A novel Multi-spatial analysis of flow and transport in porous media

Zheng He, Xuan Gu, Xiaoyu Sun[a, *], Jianxin Teng, Binsheng Wang

College of Aerospace and Civil Engineering, Harbin Engineering University, Harbin, 150001 China

[a]Correspondent author: sunxiaoyu520634@163.com

Abstract

DuMux is a free and open-source simulator for flow and transport processes in porous media, based on the Distributed and Unified Numerics Environment (DUNE). Its main intention is to provide a sustainable and consistent framework for the implementation and application of model concepts, constitutive relations, discretizations, and solvers. The paper provides an overview of DuMux with the focus on software-related aspects. Selected examples highlight the multi-scale and the parallel capabilities.

Keywords

Porous media simulator; Open source; Multi-scale; Multi-physics.

1. Introduction

The quality of any type of computational modeling crucially depends on the quality of the employed software framework. Research codes very often fail to be developed and maintained in a continuous manner. On the contrary, software development at academic institutions usually is highly fragmented and driven by individual short-term needs. Furthermore, work is often done redundantly, diverting resources from the original focus of research projects by a need for reinventing the wheel. We are therefore convinced that the free and open source (FLOSS) idea provides a chance for sustainable high quality software development also in academia.

DuMux is a simulator for flow and transport processes in porous media. It is built on top of DUNE, the Distributed and Unified Numerics Environment, a modular toolbox for solving partial differential equations with grid-based methods [1-2]. DuMux is licensed under the terms and conditions of the GNU General Public License (GPL) version 2 or later [3]. Stable releases are available for download [4], and anonymous read-access to the Subversion repository is granted.

This paper is structured as follows: in the remaining part of this introduction, the vision, concept and design ideas behind DuMux are presented. In Section 2, the common base of all DuMux models is outlined. Section 3 describes the available models in DuMux. In Section4 some examples highlighting the capabilities of DuMux are given. We summarize and give an outlook in Section 5. Concerning the notation it should be mentioned that quantities that have a direct representation within the code base of DuMux – be it a class name or the name of a folder containing a model – are set in typewriter.

2. Simulation

Modularity is the leitmotif to design the code. DuMux provides shelves of modularized objects, enabling the user to choose the appropriate parts according to the problem at hand. This can be compared to somebody, who can easily grab a different shirt out of a shelf without changing the trousers and without ending up with a combination of clothes that do not match. Following that design idea, DuMux is meant to provide a collection of shelves each holding interchangeable alternatives, that are still fully interactive to other shelves by using common interfaces. The user is able to select each part of the implementation at each shelve through an efficient compile-time property system (Section 2.4). Part of this modular setup (seeFig. 1) is the shelves from which to choose model concepts (Section 3),numerical schemes,control strategies for the simulation (Section 2.3),multitude of substances, material laws (Section 2.2),small and large–scale examples and applications.

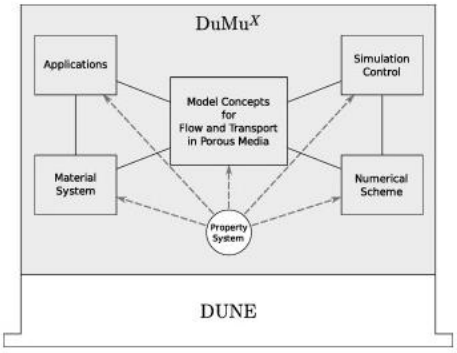
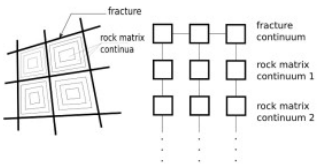


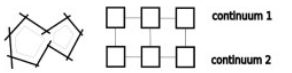
Fig. 1. Modular design of DuMux.

2pDFM. The model simulates the two-phase flow in fractured porous systems using a discrete fracture model (DFM) approach [5], with a lower dimensional representation for the fractures [6]. The representative fracture network of the DFM is reconstructed with a geostatistic fracture generator [7].

2pMINC. In contrast to the 2pDFM, this model does not discretize the fractures but simulates the two-phase flow in fractured porous media using the MINC method[8-9]. The fractures are treated as an equivalent homogeneous porous medium, which requires the determination of appropriate effective parameters and transfer functions between continua, but reduces considerably the geometrical complexity of the problem, see Fig. 2a.



(a) MINC discretization of a fractured porous system



	2pDFM	1p2cDouble Continuum	1DPipe3D PorousFlow
cont. 1	Discrete Fracture System	Capillary blood vessels	Pipe System
cont. 2	Rock Matrix Continuum	Biological Tissue	Rock Matrix Continuum

(b) Example of a double-continuum discretization for various applications

Fig. 2. Sketch of the relation between model concepts and the continuum representation. Schematic diagrams of connectivity for the continuum models.

One basic idea of the general framework is that any of the fine-scale models, which are available in DuMux, can be chosen for the local calculations without any changes in these model implementations. A second idea is that different kinds of post-processing routines, which calculate the effective parameters, are implemented once, and can be easily reused as well as combined in different kinds of local(-global) upscaling methods. Thirdly, depending on the upscaling approach one of the fine-scale model implementations could also be used on the coarse scale (with different parameters) or new coarse-scale models can be implemented, which again might be combined respectively with the different fine-scale and post-processing methods. Fig. 3 shows the general flow of a local(-global) multi-scale simulation in DuMux. In the following paragraphs, we describe the conceptual and the mathematical model for one exemplary implementation of a local upscaling method and show some results.

The models of this example are discretized by a cell-centered finite volume method. To account for the full tensor permeabilities a MPFA o-method is used for discretization of the pressure equation. The setup of the simple test-example (2-D) is shown in Fig. 4a. The computational domain is the unit square. The heterogeneous permeability field (Fig. 4b) is randomly generated by the open source tool GSTAT [10], which can be used for geostatistical modeling. A uniform fine grid of 100 × 100 elements is used, while the coarse grid consists of 10 × 10 elements. Quantities other than intrinsic permeabilities are assumed to be homogeneous over the entire model domain and not included in the upscaling procedure. The relative permeabilities are linear functions of saturation and we assume a constant porosity of 0.2. Initially, the domain is fully water-saturated, and replaced by oil from the left. The result of the coarse-scale simulation is shown in Fig. 5b and shows good agreement with the

fine-scale reference solution depicted in Fig. 5a. Even for the quite simple and small test problem a significant speed-up of the multi-scale model (> factor 40) can be achieved. Since the emphasis of this example is not to discuss the quality of existing and well established upscaling methods, but to show that they can be easily implemented and used within the DuMux multi-scale framework, we refer to [11] for details.

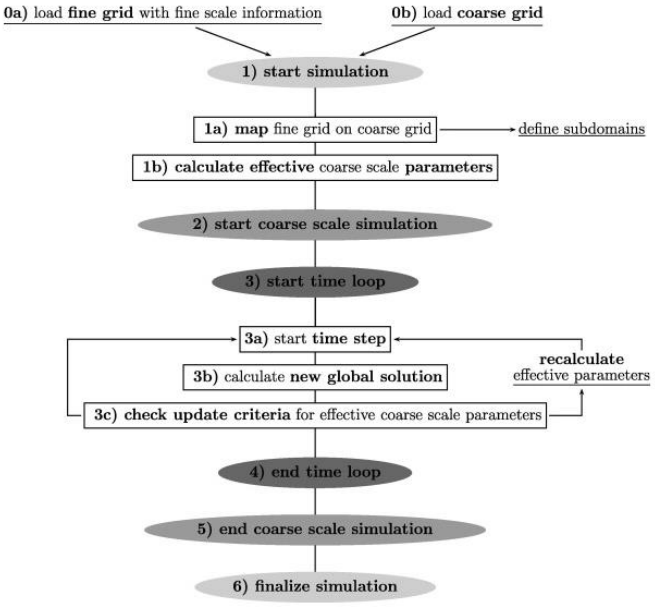


Fig. 3. Flow of a multi-scale simulation in DuMux.

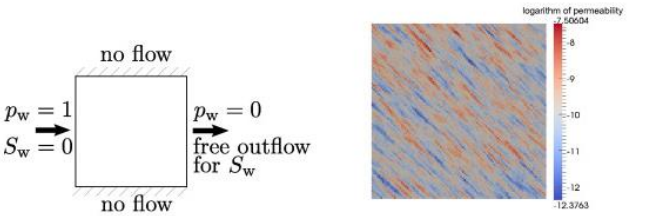


Fig. 4. Setup of the multi-scale simulation example.

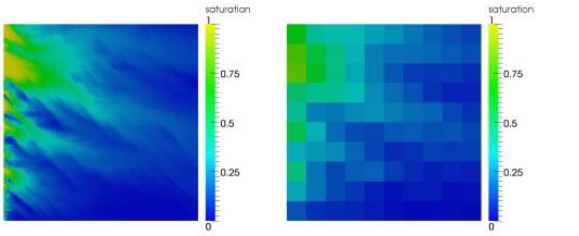


Fig. 5. Saturation distribution at $t = tend$.

The model domain and the position of the injection well are shown in Fig. 6. The domain has lateral dimensions of 9600 m × 8900 m and a varying thickness of 90–140 m. The fault within the aquifer is assumed to be an infinitely permeable fault and is included as a Dirichlet boundary with hydrostatic pressure and geothermal temperature distribution.

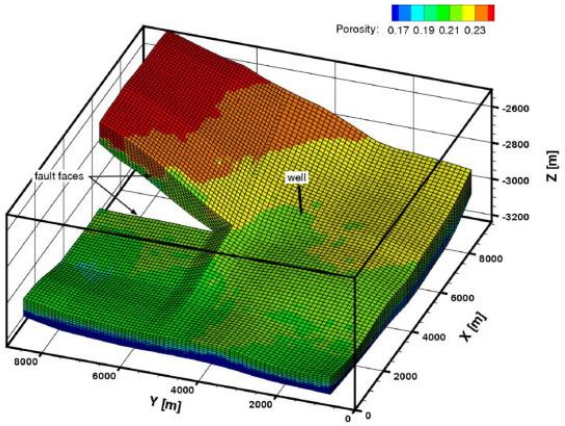
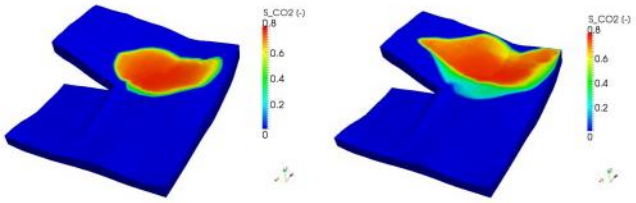


Fig. 6. Aquifer geometry and porosity distribution of the Benchmark Problem 3.1 given in [12]. The aquifer lies in a depth between 2500 m and 3200 m, the position of the injection is x = 5440 m and y = 3300 m.

3. Results And Discussion

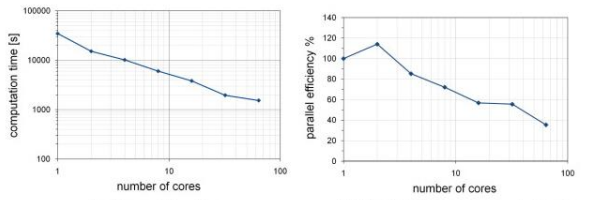
Fig. 7 shows the resulting CO2 saturation distribution for the fine grid after 25 years (injection period) and after 50 years. The simulation results fit into the range of the results given in the benchmark study [12].



(a) CO_2 plume after 25 years (b) CO_2 plume after 50 years

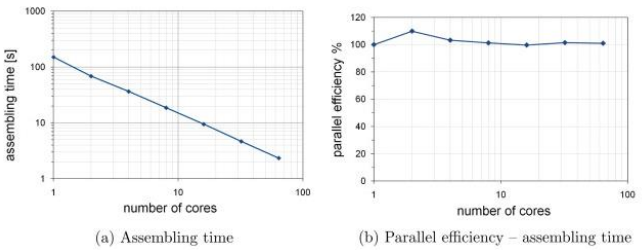
Fig. 7. CO2 saturation distribution after 25 years (end of the injection period) and after 50 years.

A more detailed description of the parallel performance is based on the coarse grid and given in Fig. 8, Fig. 9 ; Fig. 10. In Fig. 8a, the computation time for the simulation of the benchmark problem is plotted against the number of cores applied in each simulation. The computation time is decreasing for an increasing number of cores. The parallel efficiency, which is shown in Fig. 8b, is the product of the computation time and the number of cores divided by the computation time of the simulation on one core. For two cores the parallel efficiency increases above 100%. One reason for this behavior could be the fact that the performance of the applied scheme is very sensitive to the amount of available cache. For the simulation on two cores, the available cache is doubled and thus the parallel efficiency increases. However, for more than two processes, we observe that the parallel efficiency decreases for an increasing number of cores.



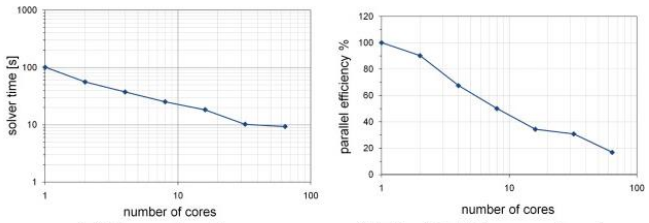
(a) Computation time (b) Parallel efficiency – computation time

Fig. 8. Computation time for the simulation of the Benchmark Problem 3.1 versus number of cores (a) and parallel efficiency (b).



(a) Assembling time (b) Parallel efficiency – assembling time

Fig. 9. Average assembling time per time step versus number of cores (a) and parallel efficiency (b).



(a) Linear solver time (b) Parallel efficiency – solver time

Fig. 10. Average time required by the linear solver per time step versus number of cores (a) and parallel efficiency (b).

In order to investigate this further, the two main parts, namely matrix assembly and linear solver, are examined separately.Fig. 9a illustrates the average time required per time step for assembling the global stiffness matrix with respect to the number of cores applied in each simulation. For an increasing number of cores the assembling time continuously decreases. The parallel efficiency with respect to assembling time is approximately constant around 100% for all simulations (Fig. 9b), thus, with respect to assembling, linear scaling is observed. This behavior can be expected, due to the facts that no communication is needed during the assembly process and that the grid is partitioned in a non-overlapping manner. Moreover, due to the computation-intensive constitutive relations, it can be assumed that comparatively few accesses to slow memory are necessary.

4. Conclution

This paper has been devoted to introduce DuMux, a free and open-source simulator for flow and transport processes in porous media. The vision, concept and design ideas have been presented. The common base of all DuMux models has been outlined, most importantly, the employed framework provided by DUNE, the Distributed and Unified Numerics Environment, as well as the flexible and extendable material system, the advanced simulation control, and the alternative to traits classes. The available models have been described, distinguishing between decoupled and fully coupled implicit approaches and briefly sketching the multi-scale and multi-physics capabilities. Two examples have been presented, one discussing the multi-scale framework in more detail and one focusing on a large scale application.

Acknowledgements

This work is supported by the National Natural Science Foundation of China（No. 11602066） and the National Science Foundation of Heilongjiang Province of China (QC2015058 and 42400621-1-15047), the Fundamental Research Funds for the Central Universities.

References

[1] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, et al. A generic grid interface for parallel and adaptive scientific computing. I. Abstract framework Computing, 82 (2–3) (2008), pp. 103–119

[2] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, et al. A generic grid interface for parallel and adaptive scientific computing. II. Implementation and tests in DUNE Computing, 82 (2–3) (2008), pp. 121–138
[3] GNU General Public License version 2. URL .
[4] The Dumux website. URL .
[5] B. Berkowitz Characterizing flow and transport in fractured geological media: A review Adv Water Resour, 25 (8–12) (2002), pp. 861–884
[6] V. Reichenberger, H. Jakobs, P. Bastian, R. Helmig A mixed-dimensional finite volume method for two-phase flow in fractured porous media Adv Water Resour, 29 (7) (2006), pp. 1020–1036
[7] Assteerawatt A. Flow and transport modelling of fractured aquifers based on geostatistical approach. PhD thesis; July 2008.
[8] K. Pruess, T.N. Narasimhan A practical method for modeling fluid and heat flow in fractured porous media Soc Petrol Eng, 25 (1) (1985), pp. 14–26
[9] Pruess K. Brief guide to the MINC-Method for modeling flow and transport in fractured media. Tech. Rep. LBNL-32195 Lawrence Berkeley National Laboratory; 1992.
[10] J.E. Pebesma, C.G. Wesseling Gstat: a program for geostatistical modelling, prediction and simulation Comput Geosci, 24 (1998), pp. 17–31
[11] Flemisch B, Wolff M. Upscaling of porous media flow with DUMUX and Dune-multidomaingrid. In: Dedner A, Flemisch B, Klöfkorn R, editors. Advances in DUNE: proceedings of the DUNE user meeting 2010. Springer; submitted for publication.
[12] H. Class, A. Ebigbo, R. Helmig, H.K. Dahle, J.M. Nordbotten, M.A. Celia, et al.A benchmark study on problems related to CO2 storage in geologic formations Comput Geosci, 13 (4) (2009), pp. 409–434