Received: 8 October 2022 | Accepted: 12 January 2023

DOI: 10.1002/pamm.202200225

Comparing methods for permeability computation of porous materials and their limitations

David Krach[1,2,*] and **Holger Steeb**[1,2]

[1] Institute of Applied Mechanics (MIB), Pfaffenwaldring 7, 70569 Stuttgart, Germany

[2] Stuttgart Center for Simulation Science, Pfaffenwaldring 5A, 70569 Stuttgart, Germany

Efficient numerical simulations of fluid flow on the pore scale allow for the numerical estimation of effective material properties of porous media, e.g. intrinsic permeability or tortuosity. These parameters are essential for various applications where hydro-mechanical properties on larger scales have to be known. Numerical tools based intrinsically on pore scale simulations are known e.g. as Digital Rock Physics in geosciences and have even more and more replaced physical experiments. For these reasons, the validation of numerical methods as well as the establishment of clear limits regarding the application areas play an important role. Here, we compute single-phase flow through a porous matrix, e.g. irregular sphere packings, sandstones, artificially created thin porous media, on the pore scale. Therefore we implement on the one hand a Smoothed Particle Hydrodynamics algorithm for solving the Navier-Stokes equations and on the other hand a Finite Difference solver for the Stokes equations. Both methods work directly and seamlessly on voxel data of porous materials which are generated by μXRCT-scans or by microfluidic experiments that have undergone segmentation and binarization. We compare both solvers from a parallel performance point of view as well as their results for flows in the Darcy regime. In addition, we investigate the limitations of the solvers using the example of a porous material whose pore geometry changes over time and precipitation affects the flow conditions.

1 Introduction

Numerical Simulations are important tools to better predict and understand the effective physical processes within different kind of porous materials. Application of characteristic parameters determined by simulations can be found in various fields such as mechanics, material science, earth and environmental sciences, and in almost all branches of engineering. With different simulation methods we aim at determining properties which are related to the fluid flow throughout the materials pores, e.g. intrinsic permeability or tortuosity.

Solving flow problems in large geometries requires not only a mathematically accurate model but also computational resources that depend on the size of the problem as well as the computational method. Accordingly, all methods must meet quality criteria in terms of performance. Both aspects are discussed in the following chapters and subsequently applied to application examples for numerically and mechanically challenging flow conditions in time-varying porous materials.

2 Mathematical models to solve single-phase flow

Fluid mechanical problems in porous materials are complex and for their solution it is therefore necessary to make assumptions adapted to the problem and computational method, which simplify the equations describing the flow or reduce the number of unknown quantities. We presuppose negligible deformation of the solid phase, isothermal processes, constant fluid viscosity $\mu^{\mathfrak{f}}$, and no-slip no-penetration boundary conditions on all fluid-solid interfaces.

2.1 Governing equations for single-phase fluid flow

The governing equations are derived by a balance equation for the conservation of mass

$$\frac{\partial \varrho^{\mathfrak{f}}}{\partial t} + \operatorname{div}(\varrho^{\mathfrak{f}}\,\mathbf{v}_{\mathfrak{f}}) \,=\, 0 \tag{1}$$

and the local form of the balance of linear momentum

$$\varrho^{\mathfrak{f}}\,\dot{\mathbf{v}}_{\mathfrak{f}} - \operatorname{div}\mathbf{T}^{\mathfrak{f}} - \varrho^{\mathfrak{f}}\,\mathbf{b} \,=\, \mathbf{0}\,, \tag{2}$$

where $\mathbf{v}_{\mathfrak{f}}$, $\mathbf{T}^{\mathfrak{f}}$, $\mathbf{b}$ denote the velocity of the fluid, the Cauchy stress tensor and a body force vector, respectively. Introducing a constitutive material law for the fluid, $\mathbf{T}^{\mathfrak{f}}$ can be split into the deviatoric part $\mathbf{T}^{\mathfrak{f}}_{neq}$, that brings in a viscosity term, and a volumetric stress $\mathbf{T}^{\mathfrak{f}}_{eq}$ by which the pore fluid pressure is introduced

$$\mathbf{T}^{\mathfrak{f}} \,=\, \mathbf{T}^{\mathfrak{f}}_{neq} + \mathbf{T}^{\mathfrak{f}}_{eq} \,=\, 2\,\mu^{\mathfrak{f}}\operatorname{dev}(\mathbf{D}^{\mathfrak{f}}) - p\,\mathbf{I}\,. \tag{3}$$

[*] Corresponding author: David Krach, mail david.krach@mib.uni-stuttgart.de, phone +49 711 685 66272, fax +49 711 685 66347

By inserting the symmetric part of the velocity gradient $\mathbf{D}^{\mathfrak{f}} = \frac{1}{2}(\operatorname{grad}\mathbf{v}_{\mathfrak{f}} + \operatorname{grad}^T\mathbf{v}_{\mathfrak{f}})$ and applying the divergence operator to the total stresses $\mathbf{T}^{\mathfrak{f}}$, we finally obtain the Navier-Stokes equations

$$\varrho^{\mathfrak{f}}\,\dot{\mathbf{v}}_{\mathfrak{f}} + \operatorname{grad} p^{\mathfrak{f}} = \mu^{\mathfrak{f}} \operatorname{div}\,(\operatorname{grad}\mathbf{v}_{\mathfrak{f}}) + \varrho^{\mathfrak{f}}\,\mathbf{b}\,. \tag{4}$$

Neglecting the convective term and assuming steady state flow leads to the Stokes equations

$$\operatorname{grad} p^{\mathfrak{f}} = \mu^{\mathfrak{f}} \operatorname{div}\,(\operatorname{grad}\mathbf{v}_{\mathfrak{f}}) + \varrho^{\mathfrak{f}}\,\mathbf{b}\,. \tag{5}$$

2.2 Finite Difference scheme for Stokes equations

Solving Eq. (1) and Eq. (5) with the Finite Difference method is known to cause problems due to the special role of pressure in these equations. We supplement the equations accordingly with an artificial compressibility and use a so-called pseudo-unsteady method [11]. This violates the prior made assumption, however, it can be shown that the solution for $t \to \infty$ approximates the correct solution in steady state [4, 5, 11]. The dimensionless, weakly compressible Stokes equations, in which the body force is already neglected can be obtained from the linearized Eqs. (1), (4) and an additional linear equation of state for the fluid pressure. Dimensionless quantities for velocity, pressure are written without any index ($\mathfrak{f}$) in order to distinguish them from the physical quantities in the previous equations. We obtain

$$\frac{\partial\mathbf{v}}{\partial t} = -\mathrm{grad}p + \frac{1}{\mathrm{Re}}\Delta\mathbf{v} \qquad \text{and} \qquad \frac{\partial p}{\partial t} = -c^2 \operatorname{div}\mathbf{v} \tag{6}$$

where $\mathrm{Re} := (\rho^{\mathrm{ref}} v_{\mathrm{ref}} L^{\mathrm{ref}})/\mu^{\mathfrak{f}}$ is the dimensionless Reynolds-Number, which must be chosen small according to the creeping flow conditions. Consistent with the dimensionless equations, we introduce the (dimensionless) velocity $c = v_{\mathrm{ref}}\,\rho^{\mathrm{ref}}/K^{\mathfrak{f}}$ which could be interpreted as the relation of the characteristic flow velocity to the speed of sound of the fluid (with compressibility $K^{\mathfrak{f}}$). Using a Taylor approximation and applied to a MAC grid (see Fig. 1), the following component-wise equations are obtained.

$$\frac{\partial v_1}{\partial t} = -\frac{\partial p}{\partial x_1} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v_1}{\partial x_1^2} + \frac{\partial^2 v_1}{\partial x_2^2} + \frac{\partial^2 v_1}{\partial x_3^2}\right)$$

$$\frac{\partial v_2}{\partial t} = -\frac{\partial p}{\partial x_2} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v_2}{\partial x_1^2} + \frac{\partial^2 v_2}{\partial x_2^2} + \frac{\partial^2 v_2}{\partial x_3^2}\right)$$

$$\frac{\partial v_3}{\partial t} = -\frac{\partial p}{\partial x_3} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v_3}{\partial x_1^2} + \frac{\partial^2 v_3}{\partial x_2^2} + \frac{\partial^2 v_3}{\partial x_3^2}\right)$$

$$\frac{\partial p}{\partial t} = -c^2\left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3}\right) \tag{7}$$

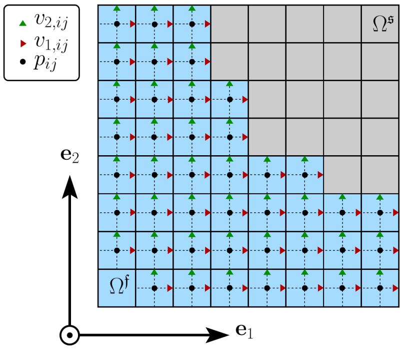

Fig. 1: Fluid-solid boundary on a MAC grid.

Discretized fluid and solid domains, e.g. of the pore space of a porous material, are denoted by $\Omega^{\mathfrak{f}}$ and $\Omega^{\mathfrak{s}}$, respectively, and the artificial speed of sound is chosen to be $c = 1.5 \cdot 10^6$ according to [3]. Note that the dimension $c \approx 10^6$ fits well to the physical ratio of the speed of sound in water to the assumed characteristic velocity of creeping flow conditions.

2.3 Smoothed Particle Hydrodynamics for Navier-Stokes equations

Smoothed Particle Hydrodynamics (SPH) is a fully mesh-free Lagrangian method that is well established for different types of flow problems [9]. SPH computes field functions based on weighted contributions of neighboring particles rather than by solving linear systems of equations. The concept of SPH is based on an approximation of a field function $f(\mathbf{x})$ more specifically the identity

$$f(\mathbf{x}) \;=\; \int_\Omega f(\mathbf{x}')\delta(\mathbf{x}-\mathbf{x}')d\mathbf{x}' \qquad \to \qquad f(\mathbf{x}) \;\approx\; \int_\Omega f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \tag{8}$$

including the Dirac delta function $\delta(\mathbf{x}-\mathbf{x}')$ that is substituted by means of a smoothing kernel $W(\mathbf{x}-\mathbf{x}',h)$. Therein $\mathbf{x}$ and $\mathbf{x}'$ are position vectors of the SPH particles and h is the so-called smoothing length of the compact-support kernel that determines a sphere of influence of a single particle and the number of neighbors. The convolution integral is numerically evaluated using Trapezoidal rule introducing discrete entities of the continuum with volume V_i. Particle wise field functions f_i are approximated only at collocation points

$$f(\mathbf{x}) \;\approx\; \int_\Omega f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\mathbf{x}' \qquad \rightarrow \qquad f_i \;=\; \sum_j^N f_j W(\mathbf{x}_i-\mathbf{x}_j,h)V_j\,. \tag{9}$$

N denotes the number of neighboring particles. With the help of this discretization method and the derivation of the kernel function, divergence and gradient operators can likewise be discretized. We obtain

$$\operatorname{grad} f(\mathbf{x}) \;=\; \sum_j^N f(x_j)\operatorname{grad} W(\mathbf{x}-\mathbf{x}',h)\,V_j \quad \text{as well as} \quad \operatorname{div}\mathbf{f}(\mathbf{x}) \;=\; \sum_j^N \mathbf{f}(x_j)\cdot\operatorname{grad} W(\mathbf{x}-\mathbf{x}',h)\,V_j \tag{10}$$

and see directly that the gradients of field functions do not have to be computed but merely the gradients of the kernel, which is an intrinsic advantage of the SPH method. By application of Eq. (9) and Eq. (10) to Eq. (1) and Eq. (4) one receives the discrete Navier-Stokes equations, here in non-dimensionless form

$$\mathbf{v}_i = -\sum_j^N m_j\left(\frac{p_i}{{\rho_i}^2}+\frac{p_j}{{\rho_j}^2}\right)\frac{\mathbf{x}_{ij}}{r_{ij}}\frac{\partial W_{ij}}{\partial r_{ij}} + \sum_j^N \frac{m_j(\mu_i-\mu_j)(\mathbf{v}_i-\mathbf{v}_j)}{\rho_i\rho_j}\left(\frac{1}{r_{ij}}\frac{\partial W_{ij}}{\partial r_{ij}}\right)+\mathbf{b} \tag{11}$$

with the discrete version of the mass balance

$$\dot{\rho}_i \;=\; \sum_j^N m_j(\mathbf{v}_i-\mathbf{v}_j)\cdot\frac{\mathbf{x}_{ij}}{r_{ij}}\frac{\partial W_{ij}}{\partial r_{ij}}\,, \quad \text{with} \quad r_{ij} = ||\mathbf{x}_i-\mathbf{x}_j||\,, \tag{12}$$

whereby the continuous field functions are transformed into particle and inter-particle forces.

2.4 Computing intrinsic permeability $\mathbf{k}^I$

The intrinsic 2nd order permeability tensor $\mathbf{k}^I = k^I_{ij}\,\mathbf{e}_i\otimes\mathbf{e}_j$ is a material specific geometrical measure which takes into account viscous momentum loss through flow in the pores of the the porous medium. Intrinsic permeability is derived by rearranging Darcy's Law

$$\mathbf{k}^I = -\frac{\mu^{\mathfrak{f}}\,\mathbf{v}_{\mathfrak{f}}}{\operatorname{grad} p} \qquad \text{with} \qquad [k^I_{ij}] = \mathrm{m}^2. \tag{13}$$

The method of its computation must be adjusted according to the two methods FD and SPH. Scalar values are computed, based on the porosity ϕ, e.g. for the component

$$\text{FD}: \qquad k^I_{11} = \frac{v_1\phi}{\mathrm{Re}} \qquad\qquad \text{SPH}: \qquad k^I_{11} = \frac{\mu^{\mathfrak{f}}\phi}{\rho^{\mathfrak{f}}g}\frac{1}{N_{\mathfrak{f}}}\sum_i^{N_{\mathfrak{f}}} v_{i,1}\,. \tag{14}$$

3 Computational aspects and implementation

3.1 `MPI` parallelized Finite Difference implementation

Since the 3-dimensional input data sets for the simulations are already voxel-based, an `MPI` implementation using Cartesian communicators can be adopted particularly well [8].

```
Result: $p$, $\mathbf{v}$-fields in porous material
MPI-Initialization, Domain Decomposition;
Input raw file;
while $\operatorname{div}\mathbf{v} > \varepsilon$ do
    Impose boundary conditions;
    Compute $p_{i,j,k}, \mathbf{v}_{i,j,k}$ ;
    Communicate $p, \mathbf{v}$ between ranks;
end
Write output;
```

Algorithm 1: Stokes Solver Algorithm.

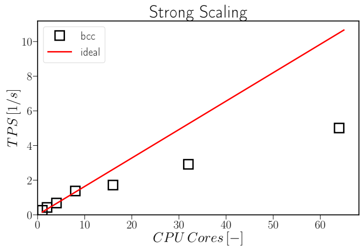

Fig. 2: Strong Scaling FD-Solver.

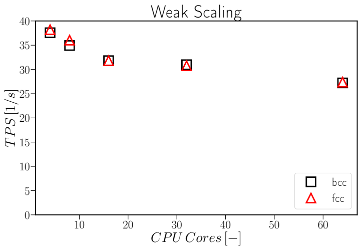

Fig. 3: Weak Scaling FD-Solver.

We implement the solver for Eqs.(7) in C++ with parallelized direct Input-Output routines for μXRCT data sets. The domain can be decomposed in all three spatial directions and distributed over an arbitrary number of computational ranks. In primary flow direction ($\mathbf{e}_1$) a pressure gradient is defined as pressure boundary condition and periodic boundary conditions for the velocity components. At all fluid-solid interfaces no-slip no-penetration boundary conditions are applied and at domain boundaries in $\mathbf{e}_2$- and $\mathbf{e}_3$-direction periodic or no-slip boundary conditions can be specified depending on the problem. The main structure of the code is given in Algorithm 1. We evaluate scalability for simple body-centered-cubic (bcc) and face-centered-cubic (fcc) spherepacking domains with 200^3 voxels. For smaller domain sizes on the individual ranks, the code scales much worse, since the routines for computing the velocity field and pressure field are comparatively fast and thus the communication generates overhead as the number of ranks increases, see Fig. 2. The study of weak scaling, Fig. 3, on the other hand, which is more relevant for our problems with large voxel numbers $> 10^9$, shows satisfactory, though not perfect, results.

3.2 SPH implementation in HOOMD-Blue

Comparing both methods, SPH is by far the one requiring more memory and computational resources. Consequently, the focus here lies in particular on the efficient implementation of a massive parallelization. The Molecular Dynamics Toolbox HOOMD-Blue [2,6] provides the basis for our CPU as well as GPU implementations of the SPH algorithm. HOOMD-Blue comprises an efficient nearest neighbor search algorithm, lean particle data structures, heuristic load balancing, a spatial domain decomposition and MPI based interface minimization to reduce communication.

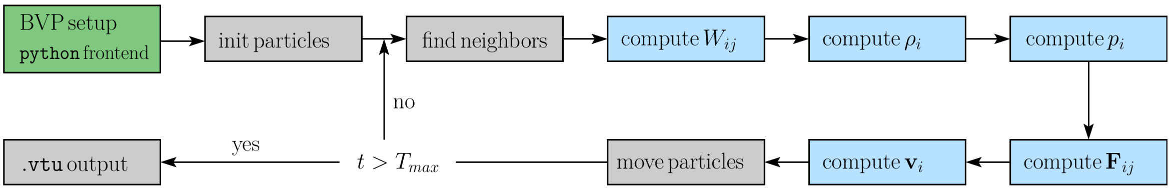

Fig. 4: SPH work-flow. Blue highlighted boxes indicate our own implementation.

Fig. 4 illustrates the program flow of the SPH code. Boundary Value Problems (BVP) are initialized with python scripts while the code is mainly written in C++. CPU and GPU implementations are written in C++ and CUDA, respectively. Boundary conditions are implemented according to Adami [1] and we employ a Velocity Verlet time integration method [13]. For more information about the implementation and especially about the strong and weak scalability on CPU and GPU clusters for different geometries we refer to recent publications of the authors, cf. Schirwon [12] and Osorno et al. [10].

4 Benchmarks

Benchmarking our codes pursues three parallel strategies. First, we test both codes with respect to simple benchmarks with existing analytic solutions, such as Poiseuille tubes. In a second step, regular sphere packings are investigated, for which analytic solutions also exist. In Fig. 5, a sweep across the porosity ϕ is presented and the normalized permeabilities plotted for two differently structured sphere packings (BCC - Body-Centered Cubic; FCC - Face-Centered Cubic). We compare the SPH and FD codes with the Kozeny-Carman equation [7]

$$k_1^{KC} = \frac{D^2}{180} \frac{\phi^3}{(1-\phi)^2} \tag{15}$$

results for porosities $0.1 \leq \phi \leq 0.9$. All simulations were performed with the same discretization (100^3 voxels).

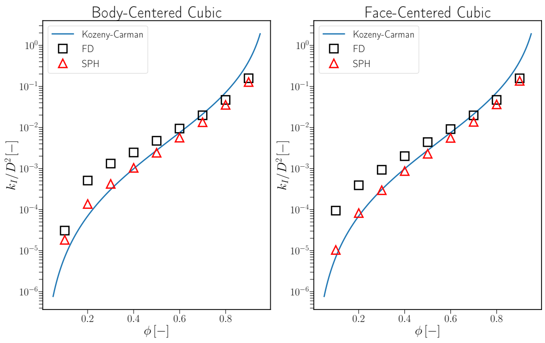

Fig. 5: Normalized permeability for different regular sphere packings in terms of a broad range of porosities.

Since no-slip boundary conditions cannot be accurately represented in the FD code, the permeability in areas of low porosity is significantly overestimated. However, it can be shown that for higher resolutions the solution of the problem tends to the analytic solution. The SPH code shows very good results for both sphere packings over all porosities. In a third step, our codes are compared with other simulation approaches for regular porous materials as used in microfluidic experiments. Our results are in good agreement with a wide variety of methods, such as

$$\text{FD}: 2.06\,, \quad \text{SPH}: 1.63\,, \quad \text{LBM}: 1.69\,, \quad \text{Homogenization}: 1.73\,, \quad \text{FEM}: 1.71 \qquad [\cdot 10^{-10}\text{m}^2] \tag{16}$$

More details on permeability determination for thin, structured porous materials can be found in Wagner et al. [14].

5 Application to clogging porous materials

Understanding and accurately predicting clogging processes at the pore scale is particularly challenging in heterogeneous porous materials. These processes may be unintended, and if unavoidable one would like to control them, for example when clogging filters. But they may also be desired, for example to block subsurface cracks. We investigate for the following use case a domain in which a chemically induced precipitation process clogs the pore space over time. Experimental data that are available online [16] is used and acquisition and experimental setup is described in [15].

5.1 Porosity-Permeability relationship

With the Finite Difference method, large geometries can be computed in its entirety. We use this to determine permeabilities of the initial configuration of the domain. In contrast to the experiment, the permeability in $\mathbf{e}_1$-direction (direction of the pressure gradient in the experiment) can be determined as well as the permeability in $\mathbf{e}_2$-direction (direction perpendicular to the pressure gradient in the experiment). Permeabilities of the domain for different time steps of the experiment are computed and normalized values are plotted over the normalized porosity. In direction of the experimentally applied pressure gradient, the decrease in permeability is less steep. In particular, after three computed snapshots ($t > 17400$s), no permeability can be determined perpendicular to the direction of the experimentally applied pressure gradient. This is due to the fact that no flow paths are open in this direction. In the pressure gradient direction, this is only the case at 4 snapshots later ($t > 32400$s).

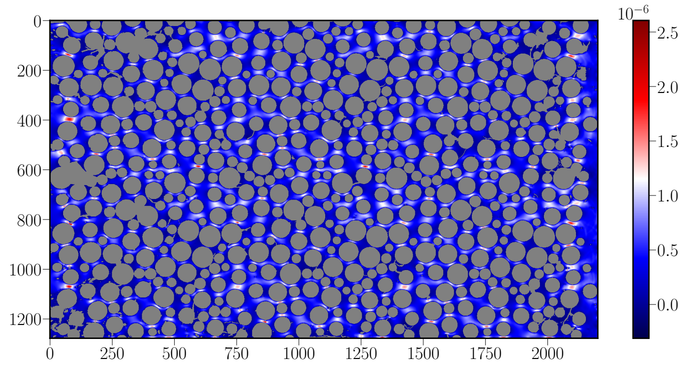

Fig. 6: Velocity component $v_{\mathfrak{f},1}$ at $t = 0$s.

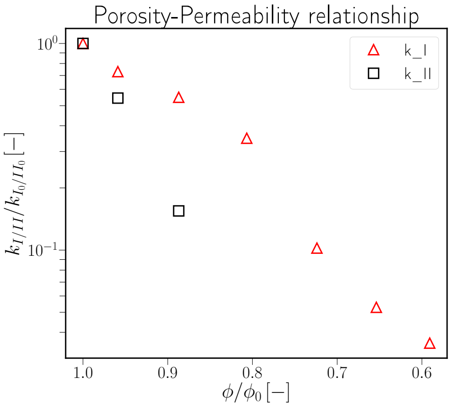

Fig. 7: Normalized porosity-permeability relationship in two main flow directions.

One simulation with 42212340 voxels requires approximately 4 hours with 64 `MPI` ranks on one CPU. Computing time required for SPH simulations cannot be justified for such domain sizes.

5.2 Formation of preferential flow paths

To illustrate the challenge of simulations at the pore scale, we consider a subdomain that includes a so-called preferential flow path. Over time (see Fig. 8 – Fig. 11), the flow-through region of the porous material narrows to a pore whose diameter is further reduced by a precipitate. The velocity component in the pressure gradient direction $v_{\mathfrak{f},1}$ is given on the right side of each figure. The evolution of the pore space on the left, where gray represents the artificial porous material, white the fluid-filled pore space and black the precipitate.

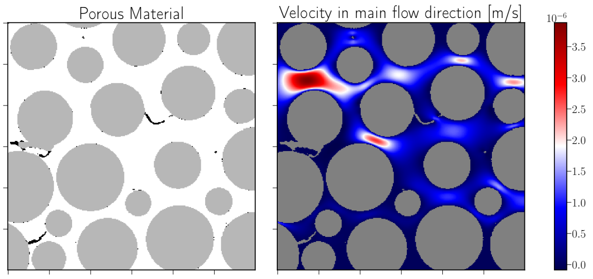

Fig. 8: Snapshot at $t = 0$s.

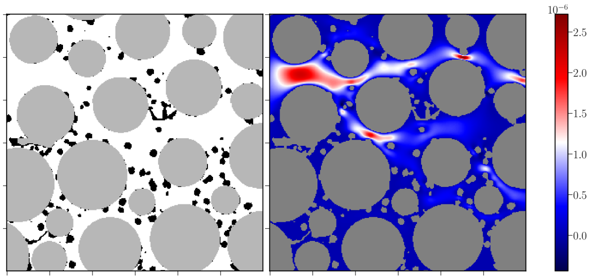

Fig. 9: Snapshot at $t = 17400$s.

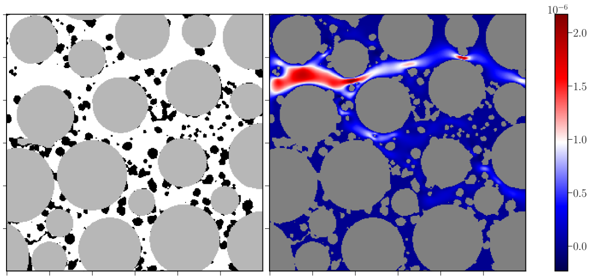

Fig. 10: Snapshot at $t = 25200$s.

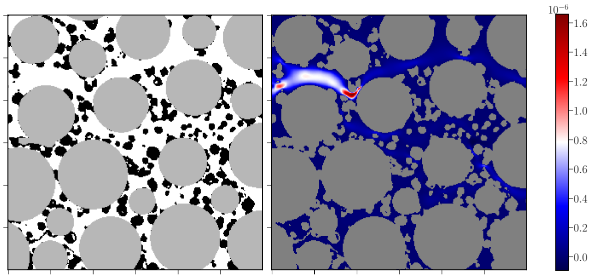

Fig. 11: Snapshot at $t = 32400$s.

The decreasing porosity and the associated reduction of the permeable area does not only cause numerical problems. Under certain circumstances, the resolution has to be adapted in order to guarantee the necessary amount of voxels even in narrow pore diameters. A fundamental question is whether, or how long, it can be justified to model this domain with the Stokes equations. For these investigations, the solution of the Navier-Stokes equation with SPH is required and these are currently in progress.

6 Conclusion and Outlook

In this paper, two fundamentally different simulation methods for flows of fluids through porous materials have been presented. Besides deriving both methods from fundamental conservation laws, their discretization and implementation are described. The implementations can correctly represent flow through different types of porous materials and are also adequately implemented from a high performance computing perspective. By definition, the methods differ due to their complexity in terms of wall time for identical problems (the FD/SPH ratio is approximately $1 : 10$). This demonstrates the fact that the problem size massively limits the usability of SPH, while the FD solver can also handle larger geometries in a reasonable time, see Fig. 6. With SPH we can only consider some smaller subdomains in which the interesting physical phenomena take place.

By means of an example it could be illustrated how both methods can contribute their advantages and disadvantages to a solidified understanding of the flow conditions and complement each other well. We follow this approach further and investigate in a next approach subdomains with relevant pore constrictions at later times of the experiment to quantify the influence/part of viscous momentum diffusion. This will then allow additional conclusions to be drawn about the limitations of the FD method.

Acknowledgements Open access funding enabled and organized by Projekt DEAL.

References

[1] S. Adami, X. Hu, N. A. Adams, J. Comput. Phys. **231(21)**, 7057–7075 (2012).

[2] J. A. Anderson, J. Glaser, and S. C. Glotzer, Comput. Mater. Sci. **173**, 109363 (2020).

[3] D. P. Bentz and N. S. Martys, A Stokes permeability solver for three-dimensional porous media. National Institute of Standards and Technology (2007).

[4] A. J. Chorin, J. Comput. Phys. **135.2**, 118-125 (1967).

[5] K. M. Gerke et al., Comput and Geosci **114**, 41-58 (2018).

[6] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, S. C. Glotzer, Comput. Phys. Commun. **192**: 97-107 (2015).

[7] J. Kozeny, Sitzber. Akad. Wiss., **136**, 271-306 (1927).

[8] Message Passing Interface Forum, Manual: https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf, (2021).

[9] J. J. Monaghan, Rep. Prog. Phys, **68(8)**, 1703 (2005).

[10] M. Osorno, M. Schirwon, N. Kijanski, R. Sivanesapillai, H. Steeb and D. Göddeke, Comput. Phys. Commun., **267**, 108059 (2021).

[11] R. Peyret and T. Taylor, Computational methods for fluid flow (Springer Science and Business Media, Berlin, 2012).

[12] M. Schirwon, Efficient simulation of challenging PDE problems on CPU and GPU clusters. (Universität Stuttgart 2021).

[13] L. Verlet, Physical Review **159(1)**, 98 (1967).

[14] A. Wagner et al., Transp. Porous Media , **138(1)**, 1-23 (2021).

[15] F. Weinhardt, J. Deng, J. Hommel, S. Vahid Dastjerdi, R. Gerlach, H. Steeb and H. Class, Transp. Porous Media, 1-23 (2022).

[16] F. Weinhardt, J. Deng, H. Steeb and H. Class, Dataset: https://doi.org/10.18419/darus-1799, (DaRUS, Universität Stuttgart 2021).