**Proceedings of the 7th GACM Colloquium on Computational Mechanics
for Young Scientists from Academia and Industry
October 11-13, 2017 in Stuttgart, Germany**

Official preCICE Adapters for Standard Open-Source Solvers

Benjamin Uekermann[1]*, Hans-Joachim Bungartz[1], Lucia Cheung Yau[1], Gerasimos Chourdakis[1] and Alexander Rusch[1]

Micro Abstract
To deal with the increasing complexity of today's multiphysics applications, the reuse of existing simulation software often becomes a necessity. Coupling to open-source simulation codes, in particular, is a time-efficient way to tackle new applications. The open-source coupling library preCICE enables such coupling in a minimally-invasive way. In this contribution, we give an overview on ready-to-use preCICE adapters for standard open-source solvers, namely CalculiX, Code_Aster, OpenFOAM, and SU2.

[1]Department of Informatics, Technical University of Munich, Garching, Germany
***Corresponding author**: uekerman@in.tum.de

Introduction

The preCICE library [1] allows to couple existing simulation software to overall multi-physics simulations in a flexible way at run-time. To this end, preCICE offers methods for equation coupling, such as quasi-Newton acceleration schemes, methods for data mapping, such as radial-basis function interpolation, and fully parallel communication means, based on either MPI or TCP/IP. preCICE is implemented in `C++`, but offers an API in most standard scientific computing languages. The software is actively developed at the Technical University of Munich and the University of Stuttgart and is published under the free LGPL license on GitHub (`http://www.precice.org`, `https://github.com/precice`). A recent article [8] summarizes the development of preCICE over the last ten years for a general audience.

To prepare an existing simulation code for coupling, preCICE has to be integrated with the solver. The high-level API of preCICE makes this step minimally-invasive. We refer to the glue code that handles the integration as "adapter", cf. Figure 1.

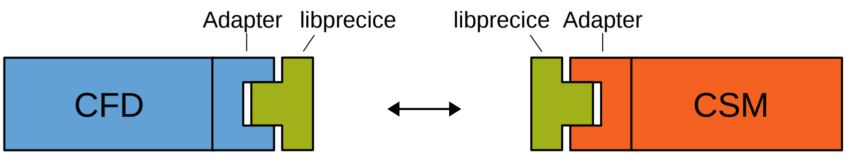

Figure 1. Coupling of a computational fluid dynamics code (CFD) to a computational structural mechanics code (CSM) via preCICE. We distinguish between the existing solver code, the adapter, which integrates preCICE in a minimally-invasive way, and the linked library.

Many users of preCICE face a similar situation: They have a sophisticated, often legacy in-house code, which they know very well and which is specialized for a certain application. To broaden the scope of applications, they want to couple their in-house code to an existing open-source code from the community. To write an adapter for the in-house code is often a relatively easy task. The users know where exactly to integrate the coupling and can copy the preCICE calls from tutorial examples. To write an adapter for the community code, however, is more cumbersome. Having good user knowledge of such a code is not enough. Developer knowledge is needed.

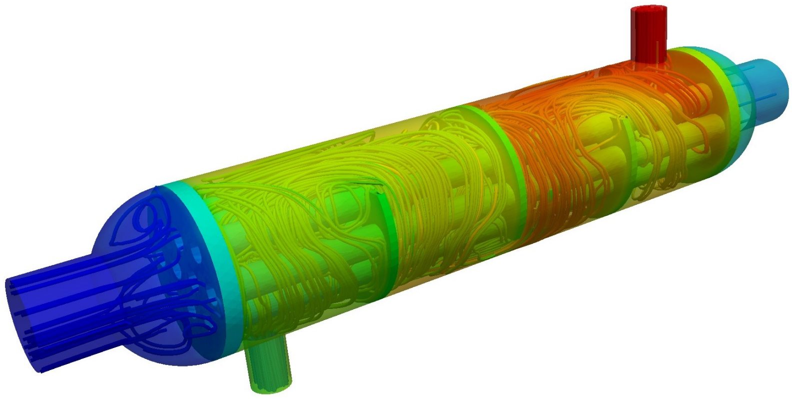

Figure 2. Temperature distribution and streamlines for a shell and tube heat exchanger. A cold fluid, running from left to right, and a hot fluid, running from top to bottom, exchange heat through a complex structure. Two instances of OpenFOAM are coupled to CalculiX via preCICE. The simulation has been carried out as part of the master thesis of Lucia Cheung [2], in cooperation with SimScale (`www.simscale.com`).

Furthermore, this is work that users do over and over again, as they all couple to the same community codes and rarely anybody is proud enough about his or her adapter to publish it. From preCICE perspective, this user experience is unsatisfactory.

Therefore, we decided to offer official preCICE adapters for several standard community solvers. Further benefits justify this effort: All Adapters can offer a unified user experience concerning building and running. Next, the coupled community codes can be used for tutorials, but also as complete coupled regression tests. Furthermore, the adapters should have fewer flaws as, during the adapter development, preCICE developers have a complete feature set of preCICE in mind whereas users often only apply a subset. Last, preCICE, of course, also benefits in terms of marketing as we can then directly approach the communities of the open-source simulation codes.

To conclude the introduction, Figure 2 shows a conjugate heat transfer simulation realized by coupling OpenFOAM to CalculiX.

1 Technical Realization of Adapters

The technically ideal realization of an adapter strongly depends on the individual simulation code. Still, open-source codes have in common that access to the complete functionality is possible. For commercial solvers (preCICE adapters currently available for ANSYS Fluent and COMSOL), access is often limited to specific function callbacks or interfaces. In this section, we give technical information about recently developed preCICE adapters for four standard community codes. Table 1 gives a first overview.

	Supp. Vers.	**Lic. of Adap.**	**Coupling Phys.**	**Solver Options**
CalculiX	2.10, 2.12	GPL	CHT, FSI	Linear and Non-Linear
Code_Aster	12	GPL	CHT	Linear and Non-Linear
OpenFOAM*	2.3–5.0	GPL	CHT, FSI	Arbitrary
SU2**	4.0.2–4.2.0	LGPL	FSI	Compressible Flow

Table 1. Official preCICE adapters for standard open-source codes. Tested version numbers of the codes are given, but upgrading to other versions should be rather simple. The adapter license is inherited from the simulation code's license. Coupling physics encompass (mechanical) fluid-structure interaction (FSI) and conjugate heat transfer (CHT). * The OpenFOAM adapter will be available by the end of 2017. ** Support of SU2 v5.0 will be available in fall 2017.

1.1 CalculiX

CalculiX [4] is a finite element solver for static, dynamic, and thermal structural mechanics problems, developed by Guido Dhont und Klaus Wittig (http://calculix.de/). The preCICE adapter builds on the master thesis of Lucia Cheung [2] and has been further extended by Alexander Rusch. As CalculiX is written in C, the adapter uses the C-API of preCICE. The adapter overwrites the main files (e.g. ccx_2.12.c and nonlingeo.c) of CalculiX and calls further adapter routines from there. The adapter is built from source with CalculiX in its native building procedure. Mechanical fluid-structure interaction (FSI) problems are supported as well as conjugate-heat transfer (CHT). For FSI, the user can choose between the linear and the nonlinear CalculiX solver. For CHT, steady-state as well as dynamic simulations are possible. During runtime, a YAML adapter configuration file specifies which variables are read and written at which coupling boundary. The choice of variables implicitly determines the type of coupling (FSI or CHT) at each coupling boundary. [2] gives more information on the adapter and validation of several coupled CHT scenarios. As a next development step, we want to extend the adapter to also support structure-structure interaction problems between, for example, the nonlinear and the linear solver of CalculiX.

1.2 Code_Aster

Code_Aster [6] is also a finite element structural mechanics code, developed by Electricité de France (http://www.code-aster.org). The software uses command files written in Python to call the actual solver routines written in Fortran. The coupled Code_Aster uses a specific command file adapter.comm and calls the actual preCICE adapter adapter.py from there using the Python-API of preCICE. The adapter was developed for CHT problems in [2], supporting dynamic and steady-state simulations. More information and validation is given in [2].

1.3 OpenFOAM

OpenFOAM [9] is a C++ library for continuum mechanics, which allows the implementation of arbitrary individual finite volume solvers. At the same time, many existing solvers are distributed as part of OpenFOAM, most of them for fluid dynamics. Many other solvers are developed in the community. In this fact lies one of the difficulties in developing a preCICE adapter for OpenFOAM – an ideal adapter should support any OpenFOAM solver. In a first step, we developed a CHT adapter for many important OpenFOAM solvers [2], by calling adapter methods directly from the main C++ solver routines. In [3], we now work on a generalization that compiles the adapter as a separate OpenFOAM function object – an additional library, which is called through callbacks directly in the OpenFOAM framework. Thus, no code changes are necessary to prepare an arbitrary existing OpenFOAM solver, only the OpenFOAM configuration file system/controlDict has to point to the compiled adapter. The user can further tailor the adapter at runtime by specifying the coupling variables and boundaries in a YAML configuration file. In theory, this enables the realization of any physical coupling. We focus our testing, however, on CHT and FSI scenarios. CHT validation is given in [2], whereas [3] describes the adapter in more detail. The publication of the adapter is planned for the end of 2017, starting with OpenFOAM versions v4.x and v5.x. Afterwards, we also plan to support earlier OpenFOAM versions starting with presumably v2.3. Another difficulty of the adapter development lies in the many available OpenFOAM variants and forks. The official preCICE adapters supports the www.openfoam.org distributions. David Blom, however, has also developed an unofficial preCICE adapter for foam-extend v3.2 (https://github.com/davidsblom/FOAM-FSI).

1.4 SU2

SU2 [5] stands for **S**tanford **U**niversity **U**nstructured and is a finite volume fluid solver (http://su2.stanford.edu/). The software originally evolved from aeronautical compressible flow applications, but due to an active community, SU2 nowadays also supports incompressible flow problems and has various multi-physics extensions. Alexander Rusch developed the preCICE adapter as part of his Bachelor thesis [7] and has maintained the adapter since. The adapter currently only supports FSI problems and only the compressible flow solver of SU2. As SU2 is written in `C++`, the adapter uses directly the `C++`-API of preCICE. Therefore, the main files of SU2, such as `SU2_CFD.cpp`, are modified to call the preCICE adapter, realized in a separate class. For every SU2 version, we provide an individual update of the main files. The adapter is built from source with SU2 in its native building procedure. For more information, refer to [7]. We currently support v4.0.2 to v4.2.0, but plan to support also the newest versions in fall 2017.

Conclusions

In this brief contribution, we gave an overview of official preCICE adapters for the popular open-source simulation codes CalculiX, Code_Aster, OpenFOAM, and SU2.

Future work will be threefold. First, we want to develop tutorial cases with the existing adapted open-source codes. A team of BGCE students (`http://www.bgce.de/`) currently works on interactive tutorials. A first static version with SU2 and CalculiX is already online (`https://github.com/precice/precice/wiki/FSI-tutorial`). Second, we want to extend the number of coupled open-source codes. Please approach us if you would like to see a particular code coupled or if you want to share your adapter. Last, we want to simplify the build process of preCICE and the adapters, for example by providing Debian packages or Docker containers.

Acknowledgements

We thankfully acknowledge the financial support of the priority program 1648 Software for Exascale Computing of the German Research Foundation (`http://www.sppexa.de/`) and our collaboration with SimScale (`https://www.simscale.com/`), which resulted in the preCICE adapters for CalculiX, Code_Aster, and OpenFOAM.

References

[1] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, and B. Uekermann. preCICE – a fully parallel library for multi-physics surface coupling. *Computers and Fluids*, 141:250–258, 2016.

[2] L. Cheung Yau. Conjugate heat transfer with the multiphysics coupling library preCICE. Master's thesis, Department of Informatics, Technical University of Munich, 2016.

[3] G. Chourdakis. A general OpenFOAM adapter for the coupling library preCICE. Master's thesis, Department of Informatics, Technical University of Munich, 2017. In preparation.

[4] G. Dhondt. *The finite element method for three-dimensional thermomechanical applications*. John Wiley & Sons, 2004.

[5] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.

[6] EDF. Finite element *code_aster*, analysis of structures and thermomechanics for studies and research. Open source on www.code-aster.org, 1989–2017.

[7] A. Rusch. Extending SU2 to fluid-structure interaction via preCICE. Bachelor's thesis, Munich School of Engineering, Technical University of Munich, 2016.

[8] B. Uekermann. The coupling library preCICE – past, present and future. *Quartl*, 80:11–20, 2016.

[9] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998.