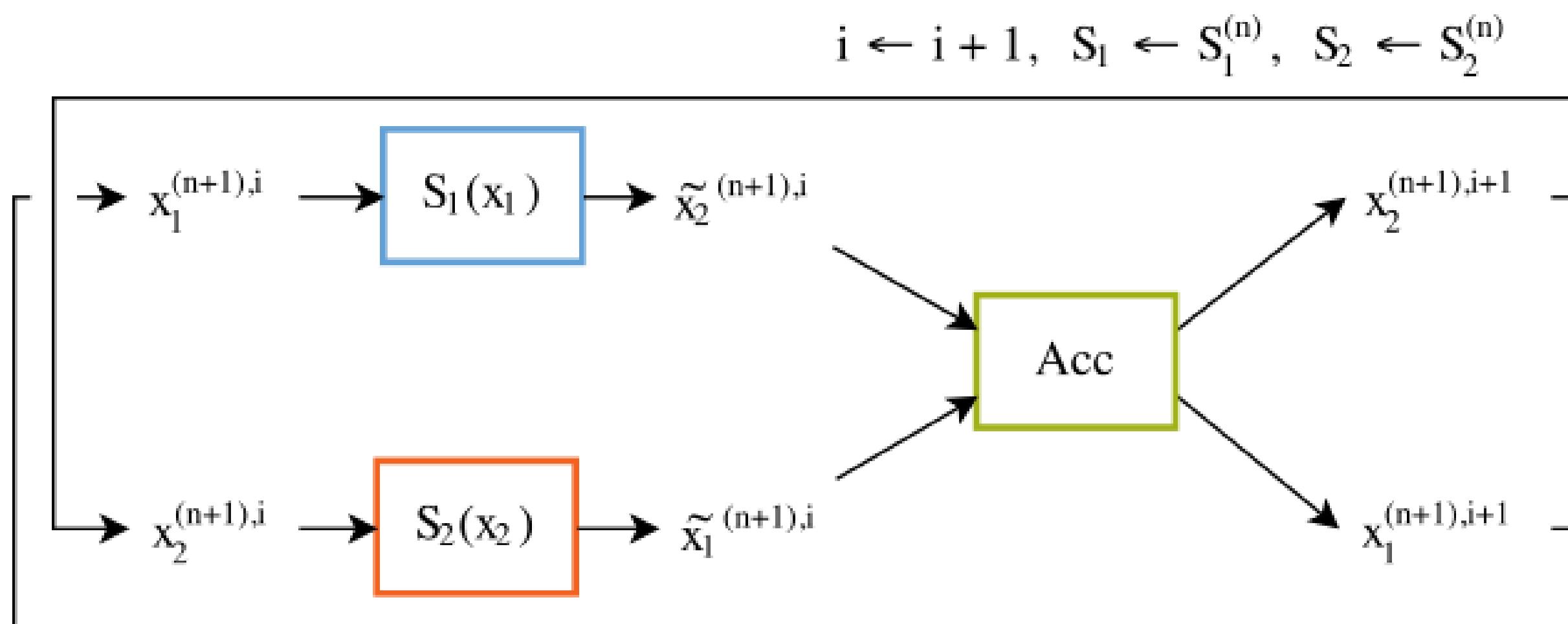
Improvement 2: Anderson acceleration

```
1 <acceleration:IQN-ILS>
2   <data name="Displacement" mesh="Solid-Mesh" />
3   <initial-relaxation value="0.5" />
4   <preconditioner type="residual-sum" />
5   <filter type="QR2" limit="1e-2" />
6   <max-used-iterations value="100" />
7   <time-windows-reused value="15" />
8 </acceleration:IQN-ILS>
```

Improvement 2: Anderson acceleration

```
1 <acceleration:IQN-ILS>
2   <data name="Displacement" mesh="Solid-Mesh" />
3   <initial-relaxation value="0.5" />
4   <preconditioner type="residual-sum" />
5   <filter type="QR2" limit="1e-2" />
6   <max-used-iterations value="100" />
7   <time-windows-reused value="15" />
8 </acceleration:IQN-ILS>
```

There are also parallel schemes



Can use fields from both / all participants

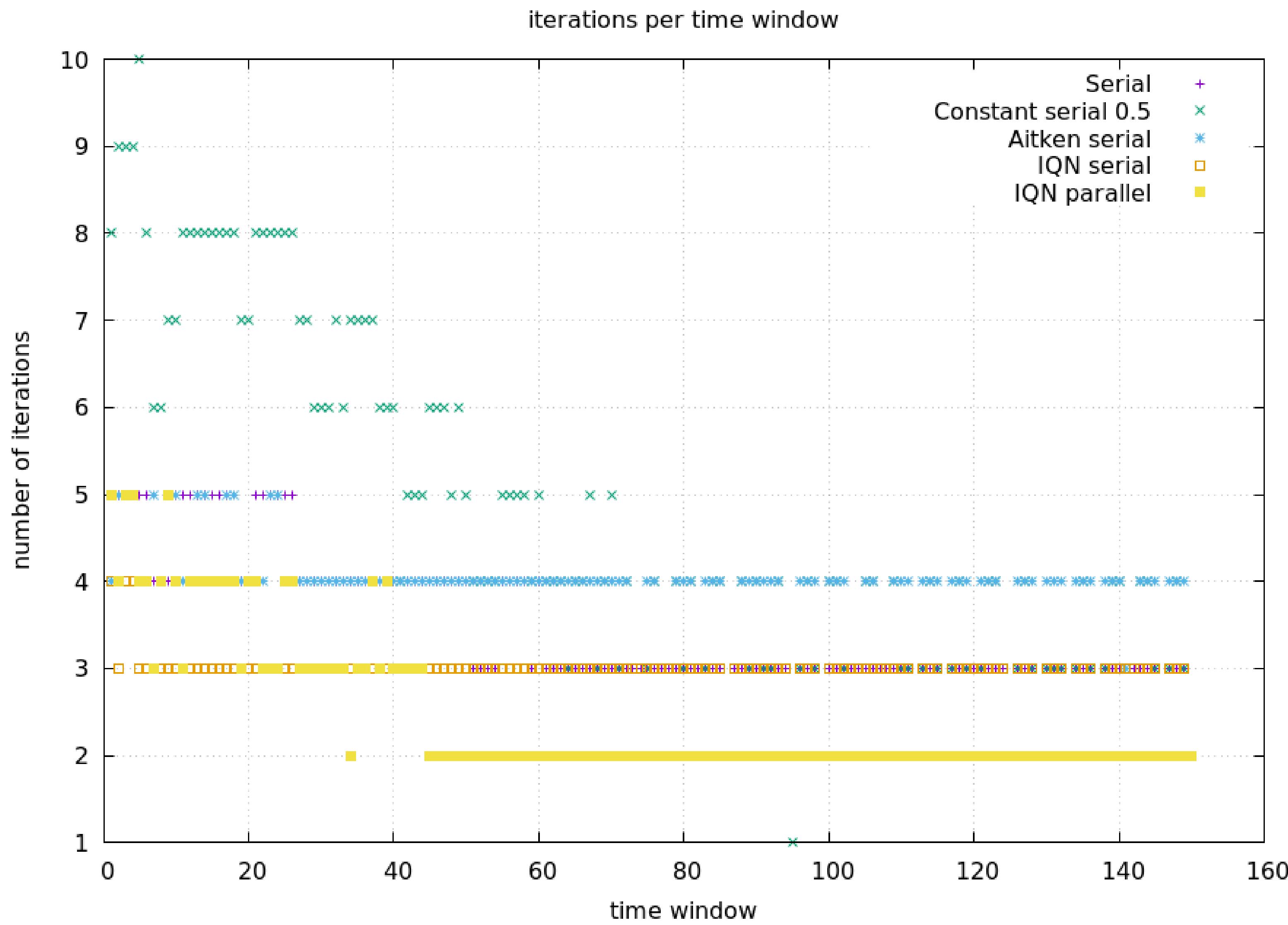
Improvement 3: Parallel coupling schemes

```
1 <coupling-scheme:parallel-implicit>
2   <time-window-size value="0.01" />
3   <max-time value="1.5" />
4   <participants first="Fluid" second="Solid" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <max-iterations value="50" />
8   <relative-convergence-measure limit="5e-3" data="Displacement" mesh
9   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
10  <acceleration:IQN-ILS>
11    <data name="Displacement" mesh="Solid-Mesh" />
12    <data name="Force" mesh="Solid-Mesh" />
13    <initial-relaxation value="0.5" />
14    <preconditioner type="residual-sum" />
15    <filter type="QR2" limit="1e-2" />
16    <max-used-iterations value="100" />
17    <time-windows-reused value="15" />
```

Improvement 3: Parallel coupling schemes

```
3   <max-time value="1.5" />
4   <participants first="Fluid" second="Solid" />
5   <exchange data="Force" mesh="Solid-Mesh" from="Fluid" to="Solid" />
6   <exchange data="Displacement" mesh="Solid-Mesh" from="Solid" to="Fl
7   <max-iterations value="50" />
8   <relative-convergence-measure limit="5e-3" data="Displacement" mesh
9   <relative-convergence-measure limit="5e-3" data="Force" mesh="Solid"
10  <acceleration:IQN-ILS>
11    <data name="Displacement" mesh="Solid-Mesh" />
12    <data name="Force" mesh="Solid-Mesh" />
13    <initial-relaxation value="0.5" />
14    <preconditioner type="residual-sum" />
15    <filter type="QR2" limit="1e-2" />
16    <max-used-iterations value="100" />
17    <time-windows-reused value="15" />
18  </acceleration:IQN-ILS>
19 </coupling-scheme:parallel-implicit>
```

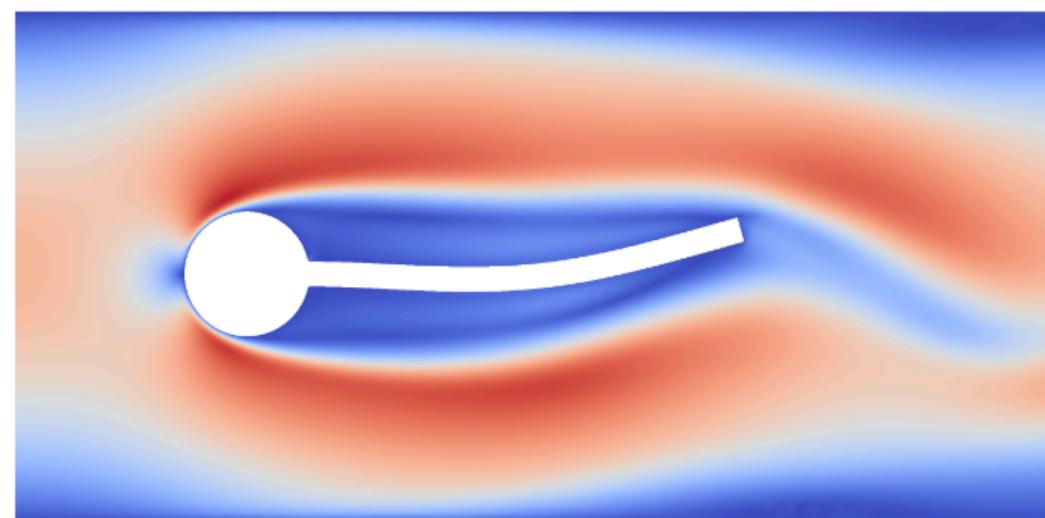
	d = 42	d = 1
serial-explicit	div	div
serial-implicit (const)	4.48	5.27
serial-implicit (Aitken)	3.74	5.30
serial-implicit (IQN)	2.95	3.07
parallel-implicit (IQN)	2.46	2.33



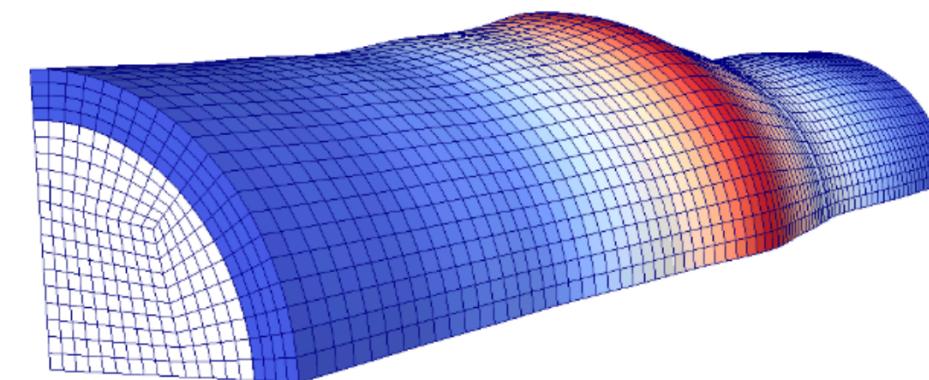
Beware: Configuration not fine-tuned, not rigorous comparison, application context.

Rigorous comparison from the literature

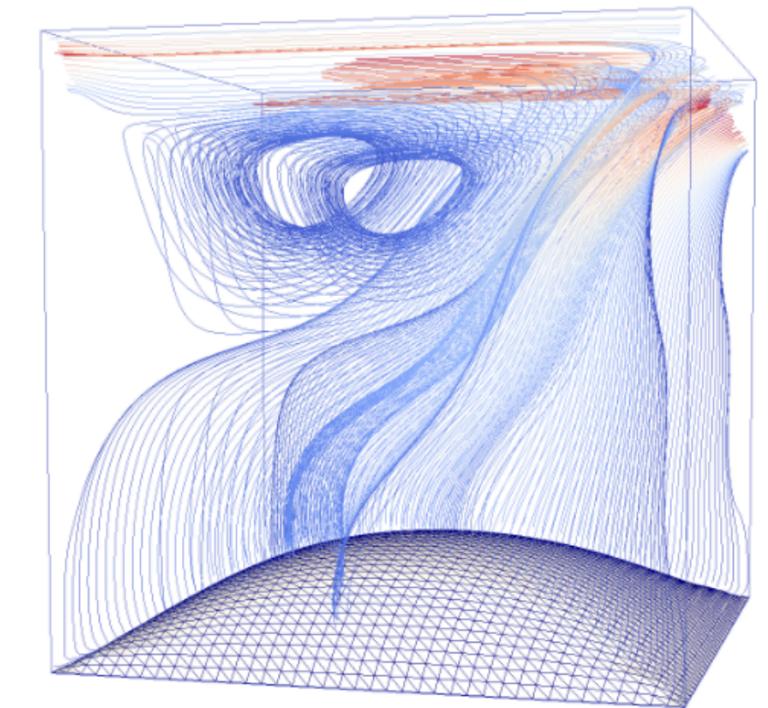
FSI3



3D-Tube



Driven Cavity



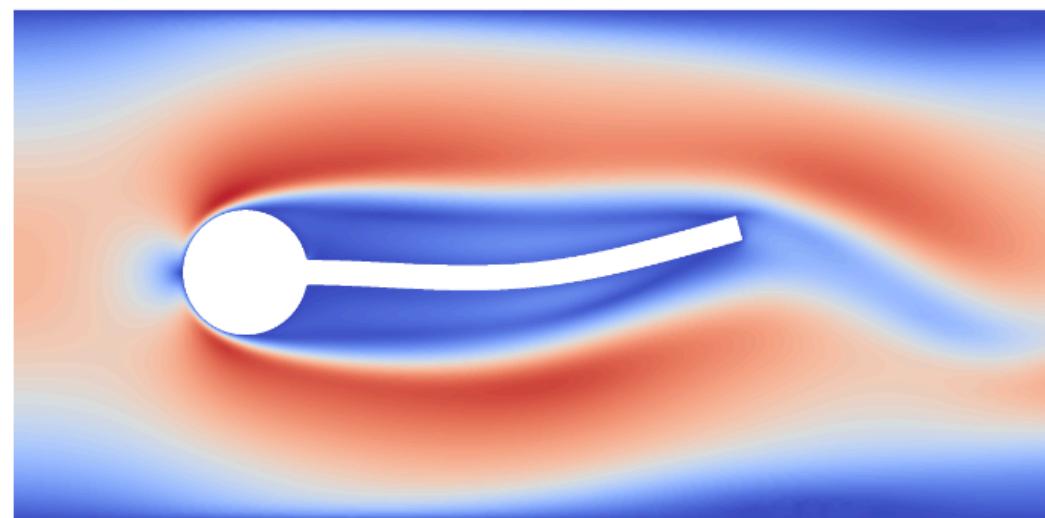
Mean Iterations	Aitken	Quasi-Newton
FSI3	17.0	3.7
3D-Tube	Div.	7.5
Driven Cavity	7.4	3.0

[http://doi.org/
10/m9ff](http://doi.org/10/m9ff)

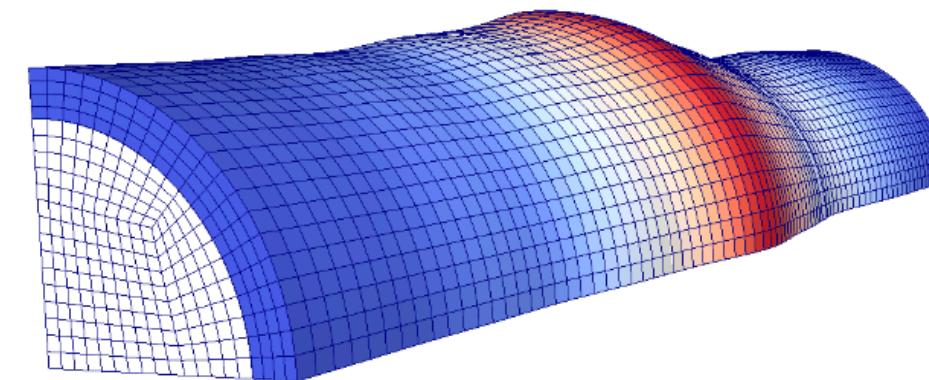
Source: Dissertation of Benjamin Uekermann (2016)

Rigorous comparison from the literature

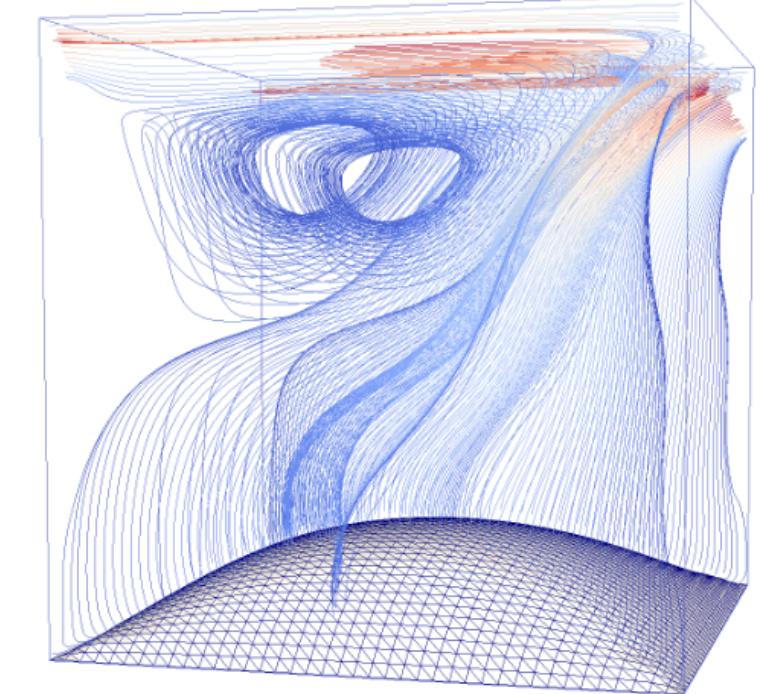
FSI3



3D-Tube



Driven Cavity



Mean Iterations	Serial Coupling	Parallel Coupling
FSI3	3.7	3.3
3D-Tube	7.5	10.4
Driven Cavity	3.0	2.0

Source: Dissertation of Benjamin Uekermann (2016)

Resources

Everything is open and user-centered

- Developed in the open: github.com/precice
- Documentation and many examples in one place
- (Diamond) open-access publications
- Public recordings of workshop talks
- Open collaboration, issues, projects, feedback, ...

Extra effort to maintain and update

Start here: precice.org



Save the date: [preCICE Workshop 2025, Hamburg, Sep 8-12](#)

Welcome to preCICE

The coupling library for partitioned multi-physics simulations.

[Star on Github](#) ★ 777

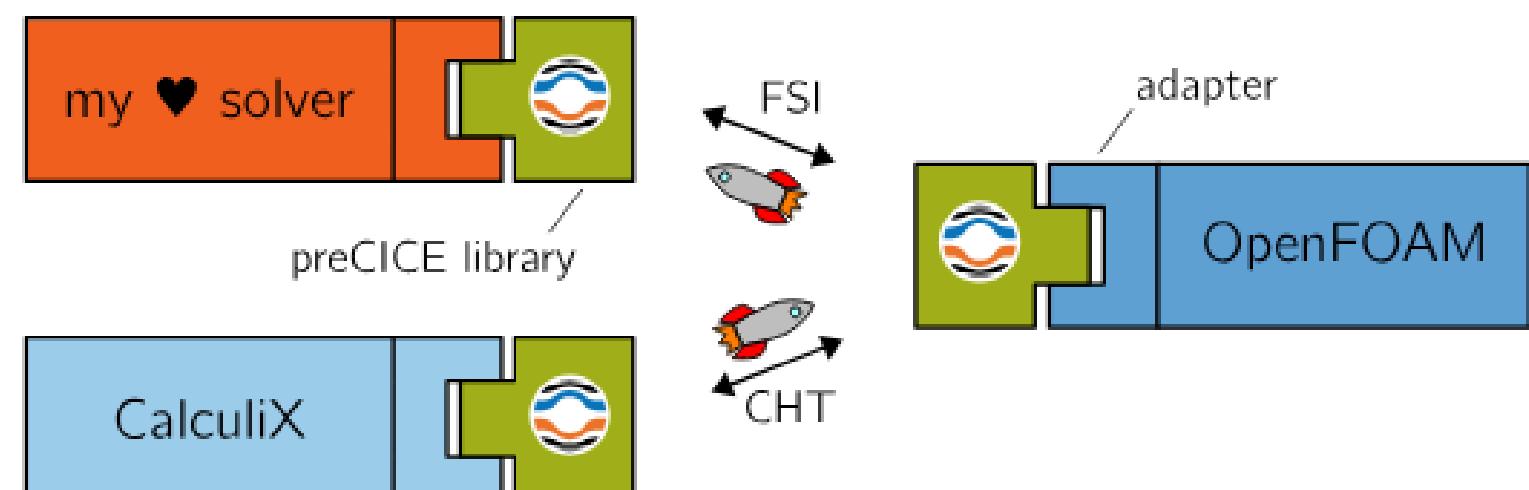
[Latest v3.1.2 \(Jun 6, 2024\)](#) 

[Get started >](#)

preCICE is an **open-source 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.

The software offers convenient methods for transient equation coupling, communication, and data mapping.



Built & hosted on GitHub Pages.
Last 7 days: 700+ visitors, 4k pageviews



Documentation



Quickstart Docs Tutorials Community Blog ↗ About

Search ...

Search by algolia



Save the date: preCICE Workshop 2025, Hamburg, Sep 8-12

Documentation

Fundamentals

Overview

Terminology

License information

Literature guide

Roadmap

Previous versions

Installation

Configuration

Tooling

Provided adapters

Couple your code

Running simulations

Dev docs

Documentation meta

The preCICE documentation

Summary: This page gives an overview of the complete preCICE documentation, including building, configuration, literature, the API, and much more.

Table of Contents

- [The big picture](#)
- [Where to find what](#)

Edit me ↗

Updated 15 May 24

The big picture

preCICE stands for Precise Code Interaction Coupling Environment. Its main component is a library that can be used for partitioned multi-physics simulations, including, but not restricted to fluid-structure interaction and conjugate heat transfer simulations. Partitioned (as opposite to monolithic) means that preCICE couples existing programs (solvers) which simulate 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, reusing existing components. preCICE runs efficiently on a wide spectrum of systems, from low-end laptops up to complete compute clusters and has [proven scalability](#) on 10000s of MPI Ranks.

Everything in one place, user-editable on GitHub, 100+ pages



Discuss & get help

 Home GitHub Mastodon YouTube Sign Up Log In

categories ► tags ► Categories Latest Top

Category	Topics
News News, announcements, "blog"-like posts	1 / month
Is preCICE for me? General questions regarding preCICE as a coupling solution.	2 / month
Installing preCICE Any issues with getting the preCICE library installed	1 / month
Using preCICE Directly using the preCICE API, configuring a new simulation	5 / month
Official adapters and tutorials Installing, configuring, and extending the official adapters, as well as running or modifying the tutorials	2 / month
Community projects Share your simulation cases for everybody to admire and try.	7
Jobs & theses market Jobs and thesis projects related to preCICE.	4

Latest

 Welcome to the preCICE Forum on Discourse Site Feedback	1 Nov 2019
 Linking problems with preCICE and boost Installing preCICE	1 3d
 Listening to your feedback: Updates on the ecosystem standardization News preeco	0 3d
 Reading the current time with the interface Using preCICE	2 7d
 Moving grid with deformation Is preCICE for me?	6 7d
 No deformation when using RBFMeshMotionSolver Non-preCICE fsi openfoam	22 7d
 Data exchange during sub-iteration	0

Active since October 2019, 500+ members



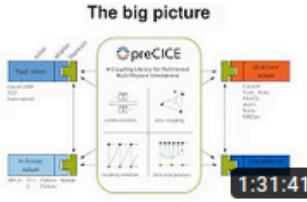
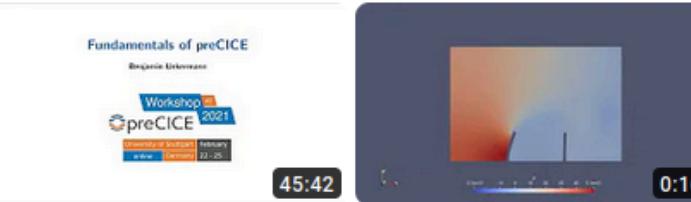
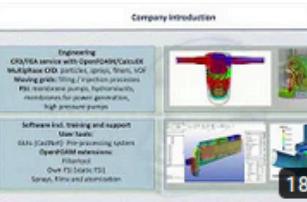
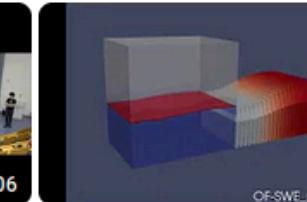
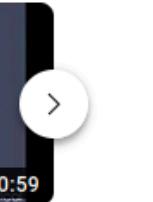
Learn: YouTube

 **preCICE Coupling**
@preCICECoupling 455 subscribers 42 videos
A coupling library for partitioned multi-physics simulations, including, but n... >

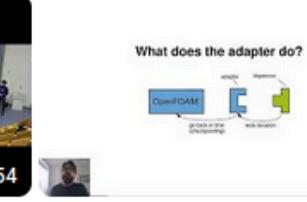
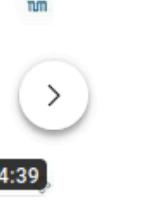
[Subscribe](#)

[HOME](#) [VIDEOS](#) [PLAYLISTS](#) [COMMUNITY](#) [CHANNELS](#) [ABOUT](#) [🔍](#)

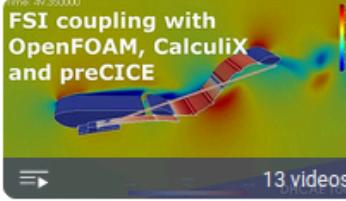
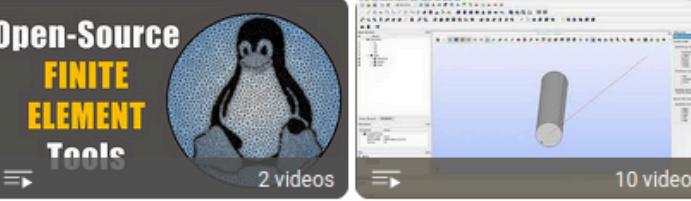
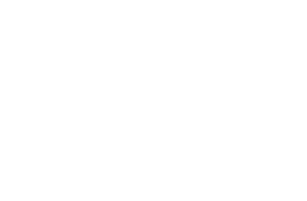
Popular videos ► [Play all](#)

 Multiphysics Modeling with the preCICE Coupling Library 2.3K views • 2 years ago	 Fundamentals of preCICE (Benjamin Uekermann,...) 1.4K views • 2 years ago	 Multicoupling with preCICE: two flaps in a channel... 1K views • 2 years ago	 Transfer of FSI coupling with preCICE, OpenFOAM and... 868 views • 2 years ago	 6-way coupling of DEM+CFD+FEM with preCIC... 781 views • 3 years ago	 Coupling of Shallow Water Equations and OpenFOAM... 700 views • 2 years ago
---	---	--	--	--	---

Talks about preCICE ► [Play all](#)
Conference talks about the preCICE coupling library

 Couple scientific simulation codes with preCICE A jour... FOSDEM 1.5K views • 4 years ago	 Fundamentals of preCICE (Benjamin Uekermann,...) preCICE Coupling 1.4K views • 2 years ago	 What is new in preCICE? (Frédéric Simonis, preCICE...) preCICE Coupling 122 views • 2 years ago	 SimTech and the Simulation of Large Systems Exzellenzcluster SimTech 669 views • 3 years ago	 High-order and multi-rate time stepping with preCICE... preCICE Coupling 231 views • 3 years ago	 The OpenFOAM-preCICE adapter (Gerasimos...) preCICE Coupling 366 views • 2 years ago
--	--	---	--	--	--

Community

 Simulations using preCICE Playlist · preCICE Coupling View full playlist	 Users talking about preCICE Playlist · preCICE Coupling View full playlist	 FSI for vascular flows using OpenFOAM, preCICE, and... Playlist · Torsten Schenkel View full playlist
---	--	---

Active since 2020, 600+ subscribers, 50+ videos

Further channels

- Mastodon: <https://fosstodon.org/@precice>
- Bluesky: <https://bsky.app/profile/precice.org>
- LinkedIn: <https://www.linkedin.com/groups/9073912/>
- RG: <https://www.researchgate.net/project/preCICE>

Related publications

European Commission | Research and Innovation

Open Research Europe

Search

Browse Gateways & Collections How to Publish About Resource Hub Blog

451 Views 180 Downloads 34 Citations

Cite Download

Home > Articles > preCICE v2: A sustainable and user-friendly coupling library

SOFTWARE TOOL ARTICLE

REVISED preCICE v2: A sustainable and user-friendly coupling library

[version 2; peer review: 2 approved]

Gerasimos Chourdakis , Kyle Davis , Benjamin Rodenberg , Miriam Schulte*, Frédéric Simonis , Benjamin Uekermann , Georg Abrams, Hans-Joachim Bungartz, Lucia Cheung Yau, Ishaaan Desai , Konrad Eder, Richard Hertich, Florian Lindner , Alexander Rusch , Dmytro Sashko, David Schneider , Amin Totounferoush , Dominik Volland, Peter Vollmer , Oguz Ziya Koseomur

This article is included in Horizon 2020 gateway

This article is included in Marie-Sklodowska-Curie Actions (MSCA) gateway

Article Authors Metrics

Open Peer Review Approval Status

Version 2 (Revision) 30 Sep 22

Version 1 29 Apr 22

1. Axelle Viré, Delft University of Technology, Cambridge, UK

2. Garth Wells

Comments on this article [All Comments \(0\)](#)

http://doi.org/
10/m9d8

OpenFOAM® Journal

Current Archives Announcements About Authors

Search

Home / Archives / Vol. 3: OpenFOAM® Journal 2023 / Full Papers

OpenFOAM-preCICE: Coupling OpenFOAM with External Solvers for Multi-Physics Simulations

Gerasimos Chourdakis
Technical University of Munich
<https://orcid.org/0000-0002-3977-1385>

David Schneider
University of Stuttgart
<https://orcid.org/0000-0002-3487-9688>

Benjamin Uekermann
University of Stuttgart
<https://orcid.org/0000-0002-1314-9969>

DOI: <https://doi.org/10.51560/ofj.v3.88>

standard, unmodified OpenFOAM adapter arbitrary code External simulation conjugate heat transfer fluid-structure interaction and more

PDF Discussion Forum Abstract Video

Sponsors

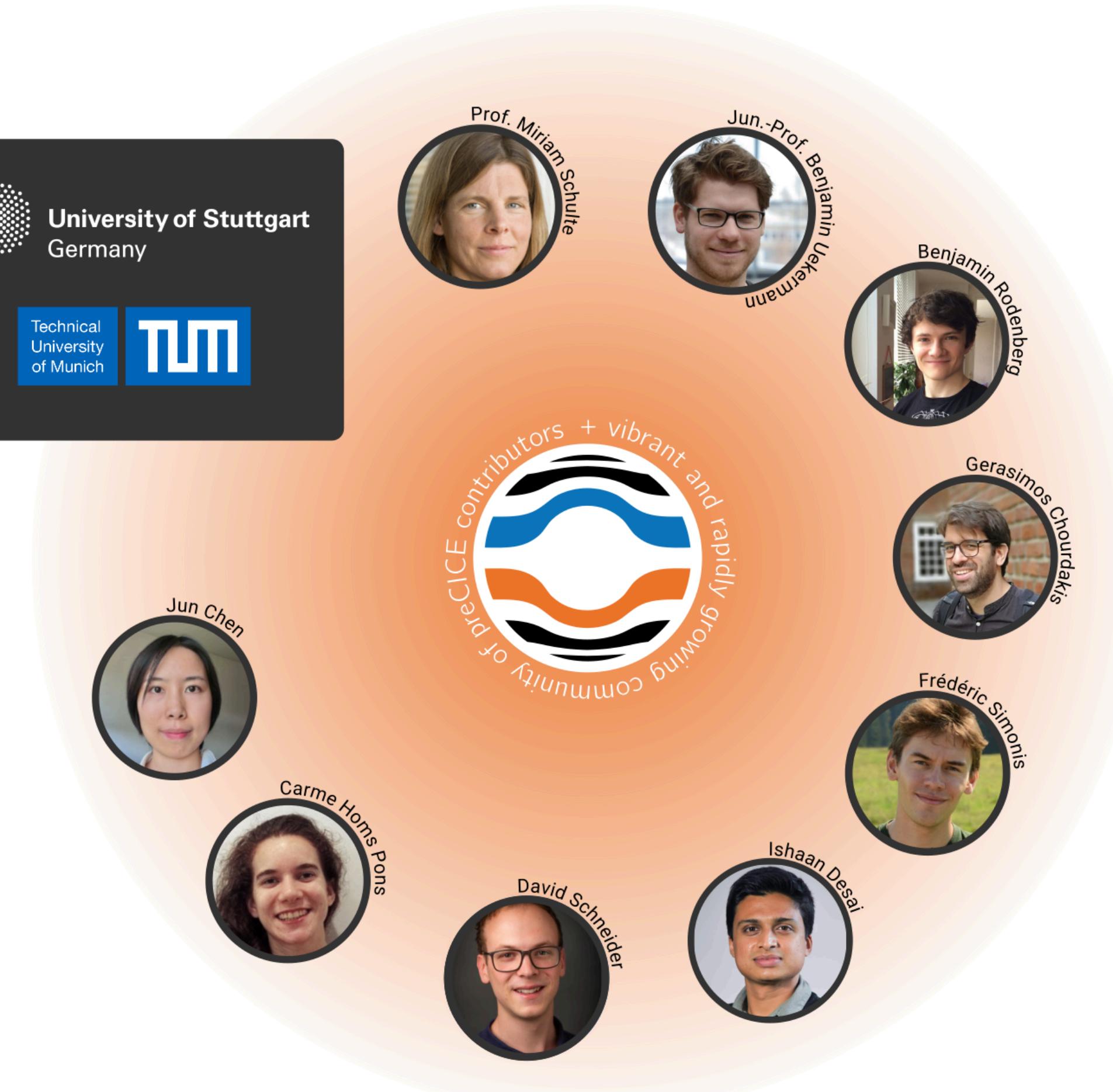
GOMPUTE
A GRIDCORE COMPANY

upstreamCFD

[http://doi.org/
10/m9fd](http://doi.org/10/m9fd)



The people behind preCICE



Current PhD topics (TUM)

- Gerasimos Chourdakis: Ecosystem, geometric multi-scale
- Benjamin Rodenberg: Time interpolation
- Frédéric Simonis: Dynamic coupling meshes

Current PhD topics (Univ. Stuttgart)

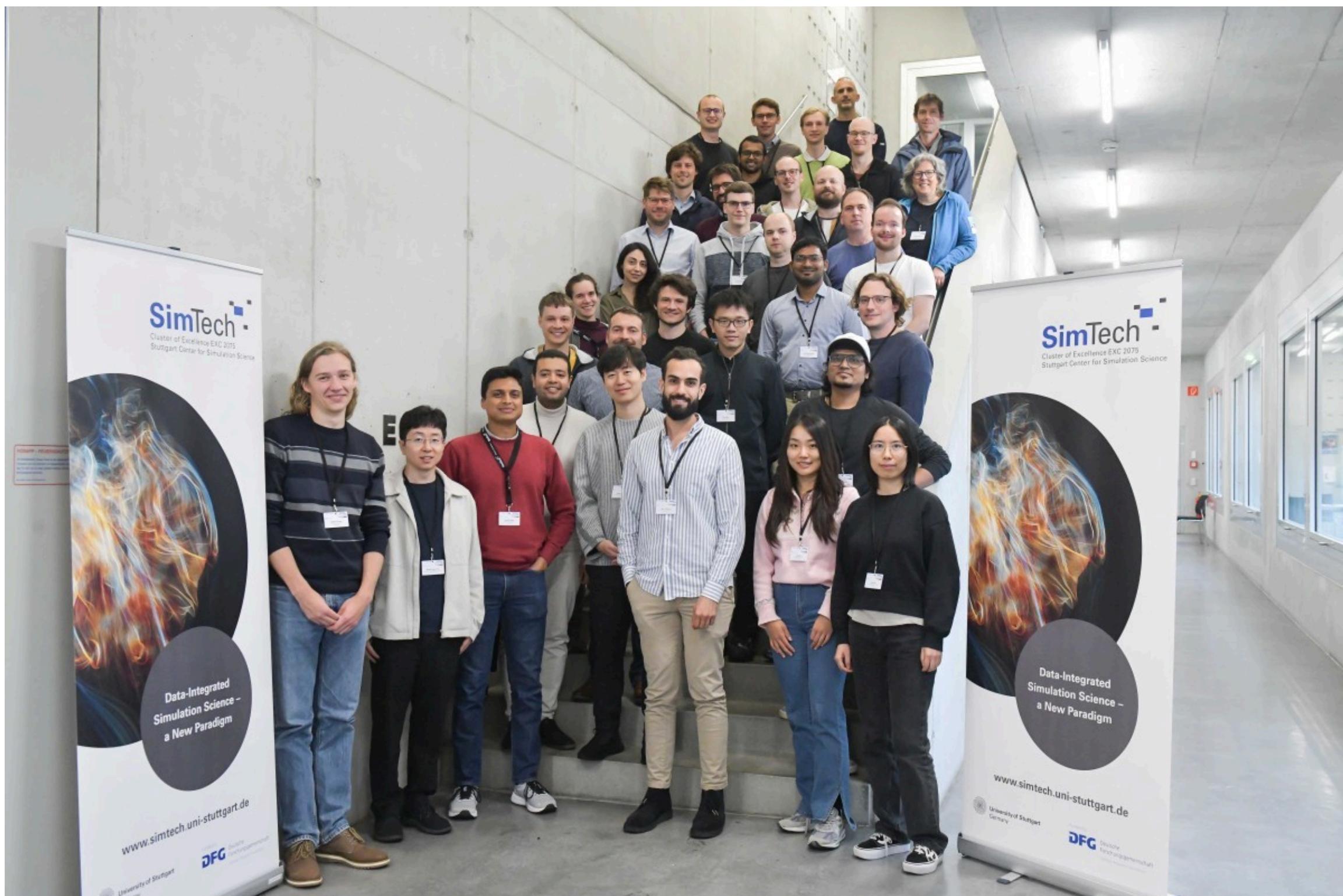
- David Schneider: Data mapping (GPUs, PoU RBF)
- Ishaan Desai: Adaptive macro-micro simulations
- Jun Chen: Porous media macro-micro simulations
- Carme Homs Pons: Musculoskeletal multi-scale simulations (OpenDiHu)

Further research and development

- Coupling with Functional Mock-Up units
- Non-mesh-related global data exchange
- Volume coupling (overlapping domains)
- Automatic acceleration configuration



Meet the people





Workshop #6

preCICE 2025

HSU Hamburg

Germany

September

8 - 12



preCICE is free because of



Research Software
Sustainability



Bundesministerium
für Wirtschaft
und Energie



EXC 2075
SimTech



This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754462

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

Summary

1. Both for developers and for users
2. Many efficient algorithms are one setting away

Gerasimos Chourdakis (US/TUM), Jun Chen(US), + many more (see precice.org/about)



This work is licensed under a [Creative Commons Attribution 4.0 International License](#).
Get these slides.