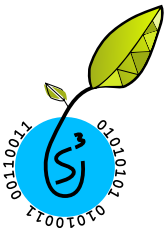



Macro-micro coupling with DuMu^{x} and preCICE

Helena Kschidock
Institute for Parallel and Distributed Systems
University of Stuttgart
April 2023

Abstract:

In this work, we present the implementation of a two-scale heat conduction problem in porous media simulation software DuMu^{x}, where macro and micro simulations are coupled using the coupling library preCICE, as well as its DuMu^{x} adapter and micro manager. We show that multiscale coupled simulations can generally be realised using these software components, and provide a tutorial case for future users. As the same example problem has an existing Nutils implementation, we also compare the results of the various combinations of Nutils and DuMu^{x} macro and micro simulations with each other.

Supervisors:

Ishaan Desai
Dr. Alexander Jaust

Examiner:

Jun.-Prof. Dr. rer. nat. Benjamin Uekermann

Contents

1 Introduction

1.1 Motivation and research questions

The coupling library preCICE [1] couples existing solver software and programs into one connected yet partitioned simulation, allowing users to select a specialized software component for each part of their problem. This can be of great use for multidomain, multiphysics and multiscale simulations. To streamline the coupling procedure and improve user experience, preCICE also develops and maintains its own adapters to many standard open-source solvers.

One of the more recent releases has been the DuMu^x adapter[1]. DuMu^x [2] is an open-source simulation software for Multi-{Phase, Component, Scale, ...} flow and transport in porous media, developed in C++. It is based on DUNE [3], a toolbox for solving PDEs with grid-based methods. The DuMu^x adapter, following the classic adapter design patter, forms the interface between preCICE and DuMu^x, providing standard preCICE functions adapted to DuMu^x syntax, as well as additional functionalities mostly related to the storage of the communicated data, and to making them accessible to the various components of the DuMu^x simulation.

While the adapter has already been used in multi-domain problems coupling between free flow and porous medium flow [4], it has not yet been tested in multiscale simulations. Especially for porous media problems as simulated by DuMu^x, these are however of particular relevance, as pore-scale phenomena can often have a significant effect on Darcy-scale behaviour. Given the extreme computational costs of simulating at pore-scale over macro-scale simulation domain, multi-scale simulations offer a more feasible alternative. Hence the importance of testing the DuMu^x adapter for such applications.

Additionally, this is of particular interest due to its potential uses in the current research surrounding adaptive macro-micro simulations with preCICE and the preCICE Micro Manager [5], a python-based tool designed to efficiently facilitate many-to-one coupling. Furthermore, the adapter has so far only been used in DuMu^x-DuMu^x coupled simulations.

All of these aspects lead to two main research questions:

- Can multi-scale coupling using DuMu^x, preCICE, the DuMu^x adapter and the Micro Manager be achieved?

[1] `https://github.com/precice/dumux-adapter`

- Is the current DuMu^x adapter capable of handling multiscale simulations? If not, what is missing and which extensions are necessary?

In response, we therefore present an example of a two-scale ("micro-macro") problem coupled via preCICE, the DuMu^x adapter and the Micro Manager, showcasing the compatibility of the software components with each other and testing the DuMu^x adapter's capabilities in a multi-scale context, as well as coupling it to a different solver for the first time. As a way to verify our numerical results, and to test the inter-solver coupling between DuMuX and Nutils, we compare our results to an existing implementation of the same test problem in Nutils, a python-based library for Finite Element Method computations [6]. We test all four possible combinations of macro- and micro simulations.

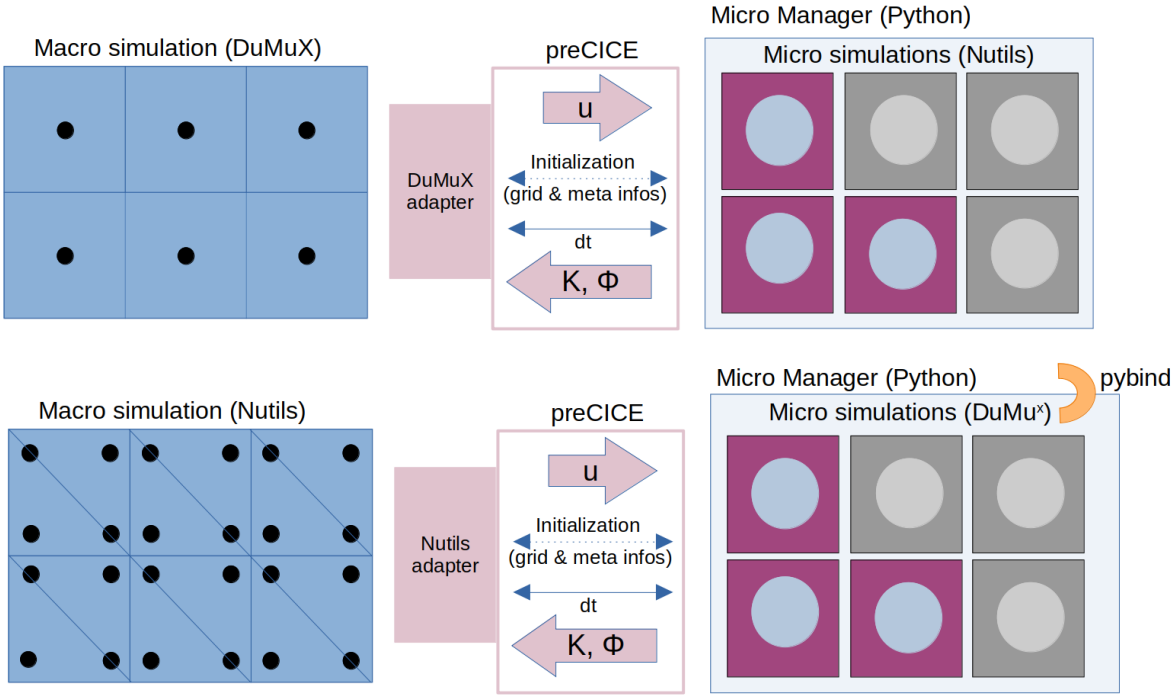


Figure 1.1: General setup of the communications chain for all four simulation components. Micro simulations may be active (purple) or inactive (grey).

1.2 The software stack

The general setup and chain of communications of the coupled simulations for all four components is shown in Figure 1.1. preCICE uses a partitioned blackbox approach to coupling, i.e. after initialization where grid information on the macro side is used to launch the correct micro simulation, only select input and output data is exchanged between the simulations. The macro simulation is coupled directly via preCICE using its corresponding preCICE adapter.

On the micro side, no adapter is used. Instead, the preCICE Micro Manager [5] handles the many-to-one coupling of the micro simulations. As it is python-based, this requires using the python bindings for DuMu^x micro simulations, which are written in C++. The Micro Manager can adaptively activate and deactivate micro simulations using on adaptivity criteria based on the input and output data communicated. Micro simulations can be launched at arbitrary points of the macro simulation. In the DuMu^x example project, there is only exactly one micro simulation per macro cell, situated at the cell center (see Figure 1.2).

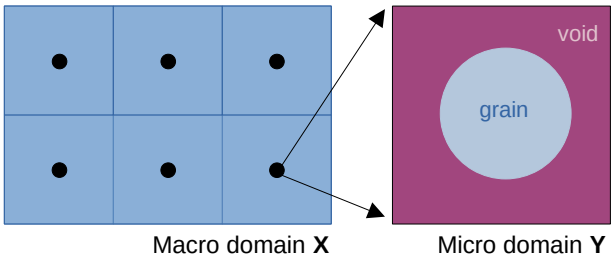


Figure 1.2: Illustration of the two-scale simulation.

2 Theory of the coupled heat problem

Porous media problems lend themselves to a multi-scale approach whenever changes in the microstructure (at pore-scale) become significant enough to impact macroscopic phenomena, such as flow behaviour or heat conductivity. Reasons for deformation on the microscale include mineral precipitation and dissolution as detailed in [7], and temperature changes, as described in the two-scale heat conduction problem in [5]. We choose the latter as our example problem to test the DuMu^{x} adapter in a realistic setting, as it has already been implemented[1] in Nutils.

In the following, the problems on both scales are explained:

2.1 The micro problem

On the micro scale, the porous medium consists of solid circular "grains" of one material, surrounded by "void" (pores) filled with a second, different material, which can be imagined as sand or a stationary fluid. The specifics of these materials beyond their respective densities ρ_g, ρ_v, specific heat capacities c_g, c_v and heat conductivities k_g, k_v mostly don't matter, as in the constructed problem there is no flow and the equations are dimensionless. Based on the current corresponding macro temperature, each simulation on the micro scale simulates the expansion or contraction of a single grain of the porous medium. We achieve this by using a phase-field approach as proposed in [8]; the changing cell geometry is represented by a phase-field indicator ϕ, which approaches 0 within the grain, and 1 in the void. Between these, a transition layer of non-vanishing width is constructed, in which ϕ changes smoothly. This eradicates the need for differentiation between time-dependent void and grain domains, as the same equations now apply consistently across the whole micro domain $\mathbf{Y}$. Each micro-simulation then essentially goes through the same four steps (for further details and the derivation of the following equations see [5] and [8]):

1. The evolution of the phase-field based on the macro temperature u is determined by solving the Allen-Cahn equation

$$\lambda^2 \partial_t \phi + \gamma P'(\phi) = \gamma \lambda^2 \nabla^2 \phi - 4\lambda\phi(1-\phi) f(u), \tag{2.1}$$

[1] `https://github.com/IshaanDesai/coupled-heat-conduction`

where $P(\phi) = 8\phi^2(1-\phi)^2$ is the double-well potential, γ is the diffusion coefficient and $f(u) = k_u\left(\frac{u^2}{u_{eq}^2} - 1\right)$ determines the rate at which grains expand or contract, dependent on a constant k_u and an equilibrium temperature u_{eq}.

2. The "cell problem"

$$\nabla \cdot ((k_g(1-\phi) + k_v\phi)(\mathbf{e}_j + \nabla\psi^j)) = 0 \quad \text{with} \quad \int_{\mathbf{Y}} \psi^j dy = 0 \tag{2.2}$$

is solved for the weights ψ^j.

3. Integrating the updated ϕ over the entire micro domain returns the porosity $\Phi = \int_{\mathbf{Y}} \phi dy$.

4. Using the calculated ψ^j, the effective upscaled conductivity matrix $\mathbf{K} = (\mathbf{K}_{ij})$ is determined by means of

$$\mathbf{K}_{ij} = \int_{\mathbf{Y}} (k_g(1-\phi) + k_v\phi)(\delta_{ij} + \partial_{y_i}\psi^j) dy. \tag{2.3}$$

Φ and $\mathbf{K}$ are then passed to the macro problem.

2.2 The macro problem

On the macro scale, there is no differentiation between the different materials; instead, we assume a homogenous "porous medium", with porosity Φ and (potentially anisotropic) effective conductivity tensor $\mathbf{K}$ as communicated by the micro problem. Each macro cell then solves the heat equation

$$\partial_t(\Phi\rho_v c_v u + (1-\Phi)\rho_g c_g u) = \nabla \cdot (\mathbf{K}\nabla u) \quad \text{in } \mathbf{X} \tag{2.4}$$

for temperature u, which is consequently passed to the corresponding micro problem.

3 The DuMu$^{\mathrm{x}}$ implementation of the macro and micro simulations

For simplicity, we use a structured, regular 2D grid in both the macro and micro simulation. At the center of each grid cell, a micro-simulation is launched, leading to a 1-to-1 correspondence between macro grid cells and micro simulations, which differs from the use of several macro simulations per (triangular) macro cell in [5].

All three of our model equations - the macro heat equation, the Allen-Cahn equation and equation of the cell problem can be brought into the form of a balance equation

$$\frac{\partial m(u)}{\partial t} + \nabla \cdot f(u, \nabla u) + q(u) = 0, \tag{3.1}$$

where u is the quantity to be solved for, m is the storage, f is the flux, and q is the source. In DuMu$^{\mathrm{x}}$, m, f and q have to be implemented depending on the specific problem and discretization. In our case, both macro and micro DuMu$^{\mathrm{x}}$ equations are spatially discretized using the same cell-centered finite volume scheme, the two-point flux approximation (TPFA). The general finite volume theory detailed here has been taken mostly from the dumux handbook[1].

3.1 Finite volumes and the CCTPFA discretization

Finite volume formulations are derived by integrating the relevant equations over a control volume K and applying the Gauss divergence theorem. For cell-centered schemes, the control volumes correspond directly to the grid elements. For (3.1), this leads to

$$\int_K \frac{\partial m(u)}{\partial t} \mathrm{d}\Omega + \int_{\partial K} \mathbf{f}(u, \nabla u) \cdot \mathbf{n} \mathrm{d}\Gamma + \int_K q \mathrm{d}x = 0, \tag{3.2}$$

and discretely,

$$M_K + \sum_{\sigma \in \partial K} F_{K,\sigma} + Q_K = 0, \tag{3.3}$$

[1] `https://dumux.org/docs/handbook/releases/3.6/dumux-handbook.pdf`

where $F_{K,\sigma}$ is a discrete approximation $F_{K,\sigma} \approx \int_\sigma \mathbf{f}(u, \nabla u) \cdot \mathbf{n} \mathrm{d}\Gamma$ of the exact flux through face σ flowing out of cell K. Amongst other aspects, finite volume schemes most strongly differ through how they define this approximative term $F_{K,\sigma}$. Our three flux terms

(i) $-(\gamma\lambda^2\nabla\phi)\cdot\mathbf{n}$ for the Allen-Cahn equation,

(ii) $-((k_g(1-\phi)+k_v\phi)\nabla\psi^j)\cdot\mathbf{n}$ for the cell problem and

(iii) $-(\mathbf{K}\nabla u)\cdot\mathbf{n}$ for the macro heat equation

can all be written in the form $(-\mathbf{\Lambda}\nabla u)\cdot\mathbf{n}$ for a 2×2 tensor $\mathbf{\Lambda}$, i.e. $F_{K,\sigma} \approx \int_\sigma (-\mathbf{\Lambda}_K \nabla u)\cdot\mathbf{n}\mathrm{d}\Gamma$. We assume Λ_K to be symmetric and positive definite, which is trivial for (i) and (ii); for (iii), off-diagonal elements approach zero. Then $(\mathbf{\Lambda}_K\nabla u)\cdot\mathbf{n} = \nabla u \cdot \mathbf{\Lambda}_K\mathbf{n}$, the directional derivative of u in direction $\mathbf{\Lambda}_K\mathbf{n}$. For TPFA, we then apply co-normal decomposition

$$\mathbf{\Lambda}_K\mathbf{n}_{K,\sigma} = t_{K,\sigma}\mathbf{d}_{K,\sigma} + \mathbf{d}^\perp_{K,\sigma} \quad \text{with} \quad t_{K,\sigma} = \frac{\mathbf{n}^T_{K,\sigma}\mathbf{\Lambda}_K\mathbf{d}_{K,\sigma}}{\mathbf{d}^T_{K,\sigma}\mathbf{d}_{K,\sigma}}, \tag{3.4}$$

where $\mathbf{d}_{K,\sigma}$ is the distance vector between the cell center and the center of the shared face between two adjacent cells and $\mathbf{d}^\intercal_{K,\sigma}\mathbf{d}^\perp_{K,\sigma} = 0$ (see Figure 3.1). The $t_{K,\sigma}$ are called transmissibilities. We neglect the second term, resulting in

$$\nabla u \cdot \mathbf{\Lambda}_K\mathbf{n}_{K,\sigma} \approx t_{K,\sigma}\nabla u\cdot\mathbf{d}_{K,\sigma} \quad \forall K,\sigma. \tag{3.5}$$

Using $\nabla u\cdot\mathbf{d}_{K,\sigma} \approx u_\sigma - u_K$, we finally arrive at our flux approximation

$$F_{K,\sigma} = -|\sigma| t_{K,\sigma}(u_\sigma - u_K). \tag{3.6}$$

We use local flux conservation $F_{K,\sigma} + F_{L,\sigma} = 0$ to derive

$$u_\sigma = \frac{t_{K,\sigma}u_K + t_{L,\sigma}u_L}{t_{K,\sigma}+t_{L,\sigma}} \tag{3.7}$$

and obtain

$$F_{K,\sigma} = |\sigma|\frac{t_{K,\sigma}t_{L,\sigma}}{t_{K,\sigma}+t_{L,\sigma}}(u_K - u_L), \tag{3.8}$$

as discrete values are generally defined at cell centers. TPFA is only consistent for cases in which $\nabla u\cdot\mathbf{d}^\perp_{K,\sigma} \approx 0$. For the regular, rectangular grid used here, this condition is always fulfilled.

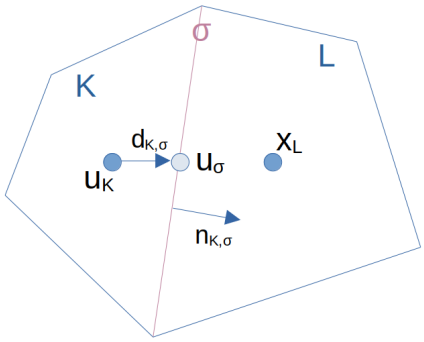


Figure 3.1: Two adjacent cells K, L, with common face σ and cell centers x_K, x_L.[2]

3.2 The DuMu^x implementation of the macro simulation

In DuMu^x, we construct the heat conduction problem as a single phase non-isothermal (`OnePNI`) porous medium flow problem, which is solved using Newton's method and an implicit Euler scheme. This allows us to utilize a lot of existing functionality of the API. The nonisothermal model uses the following energy conservation equation

$$\Phi\frac{\partial(\rho_v(h_v - p/\rho_v)S_v)}{\partial t} + (1-\Phi)\frac{\partial(\rho_g c_g u)}{\partial t} - \nabla\left(\rho_v h_v \frac{k_{rv}}{\mu_v}\mathbf{P}(\nabla p - \rho_v g)\right) - \nabla(\mathbf{K}\nabla u) - q = 0 \quad (3.9)$$

which, upon closer inspection – with enthalpy $h_v = c_v * (T - T_{\text{ref}}) + p/\rho_v$ for a reference temperature T_{ref}, zero gravity g, source term $q = 0$, saturation $S_v = 1$ and constant pressure p – reduces to Equation 2.4, the macro heat conduction equation.

Thus DuMu^x is solving the desired problem, as long as the pressure is constant, which we enforce through appropriate pressure boundaries and initial conditions, set to inhibit all flow. For the temperatures, we use adiabatic boundary conditions everywhere except the bottom left corner, where zero-Dirichlet boundary conditions are used, see 3.2

As detailed in Section 2, the macro simulation receives the conductivity tensors K and porosi-

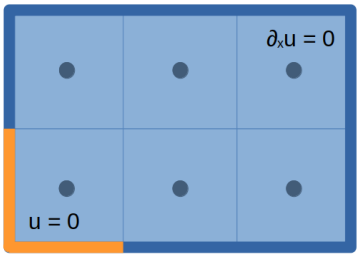


Figure 3.2: Temperature boundary conditions used in the macro problem

[2] `https://dumux.org/docs/handbook/releases/3.5/dumux-handbook.pdf`

ties ϕ from all the micro simulations. Once received, the conductivities and porosities then have to be fed into the DuMu^x simulation instead of DuMu^x's default values. The base implementation in DuMu^X assumes a static, scalar conductivity; however setting a local, potentially anisotropic conductivity tensor is essential to the example problem, especially to the asymmetric case. As the relevant equations in DuMu^x already support tensorial conductivities, we set these in each element through some modifications to the `VolumeVariables` and `EnergyVolumeVariables`.

3.3 The DuMu^x implementation of the micro simulation

In the implementation of the micro simulation, there are three non-trivial aspects: the Allen-Cahn problem, the cell problem, and the integrataion of the conductivity tensorial components (2.3), specifically the calculation of the partial derivatives $\partial_{y_i}\psi^j$ therein. We use periodic boundary conditions.

An **Allen-Cahn problem** has already been implemented as part of the DuMu^x module DUMUX-PHASEFIELD[3], which we adapt to our purposes. The initial phasefield is defined analytically as

$$\phi(x,y) = \frac{1}{(1+\exp(\frac{-4}{\lambda\sqrt{(y-y_0)^2+(x-x_0)^2}-R_0}}, \tag{3.10}$$

where (x_0, y_0) is the cell center and R_0 is the initial radius of the grain. Like the macro problem, the Allen-Cahn problem is solved using Newton's method and an implicit Euler scheme.

The **Cell Problem** is implemented as a new custom problem. Following the considerations in Section 3.1, we set storage and source terms to zero and construct our flux (ii) according to the derived approximation $F_{K,\sigma}$ in (3.8). We base our implementation on the DuMu^x TPFA implementation of a single-phase isothermal (porous medium) flow model[4]. It builds on Darcy's law and solves a mass continuity equation whose flux term

$$\nabla\left(-\rho\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g})\right) \tag{3.11}$$

with density ρ, permeability tensor $\mathbf{K}$, viscosity μ, gravity vector $\mathbf{g}$ and pressure p as the primary variable resembles our cell problem in form. The cell problem is stationary, and is solved twice per iteration (once per unit vector $\mathbf{e}_j$) using a linear solver.

The **derivative $\partial_{y_i}\psi^j$ in the K integral** had to be newly implemented as DuMu^x currently does not provide any methods to calculate spatial derivatives of primary variables for TPFA

[3]Mathis Kelm, https://git.iws.uni-stuttgart.de/dumux-appl/dumux-phasefield
[4]https://dumux.org/docs/doxygen/releases/3.6/a20035.html

discretization. We base our considerations on the discrete gradient defined in [9],

$$\nabla_K u = \frac{1}{|K|} \sum_{\sigma \in \varepsilon_K} |\sigma| (u_\sigma - u_K) \mathbf{n}_{K,\sigma}, \tag{3.12}$$

where K is the control volume (element), $|K|$ its volume, σ a face of the control volume with area $|\sigma|$. The solution on the face u_σ is calculated as in (3.7), i.e. as the harmonic average of the neighbouring solution values with the corresponding transmissibilities for $\Lambda_K = 1$.

3.4 Parameters

All simulations, both in Nutils and DuMu$^{\text{x}}$, were run with the following parameters, following our reference example in [7] as closely as possible, although we use a timestep $dt = 0.005$ instead of $dt = 0.01$ to ensure convergence.

General	
Timestep width dt	0.005
Size of the macro domain	1 0.5
Number of micro simulations	128
Adaptivity	
History parameter	0.1
Coarsening constant	0.2
Refining constant	0.05
Allen-Cahn problem	
Phasefield parameter λ	0.08
Phasefield diffusivity γ	0.01
Equilibrium concentration u_{eq}	0.05
Initial grain radius 0.4	
Cell Problem	
Solid thermal conductivity k_g	0.0
Liquid thermal conductivity k_s	1.0
Macro problem	
Initial Temperature T_0	0.5
Solid density ρ_g	1.0
Liquid density ρ_v	1.0
Solid heat capacity c_g	1.0
Liquid heat capacity c_v	1.0
Liquid reference temperature	0.5

Adaptivity parameters here refer to the adaptivity of the Micro Manager, which determines

which/how many micro simulations are active.

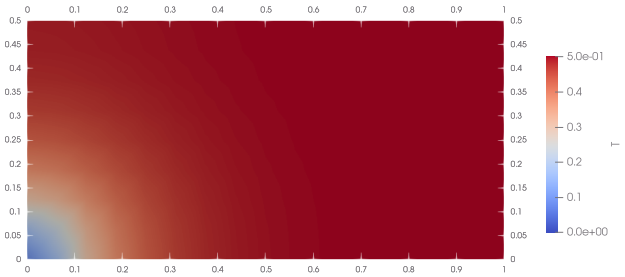
There do however remain several differences between the DuMu^x and Nutils implementations. Two aspects are most significant: Firstly, Nutils as a Finite Element solver does not launch one micro simulation per macro cell as is the case in our DuMu^x implementation; instead, each macro cell launches four micro simulations at Gauss points (see [5]). For better comparison, we therefore fix the number of total micro simulations at 128, leading to a 16×8 macro grid in DuMu^x versus a 8×4 grid in Nutils. Secondly, the Nutils micro simulation uses adaptive grid refinement, with a 10×10 grid on the coarsest level and up to 3 levels of refinement. In DuMu^x, we are instead simulating the full simulation on an 80×80 grid.

4 Results

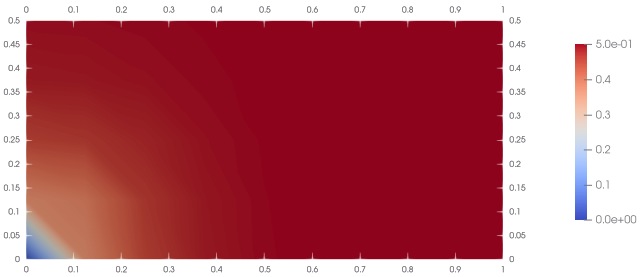
Results are shown in Figures 4.1 and 4.2 for concentrations/temperatures u, in Figure 4.3 for porosities ϕ and in Figure 4.4 for the first component $\mathbf{K}_{00}$ of the conductivity tensor. We compare all four combinations for macro- and micro solvers: Dumu$^{\text{x}}$-Dumu$^{\text{x}}$, Dumu$^{\text{x}}$-Nutils, Nutils-Dumu$^{\text{x}}$, and Nutils-Nutils.

For the concentrations/temperatures, we observe the expected overall behaviour, with lower temperatures "diffusing" out from the bottom left corner, where Dirichlet boundary conditions are set to 0.0. Lower temperatures lead to smaller grain sizes and higher porosities.

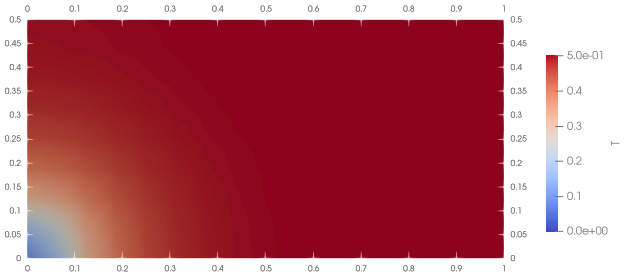
We also observe that due to the large gradient near our "sink" in the bottom left corner, at the early timestep $t = 0.05$ (Figure 4.1), the form of the grid cells still leads to the most striking feature and appears to dominate the simulation. With better resolution, this difference should however be diminished. In comparison in the later time step $t = 0.25$ (Figure 4.2), we see that the behaviour of the micro simulations now seems to dominate, with DuMu$^{\text{x}}$ micro simulations leading to a stronger/faster overall decrease in temperature in the upper left part of the domain. Of course, we would expect micro simulation behaviour more or less equivalent; looking at $\mathbf{K}_{00}$ in Figure 4.4, this is however clearly not the case, with DuMu$^{\text{x}}$ and Nutils micro simulations, while showing the same overall behaviour, also exhibiting a clear shift in values. While some of this might be attributed to the above differences between Nutils and DuMu$^{\text{x}}$, such a strong shift might also indicate a modelling or implementation error, which should be investigated further.



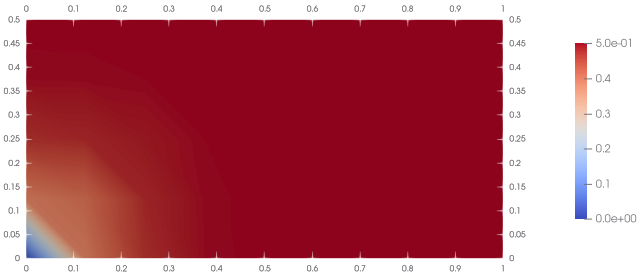
(a) DuMu^{x}-DuMu^{x}



(b) Nutils-DuMu^{x}

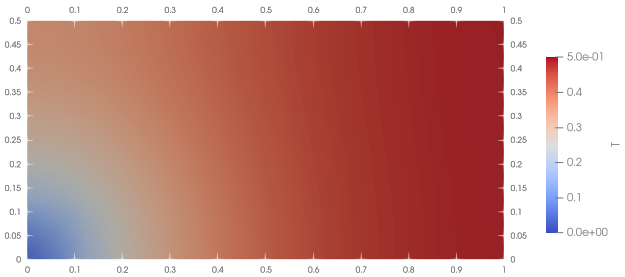


(c) DuMu^{x}-Nutils

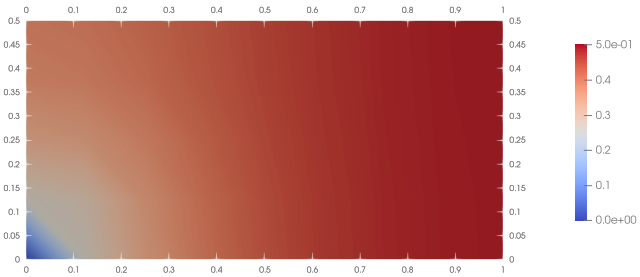


(d) Nutils-Nutils

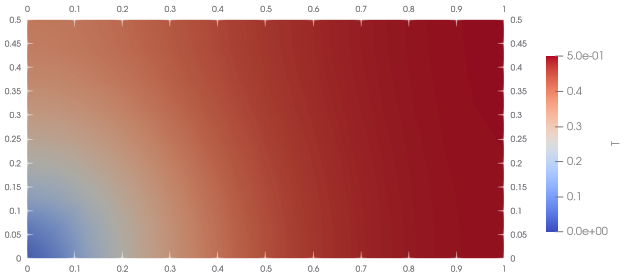
Figure 4.1: Concentrations at $t = 0.05$.



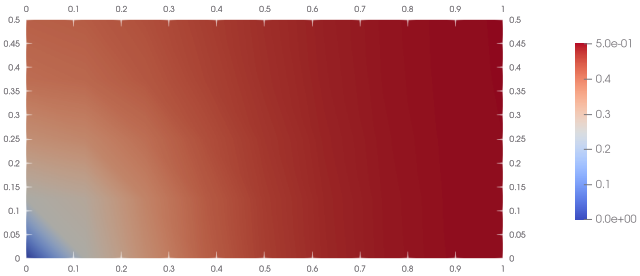
(a) DuMu^{x}-DuMu^{x}



(b) Nutils-DuMu^{x}

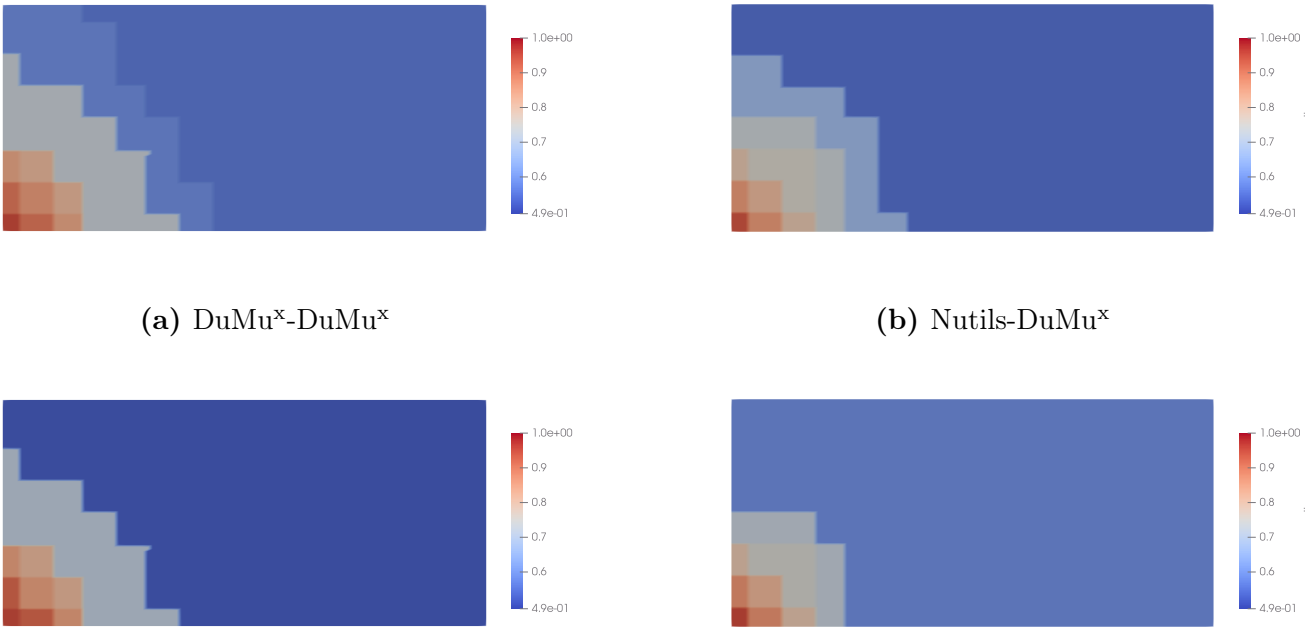


(c) DuMu^{x}-Nutils



(d) Nutils-Nutils

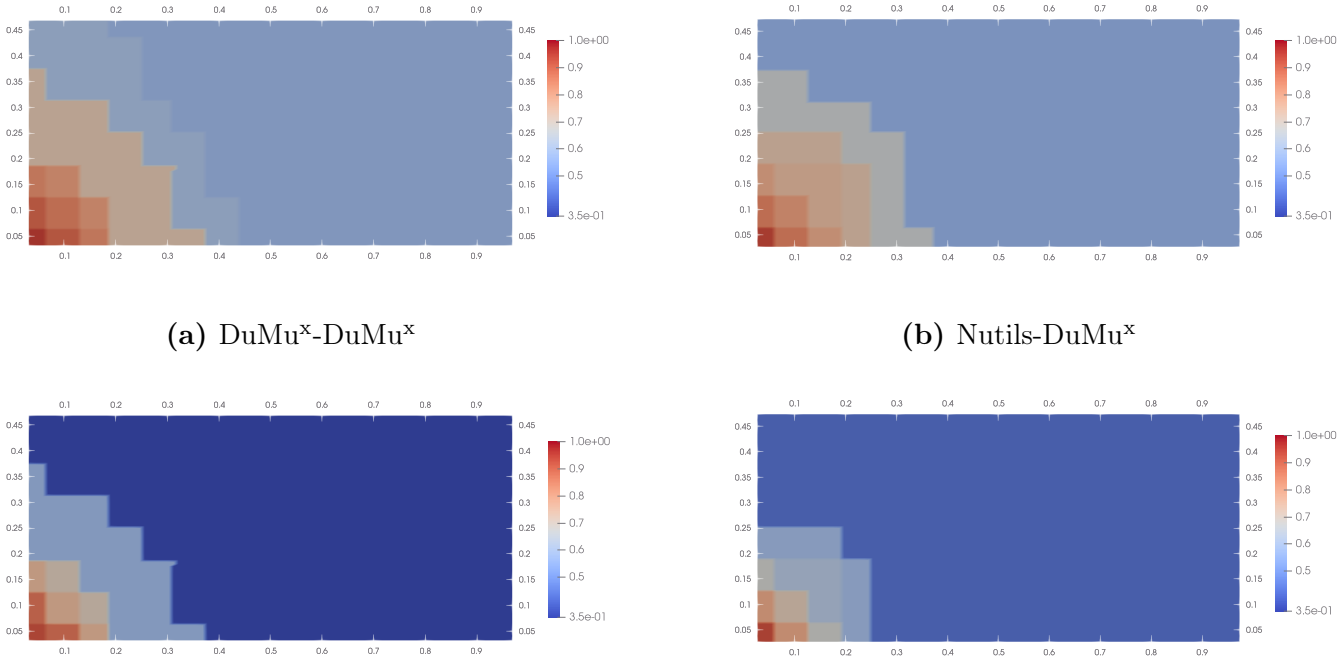
Figure 4.2: Concentrations at $t = 0.25$.



(a) DuMu^{x}-DuMu^{x} **(b)** Nutils-DuMu^{x}

(c) DuMu^{x}-Nutils **(d)** Nutils-Nutils

Figure 4.3: Porosities at $t = 0.25$.



(a) DuMu^{x}-DuMu^{x} **(b)** Nutils-DuMu^{x}

(c) DuMu^{x}-Nutils **(d)** Nutils-Nutils

Figure 4.4: First component $\mathbf{K}_{00}$ of the conductivity tensor at $t = 0.25$.

5 Conclusion

As this work shows, the DuMu^x adapter can be used in a multi-scale context and preCICE can not only successfully couple DuMu^x-DuMu^x simulations, but also succeeds in coupling between DuMu^x to other solvers. Over the course of the project, we have also improved the adapter by fixing smaller errors - such as dependencies and CMake macros - and added vector quantity support.

We have furthermore provided another example of successfully using the preCICE Micro Manager to facilitate many-to-one coupling. Set in motion by this project, C++ bindings have been added to the Micro Manager, vastly increasing its future usability. We are also showcasing the capabilites of the adaptive scheme, which allows even the python-based simulations to be run on local machines. While the results of the micro simulations of the two solvers still show differences, the entire software stack has been shown to work well together, and future projects with a similar setup are to be expected.

6 Bibliography

[1] G. Chourdakis et al. "preCICE v2: A sustainable and user-friendly coupling library [version 2; peer review: 2 approved]". In: *Open Res Europe* (2022). URL: https://doi.org/10.12688/openreseurope.14445.2.

[2] T. Koch et al. "DuMu^x 3 - an open-source simulator for solving flow and transport problems in porous media with a focus on model coupling". In: *Computers & Mathematics with Applications* (2020). ISSN: 0898-1221. DOI: 10.1016/j.camwa.2020.02.012.

[3] M. Blatt et al. "The Distributed and Unified Numerics Environment, Version 2.4". In: *Archive of Numerical Software* 4.100 (2016), pp. 13–29. ISSN: 2197-8263. DOI: 10.11588/ans.2016.100.26526. URL: http://dx.doi.org/10.11588/ans.2016.100.26526.

[4] A. Jaust et al. "Partitioned Coupling Schemes for Free-Flow and Porous-Media Applications with Sharp Interfaces". In: *Finite Volumes for Complex Applications IX - Methods, Theoretical Aspects, Examples*. Ed. by Robert Klöfkorn et al. Cham: Springer International Publishing, 2020, pp. 605–613. ISBN: 978-3-030-43651-3.

[5] I. Desai, C. Bringedal, and B. Uekermann. "A Flexible Software Approach To Simulate Two-Scale Coupled Problems". In: *(submitted)* (2022).

[6] J.S.B. van Zwieten, G.J. van Zwieten, and W. Hoitinga. *Nutils 7.0*. 2022. DOI: 10.5281/zenodo.6006701.

[7] M. Bastidas Olivares, C. Bringedal, and I. S. Pop. "A two-scale iterative scheme for a phase-field model for precipitation and dissolution in porous media". In: *Applied Mathematics and Computation* 396 (2021), p. 125933. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2020.125933. URL: https://www.sciencedirect.com/science/article/pii/S0096300320308869.

[8] C. Bringedal, L. von Wolff, and I. S. Pop. "Phase Field Modeling of Precipitation and Dissolution Processes in Porous Media: Upscaling and Numerical Experiments". In: *Multiscale Modeling & Simulation* 18.2 (2020), pp. 1076–1112. DOI: 10.1137/19M1239003. URL: https://doi.org/10.1137/19M1239003.

[9] Martin Schneider et al. “Comparison of finite-volume schemes for diffusion problems”. In: *Oil & Gas Science and Technology – Revue d’IFP Energies nouvelles* 73 (2018). Ed. by A. Anciaux-Sedrakian and Q. H. Tran, p. 82. DOI: 10.2516/ogst/2018064. URL: https://doi.org/10.2516/ogst/2018064.