

An introduction to the preCICE coupling library

Frédéric Simonis¹, Benjamin Uekermann²

6. June 2023

¹ Chair of Scientific Computing
Technical University of Munich

² Usability and Sustainability of Simulation Software
University of Stuttgart



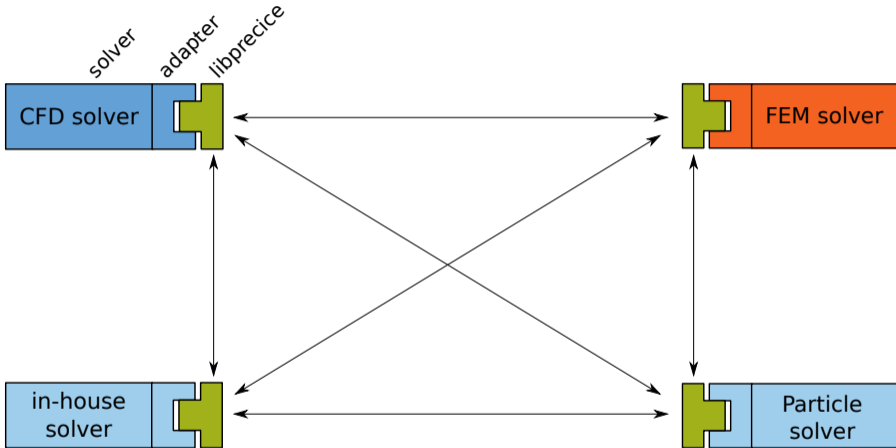
The Big Picture



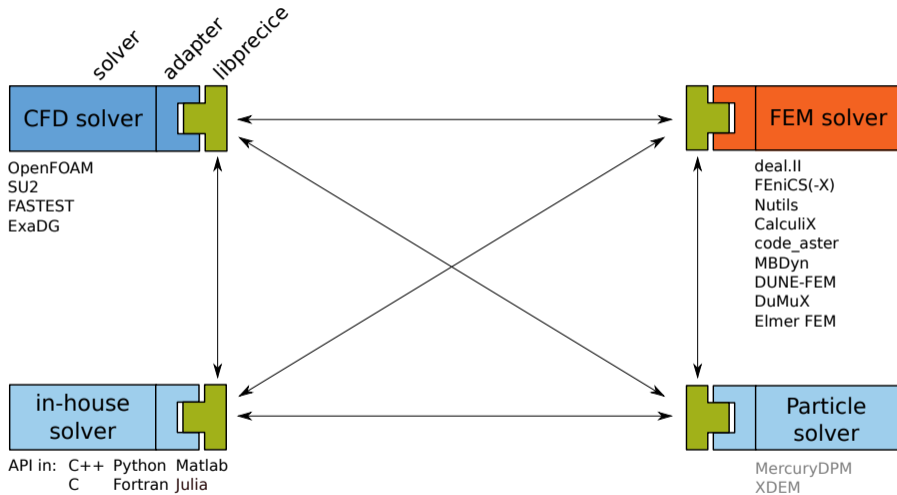
The Big Picture



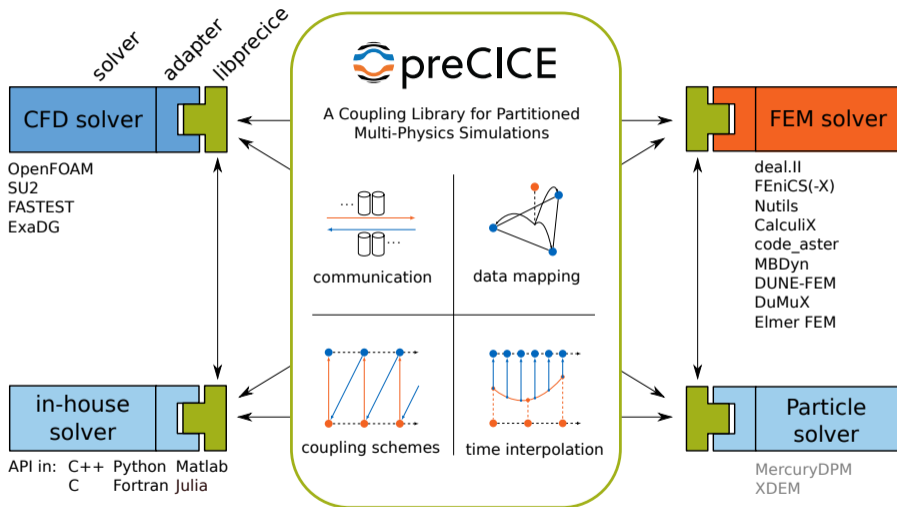
The Big Picture



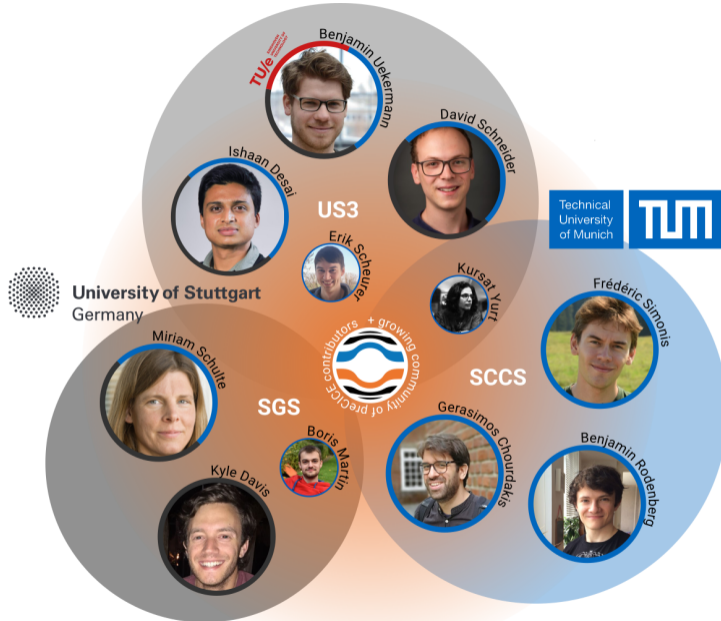
The Big Picture



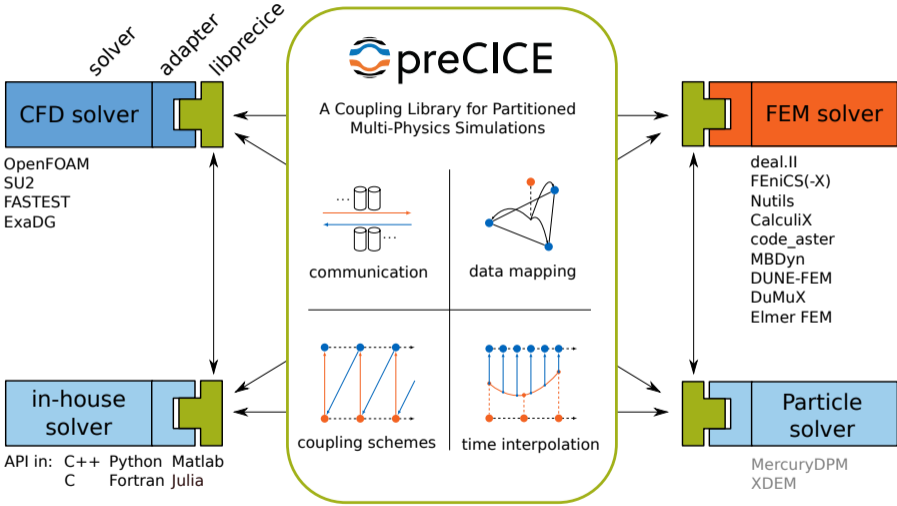
The Big Picture



Maintainers



preCICE Application Programming Interface



Application Programming Interface

```
1
2
3
4
5
6
7
8
9 while not done: # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14
15     u = solve_time_step(u, dt) # returns new solution
16
17
18
19
```

Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4
5
6
7
8
9 while not done: # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14
15     u = solve_time_step(u, dt) # returns new solution
16
17
18
19
```

Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7 participant.initialize()
8
9 while not done: # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14
15     u = solve_time_step(u, dt) # returns new solution
16
17
18
19
```

Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7 participant.initialize()
8
9 while participant.is_coupling_ongoing(): # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14
15     u = solve_time_step(u, dt) # returns new solution
16
17
18
19
```

Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7 participant.initialize()
8
9 while participant.is_coupling_ongoing(): # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14     displacements = participant.read_data(mesh, 'Displacement', vertex_ids, dt)
15     u = solve_time_step(u, dt, displacements) # returns new solution
16
17
18
19
```

Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7 participant.initialize()
8
9 while participant.is_coupling_ongoing(): # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14     displacements = participant.read_data(mesh, 'Displacement', vertex_ids, dt)
15     u = solve_time_step(u, dt, displacements) # returns new solution
16     forces = compute_forces(u) # returns 2D array with shape (n, dim)
17     participant.write_data(mesh, 'Force', vertex_ids, forces)
18
19
```

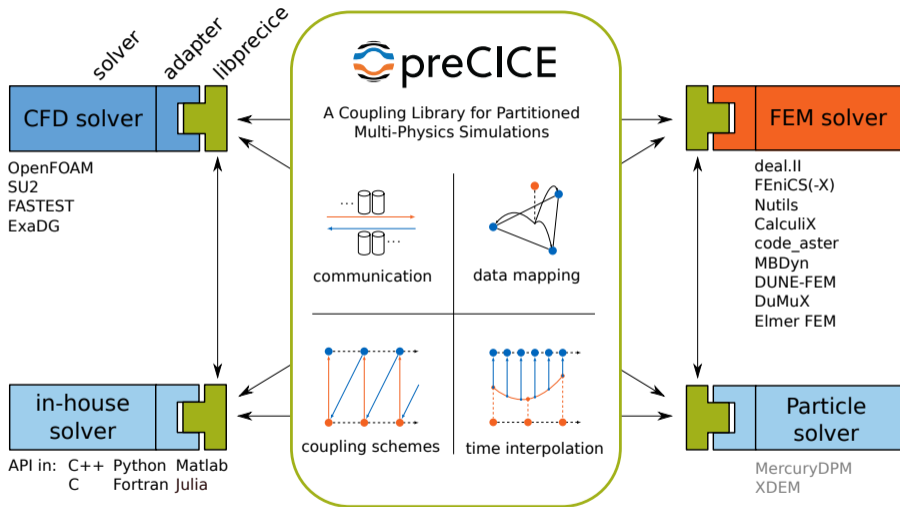
Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7 participant.initialize()
8
9 while participant.is_coupling_ongoing(): # main time loop
10     dt = compute_adaptive_dt()
11
12
13
14     displacements = participant.read_data(mesh, 'Displacement', vertex_ids, dt)
15     u = solve_time_step(u, dt, displacements) # returns new solution
16     forces = compute_forces(u) # returns 2D array with shape (n, dim)
17     participant.write_data(mesh, 'Force', vertex_ids, forces)
18
19     participant.advance(dt)
```

Application Programming Interface

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7 participant.initialize()
8
9 while participant.is_coupling_ongoing(): # main time loop
10     solver_dt = compute_adaptive_dt()
11     precice_dt = participant.get_max_time_step_size()
12     dt = min(solver_dt, precice_dt)
13
14     displacements = participant.read_data(mesh, 'Displacement', vertex_ids, dt)
15     u = solve_time_step(u, dt, displacements) # returns new solution
16     forces = compute_forces(u) # returns 2D array with shape (n, dim)
17     participant.write_data(mesh, 'Force', vertex_ids, forces)
18
19     participant.advance(dt)
```


preCICE Resources



Community

Everything

My Posts

Admin

More

Categories

Uncategorized

All categories

Tags

code_aster

mbdyn

All tags

Messages

Inbox

admins

moderators

all categories

all tags

Categories

Latest

New (78)

Unread (54)

Top

Bookmarks

✂

+ New Topic

Category Topics

News

32

News, announcements, "blog"-like posts

1 unread
2 new

Is preCICE for me?

44

General questions regarding preCICE as a coupling solution.

1 unread
2 new

Installing preCICE

68

Any issues with getting the preCICE library installed

8 unread
7 new

Using preCICE

216

Using the preCICE API, configuring a new simulation

23 unread
23 new

Official adapters and tutorials

182

Installing, configuring, extending adapters, running or modifying the tutorials

18 unread
35 new

Community projects

3

Share your simulation cases for everybody to admire and try.

Jobs & theses market

2

Jobs and thesis projects related to preCICE.

1 new

Latest

Mapping problem OpenFOAM-FEniCS on cluster

1
1h

Official adapters and tutorials
fenics fsi openfoam

Collapsible Tube (OpenFOAM+CalculiX)

18
19h

Fluid-structure interactions of 2D flexible thin palate

0
1d

Official adapters and tutorials
calculix fsi openfoam

preCICE for coupling OpenFOAM to FEnicsX with parallel computing

2
1d

Is preCICE for me?

Installation on Ubuntu 20.04

1
2d

Installing preCICE

OpenFOAM-CalculiX Coupling and olaFlow

6
2d

Official adapters and tutorials

Virtual Machine

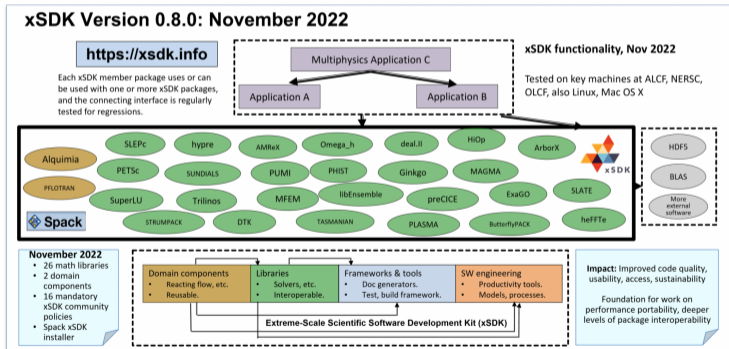
The screenshot shows a Virtual Machine environment. The desktop background is a dark purple color. On the left side, there is a vertical dock with various application icons. The main desktop area contains several folders: 'Trash', 'parview', 'File System', 'tutorials', 'Home', and 'Keyboard'. A terminal window is open in the foreground, displaying a list of tutorial cases:

```
vagrant@precicevm:~/tutorials$ ls -l
total 12
-rw-r--r-- 1 vagrant vagrant 4096 Aug 14 12:21 clean-all.sh
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 CODE_OF_CONDUCT.md
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 CONTRIBUTING.md
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 elastic-tube-1d
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 elastic-tube-3d
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 flow-over-heated-plate
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 flow-over-heated-plate-nearest-project
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 flow-over-heated-plate-steady-state
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 heat-exchanger
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 LICENSE
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 multiple-perpendicular-flaps
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 partitioned-elastic-beam
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 partitioned-heat-conduction
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 partitioned-heat-conduction-complex
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 partitioned-pipe
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 perpendicular-flap
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 quickstart
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 README.md
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 tools
-rw-r--r-- 1 vagrant vagrant 1024 Aug 14 12:21 turek-bron-fs13
vagrant@precicevm:~/tutorials$
```

The ParaView 3.8.1 application is running in the foreground. The main window displays a 3D visualization of a flow field around a central object. The color scale for 'U Magnitude' ranges from 6.2e-04 (blue) to 1.6e+00 (red). The interface includes a Pipeline Browser on the left, a Properties panel on the right, and various toolbars at the top. The title bar of the ParaView window reads 'ParaView 3.8.1'. The VM title bar reads 'precice-vm_default_1620120994434_70754 [Running] - Oracle VM VirtualBox'.

preCICE in xSDK

- ▶ Extreme-scale Scientific Software Development Kit
- ▶ Building via Spack
- ▶ Pre-installed on many supercomputers
- ▶ Among most important European math libraries



<https://xsdk.info/packages/>

The BIG changes

- ▶ Simpler and stricter API
- ▶ Clearer configuration
- ▶ Time-interpolation
- ▶ Partition of unity RBF mapping
- ▶ Performance analysis tools
- ▶ Connectivity optimization
- ▶ New Wendland C2/4 RBF
- ▶ Defaults for acceleration
- ▶ Adaptive-dynamic meshes
- ▶ Geometric multi-scale mappings

Funding

SimTech

SPPEXA

DFG



Federal Ministry
for the Environment, Nature Conservation,
Nuclear Safety and Consumer Protection

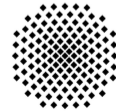


H2020 grant 754462

KONWIHR

TUM

TU/e








Universität
Stuttgart

Summary

- ▶ Plug-and-play coupling, minimally-invasive integration
- ▶ Scalability up to complete supercomputers
- ▶ State-of-the-art coupling numerics
- ▶ Extensive documentation
- ▶ Sustainable, extensible, and user-friendly ecosystem of software components

Resources

- ▶  simonis@cit.tum.de
- ▶  precice.org
- ▶  github.com/precice, LGPL3 license
- ▶  precice.discourse.group
- ▶  Chourdakis G, Davis K, Rodenberg B et al. 2022
preCICE v2: A sustainable and user-friendly coupling library
[doi:10.12688/openreseurope.14445.2](https://doi.org/10.12688/openreseurope.14445.2)

Application Programming Interface - Data Initialization

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7
8
9
10
11
12 participant.initialize()
13
14 ...
```

Application Programming Interface - Data Initialization

```
1 import precice
2 participant = precice.Participant('FluidSolver', 'precice-config.xml', rank, size)
3
4 mesh = 'Fluid-Mesh'
5 positions = ... #define interface mesh, 2D array with shape (n, dim)
6 vertex_ids = participant.set_mesh_vertices(mesh, positions)
7
8 if participant.requires_initial_data():
9     initial_forces = compute_initial_forces()
10    participant.write_data(mesh, 'Force', vertex_ids, initial_forces)
11
12 participant.initialize()
13
14 ...
```

Application Programming Interface - Implicit Coupling

```
1 ...
2 u = compute_initial_solution()
3 while participant.is_coupling_ongoing(): # main time loop
4
5
6
7
8     dt = ...
9
10    displacements = participant.read_data(...)
11    u = solve_time_step(u, dt, displacements)
12    forces = compute_forces(u)
13    participant.write_data(...)
14
15    participant.advance(dt)
16
17
18
```

Application Programming Interface - Implicit Coupling

```
1 ...
2 u = compute_initial_solution()
3 while participant.is_coupling_ongoing(): # main time loop
4
5     if participant.requires_writing_checkpoint():
6         checkpoint = u
7
8     dt = ...
9
10    displacements = participant.read_data(...)
11    u = solve_time_step(u, dt, displacements)
12    forces = compute_forces(u)
13    participant.write_data(...)
14
15    participant.advance(dt)
16
17    if participant.requires_reading_checkpoint():
18        u = checkpoint
```