

DuMu^x: DUNE for Multi-{Phase, Component, Scale, Physics, ...} Flow and Transport in Porous Media

B. Flemisch^a, M. Darcis^a, K. Erbertseder^a, B. Faigle^a, A. Lauser^a,
K. Mosthaf^a, S. Müthing^b, P. Nuske^a, A. Tatomir^a, M. Wolff^a, R. Helmig^a

^a*Universität Stuttgart, Department of Hydromechanics and Modeling of Hydrosystems,
Pfaffenwaldring 61, 70569 Stuttgart, Germany*

^b*Universität Stuttgart, Institute for Visualization and Interactive Systems,
Universitätsstraße 38, 70569 Stuttgart, Germany*

Abstract

DuMu^x 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 DuMu^x 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

1 The quality of any type of computational modeling crucially depends on
2 the quality of the employed software framework. Research codes very often
3 fail to be developed and maintained in a continuous manner. On the contrary,
4 software development at academic institutions usually is highly fragmented
5 and driven by individual short-term needs. Furthermore, work is often done
6 redundantly, diverting resources from the original focus of research projects
7 by a need for reinventing the wheel. We are therefore convinced that the free
8 and open source (FLOSS) idea provides a chance for sustainable high quality
9 software development also in academia.
10

11 DuMu^x is a simulator for flow and transport processes in porous media. It
12 is built on top of DUNE, the Distributed and Unified Numerics Environment,
13 a modular toolbox for solving partial differential equations with grid-based

14 methods, [1, 2]. DuMu^x is licensed under the terms and conditions of the
15 GNU General Public License (GPL) version 2 or later, [3]. Stable releases
16 are available for download, [4], and anonymous read-access to the Subversion
17 repository is granted.

18 DuMu^x includes several standard models of varying complexity, ranging
19 from stationary isothermal single-phase single-component flow to transient
20 non-isothermal multi-phase compositional flow. Active research is currently
21 undertaken to include multi-scale and multi-physics concepts, as well as non-
22 standard formulations like multiple continua approaches or models involving
23 interfacial area as primary state variable. All models employ efficient nonlin-
24 ear solvers in close combination with a sophisticated time step management.
25 The capabilities of DUNE are heavily exploited to offer various spatial dis-
26 cretization schemes as well as the possibility of parallel computations. The
27 applications currently targeted by DuMu^x include fuel cells, groundwater re-
28 mediation, evaporation from partially saturated soils, CO₂ storage, and drug
29 delivery into human tissue. Several scientists from diverse areas of exper-
30 tise (computer science, engineering, mathematics) are involved in the code
31 development.

32 Various porous media simulators are under ongoing development, we list
33 some examples in the following. Most prominently, ECLIPSE is a simulation
34 tool used extensively in the oil and gas industry, [5]. Another commercial tool
35 is the Generalised Equation-of-state Model compositional reservoir simulator
36 (GEM), [6]. Finite Element Heat and Mass Transfer Simulator (FEHM) is
37 a porous media fluid flow simulator developed by the Los Alamos National
38 Laboratory, [7]. Stanford's General Purpose Research Simulator (GPRS)
39 serves as a research platform for reservoir simulation, [8, 9]. Integrated Par-
40 allel Accurate Reservoir Simulators (IPARS), developed at the University of
41 Texas, is a framework for parallel models of subsurface flow and transport
42 through porous media, [10]. OpenGeoSys is a project for the development of
43 numerical methods for the simulation of thermo-hydro-mechanical-chemical
44 processes in porous and fractured media, [11]. MUFTE-UG, the predecessor
45 of DuMu^x, is capable of solving isothermal and non-isothermal multi-phase
46 flow problems including compositional effects, [12]. The simulation code
47 TOUGH2 was developed by researchers from Lawrence Berkeley National
48 Laboratory, [13]. The MATLAB Reservoir Simulation Toolbox (MRST) is
49 developed by SINTEF Applied Mathematics and is a result of their research
50 on the development of new (multiscale) computational methodologies, [14].
51 The distribution and licensing policies for the listed simulators vary from pro-

52 proprietary commercial to open source and free of charge. From the list, only
53 MRST is released under a GPL license from the Free Software Foundation.

54 This paper is structured as follows: in the remaining part of this intro-
55 duction, the vision, concept and design ideas behind DuMu^x are presented.
56 In Section 2, the common base of all DuMu^x models is outlined. Section
57 3 describes the available models in DuMu^x. In Section 4 some examples
58 highlighting the capabilities of DuMu^x are given. We summarize and give an
59 outlook in Section 5. Concerning the notation it should be mentioned that
60 quantities that have a direct representation within the code base of DuMu^x
61 — be it a class name or the name of a folder containing a model — are set
62 in `typewriter`.

63 *1.1. Vision*

64 Up to now, DuMu^x is an academic research code and thus primarily tar-
65 geted towards researchers and particularly PhD students to code, test and
66 apply new mathematical and numerical modeling approaches. Thanks to
67 the abstraction principles employed in the DUNE framework, this can be
68 achieved without any knowledge of the underlying detailed implementations.
69 Still, a profound knowledge of advanced C++ programming techniques is re-
70 quired from the current users and developers. In the future, the capabilities
71 of DuMu^x are expected to attract end-users, who are mainly interested in
72 applying existing numerical models to their concrete problem at hand. For
73 this group of users which includes prospective bachelor and master students
74 as well as partners from industry, it will be important to offer a framework al-
75 lowing general problem descriptions and a model selection without requiring
76 in-depth programming knowledge.

77 It is evident that only one research group cannot cover all aspects of
78 the computational modeling of porous media flow and transport processes.
79 Therefore, it is important to combine the expertise from different groups,
80 and free and open source software development is the most natural way to
81 achieve a sufficiently flawless integration of individual solution components.
82 For this purpose, DuMu^x is part of the recently funded Open Porous Media
83 (OPM) initiative, [15], which is dedicated to develop a simulation suite that
84 is capable of modeling industrially and scientifically relevant flow and trans-
85 port processes in porous media and to bridge the gap between the different
86 application areas of porous media modeling.

87 *1.2. Concepts and Design Ideas*

88 Modularity is the leitmotif to design the code. DuMu^x provides shelves
89 of modularized objects, enabling the user to choose the appropriate parts
90 according to the problem at hand. This can be compared to somebody, who
91 can easily grab a different shirt out of a shelf without changing the trousers
92 and without ending up with a combination of clothes that do not match.
93 Following that design idea, DuMu^x is meant to provide a collection of shelves
94 each holding interchangeable alternatives, that are still fully interactive to
95 other shelves by using common interfaces. The user is able to select each
96 part of the implementation at each shelf through an efficient compile-time
97 **property system** (Section 2.4). Part of this modular setup (see Figure 1)
98 are the shelves from which to choose

- 99 • model concepts (Section 3),
- 100 • numerical schemes,
- 101 • control strategies for the simulation (Section 2.3),
- 102 • multitude of substances, material laws (Section 2.2),
- 103 • small and large-scale examples and applications.

104 Furthermore, the following design principles are of high importance:

- 105 1. Like DUNE, DuMu^x is coded in C++ and employs high-level generic
106 programming techniques. The design principles guiding the DUNE
107 development, see Section 2.1, are also closely followed within DuMu^x.
- 108 2. The whole project is free and open source, and fully available to the
109 public. By this, the project is open to all kinds of participation from
110 different areas of expertise, which attracts new streams of ideas to
111 improve DuMu^x.
- 112 3. A framework of this size (several 100,000 lines of code) has to be in
113 good maintenance requiring a sound infrastructure: a common reposi-
114 tory with a version control system through Subversion (SVN), [16], an
115 online bugtracking and feature request system, [17], and the regular
116 submission of build and test results to a dashboard, [18].

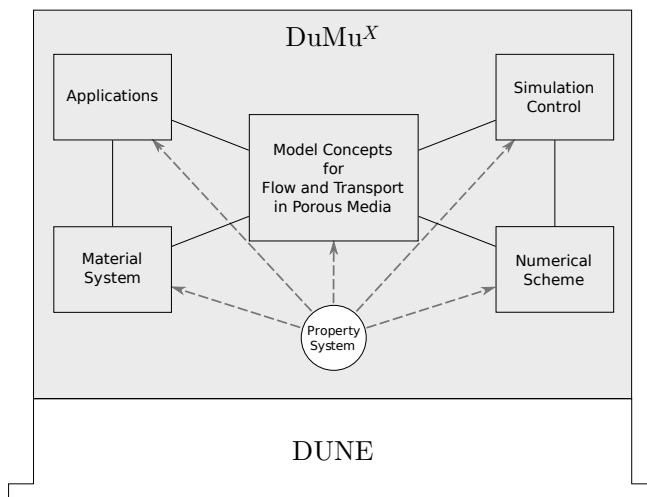


Figure 1: Modular design of DuMu^x.

117 **2. The Backbone of DuMu^x**

118 This section deals with the common structures employed by most DuMu^x
 119 models. Most prominently, these are the grid, solver and discretization in-
 120 terfaces provided by the DUNE framework described in Subsection 2.1. Of
 121 crucial importance for the ability to solve real life porous media flow prob-
 122 lems is a flexible and extendable material system, introduced in Subsection
 123 2.2, as well as a sophisticated simulation control, which is outlined in Sub-
 124 section 2.3. Finally, we propose a convenient alternative to traits classes in
 125 Subsection 2.4.

126 *2.1. DUNE*

127 DUNE, the Distributed and Unified Numerics Environment, is a modular
 128 toolbox for solving partial differential equations with grid-based methods,
 129 [1, 2]. To quote from [19]:

130 The underlying idea of DUNE is to create slim interfaces allowing
 131 an efficient use of legacy and/or new libraries. Modern C++ pro-
 132 gramming techniques enable very different implementations of the
 133 same concept (i.e. grids, solvers, ...) using a common interface
 134 at a very low overhead. Thus DUNE ensures efficiency in sci-
 135 entific computations and supports high-performance computing
 136 applications. DUNE is based on the following main principles:

- 137 • Separation of data structures and algorithms by abstract
138 interfaces.
- 139 • Efficient implementation of these interfaces using generic
140 programming techniques.
- 141 • Reuse of existing finite element packages with a large body
142 of functionality.

143 DUNE is organized as a modular system. The current release 2.0 in-
144 cludes the core modules `dune-common` (basic classes), `dune-grid` (grid in-
145 terface and implementations), `dune-istl` (iterative solver template library),
146 and `dune-localfunctions` (interface for finite element shape functions). In ad-
147 dition to these, `DuMux` also uses the DUNE external module `dune-pdelab`
148 which provides a large variety of finite element function spaces, global as-
149 sembly of residuals and operators, linear and nonlinear solvers as well as
150 explicit and implicit time discretizations based on the method of lines ap-
151 proach, [20]. Moreover, for the multi-scale and multi-physics approaches, the
152 external module `dune-multidomaingrid` is employed which supplies a meta
153 grid allowing the division of a given grid into separate sub-domains, [21].

154 The use of DUNE as basis on which `DuMux` is built on (Figure 1) offers
155 several advantages. The most important one is the ability to use a wide range
156 of different grid implementations and several linear solvers without having to
157 care about the underlying data structures of the individual implementations.
158 This particularly includes capabilities like parallelism and adaptivity, which
159 comes at minimal additional programming cost for the user. Thus, the main
160 part of the development of `DuMux` can concentrate on the implementation
161 of physical and mathematical models. The key modules of `DuMux` (Figure
162 1) are introduced in the following subsections.

163 2.2. *Material System*

164 The biggest challenges in porous media simulation are the possibly highly
165 heterogeneous distribution of parameters and the complex nonlinear material
166 laws. The `DuMux` material system constitutes a framework that allows a
167 convenient definition and usage of parameters and material laws. Due to
168 the strong interconnection of these properties, it proves difficult to achieve
169 modularity. Nevertheless, it is possible to achieve a modular structure by a
170 separation into the following parts.

171 **Components.** The term component stands for constituents of the phases
172 which can be associated with a unique chemical species, or, more generally,
173 with a group of species exploiting similar physical behavior. Each component
174 is implemented as a class consisting primarily of static member functions de-
175 scribing the physical properties of the component. This ranges from simple
176 constants like the molar mass to possibly very complex functional relation-
177 ships like the density depending on pressure and temperature.

178 **FluidSystems.** A `FluidSystem` describes the properties of the participating
179 fluid phases. This includes phase densities and viscosities as well as fugacities
180 and diffusion coefficients of components inside phases, where each phase may
181 consist of one or more components. The properties of the fluid phases usually
182 depend on their current composition which is described in a separate object
183 of type `FluidState` containing the saturation and mole fraction values. A
184 `FluidSystem` is implemented in the same way as a `Component`.

185 **FluidMatrixInteractions.** This part collects the material laws which are
186 necessary for the description of the interaction of the fluid phases with the
187 porous medium, i.e. capillarity and relative permeability. A collection of
188 standard laws is provided, including e.g. VAN GENUCHTEN and BROOKS-
189 COREY. For our extended model concepts, elaborate interfacial area - cap-
190 illary pressure - saturation - relationships are included as well as standard
191 hysteresis models. Through modular adapters, regularization schemes can
192 be imposed for extreme values. Each material law uses a set of appropriately
193 definable parameters of type `MaterialLawParams`, which may depend on the
194 location inside the domain.

195 **SpatialParameters.** This part collects all parameters that may vary de-
196 pending on the location within the porous medium. It admits a local as-
197 signment of purely intrinsic properties like porosity, permeability, or heat
198 capacity as well as of the aforementioned parameters for the material law.

199 *2.3. Simulation Control*

200 Two standard approaches for the solution of porous media problems exist:
201 a coupled fully-implicit approach and a decoupled semi-implicit approach.
202 The fully-implicit approach discretizes the original coupled balance equations
203 by an appropriate method in space and by an implicit method in time. The
204 decoupled approach manipulates the balance equations towards one equation

205 for the pressure and one or more equations for transport (of phases, com-
 206 ponents, energy, etc.), where the pressure equation is solved implicitly while
 207 the transport equations are solved explicitly. In Figure 2, the algorithmic
 208 representations of both approaches down to the element level are illustrated.

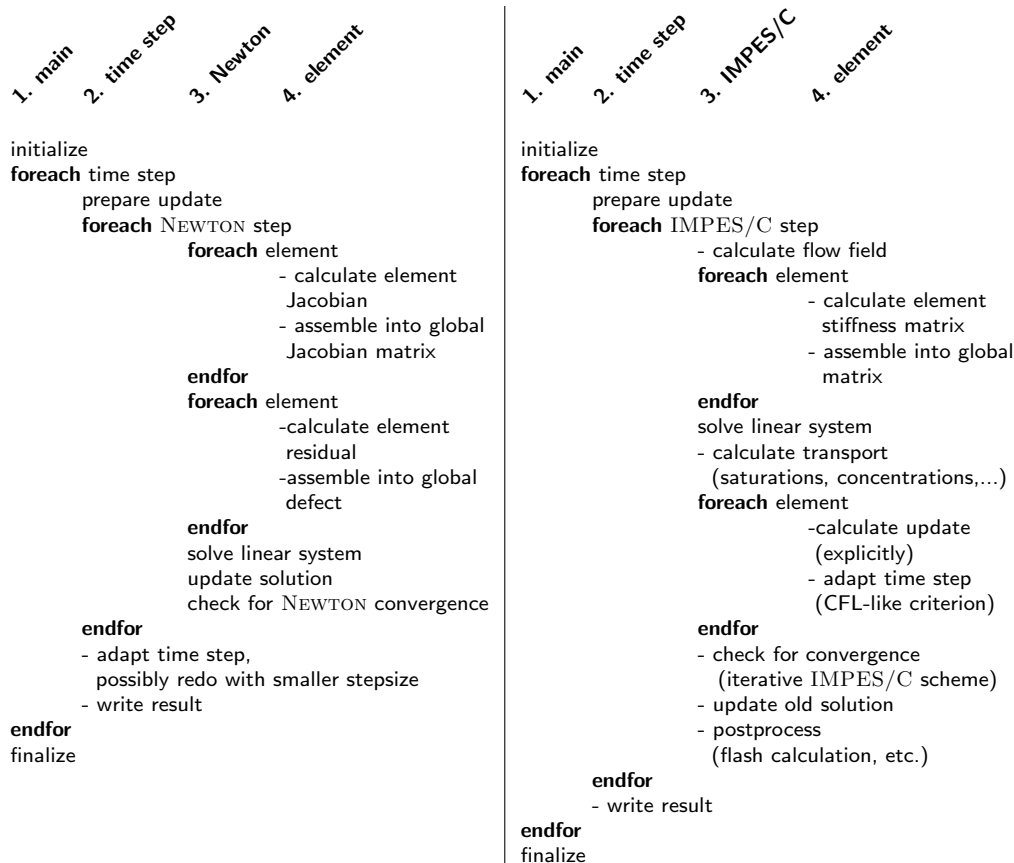


Figure 2: Structure of a coupled fully-implicit (**left**) and a decoupled semi-implicit (**right**) scheme in DuMu^x.

209 In DuMu^x, both the coupled fully-implicit and the decoupled semi-implicit
 210 models use the same code for the time-step control: The temporal domain
 211 is first divided into episodes, defined as time periods where boundary condi-
 212 tions, source terms and material parameters are differentiable with respect
 213 to time. Simulation time is then advanced by the minimum of the time-step
 214 size suggested by the underlying numerical model or the time span until the
 215 end of an episode. For the coupled fully-implicit models, the time-step size is

216 controlled based on the number of iterations required by the Newton method
217 to achieve convergence for the last time integration: The time-step size is re-
218 duced, if the number of iterations exceeds a specified threshold, whereas it is
219 increased, if the method converges within less iterations. The main influential
220 parameters are the threshold value for the Newton convergence and how it is
221 determined, and the factors for increasing and decreasing the time-step size.
222 While a default implementation for each numerical model is available, these
223 parameters can be changed problem-specific. For the decoupled models, the
224 time-step size is calculated by CFL-like criteria.

225 *2.4. Property System*

226 On the one hand, DuMu^x modules can be freely combined, on the other
227 hand, dynamic polymorphism is avoided for reasons of performance. Thus,
228 a consistent set of parameters has to be provided throughout the module
229 hierarchy at compile time. Examples for such parameters are the classes
230 including the problem description, where initial and boundary conditions are
231 typically defined on the highest level of the class hierarchy but are required
232 by the code of the low-level spatial discretization.

233 Many such parameters are typically necessary and providing all of them as
234 C++ template parameters would be very cumbersome and error-prone. One
235 option is to use traits classes. In this approach a class hierarchy is created
236 where each level stores the parameters required by the corresponding level
237 of abstraction. A fundamental problem with this approach is the inability
238 to change parameters of low levels at higher levels. This is because the
239 parameters may be defined only using those of lower levels in the traits
240 hierarchy.

241 To remedy this, the DuMu^x property system has been developed based
242 on the C++ template specialization mechanism. In this system, a hierarchy
243 of nodes – called type tags – is defined. Then all parameters are labeled and
244 attached to the appropriate nodes in this acyclic graph. The labels are called
245 property tags, whereas the parameters actually attached are called proper-
246 ties. The definition of properties may depend on arbitrary other properties
247 which may be overwritten at any higher node of the acyclic graph. The only
248 requirement for properties is that they may not exhibit cyclic dependencies.
249 This is illustrated in the following short example:

250 **Example 1.** *Assume that the type used to represent floating point values*
251 *should be parameterized as a property using the label `Scalar`. Next, assume*

252 that the property representing solution vectors is labeled **Vector**, and defined
 253 within the spatial discretization module using the **Scalar** property. In the
 254 *DuMu^x* property system, a default value for the floating point representation
 255 can be specified at the level of the spatial discretization, but it can be changed
 256 at the level of the problem definition if higher accuracy is required. In the
 257 latter case, the definition of **Vector** changes automatically.

258 3. Models

259 This section describes the models currently implemented in DuMu^x. In
 260 Table 1, an overview of the available models is given. They can be chosen
 261 – largely independent from the problem description – according to the task
 at hand. The individual models will be described in further detail here. As

Table 1: Currently available models within DuMu^x. With **p** standing for phase, **c** for component, **ni** for non-isothermal and **ia** for interfacial area.

	coupled fully-implicit	decoupled semi-implicit
<u>Standard</u>	1p, 1p2c, 2p, 2pni, 2p2c, 2p2cni, Richards	1p, 2p, 2p2c, 2p2cni, 3p3c
<u>Extended</u>	2pia, 2p2cia, 2pNc, linear-elasticity 2pDFM, 2pMINC, 1DPipe3DPorousFlow, 1p2cDoubleContinuum	
multiscale, multiphysics		

262 outlined above, we distinguish between decoupled semi-implicit and coupled
 263 fully-implicit approaches. Furthermore, a brief sketch of the multi-scale and
 264 multi-physics capabilities is given. The following nomenclature will be used
 265 from this point on: **p** stands for phase, **c** for component, **ni** for non-isothermal
 266 and **ia** for interfacial area.
 267

268 *3.1. Decoupled Semi-implicit Numerical Models*

269 As already described above, decoupled models solve a system of equations
270 in which the single equations are only weakly coupled with each other. Thus,
271 a sequential solution strategy can be applied where the standard scheme for
272 multi-phase flow in porous media is an IMPES/IMPEC algorithm (IMplicit
273 Pressure Explicit Saturation/Concentration). IMPES/IMPEC schemes first
274 implicitly solve a pressure equation to get the flow field. Afterwards, the
275 transport equations can be solved explicitly in the simplest case by using an
276 explicit Euler scheme.

277 **1p.** This model solves an elliptic pressure equation in a fully-saturated porous
278 medium. Available discretization methods are: Cell-centered finite volumes
279 with TPFA (Two-Point-Flux-Approximation), or with MPFA (Multi-Point-
280 Flux-Approximation, [22]), and mimetic finite differences, [23].

281 **2p.** Formulations for immiscible isothermal two-phase flow are implemented
282 as classical fractional flow formulation (incompressible, e.g. [24]) or decou-
283 pled phase pressure formulation (slightly compressible/incompressible, see
284 e.g. [25]). Transport equations for saturation are discretized by cell-centered
285 finite volumes, for the solution of the pressure equations various discretiza-
286 tion methods (see **1p** model) are available. An IMPES strategy is applied.

287 **2p2c/ni.** Miscible compressible two-phase flow can be modeled isothermally
288 (**2p2c**) as well as non-isothermally (**2p2cni**). Capillary pressure is at the
289 stage of this publication neglected. Instead of saturation equations, transport
290 equations for concentrations of the components (**2c** - two components) are
291 solved and flash calculations are performed afterwards to determine the phase
292 composition. Details on the formulation can be found in [26]. Both pressure
293 and transport equations for concentrations are discretized by cell-centered
294 finite volumes, and an IMPEC scheme is used for the solution of the system.

295 **3p3c.** This model for three-phase three-component flow and transport is sim-
296 ilar to the **2p2c** model.

297 *3.2. Coupled Fully-implicit Numerical Models*

298 We provide brief descriptions of standard and extended models treated
299 in a coupled fully-implicit manner.

300 *Standard models.* The standard coupled models can be subdivided into sev-
301 eral groups. It can be distinguished between the phase-based formulations
302 (1p, 2p and 2pni) and the component-based models (1p2c, 2p2c, 2p2cni).
303 Further, thermal effects can be taken into account (non-isothermal models:
304 2pni, 2p2cni). In general, a modular structure is used and the models with
305 a higher complexity inherit the functionality from the simpler ones and ext-
306 tend it accordingly. As spatial discretization, a vertex-centered finite volume
307 scheme (box method, [27]) is used. So far, the implicit Euler scheme is ap-
308 plied for the temporal discretization.

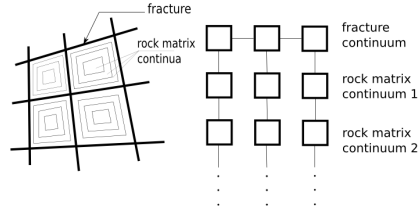
309 **interfacial area.** Usually, flow and transport in the bulk phases is mod-
310 eled using the assumption of local thermodynamic equilibrium. The inter-
311 facial area models (abbreviated 2pia and 2p2cia) extend this approach by
312 incorporating flow and evolution of the area separating the fluid phases [28].
313 This enables the description of kinetic mass and energy transfer between the
314 fluid phases (thermal and chemical non-equilibrium). A natural description
315 of hysteresis in the capillary pressure – saturation relationship is envisaged.

316 **linear-elasticity.** In the case of large pressure gradients the assumption
317 of a rigid porous medium might be violated. The linear-elasticity models ac-
318 count for the deformation of the solid matrix and its interaction with a single-
319 or two-phase flow system based on the theory of [29]. A first application of
320 the DuMu^x single-phase linear-elasticity model is described in [30].

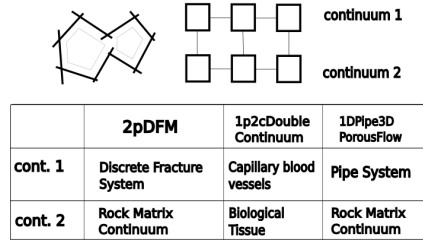
321 **2pNc.** In general it is desirable not to write a new model whenever an addi-
322 tional component is to be modeled. This, as well as the general prevention
323 of switches in the primary variables – in the case of one phase disappearing
324 – are the distinctions of the 2Nc model, with Nc standing for N-number of
325 components .

326 **2pDFM.** The model simulates the two-phase flow in fractured porous systems
327 using a discrete fracture model (DFM) approach [31], with a lower dimen-
328 sional representation for the fractures [32]. The representative fracture net-
329 work of the DFM is reconstructed with a geostatistic fracture generator [33].

330 **2pMINC.** In contrast to the 2pDFM, this model does not discretize the fractures
331 but simulates the two-phase flow in fractured porous media using the MINC
332 method [34], [35].The fractures are treated as an equivalent homogeneous



(a) MINC discretization of a fractured porous system



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

Figure 3: Sketch of the relation between model concepts and the continuum representation. Schematic diagrams of connectivity for the continuum models

333 porous medium which requires the determination of appropriate effective pa-
 334 rameters and transfer functions between continua, but reduces considerably
 335 the geometrical complexity of the problem, see Figure 3a.

336 **1p2cDoubleContinuum**. It models the flow, transport and reaction processes
 337 through the terminal vascular bed (capillary vessels) and the surrounding
 338 biological tissues as an application to pulmonary cancer therapy. This is
 339 done by using the 1p2c model concept, where both the capillary vessels and
 340 the surrounding tissue are described as two interacting porous media.

341 **1DPipe3DPorousFlow**. It simulates a coupled flow and transport system com-
 342 posed of quasi one-dimensional hollow structures embedded into a 3D porous
 343 medium [36]. The hollow structures are expressed with the cross-sectionally
 344 averaged one-dimensional pipe flow equation, like Hagen-Poiseuille, whereas
 345 for the flow within the porous medium the Darcy law is used.

346 The **2pDFM**, **1p2cDoubleContinuum**, and **1DPipe3DPorousFlow** models
 347 use the coupling strategy of a standard double-continuum approach, that
 348 is, the exchange terms between the two continua are implemented by addi-
 349 tional source/sink terms, see Figure 3b.

350 3.3. Multi-Scale and Multi-Physics Models

351 The modular concept of DuMu^x allows the combination of the single mod-
 352 els described before in multi-scale and/or multi-physics concepts. Multi-scale
 353 methods combine simulations on different length or time scales dependent on
 354 the processes that are to be modeled and on the information available. In

355 DuMu^x, this can be done in the context of classical numerical upscaling (or
356 downscaling) methods (e.g. permeability upscaling) or in combination with
357 multi-physics strategies. In space, multi-physics can be volume coupling (see
358 e.g. multi-continuum models) or surface coupling of different model domains
359 depending on the occurring processes that dominate in a sub-domain. In
360 time, it could be a sequential solution strategy, where the model type is
361 switched if the dominating processes change. The idea of both multi-scale
362 and multi-physics methods is to be able to model complex processes occurring
363 in large model domains. Examples on the capabilities of DuMu^x concerning
364 multi-scale and multi-physics ideas can be found in Section 4.

365 4. Selected Examples

366 We present two examples that demonstrate the capabilities of DuMu^x
367 for applications, relevant for the simulation of multi-phase flow in porous
368 media. The multi-scale example presents an approach allowing the coarse-
369 scale description of flow and transport in a fine-scale heterogeneous domain.
370 This is accomplished by exploiting the capabilities of DUNE and DuMu^x for
371 handling grids on multiple scales and solving small flow problems in order to
372 incorporate the influence of sub-scale heterogeneities on the coarse scale.

373 Many real life applications require large modeling domains on the one
374 hand and high model complexity on the other hand. Typically, this results
375 in computationally demanding tasks. Therefore, the second example focuses
376 mainly on the parallelization of code, which comes at little additional effort to
377 the programmer. Furthermore, it is shown that the single models (described
378 in Section 3) can be applied sequentially.

379 4.1. Multi-Scale

380 The multi-scale framework of DuMu^x that is illustrated in the following is
381 a general implementation for local-global upscaling (downscaling) methods.
382 It allows to carry out local fine-scale simulations – numerical experiments –
383 to estimate effective parameters, which can be directly used in global coarse-
384 scale simulations. In the following part the term fine-scale is used for the
385 scale of the finest heterogeneities that are accounted for (*Darcy-scale* on
386 which the previously described models (Section 3) are valid), whereas the
387 term coarse-scale indicates the scale to which an upscaling is applied.

388 One basic idea is that any of the fine-scale models which are available
389 in DuMu^x can be chosen for the local calculations without any changes in

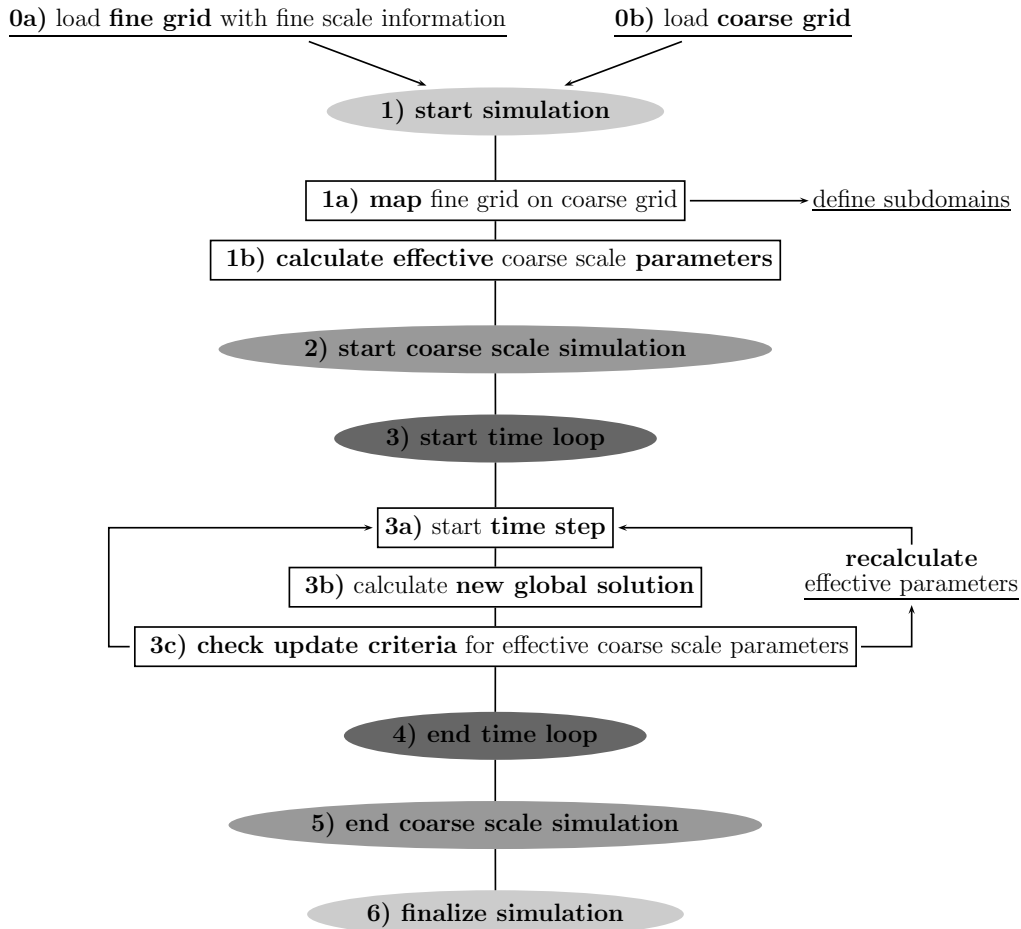


Figure 4: Flow of a multi-scale simulation in DuMu^x.

390 these model implementations. A second idea is that different kinds of post-
 391 processing routines which calculate the effective parameters are implemented
 392 once, and can be easily reused as well as combined in different kinds of
 393 local(-global) upscaling methods. Thirdly, depending on the upscaling ap-
 394 proach one of the fine-scale model implementations could also be used on the
 395 coarse scale (with different parameters) or new coarse-scale models can be
 396 implemented, which again might be combined respectively with the differ-
 397 ent fine-scale and post-processing methods. Figure 4 shows the general flow
 398 of a local(-global) multi-scale simulation in DuMu^x. In the following para-
 399 graphs, we describe the conceptual and the mathematical model and show

400 some results.

401 *Conceptual Model.* The multi-scale approach shown here is a local intrinsic
 402 permeability upscaling method (see e.g. [37, 38]), where a heterogeneous
 403 fine-scale permeability field is assumed to be known. It divides the global
 404 model domain into sub-domains assigned to coarse grid blocks (Figure 4, 1a)).
 405 The grids for the single sub-domains are managed using the DUNE module
 406 `dune-multidomaingrid` [21]. Effective coarse-scale intrinsic permeabilities
 407 are calculated for every coarse grid block using the results of local fine-scale
 408 flow simulations carried out on each of the sub-domains. As the fine-scale
 409 intrinsic permeability field is not dependent on saturation or pressure if the
 410 porous matrix is rigid, it is possible to calculate effective permeabilities that
 411 are also independent of the coarse-scale pressure and saturation field. This
 412 can be done in a preprocessing step (Figure 4, 1b)). After the preprocessing
 413 the global coarse-scale simulation can be directly started using the newly
 414 calculated effective intrinsic permeabilities (Figure 4, 2)). If an adaptive
 415 local-global approach is used in which effective permeabilities are updated
 416 depending on the flow field, a criterion which triggers the recalculation has
 417 to be checked at the end of every time step (Figure 4, 2c)). In this exam-
 418 ple a full tensor effective intrinsic permeability is calculated only once in a
 419 preprocessing step.

420 *Mathematical Model.* An isothermal incompressible two-phase flow model is
 421 used in a decoupled formulation (decoupled - 2p) and in the simplest form,
 422 neglecting capillary pressure as well as gravity. The following equations have
 423 to be solved:

$$\nabla \cdot (-\lambda_t \mathbf{K} \text{grad } p_w) = \sum_{\alpha} q_{\alpha}, \quad \alpha \in \{\text{w}, \text{n}\}, \quad (1)$$

424

$$\phi \frac{\partial (S_w)}{\partial t} + \text{div} (-\lambda_w \mathbf{K} \text{grad } p_w) = q_w, \quad (2)$$

425 where \mathbf{K} is the intrinsic permeability, ϕ is the porosity, λ_{α} is the mobility of
 426 phase α and $\lambda_t = \sum_{\alpha} \lambda_{\alpha}$ is the total mobility, p_w is the pressure of the wetting
 427 phase, S_w is the saturation of the wetting phase, and q_{α} source or sink term
 428 of phase α . Equation 1 is called pressure equation and Equation 2 is called
 429 saturation equation. The same set of equations is used for both, fine scale
 430 and coarse scale, while effective (upscaled) coefficients are used on the coarse
 431 scale (e. g. $\mathbf{K} = \mathbf{K}_{\text{eff}}$, see [38]).

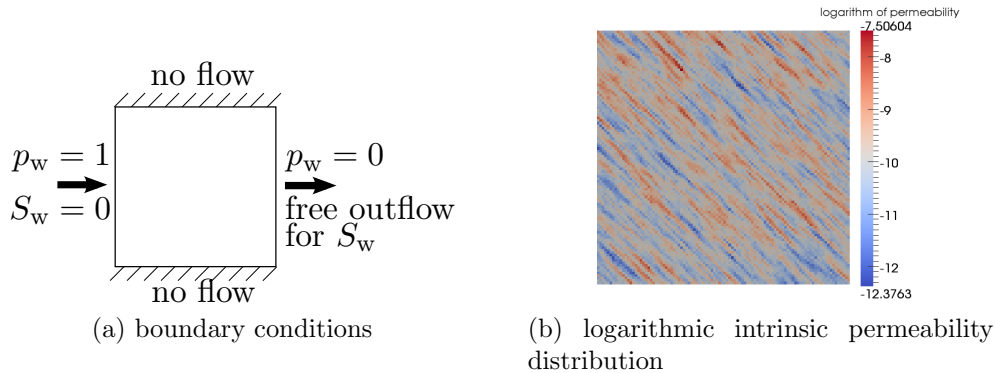


Figure 5: Setup of the multi-scale simulation example.

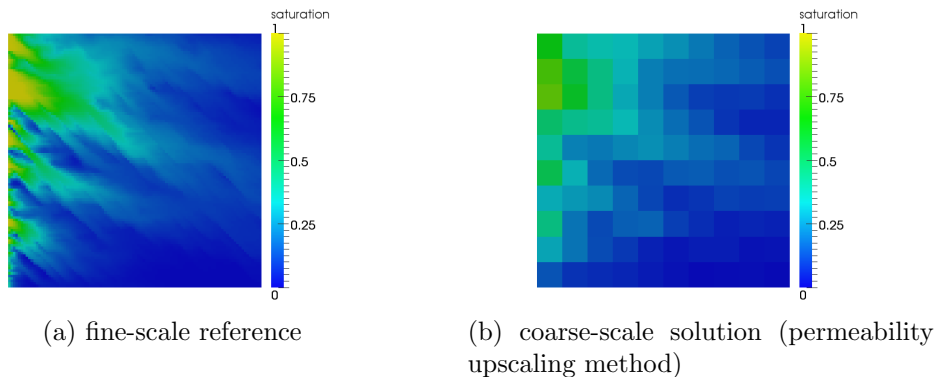


Figure 6: Saturation distribution at $t = t_{end}$.

432 *Results.* The models of this example are discretized by a cell-centered finite
 433 volume method. To account for the full tensor permeabilities a MPFA o-
 434 m-method is used (see [22], [39]) for discretization of the pressure equation.
 435 The setup of the simple test-example (2-D) is shown in Figure 5a. The
 436 heterogeneous permeability field (Figure 5b) is randomly generated by the
 437 open source tool GSTAT [40], which can be used for geostatistical modeling.
 438 The result of the coarse-scale simulation is shown in Figure 6b and shows
 439 good agreement with the fine-scale reference solution depicted in Figure 6a.
 440 Even for the quite simple and small test problem a significant speed-up of
 441 the multi-scale model ($>$ factor 40) can be achieved.

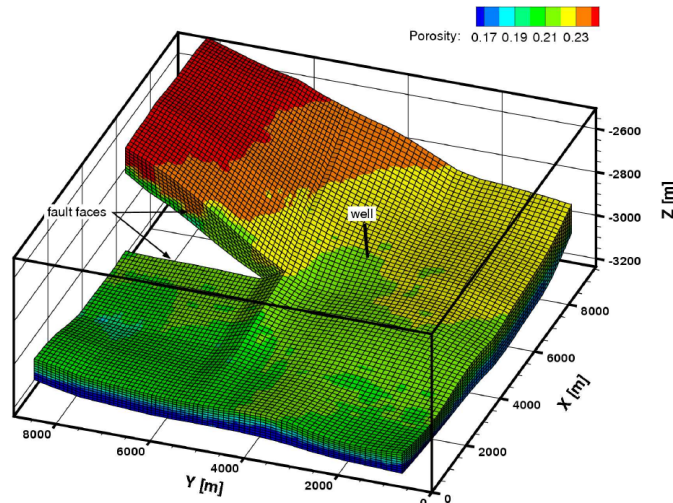


Figure 7: Geometry and porosity distribution of the Benchmark Problem 3.1 given in [41].

442 *4.2. CO₂ Storage – A Large Scale Example*

443 As a large scale application example, the injection of CO₂ into a het-
 444 erogeneous saline aquifer is modeled with DuMu^x. The geometry and the
 445 hydraulic parameters are chosen according to the Benchmark Problem 3.1
 446 given in [41]. The model domain and the position of the injection well are
 447 shown in Figure 7. CO₂ is injected for 25 years with a rate of 15 kg/s over
 448 the bottom 50 m of the vertical injection well. After the injection stop, the
 449 migration of the CO₂ plume is modeled for another 25 years. We present
 450 the conceptual and the mathematical model as well as some results in the
 451 following paragraphs.

452 *Conceptual Model.* In order to reduce the computational effort, the simula-
 453 tion of the injection and post-injection period is performed with a sequentially
 454 coupled model. Sequential model coupling can be applied if the dominat-
 455 ing physical processes change over time. With respect to CO₂ storage it is
 456 possible to distinguish such different time periods. The injection period is
 457 governed by viscous and buoyancy driven multi-phase flow processes, which
 458 are hardly influenced by compositional processes such as CO₂ dissolution,
 459 diffusion and density-driven convection. After injection stop, the composi-
 460 tional effects become increasingly important and finally dominate the flow
 461 processes.

462 The sequential model applied here consists of two standard coupled fully-
 463 implicit models (Section 3.2). The injection period is simulated with a two-
 464 phase model (2pni) and the post-injection period is simulated with a more
 465 complex two-phase, two-component model (2p2cni). Both models include
 466 an energy balance equation to take into account non-isothermal effects e.g.
 467 the cooling caused by a CO₂ injection temperature, which is often below the
 468 reservoir temperature, or the reestablishment of the geothermal temperature
 469 gradient after injection stop by heat conduction.

470 Due to the modular structure of DuMu^x the setup of a sequential model
 471 only requires the implementation of a suitable interface which allows a rea-
 472 sonable transfer of the primary variables between the coupled models. For
 473 more details on sequential model coupling in general and its application for
 474 the simulation of CO₂ storage, the reader is referred to [42], [43].

475 *Mathematical Model.* In the model of the first time period (2pni), the fol-
 476 lowing mass balance equation is solved for both the wetting brine phase w
 477 and the non-wetting CO₂ phase n,

$$\phi \frac{\partial(\varrho_\alpha S_\alpha)}{\partial t} - \text{div} \{ \varrho_\alpha \lambda_\alpha \mathbf{K} (\text{grad } p_\alpha - \varrho_\alpha \mathbf{g}) \} = q_\alpha, \quad \alpha \in \{w, n\}. \quad (3)$$

478 Here, ϱ_α represents the phase densities. Additionally, the energy balance
 479 equation of the fluid-solid mixture is solved to describe the non-isothermal
 480 processes. Assuming local thermal equilibrium, the energy balance equation
 481 can be written as

$$\begin{aligned} & \phi \frac{\partial(\sum_\alpha \varrho_\alpha u_\alpha S_\alpha)}{\partial t} + (1 - \phi) \frac{\partial(\varrho_s c_s T)}{\partial t} \\ & - \sum_\alpha \text{div} \{ \varrho_\alpha h_\alpha \lambda_\alpha \mathbf{K} (\text{grad } p_\alpha - \varrho_\alpha \mathbf{g}) \} - \text{div}(\lambda_{\text{pm}} \text{grad } T) = q_h, \end{aligned} \quad (4)$$

482 including a summation over the phases $\alpha \in \{w, n\}$. T is the temperature, u_α
 483 and h_α stand for the internal energy and the enthalpy of the fluid phases. The
 484 density ϱ_s and the heat capacity c_s are properties of the rock matrix and λ_{pm}
 485 is the saturation-dependent thermal conductivity of the porous medium. To
 486 close this system of equations, the following auxiliary conditions are applied:

$$\sum_\alpha S_\alpha = 1 \quad \text{and} \quad p_w = p_n - p_c(S_w), \quad (5)$$

487 with the primary drainage capillary pressure p_c .

488 The model of the second time period (**2p2cni** model) solves composi-
 489 tional mass balances for the components CO₂ and brine. Brine is a pseudo-
 490 component and represents water with a certain salinity. The component mass
 491 balances are described by

$$\begin{aligned} & \phi \frac{\partial(\sum_{\alpha} \varrho_{\alpha} X_{\alpha}^{\kappa} S_{\alpha})}{\partial t} - \sum_{\alpha} \operatorname{div} \{ \varrho_{\alpha} X_{\alpha}^{\kappa} \lambda_{\alpha} \mathbf{K}(\operatorname{grad} p_{\alpha} - \varrho_{\alpha} \mathbf{g}) \} \\ & - \sum_{\alpha} \operatorname{div} \{ D_{\alpha, \text{pm}}^{\kappa} \varrho_{\alpha} \operatorname{grad} X_{\alpha}^{\kappa} \} = \sum_{\alpha} q_{\alpha}^{\kappa} \quad \kappa \in \{\text{brine}, \text{CO}_2\}. \end{aligned} \quad (6)$$

492 Here, X_{α}^{κ} is the mass fraction of the component κ , $D_{\alpha, \text{pm}}^{\kappa}$ is the porous
 493 medium diffusion coefficient and q_{α}^{κ} is the source or sink term of the compo-
 494 nent κ in the phase α . The energy equation of the **2p2cni** model is similar to
 495 Equation (5), but the phase enthalpies and internal energies are functions of
 496 the dissolved components in addition to pressure and temperature. Besides
 497 the equations given in (5), a further auxiliary condition needs to be fulfilled
 498 for each phase in the **2p2cni** model:

$$\sum_{\kappa} X_{\alpha}^{\kappa} = 1. \quad (7)$$

499 The fluid properties of CO₂ are calculated as functions of pressure and tem-
 500 perature. The properties of brine additionally depend on the salinity, and in
 501 the compositional model, on the CO₂ mass fraction. The mutual solubilities
 502 of water and CO₂ are calculated according to [44]. For further information
 503 on the fluid property functions, the reader is referred to [45].

504 *Results.* The simulation is performed on a grid with 469,813 vertices lead-
 505 ing to 1,409,439 degrees of freedom. For the parallelization of the DuMu^x
 506 models, almost no additional implementation work is required, since DUNE
 507 provides arbitrary data decomposition in a generic way and the employed
 508 assembly operator from `dune-pdelab` and linear solvers from `dune-istl` are
 509 designed correspondingly. To test the parallel scaling, several simulations
 510 with a differing number of cores are carried out.

511 Figure 8 shows the resulting CO₂ saturation distribution after 25 years
 512 (injection period) and after 50 years. The simulation results fit into the
 513 range of the results given in the benchmark study [41]. In Figure 9a, the
 514 computation time for the simulation of the benchmark problem is plotted
 515 against the number of cores applied in each simulation. The computation

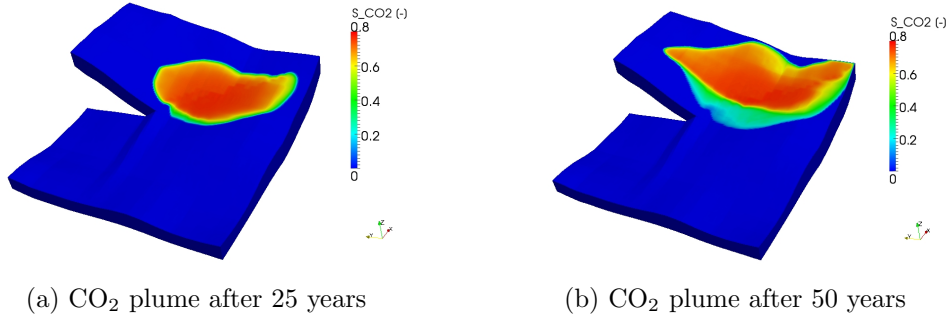


Figure 8: CO₂ saturation distribution after 25 years (end of the injection period) and after 50 years.

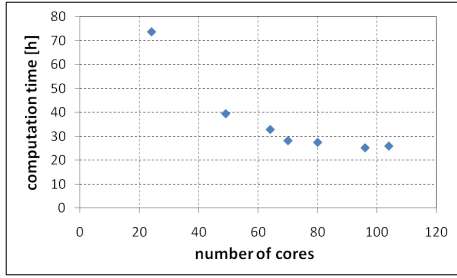
516 time is decreasing for an increasing number of cores up to a value of 96
 517 cores. For a larger number the computation time slightly increases, which
 518 is due to a sub-optimal linear solver applied in the current implementation.
 519 In particular, a BiCGSTAB solver is used, preconditioned by an additive
 520 Schwarz method consisting of ILU applications locally on each process. No
 521 coarse grid correction is used and a linear parallel speed-up is not possible.
 522 This can also be seen by the increasing total CPU time which is plotted in
 523 Figure 9b.

524 In order to investigate this further, Figure 10a illustrates the average time
 525 required for assembling the global stiffness matrix with respect to the number
 526 of cores applied in each simulation. For an increasing number of cores the
 527 assembling time continuously decreases. The product of assembling time and
 528 number of cores is approximately constant for all simulations (Figure 10b),
 529 thus, with respect to assembling, linear scaling is observed.

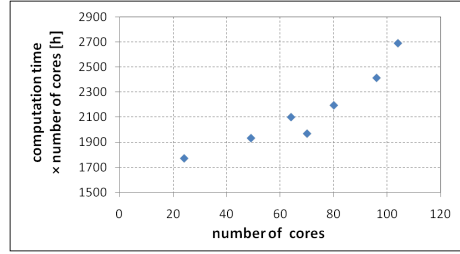
530 As already mentioned, the average time required for the solution of the
 531 global stiffness matrix by the iterative solver is not scaling linearly with
 532 the number of cores (Figure 11). An algebraic multigrid solver for non-
 533 overlapping grids is currently developed within DUNE and will be available
 534 for parallel simulations with DuMu^x in the future.

535 5. Summary and Outlook

536 This paper has been devoted to introduce DuMu^x, a free and open-source
 537 simulator for flow and transport processes in porous media. The vision, con-
 538 cept and design ideas have been presented. The common base of all DuMu^x

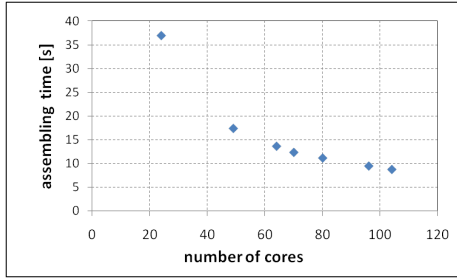


(a) Computation time

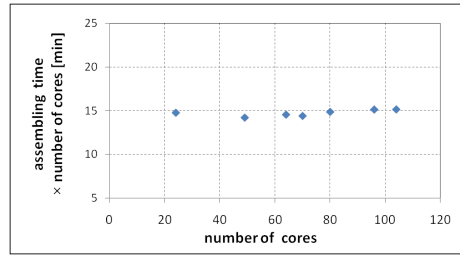


(b) Total CPU hours

Figure 9: Computation time for the simulation of the Benchmark Problem 3.1 versus number of cores (a) and total CPU hours (computation time \times number of cores) versus number of cores (b).

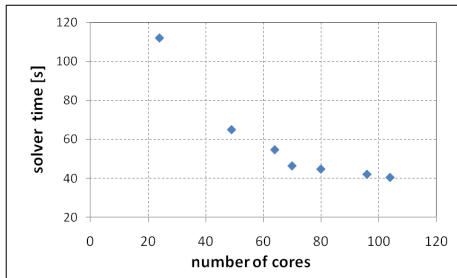


(a) Assembling time

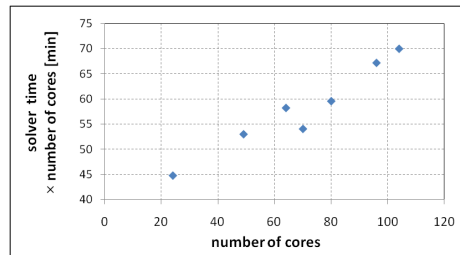


(b) Total CPU assembling time

Figure 10: Average assembling time per nonlinear iteration versus number of cores in seconds (a) and total CPU assembling time (assembling time \times number of cores) versus number of cores (b).



(a) Linear solver time



(b) Total linear solver time

Figure 11: Average time required by the linear solver per nonlinear iteration versus number of cores (a) and total CPU time required by the linear solver (linear solver time \times number of cores) versus number of cores (b).

539 models has been outlined, most importantly, the employed framework pro-
540 vided by DUNE, the Distributed and Unified Numerics Environment, as well
541 as the flexible and extendable material system, the advanced simulation con-
542 trol, and the alternative to traits classes. The available models have been
543 described, distinguishing between decoupled and fully coupled implicit ap-
544 proaches and briefly sketching the multi-scale and multi-physics capabilities.
545 Two examples have been presented, one discussing the multi-scale framework
546 in more detail and one focussing on a large scale application.

547 In the future, the modeling capabilities will be further extended. In-
548 creased emphasis will be devoted to the efficiency and robustness of the im-
549 plemented numerical models. An enhanced user-friendliness will attract more
550 users from outside of the developers group. Within the Open Porous Media
551 (OPM) initiative, DuMu^x will become part of a simulation suite integrating
552 the expertise of several academic and industrial partners.

553 *Acknowledgments.* This work was partially funded by the German research
554 foundation (DFG) and the German Federal Ministry of Education and Re-
555 search (BMBF). Some of the authors are members of the DFG International
556 Research Unit MUSIS (FOR 1083), the Cluster of Excellence SimTech and of
557 the International Research Training Group NUPUS, all funded by the Ger-
558 man Research Foundation (DFG). Others are members of the CO2-MoPa
559 joint project, BMBF's IPSWaT programme and of the FORSYS-Partner
560 project. We thank the DFG, the BMBF and the partners from industry for
561 their support.

562 **References**

- 563 [1] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger,
564 O. Sander, A generic grid interface for parallel and adaptive scientific
565 computing. I. Abstract framework, Computing 82 (2-3) (2008) 103–119.
566 doi:10.1007/s00607-008-0003-x.
- 567 [2] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhu-
568 ber, M. Ohlberger, O. Sander, A generic grid interface for parallel and
569 adaptive scientific computing. II. Implementation and tests in DUNE,
570 Computing 82 (2-3) (2008) 121–138. doi:10.1007/s00607-008-0004-9.
- 571 [3] GNU General Public License version 2.
572 URL <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

- 573 [4] The Dumux website.
574 URL <http://dumux.org/>
- 575 [5] Schlumberger, Eclipse technical description, 2007.
- 576 [6] Computer Modelling Group, GEM user guide.
577 URL http://www.cmgroup.com/software/brochures/GEM_FactSheet.pdf
- 578 [7] B. Robinson, H. Viswanathan, A. Valocchi, Efficient numerical tech-
579 niques for modeling multi-component groundwater transport based upon
580 simultaneous solution of strongly coupled subsets of chemical compo-
581 nents, *Adv. Water Res.* 23 (2000) 307–324.
- 582 [8] H. Cao, Development of techniques for general purpose simulators,
583 Ph.D. thesis, Stanford University (2002).
- 584 [9] Y. Jiang, Techniques for modeling complex reservoirs and advanced
585 wells, Ph.D. thesis, Stanford University (2007).
- 586 [10] J. Wheeler, M. Wheeler, Integrated parallel and accurate reservoir sim-
587 ulator, Tech. rep., TICAM01-25, CSM, University of Texas at Austin
588 (2001).
- 589 [11] The OpenGeoSys project.
590 URL http://www.ufz.de/data/OGS-5_concept_V110981.pdf
- 591 [12] A. Assteerawatt, P. Bastian, A. Bielinski, T. Breiting, H. Class,
592 A. Ebigbo, H. Eichel, S. Freiboth, R. Helmig, A. Kopp, J. Niessner,
593 S. Ochs, A. Papafotiou, M. Paul, H. Sheta, D. Werner, U. Ölmann,
594 MUFTE-UG: structure, applications and numerical methods, Newslet-
595 ter, International Groundwater Modeling Centre, Colorado School of
596 Mines 23(2), 2005.
- 597 [13] K. Pruess, The TOUGH codes – a family of simulation tools for multi-
598 phase flow and transport processes in permeable media, *Vadose Zone J.*
599 3 (2004) 738–746.
- 600 [14] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen,
601 B. Skaflestad, Discretisation on complex grids – open source matlab im-
602 plementation, in: *Proceedings of ECMOR XII*, Oxford, UK, 6–9 Septem-
603 ber 2010, 2010.

- 604 [15] Open Porous Media Initiative.
605 URL <http://www.sintef.no/Projectweb/GeoScale/Simulators/OPM/>
- 606 [16] Apache Subversion.
607 URL <http://subversion.apache.org/>
- 608 [17] Flyspray bugtracking system.
609 URL <http://flyspray.org/>
- 610 [18] Cdash software testing server.
611 URL <http://www.cdash.org/>
- 612 [19] The DUNE website.
613 URL <http://www.dune-project.org/>
- 614 [20] P. Bastian, F. Heimann, S. Marnach, Generic implementation of finite
615 element methods in the distributed and unified numerics environment
616 DUNE, *Kybernetika* 46 (2) (2010) 294–315.
- 617 [21] S. Müthing, dune-multidomaingrid.
618 URL <http://gitorious.org/dune-multidomaingrid>
- 619 [22] I. Aavatsmark, An introduction to multipoint flux approximations for
620 quadrilateral grids, *Computational Geosciences* 6 (2002) 405 – 432.
621 doi:10.1023/A:1021291114475.
- 622 [23] B. Flemisch, R. Helmig, Numerical investigation of a mimetic finite dif-
623 ference method, in: R. Eymard, J. Hérard (Eds.), *Finite volumes for
624 complex applications V - problems and perspectives*, Wiley - VCH, 2008,
625 pp. 815–824.
- 626 [24] P. Binning, M. A. Celia, Practical implementation of the fractional flow
627 approach to multi-phase flow simulation, *Advances in Water Resources*
628 22 (5) (1999) 461–478. doi:10.1016/S0309-1708(98)00022-0.
- 629 [25] Z. Chen, G. Huan, Y. Ma, *Computational Methods for Multiphase Flows
630 in Porous Media*, SIAM, Computational Science & Engineering, 2006.
- 631 [26] J. Fritz, B. Flemisch, R. Helmig, Multiphysics modeling of advection-
632 dominated two-phase compositional flow in porous media, *International
633 Journal of Numerical Analysis & Modeling*. To appear 2010.

- 634 [27] R. Huber, R. Helmig, Node-centered finite volume discretizations for the
635 numerical simulation of multiphase flow in heterogeneous porous media,
636 *Comput. Geosci.* 4 (2000) 141–164.
- 637 [28] J. Niessner, S. M. Hassanizadeh, Modeling kinetic interphase mass trans-
638 fer for two-phase flow in porous media including fluid-fluid interfacial
639 area, *Transport in Porous Media* doi:10.1007/s11242-009-9.
- 640 [29] M. A. Biot, Theory of elasticity and consolidation for a porous
641 anisotropic solid, *Journal of Applied Physics* 25 (1955) 182–185.
- 642 [30] K. H. Støverud, S. M. Hassanizadeh, R. Helmig, M. Darcis, Modeling
643 convection-enhanced drug delivery into brain tissue including informa-
644 tion from magnetic resonance imaging, submitted to: *Biomechanics and*
645 *Modeling in Mechanobiology*.
- 646 [31] B. Berkowitz, Characterizing flow and transport in fractured geological
647 media: A review, *Advances in Water Resources* 25 (8-12) (2002) 861–
648 884. doi:10.1016/S0309-1708(02)00042-8.
- 649 [32] V. Reichenberger, H. Jakobs, P. Bastian, R. Helmig, A mixed-
650 dimensional finite volume method for two-phase flow in fractured
651 porous media, *Advances in Water Resources* 29 (7) (2006) 1020–1036.
652 doi:10.1016/j.advwatres.2005.09.001.
- 653 [33] A. Assteerawatt, Flow and transport modelling of fractured aquifers
654 based on geostatistical approach, Ph.D. thesis (Jul. 2008).
- 655 [34] K. Pruess, T. N. Narasimhan, A practical method for modeling fluid and
656 heat flow in fractured porous media, Society of Petroleum Engineers.
- 657 [35] K. Pruess, Brief guide to the MINC-Method for modeling flow and trans-
658 port in fractured media, Tech. Rep. LBNL-32195 Lawrence Berkeley
659 National Laboratory (1992).
- 660 [36] O. Dogan, H. Class, R. Helmig, Different concepts for the coupling of
661 porous-media flow with lower-dimensional pipe flow, *Computer Model-
662 ing in Engineering & Sciences* 53 (3) (2009) 207–233.
- 663 [37] L. J. Durlofsky, Numerical calculation of equivalent grid block perme-
664 ability tensors for heterogeneous porous media, *Water Resources Re-
665 search* 27 (5) (1991) 699–708.

- 666 [38] X. H. Wen, L. J. Durlafsky, M. G. Edwards, Use of border regions for
667 improved permeability upscaling, *Mathematical Geology* 35 (5) (2003)
668 521–547. doi:10.1023/A:1026230617943.
- 669 [39] G. T. Eigestad, R. A. Klausen, On the convergence of the multi-point
670 flux approximation o-method: Numerical experiments for discontinuous
671 permeability., *Numerical methods for partial differential equations* 21 (6)
672 (2005) 1079–1098. doi:10.1002/num.20079.
- 673 [40] J. E. Pebesma, C. G. Wesseling, Gstat: a program for geostatistical mod-
674 elling, prediction and simulation., *Computers & Geosciences* 24 (1998)
675 17–31.
- 676 [41] H. Class, A. Ebigbo, R. Helmig, H. K. Dahle, J. M. Nordbotten, M. A.
677 Celia, P. Audigane, M. Darcis, J. Ennis-King, Y. Fan, B. Flemisch,
678 S. E. Gasda, M. Jin, S. Krug, D. Labregere, B. A. Naderi, R. J. Pawar,
679 A. Sbai, S. G. Thomas, L. Trenty, L. Wei, A benchmark study on prob-
680 lems related to CO₂ storage in geologic formations, *Computational geo-*
681 *sciences* 13 (4) (2009) 409–434.
- 682 [42] H. Class, R. Helmig, I. Neuweiler, Sequential coupling of models for
683 contaminant spreading in the vadose zone, *Vadose Zone Journal* 7 (2).
- 684 [43] M. Darcis, H. Class, B. Flemisch, R. Helmig, Sequential model coupling
685 for feasibility studies of CO₂ storage in deep saline aquifers, submitted
686 to: *Oil & Gas Science and Technology – Rev. IFP*.
- 687 [44] N. Spycher, K. Pruess, CO₂-H₂O mixtures in the geological sequestra-
688 tion of CO₂. ii. partitioning in chloride brines at 12–100°C and up to 600
689 bar, *Geochimica et Cosmochimica Acta* 69 (13) (2005) 3309–3320.
- 690 [45] A. Bielinski, Numerical simulation of CO₂ sequestration in geological
691 formations, Ph.D. thesis (2006).