Partitioned Fluid-Structure Interaction with the Coupling Library preCICE

CCP-WSI
Code Developers' Workshop 2
April 8, 2021

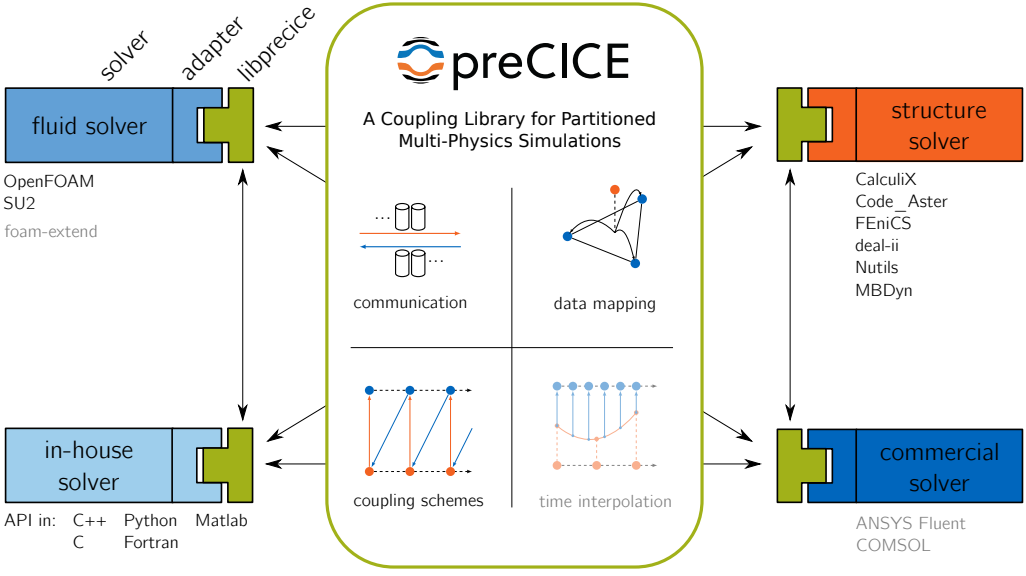
Gerasimos Chourdakis (TUM), Benjamin Uekermann (U Stuttgart)

Outline

1. Fundamentals
2. Live Demo
3. What is an Adapter?
4. Resources
5. Coupling Schemes

1. Fundamentals

The Big Picture



Who uses preCICE?

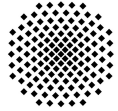
- We know of ≈80 groups that use preCICE
- 50% academia, 25% non-academic research centers, 25% industry
- Many different applications: aerodynamics, astronautics, automotive manufacturing, wind energy, biomechanics, marine engineering, nuclear fusion, reactor safety, ...
- 50% fluid-structure interaction

Who develops preCICE?

Still 100% academia

Funding

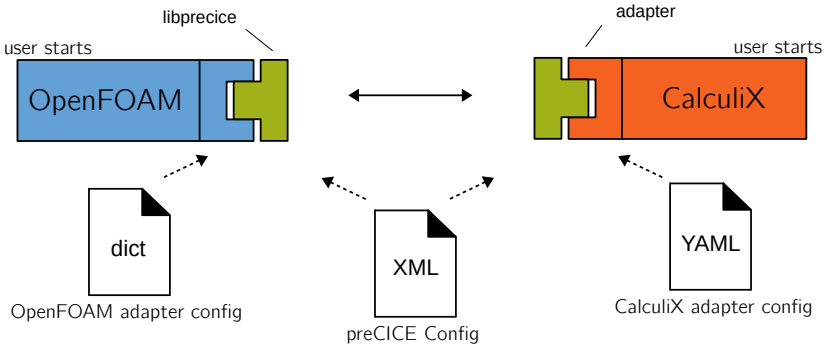
H2020 grant 754462



Universität
Stuttgart

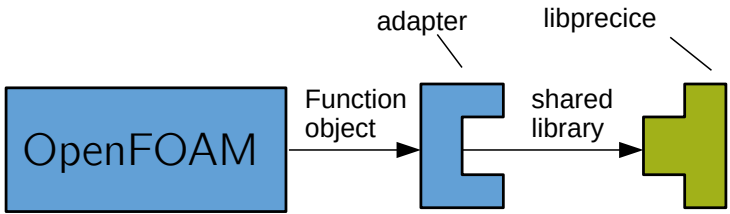
2. Live Demo

How does it look like?

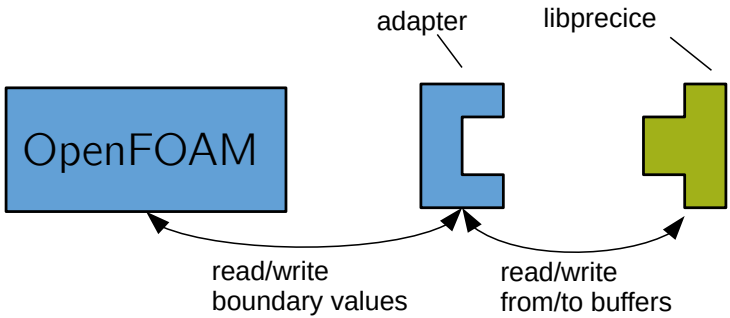


3. What is an Adapter?

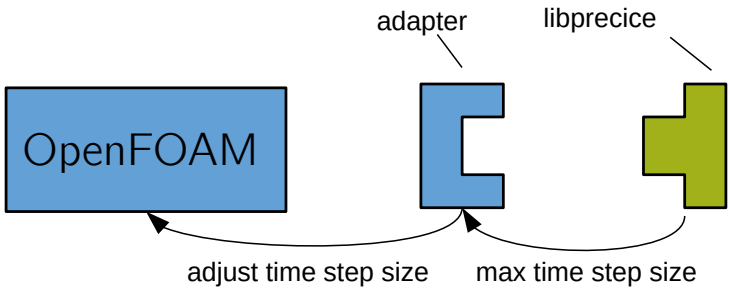
What is an adapter?



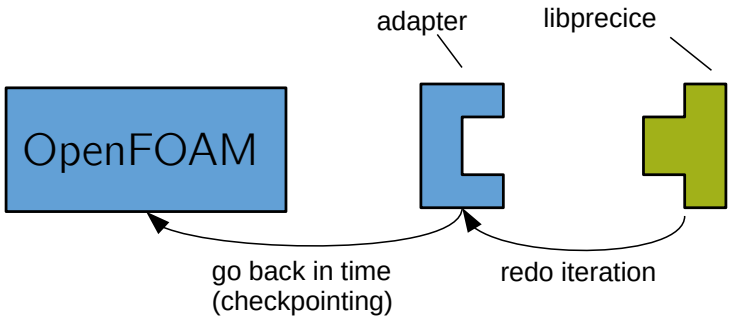
What is an adapter?



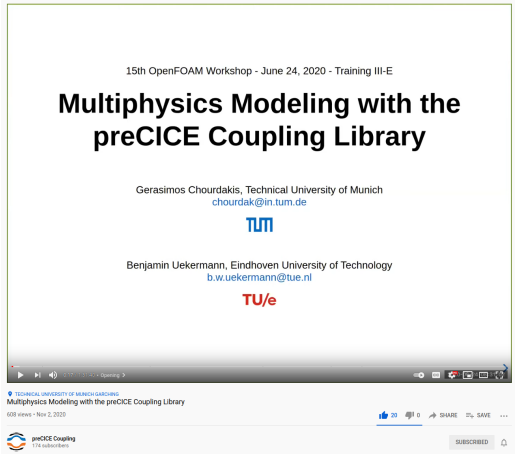
What is an adapter?



What is an adapter?



Learn more about preCICE and OpenFOAM



Watch the recording of our OpenFOAM Workshop 2020 training session on YouTube
(or join the next one at OFW16: 8-11 June, 2021)

4. Resources

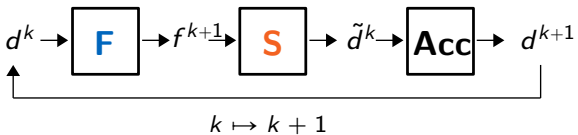
Resources

Live demo: preCICE website (`https://www.precice.org/`)
and forum (`https://precice.discourse.group/`)

5. Coupling Schemes

Serial-Implicit Coupling

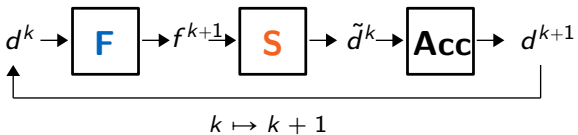
- Fluid solver $F : d \mapsto f$
- Solid solver $S : f \mapsto d$



[1] Mehl et al. Parallel Coupling Numerics for Partitioned Fluid-Structure Interaction Simulations. (2016)
[2] Bungartz et al. A Plug-and-Play Coupling Approach for Parallel Multi-Field Simulations. (2015)

Serial-Implicit Coupling

- Fluid solver $F : d \mapsto f$
- Solid solver $S : f \mapsto d$



- There is also explicit coupling
- There is also parallel coupling[1]
- Can be generalized to multiple coupled codes[2]

[1] Mehl et al. Parallel Coupling Numerics for Partitioned Fluid-Structure Interaction Simulations. (2016)
[2] Bungartz et al. A Plug-and-Play Coupling Approach for Parallel Multi-Field Simulations. (2015)

Aitken Acceleration

- Coupled problem: $F : d \mapsto f,\ S : f \mapsto d \quad \rightsquigarrow \quad (S \circ F)(d) \overset{!}{=} d$
- General fixed-point equation: $H(x) = x$

Aitken Acceleration

- Coupled problem: $F : d \mapsto f,\ S : f \mapsto d \quad \rightsquigarrow \quad (S \circ F)(d) \overset{!}{=} d$
- General fixed-point equation: $H(x) = x$
- Fixed-point iteration: $x^{k+1} = H(x^k)$

Aitken Acceleration

- Coupled problem: $F : d \mapsto f,\ S : f \mapsto d \quad \rightsquigarrow \quad (S \circ F)(d) \overset{!}{=} d$
- General fixed-point equation: $H(x) = x$
- Fixed-point iteration: $x^{k+1} = H(x^k)$
- Underrelaxation: $x^{k+1} = \omega H(x^k) + (1 - \omega)x^k\ ,\ \ \omega \in (0; 1)$
- Aitken underrelaxation: compute adaptive ω^k

Quasi-Newton Acceleration

- Non-linear problem: $H(x) = x \;\Leftrightarrow\; R(x) := H(x) - x \overset{!}{=} 0$

[3] Degroote et al. Performance of a New Partitioned Procedure versus a Monolithic Procedure in FSI. (2009)
[4] Lindner et al. A Comparison of Various Quasi-Newton Schemes for Partitioned FSI. (2015)
[5] Haelterman et al. Improving the Performance of the Partitioned QN-ILS Procedure for FSI Problems: Filtering. (2016)

Quasi-Newton Acceleration

- Non-linear problem: $H(x) = x \;\Leftrightarrow\; R(x) := H(x) - x \overset{!}{=} 0$
- Newton: $J_R \Delta x^k = -R(x^k)$ and $x^{k+1} = x^k + \Delta x^k$

[3] Degroote et al. Performance of a New Partitioned Procedure versus a Monolithic Procedure in FSI. (2009)
[4] Lindner et al. A Comparison of Various Quasi-Newton Schemes for Partitioned FSI. (2015)
[5] Haelterman et al. Improving the Performance of the Partitioned QN-ILS Procedure for FSI Problems: Filtering. (2016)

Quasi-Newton Acceleration

- Non-linear problem: $H(x) = x \;\Leftrightarrow\; R(x) := H(x) - x \overset{!}{=} 0$
- Newton: $J_R \Delta x^k = -R(x^k)$ and $x^{k+1} = x^k + \Delta x^k$
- Jacobian unknown $\rightsquigarrow$ approximate with past iterates

[3] Degroote et al. Performance of a New Partitioned Procedure versus a Monolithic Procedure in FSI. (2009)
[4] Lindner et al. A Comparison of Various Quasi-Newton Schemes for Partitioned FSI. (2015)
[5] Haelterman et al. Improving the Performance of the Partitioned QN-ILS Procedure for FSI Problems: Filtering. (2016)

Quasi-Newton Acceleration

- Non-linear problem: $H(x) = x \;\Leftrightarrow\; R(x) := H(x) - x \overset{!}{=} 0$
- Newton: $J_R \Delta x^k = -R(x^k)$ and $x^{k+1} = x^k + \Delta x^k$
- Jacobian unknown $\rightsquigarrow$ approximate with past iterates
- Different variants IQN-ILS[3] (explicit information from previous time windows) and IQN-IMVJ[4] (implicit information from previous time windows)

[3] Degroote et al. Performance of a New Partitioned Procedure versus a Monolithic Procedure in FSI. (2009)
[4] Lindner et al. A Comparison of Various Quasi-Newton Schemes for Partitioned FSI. (2015)
[5] Haelterman et al. Improving the Performance of the Partitioned QN-ILS Procedure for FSI Problems: Filtering. (2016)

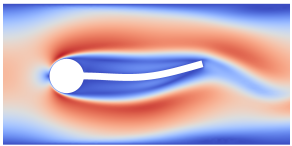
Quasi-Newton Acceleration

- Non-linear problem: $H(x) = x \;\Leftrightarrow\; R(x) := H(x) - x \overset{!}{=} 0$
- Newton: $J_R \Delta x^k = -R(x^k)$ and $x^{k+1} = x^k + \Delta x^k$
- Jacobian unknown $\rightsquigarrow$ approximate with past iterates
- Different variants IQN-ILS[3] (explicit information from previous time windows) and IQN-IMVJ[4] (implicit information from previous time windows)
- Can be improved with a filter[5]

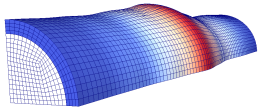
[3] Degroote et al. Performance of a New Partitioned Procedure versus a Monolithic Procedure in FSI. (2009)
[4] Lindner et al. A Comparison of Various Quasi-Newton Schemes for Partitioned FSI. (2015)
[5] Haelterman et al. Improving the Performance of the Partitioned QN-ILS Procedure for FSI Problems: Filtering. (2016)

Numerical Results

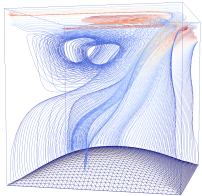
FSI3



3D-Tube



Driven Cavity



Mean Iterations[6]	Aitken	Quasi-Newton
FSI3	17.0	3.7
3D-Tube	Div.	7.5
Driven Cavity	7.4	3.0

[6] Uekermann. Partitioned FSI on Massively Parallel Systems. (2016)

Summary

- Minimally invasive integration (library approach, API in C++, C, Fortran, Python, Matlab)
- Many ready-to-use adapters available (OpenFOAM, FEniCS, Nutils, deal.II, SU2, ...)
- Coupling of arbitrary many programs
- Extensive documentation and testcases
- State-of-the-art numerical methods (quasi-Newton coupling, radial-basis function mapping)
- Scalability up to complete supercomputers
- Fast growing and active community

Links

- https://www.precice.org/
- https://github.com/precice/
- https://precice.discourse.group
- @preCICE_org
- https://www.youtube.com/c/preCICECoupling
- chourdak@in.tum.de, benjamin.uekermann@ipvs.uni-stuttgart.de