The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-–9 June 2022, Oslo, Norway

A FLEXIBLE SOFTWARE APPROACH TO SIMULATE TWO-SCALE COUPLED PROBLEMS

ISHAAN DESAI[1], CARINA BRINGEDAL[2] AND BENJAMIN UEKERMANN[1]

[1] Institute for Parallel and Distributed Systems (IPVS)
University of Stuttgart
Universitätstraße 38, 70569, Stuttgart
e-mail: {ishaan.desai,benjamin.uekermann}@ipvs.uni-stuttgart.de

[2] Institute for Modelling Hydraulic and Environmental Systems (IWS)
University of Stuttgart
Pfaffenwaldring 61, 70569, Stuttgart
email: carina.bringedal@iws.uni-stuttgart.de

Key words: multiphysics, multiscale, research software, preCICE, porous media

Abstract. Many multiscale simulation problems require a many-to-one coupling between different scales. For such coupled problems, researchers oftentimes focus on the coupling methodology, but largely ignore software engineering and high-performance computing aspects. This can lead to inefficient use of hardware resources, on the one hand, but also inefficient use of human resources as solutions to typical technical coupling problems are constantly reinvented.

This work proposes a flexible and application-agnostic software framework to couple independent simulation codes in a many-to-one fashion. To this end, we introduce a prototype of a new lightweight software component called Micro Manager, which allows us to reuse the coupling library preCICE for two-scale coupled problems. We demonstrate the applicability of the framework by a two-scale coupled heat conduction problem.

1 INTRODUCTION

For many multiscale scenarios in simulation technology, the physics on different scales can be viewed as separate problems that interact with each other. The simulation software for such scenarios involves solving the physics on the individual scales and devising some form of coupling between the scales. This work presents a software framework to couple simulations at two scales in an application-agnostic way. Avoiding reinventing coupling strategies for each application is the main motivation of developing such a framework. Two-scale simulations encompass a broad range of techniques and approaches, and this work targets a specific class of two-scale simulations.

Most two-scale phenomena have a well-defined scale separation. The two scales are commonly referred to as macro and micro scales. A two-scale multiphysics problem can be seen as a macro-micro coupled problem. There are several application areas where such macro-micro coupled simulations have already been done, for example, porous media [1, 2], dual-phase steel simulation [3], computational mechanics [4] and biomechanics [5]. In each of these applications, the coupling methodologies are mostly developed from scratch. These methodologies typically involve efficient data transfer and communication between the scales, different coupling schemes, and technical solutions for how to combine different programming languages. This work builds on the functionality of the coupling library preCICE [6] to develop a software framework that can facilitate application-agnostic macro-micro coupling.

The development of a flexible macro-micro coupling software has been previously discussed from different perspectives. Groen et al. (2019) [7] states that the field of generic multiscale coupling software is still maturing. Even though software packages such as MUSCLE3 [8], MUI [9], or Amuse [10] exist, they are all tailored to particular multiscale computing patterns. The macro-micro coupling we are addressing in this work falls into the heterogeneous multiscale computing pattern [11]. For this pattern, high-performance computing (HPC) software is still rare [11]. There exist, however, already application-tailored solutions for massively parallel simulations, for example Klawonn et al. (2020) [3] or Di Natale et al. (2019) [12].

The proposed software framework has two parts: the preCICE coupling library and a newly developed so-called Micro Manager[1]. Partitioned black-box coupling between two or more simulation codes can already be done with preCICE. The Micro Manager controls a set of micro-scale simulations and couples them to a macro-scale simulation using preCICE. This way, we can reuse the sophisticated and efficient coupling solutions of preCICE.

The macro-micro coupling framework is explained in a step-by-step manner. Section 2 introduces the software components used in this work. Section 3 describes a two-scale heat conduction example problem, which is developed to showcase and test the applicability of the software framework. The macro-scale problem solves the heat equation using the conductivity and material amounts from micro-scale problems connected to each point of the macro problem. Section 4 shows results of the simulation runs of the model problem.

2 SOFTWARE

There are two software components in this work, namely preCICE and a newly developed software package called the Micro Manager. The coupling between one macro simulation and several micro simulations using preCICE and the Micro Manager is shown in Figure 1. This section presents both the used and the developed software packages.

2.1 preCICE coupling library

preCICE facilitates coupling between two or more simulation codes to perform multiphysics simulations. The coupling is done in a minimally-invasive black-box fashion. Through its application programming interface (API), preCICE steers the coupled codes and handles the coupling numerics. preCICE does fully distributed point-to-point communication and offers functionality such as data mapping methods and coupling schemes [6]. Typical examples of coupled scenarios are fluid-structure interaction and conjugate heat transfer, but preCICE is not restricted to these applications. preCICE is an open-source project[2] with extensive documentation[3] and tutorial

[1] `https://github.com/precice/micro-manager`

[2] `https://github.com/precice`

[3] `https://precice.org/docs.html`

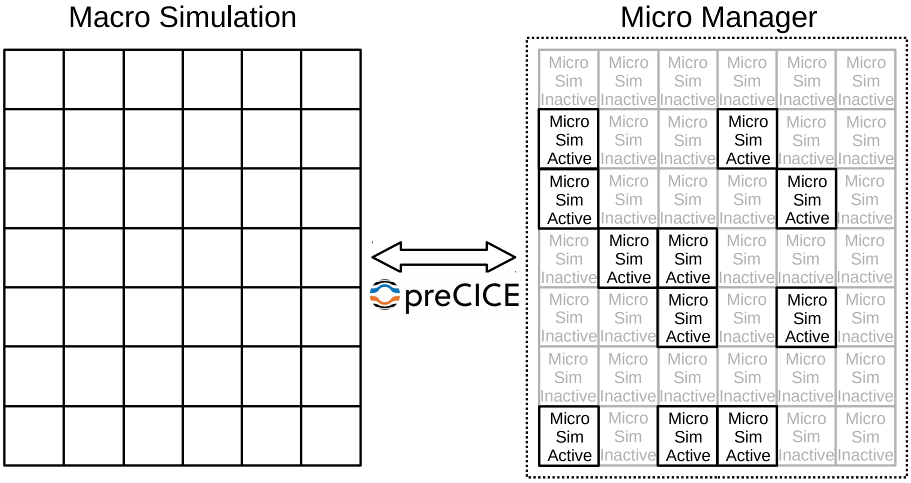

Figure 1: A macro simulation is coupled to the Micro Manager which in turn controls a set of micro simulations. The coupling is done via preCICE. The *active* and *inactive* micro simulations indicate that the Micro Manager can run them adaptively.

cases[4].

2.2 Micro Manager

The Micro Manager controls all micro simulations and couples them to a macro simulation through preCICE. To this end, the micro code needs to be converted to a callable library with a specific API, shown in Listing 1 as it would be for a micro problem code for the model problem explained in section 3. The functions `solve(...)`, `initialize(...)`, `save_checkpoint()` and `reload_checkpoint()` are part of the Micro Manager API, which means that the functions need to have those particular names and signatures so that the Micro Manager can call them, give them data and use the returned data. More details about the API can be found in the Micro Manager README [5].

The Micro Manager itself is coupling to the macro simulation using preCICE. The data transfer between the Micro Manager and each individual micro simulation is done in-memory. We present a first version of the Micro Manager, in which all micro simulations are run in each time step. The Micro Manager is configured with a JSON (JavaScript Object Notation) file. A configuration is shown in Listing 2.

The configuration file has two components, `coupling_params` and `simulation_params`. In `coupling_params`, `config_file_name` is the name of the preCICE XML configuration file, `macro_mesh_name` is the name of the macro mesh as stated in the preCICE configuration file, and, `read_data_names` and `write_data_names` are dictionaries with the names of the data as keys and strings *scalar* or *vector* as values, depending on whether the data is scalar or vector. In `simulation_params`, the user needs to set the entity `macro_domain_bounds` which are the minimum and maximum limits of the macro domain in all axis. For a 2D simula-

[4] https://github.com/precice/tutorials

[5] https://github.com/precice/micro-manager#readme

```python
class MicroSimulation:
    def __init__(self):
        self._dims = 2
        self._porosity = None
        self._conductivity = None
        self._checkpoint = None

    def initialize(self):
        self._porosity = 0
        self._conductivity = []

    def solve(self, temperature, dt):
        self._porosity = solve_allen_cahn(temperature, dt)
        self._conductivity = solve_heat_eqn(self._porosity)

        return {"porosity": self._porosity, "conductivity_i": self._conductivity[0],
        ↪  "conductivity_j": self._conductivity[1]}

    def save_checkpoint(self):
        print("Saving state of micro problem")
        self._checkpoint = self._porosity

    def reload_checkpoint(self):
        print("Reverting to old state of micro problem")
        self._porosity = self._checkpoint
```

Listing 1: Example of micro simulation code as a callable Python class

```json
{
  "micro_file_name": "micro_heat",
  "coupling_params": {
    "config_file_name": "precice-config.xml",
    "macro_mesh_name": "macro-mesh",
    "read_data_names": {"temperature": "scalar"},
    "write_data_names": {"porosity": "scalar", "conductivity_i": "vector",
    ↪  "conductivity_j": "vector"}
  },
  "simulation_params": {
    "macro_domain_bounds": [0.0, 1.0, 0.0, 1.0, 0.0, 1.0]
  }
}
```

Listing 2: JSON configuration file of Micro Manager

tion the format is `[xmin, xmax, ymin, ymax]` and correspondingly for a 3D simulation it is `[xmin, xmax, ymin, ymax, zmin, zmax]`. Using these bounds, the Micro Manager extracts the coordinates of the points on the macro domain from preCICE.

Future versions of the Micro Manager will be capable of adaptively running micro simulations. Adaptive means classifying the micro simulations as active and inactive, running the active ones and using their output to generate the full output field of all micro simulations. Various strategies for selecting the adaptivity criteria have been proposed [13, 14] and will be implemented in the Micro Manager. The Micro Manager is already parallelized with the Message Passing Interface (MPI) for Python [15], and it is foreseen that the manager will have dynamic load-balancing strategies when adaptivity is used for parallelized scenarios.

3 MODEL PROBLEM

The new macro-micro software framework is tested using a two-scale heat conduction example, motivated by Bastidas et al. (2021) [13]. We consider a macro domain with an underlying micro structure of two materials. The material properties are known, but the micro structure evolves in an a priori unknown way. The existence of such an evolving micro structure means that the constitutive properties necessary on the macro scale cannot be directly estimated or taken from literature. At each computation point of the macro problem, a micro problem is solved and the constituent quantities are upscaled. Such a macro-micro problem setting is possible for a system with clear scale separation. That means if the micro length is l and the macro length scale is L, we assume that $l/L \ll 1$. The macro domain is denoted by $\mathbf{X}$ and the micro domain is denoted by $\mathbf{Y}$.

3.1 Problem on macro scale

On the macro scale, the heat equation is solved on a two dimensional rectangular domain, see Figure 2. Let $\mathbf{K}$ be the potentially anisotropic thermal conductivity tensor, ρ the molar density and c the specific heat. We know a priori that a micro structure with two materials g (grain) and s (solid) exists, and hence the macro problem needs to be suitably modified to incorporate this. We need to use the density and specific heat of both the materials. We define a ratio $\Phi(x)$ which is the relative amount of material s to the total material in a micro domain, which we henceforth call porosity. We use Φ to write the heat equation as

$$\partial_t(\Phi \rho_s c_s u + (1-\Phi)\rho_g c_g u) = \nabla \cdot (\mathbf{K} \nabla u) \quad \text{in} \quad \mathbf{X}, \tag{1}$$

where u is the temperature to be solved for. The quantities $\mathbf{K}$ and Φ will be acquired by solving micro problems.

3.2 Problem on micro scale

We assume that a micro structure exists in the macro domain, as shown in Figure 2. We choose the domain of the micro simulation as a unit square as it is only a representative simulation and not an actual zoomed-in domain of the macro scale. The micro domain is made up of a material having conductivity k_g which is embedded in another material having conductivity k_s. For simplicity reasons, the material g is chosen to have a circular shape. The micro simulation has periodic boundary conditions on all boundaries. The grain geometry is mathematically represented by introducing a phase-field variable ϕ which takes the value 0 in the grain domain and the value of 1 outside it. Using this phase-field variable, the heat equation in the micro domain

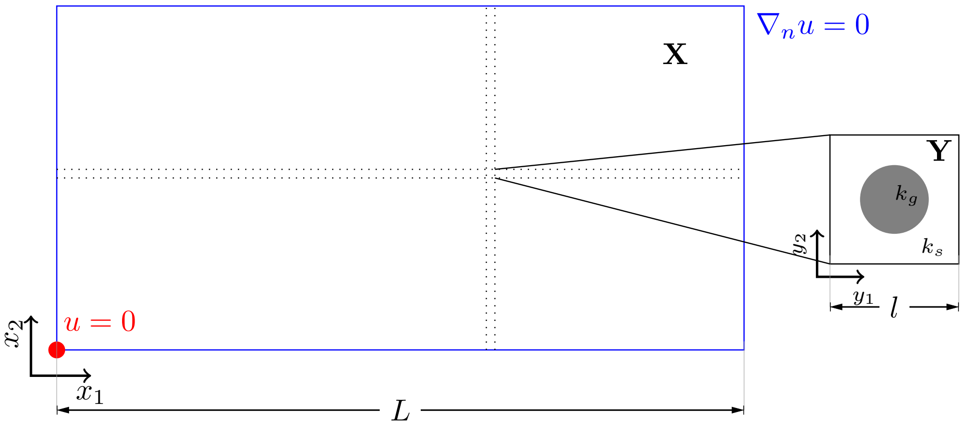

Figure 2: Macro domain **X** with boundary conditions and a representative micro domain **Y** with a heterogeneous grain structure.

can be written as

$$\partial_t(\rho_s c_s \phi u + \rho_g c_g (1-\phi)u) = \nabla \cdot ((k_s\phi + k_g(1-\phi))\nabla u) \quad \text{in} \quad \mathbf{Y} = [0,1]^2. \tag{2}$$

This equation simplifies to the equation of the corresponding material depending on whether the value of ϕ is closer to 0 or 1. The phase-field approach is used to model the heat conduction through the two material domains. The macro problem is already quasi-non-dimensional, and more details on the formulation of the equations can be found in Bringedal et al. (2020) [1]. We solve the following equation

$$\nabla \cdot ((k_g(1-\phi) + k_s\phi)(e_j + \nabla\psi^j)) = 0, \tag{3}$$

Eq. (3) is known as the cell problem for ψ^j and it appears from two-scale homogenization of Eq. (2). For more details on homogenization of such problems, we refer to Bringedal et al. (2020) [1]. To guarantee uniqueness, we constrain the weights in the following way

$$\int_{\mathbf{Y}} \psi^j dy = 0. \tag{4}$$

The components of the effective upscaled conductivity matrix $\mathbf{K}$ are calculated from the weights as

$$\mathbf{K}_{ij} = \int_{\mathbf{Y}} (k_g(1-\phi) + k_s\phi)(\delta_{ij} + \partial_{y_i}\psi^j)dy. \tag{5}$$

The initial micro structure is set using an analytical representation of the phase field ϕ

$$\phi(y_1, y_2) = \frac{1}{1 + \exp(\frac{-4}{\lambda}\sqrt{(y_1 - y_{0,1})^2 + (y_2 - y_{0,2})^2 - r_0^2})}, \tag{6}$$

where (y_1, y_2) are the micro coordinates, $(y_{0,1}, y_{0,2})$ is the center of the grain, r_0 is the initial radius of the grain and $\lambda > 0$ is related to the width of the transition layer between the two materials. To calculate the evolution of the phase field, we use the Allen-Cahn equation with a reaction rate f related to the macro-scale temperature in the following way

$$\lambda^2 \partial_t \phi + \gamma P'(\phi) = \gamma \lambda^2 \nabla^2 \phi - 4\lambda\phi(1-\phi)f(u), \tag{7}$$

where $P(\phi) = 8\phi^2(1-\phi)^2$ is the double-well potential, γ is the diffusion coefficient and $f(u)$ is a reaction rate, which is formulated as

$$f(u) = k_u \left(\frac{u^2}{u_{eq}^2} - 1 \right), \tag{8}$$

where k_u is a constant deciding the speed of the grain expansion or contraction and u_{eq} is a chosen equilibrium temperature. This reaction rate term is artificially constructed to couple micro problem to a macro entity, which in this case is the macro temperature u. The porosity Φ is calculated from the phase field in the following way

$$\Phi = \int_{\mathbf{Y}} \phi dy. \tag{9}$$

3.3 Solving the macro-micro problem

Both the macro and micro problem are solved using the finite element library Nutils [16]. The macro and micro codes are publicly available[6]. Initially the macro field u is set at 0.5. Boundary conditions for the macro domain are as shown in Figure 2. There is a point Dirichlet boundary condition at the lower left corner, and otherwise the macro domain has zero flux boundary conditions. All micro simulations have an initial grain of material g of radius $r_0 = 0.25$. The material conductivity values are chosen as $k_g = 0$ and $k_s = 1.0$. For simplicity, the ρ and c on both scales are chosen as 1.0. On the micro scale $\gamma = 0.01$ and $u_{eq} = 0.5$.

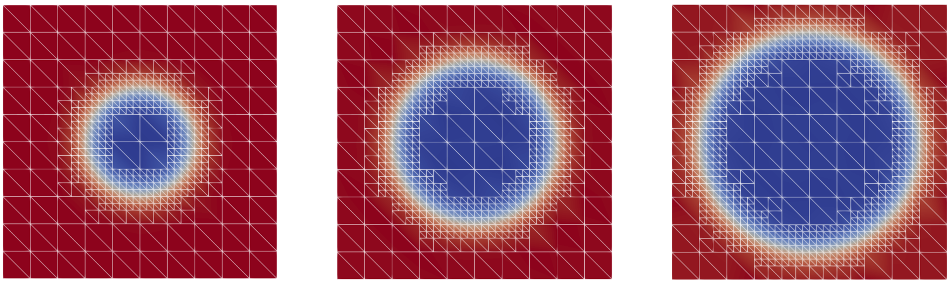

Figure 3: Grain micro structure represented by a phase field in micro problem. Adaptive mesh refinement is used to accurately resolve the phase field at the diffuse interface layer. The hanging nodes seen here are handled in Nutils using a hierarchical basis.

The phase field over the material transition layer needs to be resolved accurately. The material transition layer should have a width of at least three mesh points to be resolved accurately. The grain is represented by having a thin transition layer, which means that the mesh needs to be

[6] Coupled heat conduction code: `https://github.com/IshaanDesai/coupled-heat-conduction`

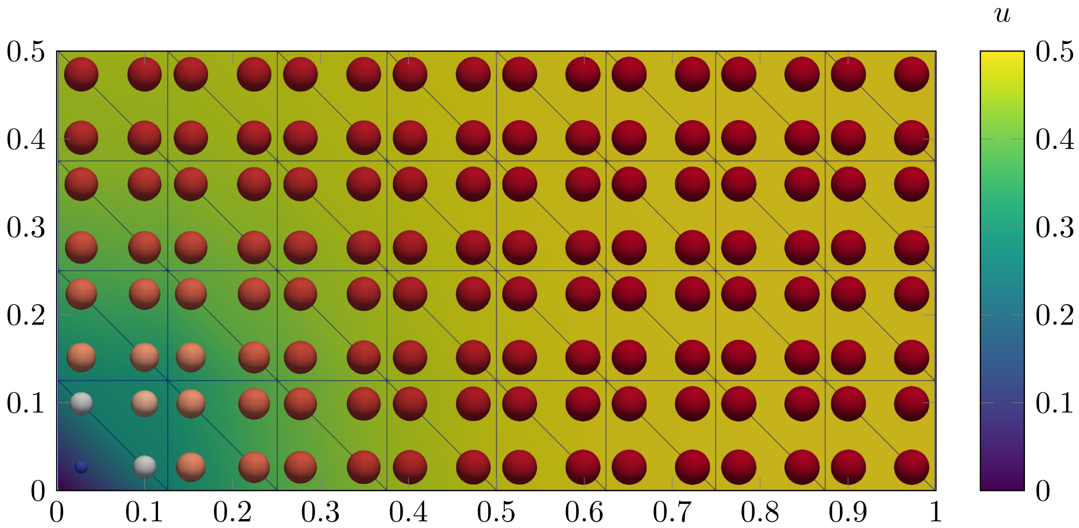

Figure 4: Macro temperature u (background color) and macro grains (overlay circles) scaled by size at time $t = 0.25$. As expected, the lower temperature in the lower left corner results in smaller grain sizes.

very fine. To reduce the computational cost of each micro problem, adaptive mesh refinement is used. The mesh is refined where $0.1 < \phi < 0.9$. Mesh refinement is done in a recursive manner up to 2 levels. A refining and coarsening algorithm similar to Bastidas et al. (2021) [13] is employed to accurately resolve an evolving grain, as shown in Figure 3.

4 RESULTS

The macro problem is run on a single processor and the micro manager is run on 48 processors using MPI parallelization. Using 48 processors to run the micro manager leads to having approximately 2 or 3 micro simulations per processor. The simulation is run till $t = 0.25$ with a time step of 0.01. A serial implicit coupling scheme is used along with an iterative quasi-Newton acceleration scheme. We choose this acceleration scheme as it gave efficient and stable results. From Figure 4 we see that as the value of u decreases, the grain in the micro problem contracts.

This process is similar to the dissolution process considered by Bastidas et al. (2021) [13]. The grain is chosen to be of a lower conductivity than the material around it, hence a shrinking grain leads to a corresponding increase in the upscaled effective conductivity. Figure 5 shows that the porosity on the macro scale increases as the grain contracts. The results are preliminary and have not been validated against a benchmark case.

5 CONCLUSIONS AND OUTLOOK

We presented a software framework for generic two-scale coupled problems. The main focus is to show a preliminary working version of the Micro Manager with a two-scale coupled problem. The model problem is tightly coupled. To get a stable simulation, we used an implicit coupling scheme from preCICE. preCICE offers several variants of acceleration schemes which were used to tune the implicit coupling to obtain a stable and accurate solution. The results show that the Micro Manager works in parallel and is capable of bi-directional implicit coupling. Despite the

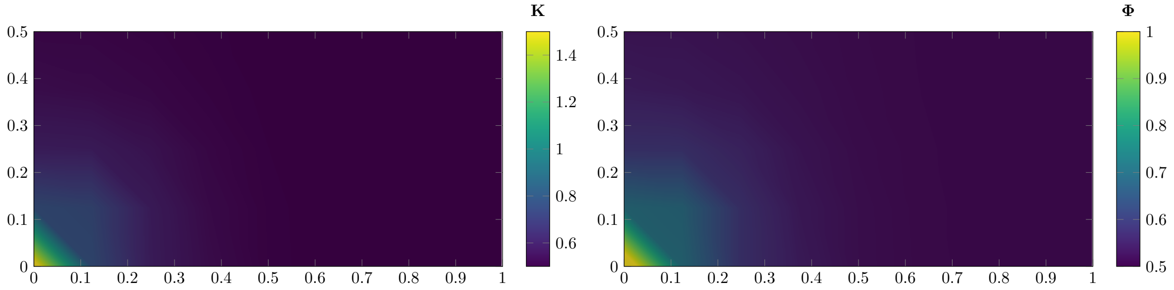

Figure 5: Effective conductivity $\mathbf{K}$ and porosity Φ on the macro scale at time $t = 0.25$.

heterogeneity of the model problem, we observe that several micro simulations are similar to each other. In the future, having adaptive initialization of micro problems would greatly reduce the number of micro problems which need to be solved in each time step [13]. The Micro Manager partitions the macro domain at the start of the simulation, hence adaptivity will lead to a load imbalance at some point. The Micro Manager will need to have dynamic load balancing to be a scalable software. Figure 4 shows that due to the variation of grain sizes with space and time, there is a considerable variation in the mesh size of the micro problems. This further highlights the need of having dynamic load balancing capability in the micro manager. To showcase the application-agnostic nature of our approach, we plan to apply it also to other two-scale coupled problems, such as porous media with flow, human muscle models and human liver models.

6 ACKNOWLEDGMENTS

We are funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

REFERENCES

[1] C. Bringedal, L. von Wolff, I. S. Pop, Phase field modeling of precipitation and dissolution processes in porous media: Upscaling and numerical experiments, Multiscale Modeling & Simulation 18 (2) (2020) 1076–1112. doi:10.1137/19M1239003.

[2] C. Bringedal, I. Berre, I. S. Pop, F. A. Radu, Upscaling of non-isothermal reactive porous media flow with changing porosity, Transport in Porous Media 114 (2) (2020) 371–393. doi:10.1007/s11242-015-0530-9.

[3] A. Klawonn, M. Lanser, M. Uran, O. Rheinbach, S. Köhler, J. Schröder, L. Scheunemann, D. Brands, D. Balzani, A. R. Gandhi, G. Wellein, M. Wittmann, O. Schenk, R. Janalík, EXASTEEL: Towards a Virtual Laboratory for the Multiscale Simulation of Dual-Phase Steel Using High-Performance Computing, 2020, pp. 351–404. doi:10.1007/978-3-030-47956-5_13.

[4] F. Fritzen, M. Fernández, F. Larsson, On-the-fly adaptivity for nonlinear twoscale simulations using artificial neural networks and reduced order modeling, Frontiers in Materials 6 (2019). doi:10.3389/fmats.2019.00075.

[5] L. Lambers, M. Suditsch, A. Wagner, T. Ricken, A multiscale and multiphase model of function-perfusion growth processes in the human liver, PAMM 20 (1) (2021) e202000290. doi:10.1002/pamm.202000290.

[6] G. Chourdakis, K. Davis, B. Rodenberg, M. Schulte, F. Simonis, B. Uekermann, et al., preCICE v2: A sustainable and user-friendly coupling library [version 1; peer review: 2 approved], Open Research Europe 2 (51) (2022). doi:10.12688/openreseurope.14445.1.

[7] D. Groen, J. Knap, P. Neumann, D. Suleimenova, L. Veen, K. Leiter, Mastering the scales: a survey on the benefits of multiscale computing software, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 377 (2142) (2019) 20180147. doi:10.1098/rsta.2018.0147.

[8] L. E. Veen, A. G. Hoekstra, Easing multiscale model design and coupling with muscle 3, in: V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, J. Teixeira (Eds.), Computational Science – ICCS 2020, Springer International Publishing, Cham, 2020, pp. 425–438. doi:10.1007/978-3-030-50433-5_33.

[9] Y.-H. Tang, S. Kudo, X. Bian, Z. Li, G. E. Karniadakis, Multiscale universal interface: A concurrent framework for coupling heterogeneous solvers, Journal of Computational Physics 297 (2015) 13–31. doi:10.1016/j.jcp.2015.05.004.

[10] Pelupessy, F. I., van Elteren, A., de Vries, N., McMillan, S. L. W., Drost, N., Portegies Zwart, S. F., The astrophysical multipurpose software environment, A&A 557 (2013) A84. doi:10.1051/0004-6361/201321252.

[11] S. Alowayyed, D. Groen, P. V. Coveney, A. G. Hoekstra, Multiscale computing in the exascale era, Journal of Computational Science 22 (2017) 15–25. doi:10.1016/j.jocs.2017.07.004.

[12] F. Di Natale, H. Bhatia, et al., A massively parallel infrastructure for adaptive multiscale simulations: Modeling ras initiation pathway for cancer, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '19, Association for Computing Machinery, New York, NY, USA, 2019. doi:10.1145/3295500.3356197.

[13] M. Bastidas, C. Bringedal, I. S. Pop, A two-scale iterative scheme for a phase-field model for precipitation and dissolution in porous media, Applied Mathematics and Computation 396 (2021) 125933. doi:10.1016/j.amc.2020.125933.

[14] M. Redeker, C. Eck, A fast and accurate adaptive solution strategy for two-scale models with continuous inter-scale dependencies, Journal of Computational Physics 240 (2013) 268–283. doi:10.1016/j.jcp.2012.12.025.

[15] L. Dalcin, Y.-L. L. Fang, mpi4py: Status update after 12 years of development, Computing in Science and Engineering 23 (4) (2021) 47–54. doi:10.1109/MCSE.2021.3083216.

[16] G. van Zwieten, J. van Zwieten, W. Hoitinga, Nutils v7.0 (Jan. 2022). doi:10.5281/zenodo.6006701.