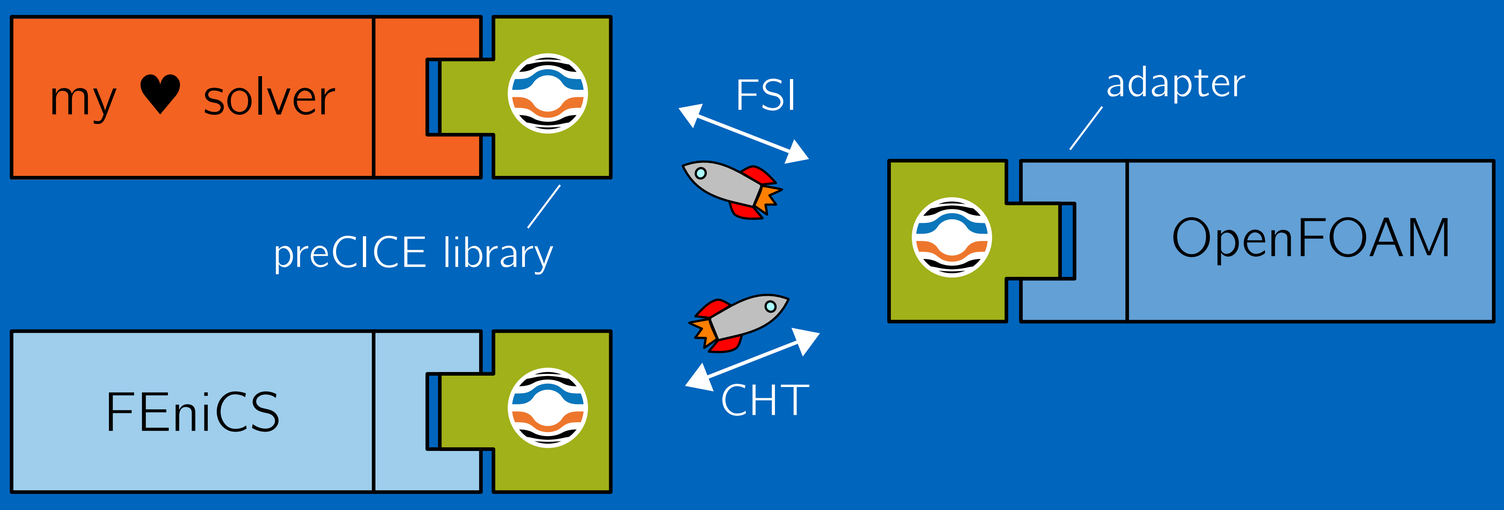
Coupling simulation codes

from accurately and efficiently,[1]

to **easily** and **sustainably**[2] is crucial and non-trivial.



Plain language summary

Single computer models are often not enough to describe phenomena and processes across science and engineering. The software package preCICE allows programmers and end users to combine existing models with minimal effort, producing accurate results and being portable from a laptop to a large supercomputer. With recent improvements and extensions in code and user experience, preCICE is now being sustainably developed and actively used by a vivid community spanning from aerospace and biomedical engineering to climate research.

preCICE

A sustainable and user-friendly coupling ecosystem for partitioned simulations

From a prototype to sustainable, easy-to-use software (with users!)

- Library approach: drop-in, no need to adopt design
- Packaging and dependencies: standard and minimal
- Ready-to-use integrations: users are not developers
- Documentation: next to the code, rendered as one
- Tests and continuous integration, code reviews
- Tutorial cases, forum, trainings, workshops

API example (simplified)

```python
import precice
vertices = precice.set_mesh_vertices(positions)
precice_dt = precice.initialize()
while precice.is_coupling_ongoing():
    solve_as_usual()
    precice.write_data("force", vertices)
    precice.advance()
    precice.read_data("displacement", vertices)
```

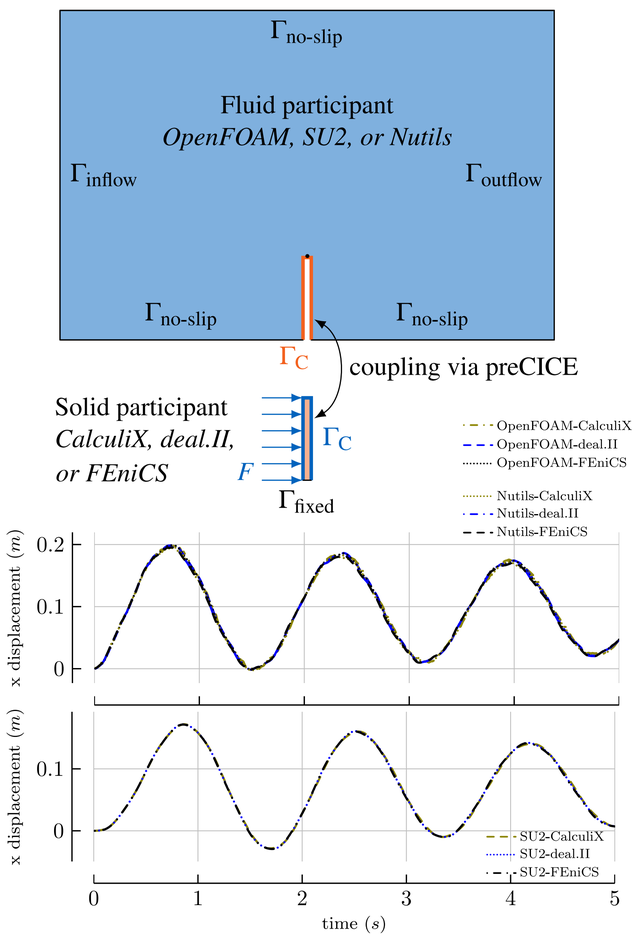
An ecosystem of components

- API in C++, C, Fortran, Python, Julia, MATLAB
- Ready-to-use adapters for CalculiX, code_aster, deal.II, DUNE, FEniCS, OpenFOAM, SU2, ...
- Tutorial cases for arbitrary solver combinations
- Automatically generated Vagrant box (VM) with all components preinstalled
- Website with documentation sourced from repos

Citable preCICE distribution on DaRUS [3]

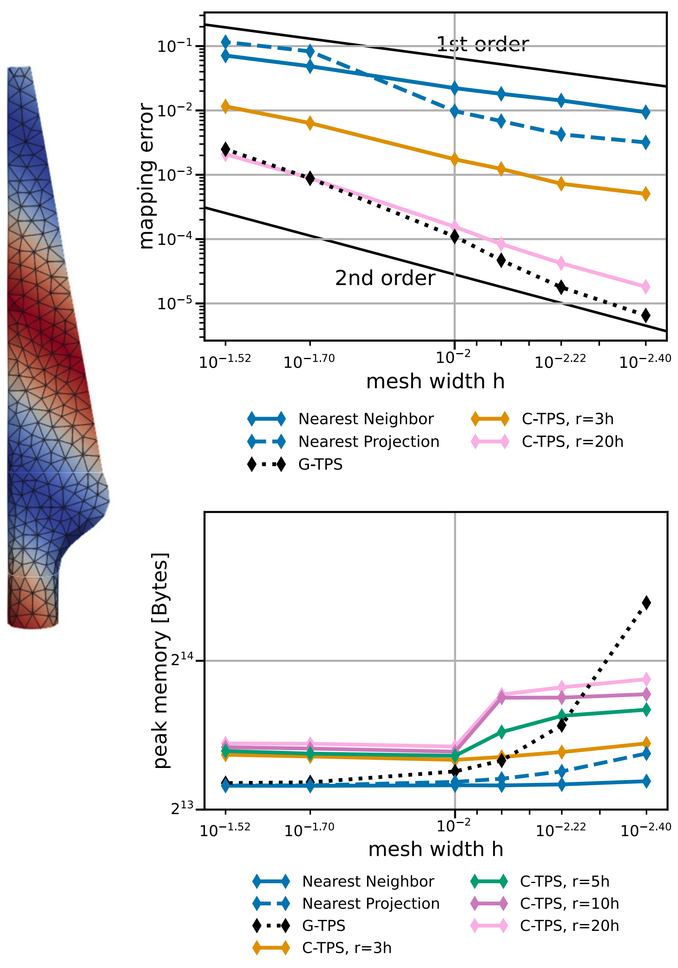
Example: Fluid-structure interaction

Choose your CFD (compressible / incompressible) and CSM codes at runtime: A preCICE tutorial.



Data mapping: Compact RBF with thin plate splines is a good option

Mapping comparisons evaluating a periodic function on coarse (4k vertices, shown) and fine (170k vert.) meshes of a wind turbine blade, using ASTE.



Towards testing everything together

CI for library and some components using GitHub Actions. Partitioned approach makes CI complex.

G. Chourdakis*, K. Davis, B. Rodenberg, M. Schulte, F. Simonis, B. Uekermann,

G. Abrams, H. J. Bungartz, L. Cheung Yau, I. Desai, K. Eder, R. Hertrich, F. Lindner, A. Rusch, D. Sashko, D. Schneider, A. Totounferoush, P. Vollmer, O. Ziya Koseomur

gerasimos.chourdakis@tum.de

Chair of Scientific Computing in Computer Science
School of Computation, Information and Technology
Technical University of Munich

Get paper and poster:
http://go.tum.de/369322

[1] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, B. Uekermann (2016). preCICE – A fully parallel library for multi-physics surface coupling. In Comput. & Fluids, 141, 250–258. https://doi.org/10.1016/j.compfluid.2016.04.003

[2] G. Chourdakis, K. Davis, B. Rodenberg, M. Schulte, F. Simonis, B. Uekermann, et al., preCICE v2: A sustainable and user-friendly coupling library [version 2; peer review: 2 approved] Open Res Europe 2022, 2:51, https://doi.org/10.12688/openreseurope.14445.2

[3] G. Chourdakis, K. Davis, I. Desai, B. Rodenberg, D. Schneider, F. Simonis, B. Uekermann, M. Firmbach, A. Jaust, C. Lorenz, B. Martin, M. Olesen, O. Ziya Koseomur (2022) preCICE Distribution Version v2202.0 https://doi.org/10.18419/darus-2613, DaRUS, V1.

University of Stuttgart
Germany

Technical University of Munich