Computers and Fluids 141 (2016) 250–258

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

preCICE – A fully parallel library for multi-physics surface coupling

Hans-Joachim Bungartz[a,1], Florian Lindner[b], Bernhard Gatzhammer[a,1], Miriam Mehl[b,*], Klaudius Scheufele[b], Alexander Shukaev[a,1], Benjamin Uekermann[a,1]

[a] Department of Computer Science, Technical University of Munich, Boltzmannstraße 3, Garching, Germany
[b] Institute for Parallel and Distributed Systems, University of Stuttgart, Universitätsstraße 38, Stuttgart, Germany

ARTICLE INFO

Article history:
Received 30 October 2015
Accepted 2 April 2016
Available online 19 April 2016

Keywords:
Partitioned multi-physics
Strong coupling
Non-matching grids
Inter-code communication
Quasi-Newton
Radial basis functions
High performance computing

ABSTRACT

In the emerging field of multi-physics simulations, we often face the challenge to establish new connections between physical fields, to add additional aspects to existing models, or to exchange a solver for one of the involved physical fields. If in such cases a fast prototyping of a coupled simulation environment is required, a partitioned setup using existing codes for each physical field is the optimal choice. As accurate models require also accurate numerics, multi-physics simulations typically use very high grid resolutions and, accordingly, are run on massively parallel computers. Here, we face the challenge to combine flexibility with parallel scalability and hardware efficiency. In this paper, we present the coupling tool preCICE which offers the complete coupling functionality required for a fast development of a multi-physics environment using existing, possibly black-box solvers. We hereby restrict ourselves to bidirectional surface coupling which is too expensive to be done via file communication, but in contrast to volume coupling still a candidate for distributed memory parallelism between the involved solvers. The paper gives an overview of the numerical functionalities implemented in preCICE as well as the user interfaces, i.e., the application programming interface and configuration options. Our numerical examples and the list of different open-source and commercial codes that have already been used with preCICE in coupled simulations show the high flexibility, the correctness, and the high performance and parallel scalability of coupled simulations with preCICE as the coupling unit.

1. Introduction

preCICE has been developed as a tool providing all functionality required for the realization of a plug-and-play multi-physics simulation environment based on a selection of existing and trusted single-physics solvers. That means, we target simulations where several types of equations are involved such as fluid-structure, fluid-acoustics, fluid-solid thermodynamics, porous-free flow, robot-soft tissue interactions and many more that cannot all be listed here. For all these examples, a high flexibility is often required. This holds in terms of fast prototyping using existing solvers and ready-to-use coupling methods, in terms of exchanging solvers by better, newer, or more suitable ones, and in terms of exchanging physical fields or adding further effects. In all these cases, a coupling tool that provides technical communication, data mapping between non-matching grids, and coupling iteration numerics in an easy-to-use way speeds up the development process. In addition, it enables non-coupling-experts to achieve stable solutions with minimal effort, and avoids multiple re-implementations of the same functionality. To be widely applicable, the coupling tool has to be suited also for black-box and parallel solvers.

The predecessor of preCICE, FSI*ce has been designed as a dedicated tool for the surface coupling in fluid-structure interaction (FSI) simulations. FSI is an ideal application example as it features severe stability issues, typically different types of meshes (Eulerian, Lagrangian, structured, unstructured, static, dynamic), different discretizations (finite elements/volumes/differences), and often also different scales in time and space. In contrast to FSI*ce, preCICE is not based on a client-server concept but implemented as a library providing direct communication between solver processes. In addition, preCICE is intended to be usable for virtually any surface coupled multi-physics simulation, not only for FSI simulations. Our focus is clearly on bidirectional surface coupling as communication is prohibitively expensive for volume coupled problems. However, parts of the preCICE functionality such as the iterative quasi-Newton solvers can also be applied for volume coupled problems

* Corresponding author. Tel.: +49 1573 4509626.
E-mail addresses: bungartz@in.tum.de (H.-J. Bungartz), florian.lindner@ipvs.uni-stuttgart.de (F. Lindner), miriam.mehl@ipvs.uni-stuttgart.de (M. Mehl), klaudius.scheufele@ipvs.uni-stuttgart.de (K. Scheufele), uekerman@in.tum.de (B. Uekermann).
[1] Supported by the Institute for Advanced Study, Technical University of Munich http://www.ias.tum.de

http://dx.doi.org/10.1016/j.compfluid.2016.04.003
0045-7930/

if communication of volume data across the network is avoided. Unidirectionally coupled problems are not in the focus of preCICE as they can be easily and efficiently tackled with much simpler tools, even via 'communication' over files.

The current version of preCICE relies solely on input and output data of the involved single-physics solvers to set up a multi-physics coupling in the sense of an strong/implicit coupling, i.e., the build-in preCICE iterators stop if and only if the coupling conditions are fulfilled. This recovers the solution of the monolithic (fully coupled) system of equations in a partitioned approach. For cases, in which such a strong coupling is not required, the user can explicitly configure preCICE to use explicit coupling. As the motivation to use a multi-physics model in general is to increase the accuracy compared to a simpler single-physics model, it is crucial for a simulation environment that the integration of parallel single-physics solvers is not only supported but also yields a good overall parallel scalability of the coupled simulation runs [21]. For this purpose, preCICE has recently undergone some major improvements in the communication with parallel solvers and in the parallel realization of the coupling functionality itself.

Since multi-physics simulations substantially gained importance throughout the last years, many tools to support the setup of a suitable software environment have been developed. We cannot give a complete list here, but rather focus on some that have similar aims as preCICE. In particular, we restrict to describe tools suitable for black-box solver coupling.

Among the first and for a long time the standard tool is the commercial software MpCCI [1] that was first developed as a communication library for inter-code communication of surface data, but in the meantime has been extended with data mapping functionalities for non-matching grids at the coupling surface. It uses a library approach for solver steering, but a framework approach for data exchange. Many ready-to-use adapters for commercial solvers are provided. The client-server approach limits the parallel scalability. Coupling numerics comprise only simple schemes such as Aitken underrelaxation.

ADVENTURE_Coupler is an open-source tool developed in Japan particularly to be used within the ADVanced ENgineering analysis Tool for Ultra large REal world (ADVENTURE) [20], an open-source general-purpose computational mechanics system for large-scale analysis and design. In contrast to preCICE, ADVENTURE_Coupler is based on a client-server approach. Communication via the server doubles the number of necessary communication operations compared to a direct point-to-point communication between solvers, but the coupling server is an MPI parallel executable itself such that there is no communication bottleneck at a single server process. Besides mapping and communication, the coupler also provides a Broyden-based quasi-Newton method to solve interface fixed-point equations.

EMPIRE (Enhanced Multi Physics Interface Research Engine) is another open-source code relying on the client-server approach. It has been developed based on the same ideas as the preCICE predecessor FSI*ce. It implements inter-code communication using MPI, sophisticated data mapping options for non-matching grids at the coupling surface, and quasi-Newton methods as well as Aitken underrelaxation for the iterative solution of the interface fixed-point equation. However, massively parallel simulations are not possible with EMPIRE.

OpenPALM [24] is a very advanced and well-established open-source software for massively parallel HPC applications combining multiple executables that require extensive data communication during their execution. For interface numerics, OpenPALM offers interfaces to several linear algebra libraries such as BLAS and LAPACK that can be used to establish user-defined coupling numerics. However, ready-to-use coupling iterations including the respective solvers are, to our best knowledge, not provided.

The OASIS coupler [29] based on the Model Coupling Toolkit (MCT) [22] is an open-source code coupler for massively parallel climate simulation. It offers fully parallel coupling field interpolation and intercommunication, but no dedicated coupling iteration methods.

The Physics Integration KErnerls (PIKE) are part of the Trilinos library and provide code coupling drivers for sophisticated fixed-point iteration schemes. The technical communication is implemented in the Data Transfer Kit (DTK) [28] providing efficient point-to-point communication pattern. Users can choose from various data mapping options for volume and surface coupling of non-matching grids.

Summarizing the main features of preCICE compared to the mentioned tools, we can state that preCICE is open-source and not developed for use with dedicated solvers. It provides all components necessary for multi-physics coupling, i.e., technical inter-code communication via MPI or TCP/IP, various methods for data mapping between non-matching grids, and efficient and robust coupling iterations based on quasi-Newton methods applied to interface fixed-point equations. preCICE can be used with black-box commercial solvers as well as of course also with open-source and inhouse solvers. To support parallel solvers, preCICE offers efficient point-to-point communication without having to go through a server instance. In contrast to all mentioned coupling tools, preCICE features a high-level API, which makes integration into existing code minimally invasive and allows for a high flexibility at runtime.

In the remainder of this paper, we give an overview of the functionalities offered by preCICE and its application user interface. In the last sections, we present performance data for example applications on SuperMUC at the Leibniz Supercomputing Center in Garching.

2. Functionality

As already mentioned, preCICE provides implementations for three main ingredients of a partitioned multi-physics simulation: 1) iterative methods for solving an interface fixed-point equation, 2) data mapping for interpolating data between non-matching grids at the coupling surface, and 3) implementations of the physical data communication between processes of several solver codes. All functionalities are provided in a single library API. Fig. 1 shows this concept in a schematic way.

2.1. Iterative solvers for interface fixed-point equations

Depending on the strength of the physical coupling between the involved single-physics fields in a multi-physics simulation, a time step of the overall partitioned simulation can either be done with only a small and fixed number of calls of time steps in each involved solver or requires an iterative procedure to achieve convergence to the monolithic solution of the coupled system in each time step. Acccordingly, preCICE offers both possibilities: explicit coupling schemes come with a fixed number of solver calls per time step, implicit coupling schemes iterate over a coupling equation until convergence. In the following, we assume that we have two involved solvers. Their routines for a single time step induce mappings S_1 and S_2 which map elements of vectorspaces X_1 and X_2 describing data at the coupling surface to output values at the coupling surface. Hereby, the output of S_1 is of the type that is required as an input by S_2 and vice versa, i.e., we have

$$S_1 : X_1 \to X_2 \quad \text{and} \quad S_2 : X_2 \to X_1. \tag{1}$$

Many partitioned coupling formulations are based on such a setting, among them all Dirichlet–Neumann type couplings. As an example, fluids and structures are usually coupled this way with