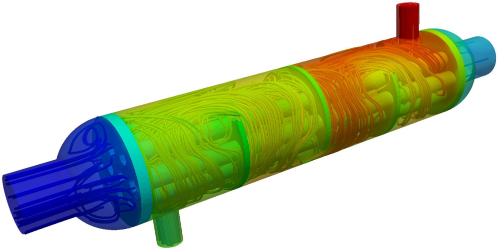
preCICE – A Library for Flexible Coupling of Multi-Physics Problems

Example: Shell-And-Tube Heat Exchanger

- Conjugate heat transfer between cold and hot fluid through complex structure

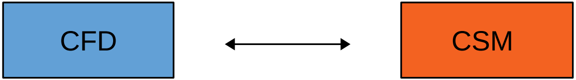


- Partitioned coupling: Usage of three independent solvers
- Reuse of existing solvers

OpenFOAM ⟷ CalculiX ⟷ OpenFOAM

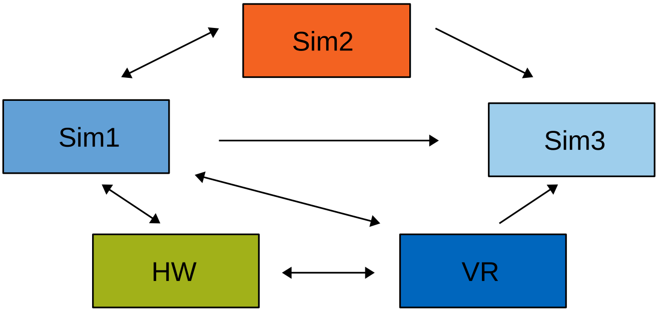
- Joint work with Lucia Cheung (TUM) and Babak Golami (SimScale)

The Present of Multi-Physics Simulations



- Two coupled components
- Weakly-coupled or uni-directional interaction
- Only moderately-parallel runs ($<$ 64 cores)
- Performance of coupling not considered important
  - Writing exchange data to files
  - Communication through central server process
  - Coupling numerics computed in serial
  - Inefficient coupling algorithms

The Future of Multi-Physics Simulations



- Many coupled components, coupling to hardware (HW) and virtual realities (VR), existing codes
- Various interactions (strong and weak ones)
- Massively parallel runs mandatory
- Performance of coupling highly important

Conclusion

Today's coupling approaches (coupling tools and monolithic simulation software) cannot deal with the future **flexibility**, **complexity**, and **efficiency** requirements.

Our Vision

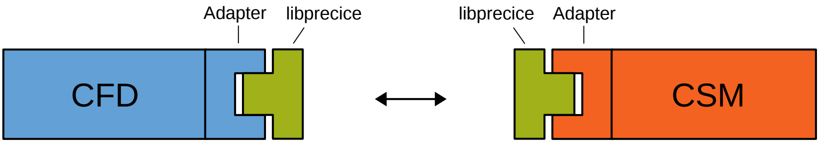
Provide solutions for the future of multi-physics simulations

Solution Strategy

- Black-box coupling $\Rightarrow$ Minimal information from coupled solvers
- Outsource complexity of coupling to “third” tool
- Pure library approach $\Rightarrow$ Minimally-invasive integration into coupled solvers
- Sophisticated coupling algorithms
- All coupling parts run efficiently on massively parallel systems
- Open-source strategy

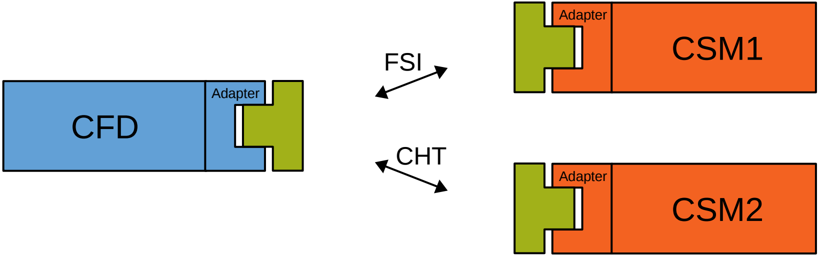
preCICE – A Plug-and-Play Coupling Library

As simple as this, . . .



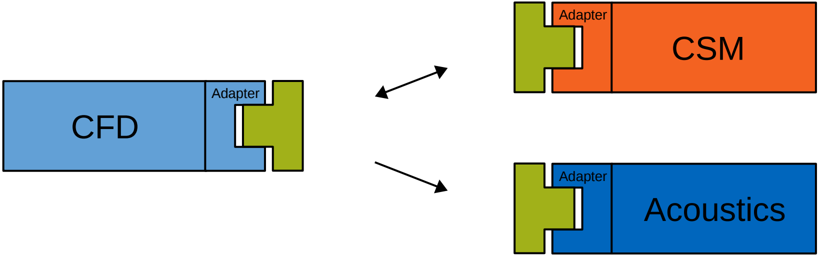
preCICE – A Plug-and-Play Coupling Library

. . . but also working for more components, . . .



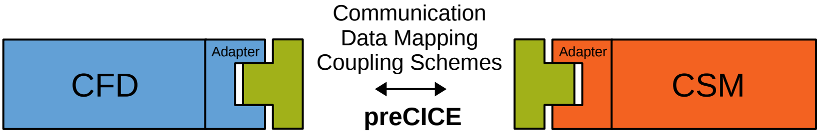
preCICE – A Plug-and-Play Coupling Library

. . . or different physical systems.



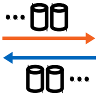
preCICE – A Plug-and-Play Coupling Library

To this end, we need three building blocks:



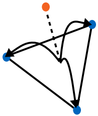
Features

Communication



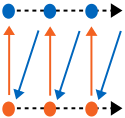
- MPI or TCP/IP (boost.asio)
- Asynchronous
- Fully parallel (no central instance)

Data Mapping



- Projection-based mapping (nearest neighbor or nearest projection)
- Radial-basis-function mapping (pure black-box)
- Consistent or conservative

Coupling Schemes



- Explicit or implicit
- Multi-coupling (more than two coupled solvers)
- Subcycling (different timestep sizes)
- Quasi-Newton methods

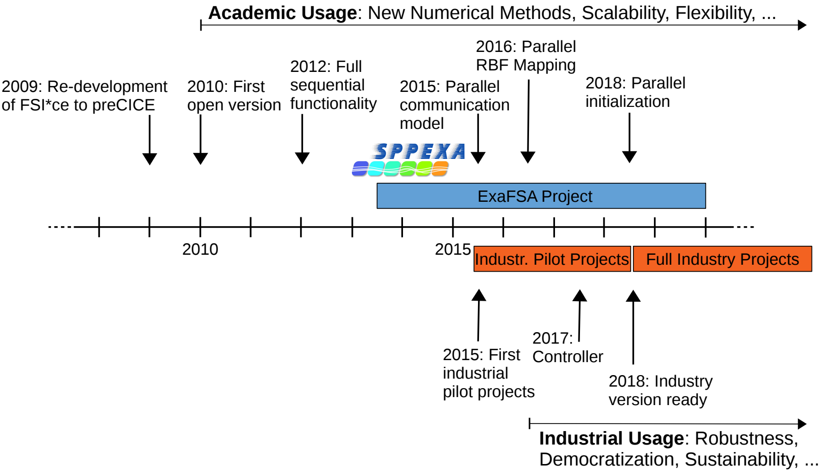
Coupled Simulation Software

ANSYS Fluent	CFD	commercial
Ateles	CFD	in-house (University of Siegen)
Alya System	CFD, CSM	in-house (Barcelona Supercomputing Center)
CalculiX	CSM	open-source
CARAT++	CSM	in-house (TU München)
Code_Aster	CSM	open-source
COMSOL	CFD, CSM	commercial
deal.II	CSM	open-source
FASTEST	CFD	in-house (TU Darmstadt)
FEAP	CSM	in-house (adapted at TU Darmstadt)
muSICS	CFD	in-house (A*STAR Singapore)
OpenFOAM	CFD	open-source
SU2	CFD	open-source

New Adapters in 30 Lines of Code

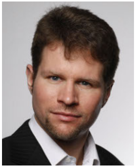
```
turn_on()
precice_create("NASTIN", "precice_config.xml", index, size)
init_unknowns_and_grid()
precice_set_vertices(meshID, N, pos(dim*N), vertIDs(N))
precice_initialize()
while time loop ≠ done do
   begin_timestep()
   while precice_not_converged() do
      solve_timestep()
      precice_write_bvdata(stresID, N, vertIDs, stres(dim*N))
      precice_advance()
      precice_read_bvdata(displID, N, vertIDs, displ(dim*N))
   end while
   end_timestep()
end while
turn_off()
precice_finalize()
```

10 Years of Development



The Team

Miriam Mehl
U Stuttgart



Florian Lindner
U Stuttgart

Amin Totounferoush
U Stuttgart

Alexander Rusch
ETH/COPLON

Hans Bungartz
TUM

Benjamin Rüth
TUM

Gerasimos Chourdakis
TUM

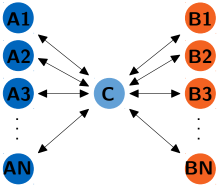
Benjamin Uekermann
TUM/COPLON

Previous contributors:

- Bernhard Gatzhammer, Klaudius Scheufele, Lucia Cheung, Alexander Shukaev, Peter Vollmer, Georg Abrams, . . .

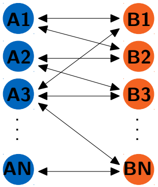
Example 1: Scalability Test

Server-Based Concept



- Complete communication through central server process
- Interface computations on server (in sequential)
- $\Rightarrow$ Coupling becomes bottleneck for overall simulation already on moderate parallel systems

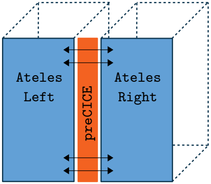
Our Peer-To-Peer Concept

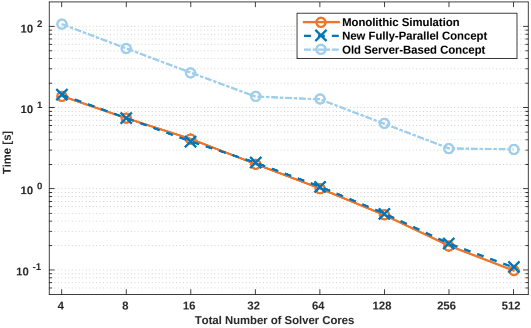


- No central entity
- $\Rightarrow$ Easier to handle (user does not need to care about server)
- $\Rightarrow$ No scaling issues

Example 1: Scalability Test

- Travelling density pulse (Euler equations) through artificial coupling interface
- DG solver Ateles (U Siegen), $7.1 \cdot 10^6$ dofs
- Nearest neighbor mapping and communication

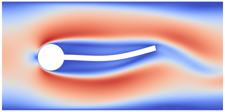




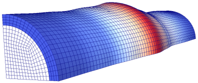
Example 2: Performance Tests of Quasi-Newton Coupling

Our quasi-Newton approach is more efficient and more robust than standard dynamic (Aitken) underrelaxation approaches.

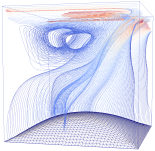
FSI3



3D-Tube



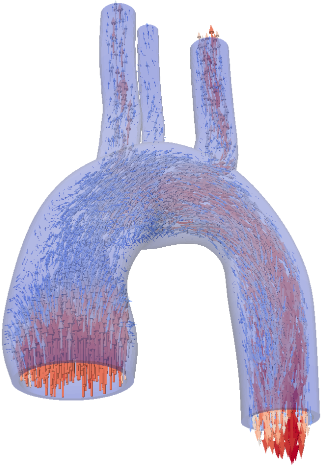
Driven Cavity



Mean Iterations	Aitken	Quasi-Newton
FSI3	17.0	3.3
3D-Tube	Div.	7.5
Driven Cavity	7.4	2.0

Example 2: Performance Tests of Quasi-Newton Coupling

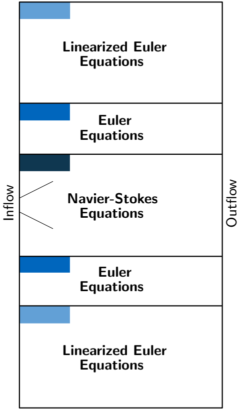
- Quasi-Newton can even handle biomedical applications, such as an Aortic bloodflow
- Stable coupling (no added-mass instabilities)
- Six times less iterations than Aitken

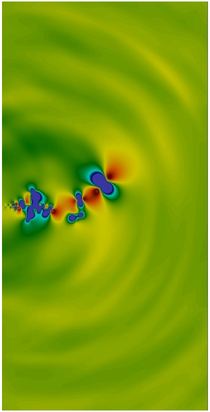


- Joint work with Juan-Carlos Cajas (Barcelona Supercomputing Center)
- Geometry by Jordi Martorell

Example 3: Multi-Fluid Coupling

- Besides FSI, many other possible applications of preCICE
- Simulation of a subsonic jet
- Explicit, parallel coupling between three fluid solvers
- Linearized Euler 28 times cheaper than Navier-Stokes

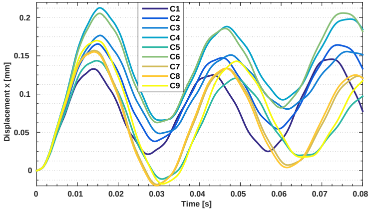




- Joint work with Verena Krupp (University of Siegen)

Example 4: Coupling of many Components

- Step-wise approximation of a brush seal by multiple cylinders
- Each cylinder simulated with individual structure solver $\Rightarrow$ reuse of setup
- Stable and efficient coupling of all 10 components (9 cylinders and fluid solver)



- Joint work with Alexander Fuchs (TUM LTF / MTU)

Contact

- Write to the preCICE mailing list or in our Gitter chatroom
- Open-source project: `www.precice.org`
- Source code: `github.com/precice`
- Bungartz, H.-J., Lindner, F., Gatzhammer, B., Mehl, M., Scheufele, K., Shukaev, A. and Uekermann, B. preCICE – A fully parallel library for multi-physics surface coupling. *Computers and Fluids*, Vol. **141**, pp 250–258, 2016.