

On h -refined meshless solution to Navier-Stokes problem in porous media: comparing meshless Lattice Boltzmann Method with ACM RBF-FD approach

Dawid Strzelczyk^{1,*}, Miha Rot^{2,3}, Gregor Kosec² and Maciej Matyka^{1,2}

¹ *Institute of Theoretical Physics, Faculty of Physics and Astronomy, University of Wrocław, pl. M. Borna 9, 50-204, Wrocław, Poland*

² *Parallel and Distributed Systems Laboratory, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*

³ *Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia*

* dawid.strzelczyk@uwr.edu.pl

Abstract In this paper, two mesh-free CFD solvers for pore-scale fluid flow through porous media are considered, namely the Lattice Boltzmann Method with the two relaxation time collision term and the direct Navier-Stokes solver under the artificial compressibility limit. The porous media is built with a regular arrangement of spherical grains with variable radii, which allows control of the porosity. Both solvers use the same h -refined meshless spatial discretization to adequately capture the underlying geometry and the same Radial Basis Function (RBF) method to approximate the involved fields and partial differential operators. First, the results are compared with the data from the literature in terms of drag coefficient and permeability at different porosities achieving excellent agreement with the reported results. Next, the simulations are extended beyond the porosity range reported in the literature using proposed h -refined CFD solvers. The results are supported by convergence and timing analyses and discussions on meshless parameters such as stencil size and refinement settings.

Keywords: fluid flow; meshfree methods; radial basis functions; Lattice Boltzmann Method; Navier-Stokes equations; artificial compressibility; porous media;

1 Introduction

Porous media are ubiquitous and the fluid flow through them is of paramount importance for science, technology, and life. It concerns such fundamental issues as the health and performance of the human body, which is composed of 70% fluids, where the lungs and the arterial system can indeed be treated as porous systems [1], or solidification processes, where the transport of solutes in mushy (porous) regime governs the macrosegregation effect [2], and after all, study of radioactive waste seepage in radioactive waste disposal facilities [3], to name just a few examples.

Research in the field of fluid dynamics (porous or free fluid) is not possible with analytical methods and therefore requires sophisticated experimental methods, complex differential calculus methods, or the use of numerical methods. Traditionally, problems in fluid dynamics have been approached numerically using mesh-based methods, where, despite the long history of research on mesh generators [4], meshing remains a difficult problem that cannot be fully automated and therefore often requires significant human assistance. As a matter of fact, the meshing of irregularly shaped 3D geometries, especially when dealing with complex geometries such as porous media [5], is one of the most complex and time-consuming steps in the overall mesh-based numerical solution [6].

In response to the complexity of meshing, the development of numerical methods took two principal directions. In addition to the development of specific methods and algorithms for meshing [4], a class of numerical methods based on the meshless principle has been developed [7]. The conceptual difference between mesh-based and meshless methods lies in the consideration of the relationships between the computational nodes. Meshless methods define the relationship between the nodes completely using only the internodal distances, thus freeing themselves from the constraints of using meshes. An important implication of this simplification is that meshless methods can work with scattered nodes. Although it is generally recognized that certain rules must be followed when generating such scattered nodes [8], the positioning of the nodes is significantly less complex compared to meshing [9] and can be automated regardless of the dimensionality or shape of the domain under consideration [8, 10]. Moreover, the h -adaptivity in the meshless setup is almost free, since the nodes can be distributed with variable internodal distance without any special treatment [11, 8, 12]. In recent years, a specific

meshless method has gained popularity, namely a radial basis function (RBF) generated finite differences (FD) [13]. The hallmark of the RBF-FD is an approximation based on the combination of polynomials that ensure consistency up to their order [14] and RBF (typically polyharmonic splines) that help stabilize the method. Although the idea is not entirely new, recent theoretical and experimental research has taken the understanding of such a numerical approach to a new level [14, 15, 12, 11, 16].

In fluid dynamics, different meshless approaches have been successfully applied to the Navier-Stokes problem, namely methods based on direct solving of Navier-Stokes equation using different meshless approximations and pressure-velocity couplings [16, 17, 18] and Meshless lattice Boltzmann methods [19, 20]. Each of these methods has different properties and is used differently for certain physical problems. It is important to understand and employ them in a proper way to utilize their strengths and avoid weaknesses in the given context, which in our case is the fluid flow through the irregular, complex pores of porosity-dependent porous media.

The aim of this work is to formulate and implement a direct meshless Navier-Stokes solver under the artificial compressibility limit and meshless Lattice Boltzmann Method with the two relaxation time collision. Both methods use the same h -refined meshless spatial discretization and the same RBF method to approximate the involved fields and partial differential operators [21] for a fair comparison. We test both methods on the classical problem of flow through idealized three-dimensional porous media represented as a periodic array of spheres and compare meshless results with the previously recognized benchmarks, i.e. smoothed particle hydrodynamics method based solution provided by Holmes, et al. [22] and solution provided by Larson and Higdon [23]. We are particularly interested in extending the simulations to the limiting cases of the model, i.e. low porosity, high porosity, and touching limit of the singularity contact point. Under these conditions, we discuss the strengths and weaknesses of both formulations in terms of numerical accuracy and stability, convergence rate, complexity, and robustness using refined scattered nodes. Furthermore, we identify regions where both approaches have problems and discuss possible overlaps and gap-filling between the two. Finally, we extend the spectra of available benchmark data by expanding the range of porosity in the simulations.

2 Methods

We study the pore scale flow through a porous medium by modeling it with the incompressible Navier-Stokes system of equations

$$\begin{aligned}\nabla \cdot \mathbf{v} &= 0, \\ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} &= -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g}.\end{aligned}\tag{1}$$

where $\mathbf{v}(t, \mathbf{x})$ is the velocity field, $p(t, \mathbf{x})$ is the pressure field, and ρ , ν and $\mathbf{g}$ are the density, the kinematic viscosity and the body force, respectively. To solve the equations, we take two approaches. The first is to solve the Boltzmann transport equation with the meshless Lattice Boltzmann Method (MLBM) described in Sec. 2.2 and then, by the virtue of Chapman-Enskog expansion[24], derive $\mathbf{v}$ and p fields from its solution. The other is to solve Eq. (1) directly, using a meshless Navier-Stokes (MNS) solver described in Section. 2.3. The pseudocodes for MNS and MLBM are provided in Appendix A. Both methods rely on the meshless RBF-FD approximation framework; either to interpolate values between the Eulerian and the Lagrangian nodes in LBM or to approximate the spatial derivatives in MNS. The principles of the approximation method and the description of the node positioning algorithm used in the present study are provided in Sec. 2.1.

2.1 Meshless discretization and approximation

The discretization of the computational domain for the meshless approximation is achieved by positioning N scattered *computational nodes* $\mathbf{x}_i \in \mathbb{R}^D$, with D denoting the spatial dimension. The nodes are placed with a variable density, defined with the inter-nodal distance function $h(\mathbf{x})$, utilizing a dedicated meshless dimension-independent variable density (DIVG) node positioning algorithm [25] with an extension to account for the periodic nature of the domain. DIVG is an iterative algorithm based on an *expansion queue* that results in discretization proceeding as an advancing front away from the initial nodes used to seed the queue. At each iteration, we attempt to expand the discretization with candidate nodes laying on a circle with radius $h(\mathbf{x}_d)$ around the de-queued node position $\mathbf{x}_d$. The candidates that fall inside the domain and not too close to existing nodes, based on their local $h(\mathbf{x}_i)$, are then added to the discretization and to the queue for further expansion which continues until the queue is empty.

No further information, e.g. meshing, regarding the node cloud is required as the *stencil* S_i , i.e. the domain for the local approximation in the i -th computational node, is constructed solely based on the inter-nodal distance. Although more complex strategies for stencil construction exist [26] we use the simplest and populate S_i with indices of N_L closest neighbors to the i -th computational node.

After the stencils for computational nodes are constructed we can use them to form a generalized finite difference approximation that numerically approximates the linear differential operator $\mathcal{L}$ in the central node

$$(\mathcal{L}u)_i \approx \sum_{j=1}^{N_L} w_{i,j} u_{S_i(j)}, \quad (2)$$

based on the function values u in stencil nodes and weights w that are pre-computed by demanding the exactness of Eq. (2) for a set of basis functions. Note that the same general framework used for differential operators can also be used with $\mathcal{L} = 1$ for approximation of function values. We use radial basis functions

$$\phi(i, j) = \phi\left(\frac{\|\mathbf{x}_j - \mathbf{x}_i\|_2}{\delta_i}\right), \quad (3)$$

with a local scaling factor δ_i that decouples the approximation from the choice of the coordinate system (especially important for RBF with a shape parameter[27]). It can be set to an arbitrary local measure of distance, e.g., the distance between the central node and its closest neighbor. Replacing u with the RBF in the exact form of Eq. (2) leads to a local linear system $\mathbf{A}\mathbf{w}_i = \mathbf{b}$

$$\begin{bmatrix} \phi(S_i(1), S_i(1)) & \cdots & \phi(S_i(1), S_i(N_L)) \\ \vdots & \ddots & \vdots \\ \phi(S_i(N_L), S_i(1)) & \cdots & \phi(S_i(N_L), S_i(N_L)) \end{bmatrix} \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,N_L} \end{bmatrix} = \begin{bmatrix} (\mathcal{L}\phi)(i, S_i(1)) \\ \vdots \\ (\mathcal{L}\phi)(i, S_i(N_L)) \end{bmatrix}, \quad (4)$$

for each stencil with the solution providing approximation weights $\mathbf{w}_i$. The right hand side vector $\mathbf{b}$ is formed by applying the linear operator $\mathcal{L}$ to the basis function and evaluating the result with an argument analogous to Eq. (3).

We use an RBF with no shape parameter, the polyharmonic spline (PHS)

$$\phi(r) = r^k, \quad (5)$$

with odd order k , to avoid additional problems with parameter tuning. Local matrices constructed with polyharmonic RBF are only conditionally positive definite and need to be augmented with monomials [28, 29]. The system is expanded with $N_p = \binom{m+D}{m}$ monomials¹ p , where m denotes the monomial order. The monomials are scaled similarly to RBFs in Eq. (3)

$$p_l(i, j) = p_l\left(\frac{\mathbf{x}_j - \mathbf{x}_i}{\delta_i}\right). \quad (6)$$

The linear system from Eq. (4) is augmented with monomials

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_i \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \quad (7)$$

$$\mathbf{P} = \begin{bmatrix} p_1(S_i(1), S_i(1)) & \cdots & p_{N_p}(S_i(1), S_i(1)) \\ \vdots & \ddots & \vdots \\ p_1(S_i(N_L), S_i(1)) & \cdots & p_{N_p}(S_i(N_L), S_i(1)) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} (\mathcal{L}p_1)(S_i(1), S_i(1)) \\ \vdots \\ (\mathcal{L}p_{N_p})(S_i(1), S_i(1)) \end{bmatrix},$$

with the additional weights $\boldsymbol{\lambda}$ treated as Lagrange multipliers and discarded after computation.

Augmentation with an order of at least $m = \frac{k-1}{2}$ is required to guarantee the positive definiteness for a PHS with order k . Higher orders provide better convergence characteristics [30] at the cost of increased computational complexity, since the required stencil size is $N_L \geq N_p$, with $N_L > 2N_p$ as the often recommended value [14]. We use a slightly larger stencil size $N_L = 25$ than the recommended minimum 20 based on observations in 3.4. In the remainder of the paper, we use approximations with $k = 3$ and $m = 2$ unless otherwise specified.

2.2 Meshless Lattice Boltzmann Method

In the present study we use the meshless Lattice Boltzmann Method the principles of which are outlined in [31, 20]. It solves the discrete velocity Boltzmann equation[32]:

$$f_k(t+1, \mathbf{x}) = f_k^{\text{post}}(t, \mathbf{x} + \mathbf{e}_{k'}), \quad k = 0, 1, \dots, q-1 \quad (8)$$

¹The 6 monomials in $D = 3$ case with $m = 2$ would be $p = \{1, x, y, z, x^2, xy, xz, y^2, yz, z^2\}$.

in a semi-Lagrangian way. In the above equation, f_k is the k -th distribution function, $\mathbf{e}_k = -\mathbf{e}_{k'}$ is the k -th streaming direction and its opposite, respectively, and superscript "post" denotes post-collision distribution. We implement compressible D3Q15 model ($q=15$) with the following set of discrete streaming directions:

$$\begin{array}{cc|cc} k & \mathbf{e}_k & k & \mathbf{e}_k \\ \hline 0 & (0, 0, 0) & 7 & (1, 1, 1) \\ 1 & (1, 0, 0) & 9 & (1, 1, -1) \\ 3 & (0, 1, 0) & 11 & (1, -1, 1) \\ 5 & (0, 0, 1) & 13 & (1, -1, -1) \end{array} \quad , \quad \mathbf{e}_k = -\mathbf{e}_{k-1} \text{ for } k = 2, 4, \dots, 14 \quad (9)$$

Using two relaxation time (TRT) collision kernel[33, 24], post-collision distributions have the form of:

$$f_k^{\text{post}}(t, \mathbf{x}) = f_k(t, \mathbf{x}) - \frac{1}{\tau^+} (f_k^+(t, \mathbf{x}) - f_k^{\text{eq},+}(t, \mathbf{x})) - \frac{1}{\tau^-} (f_k^-(t, \mathbf{x}) - f_k^{\text{eq},-}(t, \mathbf{x})) + F_k \quad (10)$$

where τ^+ and τ^- are symmetric and anti-symmetric relaxation parameters related by:

$$\tau^- = \frac{1}{2} + \frac{\Lambda}{\tau^+ - 0.5}, \quad (11)$$

with the value of $\Lambda = 1/4$ used in the present study for its positive effect on the simulation stability. Unless otherwise stated we use the value of $\tau^+ = 1$. f_k^+ and f_k^- are the symmetric and anti-symmetric components of the k -th distribution function:

$$\begin{aligned} f_k^+ &= \frac{1}{2}(f_k + f_{k'}) \\ f_k^- &= \frac{1}{2}(f_k - f_{k'}), \end{aligned} \quad (12)$$

and $f_k^{\text{eq},+}$, $f_k^{\text{eq},-}$ are symmetric and anti-symmetric components of the k -th equilibrium distribution function:

$$\begin{aligned} f_k^{\text{eq},+} &= \frac{1}{2}(f_k^{\text{eq}} + f_{k'}^{\text{eq}}) \\ f_k^{\text{eq},-} &= \frac{1}{2}(f_k^{\text{eq}} - f_{k'}^{\text{eq}}). \end{aligned} \quad (13)$$

The relaxation time τ^+ defines the kinematic viscosity of the considered fluid:

$$\nu_{lb} = c_s^2 \left(\tau^+ - \frac{1}{2} \right), \quad (14)$$

where the lattice speed of sound is $c_s = 1/\sqrt{3}$ and the subscript lb denotes a macroscopic quantity in LBM model units (non-dimensional), to distinguish it from the same quantity in physical units which will be denoted without any subscript. We use the discrete equilibrium distribution function conserving the first two moments of the local distributions:

$$f_k^{\text{eq}} = \rho_{lb} \omega_k \left[1 + \frac{\mathbf{e}_k \cdot \mathbf{v}_{lb}}{c_s^2} + \frac{(\mathbf{e}_k \cdot \mathbf{v}_{lb})^2}{2c_s^4} - \frac{\mathbf{v}_{lb}^2}{2c_s^2} \right], \quad (15)$$

where lattice weights ω_k have the values of:

$$\omega = \begin{cases} 2/9, & k = 0 \\ 1/9, & k = 1, \dots, 6 \\ 1/72, & k = 7, \dots, 14 \end{cases}, \quad (16)$$

and ρ_{lb} and $\mathbf{v}_{lb}$ are local fluid macroscopic density and velocity vector, respectively:

$$\begin{aligned} \rho_{lb} &= \sum_{k=0}^{q-1} f_k \\ \mathbf{v}_{lb} &= \frac{1}{\rho_{lb}} \sum_{k=0}^{q-1} f_k \mathbf{e}_k \end{aligned} \quad (17)$$

To obtain the relative pressure we use the ideal gas equation of state with the deviation of the density from its mean:

$$p_{lb} = (\rho_{lb} - \langle \rho_{lb} \rangle) c_s^2. \quad (18)$$

We implement acceleration using first-order discretization in velocity space:

$$F_k = \omega_k \frac{\mathbf{e}_k \cdot \mathbf{F}_{lb}}{c_s^2} \quad (19)$$

where $\mathbf{F}_{lb} = \rho_{lb} \mathbf{g}_{lb}$ is the body force defined in terms of acceleration $\mathbf{g}_{lb}$.

The conversion from LBM units to physical units is achieved by the multiplication of a quantity by a suitable conversion factor:

$$\begin{aligned} L &= \delta x L_{lb}, & \text{length,} & & \nu &= \frac{\delta x^2}{\delta t} \nu_{lb}, & \text{viscosity,} \\ T &= \delta t T_{lb}, & \text{time,} & & g &= \frac{\delta x}{\delta t^2} g_{lb}, & \text{acceleration,} \\ v &= \frac{\delta x}{\delta t} v_{lb}, & \text{velocity,} & & \rho &= \rho_{ref} \rho_{lb}, & \text{density,} \\ & & & & p &= \rho_{ref} \frac{\delta x^2}{\delta t^2} p_{lb}, & \text{pressure} \end{aligned} \quad (20)$$

where δx , δt , ρ_{ref} are the streaming distance length, the timestep length and the reference density, respectively, all in physical units. By virtue of Chapman-Enskog expansion [24], the macroscopic velocities and pressure obtained from the solution of the discrete velocity Boltzmann equation are the solution to the weakly compressible Navier-Stokes equation.

Since the flow domain is discretized with a set of scattered nodes, it is convenient to reformulate the streaming step (Eq. (8)) into physical units and speak of the streaming distances $\delta \mathbf{x}_k = \mathbf{e}_k \delta x$ rather than streaming directions (see Fig. 1). At the same time, the collision step (Eq. (10)) is still solved non-dimensionalized. In this way, the positions of the departure (Lagrangian) nodes for f_k^{post} can be related to the meshless space discretization (Eulerian) nodes as $\mathbf{x} + \delta \mathbf{x}_k$. In contrast to the lattice-based LBM, they do not need to coincide with the Eulerian nodes. Unless otherwise stated, in the presented MLBM setups we used the streaming distance equal to half of the minimal internodal distance, i.e. $\delta x = h_{min}/2$ (compare with Eq. (26)). Along with the relaxation time τ^+ and the physical kinematic viscosity ν it determines the timestep length δt according to the viscosity conversion from Eq. (20)².

To obtain the value of the post-collision distribution function in such a case, we use a method of meshless interpolation described in Sec. 2.1 ($\mathcal{L} = 1$ in Eq. (2)).

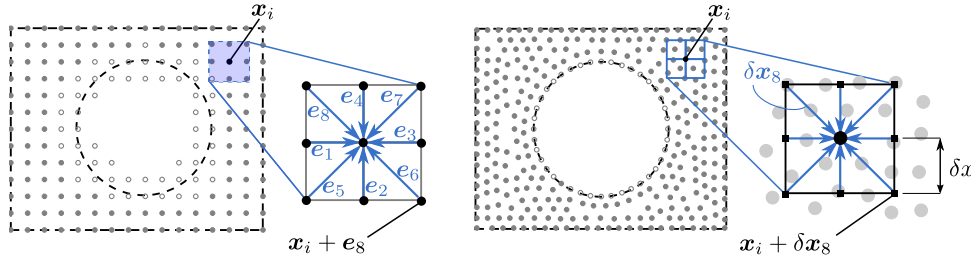


Figure 1. Graphical interpretation of differences between the space discretization of a circle in standard (lattice-based) LBM (*left*) and MLBM (*right*). The open symbols in LBM case represent the nodes whose links are cut by the circle's boundary. In MLBM the open symbols represent the boundary nodes lying exactly on the circle's surface.

In MLBM the no-slip condition is achieved through the multireflection bounce-back proposed by Ginzburg and d'Humières [34] applied to the boundary nodes. As the discretization nodes lie exactly on the boundary, the formula for the bounce-back simplifies to:

$$f_k(t + \delta t, \mathbf{x}) = f_{k'}(t + \delta t, \mathbf{x}). \quad (21)$$

The initial conditions for all simulations presented in this work were equilibrium populations (Eq. (15)) parametrized with zero macroscopic velocities $\mathbf{v}_{lb}$ and unit density ρ_{lb} . The Mach number did not exceed $2 \cdot 10^{-4}$ in all simulations in the steady state.

2.3 Meshless Navier-Stokes solver

One of the main problems in directly solving the Eq. (1) lies in implementing the pressure-velocity coupling procedure that ensure continuity. We use one of the fundamental approaches, the artificial compressibility method (ACM)[35, 36], first introduced by Chorin [37] in 1967, that relies on transiently introducing a slight compressibility into the system. The main benefit of this method from the computational standpoint is that

²The actual values of δt that we use range from $1.04 \cdot 10^{-6}$ to $1.67 \cdot 10^{-5}$.

it avoids solving the global pressure Poisson equation required by pressure projection[38], the other main class of pressure-velocity coupling methods, thus allowing for perfect parallelization within a single time step. Additionally, the selection of ACM for pressure-velocity coupling is motivated by its innate similarities to LBM [39].

We use the explicit Euler method for the temporal discretization of the Navier-Stokes equation

$$\mathbf{v}' = \mathbf{v} + \delta t (\nu \nabla^2 \mathbf{v} - \mathbf{v} \cdot \nabla \mathbf{v} + \mathbf{g}), \quad (22)$$

where δt is the time step, μ the viscosity and ρ the density. The time step³ $\delta t = 0.1 \frac{h_{\min}^2}{2\nu}$ is set as a function of the smallest inter-nodal distance $h_{\min}$. Note that the pressure term is omitted while calculating the intermediate velocity $\mathbf{v}'$. This is because the subsequent pressure-velocity coupling

$$p \leftarrow p - \delta t C^2 \rho (\nabla \cdot \mathbf{v}), \quad (23)$$

$$\mathbf{v} \leftarrow \mathbf{v}' - \frac{\delta t}{\rho} \nabla p, \quad (24)$$

is done iteratively. The magnitude of the AC is determined with the artificial speed of sound C [40]

$$C = \beta \max(\max_i(\|\mathbf{v}_i\|_2), \|\mathbf{v}_{ref}\|_2), \quad (25)$$

where β is the compressibility parameter, and $\mathbf{v}_{ref}$ a reference velocity that prevents instabilities due to C reaching zero in stagnation points. We use $\beta = 10$ for all computations presented in this paper. If we sought a time-accurate solution the pressure-velocity coupling iteration would have to be iterated until the maximum divergence of the velocity field dropped below a pre-determined threshold, but since we are dealing with a steady state problem, the number of iterations can be limited. We chose to perform $n_p = 3$ pressure correction iterations for each velocity iteration as a reasonable compromise that empirically provided the fastest convergence to a steady state.

The non-slip boundary condition (BC) is enforced by setting boundary node values according to the Dirichlet BC $\mathbf{v} = 0$ for velocity and the Neumann BC $\frac{\partial p}{\partial \mathbf{n}} = 0$ for pressure.

2.4 Case definition

We consider a flow through a three-dimensional, infinite, periodic, simple cubic (SC) array of spheres. All the spheres in the domain have the same radius and we consider cases with radii ranging from $r = 0.04$ to $r = 0.69$. This gives the porosity range $\varphi \in [0.05, 0.9997]$ and a natural transition from the overlapping to the diluted regime of the system. The flow is forced with a constant acceleration $\mathbf{g} = [0.1, 0, 0]$. In actual computations, it is thus sufficient to reduce the computational domain to the fluid-occupied part of a single periodic cell with the side length $d = 1$. The no-slip boundary condition is imposed on the spheres' walls and periodic boundary conditions are imposed in all three directions via the periodic search of stencils' members.

The discretization is refined locally by decreasing the internodal distance towards the spheres' surface. We implement the refinement by specifying the function of the target distance between the nodes:

$$h(\mathbf{x}) = h_{\max} - \exp\left(-\frac{\tilde{\phi}^2(\mathbf{x})}{2\varepsilon^2}\right) (h_{\max} - h_{\min}), \quad \tilde{\phi}(\mathbf{x}) = \phi(\mathbf{x}) \left(\sqrt{3}/2 - r\right) \quad (26)$$

where $\phi(\mathbf{x})$ is the distance of the point $\mathbf{x}$ to the closest spherical obstacle, $h_{\min}$ and $h_{\max}$ are minimum and maximum values of the target distance, respectively and ε is the shape parameter. Note that $h(\mathbf{x})$ reaches the minimum on the spheres' surface ($\phi(\mathbf{x}) = 0$) and maximum away from it. The shape of the refinement function for a selection of parameters is shown in (Fig. 2). Unless otherwise stated we use the values of $h_{\min} = 0.5h_{\max}$ and $\varepsilon = 0.295$. Example visualizations of the discretizations are shown in Fig. 3.

To allow for a reliable comparison between MLBM and meshless Navier-Stokes results we use the stopping criterion based on the relative change of permeability k/d^2

$$|\Delta|_{k/d^2} = \frac{1}{(k/d^2)(t)} \frac{|(k/d^2)(t) - (k/d^2)(t - \Delta t)|}{\Delta t} \quad (27)$$

calculated every 10^4 timesteps with $\Delta t = 10^4 \delta t$ and we stop the simulations when $|\Delta|_{k/d^2} < 10^{-2}$. The permeability is defined as:

$$k/d^2 = \frac{q\nu}{|\mathbf{g}|d^2}, \quad q = \frac{1}{d^2} \cdot \frac{1}{5} \sum_{i=1}^5 \int_{P_i} dA v_0(\mathbf{x}) \quad (28)$$

³The actual values of δt that we use range from $1.25 \cdot 10^{-6}$ to $2 \cdot 10^{-5}$.

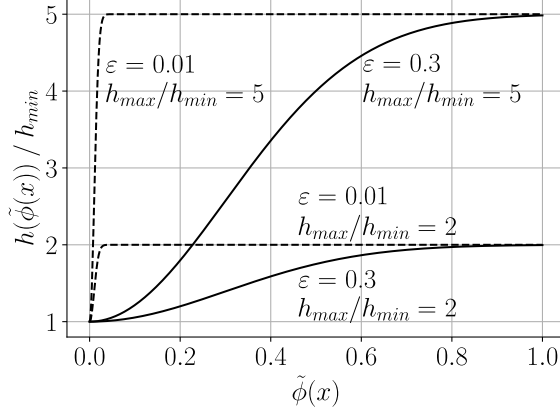


Figure 2. The average distance between points as a function of the normalized distance function value for various shape parameters and $h_{\max}/h_{\min}$ ratios.

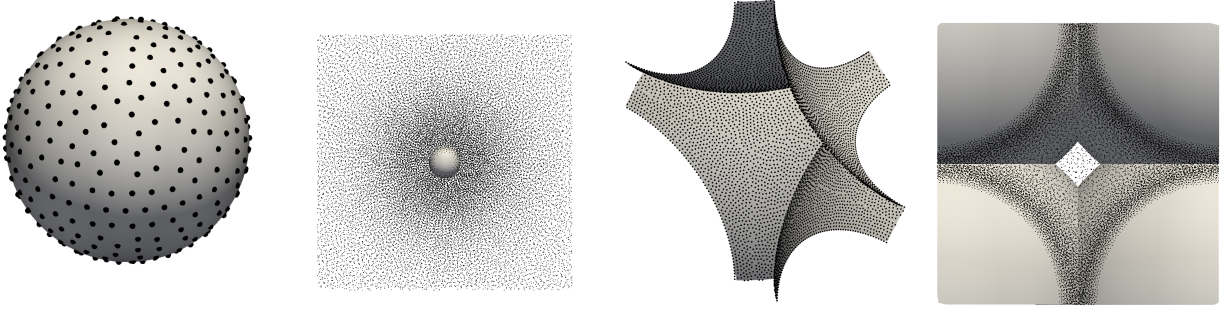


Figure 3. Meshless space discretizations used in the calculations. *Two leftmost pictures:* $r = 0.062$, *two rightmost pictures:* $r = 0.6526$. Each pair consists of the visualization of the boundary discretization and the subset of nodes with $y \in [0.45; 0.55]$ viewed along $+y$ direction. The spheres' boundaries are rendered as grey surfaces.

where q is the mean x -component of the velocity averaged over five cross-sections $P_i = \{(x, y, z) \in \Omega : x \in \{0, 0.2, 0.4, 0.6, 0.8\}\}$

In both methods, for the calculation of surface forces acting on the spheres we use the stress tensor σ_{ij} at boundary nodes:

$$\sigma_{ij} = -p\delta_{ij} + \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (29)$$

The integration of the stresses over the spheres' surface assumes equal area ΔA assigned to each boundary node:

$$\Delta A = A/N_b \quad (30)$$

where N_b is the number of boundary nodes and A is the area of the obstacles in a unit cell. To obtain the hydrodynamic force on the obstacle we sum:

$$F_i = \Delta A \sum_{n \in I_b} (\sigma_{ij} \hat{n}_j)_{x_n} \quad (31)$$

where $\hat{n}_i$ is a local normal vector and I_b is the set of boundary nodes indices.

3 Results and Discussion

3.1 Analysis of the velocity and stresses fields

We first present the visualizations of the velocity fields obtained for diluted and overlapping systems in Fig. 4. For the small sphere case, we observe an undisturbed flow away from the obstacle and a boundary layer near the sphere's surface. For the large sphere case, a strong channelization is visible with a considerable portion of the fluid volume excluded from the percolation where recirculation occurs. Also, the value of the maximum velocity magnitude differs by two orders of magnitude between the two cases.

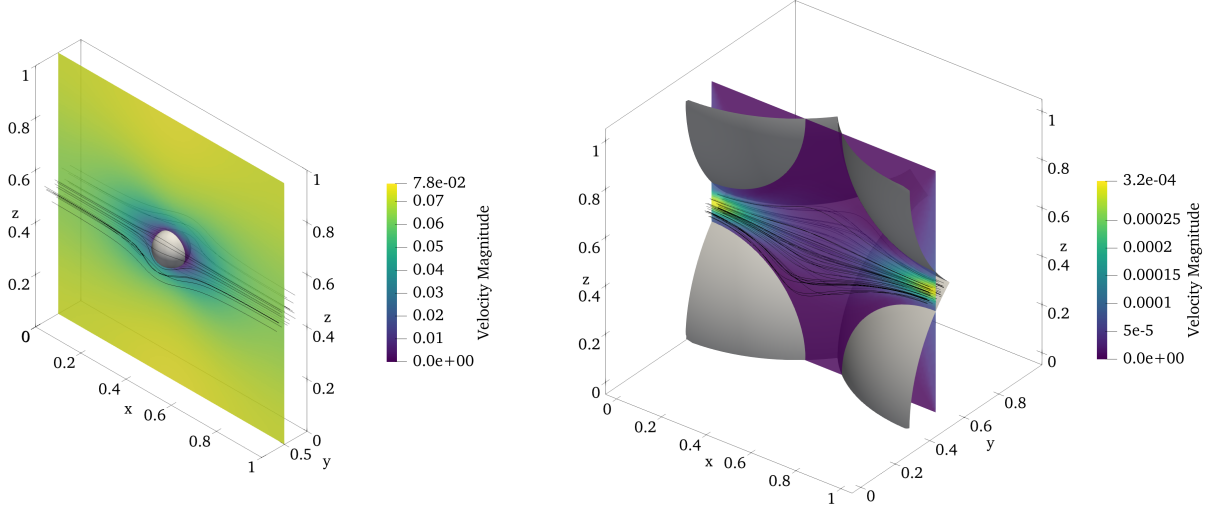


Figure 4. Streamlines of the velocity field and a cross section of the velocity magnitude field for $r = 0.062$ (left) and $r = 0.6526$ (right) case.

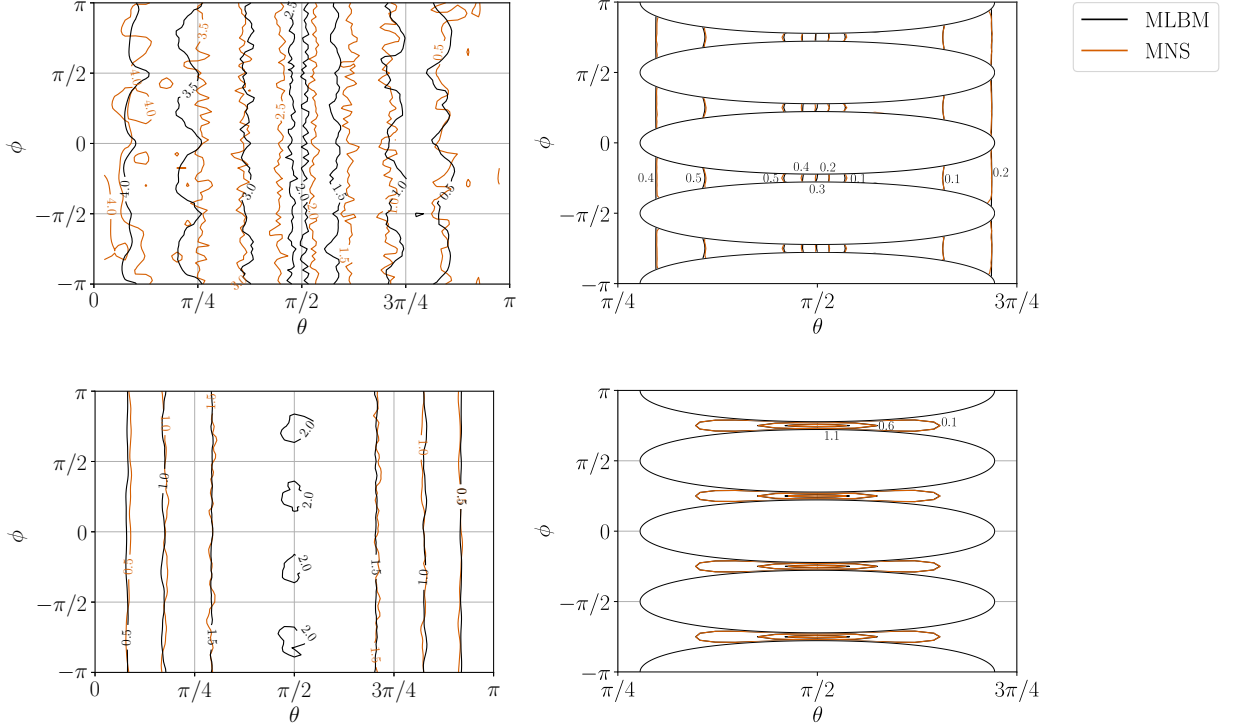


Figure 5. Pressure (top) and tangent force (Eq. (31)) magnitude (bottom) contours on the sphere's surface for $r = 0.062$ (left) and $r = 0.6526$ (right) obtained with MLBM and MNS. The contours are presented in a spherical coordinate system with the origin at the sphere's center. θ is the polar angle measured from the negative direction of the x -axis and ϕ is the azimuthal angle measured from the positive direction of the z -axis. The blank ellipses in the right column result from the intersection between the obstacle and the periodic cubic cell.

Initially, we performed a set of simulations for $|g| = 49$ as indicated in [22]. We noticed, however, that both results obtained with MLBM and MNS for $|g| = 49$ exhibit an asymmetry about the $x = 0.5$ plane in the x -component of the velocity field at high porosities (data not shown). It can be associated with inertial effects in the flow. If we define the Reynolds number as:

$$Re = \frac{2rq}{\nu}, \quad (32)$$

we find that for $g = 49$ and $r = 0.062$ the Reynolds number is approximately 4. This is already when inertial

effects appear in the flow past a single sphere[41]. This motivated us to decrease the acceleration to $|g| = 0.1$ to ensure Darcy regime in the whole range of porosities. Now, for the lower acceleration, the Reynolds number for the same system is on the order of 10^{-2} .

Fig. 5 shows the isocontours of pressure and tangent force magnitude on the sphere's surface for $r = 0.062$ and $r = 0.6526$. Results obtained with both methods are compliant with each other in terms of the values of the quantities. The most pronounced difference between the two methods is visible in the tangent force isocontours for $r = 0.062$ case, where MNS records values ≥ 2 , not present in MLBM solution. Nevertheless, this level of compliance between the methods is satisfactory, especially as MLBM and MNS use vastly different strategies to calculate the pressure (see Sec. 2). Moreover, the use of boundary-compliant discretization in MLBM allows for a direct comparison of σ_{ij} with the Navier-Stokes solver, inaccessible in the lattice-based LBM due to the staircase approximation of boundaries[42, 43].

3.2 Convergence of permeability and drag coefficient

We next check the convergence of the permeability k/d^2 (Eq. (28)) and the drag coefficient K with the space discretization refinement for various sphere radii. The drag coefficient is the non-dimensionalized drag force:

$$K = \frac{F_d}{6\pi r \varphi q \mu}. \quad (33)$$

We use the minimal distance between nodes $h_{\min} \in \{0.02, 0.014, 0.01, 0.007, 0.005\}$.

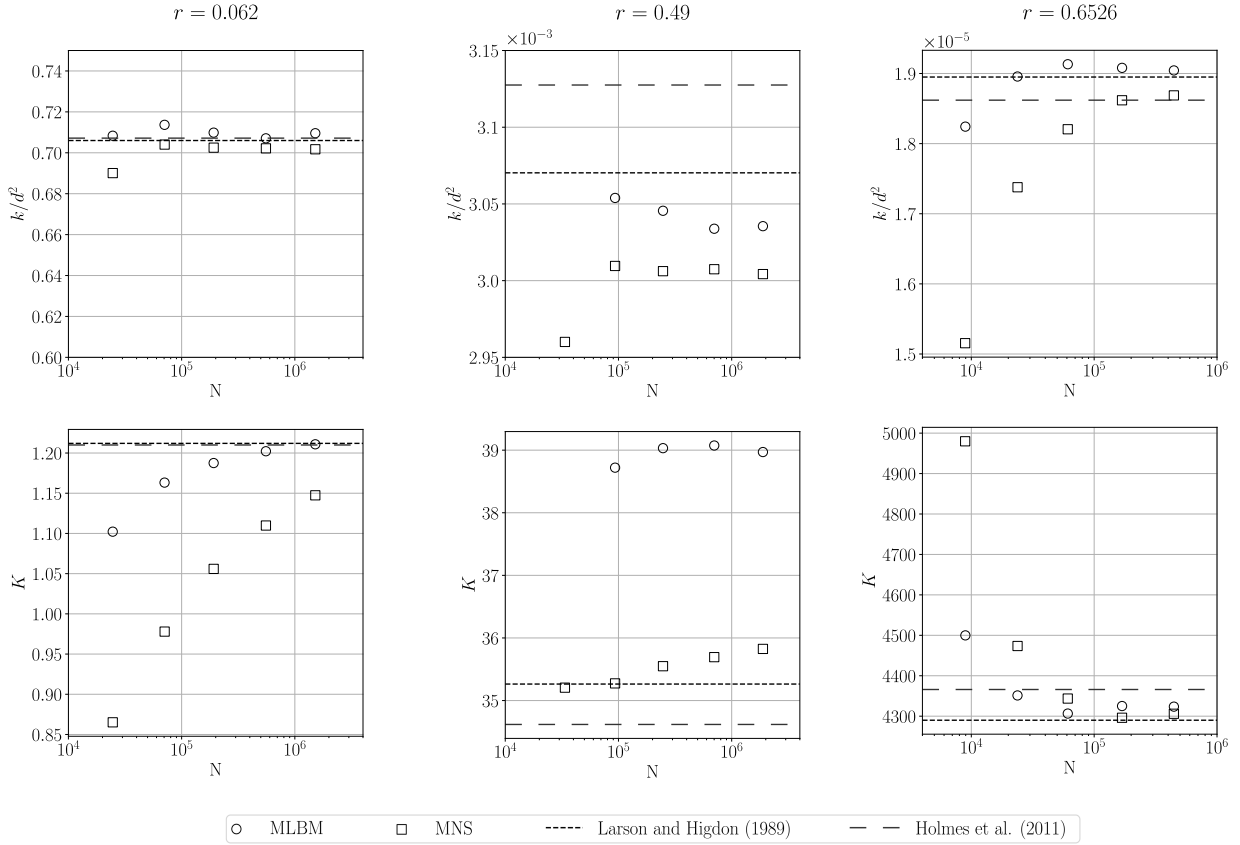


Figure 6. The convergence of the dimensionless permeability k/d^2 and drag coefficient K obtained with MLBM and MNS in 3D case for sphere radii $r = 0.062$ (left), $r = 0.49$ (middle) and $r = 0.6526$ (right). The horizontal dashed lines are the values obtained by Larson and Higdon [23] and Holmes et al. [22] (interpolated for $r = 0.49$).

The results are presented in Fig. 6 along with literature data [23, 22]. For $r = 0.49$ (near the touching limit) we interpolate the reference values of K from the above works. To calculate the reference k/d^2 we use the relation:

$$k/d^2 = \frac{1}{K} \left(\frac{d}{6\pi r} \right). \quad (34)$$

For $r = 0.062$, in the considered range of discretization parameters and for both tested methods, the values of permeability k/d^2 comply with the reference values within about 1% relative difference. The drag coefficient K converges towards the reference values in a stable manner.

For $r = 0.49$, both methods converge to values of the permeability slightly off the reference values ($< 3\%$ relative difference). The results of the drag force coefficient show a much higher discrepancy for MLBM (about 10% compared to the reference). We note that for MLBM the $h_{\min} = 0.02$ ($N \approx 3 \cdot 10^4$) case went unstable. To obtain stable solutions, the streaming distance needed to be changed to $\delta x = 0.005$ for $h_{\min} = 0.014$ and the stencil size to $N_L = 26$ for $h_{\min} = 0.01$ and to $N_L = 35$ for $h_{\min} = 0.007$. The possible causes of the instability are further investigated in Sec. 3.3.

When large spheres are considered, permeability, as well as the drag coefficient obtained with both methods, converge to values close to those from the references. However, bigger differences from the reference values for coarse discretizations are visible in MNS.

Concerning those results, one can assess the efficiency advantage of the use of space discretization refinement as follows. The work of Holmes and others[22] reports the internodal distance approximately equal to $\tilde{h} = 4.1 \cdot 10^{-3}$ for the porosity $\varphi = 0.1$ ($r = 0.6526$). Our results show that to achieve similar values of k/d^2 and K for this case we need to have $h_{\min} = 7 \cdot 10^{-3}$, which gives $N \approx 1.7 \cdot 10^5$ nodes. Assuming the least dense, simple cubic packing of points in Holmes and others' work one can estimate the lower bound for the number of nodes in their simulation as about an order of magnitude higher:

$$\tilde{N} = \frac{\varphi d^3}{\tilde{h}^3} = \frac{0.1 \cdot 1}{68.9 \cdot 10^{-9}} = 1.45 \cdot 10^6. \quad (35)$$

Further in the text, we show that the number of nodes in the meshless solutions can be further decreased with no significant rise in the error of K .

3.3 Permeability and drag coefficient for various porosities

Next, we calculate the permeability and drag coefficient in a wide range of solid volume fractions. We choose the range $r \in [0.04, 0.69]$. Fig. 7 shows the values of k/d^2 and K obtained for $h_{\min} = 0.01$ compared to the results of Holmes and others[22]. An excellent match between the two methods and the reference values is seen. This can be referred to the results of He and others [44] where steady-state LBM and ACM solutions of the velocity and the pressure field are compliant with each other. MLBM and MNS are capable of extending the analysis range beyond those investigated by the previous authors (denoted by red dashed lines) without any changes to the refinement function, Eq. (26). The meshless discretization with a proper refinement of the narrow pore throats could open the possibilities to consider even lower porosities. The lack of MLBM data at $\varphi = 0.4$ ($r = 0.528$) was caused by the divergence of the simulation.

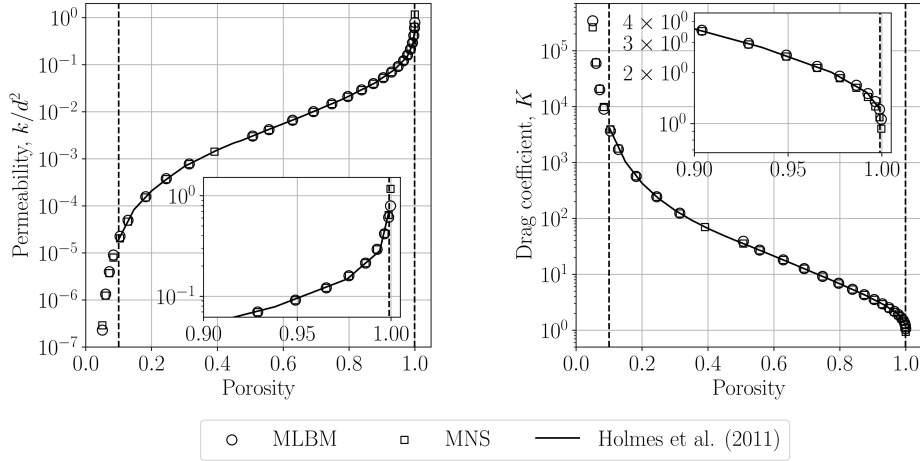


Figure 7. The values of the dimensionless permeability k/d^2 and drag coefficient K obtained with MLBM and MNS as a function of porosity. The solid line denotes the values obtained by Holmes et al. [22] in the range from $\varphi = 0.1$ to $\varphi = 0.999$ (indicated by vertical dashed lines).

To further investigate this issue we show the permeability and the drag coefficient values obtained for the near-touching limit cases ($r \approx 0.5$) in Fig. 8. Away from the $r = 0.5$ the values obtained with both methods are compliant with one another. For $r \rightarrow 0.5$ MNS faces no problems with the solution stability, nor does it exhibit irregular behavior of the coefficients' values. On the other hand, MLBM simulations go unstable for the cases $r = 0.5, 0.505$ and 0.510 . To obtain a stable solution for radii close to the problematic range we need to increase the stencil size up to $N_L = 40$ and change the rule for determining which populations of the boundary nodes are assumed unknown during the streaming. We no longer use the local normal vectors at the boundary nodes, rather we explicitly check which Lagrangian nodes lie inside the solid. Those altered MLBM setups are

denoted with filled symbols in Fig. 8. We attribute the difference in the behavior of the two methods in the touching limit case to the implementation of the no-slip boundary condition. In MNS, the Dirichlet boundary $\mathbf{v} = \mathbf{0}$ needs no approximation or information about the orientation of the local normal on the boundary, thus it introduces no error to the velocity field. MLBM faces growing approximation errors as the spheres get closer to each other and in the narrow volumes adjacent to the obstacle's intersections for $r > 0.5$ which might produce unbalanced stencils due to visibility criterion [45]. Those can be further amplified by too few discretization points across the narrow fluid volumes compared to the minimal number of nodes used in the literature in e.g. Poiseuille flow[46, 24].

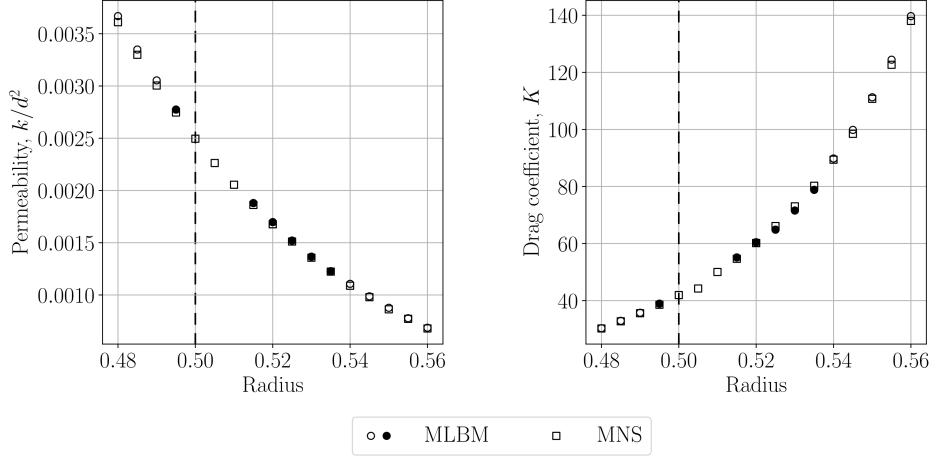


Figure 8. The values of the permeability (*left*) and the drag coefficient (*right*) in the near-touching limit cases. The filled symbols denote the MLBM cases where the bounceback criterion or the interpolation parameters needed to be changed in order to obtain stable simulations. The dashed line indicates the touching limit.

3.4 Influence of the refinement parameters and the stencil size

The advantage of meshless discretizations lies in the flexible positioning of nodes and easy manipulation of the approximation accuracy via the number of stencil members. First, we investigate the stability of the simulations and the relative errors of the drag coefficient K obtained with various combinations of ε and $h_{\max}/h_{\min}$ ratio (Eq. (26)) for small and large spheres. In general, lower values of ε and higher values of $h_{\max}/h_{\min}$ correspond to more aggressive refinement from the bulk towards the sphere's surface (Fig. 2). The relative error, defined as:

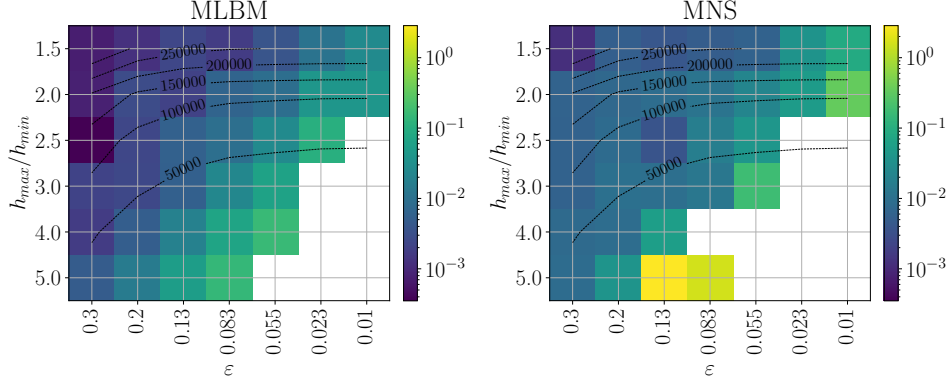
$$E(h_{\max}/h_{\min}, \varepsilon) = \frac{|K(h_{\max}/h_{\min}, \varepsilon) - K_{\text{ref}}|}{K_{\text{ref}}}, \quad (36)$$

is shown in Fig. 9. $K_{\text{ref}} \equiv K(1, 1)$ is the value of K for the constant nodes density in space. For all simulations, we used $h_{\min} = 0.01$. No error value is plotted for setups when the simulations diverged or when $E > 2$. On top of the errors map we plot isolines of several chosen numbers of nodes in the domain N . For comparison, in Fig. 9 we also report the number of nodes for the non-refined case, $N_{1,1}$.

For the considered refinement parameters, both methods show similar regions of stable solutions. This suggests that this is by large the meshless approximation stability/accuracy which determines the stability for both methods. The values of error E approximately follow the isocontours of the number of nodes with a noticeable drop of the error for large ε at a constant $h_{\max}/h_{\min}$ ratio in the case of $r = 0.062$. This is most probably due to a too small volume of fine discretization near the sphere to properly handle gradients in the relatively thick boundary layer. In this case, MLBM exhibits an extended area of stability than MNS. For the convergent cases of $r = 0.6526$ the error values are comparable between the methods. For $r = 0.062$, MNS exhibits a larger set of stable solutions for low $h_{\max}/h_{\min}$ and ε than MLBM, with higher values of errors in the range of medium values of those parameters. A comparison with the number of nodes in the non-refined case ($N_{1,1}$) reveals that the local refinement brings great advantages in the case of high-porosity sample - approximately 15-fold reduction of the node count comes with the cost of less than 1% error for $h_{\max}/h_{\min} = 4$, $\varepsilon = 0.3$ case.

Fig. 10 shows the obtained values of the drag coefficient for various stencil sizes N_L and refinement function shape parameters ε . In the case of MLBM, the value of K shows no significant dependence on the stencil size, regardless of the refinement aggressiveness. MNS can provide stable solutions with smaller stencil sizes but exhibits a stronger dependence of K on N_L . This can be explained by the fact that smaller ε , i.e. large gradients in node density, result in unbalanced stencils due to them being constructed from N_L closest nodes irrespective

$$r = 0.062, N_{1,1} = 828724$$



$$r = 0.6526, N_{1,1} = 89591$$

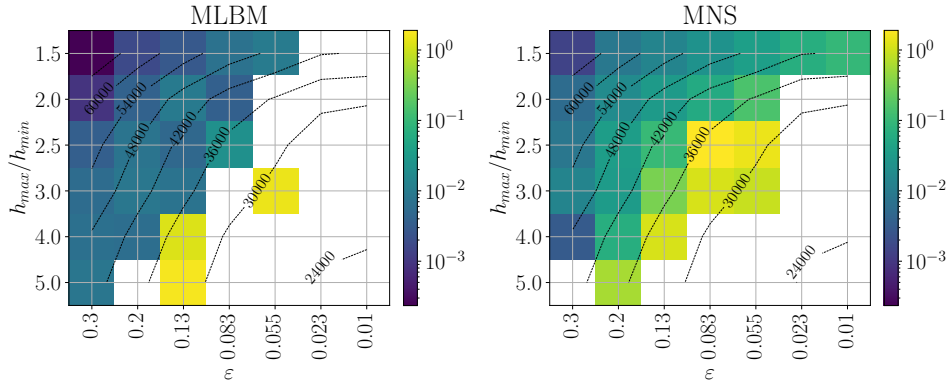


Figure 9. Relative error of the drag coefficient for MLBM and MNS solution (Eq. (36)) for various refinement function parameters. $N_{1,1}$ is the number of nodes in the non-refined discretization. Dashed lines with labels are isocontours of the number of Eulerian nodes N .

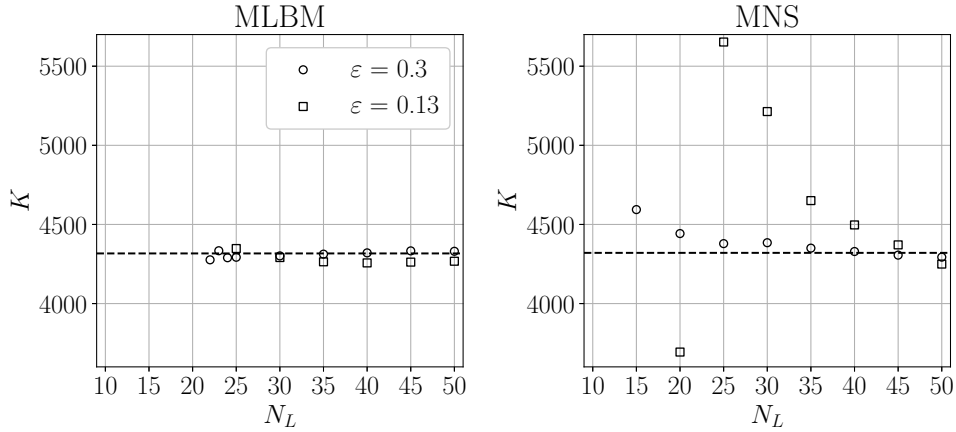


Figure 10. The values of the drag coefficient K for $r = 0.6526$, $h_{\max}/h_{\min} = 3$ case versus the stencil size N_L for various shape parameters ε for both solvers. The dashed lines are the reference $K_{\text{ref}} \equiv K(1, 1)$ values.

of direction. Unbalanced stencils are not as problematic for the RBF-FD method in value interpolation used by MLBM but can cause significant problems for 1st and 2nd-order derivative approximation used in MNS.

3.5 Pointwise comparison of the meshless solvers

We next investigate the pointwise relative difference between the velocity fields obtained with MLBM and MNS. A single simulation with $r = 0.062$ and $r = 0.6526$ was run with each solver using exactly the same space

discretization. We define the pointwise difference of the velocity field as:

$$\delta(\mathbf{x}) = \frac{|\mathbf{v}_{\text{MLBM}} - \mathbf{v}_{\text{MNS}}|}{|\mathbf{v}_{\text{MNS}}|} = \frac{\Delta v}{|\mathbf{v}_{\text{MNS}}|} \quad (37)$$

where the subscript denotes the method used to obtain the velocity value and $|\cdot|$ is the Euclidean vector norm. Fig. 11 shows isosurfaces of $\delta(\mathbf{x})$ for each sphere radius. For $r = 0.062$ the lowest difference isosurface corresponds to the boundary layer around the sphere, being stretched in the x -direction. The larger values of $\delta(\mathbf{x})$ towards the surface are caused by the normalization by near-zero velocity values (compare with the right subplot of Fig. 12 showing that the absolute value of the difference falls down towards the surface) with the maxima occurring in the same places as the maxima of the tangent force of MNS solution in Fig. 5. More drag in the MNS solution may explain the slightly lower x -velocity visible in the right plot of Fig. 12. In the case of $r = 0.6526$ the largest differences occur in the concave parts of the domain, near the spheres' intersections, where vortices are expected to emerge in the velocity field, even for a very low Reynolds number [47]. The fact that the main point-wise differences shown in Fig. 11 and Fig. 12 occur in areas with stagnant flow indicates that it may be caused by known ACM problems with pressure oscillations in the low Reynolds regime [48]. This is supported by the oscillations in the surface pressure field shown in Fig. 5. Oscillatory problems could be resolved by resorting to entropically dampened artificial compressibility [48], a more advanced form of the method.

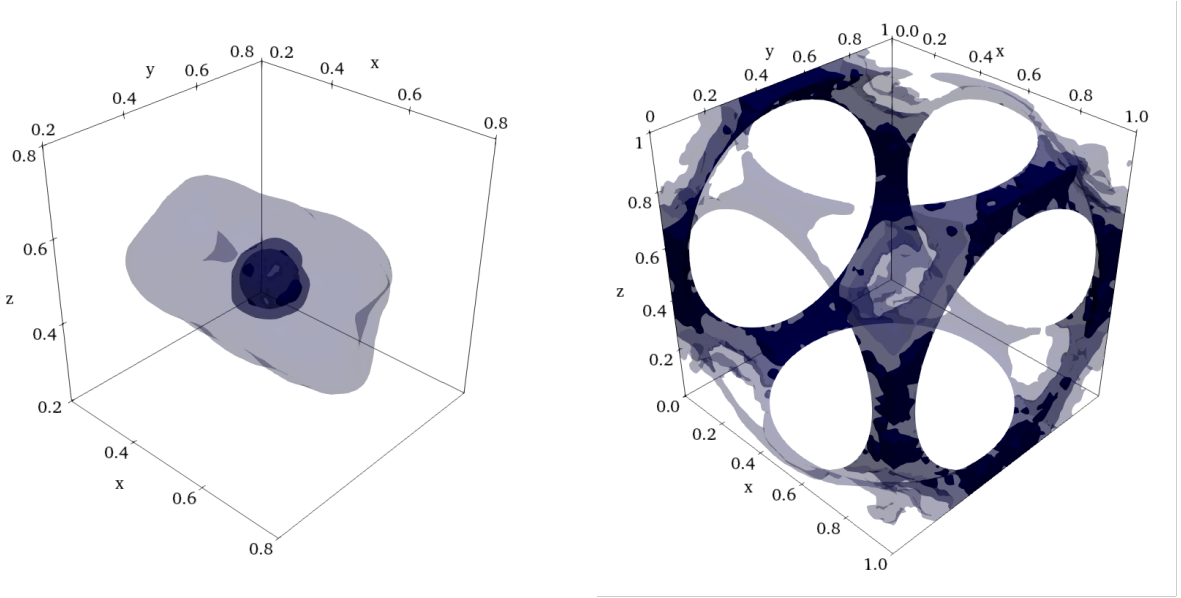


Figure 11. Relative pointwise velocity difference between the two solvers $\delta(\mathbf{x})$ for $r = 0.062$ (left) and $r = 0.6526$ (right). The isocontours correspond to the values: 0.1, 0.05, 0.01 for $r = 0.062$ and 1, 0.25, 0.05 for $r = 0.6526$, from the darkest to the brightest. Note the sphere surface is not rendered but the isosurfaces are trimmed by it.

3.6 Complexity and timings analysis

Algorithmical complexity

The cost of a single timestep of the described MLBM algorithm can be expressed as:

$$C_{\text{MLBM}} = N \left(\underbrace{q(56)}_{\text{I}} + \underbrace{+ 2N_L - 1}_{\text{II}} + \underbrace{q - 1 + 6q}_{\text{III}} \right) = N(q(62 + 2N_L) - 1) \quad (38)$$

where terms denoted by I, II, and III correspond to the collision, the interpolation, and the calculation of the macroscopic variables according to Eq. (17), respectively. In the case of the present MNS implementation, it is:

$$C_{\text{MNS}} = N \left(\underbrace{6(2N_L - 1) + 4}_{\text{I}} + \underbrace{+ n_p(4(2N_L - 1) + 4)}_{\text{II}} \right) = N(N_L(8n_p + 12) - 2) \quad (39)$$

where the terms denoted by I and II correspond to the update of the velocity field and the pressure correction steps, respectively. It is immediately seen that the number of the stencil members N_L contributes significantly

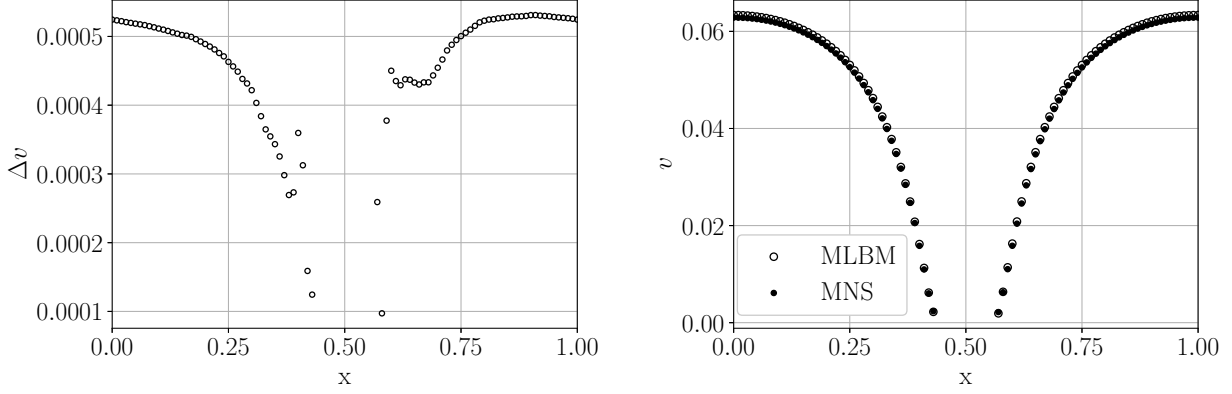


Figure 12. The absolute value of the difference of the velocity magnitude between MLBM and MNS Δv (*left*) and the values of the velocity magnitude for each method (*right*) along the line $y, z = 0.5$ for the $r = 0.062$ case.

to both values. In MLBM its contribution is multiplied by the number of discrete velocities q , while in MNS by terms corresponding to the number of macroscopic fields and pressure correction steps n_p . In MLBM, q multiplies also a large factor of 62 related to the calculation of the equilibrium distributions. In fact, the asymptotic complexity of each method is $\mathcal{O}(NN_L q)$ for MLBM (in line with the findings of e.g. Musavi and Ashrafizaadeh [19]) and $\mathcal{O}(NN_L n_p)$ for MNS. However, for the asymptotic regimes to be reached, extremely large values of N_L , q and n_p would need to be used. It is therefore more useful to consider the values of complexities for some certain, smaller values of those parameters. Fig. 13 shows the ratio of the two complexities for a constant number of pressure correction steps n_p and discrete velocities q . The complexity of both methods depends on the choice of the free parameters. For small stencils, the overhead from the calculation of equilibrium in MLBM makes it more complex than MNS.

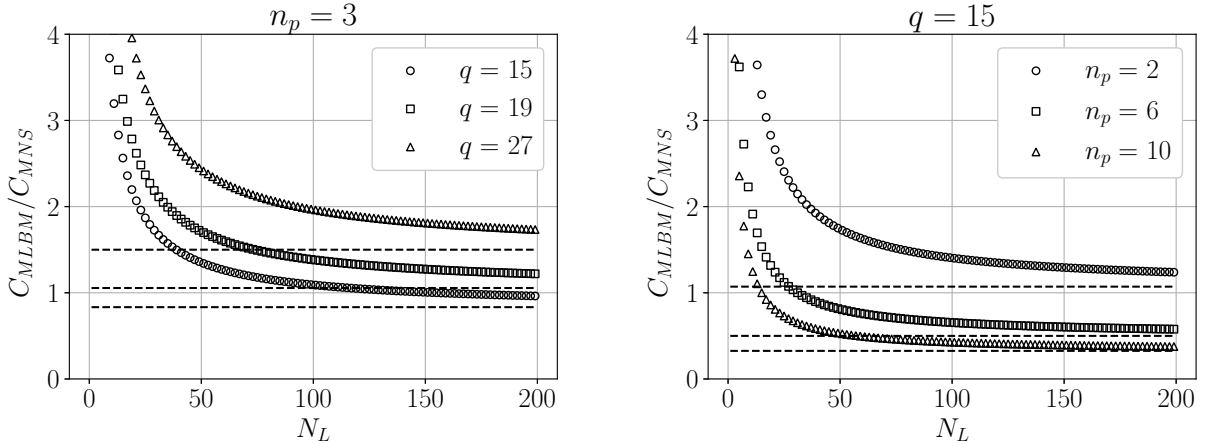


Figure 13. The ratio of MLBM complexity to MNS complexity defined by Eqs. (38) and (39) for three velocity discretizations used in MLBM (*left*) and for a range of pressure correction step numbers n_p (*right*). The horizontal dashed lines denote the asymptotes of the ratio for $N_L \rightarrow +\infty$.

It is important to notice that the size of the monomial subset of the interpolation basis does not influence the approximation step's complexity directly. It does, however, impose limits on the minimal number of nodes in interpolation supports, as noted in Section 2.1. The approximation step makes use of the weights vectors, Eq. (7), which need to be calculated only once, in the preprocessing step. The computational cost of this is of the order of $\mathcal{O}(N(N_L + m)^3)$ which has to be multiplied by the number of populations in MLBM or the number of approximated operators in MNS.

Timings

To provide a more practical quantification of the performance of the presented methods we measure the time needed to achieve a converged solution (defined as in the previous tests) with each of them. Fig. 14 shows the algorithms' execution times for the case of $r = 0.062$ and $r = 0.6526$ versus the number of nodes in the discretization N . We perform the analysis for single-threaded runs, observing that the scaling is of the order

1.5 for both methods and compliant with the theoretical predictions. In the large sphere case, the MNS needs approximately 3-folds more time to complete the calculations compared to the MLBM. We note, however, that the presented execution time implicitly contains the amount of the physical time needed for the system to reach steady-state. MLBM needed about 3.57s and 0.04s to reach the stopping criterion for $r = 0.062$ and $r = 0.6526$, respectively, at $h_{\min} = 0.007$. Those times for MNS are 3.58s and 0.39s. In the case of $r = 0.6526$, the difference of 0.35s in the physical time is what accounts for the larger execution times of MNS in Fig. 14. In case $r = 0.062$, where the physical time of the simulations is comparable between the methods, the estimated execution times were about three times larger for MLBM than for MNS. The issues of pressure oscillations mentioned in Sec.3.5 may explain the difference in the physical time required to reach a steady state in the $r = 0.6526$ case with large areas of stagnant flow.

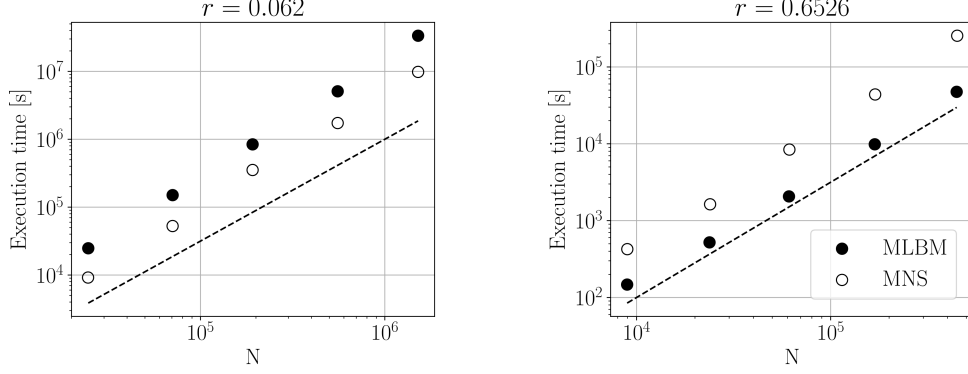


Figure 14. Execution times in seconds of $r = 0.062$ (left) and $r = 0.6526$ (right) case for single-threaded runs versus the number of nodes in the domain N . Open symbols denote meshless Navier-Stokes timings and filled symbols denote MLBM timings. The stopping criterion was the same as in the remaining tests. In both methods, the execution time is inversely proportional to the timestep length δt . In both methods the diffusive scaling of the timestep length was used ($\delta t \propto h(\mathbf{x})^2$ in MNS and $\delta t \propto \delta x^2 \propto h(\mathbf{x})^2$ in MLBM) and as $\delta x \propto N^{1/3}$ one obtains that the execution time should scale as $N^{1.5}$. The dashed lines denote the 1.5-order slope. The execution times for $r = 0.062$ were extrapolated based on the physical time at which the simulation reached the stopping criterion and the execution time for a fixed number of iterations.

4 Conclusions

The paper compares two meshless CFD solvers for the simulation of fluid flow through porous media, the meshless Lattice-Boltzmann method with two relaxation-time collision terms and a direct Navier-Stokes solver under the artificial compressibility limit, focusing on a periodic 3D porous medium modeled as a cubic array of spheres. Both methods utilize point clouds of the same point density in space, provided by an iterative, advancing-front dimension-independent algorithm that allows variable internodal distances (needed for h -refinement) and Radial Basis Function (RBF) approximation for the field and operator approximation. In terms of methods, the paper discusses the meshless discretization and approximation techniques as well as the specifics of the implementation of the considered CFD methods.

Through a convergence analysis of the permeability and drag coefficient analysis across various porosities, we demonstrate the effectiveness of both methods in accurately predicting the flow characteristics. Both methods are convergent and provide results that agree with reference data from the literature. This has encouraged us to extend the range of solvable problems beyond the available benchmark data to lower porosities.

Furthermore, we investigate the sensitivity of the drag coefficient to the refinement parameters, with both methods showing a similar stability range. We demonstrate the advantages of meshless refined discretization by significantly reducing the number of nodes with minimal increase in error.

When analyzing the effect of the stencil size, we find that the meshless Lattice-Boltzmann method achieves good results even with a relatively small stencil size compared to the direct solver. This is to some extent to be expected, since the Lattice-Boltzmann method uses the RBF approximation only for field approximation, while the direct solver approximates all differential operators involved.

The paper also delves into the complexity and timing analyses of both solvers. We find the complexity of both methods linearly dependent on the stencil size and the space discretization size with coefficients related to the equilibrium VDF form (LBM) and the number of pressure steps (MNS). Although the MNS is seemingly less complex in terms of the sheer number of operations, it requires a larger stencil size for stability reasons. Moreover, the actual execution times strongly depend on the number of iterations required to reach the steady

state, which is different for both methods for different porosities. The message here is that both methods are in the same order of magnitude in terms of computational complexity.

5 Acknowledgments

The authors acknowledge the financial support from the Slovenian Research and Innovation Agency (ARIS) research core funding No. P2-0095 and the Young Researcher programme PR-10468. Funded by National Science Centre, Poland under the OPUS call in the Weave programme 2021/43/I/ST3/00228. This research was funded in whole or in part by National Science Centre (2021/43/I/ST3/00228). For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

A MLBM and MNS algorithms comparison

MLBM algorithm

```

1: while the stopping criterion is not met do
2:   // COLLIDE
3:   for all Eulerian points  $x_i$  do
4:     for each lattice velocity  $c_k$  do
5:       Calculate  $f_k^{\text{eq}}$  (Eq. (15));
6:       Calculate  $f_k^{\text{post}}$  (Eq. (10));
7:     end for
8:   end for
9:   // STREAM
10:  for all Eulerian points  $x_i$  do
11:    for each Lagrangian point  $x_i + \delta x_k$  do
12:      Interpolate  $f_{k'}^{\text{post}}$  to  $x_i + \delta x_k$ 
(Eq. (2));
13:      Overwrite  $f_{k'}(x_i)$  with  $f_{k'}^{\text{post}}(x_i + \delta x_k)$  (Eq. (8));
14:    end for
15:    Update  $\rho_{lb}, v_{lb}$  (Eq. (17));
16:  end for
17: end while

```

MNS algorithm

```

1: while the stopping criterion is not met do
2:   // UPDATE VELOCITY
3:   for all computational points  $x_i$  do
4:     Calculate intermediate velocity  $v'$ 
(Eq. (22));
5:   end for
6:   // PRESSURE-VELOCITY COUPLING
7:   for a predetermined number of iterations do
8:     for all computational points  $x_i$  do
9:       Update pressure (Eq. (23));
10:    end for
11:    for all computational points  $x_i$  do
12:      Update velocity with the new pres-
sure gradient Eq. (24);
13:    end for
14:    Update  $C$  if required (Eq. (25));
15:  end for
16: end while
17:

```

References

- [1] A. F. Miguel, Lungs as a natural porous media: architecture, airflow characteristics and transport of suspended particles, in: Heat and mass transfer in porous media, Springer, 2011, pp. 115–137.
- [2] H. Whitesell, R. Overfelt, Influence of solidification variables on the microstructure, macrosegregation, and porosity of directionally solidified mar-m247, Materials Science and Engineering: A 318 (1-2) (2001) 264–276.
- [3] R. A. Patel, J. Perko, D. Jacques, G. De Schutter, G. Ye, K. Van Breugel, A three-dimensional lattice boltzmann method based reactive transport model to simulate changes in cement paste microstructure due to calcium leaching, Construction and Building Materials 166 (2018) 158–170.
- [4] A. K. Singh, R. K. Gupta, M. Kumar, K. Yadav, Review of finite element mesh generation methods, in: AIP Conference Proceedings, Vol. 2782, AIP Publishing, 2023.
- [5] J. Klostermann, R. Schwarze, C. Brückner, Meshing of porous foam structures on the micro-scale, Engineering with Computers 29 (2013) 95–110.
- [6] G.-R. Liu, Y.-T. Gu, An introduction to meshfree methods and their programming, Springer Science & Business Media, 2005.
- [7] V. P. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: a review and computer implementation aspects, Math. Comput. Simul 79 (3) (2008) 763–813. doi:10.1016/j.matcom.2008.01.003. URL <https://sourceforge.net/projects/elemfregalerkin/>

- [8] J. Slak, G. Kosec, Medusa: A C++ Library for solving PDEs using Strong Form Mesh-Free methods, *arXiv:1912.13282*.
- [9] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, *The finite element method: its basis and fundamentals*, Elsevier, 2005.
- [10] M. Depolli, J. Slak, G. Kosec, Parallel domain discretization algorithm for RBF-FD and other meshless numerical methods for solving PDEs, *Computers & Structures* 264 (2022) 106773. doi:10.1016/j.compstruc.2022.106773.
URL <https://linkinghub.elsevier.com/retrieve/pii/S0045794922000335>
- [11] D. T. Oanh, N. M. Tuong, An approach to adaptive refinement for the RBF-FD method for 2D elliptic equations, *Applied Numerical Mathematics* 178 (2022) 123–154. doi:10.1016/j.apnum.2022.03.015.
URL <https://linkinghub.elsevier.com/retrieve/pii/S0168927422000770>
- [12] Strong form mesh-free hp-adaptive solution of linear elasticity problem doi:10.1007/s00366-023-01843-6.
URL <https://link.springer.com/10.1007/s00366-023-01843-6>
- [13] A. I. Tolstykh, D. A. Shirobokov, On using radial basis functions in a “finite difference mode” with applications to elasticity problems, *Computational Mechanics* 33 (1) (2003) 68–79, publisher: Springer. doi:10.1007/s00466-003-0501-9.
- [14] V. Bayona, N. Flyer, B. Fornberg, G. A. Barnett, On the Role of Polynomials in RBF-FD Approximations: II. Numerical Solution of Elliptic PDEs, *Journal of Computational Physics* 332 (2017) 257–273. doi:10.1016/j.jcp.2016.12.008.
- [15] J. Slak, G. Kosec, Adaptive radial basis function-generated finite differences method for contact problems, *International Journal for Numerical Methods in Engineering* 119 (7) (2019) 661–686. doi:10.1002/nme.6067.
URL <https://onlinelibrary.wiley.com/doi/10.1002/nme.6067>
- [16] T. Chu, O. T. Schmidt, RBF-FD discretization of the Navier-Stokes equations on scattered but staggered nodes, *Journal of Computational Physics* 474 (2023) 111756, arXiv:2206.06495 [physics]. doi:10.1016/j.jcp.2022.111756.
URL <http://arxiv.org/abs/2206.06495>
- [17] P. P. Chinchapatnam, K. Djidjeli, P. B. Nair, M. Tan, A compact RBF-FD based meshless method for the incompressible Navier–Stokes equations, *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 223 (3) (2009) 275–290. doi:10.1243/14750902JEME151.
URL <http://journals.sagepub.com/doi/10.1243/14750902JEME151>
- [18] R. Zamolo, D. Miotti, E. Nobile, Accurate rbf-fd meshless numerical simulation of thermo-fluid problems for generic 3d geometries, in: *Journal of Physics: Conference Series*, Vol. 2509, IOP Publishing, 2023, p. 012002.
- [19] S. H. Musavi, M. Ashrafzaadeh, Meshless lattice boltzmann method for the simulation of fluid flows, *Physical Review E* 91 (2) (2015) 023310.
- [20] D. Strzelczyk, M. Matyka, Study of the convergence of the meshless lattice boltzmann method in taylor–green, annular channel and a porous medium flows, *Computers & Fluids* 269 (2024) 106122. doi:10.1016/j.compfluid.2023.106122.
URL <http://dx.doi.org/10.1016/j.compfluid.2023.106122>
- [21] J. Slak, G. Kosec, Medusa: A C++ Library for Solving PDEs Using Strong Form Mesh-free Methods, *ACM Transactions on Mathematical Software* 47 (3) (2021) 1–25. doi:10.1145/3450966.
URL <https://dl.acm.org/doi/10.1145/3450966>
- [22] D. W. Holmes, J. R. Williams, P. Tilke, Smooth particle hydrodynamics simulations of low reynolds number flows through porous media, *International Journal for Numerical and Analytical Methods in Geomechanics* 35 (4) (2011) 419–437. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nag.898>, doi:<https://doi.org/10.1002/nag.898>.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nag.898>

- [23] R. E. Larson, J. J. L. Higdon, A periodic grain consolidation model of porous media, *Physics of Fluids A: Fluid Dynamics* 1 (1) (1989) 38–46. [arXiv:https://doi.org/10.1063/1.857545](https://doi.org/10.1063/1.857545), doi:10.1063/1.857545.
URL <https://doi.org/10.1063/1.857545>
- [24] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Vigen, *The Lattice Boltzmann Method - Principles and Practice*, 2016. doi:10.1007/978-3-319-44649-3.
- [25] J. Slak, G. Kosec, On generation of node distributions for meshless PDE discretizations, *SIAM Journal on Scientific Computing* 41 (5) (2019) A3202–A3229. doi:10.1137/18M1231456.
- [26] O. Davydov, D. T. Oanh, N. M. Tuong, Improved stencil selection for meshless finite difference methods in 3d, *Journal of Computational and Applied Mathematics* 425 (2023) 115031. doi:<https://doi.org/10.1016/j.cam.2022.115031>.
- [27] E. Lehto, V. Shankar, G. B. Wright, A radial basis function (rbf) compact finite difference (fd) scheme for reaction-diffusion equations on surfaces, *SIAM Journal on Scientific Computing* 39 (5) (2017) A2129–A2151. [arXiv:https://doi.org/10.1137/16M1095457](https://doi.org/10.1137/16M1095457), doi:10.1137/16M1095457.
URL <https://doi.org/10.1137/16M1095457>
- [28] N. Flyer, B. Fornberg, V. Bayona, G. A. Barnett, On the role of polynomials in rbf-fd approximations: I. interpolation and accuracy, *Journal of Computational Physics* 321 (2016) 21–38. doi:<https://doi.org/10.1016/j.jcp.2016.05.026>.
- [29] J. Slak, Adaptive rbf-fd method, Ph.D. thesis, University of Ljubljana (2020).
URL <https://repozitorij.uni-lj.si/IzpisGradiva.php?lang=eng&id=121511>
- [30] M. Jančič, J. Slak, G. Kosec, Monomial augmentation guidelines for rbf-fd from accuracy versus computational time perspective, *Journal of Scientific Computing* 87 (1) (2021) 9. doi:10.1007/s10915-020-01401-y.
- [31] X. Lin, J. Wu, T. Zhang, A mesh-free radial basis function-based semi-Lagrangian lattice Boltzmann method for incompressible flows, *International Journal for Numerical Methods in Fluids* 91 (4) (2019) 198–211. doi:10.1002/flid.4749.
- [32] S. Succi, *The Lattice Boltzmann Equation: For Complex States of Flowing Matter*, Oxford University Press, 2018. doi:10.1093/oso/9780199592357.001.0001.
- [33] I. Ginzburg, Two-relaxation-time lattice boltzmann scheme: about parametrization, velocity, pressure and mixed boundary conditions, *Communications in Computational Physics* 3 (2008) 427–478.
- [34] I. Ginzburg, D. d’Humières, Multireflection boundary conditions for lattice Boltzmann models, *Physical Review E* 68 (6) (2003) 066614. doi:10.1103/PhysRevE.68.066614.
URL <https://link.aps.org/doi/10.1103/PhysRevE.68.066614>
- [35] W. Trojak, N. Vadlamani, J. Tyacke, F. Witherden, A. Jameson, Artificial compressibility approaches in flux reconstruction for incompressible viscous flow simulations, *Computers & Fluids* 247 (2022) 105634. doi:<https://doi.org/10.1016/j.compfluid.2022.105634>.
URL <https://www.sciencedirect.com/science/article/pii/S0045793022002304>
- [36] G. Kosec, A local numerical solution of a fluid-flow problem on an irregular domain, *Advances in Engineering Software* 120 (2018) 36–44, civil-Comp - Part 2. doi:<https://doi.org/10.1016/j.advengsoft.2016.05.010>.
URL <https://www.sciencedirect.com/science/article/pii/S0965997816301065>
- [37] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *Journal of Computational Physics* 2 (1) (1967) 12–26. doi:[https://doi.org/10.1016/0021-9991\(67\)90037-X](https://doi.org/10.1016/0021-9991(67)90037-X).
URL <https://www.sciencedirect.com/science/article/pii/002199916790037X>
- [38] Q. W. Ma, Y. Zhou, S. Yan, A review on approaches to solving poisson’s equation in projection-based meshless methods for modelling strongly nonlinear water waves, *Journal of Ocean Engineering and Marine Energy* 2 (3) (2016) 279–299. doi:10.1007/s40722-016-0063-5.
URL <https://doi.org/10.1007/s40722-016-0063-5>

- [39] T. Ohwada, P. Asinari, D. Yabusaki, Artificial compressibility method and lattice boltzmann method: Similarities and differences, *Computers & Mathematics with Applications* 61 (12) (2011) 3461–3474, mesoscopic Methods for Engineering and Science — Proceedings of ICMMES-09. doi:<https://doi.org/10.1016/j.camwa.2010.08.032>.
URL <https://www.sciencedirect.com/science/article/pii/S0898122110005961>
- [40] M. M. Rahman, T. Siikonen, An artificial compressibility method for viscous incompressible and low mach number flows, *International Journal for Numerical Methods in Engineering* 75 (11) (2008) 1320–1340. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2302>, doi:<https://doi.org/10.1002/nme.2302>.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2302>
- [41] J. Almedeij, Drag coefficient of flow around a sphere: Matching asymptotically the wide trend, *Powder Technology* 186 (3) (2008) 218–223.
- [42] B. Stahl, B. Chopard, J. Latt, Measurements of wall shear stress with the lattice boltzmann method and staircase approximation of boundaries, *Computers & Fluids* 39 (9) (2010) 1625–1633. doi:<https://doi.org/10.1016/j.compfluid.2010.05.015>.
URL <https://www.sciencedirect.com/science/article/pii/S0045793010001283>
- [43] M. Matyka, Z. Koza, Ł. Mirosław, Wall orientation and shear stress in the lattice boltzmann model, *Computers & Fluids* 73 (2013) 115–123.
- [44] X. He, G. D. Doolen, T. Clark, Comparison of the lattice boltzmann method and the artificial compressibility method for navier–stokes equations, *Journal of Computational Physics* 179 (2) (2002) 439–451.
- [45] T. Jacquemin, S. P. A. Bordas, A unified algorithm for the selection of collocation stencils for convex, concave, and singular problems, *International Journal for Numerical Methods in Engineering* 122 (16) (2021) 4292–4312. doi:[10.1002/nme.6703](https://doi.org/10.1002/nme.6703).
URL <https://onlinelibrary.wiley.com/doi/10.1002/nme.6703>
- [46] X. He, Q. Zou, L.-S. Luo, M. Dembo, Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice boltzmann bkg model, *Journal of Statistical Physics* 87 (1997) 115–136. doi:[10.1007/BF02181482](https://doi.org/10.1007/BF02181482).
- [47] M. Agnaou, D. Lasseux, A. Ahmadi, Origin of the inertial deviation from darcy’s law: An investigation from a microscopic flow analysis on two-dimensional model structures, *Phys. Rev. E* 96 (2017) 043105. doi:[10.1103/PhysRevE.96.043105](https://doi.org/10.1103/PhysRevE.96.043105).
URL <https://link.aps.org/doi/10.1103/PhysRevE.96.043105>
- [48] J. R. Clausen, Entropically damped form of artificial compressibility for explicit simulation of incompressible flow, *Phys. Rev. E* 87 (2013) 013309. doi:[10.1103/PhysRevE.87.013309](https://doi.org/10.1103/PhysRevE.87.013309).
URL <https://link.aps.org/doi/10.1103/PhysRevE.87.013309>