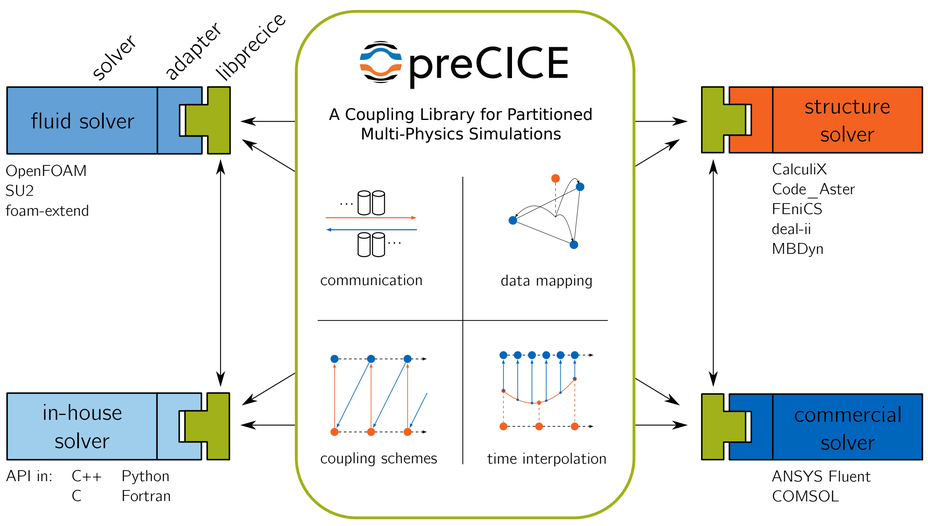
Energy Technology
Department of Mechanical Engineering
Eindhoven University of Technology

preCICE: A dependable open-source coupling library for partitioned multi-physics simulations

Hans-Joachim Bungartz[1], **Gerasimos Chourdakis**[1], **Florian Lindner**[2], **Miriam Mehl**[2], **Benjamin Rüth**[1], **Dmytro Sashko**[1], **Frédéric Simonis**[1], **Benjamin Uekermann**[3]
[1]{bungartz, simonis, chourdak, sashko}@in.tum.de, Chair of Scientific Computing, Technical University of Munich
[2]{Miriam.Mehl, Florian.Lindner}@ipvs.uni-stuttgart.de, Institute for Parallel and Distributed Systemes, University of Stuttgart
[3]b.w.uekermann@tue.nl, Department of Mechanical Engineering, Eindhoven University of Technology

preCICE Coupling Library[1]



preCICE (Precise Code Interaction Coupling Environment) is a coupling library for partitioned multi-physics simulations. Its minimally invasive API and scalability on massively parallel systems allow for rapid adaptation, and thus offers the flexibility needed to keep a decent time-to-solution for complex multi-physics scenarios. As a common interface, it encourages collaboration between researchers and ensures compatibility and thus the sustainability of both modern and legacy code.

Application Programming Interface

```
turnOnSolver(); //e.g. setup and partition mesh
SolverInterface precice("FluidSolver",rank,size);
precice.configure("precice-config.xml");

int meshID = precice.getMeshID("FluidMesh");
int vertexSize; // number of vertices at interface
// determine vertexSize
double* coords = new double[vertexSize*dim];
// determine coordinates
int* vertexIDs = new int[vertexSize];
precice.setMeshVertices(meshID, vertexSize,
                        coords, vertexIDs);
delete[] coords;

int displID = precice.getDataID("Displs", meshID);
int forceID = precice.getDataID("Forces", meshID);
double* forces = new double[vertexSize*dim];
double* displacements = new double[vertexSize*dim];

double dt; // solver timestep size
double maxDt; // maximum precice timestep size

maxDt = precice.initialize()

while (running() && precice.isCouplingOngoing()){
  if(precice.isActionRequired("WriteCheckpoint")){
    saveCheckpoint(); // save internal state of solver
  }

  precice.readBlockVectorData(displID, vertexSize,
                              vertexIDs, displacements);
  setDisplacements(displacements);
  dt = beginTimeStep(); // e.g. compute adaptive dt
  dt = min(maxDt, dt);
  computeSolution(dt);
  precice.writeBlockVectorData(forceID, vertexSize,
                               vertexIDs, forces);
  maxDt = precice.advance(dt); // coupling
  if(precice.isActionRequired("ReadCheckpoint")){
    reloadCheckpoint();
  }
  else{ // timestep converged
    endTimeStep(); // update variables, increment time
  }
}
precice.finalize();
delete[] vertexIDs, forces, displacements;
turnOffSolver();
```

A fluid solver adapted for fluid-structure interaction. Original calls marked in blue.

Developers

Gerasimos Chourdakis
Technical University of Munich
OpenFOAM, DevOps
Since 2017

Kyle Davis
University of Stuttgart
CalculiX
Since 2019

Florian Lindner
University of Stuttgart
RBF, MPI, Profiling
2014-2019

Alexander Rusch
ETH Zürich
SU2, CalculiX
2016-2018

Benjamin Rueth
Technical University of Munich
Time Integration, FEniCS
Since 2017

Dmytro Sashko
Technical University of Munich
DevOps, System Tests
2018-2019

Frédéric Simonis
Technical University of Munich
Meshes, Build System
Since 2018

Amin Totounferoush
University of Stuttgart
Parallel Initialization
Since 2017

Benjamin Uekermann
Eindhoven University of Tech.
HPC, Quasi-Newton
Since 2012

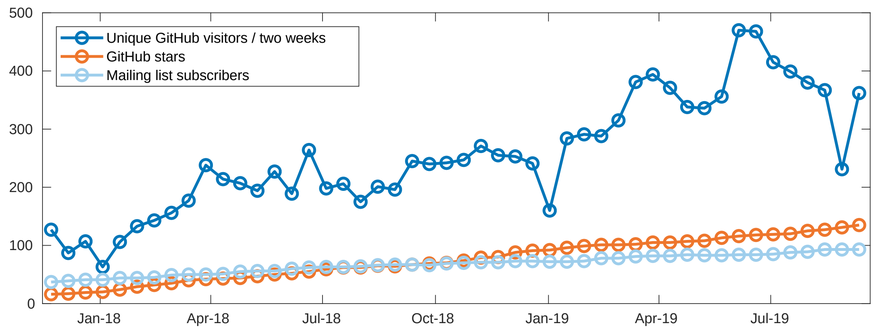
Previous main contributors: Bernhard Gatzhammer (alumnus of TUM), Klaudius Scheufele (alumnus of University of Stuttgart), and many more researchers and students!

Users

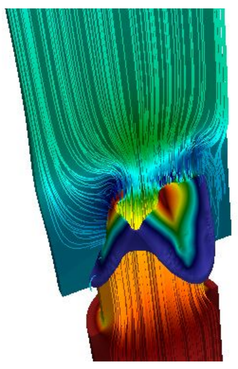
- LSM & STS, U Siegen, Germany
- SC & FNB, TU Darmstadt, Germany
- SCpA, CIRA, Italy
- Cardiothoracic Surgery, UFS, South Africa
- A*STAR, Singapore
- NRG, Petten, The Netherlands
- Aerodynamics & Wind Energy (KITE Power), TU Delft, The Netherlands
- Mechanical and Aeronautical Eng., University of Manchester, UK
- University of Strathclyde, Glasgow, UK
- FAST, KIT, Germany
- AIT, Ranshofen, Austria
- GRS, Garching, Germany
- MTU Aero Engines, Munich, Germany
- Temasek Laboratories, National University of Singapore
- Helicopter Technology & Astronautics, TUM, Germany
- IAG, University of Stuttgart, Germany
- CTTC UPC, Barcelona, Spain
- Amirkabir U. of Technology, Iran
- Noise & Vibration Research Group, KU Leuven, Belgium
- BITS Pilani, India

Upcoming:

- Numerical Analysis, Lund, Sweden
- ATA Engineering Inc., USA
- Aviation, MSU Denver, USA
- IMVT & IWS & MechBau, University of Stuttgart
- Engineering Science, U of Luxembourg
- Renewable and Sustainable Energy Systems & Hydrogeology, TUM, Germany
- Dive Solutions, Berlin, Germany

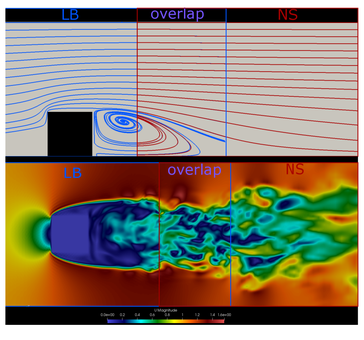


Evaluation of Heart Valve Biomechanics



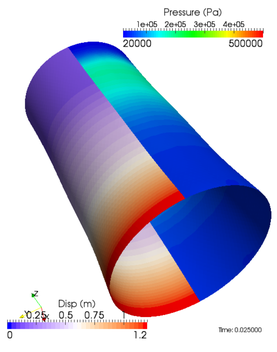
[2] University of the Free State, ZA

Hybrid Methods for Wind modelling in urban areas



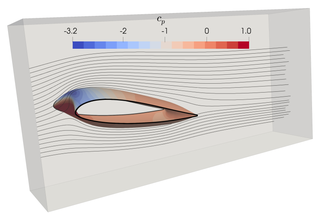
[3] The University of Manchester, GB

Simulation of High Impact Loads on Structures



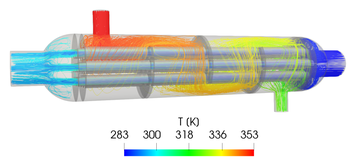
[4] A*STAR, SG

Fluid-Structure Interaction of Inflatable Wing Sections



[5] TU Delft, NL

Shell and Tube Heat Exchanger



[6,7,8] TU Munich/SimScale, DE

Reliabilty and Usability

Stable Foundation
Boost, nlohmann/json, libxml2, MPI, PETSc

Integration and Installation
CMake, pkg-config, pip, Debian, Spack

Code Quality
- Code coverage lcov, codecov.io
- Static analysis and modernisation clang-tidy

Documentation
Extensive GitHub Wiki, Doxygen

Community and Outreach

Communication
- Face-to-face
- Chat on Gitter
- Broadcast on mailing list
- Discourse forum
- GitHub Issues/Pull Requests

Culture
- Regular short telcos
- Regular structured meetings
- Local casual coding evenings
- Fixed release schedule
- Peer-reviews
- Open discussions

Community
- ECCOMAS minisymposia 2018, 2019, 2020
- **preCICE Workshop 17-18 Feb 2020, Munich**
- Cross-advertising via testimonials
- Valuable asset to teaching
- Contributions to adapters
- Common interface encourages collaboration

References:
[1] H.-J. Bungartz, B. Gatzhammer, F. Lindner, M. Mehl, K. Scheufele, A. Shukaev, and B. Uekermann. preCICE – A Fully Parallel Library for Multi-Physics Surface Coupling. *Computers and Fluids*, **141**, p. 250–258, 2016.
[2] K. Davis. Numerical and experimental investigation of the hemodynamics of an artificial heart valve. Thesis (MEng)–Stellenbosch University, 2018.
[3] M. Camps Santasmasas, A. Revell, B. Parslew, A. Harwood, and W. Crowther. Dual Navier-Stokes / Lattice-Boltzmann method for urban wind flow. *12th Int. ERCOFTAC Symp. on Eng. Turbulence Modelling and Measurements*, 2018.
[4] V.-T. Nguyen, B. Gatzhammer. A fluid structure interactions partitioned approach for simulations of explosive impacts on deformable structures. *Int. J. of Impact Eng.*, **80**, p. 65–75, 2015.
[5] M. Folkersma, R. Schmehl, A. Viré. Fluid-Structure Interaction Simulations on Kites. *Int. Airborne Wind Energy Conference*, 2017.
[6] L. Cheung Yau. Conjugate Heat Transfer with the Multiphysics Coupling Library preCICE. Master's thesis – Institut für Informatik, Technische Universität München, 2016.
[7] G. Chourdakis. A general OpenFOAM adapter for the coupling library preCICE. Master's thesis, Institut für Informatik, Technische Universität München, 2017.
[8] A. Rusch, B. Uekermann. Comparing OpenFOAM's Intrinsic Conjugate Heat Transfer Solver with preCICE-Coupled Simulations. *Unpublished white paper*, 2018.

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.

European Commission

EuroTechPostdoc Programme
Excellence in Science and Technology

www.precice.org github.com/precice