Computer Methods in Applied Mechanics and Engineering 419 (2024) 116655

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A new numerical *mesoscopic scale* one-domain approach solver for free fluid/porous medium interaction

Costanza Aricò [a,*], Rainer Helmig [b], Daniele Puleo [a], Martin Schneider [b]

[a] *Department of Engineering, University of Palermo, viale delle Scienze, Palermo, 90128, Italy*
[b] *Institute for Modelling Hydraulic and Environmental Systems (IWS), Department of Hydromechanics and Modelling of Hydrosystems, University of Stuttgart, Pfaffenwaldring 61, Stuttgart, D-70569, Germany*

ARTICLE INFO

Keywords:
Free fluid
Porous medium
Coupling
Mesoscopic scale
Anisotropy
Numerical scheme

ABSTRACT

A new numerical continuum *one-domain* approach (ODA) solver is presented for the simulation of the transfer processes between a free fluid and a porous medium. The solver is developed in the *mesoscopic* scale framework, where a continuous variation of the physical parameters of the porous medium (e.g., porosity and permeability) is assumed. The Navier–Stokes–Brinkman equations are solved along with the continuity equation, under the hypothesis of incompressible fluid. The porous medium is assumed to be fully saturated and can potentially be anisotropic. The domain is discretized with unstructured meshes allowing local refinements. A fractional time step procedure is applied, where one predictor and two corrector steps are solved within each time iteration. The predictor step is solved in the framework of a marching in space and time procedure, with some important numerical advantages. The two corrector steps require the solution of large linear systems, whose matrices are sparse, symmetric and positive definite, with $\mathcal{M}$-matrix property over Delaunay-meshes. A fast and efficient solution is obtained using a preconditioned conjugate gradient method. The discretization adopted for the two corrector steps can be regarded as a Two-Point-Flux-Approximation (TPFA) scheme, which, unlike the standard TPFA schemes, does not require the grid mesh to be $\mathbf{K}$-orthogonal, (with $\mathbf{K}$ the anisotropy tensor). As demonstrated with the provided test cases, the proposed scheme correctly retains the anisotropy effects within the porous medium. Furthermore, it overcomes the restrictions of existing mesoscopic scale one-domain approaches proposed in the literature.

1. Introduction

Momentum transfer at the interface between free fluid and porous media is of significance for various applications. Indeed, interface transport processes are involved in different industrial, environmental and biological/biomedical applications, as for example passive control devices using porous coating, heat exchangers, fuel cells, filtration and drying processes, groundwater pollution, flows in fractured media, geothermal systems, flows in biological tissues and related medical drugs transport problems. The study of fluid–porous interface momentum transfer is also crucial for the development of mathematical and numerical models involving additional transfer processes, e.g., passive solute or heat and pollutant transport.

The study of the fluid–porous interface transfer can be performed at different scales [1]. At the microscopic pore scale, the flow in the free fluid region and in the void spaces of the porous medium is governed by the classical (Navier)-Stokes equations along with boundary conditions at the interface between the fluid phase and the solid phase within the permeable region (e.g., no-slip

[*] Corresponding author.
E-mail address: costanza.arico@unipa.it (C. Aricò).

https://doi.org/10.1016/j.cma.2023.116655
Received 23 May 2023; Received in revised form 17 November 2023; Accepted 19 November 2023
Available online 22 November 2023
0045-7825/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

velocity condition). Such a pore scale approach has two limitations, (1) only small-scale problems can be simulated due to the high computational effort, caused by the discretization of the microscopic void spaces, which requires a significant number of mesh elements, and (2) often, detailed knowledge of the pore geometry in the entire domain is not known, even with advanced image acquisition technologies. This is the reason why descriptions at the mesoscopic and macroscopic scales are usually introduced according to two popular approaches, both derived from the volume averaging of pore-scale governing equations.

In the *two-domain* approach (TDA), at the macroscopic scale, the bulk fluid and porous regions are separated by a sharp interface. Two different sets of governing equations are applied in each of the bulk regions, namely the Darcy equation in the porous domain and the (Navier)-Stokes equations in the free fluid domain. Due to the different character of the corresponding partial differential equations, specific boundary conditions have to be imposed at the common interface that guarantee conservation of the fluxes together with appropriate slip conditions for tangential component of the free fluid velocity [2]. In the pioneering work of Beavers and Joseph [3], such a slip boundary condition was experimentally derived for parallel flow conditions when coupling Stokes with Darcy flow. Ochoa-Tapia and Whitaker [4,5] coupled the Stokes and Brinkman equations at the interface, assuming continuous tangential velocity but discontinuous tangential shear stress. Other interface boundary conditions have been obtained by using homogenization methods, see for example [6–8] and references therein. Several numerical techniques have been proposed in the literature for TDA. In [9], the authors solve the Navier–Stokes and Darcy equations in the framework of a finite element scheme. They decouple the problem and, at each time step, iteratively solve two subproblems via a suitable splitting of the interface condition. In [10] the authors present a locally conservative method by coupling mixed finite element methods for the Darcy region with Discontinuous Galerkin methods for the Stokes region. In [11] the authors couple mixed methods for the porous Darcy region with finite elements methods for the fluid Stokes region. They also prove the existence of weak solutions and the coupled problem is uncoupled into steps involving porous media and fluid flow subproblems. In [12], the Darcy–Stokes problem is solved via a non-overlapping domain decomposition method. The computational domain is divided into multiple subdomains where coupled local problems of lower complexity are solved. In [13,14] the authors present mortar multiscale numerical methods for coupling Stokes and Darcy flows with the Beavers–Joseph–Saffman interface condition. In [15], discontinuous Galerkin methods and mixed Finite Elements methods are applied in the Stokes and Darcy regions, respectively. Numerical challenges related to the coupling of Galerkin approximations of both Stokes and Darcy problems with mixed Finite Element formulations are discussed in [15]. Indeed, combining Raviart–Thomas Finite Element velocity spaces [16] with piecewise constant or linear pressure fields can satisfy the *inf-sup* conditions [15]. In [17], such discretization allows us to obtain the correct solution for the Darcy equation, but is not suitable for the Stokes problems. An alternative is to invoke the Darcy's law in the mass conservation equation, leading to an elliptic pressure Poisson problem, which can be easily approximated by Galerkin techniques. Unfortunately, this technique leads to a loss of accuracy for the velocity solution, as well as to a weak enforcement of the mass conservation equation [17].

In the continuum *one-domain* approach (ODA), a "fictitious" equivalent single medium replaces the fluid and solid phases, and one set of governing equations, valid everywhere in the domain, is used to model the transfer processes. At the mesoscopic scale, the transition from the bulk fluid to the bulk porous region is modeled using a transition zone (or transition layer, TL) located in between these two regions [1,18,19]. Within this transition layer the change of effective macroscopic properties of the permeable medium, such as porosity ϵ and permeability K, is modeled with appropriate continuous transition functions. The set of governing equations is often denoted as (Navier–)Stokes–Brinkman, (Navier–)Stokes–Darcy, Darcy–Brinkman or Brinkman equations [18,19]. These are derived by averaging the governing pore-scale equations over a Representative Elementary Volume (REV) [20,21]. The REV characteristic size is much smaller than that of the investigated domain, but much larger than the characteristic pore-scale size. In [22] the authors solve the transfer momentum problem in a three-layer 1D channel, including two external bulk homogeneous porous and fluid regions, separated by a heterogeneous transition zone, with variable ϵ and K. They give analytical velocity expressions in the three zones. In [23] the authors present an ODA based on the Stokes–Brinkman model. By using the method of matched asymptotic expansion, the asymptotic solution for vanishing transition layer thickness is investigated. In [24,25], an ODA is presented for 1D problems, where a macroscopic momentum equation, with Darcy form, applicable everywhere in the system, is solved along a homogenization closure problem, to obtain the transition layer permeability profile. The porous medium is assumed to be periodic and periodicity conditions are imposed for the closure problem. The ODA model is derived under the assumptions of constant pressure gradient and a given convective fluid velocity in the inertial term of the momentum equation. Most of the mesoscopic ODA models proposed in the recent literature are analytically solved under specific geometrical conditions and by assuming simplified boundary conditions, e.g., 1D flow, periodicity of the flow and the porous medium, steady-state flow or Stokes flow regime (low Reynolds number), (e.g., [22–25]).

Another macroscopic ODA approach uses penalization, such that the so-called *penalized* Navier–Stokes equations are applied, with an extra penalizing Darcy term in the momentum equation, which accounts for the drag force of the solid particles of the porous medium over the fluid [26–32]. The Darcy term is a function of the porosity and inverse of the permeability. This term is applied only within the porous region, while in the clear fluid region it vanishes, such that the classical (Navier)-Stokes equations are solved there. This implies that no continuous transition of porosity and permeability is assumed close to the free fluid–porous medium interface, but instead a discontinuous change is considered. These discontinuities induce an interfacial stress jump, and an additional stress arises within the porous medium due to the Darcy term in the governing momentum equations. Such penalized approaches have been widely applied in the literature since they adopt well-consolidated numerical procedures for the classical (Navier)-Stokes equations. Several numerical procedures have been proposed in the literature for the macroscopic ODA. Stabilized Finite Element methods (e.g., Galerkin/Least-Squares methods, Streamline-Upwind/Petrov–Galerkin or Pressure-Stabilizing/Petrov–Galerkin methods, Pressure Gradient Projection methods or Variational Multi-Scale methods) have been successfully applied either for Darcy or Navier–Stokes equations ([32] and the references therein).

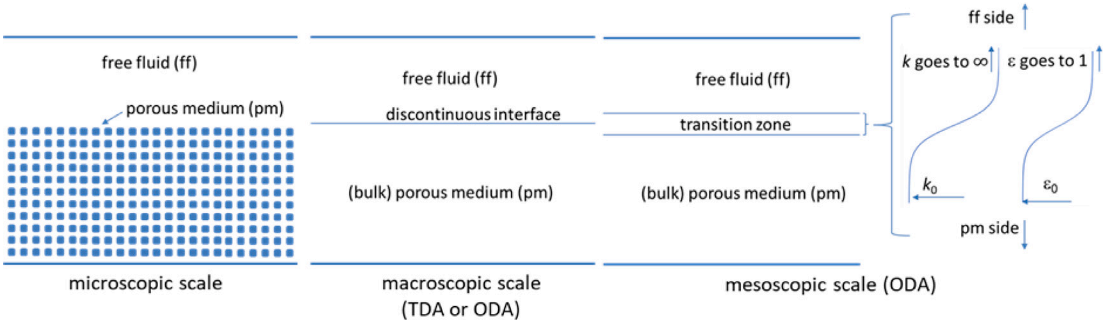


Fig. 1. Pore scale reference configuration (left) and different REV-scale approaches. Sharp interface approaches are depicted in the middle picture (TDA or macroscopic ODA), whereas the ODA that uses a transition zone is shown in the right picture. There, some possible transition functions are plotted, where k_0 and ϵ_0 denote the porosity and permeability values of the bulk porous medium.

These different approaches are schematically depicted in Fig. 1 together with the reference configuration on the pore scale.

Each of these approaches has advantages and drawbacks. If on the one hand the sets of governing equations in the TDA are well-known and consolidated numerical tools can be applied, a good match of the results near the interface is obtained when choosing appropriate interface coupling conditions [18]. Many applications present a gradual variation of the macroscopic properties (porosity, permeability, etc.) of the porous medium, without any abrupt change between the two bulk fluids and permeable regions [18], so the ODA should be more suitable. The principal limitation of the ODA is related to the difficulty in predicting the spatial variation of ϵ and K in the transition layer.

In the present paper, we propose a new numerical ODA solver at the mesoscopic level, where we do not restrict the porous medium to be isotropic, but also consider anisotropy. The proposed algorithm overcomes the restrictions of the mesoscopic ODA recently proposed in the literature, previously mentioned. This scheme has some important numerical features, which are further discussed in the following Sections. Another motivation is related to the significant discrepancies of the results provided by the present model and the outputs of the widely applied "penalized" macroscopic ODA in some of the presented applications in Section 4.

We solve the continuity and the Navier–Stokes–Brinkman equations, applying a fractional time step procedure, where a prediction and two correction problems are sequentially solved within each time iteration. The computational domain is discretized using unstructured meshes, allowing mesh refinement at the transition layer. The solution of the prediction problem is performed by a Marching in Space and Time (MAST) procedure. This is a Finite Volume algorithm, recently presented for the solution of shallow water and groundwater problems, as well as Navier–Stokes flow applications (see for example [33–37] and references therein). The two correction problems involve a fast and efficient solution of large linear systems since the associated matrices are sparse, symmetric, positive definite and diagonally dominant. As discussed in the following Sections, the algorithm proposed for the discretization of the correction problems can be regarded as a Two-Point-Flux Approximation (TPFA) scheme, which retains the anisotropic properties of the porous medium, but, unlike the standard TPFA scheme, it does not require the computational grid to be aligned with the principal anisotropy directions. The proposed method is *strongly* conservative, in the sense that the velocity solution is divergence free pointwise inside each mesh cell, and local and global mass balance are always guaranteed. The method is suitable for simulation of multi-dimensional unsteady flow problems.

The paper is organized as follows. The governing equations and the characteristics of the discretizing mesh are presented in Section 2, in Section 3 we provide the algorithmic details of the new ODA solver, and in Section 4 some numerical applications, including the analysis of the convergence order and the computational costs, as well as some "real-world" applications, are presented.

2. Governing equations

We assume a Newtonian incompressible fluid with density ρ_l inside and around a saturated porous medium, which is assumed to be rigid, with solid particles fixed in space. At the mesoscopic scale, the fluid and solid phases are described as a single continuous medium, derived by averaging the micro-scale Navier–Stokes equations over a REV [20,21]. This yields the following governing equations [1,19]

$$\nabla \cdot \left(\rho_l \mathbf{u}\right) = 0, \tag{1a}$$

$$\rho_l \frac{\partial \mathbf{u}}{\partial t} + \rho_l \mathbf{u} \cdot \nabla \left(\frac{\mathbf{u}}{\epsilon}\right) = -\epsilon \left(\nabla p^l - \rho_l \mathbf{g}\right) + \mu \nabla^2 \mathbf{u} - \mu \epsilon \mathfrak{K} \mathbf{u}, \tag{1b}$$

where t is time, $\mathbf{x}$ is the spatial coordinate vector, $\mathbf{u}$ is the surface average fluid velocity, with u and v its x and y components, p^l is the intrinsic averaged fluid pressure [19], $\mathbf{g}$ is the gravitational acceleration, downward oriented, with g the absolute value of its vertical component, μ is the dynamic fluid viscosity, ϵ is the porosity of the porous medium, and $\mathfrak{K}$ is the inverse of the permeability tensor of the porous medium, $\mathbf{K}$, symmetric and positive definite. The last term on the r.h.s. of Eq. (1b) represents a drag force due to the microscopic momentum exchange of the fluid with the solid particles of the permeable matrix. According to [4,19], it is related to μ, to the relative velocity between the fluid and the solid grains and to the permeability of the porous medium. In Eq. (1)

symbols $\nabla\cdot$ and ∇ mark the divergence of a vector or a tensor, and the gradient of a scalar quantity or a tensor, respectively, and symbol ∇^2 is the Laplacian operator.

Dividing Eqs. (1a) and (1b) by ρ_l and setting $\Psi = \frac{p^l}{\rho_l} - gy$, we obtain

$$\nabla \cdot \mathbf{u} = 0, \tag{2a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \left(\frac{\mathbf{u}}{\epsilon} \right) = -\epsilon \left(\nabla \Psi \right) + \nu \nabla^2 \mathbf{u} - \nu \epsilon \mathfrak{K} \mathbf{u}, \tag{2b}$$

with the kinematic fluid viscosity $\nu = \mu / \rho_l$. We solve the system (1) for the unknowns $\mathbf{u}$ and Ψ, in the computational domain Ω, and let Γ be its boundary. Three types of boundary conditions (BCs) can be assigned over $\Gamma = \Gamma_e \cup \Gamma_n \cup \Gamma_f$. Γ_e is the portion where we assign *essential* BCs (i.e., Dirichlet BCs for the velocity), Γ_n the portion where we assign *natural* BCs, (i.e., boundary stress vectors), and Γ_f the portion where we assign *free-slip* BCs, a combination of the previous ones. In Eq. (3a) we formulate the boundary and initial conditions (ICs) needed for the solution of system (2) to be well-posed.

$$\mathbf{u}\left(\mathbf{x}\right) = \mathbf{u}_b \qquad \forall \mathbf{x} \in \Gamma_e, \qquad t \geq 0, \tag{3a}$$

$$\left(\Psi - 2\nu \frac{\partial u_n}{\partial n} \right) = \sigma_n, \quad -\nu \left(\frac{\partial u_n}{\partial s} + \frac{\partial u_s}{\partial n} \right) = \sigma_s \qquad \forall \mathbf{x} \in \Gamma_n, \qquad t \geq 0, \tag{3b}$$

$$u_n = 0, \qquad -\nu \left(\frac{\partial u_s}{\partial n} \right) = 0 \qquad \forall \mathbf{x} \in \Gamma_f, \qquad t \geq 0, \tag{3c}$$

$$\mathbf{u} = \mathbf{u}_0 \qquad \text{with} \qquad \nabla \cdot \mathbf{u}_0 = 0, \qquad \Psi = \Psi_0 \qquad \forall \mathbf{x} \in \Omega, \qquad t = 0, \tag{3d}$$

where $\mathbf{u}_b$ is the velocity vector imposed over Γ_e, the directions $\mathbf{n}$ and $\mathbf{s}$ are normal (outward oriented) and tangent to the boundary, respectively, σ_n and σ_s are the normal and tangential components of the stress at the boundary, respectively, u_n and u_s are the components of $\mathbf{u}$ along $\mathbf{n}$ and $\mathbf{s}$, and sub-index 0 marks the initial values of $\mathbf{u}$ and Ψ in Ω. The kinematic pressure is prescribed indirectly via Eq. (3b). The viscous component of σ_n and σ_s in Eq. (3b) are set to zero.

3. Numerical algorithm

In Section 3.1 we present a general overview of the proposed algorithm, while we refer to Section 3.2 those readers interested in the numerical details of the algorithm steps.

3.1. General algorithm overview

System (2) is solved by applying a fractional time step procedure, where one predictor and two corrector problems are solved sequentially.

Using the splittings

$$\nabla \Psi = \nabla \Psi^k + \nabla \Psi - \nabla \Psi^k, \tag{4}$$

$$\nu \nabla^2 \mathbf{u} = \nu \nabla^2 \mathbf{u}^{k-1/3} + \nu \nabla^2 \mathbf{u} - \nu \nabla^2 \mathbf{u}^{k-1/3}, \tag{5}$$

after simple manipulations, setting $\mathbf{m} = \nu\epsilon\mathfrak{K}$ and $\mathbf{M}_0 = \mathbf{I} + \mathbf{m}\Delta t$ (with $\mathbf{I}$ the identity matrix), the time discretization form of Eq. (2b) becomes

$$\mathbf{M}_0 \frac{\left(\mathbf{u}^{k+1/3} - \mathbf{u}^k\right)}{\Delta t} + \mathbf{u}^{k+1/3} \cdot \nabla \left(\frac{\mathbf{u}^{k+1/3}}{\epsilon} \right) + \epsilon \nabla \Psi^k - \nu \nabla^2 \mathbf{u}^{k-1/3} + \mathbf{m}\mathbf{u}^k = 0, \tag{6a}$$

$$\mathbf{M}_0 \frac{\left(\mathbf{u}^{k+2/3} - \mathbf{u}^{k+1/3}\right)}{\Delta t} - \nu \nabla^2 \mathbf{u}^{k+2/3} + \nu \nabla^2 \mathbf{u}^{k-1/3} = 0, \tag{6b}$$

$$\mathbf{M}_0 \frac{\left(\mathbf{u}^{k+1} - \mathbf{u}^{k+2/3}\right)}{\Delta t} + \epsilon \nabla \Psi^{k+1} - \epsilon \nabla \Psi^k = 0. \tag{6c}$$

where Eq. (6a) is the predictor problem (PP) and Eqs. (6b) and (6c) are the 1st and 2nd corrector problems (CP1 and CP2), respectively. In the following Sections, the symbols t^k, $t^{k+1/3}$, $t^{k+2/3}$ and t^{k+1} mark the beginning of the time step, as well as the end of PP, CP1 and CP2, respectively. The symbol $t^{k-1/3}$ marks the end of the CP2 of the previous time step. In this sense, $t^{k+1/3}$ and $t^{k+2/3}$ are the first and second approximations for the full time step of the current time iteration, respectively, and $t^{k-1/3}$ is the second approximation for the full time step of the previous time iteration. $(\cdot)^k, (\cdot)^{k+1/3}, (\cdot)^{k+2/3}, (\cdot)^{k+1}, (\cdot)^{k-1/3}$ are the values of variable $(\cdot)$, (with $(\cdot) = \mathbf{u}$ or Ψ) at the corresponding time levels.

We discretize the domain Ω using unstructured triangulations Ω_T of N_T non-overlapping triangles and N nodes. The computational mesh satisfies the extended Delaunay property as defined in [37] (see Fig. 1 in the referred paper), which can always be obtained in the 2D case (see [38] and literature therein). The reason why we use Delaunay meshes will be explained in the following Sections. A triangle e is called the (computational) cell or (computational) element and the triangle side is called the (element) interface.

In the present paper, we specifically adapt the procedure proposed in [36,37] to account for the modified system (1)–(2) of governing equations, compared to the classical Navier–Stokes equations.

Inside each triangle e the velocity vector computed at the end of each time iteration $\mathbf{u}_e^{k+1}$ (which is equal to the vector at the beginning of the next time iteration) is assumed $\mathbf{u}_e^{k+1} \in \Re_e$, where $\Re_e$ is the lowest-order Raviart–Thomas (**RT0**) space function [16], whose basic properties are briefly summarized in Appendix A. Thanks to the **RT0** properties, the velocity components are piecewise constant inside each triangle e if $\sum_{j=1}^{3} Q_j^e = 0$ (where Q_j^e is the normal flux crossing side j of e, positive outward, i.e., one of the three DOFs of the $\Re_e$ space). If this condition is satisfied, $\nabla \cdot \mathbf{u}_e = 0\ \forall \mathbf{x} \in e,\ \forall e \in \Omega_T$, and, if the normal fluxes of two neighboring triangles are equal in value and opposite in sign along the common side, both local and global mass continuity are preserved.

The kinematic pressure Ψ is assumed to be piecewise linear inside each triangle e according to the nodal values, as explained in [36,37].

Below we give the outline of the proposed algorithm, whose numerical details are reported in Section 3.2. In Fig. 2 we show the sequence of the algorithm steps, along with the associated representation of the velocity field within any cell $e \in \Omega_T$.

- Beginning of each time iteration (time level t^k). $\mathbf{u}_e^k$ is the solution $\mathbf{u}_e^{k+1}$ of the previous time iteration. $\mathbf{u}_e^k \in \Re_e$, the corresponding normal fluxes are continuous along each element interface and $\nabla \cdot \mathbf{u}_e = 0,\ \forall \mathbf{x} \in e,\ \forall e \in \Omega_T$.
- Solution of the PP (from time level t^k to time level $t^{k+1/3}$). After integration in space over each mesh element of Eq. (6a), PP is solved in its time integral form. A "local element update" of the velocity field is performed, by computing a piecewise constant correction $\Delta\hat{\mathbf{u}}$ of $\mathbf{u}_e$ within each e (see Fig. 2). This disrupts the continuity of the normal fluxes (DOFs of $\Re_e$) at each element interface. At the end of PP (time level $t^{k+1/3}$) $\mathbf{u}^{k+1/3}$ is piecewise constant, but does not satisfy local and global mass balance. Numerical details of this prediction problem are presented in Section 3.2.1.
- Solution of the CP1 (from time level $t^{k+1/3}$ to time level $t^{k+2/3}$). After integration in space over each mesh element, we solve Eq. (6b) in its differential form. Starting from the solution $\mathbf{u}_e^{k+1/3}$, we perform, $\forall e \in \Omega_T$, a "local element update" of the velocity field by computing a second piecewise constant correction $\Delta\tilde{\mathbf{u}}$ of $\mathbf{u}_e$ within each e (see Fig. 2). As for the PP, at the end of CP1 (time level $t^{k+2/3}$) $\mathbf{u}^{k+2/3}$ is piecewise constant, but does not satisfy local and global mass balance. After the solution of CP1, before CP2, (time level $t^{k+1/3}$) we re-establish the normal flux continuity at each element interface, by averaging the fluxes computed according to the velocity field $\mathbf{u}^{k+2/3}$ in the two elements sharing the side. The averaged normal fluxes do not satisfy the mass balance since $\mathbf{u}^{k+2/3}$ has been obtained by solving, during the PP and CP1, momentum equations only. The **RT0** velocity vector, $\mathbf{u}_{RT0}^{k+2/3}$, associated to the continuous normal fluxes is piecewise linear within each cell but not divergence free (see Fig. 2). Numerical details of the CP1 are given in Section 3.2.2.
- Solution of the CP2 (from time level $t^{k+2/3}$ to time level t^{k+1}). After integration in space over each mesh element, we solve Eq. (6c) in its differential form. We re-establish local and global mass balance by adding corrective normal fluxes to those computed at the end of CP1. These corrective fluxes are calculated to impose the divergence free condition of the final velocity $\mathbf{u}^{k+1}$. After the solution of CP2, $\mathbf{u}_e^{k+1} \in \Re_e$ and it is piecewise constant within each cell e (see Fig. 2). Numerical details of the CP2 are presented in Section 3.2.3.

The prediction problem is solved by applying a Finite Volume MArching in Space and Time procedure (MAST). As mentioned in the Introduction, this has already been applied in other contexts. One of the main advantages of this procedure is that it performs a sequential solution of small Ordinary Differential Equations (ODEs) systems, one for each computational cell. This allows an "explicit handling" of the non-linear convective inertial momentum terms in Eq. (1b), (i.e., the convective inertial terms within each cell are updated in the time interval $\left[t^k, t^{k+1/3}\right]$ separately from the other cells) avoiding the solution of large systems with non-symmetric matrices as in other numerical schemes, e.g., [33–37,39]. The MAST procedure has shown numerical stability for Courant–Friedrichs–Levy (CFL) numbers greater than 1 (see [33–37] and references therein).

In both corrector problems, large linear systems of dimension N_T are solved, with sparse, symmetric, and, if the Delaunay mesh property holds, also positive definite matrices. This ensures that the system matrices are $\mathcal{M}$-matrices, which avoids nonphysical oscillations in the numerical solution [40]. We apply a mass lumping procedure, similar to the one proposed in [41] for Mixed Hybrid Finite Element, which is well-suited if the Delaunay mesh property holds [36,37]. The coefficients of the matrices of CP1 and CP2 are constant in time, which makes computations efficient, since their assembly and factorization are performed only once, before the beginning of the time loop.

3.2. Numerical details of the algorithm steps

In the following Sections, A_e marks the area of element e, ep is one of the neighboring elements of e, and the common side is marked as j and jp in the local numeration of e and ep, respectively ($j,\ jp = 1,\ 2,\ 3$). The symbols $\bar{(\cdot)}_e$ and $\bar{(\cdot)}_j$ denote the spatial average of variable $(\cdot)$ inside each element e and along side j, respectively, computed according to the nodal values of $(\cdot)$. If $(\cdot)$ is a tensor, the symbols $\bar{(\cdot)}_e$ and $\bar{(\cdot)}_j$ denote the average values of the tensor coefficients, computed according to the coefficients in the three nodes of triangle e and the two nodes of side j, respectively.

3.2.1. Predictor problem

Integrating in space Eq. (6a) over each element $e \in \Omega_T$ and left-multiplying by matrix $\bar{\mathbf{M}}_0^{-1}$, we obtain

$$\begin{aligned} &\frac{\left(\mathbf{u}^{k+1/3} - \mathbf{u}^k\right)}{\Delta t} A_e + \bar{\mathbf{M}}_0^{-1} \int_{A_e} \left(\mathbf{u}^{k+1/3} \cdot \nabla \left(\frac{\mathbf{u}^{k+1/3}}{\epsilon}\right)\right) \mathrm{d}A + \\ &\bar{\mathbf{M}}_0^{-1} \int_{A_e} \left(\epsilon \nabla \Psi^k - \nu \nabla^2 \mathbf{u}^{k-1/3} + \mathbf{m}\mathbf{u}^k\right) \mathrm{d}A = 0 \end{aligned} . \tag{7}$$

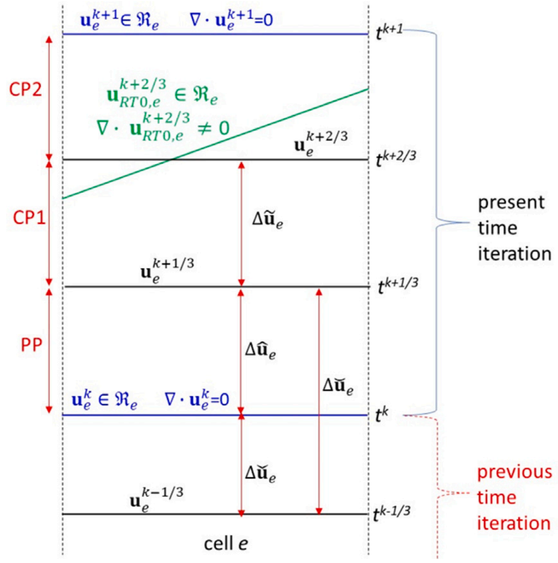

$\mathbf{u}_e^{k+1} \in \Re_e$ $\nabla \cdot \mathbf{u}_e^{k+1}=0$
t^{k+1}
CP2
$\mathbf{u}_{RT0,e}^{k+2/3} \in \Re_e$
$\nabla \cdot \mathbf{u}_{RT0,e}^{k+2/3} \neq 0$
$\mathbf{u}_e^{k+2/3}$
$t^{k+2/3}$
CP1
$\Delta\tilde{\mathbf{u}}_e$
present
time
iteration
$\mathbf{u}_e^{k+1/3}$
$t^{k+1/3}$
PP
$\Delta\hat{\mathbf{u}}_e$
$\Delta\breve{\mathbf{u}}_e$
$\mathbf{u}_e^{k} \in \Re_e$ $\nabla \cdot \mathbf{u}_e^{k}=0$
t^{k}
$\Delta\breve{\mathbf{u}}_e$
previous
time
iteration
$\mathbf{u}_e^{k-1/3}$
$t^{k-1/3}$
cell e

Fig. 2. 1D sketch of the velocity update within cell e during the algorithm steps for the current time iteration.

According to the MAST approach, the scalar momentum equations along the x and y directions are solved inside each triangle e separately from those of the other cells. This is possible if at the beginning of each time step (time level t^k) we perform a sorting operation of all the cells in the domain according to the direction of the velocity vector $\mathbf{u}^k$. This is a fast operation which always allows the cell sorting, even if recirculations occur in the flow field (see details in Section 3.1 and Appendix 1 of [36]). At the end of the sorting operation, a rank R_e is assigned to each cell, such that $R_e > R_{ep}$, with R_{ep} the rank of any neighboring ep triangles, whose common side shared with e is crossed by a flux entering e from ep.

The solution of Eq. (7) depends on the pressure gradient and viscous and drag forces obtained from the previous time step, and, thanks to the sorting operation, on the incoming momentum flux from neighboring cells ep with $R_e > R_{ep}$. These are the reasons why Eq. (7) can be solved within the interval $[0 - \Delta t]$ (e.g. from t^k to $t^{k+1/3}$), as a system of two Ordinary Differential Equations (ODEs) for the u and v unknowns, $\forall e \in \Omega_T$ [33–37].

As specified in Section 3.1, at time level t^k velocity $\mathbf{u}_e^k \in \Re_e$ is divergence free. $\forall e \in \Omega_T$ we define a piecewise constant velocity vector correction $\Delta\hat{\mathbf{u}}_e$, such that

$$\mathbf{u}_e(t) = \mathbf{u}_e^k + \Delta\hat{\mathbf{u}}_e(t) \quad 0 \le t \le \Delta t \quad \text{with} \quad \Delta\hat{\mathbf{u}}_e(0) = 0. \tag{8}$$

Applying the divergence theorem to the second term on the l.h.s. of Eq. (7),

$$\int_{A_e} \left(\mathbf{u}^{k+1/3} \cdot \nabla \left(\frac{\mathbf{u}^{k+1/3}}{\epsilon} \right) \right) \mathrm{d}A = \oint_{L_e} \frac{\mathbf{u}^{k+1/3}}{\epsilon} \left(\mathbf{u}^{k+1/3} \cdot \mathbf{n} \right) \, \mathrm{d}l, \tag{9}$$

where $\oint_{L_e}$ represents the integral over the three sides of triangle e, we rewrite the momentum equilibrium equation as

$$\begin{aligned} &\frac{d\left(\Delta\hat{\mathbf{u}}_e\right)}{dt} A_e + \sum_{j=1}^{3} \phi_j \bar{\mathbf{M}}_{0,j}^{-1} \mathfrak{M}_j^{e,out}(t) + \sum_{j=1}^{3} \left(1 - \phi_j\right) \bar{\mathbf{M}}_{0,j}^{-1} \bar{\mathfrak{M}}_{0,j}^{e,in}(t) + \\ &\qquad \bar{\mathbf{M}}_{0,e}^{-1} \left(\mathfrak{S}_{\Psi,e}^{k} + \mathfrak{V}_e^{k-1/3} + \bar{\mathbf{m}}_e \mathbf{u}_e^k \right) A_e = 0 \end{aligned}, \tag{10}$$

where $\mathfrak{M}_j^{e,out}$ is the leaving momentum flux from side j of e to ep with $R_{ep} > R_e$,

$$\mathfrak{M}_j^{e,out}(t) = l_j^e \frac{\mathbf{u}_e(t)}{\bar{\epsilon}_j} \max\left(0, \mathbf{u}_e(t) \cdot \mathbf{n}_j\right), \tag{11}$$

and l_j^e is the length of side j. $\bar{\mathfrak{M}}_j^{e,in}$ in Eq. (10) is the mean in time value of the incoming momentum flux crossing side j, oriented from ep to e, with $R_{ep} < R_e$, known from the solution of the previously solved cells and computed as specified in Section 3.1 in [36], $\phi_j = 1$ if e shares its side j with triangles ep with $R_{ep} > R_e$, or if it is a boundary side with leaving momentum flux, in the opposite case $\phi_j = 0$. $\mathfrak{V}_e^{k-1/3}$ and $\mathfrak{S}_{\Psi,e}^{k}$ are the sum of viscous and kinematic pressure forces, respectively, computed in cell e during the previous time step, as better specified in Sections 3.2.2 and 3.2.3.

Eq. (10) is called *MAST forward step*. The ODEs systems (10) are sequentially solved, one for each cell. We apply a Runge–Kutta method with adjustable time-step size within the interval $[0, \Delta t]$ [42]. We proceed, in the sequential solution, from the triangles with the smallest rank to the triangles with the highest rank. The direction of $\mathbf{u}_e$ could change within $[0 - \Delta t]$ during the solution of system (10), and we could compute momentum fluxes going from e to ep with $R_{ep} < R_e$. These momentum fluxes are neglected during the *MAST forward step*. To restore the momentum balance, after the *MAST forward step*, we perform a *MAST backward step* proceeding from the cells with the highest rank to the cells with the lowest rank. During the *MAST backward step* only the inertial terms are retained,

$$\frac{d\left(\Delta\hat{\mathbf{u}}_e\right)}{dt}A_e + \sum_{j=1}^{3}\ \left(1-\phi_j\right)\bar{\mathbf{M}}_{0,j}^{-1}\mathfrak{M}_j^{e,out}\left(t\right)\ + \sum_{j=1}^{3}\ \phi_j\bar{\mathbf{M}}_{0,j}^{-1}\bar{\mathfrak{M}}_j^{e,in}\left(t\right) = 0. \tag{12}$$

The initial solution of the *MAST backward step* is the final solution computed at the end of the *MAST forward step*. The boundary conditions of the predictor problem are assigned as specified in [36]. As mentioned in Section 3.1, the "local element update" by the computation of $\Delta\hat{\mathbf{u}}_e$ $\forall e \in \Omega_T$ disrupts the continuity of the normal fluxes at each element interface, and velocity $\mathbf{u}^{k+1/3}$ is not divergence free.

3.2.2. 1st corrector problem

Integrating Eq. (6b) in space over each element $e \in \Omega_T$ we get

$$\bar{\mathbf{M}}_e\left(\mathbf{u}^{k+2/3} - \mathbf{u}^{k+1/3}\right)A_e = \nu\int_{A_e}\nabla^2\mathbf{u}^{k+2/3}\mathrm{d}A_e - \nu\int_{A_e}\nabla^2\mathbf{u}^{k-1/3}\mathrm{d}A_e, \tag{13}$$

where matrix $\mathbf{M} = \mathbf{M}_0/\Delta t$, with matrix $\mathbf{M}_0$ defined in Section 3.1. Applying the divergence theorem to the integrals on the r.h.s. of Eq. (13), we get Eq. (14), which forms a system to solve from $t^{k+1/3}$ to $t^{k+2/3}$ for the u and v unknowns $\forall e \in \Omega_T$,

$$\bar{\mathbf{M}}_e\left(\mathbf{u}^{k+2/3} - \mathbf{u}^{k+1/3}\right)A_e = \nu\oint_{L_e}\frac{\partial\mathbf{u}_e^{k+2/3}}{\partial n}dl\ - \nu\oint_{L_e}\frac{\partial\mathbf{u}_e^{k-1/3}}{\partial n}dl. \tag{14}$$

$\frac{\partial\mathbf{u}}{\partial n}$ is the derivative of the velocity vector along the orthogonal direction to the boundary L_e of triangle e and the other symbols have been already specified. As mentioned at the end of Section 3.1, inside each triangle e, we apply a mass-lumping Mixed Hybrid Finite Element procedure to solve system (14), setting (see [36,37] and literature therein and [41])

$$\nu\oint_{L_e}\frac{\partial\mathbf{u}_e^*}{\partial n}dl\ = \nu\sum_{j=1}^{3}\frac{\left(\mathbf{u}_e^* - \mathbf{u}_{ep}^*\right)}{d_{e,ep}}l_j^e \qquad * = \begin{cases} k+2/3 \\ k-1/3, \end{cases} \tag{15}$$

where $d_{e,ep} = \left|\mathbf{x}_e - \mathbf{x}_{ep}\right|sign$, $\left|\mathbf{x}_e - \mathbf{x}_{ep}\right|$ is the distance between the circumcenters of triangles e and ep and $sign = 1$ or -1 depending on whether or not the mesh satisfies the Delaunay property [36,37]. The other symbols have been previously specified. $\forall e \in \Omega_T$ we introduce two new piecewise constant vectors $\Delta\tilde{\mathbf{u}}_e$ and $\Delta\check{\mathbf{u}}_e$, such that

$$\mathbf{u}_e^{k+2/3} - \mathbf{u}_e^{k+1/3} = \Delta\tilde{\mathbf{u}}_e, \qquad \mathbf{u}_e^{k+1/3} - \mathbf{u}_e^{k-1/3} = \Delta\check{\mathbf{u}}_e, \tag{16}$$

where $\Delta\check{\mathbf{u}}_e = \Delta\hat{\mathbf{u}}_e + \Delta\check{\mathbf{u}}_e$, with $\Delta\check{\mathbf{u}}_e = \mathbf{u}_e^k - \mathbf{u}_e^{k-1/3}$ is known. After some manipulations, system (14) can be written as

$$\bar{\mathbf{M}}_e\Delta\tilde{\mathbf{u}}_e\ A_e = \nu\sum_{j=1}^{3}\frac{\Delta\tilde{\mathbf{u}}_e - \Delta\tilde{\mathbf{u}}_{ep}}{d_{e,ep}}l_j^e - \nu\sum_{j=1}^{3}\frac{\Delta\check{\mathbf{u}}_e - \Delta\check{\mathbf{u}}_{ep}}{d_{e,ep}}l_j^e. \tag{17}$$

We solve one system (17) for each component of $\Delta\tilde{\mathbf{u}}_e$ unknown, $\Delta\tilde{u}_e$ and $\Delta\tilde{v}_e$, respectively,

$$\begin{cases} \bar{\mathbf{M}}_{x,e}\begin{pmatrix}\Delta\tilde{u}_e \\ \Delta\tilde{v}_e\end{pmatrix}A_e = \nu\sum_{j=1}^{3}\frac{\Delta\tilde{u}_e - \Delta\tilde{u}_{ep}}{d_{e,ep}}l_j^e - \nu\sum_{j=1}^{3}\frac{\Delta\check{u}_e - \Delta\check{u}_{ep}}{d_{e,ep}}l_j^e \\ \bar{\mathbf{M}}_{y,e}\begin{pmatrix}\Delta\tilde{u}_e \\ \Delta\tilde{v}_e\end{pmatrix}A_e = \nu\sum_{j=1}^{3}\frac{\Delta\tilde{v}_e - \Delta\tilde{v}_{ep}}{d_{e,ep}}l_j^e - \nu\sum_{j=1}^{3}\frac{\Delta\check{v}_e - \Delta\check{v}_{ep}}{d_{e,ep}}l_j^e, \end{cases} \tag{18a}$$

$$\text{with} \qquad \bar{\mathbf{M}}_{x,e} = \begin{pmatrix}\bar{M}_{1,1}^e \\ \bar{M}_{1,2}^e\end{pmatrix}^T \qquad \bar{\mathbf{M}}_{y,e} = \begin{pmatrix}\bar{M}_{2,1}^e \\ \bar{M}_{2,2}^e\end{pmatrix}^T, \tag{18b}$$

where $\bar{M}_{i,j}^e$, $i, j = 1, 2$, is the $(i,j)^{th}$ coefficient of matrix $\bar{\mathbf{M}}_e$. We apply the iterative procedure described in Appendix B. Typically three/four iterations are enough to satisfy the convergence of the iterative procedure and the computational effort required for solving the CP1 step is very small compared to the other algorithm steps, as shown in Section 4.1. BCs of CP1 are set as in [36]. After solving the systems (18), we update the velocity at time level $t^{k+2/3}$ according to the first relationship in Eq. (16). Due to the "local element update" operation performed during the CP1, normal flux continuity at each element interface is not yet recovered and $\mathbf{u}_e^{k+2/3}$ is not divergence free.

The sum of the viscous forces $\mathfrak{V}_e^{k-1/3}$ for the next time iteration in Eq. (10) is computed as

$$\mathfrak{V}_e^{k-1/3} = \nu\sum_{j=1}^{3}\frac{\left(\mathbf{u}_e^{k+2/3} - \mathbf{u}_{ep}^{k+2/3}\right)}{d_{e,ep}}l_j^e. \tag{19}$$

At the end of CP1, (time level $t^{k+2/3}$) we compute $\forall e \in \Omega_T$ the velocity vector $\mathbf{u}_{RT0,e}^{k+2/3} \in \Re_e$ according to Eq. (A.1), with the normal flux Q_j^e given in Eq. (20)

$$Q_j^e = \bar{F}l_j^e = \frac{\left(\mathbf{u}_e^{k+2/3} \cdot \mathbf{n}_j^e\right) A_{ep} - \left(\mathbf{u}_{ep}^{k+2/3} \cdot \mathbf{n}_{jp}^{ep}\right) A_e}{\left(A_e + A_{ep}\right)} l_j^e \qquad j, jp = 1, 2, 3, \tag{20}$$

and $\bar{F}l_j^e$ is the weighted mean flux crossing side j of e computed according to the fluxes $\mathbf{u}_e^{k+2/3} \cdot \mathbf{n}_j^e l_j^e$ and $\mathbf{u}_{ep}^{k+2/3} \cdot \mathbf{n}_{ep}^{k+2/3} l_j^e$ crossing the common side shared by e and ep at time level $t^{k+2/3}$. According to Eq. (20), the fluxes $\bar{F}l_j^e$ are continuous, i.e, $\bar{F}l_j^e = -\bar{F}l_{jp}^{ep}$, but do not satisfy mass balance, i.e., $\sum_{j=1}^3 \bar{F}l_j^e \neq 0$. This is because so far, the velocity field has been updated from the initial state $\mathbf{u}_e^k$, by solving, in the PP and CP1, momentum equilibrium equations only. This implies that $\nabla \cdot \mathbf{u}_{RT0,e}^{k+2/3} \neq 0$ and velocity $\mathbf{u}_{RT0,e}^{k+2/3}$ is piecewise linear inside e. The velocity $\mathbf{u}_{RT0,e}^{k+2/3}$, as well as the continuous fluxes $\bar{F}l_j^e$ are used for the solution of the 2nd correction problem, as explained in the following Section.

The systems (18) can be written in vector–matrix form as

$$\mathcal{A}^{x,CP1} \Delta\tilde{\mathbf{u}}_x = \mathcal{B}^{x,CP1}, \tag{21a}$$

$$\mathcal{A}^{y,CP1} \Delta\tilde{\mathbf{v}}_y = \mathcal{B}^{y,CP1}, \tag{21b}$$

where the coefficients of vectors $\Delta\tilde{\mathbf{u}}_x$ and $\Delta\tilde{\mathbf{v}}_y$ are the x and y components of $\Delta\tilde{\mathbf{u}}$ in each cell e, and the coefficients of matrices $\mathcal{A}^{x,CP1}$ and $\mathcal{A}^{y,CP1}$, as well as of the source term vectors $\mathcal{B}^{x,CP1}$ and $\mathcal{B}^{y,CP1}$, are given in Appendix B. The matrices of systems (18) are sparse and symmetric. If the mesh satisfies the Delaunay property, the matrices are also positive-definite, so that the $\mathcal{M}$-matrix property is guaranteed [40] (see also Appendix B). The systems (18) are solved by a fast and efficient Preconditioned Conjugate Gradient (PCG) method [43,44] with an incomplete Cholesky factorization [45,46]. The matrix coefficients only depend on ν, $M_{i,j}^e$ and geometrical quantities, so that the matrices of systems (18) are only factorized once, before the beginning of the time loop, saving a lot of computational effort.

3.2.3. 2nd corrector problem

Eq. (6c) is rewritten as

$$\mathbf{M}\left(\mathbf{u}^{k+1} - \mathbf{u}^{k+2/3}\right) + \epsilon\left(\nabla\Psi^{k+1} - \nabla\Psi^k\right) = 0, \tag{22}$$

with matrix $\mathbf{M}$ defined in Section 3.2.2. Introducing the scalar variable η such that

$$\mathbf{M}\left(\mathbf{u}^{k+1} - \mathbf{u}_{RT0}^{k+2/3}\right) = \epsilon\nabla\eta, \tag{23}$$

and left-multiplying both sides of Eq. (23) by $\mathbf{M}^{-1}$, we obtain

$$\left(\mathbf{u}^{k+1} - \mathbf{u}_{RT0}^{k+2/3}\right) = \epsilon\mathbf{M}^{-1}\nabla\eta, \tag{24}$$

with $\mathbf{u}_{RT0}^{k+2/3}$ defined as in Section 3.2.2. Since $\mathbf{u}_{RT0}^{k+2/3}$ is piecewise linear within each triangle e (see end of Section 3.2.2), we see from Eq. (23) that η is piecewise quadratic within triangle e, while, from Eq. (22), Ψ is piecewise linear within e. Taking the divergence of Eq. (24) we get

$$\nabla \cdot \left(\mathbf{u}^{k+1} - \mathbf{u}_{RT0}^{k+2/3}\right) = \nabla \cdot (\boldsymbol{\Xi}\nabla\eta) \quad \text{with} \quad \boldsymbol{\Xi} = \epsilon\mathbf{M}^{-1}. \tag{25}$$

Setting $\nabla \cdot \mathbf{u}^{k+1} = 0$, integrating in space over each element e and applying the divergence theorem, we obtain from Eq. (25)

$$\sum_{j=1}^{3}\left(\bar{F}l_j^e\right) + \oint_{L_e} (\bar{\boldsymbol{\Xi}}\nabla\eta) \cdot \mathbf{n}\, \mathrm{d}l = 0 \quad \text{with} \quad \oint_{L_e} (\bar{\boldsymbol{\Xi}}\nabla\eta) \cdot \mathbf{n}\, \mathrm{d}l = \sum_{j=1}^{3} \left(\bar{\boldsymbol{\Xi}}_j\nabla\eta\right) \cdot \mathbf{n}_j\, l_j^e, \tag{26}$$

where the average normal flux $\bar{F}l_j^e$ crossing side j of cell e has been defined in Eq. (20). Since matrix $\bar{\boldsymbol{\Xi}}_j$ is symmetric and positive definite, we have $\left(\bar{\boldsymbol{\Xi}}_j\nabla\eta\right) \cdot \mathbf{n}_j = \left(\bar{\boldsymbol{\Xi}}_j\mathbf{n}_j\right) \cdot \nabla\eta$, and we set $\bar{\boldsymbol{\Xi}}_j\mathbf{n}_j = \mathbf{d}_j$. Decomposing vector $\mathbf{d}_j$ along the normal and tangential directions to side j ($\mathbf{n}_j$ and $\boldsymbol{\tau}_j$ in Fig. 3), $\mathbf{d}_j = \mathbf{d}_{j,n} + \mathbf{d}_{j,\tau}$, and applying a co-normal decomposition of vector $\mathbf{d}_{j,\tau}$ along the normal directions to the other two sides of cell e ($\mathbf{n}_1$ and $\mathbf{n}_2$ in Fig. 3), after some manipulations whose details are given in Appendix C, Eq. (26) is discretized as

$$\begin{gathered}\sum_{j=1}^{3}\left(\bar{F}l_j^e\right) = \sum_{j=1}^{3}\left(\frac{\eta_e - \eta_{ep}}{d_{e,ep}} d_{j,n}\right) l_j^e + \\ \frac{1}{2}\sum_{j=1}^{3}\left(\sum_{l=1,2}\frac{\eta_e - \eta_{ep_l}}{d_{e,ep_l}} d_{\tau_{n_l}}\alpha_l + \sum_{m=3,4}\frac{\eta_{ep_m} - \eta_{ep}}{d_{ep,ep_m}} d_{\tau_{n_m}}\alpha_m\right) l_j^e\end{gathered}, \tag{27}$$

which forms a system to be solved for the η unknowns. With the help of Appendix C and Fig. 3, $d_{j,n} = \mathbf{d}_j \cdot \mathbf{n}_j$, $d_{\tau_{n_{l(m)}}} = \mathbf{d}_{j,\tau} \cdot \mathbf{n}_{l(m)}$ (l = 1,2, m = 3,4), η_e, η_{ep}, $\eta_{ep_{l(m)}}$ are the η unknowns in the circumcenters of cells e, ep, and $ep_{l(m)}$, respectively, and $d_{e,ep}$, d_{e,ep_l} and d_{e,ep_m} are the distances of the circumcenters of cells e and ep, e and ep_l, ep and ep_m respectively, times +1 or −1, depending on whether the mesh satisfies or not the Delaunay property (see also Section 3.2.2). $\alpha_{l(m)} = 1$ if $\mathbf{d}_{j,\tau} \cdot \mathbf{n}_{l(m)} > 0$ otherwise $\alpha_{l(m)} = -1$.

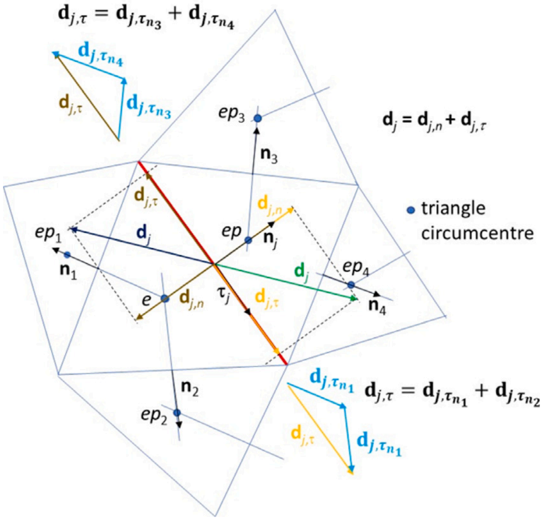

$\mathbf{d}_{j,\tau} = \mathbf{d}_{j,\tau_{n_3}} + \mathbf{d}_{j,\tau_{n_4}}$
$\mathbf{d}_{j,\tau_{n_4}}$
$\mathbf{d}_{j,\tau}$
$\mathbf{d}_{j,\tau_{n_3}}$
ep_3
$\mathbf{d}_j = \mathbf{d}_{j,n} + \mathbf{d}_{j,\tau}$
$\mathbf{n}_3$
$\mathbf{d}_{j,\tau}$
$\mathbf{d}_{j,n}$
● triangle circumcentre
ep_1
$\mathbf{d}_j$
ep
$\mathbf{n}_j$
$\mathbf{n}_1$
$\mathbf{d}_j$
ep_4
e
$\mathbf{d}_{j,n}$
τ_j
$\mathbf{d}_{j,\tau}$
$\mathbf{n}_4$
$\mathbf{d}_{j,\tau_{n_1}}$
$\mathbf{d}_{j,\tau} = \mathbf{d}_{j,\tau_{n_1}} + \mathbf{d}_{j,\tau_{n_2}}$
$\mathbf{n}_2$
ep_2
$\mathbf{d}_{j,\tau}$
$\mathbf{d}_{j,\tau_{n_1}}$

Fig. 3. CP2, calculation of $\left(\boldsymbol{\Xi}_j \mathbf{n}_j\right) \cdot \nabla \eta$.

Instead of solving system (27), we solve an approximation of it, according to the following strategy. We first compute an approximate solution of η, denoted as $\tilde{\eta}$, by solving system (28),

$$\sum_{j=1}^{3} \left(\frac{\tilde{\eta}_e - \tilde{\eta}_{ep}}{d_{e,ep}} d_{j,n} \right) l_j^e = \sum_{j=1}^{3} \left(\bar{Fl}_j^e \right). \tag{28}$$

With this, the η solution can be obtained by solving system (29a),

$$\sum_{j=1}^{3} \left(Fl_j^{\eta,e} \right) = \sum_{j=1}^{3} \left(\bar{Fl}_j^e \right), \tag{29a}$$

$$\text{with} \quad \begin{aligned} Fl_j^{\eta,e} = \int_{l_{j,e}} (\Xi \nabla \eta) \cdot \mathbf{n}_j dl_j = \frac{\eta_e - \eta_{ep}}{d_{e,ep}} d_n l_j^e + \\ \frac{1}{2} \left(\sum_{l=1,2} \frac{\tilde{\eta}_e - \tilde{\eta}_{ep_l}}{d_{e,ep_l}} d_{\tau_{n_l}} \alpha_l + \sum_{m=3,4} \frac{\tilde{\eta}_{ep} - \tilde{\eta}_{ep_m}}{d_{ep,ep_m}} d_{\tau_{n_m}} \alpha_m \right) l_j^e \end{aligned}, \tag{29b}$$

where $Fl_j^{\eta,e}$ is the flux crossing side j of e due to $\Xi \nabla \eta$. Systems (28)–(29) are an approximation of system (27). The advantage of solving the approximation systems instead of system (27), will be explained at the end of this subsection.

The same spatial discretization as in Eq. (15) has been applied. Eq. (29b) implies that fluxes $Fl_j^{\eta,e}$, as well as $\bar{FL}_j^e$, are continuous for the two neighbor cells e and ep, and Eq. (26) implies that $\sum_{j=1}^{3} \left(Fl_j^{\eta,e} + \bar{FL}_j^e \right) = 0$, $\forall e \in \Omega_T$. Mass conservation along the three sides and inside each cell e, $\forall e \in \Omega_T$ is finally recovered at time level t^{k+1}.

Observe in Eq. (29b) that flux $Fl_j^{\eta,e}$ depends on the six values of η in cells e, ep and $ep_{l(m)}$ ($l = 1{,}2$, $m = 3{,}4$). Since the η values in cells $ep_{l(m)}$ are assumed to be known in the solution of system (28), $Fl_j^{\eta,e}$ depends only on the two unknowns η_e and η_{ep}. For this reason, due to the splitting strategy operated in Eqs. (28) and (29a), the flux discretization scheme in Eq. (29b) can be regarded, *de facto*, as a Two-Point-Flux-Approximation scheme (TPFA).

After the solution of systems (28)–(29), we obtain the velocity vector at the end of the time step $\forall e \in \Omega_T$, $\mathbf{u}_e^{k+1}$, as in Eq. (24), where $\Xi \nabla \eta$, as well as $\mathbf{u}_{RT0}^{k+2/3}$, is a **RT0** function,

$$\mathbf{u}_{RT0}^{k+2/3} = \sum_{j=1}^{3} \left(\mathbf{w}_j^e \bar{FL}_j^e \right), \qquad \Xi \nabla \eta = \sum_{j=1}^{3} \left(\mathbf{w}_j^e Fl_j^{\eta} \right), \tag{30}$$

and we get

$$\mathbf{u}_e^{k+1} = \sum_{j=1}^{3} \mathbf{w}_j^e \left(\bar{Fl}_j^e + Fl_j^{\eta} \right). \tag{31}$$

According to the properties of the **RT0** functions (see Eqs. (A.1) and (A.2a)), $\nabla \cdot \mathbf{u}_e^{k+1} = 0$ $\forall \mathbf{x} \in e$, $\forall e \in \Omega_T$, and the method is *strongly* conservative (see Section 1). Once $\mathbf{u}_e^{k+1}$ is known, we get from Eq. (22), term $\bar{\epsilon}_e \nabla \Psi^{k+1}$, $\forall e \in \Omega_T$, as,

$$\bar{\epsilon}_e \nabla \Psi_e^{k+1} = \bar{\epsilon}_e \nabla \Psi_e^k + \bar{\mathbf{M}}_e \left(\mathbf{u}_e^{k+1} - \mathbf{u}_e^{k+2/3} \right), \tag{32}$$

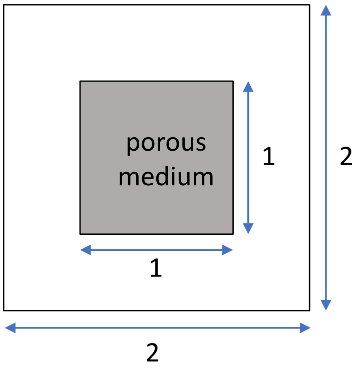


Fig. 4. Test 1. Computational domain.

where the symbols have been specified before. Since the kinematic pressure is not needed to update the solution at each time step, we compute the nodal values of Ψ at target simulation times only, as explained in Section 5 in [36] and in Eqs. (36) in [37]. In the framework of a linear Galerkin Finite element scheme, we set the kinematic pressure distribution $\Psi = \sum_{e=1}^{N_T} \omega_i \tilde{\Psi}_i$, with ω_i and $\tilde{\Psi}_i$ the shape function and the nodal unknown kinematic pressure values, respectively. We minimize the difference between the gradients computed by Eq. (32) and the gradient of the unknown $\tilde{\Psi}$ distribution. More details in the referred papers.

We set in Eq. (10) of the PP of the next time iteration

$$\mathfrak{G}_{\eta,e}^{k} = \bar{\epsilon}_e \nabla \Psi^{k+1} A_e. \tag{33}$$

The boundary conditions of the CP2 are assigned as specified in [36]. For sides $j \in \Gamma_n$ we wet

$$\eta = \Psi_b^{k+1} - \Psi_b^{k} \tag{34}$$

with Ψ_b^k and Ψ_b^{k+1} the values of the kinematic pressure assigned at the boundary (the values of the normal stress component σ_n in Eq. (3b) at time levels k and k+1). Eq. (34) is consistent with the BCs in Eq. (3b). The corrective fluxes at the same boundary sides are computed a-posteriori via Eqs. (28) and (29b). See details in [36].

The systems (28)–(29a) can be written in vector–matrix form as

$$\boldsymbol{\mathcal{A}}^{CP2}\ \tilde{\boldsymbol{\eta}} = \boldsymbol{\mathcal{B}}_1^{CP2}, \tag{35a}$$

$$\boldsymbol{\mathcal{A}}^{CP2}\ \boldsymbol{\eta} = \boldsymbol{\mathcal{B}}_2^{CP2}, \tag{35b}$$

where the coefficients of vectors $\tilde{\boldsymbol{\eta}}$ and $\boldsymbol{\eta}$ are the value of $\tilde{\eta}$ and η in each cell e and the coefficients of matrix $\boldsymbol{\mathcal{A}}^{CP2}$ and of the source term vectors $\boldsymbol{\mathcal{B}}_{1(2)}^{CP2}$ are given in Appendix C. Matrix $\boldsymbol{\mathcal{A}}^{CP2}$ is sparse and symmetric (see Appendix C), yielding the same beneficial matrix properties as for the CP1 problem if the mesh satisfies the Delaunay property. Therefore, systems (28)–(29a) are solved by applying the same procedure as in Section 3.2.2. Moreover, this again allows to perform factorization of matrix $\boldsymbol{\mathcal{A}}^{CP2}$ only once.

The matrix associated with the original system (27) is not symmetric, and the advantage of splitting system (27) into systems (28)–(29a) is twofold: (1) numerical stability is achieved, since matrix $\boldsymbol{\mathcal{A}}^{CP2}$ is a $\mathcal{M}$-matrix and (2) there is a fast and efficient solution of the PCG method, compared to the standard GMRES, BiCG, CGSquared, or BiCGStab methods, usually applied for the solution of non-symmetric matrix systems.

4. Numerical tests and analysis of the results

We present five numerical tests. In the first test we analyze the convergence order of the proposed algorithm and the required computational (CPU) times. In the second test, we compare the solution of the presented algorithm with an analytical one provided in the literature for Stokes flow regime and different geometrical domain configurations. The third, fourth and fifth test are related to "real-world" applications. In the third and fourth test we compare our numerical solution with averaged pore-scale results for the Stokes and Navier–Stokes flow regimes. In the last test, we provide a *showcase* with different anisotropy tensors, for different Reynolds numbers and a comparison with the solution provided by a TDA code developed in the framework of the open-source software package DuMu$^{\text{x}}$ [47] which applies a Multi-Point-Flux-Approximation (MPFA) scheme for calculating the solution in the porous region.

As ICs for all the presented test cases, we adopt zero velocity and zero kinematic pressure in the domain.

A very fast off-line *in-house* procedure is adopted to generate the computational mesh and the input data, and assign the ICs and BCs [37]. The output model results are processed with Paraview [48]. In the presented tests, we neglect gravitational effects.

In the following Sections, δ_0 and δ_{TL} mark the mesh sizes adopted to discretize the bulk free fluid and porous regions, as well as the transition layer, respectively, while d_{TL} is the width of the transition layer.

Table 1
Test 1. Parameter values in Eq. (37).

a_1	a_2	b_1	b_2	c_1	c_2
5×10^{-4}	1.5×10^{-3}	-3.2×10^{-3}	-8×10^{-4}	-1.95×10^{-3}	4×10^{-3}

4.1. Test 1. Convergence test

We assume a 2D domain $\Omega = [0,2]^2$ m^2 with an internal square porous region Ω_{pm} $[0.5, 1.5]^2$ m^2 (see Fig. 4). In the bulk porous region, the porosity is $\epsilon = \epsilon_0 = 0.4$, and the anisotropy tensor $\mathbf{K}$ is given by Eq. (36)

$$\mathbf{K} = \mathbf{RAR}^{-1}, \quad \mathbf{A} = \begin{pmatrix} \frac{k}{\beta} & 0 \\ 0 & k \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \tag{36}$$

with $\beta = 100$, $\alpha = -\pi/4$, such that $k = 1.0 \times 10^{-6}$ m^2. Matrix $\mathbf{K}$ is symmetric and positive definite. Kinematic fluid viscosity is $\nu = 1.5 \times 10^{-5}$ m^2/s. The pressure field is given by

$$\Psi = a_1 + b_1 x + c_1 x^2 + a_2 + b_2 y + c_2 y^2, \tag{37}$$

where the values of the coefficients a_i, b_i and c_i $(i = 1,2)$ are listed in Table 1. The analytical velocity solution is constructed to be divergence free and continuous at the interface between free fluid and porous regions, with velocity components

$$\begin{aligned} u &= -(K_{1,1}\frac{\partial\Psi}{\partial x} + K_{1,2}\frac{\partial\Psi}{\partial y}), \\ v &= -(K_{2,1}\frac{\partial\Psi}{\partial x} + K_{2,2}\frac{\partial\Psi}{\partial y}). \end{aligned} \tag{38}$$

The Reynolds number is computed as $Re = \frac{\|\mathrm{u}\|_{max} H_{pm}}{\nu}$, where $\|\mathrm{u}\|_{max}$ is the maximum value of the velocity vector magnitude and H_{pm} is the side of the porous domain, such that $Re \simeq 267$. We assume $\Gamma_n = \Gamma$, except in the upper right corner, where we assign the Dirichlet condition for Ψ equal to the value given by Eq. (37).

In our scheme, the outer boundary of the transition zone overlaps the outer contour of Ω_{pm}. A continuous variation of the porosity ϵ and the coefficients $\mathfrak{K}_{i,j}$ of the inverse permeability tensor is assumed within the transition layer as in Eq. (39)

$$\epsilon(\mathbf{x}) = \frac{1}{2}\left(1 - \epsilon_0\right)\tanh((d + d_\epsilon)\theta_\epsilon) + \frac{1}{2}\left(1 + \epsilon_0\right), \tag{39a}$$

$$\mathfrak{K}_{i,j}(\mathbf{x}) = \mathfrak{K}^0_{i,j}\frac{1}{2}\left(1 - \tanh((d + d_k)\theta_K)\right), \tag{39b}$$

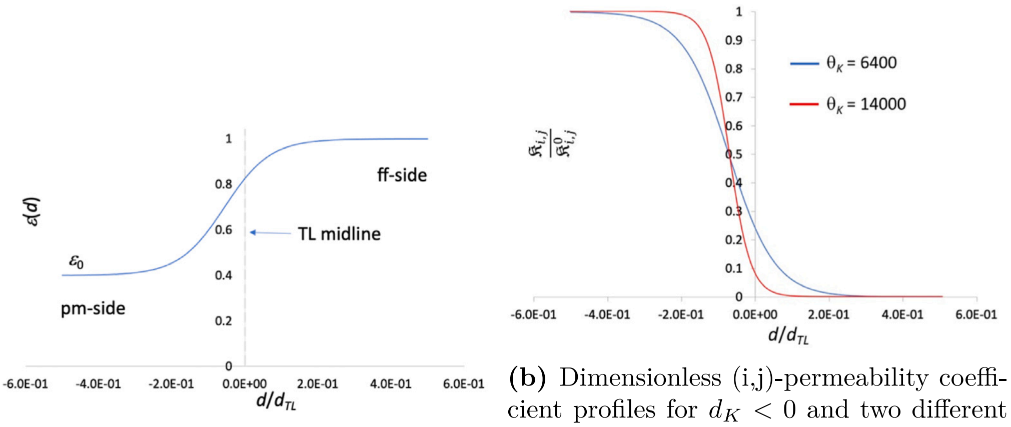
where $\mathfrak{K}^0_{i,j}$ are the coefficients of the inverse of the permeability tensor within the bulk Ω_{pm}, computed according to Eq. (36) and $d = |\mathbf{x}_P - \mathbf{x}_{TL,mid}|\, sign$, where $|\mathbf{x}_P - \mathbf{x}_{TL,mid}|$ is the distance of any point P within TL from the center-line of TL and $sign = 1$ or -1 depending on whether point P is located in the half-region of TL on the side of the porous medium or the free fluid region. d_ϵ and d_K are scalar values which control the symmetry of the profiles of ϵ and $\frac{1}{K_{i,j}}$ with respect to the center-line of TL. Depending on whether they assume positive or negative values, the profiles are shifted towards the free fluid region or to the porous medium region. θ_ϵ and θ_K are positive scalar values which control the slope of the profiles (the larger they are, the steeper the profiles). Fig. 5(a) shows a porosity profile computed setting a negative value of d_ϵ, while in Fig. 5(b) we plot two dimensionless permeability profiles corresponding to a negative d_K value.

Generally, the solution of the model is affected by both position and size of the transition zone. This is because the extension of the portion of the computational domain where the term $\mu\epsilon\mathfrak{K}\mathbf{u}$ in the momentum equation (see Eqs. (1) and (2)) is active, depends on the position and size of the transition layer.

We initially assumed the width of TL to be $d_{TL} = H_{pm}/25$, and the domain is discretize with a coarse mesh ($N_T = 16260$ triangles and $N = 8292$), whose maximum values of mesh sizes are $\delta_0 = 0.025$ m and $\delta_{TL} = 0.02$ m, respectively. Starting from this coarse mesh, we progressively performed four refinement operations by halving $\delta_0 =$, leaving d_{TL} unchanged and setting $\delta_0 = \delta_{TL}$. The adopted time step size is $\Delta t = 2.25 \times 10^{-1}$ s. The maximum CFL value, $CFL_{max} = \|\mathrm{u}\|_{max}\Delta t/\sqrt{A_e}$ was computed in TL, and $CFL_{max} \simeq 1.387$. At each mesh refinement, we halved Δt, to avoid increases of CFL_{max}. We set both d_ϵ and d_K to zero, and $\theta_\epsilon = \theta_K = 200$ in Eq. (39). The L_2 norm of the errors of the computed solutions for u, v and Ψ with respect to the exact solutions given in Eqs. (37) and (38) is computed as

$$\begin{aligned} L_2(err(q)) &= \sqrt{\sum_{e=1}^{N_T} A_e \left(q_e^n - q_e^{ex}\right)^2}, \\ L_2(err(\Psi)) &= \sqrt{\sum_{i=1}^{N} A_i \left(\Psi_i^n - \Psi_i^{ex}\right)^2}, \end{aligned} \tag{40}$$

where $q = u$ or v, A_i is the area of the Voronoi polygon associated with node i and superscripts n and ex mark the numerical and exact solutions, respectively. q_e^n and q_e^{ex} are computed in the circumcenter of each cell, while Ψ_i^n and Ψ_i^{ex} are computed in the mesh



(a) Porosity profile for $d_\epsilon < 0$

(b) Dimensionless (i,j)-permeability coefficient profiles for $d_K < 0$ and two different values of θ_K

Fig. 5. Porosity and permeability profiles.

nodes. If we call h_l the mesh size associated with the lth refinement, and assume that the error associated with the lth refinement is proportional to a power r_c of h_l, we compute the spatial rate of convergence r_c by comparing the errors obtained for the two meshes with the two consecutive linear sizes h_l and h_{l+1},

$$r_c = \ln\left[\frac{L_2\left(err_l\right)}{L_2\left(err_{l+1}\right)}\right] / \ln\left[\frac{h_l}{h_{l+1}}\right]. \tag{41}$$

In Table 2 we list the L_2 norms of the errors and the convergence order r_c of the velocity components is close to 1, due to the piecewise constant approximation of the velocity inside each triangle. r_c of Ψ is smaller than 2. The reason could be that, due to the lack of a specific equation for Ψ in the governing equations, this is indirectly computed from the pressure gradients, as described at the end of Section 3.2.3.

We also investigate how the size of TL, d_{TL}, affects the computed results compared to the exact ones. Starting from the 2nd mesh refinement level, we progressively halved d_{TL} as well as its mesh size δ_{TL}, without changing δ_0 in the free fluid and the bulk porous regions. At each refinement of d_{TL}, we also halved Δt for the aforementioned reasons. Since the assigned analytical solution of the velocity vector depends on the values of the permeability coefficients in the bulk porous region, without any transition of their values close to the interface with the fluid region (see Eq. (39)), we expect the numerical solution to get closer and closer to the exact one by refining d_{TL}. This is confirmed by the results in Table 3.

We also investigated the computational (CPU) times required by the algorithm steps, PP, CP1, CP2 and sorting cell operation, $\overline{CPU}_{PP}$, $\overline{CPU}_{CP1}$, $\overline{CPU}_{CP2}$, $\overline{CPU}_{\Psi}$ and $\overline{CPU}_{srt}$, respectively. Given two real scalar numbers c and ω, we express the mean value of $\overline{CPU}_{step}$ per time iteration as

$$\overline{CPU}_{step} = \exp(c)\, N_T^{\omega} \qquad \text{or} \qquad \overline{CPU}_{step} = \exp(c)\, N^{\omega}. \tag{42}$$

where "*step*" in Eq. (42) corresponds to *PP*, or *CP1*, or *CP2*, or Ψ, or *srt*. A single Intel(R) Core(TM) i7-9700K processor at 3.40 GHz was used for the simulation runs. In Fig. 6 we show the computational times in bi-logarithmic scales. Due to the *explicit nature* of the predictor step (i.e., the sequential solution of the ODEs systems during the MAST forward and backward steps), its growth with N_T is almost linear (the γ exponent is slightly smaller than 1). Since the two corrector steps, as well as the computation of the kinematic pressure require the solution of linear systems, their growth with N_T (or N) is more than linearly proportional (the associated exponents γ are slightly higher than 1, ranging from 1.1307 to 1.1533). $\overline{CPU}_{CP1}$ and $\overline{CPU}_{srt}$ are approximately 1 and 2–3 magnitude orders smaller than $\overline{CPU}_{PP}$ and $\overline{CPU}_{CP2}$. The difference between $\overline{CPU}_{CP1}$ and $\overline{CPU}_{CP2}$ is because the storage term is missing in Eqs. (28) and (29) of the CP2. This implies that the condition of strict diagonal dominance is obtained for each row of the matrices of the systems of CP1. On the contrary, this condition only applies to the rows associated with the Dirichlet BCs for η for the CP2 system matrix.

4.2. Test 2. Comparison with analytical solution for Stokes flow regime in a three-layer channel

We deal with a 1D Stokes flow regime along the x direction in a three-layer channel with total depth H, where the transition layer, TL, is in between a bulk porous and a clear fluid region (see Fig. 7(a)). This problem is proposed in [49]. The spatial distribution of the isotropic permeability K is given in Eq. (43a)

$$\frac{1}{K(y)} = 0 \qquad \text{in clear fluid region, layer 1,} \qquad 0 \le y \le \zeta H, \tag{43a}$$

Table 2
Test1. L_2 norms of errors and r_c for mesh refinement.

Mesh details		u		v		Ψ	
N	N_T	L_2	r_c	L_2	r_c	L_2	r_c
8292	16 260	1.20×10^{-3}	–	1.21×10^{-3}	–	3.45×10^{-2}	–
29069	57 512	6.06×10^{-4}	9.88×10^{-1}	6.06×10^{-4}	9.98×10^{-1}	1.3×10^{-2}	1.41
115110	228 970	3×10^{-4}	1.01	3.01×10^{-4}	1.01	4.6×10^{-3}	1.5
423918	845 398	1.45×10^{-4}	1.05	1.44×10^{-4}	1.06	1.5×10^{-3}	1.62
1647132	3 289 420	6.88×10^{-5}	1.07	6.86×10^{-5}	1.07	4.56×10^{-4}	1.72

Table 3
Test1. L_2 norms of errors for d_{TL} refinement.

N	N_T	$\theta_\epsilon = \theta_K$	$L_{2,u}$	$L_{2,v}$	L_Ψ
115110	228 970	800	3×10^{-4}	3.01×10^{-4}	4.6×10^{-3}
160753	329 773	1600	1.98×10^{-4}	1.97×10^{-4}	3.01×10^{-3}
218507	435 746	3200	1.178×10^{-4}	1.177×10^{-4}	1.7×10^{-3}
323158	645 032	6400	6.01×10^{-5}	5.97×10^{-5}	7.6×10^{-4}

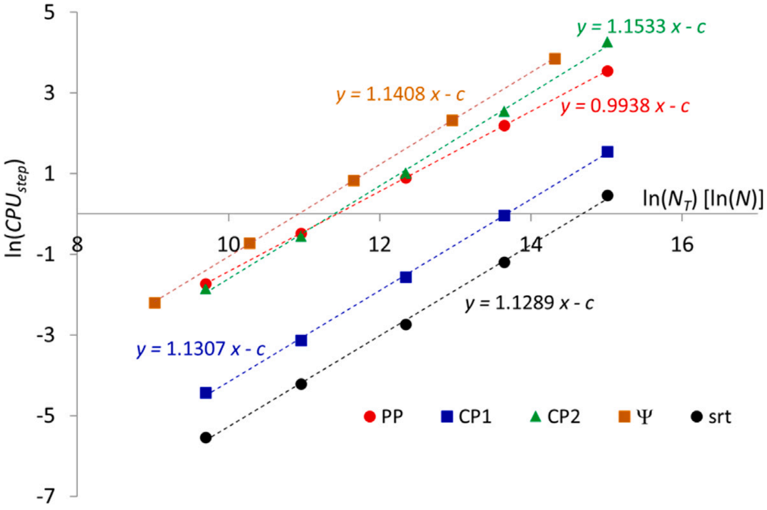


Fig. 6. Study of the CPU times of the algorithm steps.

$$\frac{1}{K(y)} = \frac{1}{K_0} \frac{y - \zeta H}{\xi H - \zeta H} \qquad \text{in transition zone, layer 2} \qquad \zeta H \leq y \leq \xi H, \tag{43b}$$

$$K(y) = K_0 \qquad \text{in bulk porous region, layer 3,} \qquad y > \xi H. \tag{43c}$$

The flow is driven by a uniform negative pressure gradient G in the three layers, yielding the following Stokes–Brinkman Equations

$$\mu \frac{\partial^2 u}{\partial y^2} + G = 0 \qquad 0 \leq y \leq \zeta H, \tag{44a}$$

$$\mu \frac{\partial^2 u}{\partial y^2} - \frac{\mu u}{K(y)} + G = 0 \qquad \zeta H \leq y \leq \xi H, \tag{44b}$$

$$\mu \frac{\partial^2 u}{\partial y^2} - \frac{\mu u}{K(y)} + G = 0 \qquad y > \xi H. \tag{44c}$$

The analytical solution of the dimensionless form of system (44a) is given by Eqq. (10) in [49]. This has been obtained by solving the system in Eqq.(15)–(18) of the same paper. In the present work, we solved the system in Eqq. (15)–(18) in [49] using the software package *Mathematica* [50]. We consider both *thin* and *fat* TL scenarios ("ttl" and "ftl", respectively), and we present simulations for small and large Darcy numbers Da values ($Da = K_0/H^2$). We set $H = 0.5$ m and an equal value of the channel length. The "ftl" and "ttl" configurations are obtained by setting $\zeta = 1/3H =$ and $\xi = 2/3H$, and $\zeta = 1/3H$ and $\xi = 1.006/3H$, respectively.

Table 4
Test 2. Parameters for the computational meshes m_i.

	δ_0 [m]	δ_{TL} [m]	N	N_T
ftl m_0	5×10^{-3}	5×10^{-3}	19 478	37 752
ftl m_1	2.5×10^{-3}	2.5×10^{-3}	51 841	102 287
ftl m_2	1.25×10^{-3}	1.25×10^{-3}	195 556	389 356
ttl m_0	5×10^{-3}	7×10^{-5}	242 750	484 234
ttl m_1	2.5×10^{-3}	7×10^{-5}	273 272	545 112
ttl m_2	1.25×10^{-3}	7×10^{-5}	402 484	803 165

Table 5
Test 2. L_2 norms of relative error of u.

Mesh	*ftl*		*ttl*	
	$Da = 2 \times 10^{-4}$	$Da = 1 \times 10^{-2}$	$Da = 2 \times 10^{-4}$	$Da = 1 \times 10^{-2}$
m_0	3.4×10^{-4}	3.51×10^{-4}	4.7×10^{-4}	4.85×10^{-4}
m_1	1.71×10^{-4}	1.74×10^{-4}	2.23×10^{-4}	2.4×10^{-4}
m_2	8.5×10^{-4}	8.7×10^{-4}	1.1×10^{-4}	1.2×10^{-4}

We set a pressure drop $\Delta\Psi = 5 \times 10^{-7}$ m^2/s^2 between the left and right end sides of the channel, while a no-slip velocity condition is assigned along the horizontal bottom and top walls. The kinematic fluid viscosity is $\nu = 1.5 \times 10^{-5}$ m^2/s. In Table 4 we list the mesh sizes δ_0 and δ_{TL} adopted for the different simulated scenarios, as well as the number of nodes N and triangles N_T of the corresponding meshes. The time step size $\Delta t = 10$ s for all the simulated scenarios. We set K_0 such that $Da = 2 \times 10^{-4}$ and $Da = 1 \times 10^{-2}$, and CFL$_{max}$ ranges from 1.42 ("ttl" case and small Da) to 2.45 ("ftl" case and large Da).

In Table 5 we list the L_2 norms of the relative errors of the horizontal velocity component for the computational meshes m_i, computed as in Eq. (40). In Fig. 7(b) we compare the numerical ODA solution against the analytical solution. The numerical solution of the presented solver fits very well the analytical one. For brevity, we show only the solutions for "ftl" with small Da and "ttl" with large Da cases, but the matching is satisfactory also for the other investigated scenarios. The numerical solution in Fig. 7(b) is related to mesh m_1 (see Table 4), but the solutions obtained for the meshes m_0 and m_2 are indistinguishable at the graphic scale, and for brevity are not shown.

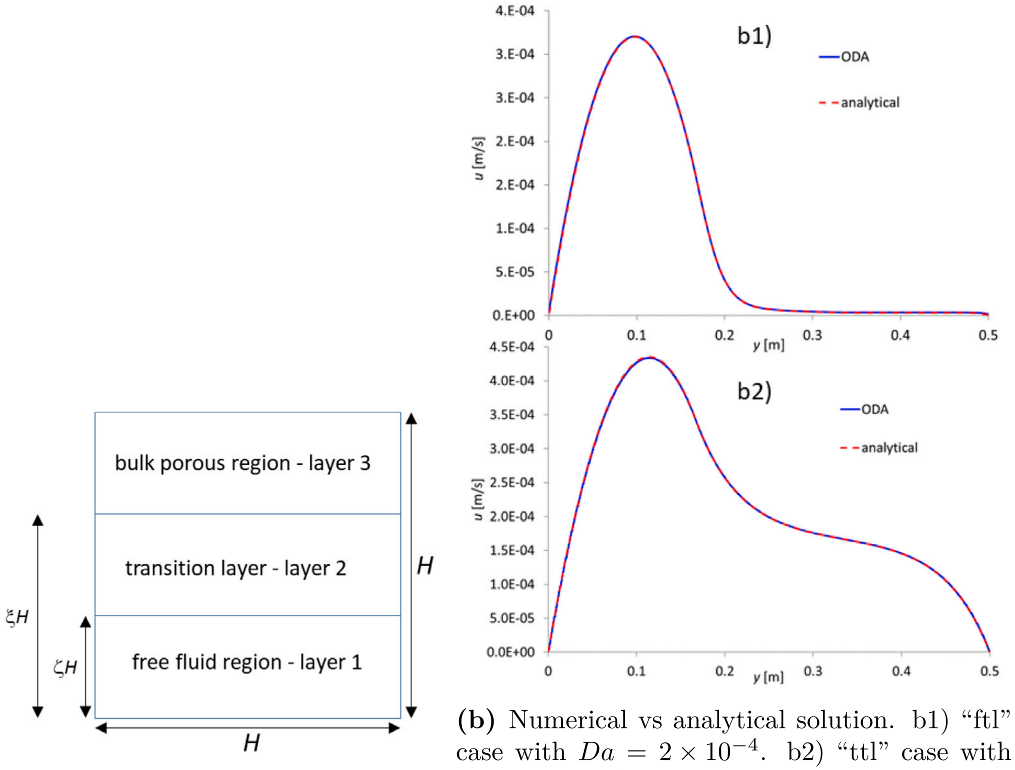
4.3. Test 3. Comparison with pore-scale results for Stokes flow regime

Test 3 is related to filtration/exfiltration processes in microfluidics, which presents several industrial, environmental or biomedical applications (e.g., membrane filtration processes, wastewater treatment, water purification, hemodialysis, …). The test case considered in this Section has been proposed in [51]. The computational domain $\Omega = (0, L) \times (0, H)$ has a free fluid region $\Omega_{ff} = (0, L)\times(\gamma, H)$ and a porous region $\Omega_{pm} = (0, L)\times(0, \gamma)$ with $L = 10.25$ mm and $H = 6$ mm. The porous region is isotropic, made up of 20 × 10 square solid inclusions of size $d_{si} = 250$ μm, such that the porosity ϵ is 0.75. $\gamma = 5$ mm is the distance between the bottom boundary and the tangential line located on top of the uppermost row of solid inclusions. This is shown in Fig. 8(a), where the geometrical setup and the assigned boundary conditions are depicted. An inflow velocity profile $\mathbf{v} = (0, V_{max}\ sin(\frac{1000}{3}\ \pi x))$ is assigned on $\Gamma_e = 1.5($ mm$, 4.5$ mm$)\times H$, with $V_{max} = 1 \times 10^{-3}$ m/s, and $\Psi = 0$ on $\Gamma_n = L\times(5.5$ mm$, H)$. The remaining portions of the boundaries are assumed to be impervious walls with no-slip velocity condition. The considered fluid is water ($\nu = 1 \times 10^{-6}$ m^2/s) and the bulk permeability is $K_0 = 3.45 \times 10^{-9}$ m^2, computed according to the geometry of the solid inclusions [52]. The Reynolds number associated with the free fluid zone, $Re = \left(V_{max}\,(H - \gamma)\right)/\nu$, is equal to 1.

To obtain a reference solution, we averaged the results of a pore-scale simulation (PSS) obtained by solving the present test case using the open-source simulator DuMu^x [47]. At the pore scale, the flow is governed by the Stokes equations in the entire domain, with no-slip BCs assigned on the boundaries of the solid inclusions. A uniform structured mesh was used for the PSS, with side $\delta_{PSS} = 1.5625 \times 10^{-5}$ m. The ODA solution is compared with the "surface average" [20,21] results of the PSS over square-shaped REVs, the size of which, l_{REV}, depend on whether the REV is located in the porous or the fluid region (see Fig. 8(b)). Within Ω_{pm}, $l_{REV} = 2d_{si}$ and REVs centroids coincide with those of the solid inclusions (yellow squares in Fig. 8(b)). Within Ω_{ff}, $l_{REV} = 8\delta_{PSS}$ (red squares in Fig. 8(b)).

Both *bottom* and *middle* TL positions were assumed. *Bottom* and *middle* positions mean that the center-line of the TL is at $y = \gamma - d_{si}/2$ or $y = \gamma$, respectively. The spatial variations of ϵ and $1/K$ within TL are given by Eq. (39).

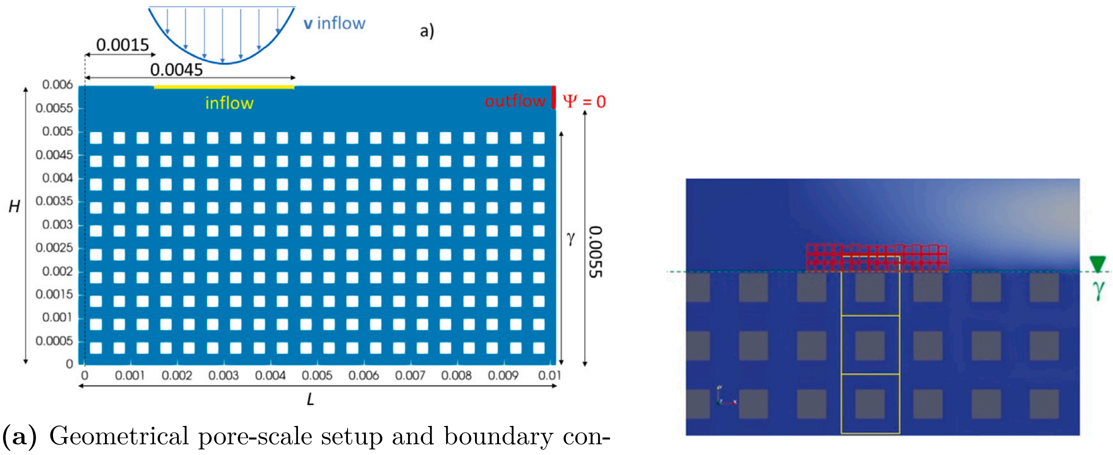
The TL width, d_{TL}, as well as the parameters needed for modeling porosity and permeability variation given by Eq. (39) were selected according to preliminary simulations performed on a fine mesh (uniform linear size of 0.0085 mm, $N = 895064$ and $N_T = 1786418$), and were adopted for TL positions. Therefore we assumed: $d_{TL} \in \left[d_{si}, 3d_{si}\right]$, $d_\epsilon = 0$, $d_K \in \left[0, 1 \times 10^{-4}\text{m}\right]$ and $\theta_\epsilon = \theta_K \in \left[7 \times 10^3, 5 \times 10^4\right]$. For each simulation we compared, along the horizontal line at $y = 0.004875$ m (the vertical coordinate of the centers of mass of the solid inclusions of the uppermost row), the distributions of u, v and Ψ computed by the ODA solver with the corresponding reference data of the averaged PSS.



(a) Setup

(b) Numerical vs analytical solution. b1) "ftl" case with $Da = 2 \times 10^{-4}$. b2) "ttl" case with $Da = 1 \times 10^{-2}$

Fig. 7. Test 2. Considered domain with three layers (a). Numerical and analytical results for two considered cases (b).



(a) Geometrical pore-scale setup and boundary condition (measures in m)

(b) REVs within Ω_{pm} and Ω_{ff}

Fig. 8. Test 3. Pore-scale setup (a). Considered REVs for averaging (b).

The ODA solutions obtained for the *middle* TL position seemed not to be sensitive to the size δ_{TL}, and, compared to the *bottom* position configuration, we observed (1) an overestimation of the velocity vector magnitude $\|\mathbf{u}\|$, up to twice in the bulk region Ω_{pm} and up to 20% in the Ω_{ff} region, and (2) an underestimation of $\|\mathbf{u}\|$ close to the interface between Ω_{pm} and Ω_{ff}, up to one magnitude order. This is why the results are not further discussed here. The best match with the averaged PSS results was obtained for the *middle* TL configuration with $d_K = 3 \times 10^{-5}$ and $\theta_\epsilon = \theta_K = 4.7 \times 10^4$.

The adopted mesh sizes are $\delta_0 = 2.5 \times 10^{-5}$ m and $\delta_{TL} = 1 \times 10^{-5}$ m. These sizes guarantee a good compromise between the accuracy of the results (maximum relative error values of the velocity components and kinematic pressure not greater than 1%

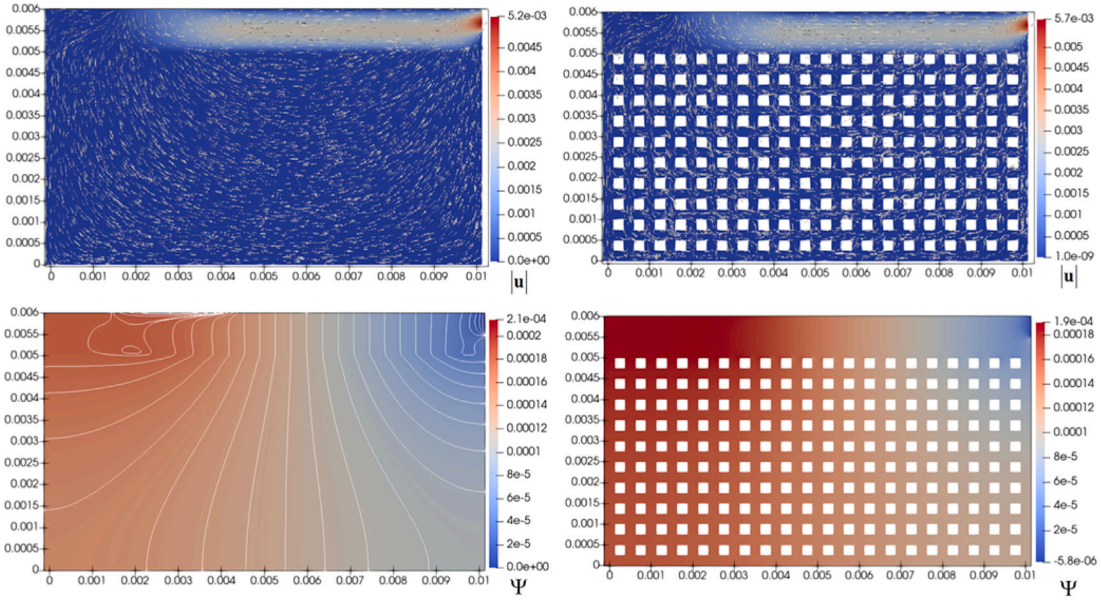


Fig. 9. Test3. Comparison between the velocity and kinematic pressure fields computed by the ODA solver (left column) and the PSS results [47] (right column). Top row: velocity (in m/s), bottom row: kinematic pressure (in $\mathrm{m}^2/\mathrm{s}^2$).

compared to the results obtained over the fine mesh above mentioned) and the computational effort. The time step size is 2.5×10^{-3} s and $CFL_{max} \simeq 1.98$.

In Fig. 9 we compare the velocity and kinematic pressure fields provided by the present ODA solver and the corresponding PSS results. The porous medium is an obstacle to the incoming flow from Γ_e, which largely deviates in the upper free fluid region, to the right, forming a channelized flow. Part of the incoming flow infiltrates the left portion of Ω_{pm} and exfiltrates through the right side, towards the outlet of the domain. The absolute value of the velocity vector in the bulk Ω_{pm} is approximately 1.5–2 orders of magnitude smaller than in Ω_{ff}. The ODA solver well reproduces the overall flow and pressure fields predicted by the PSS, with a small overestimation of Ψ close to the inflow region.

Velocity and kinematic pressure profiles are compared in Figs. 10 and 11. There, excellent agreement between the ODA and the averaged PSS solutions is observed within Ω_{pm} ($y = 0.001875$ m and $y = 0.003375$ m) and close to the interface ($y = 0.004875$ m). According to the computed v component and Ψ at $y = 0.001875$ m, $y = 0.003375$ m and $y = 0.004875$ m, as well as at $x = 0.00225$ m and $x = 0.00825$, the proposed solver correctly predicts the infiltration processes into and exfiltration processes from the porous domain. The ODA solver slightly underestimates the peak value of the u component in the Ω_{ff} region compared to the averaged PSS (positions $x = 0.00225$ m, $x = 0.00525$ m and $x = 0.00825$ m).

In Figs. 10 and 11, we also plot the solution of a *penalized* ODA solver, (see Section 1 and [26–32] for further details). This solver is obtained by assuming, in the present algorithm, a discontinuous function for ϵ and $1/K$ at the transition between Ω_{pm} and Ω_{ff}, i.e., at $y = \gamma$ (the associated results are marked with "p ODA"). This strongly affects the infiltration/exfiltration processes, as observed in these figures. The absolute value of the v component is underestimated close to the interface and along the three vertical profiles at $x = 0.00225$ m, $x = 0.00525$ m and $x = 0.00825$ m. Furthermore, nonphysical oscillations in the v profile at $x = 0.00525$ m are observed due to the interfacial stress jump. Significant underestimation of the u component within the porous medium, in the free fluid region and close to the interface, are also observed. The kinematic pressure is overestimated throughout the computational domain. This analysis shows that the "penalized" approach provides a poor estimation of the results, not only close to the interface but also within the bulk Ω_{pm} and Ω_{ff} regions, since the overall infiltration/exfiltration processes are not properly recovered, due to the choice of the discontinuous profiles of ϵ and $1/K$.

4.4. *Test 4. Comparison with pore-scale results for Navier–Stokes flow regime*

In test 4, we deal with a lid-driven cavity flow over a fibrous porous medium. The lid-driven cavity flow is an idealized paradigm of internal flows of industrial or natural processes, for example industrial microelectronics, metal casting, flows over slots on the walls of heat exchangers, dynamics of lakes,... The vortical flow structure and the related momentum transport process can be modulated by porous medium, which can be used as a passive flow control tool. The present test case has been proposed in [6]. Here we deal with a square domain $\Omega = (0, L)^2$, consisting of an upper free fluid region $\Omega_{ff} = (0, L) \times (\gamma, L)$ and a bottom porous region $\Omega_{pm} = (0, L) \times (0, \gamma)$, made of vertical fibers, with porosity $\epsilon = 0.8$, and orthotropic tensor. The setup is shown in Fig. 12, with $L = 1$ m and $\gamma = L/3$. The upper boundary moves to the right with assigned horizontal velocity $U_\infty = 1$ m/s; the other boundaries

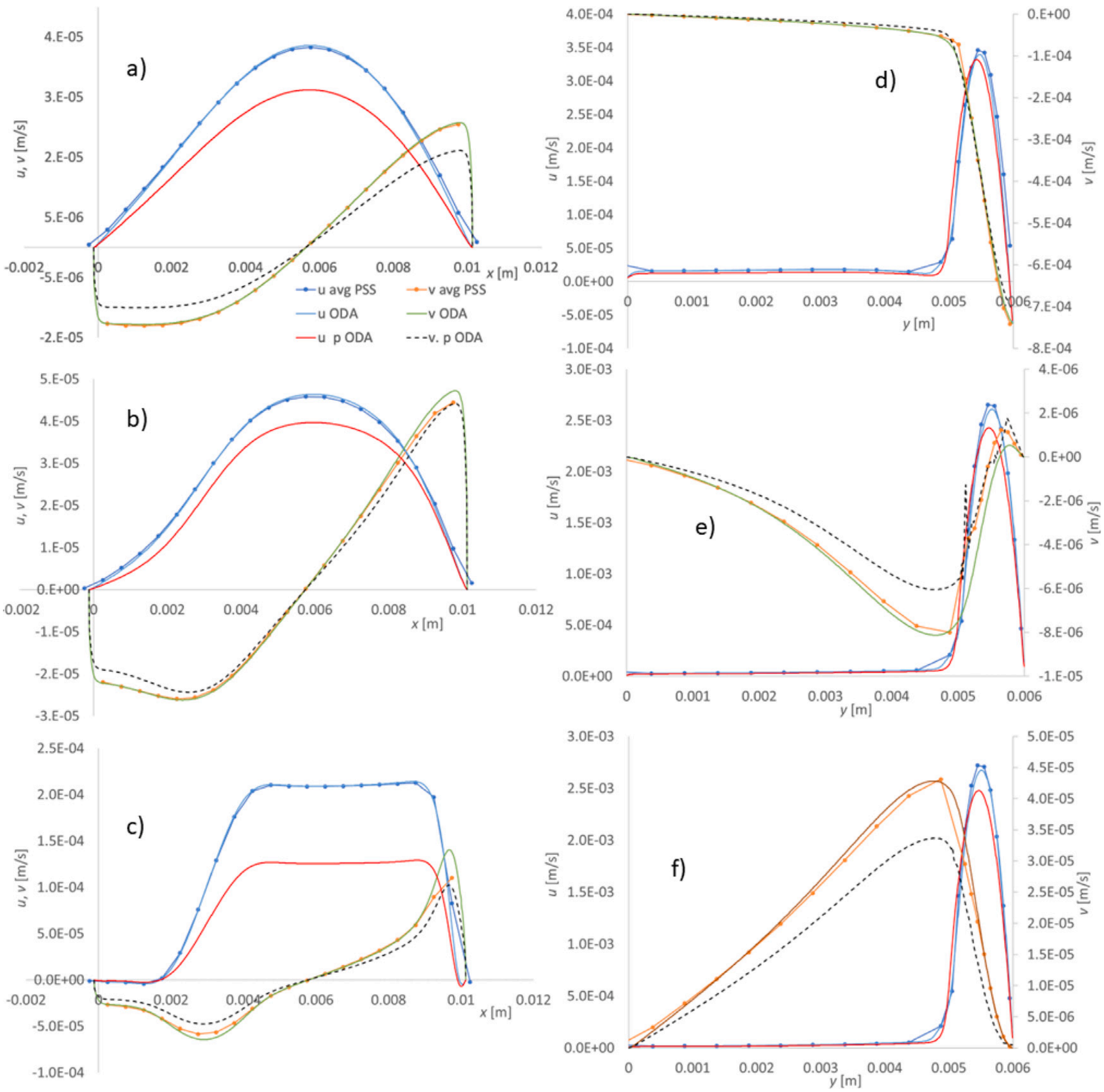


Fig. 10. Test3. Velocity components computed by the averaged PSS, the ODA solver and the penalized ODA solver (p ODA) at different positions. (a) $y = 0.001875$ m; (b) $y = 0.003375$ m; (c) $y = 0.004875$ m; (d) $x = 0.00225$ m; (e) $x = 0.00525$ m; (f) $x = 0.00825$ m.

are impervious. We ran simulations for $Re = 100$ and 1000, where $Re = \frac{U_\infty L}{\nu}$. The associated pore-scale Reynolds number within Ω_{pm} ranged between 1.75×10^{-2} and 5.125×10^{-3} [6].

The authors of [6] performed Direct Numerical Simulations (DNS) at the pore scale, where the Navier–Stokes equations with no-slip BCs on the fibers boundaries are solved, and the results have been then averaged over cubic REV volumes with side length $l_{REV} = 0.02$ m. From the averaged DNS results, they also computed the bulk permeability tensor coefficients $K^0_{i,j}, i, j = 1, 2$, as well as the extensions d_{TL_i}, measured from the top of Ω_{pm}, where $1/K_{i,j}$ decreases, almost linearly, from the bulk value to zero. More details can be found in the aforementioned paper. In Table 6 we list the values of $K^0_{i,j}$ and d_{TL_i}. Due to the orthotropy of the porous region, $K^0_{i,j} = 0$ with $i \neq j$. The averaged DNS simulations are the reference solutions for the comparison with the proposed ODA solver.

We simulated different scenarios, where the position and the size of the transition layer d_{TL} is changed (see Table 7). As before, with the nomenclature *bottom* and *middle* we refer to the position of the TL, whose top level is at $y = \gamma$ and $y = \gamma + d_{TL}/2$, respectively. We assumed linear variation of the porosity along d_{TL}, and if $d_{TL_1} \neq d_{TL_2}$, we assumed linear variation of ϵ along the average distance $\left(d_{TL_1} + d_{TL_2}\right)/2$. We performed a sensitivity analysis to the mesh size, where the reference solution was obtained over a fine mesh (uniform mesh size 0.001 m in the entire domain, $N_T = 2241989$, $N = 1122965$). The sizes δ_0 and δ_{TL} m listed in Table 7 provided a good compromise between the accuracy of the results (maximum value of the relative errors of the velocity

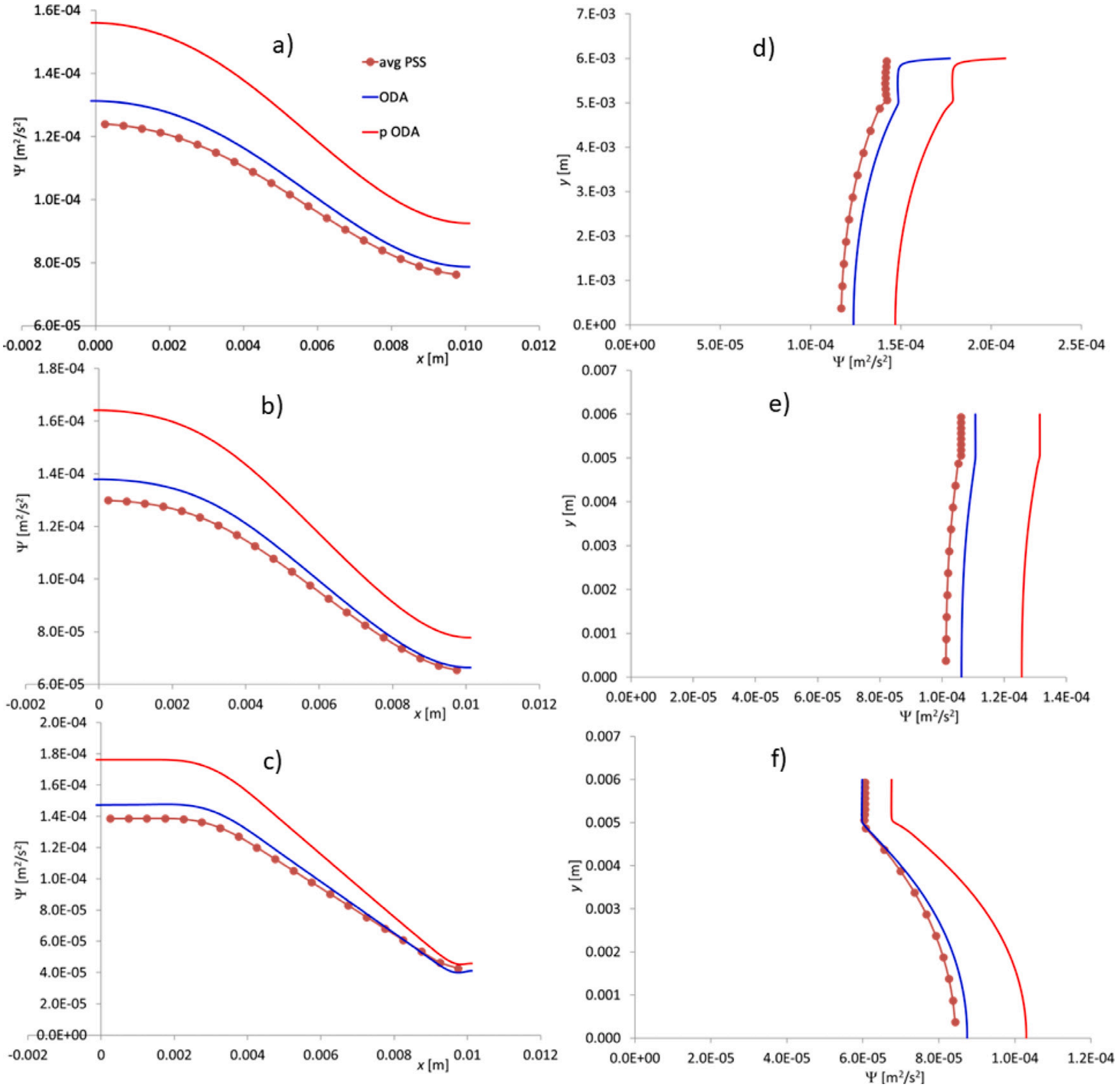


Fig. 11. Test3. Kinematic pressure computed by the averaged PSS, the ODA solver and the penalized ODA solver (p ODA) at different positions. (a) $y = 0.001875$ m; (b) $y = 0.003375$ m; (c) $y = 0.004875$ m; (d) $x = 0.00225$ m; (e) $x = 0.00525$ m; (f) $x = 0.00825$ m.

components and kinematic pressure compared to the results of the fine mesh smaller than 1%) and the computational costs. The time step size of the numerical simulation is $\Delta t = 0.02$ s and CFL_{max} ranges from 3.15 ($Re = 1000$) to 3.24 ($Re = 100$).

In Figs. 13 and 14 we compare the velocity streamlines and the pressure contours computed by the presented ODA solver with the averaged reference DNS results of [6]. Excellent agreement can be found for both Re values. As Re increases, the large vortex within Ω_{ff} moves downwards to the center of the domain, and the size of the corner vortices increases. According to the streamlines, we argue that the porous region represents an obstacle for the flow, and only a small amount of the fluid penetrates Ω_{pm}, with a velocity magnitude approximately three orders of magnitude smaller than in Ω_{ff}. The separation between the two major recirculation zones within Ω_{pm} moves to the right as Re increases from 100 to 1000. The local pressure minima are associated with the centers of the vortices, both within Ω_{ff} and Ω_{pm}.

Overall good agreement between the present model and the reference solution can be observed for the velocity components computed along the vertical center-vertical-line of the domain ($x = 0.5$ m) (see Figs. 15 and 16). The most accurate results are obtained by setting a middle position of the transition layer and the distance d_{TL_1} and d_{TL_2} to be equal to the characteristic size of the REV used for the averaging process of the DNS results (scenario 4 in Table 7). In the same Figs. 15 and 16, the solutions marked as "p ODA" are obtained in the framework of a *penalized* approach, with a discontinuity of ϵ and $1/K_{i,i}, i = 1, 2$ across the interface placed at $y = \gamma$. Due to the interfacial stress jump, u and v profiles close to interface between Ω_{pm} and Ω_{ff} are shifted. Poor estimation of the peak values is also observed in Ω_{ff}, and both velocity components are underestimated within Ω_{pm}.

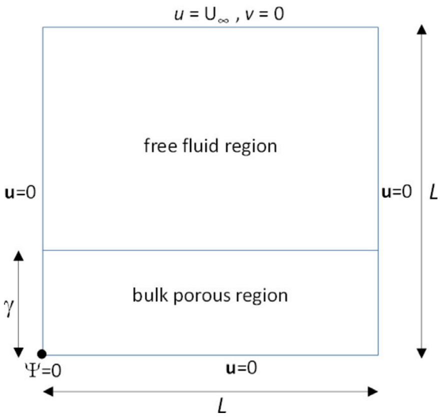


Fig. 12. Test4. Computational domain and assigned boundary conditions, with $L = 1$ m and $\gamma = L/3$.

Table 6
Test 4. Permeability coefficients (from [6]).

	$K^0_{1,1}$ [m^2]	$K^0_{2,2}$ [m^2]	d_{TL_1} [m]	d_{TL_2} [m]
Re 100	1.05×10^{-5}	2.196×10^{-2}	0.05	0.03
Re 1000	1.06×10^{-5}	2.25×10^{-5}	0.05	0.03

Table 7
Test 4. Mesh parameters for different scenarios.

Scenario	Position TL	d_{TL_1} [m]	d_{TL_2} [m]	δ_{TL} [m]	N	N_T
1	bottom	2.5 l	1.5 l	0.0015	241 187	480 781
2	bottom	1.5 l	1.5 l	0.0015	235 282	468 978
3	bottom	l	l	0.0015	233 128	464 673
4	middle	l	l	0.0015	233 948	466 351
5	bottom	0.5 l	0.5 l	0.00075	210 725	419 851
6	middle	0.5 l	0.5 l	0.00075	210 839	419 941
7	bottom	0.1 l	0.1 l	0.00015	359 986	718 352
8	bottom	0.05 l	0.05 l	0.000075	496 636	991 647
9	bottom	0.025 l	0.025 l	0.000035	793 656	1585690

4.5. Test 5. Analysis of free fluid flow over an anisotropic porous obstacle for different *Re* values

A porous obstacle invested by a fluid finds several applications (e.g., oil filters, porous coating acting as passive flow control device, porous regulating flow devices, …). The test case proposed in this Section has been proposed in [53]. Here a free fluid flows around and within an anisotropic porous obstacle Ω_{pm} with bulk porosity 0.4 (see Fig. 17). Flow is driven by a pressure drop $\Delta\Psi = \Psi_1 - \Psi_2$ between the upstream and downstream sides of the domain, while no-slip velocity BC is imposed on the bottom and top sides. This test is proposed in [53]. The kinematic viscosity of the fluid is $\nu = 1.5 \times 10^{-5}$ m^2/s. The Reynolds number is calculated as $Re = \frac{\|\mathrm{u}\|_{max} h_{ff}}{\nu}$, where $\|\mathrm{u}\|_{max}$ is the maximum value of the velocity vector magnitude in the free fluid region above the porous obstacle and h_{ff} is the fluid depth above the obstacle. We simulate two cases of $Re \ll 1$ and $Re \simeq 160$.

The anisotropic tensor $\mathbf{K}$ is computed as in Eq. (36), with coefficients $k = 1 \times 10^{-6}$ m^2, $\beta = 100$ and $\alpha \in \left[-\pi/4, \pi/4\right]$. We assume a *bottom* TL position, where the outer boundary of the TL overlaps the outer contour of the porous block (see Fig. 17) and $d_T L = 1 \times 10^{-3}$ m. We again assume that the spatial variation of the porosity and the coefficients of the inverse of the permeability tensor within TL are given by Eq. (39), with $d_\epsilon = d_K = 0$ and $\theta_\epsilon = \theta_K = 8000$.

The aim of this *showcase* test is to compare the solution of the presented ODA solver, in terms of velocity and pressure fields, as well as fluxes crossing the boundary of the porous obstacle, with the numerical solution of a TDA solver proposed in [53], which couples the Navier–Stokes equations (for compressible fluids) in Ω_{ff} to Darcy flow equation in Ω_{pm}, enforcing conservation of mass and momentum across the by interface and by applying the Beavers and Joseph slip condition at the interface [3]. A staggered-grid finite volume method is applied to discretize the Navier–Stokes equations and a MPFA finite volume method, for the discretization of the Darcy equation. The MPFA scheme is suitable to simulating anisotropic problems in porous media and does not require the computational mesh to be $\mathbf{K}$-orthogonal to the principal anisotropy directions (i.e., no specific mesh alignment along the principal direction of the permeability tensor is required). The numerical TDA-MPFA procedure is implemented in the open-source software DuMu$^{\mathrm{x}}$. More details can be found in [53].

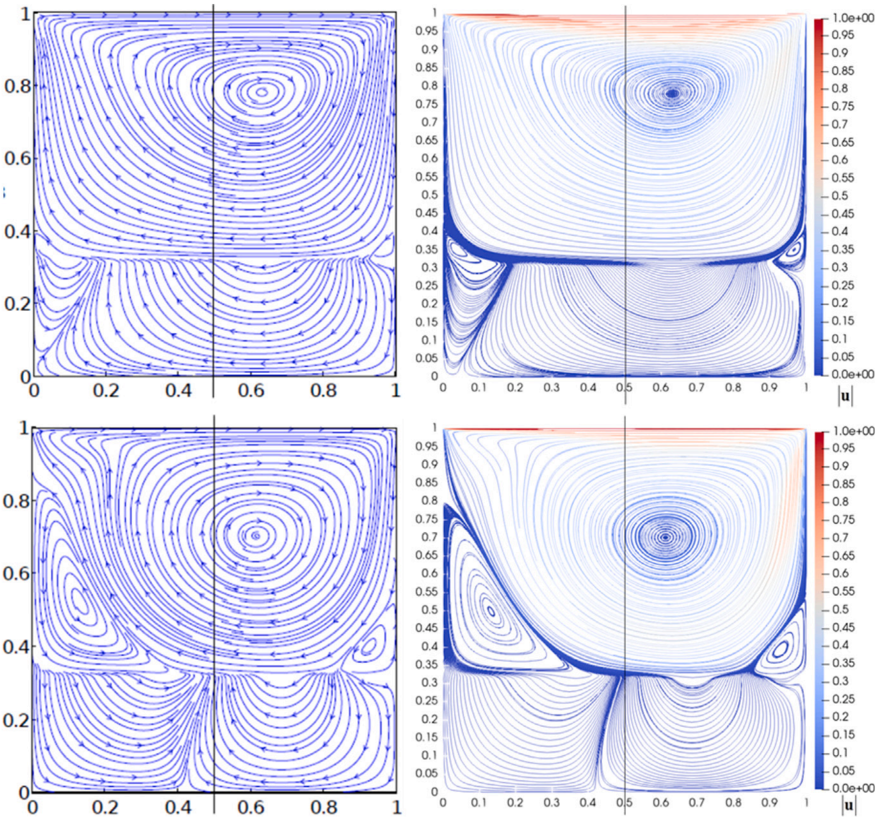


Fig. 13. Test4. Comparison between the velocity streamlines computed by the ODA solver and the averaged DNS results [6]. Left column: averaged DNS results, right column: ODA results. Top row: $Re = 100$, bottom row: $Re = 1000$ (velocity of the proposed model in m/s).

The computational domain in the case of $Re \ll 1$ is $\Omega = [0, 0.75] \times [0, 0.25]$ m^2, the porous obstacle $\Omega_{pm} = [0.25, 0.5] \times [0, 0.2]$ m^2 with pressure drop $\Delta\Psi = 1 \times 10^{-6}$ m^2/s^2 (see Fig. 17). The TDA-MPFA solver uses a structured grid with a uniform mesh size equal to 5×10^{-4} m. The adopted mesh sizes of the ODA solver runs are $\delta_0 = 5 \times 10^{-3}$ m and $\delta_{TL} = 1 \times 10^{-4}$ m, $N = 235859$ and $N_T = 468685$, $\Delta t = 4$ s and the CFL_{max} ranges from 1.13 ($\alpha = -\pi/6$) to 1.19 ($\alpha = \pi/4$).

In Fig. 18 we show the velocity and pressure fields provided by the presented ODA for $\alpha = -\pi/4$ and $\alpha = \pi/4$, in the case of $Re \ll 1$. Overall good agreement is observed with the results provided by the TDA-MPFA scheme in [53] (see Fig. 5 of the referred paper). Due to the obstacle, a channelized flow is established above it, where the highest velocity values are observed, while the velocity in the porous block is approximately 2–4 orders of magnitude smaller. The effect of anisotropy is clearly visible within Ω_{pm}, where the flow follows the principal direction of the permeability tensor, exiting ($\alpha = -\pi/4$) or entering ($\alpha = \pi/4$) at the top side. The recirculation zones simulated by the TDA-MPFA scheme within Ω_{pm} close to the bottom right corner ($\alpha = -\pi/4$) and left corner ($\alpha = \pi/4$) are slightly shifted outside Ω_{pm} in the presented ODA solutions. This could be caused by the different velocity distribution inside the TL.

For anisotropic problems, the TPFA scheme requires the computational grid to satisfy the **K**-orthogonality (see Fig. 6 in [53]). Otherwise the anisotropy is not correctly captured and the fluid flows almost horizontally in the porous block.

In the case of $Re \simeq 160$, we adopt a similar setting to the previous one, with a longer domain along the x direction. $\Omega = [0, 2.5] \times [0, 0.25]$ m^2, the porous obstacle $\Omega_{pm} = [0.4, 0.6] \times [0, 0.2]$ m^2 and pressure drop $\Delta\Psi = 2 \times 10^{-3}$ m^2/s^2. We investigate the case of $\alpha = \pi/4$. We use the same mesh sizes as before ($N = 344705$ and $N_T = 680476$), the time step size is 2.5×10^{-2} s and $CFL_{max} \simeq 9.15$. We analyzed the case of $\alpha = \pi/4$. Again, the fluid is forced to flow mainly in the narrow channel over the porous obstacle, and vortex structures within Ω_{ff} are detected after approximately 20 s. The stationary solution is achieved after a longer time ($\sim$ 200 s, in Fig. 19), compared to the case of $Re \ll 1$, with two stable countercurrent, large vortices downstream the porous block and a smaller one in front of the obstacle. Some discrepancies arise in the fluid region compared to the results provided in [53] (see Fig. 8 in the referred paper). The reason could be the different assumption of compressible fluid made in the reference study.

It is interesting to compare the fluxes across the boundaries of the porous block computed by the three numerical solvers. The case of $Re \ll 1$ is plotted in Fig. 20(a), where negative (positive) values are associated with fluxes leaving (incoming) the block. ODA and TDA-MPFA schemes provide similar results, and the discrepancies are due to the different treatment of the interface. The TDA-TPFA scheme computes correct results only for the **K**-orthogonal case, i.e. when $\alpha = 0$. Since the off-diagonal coefficients of the tensor **K** are not considered in the TDA-TPFA scheme, the associated results are independent of the direction of rotation. In

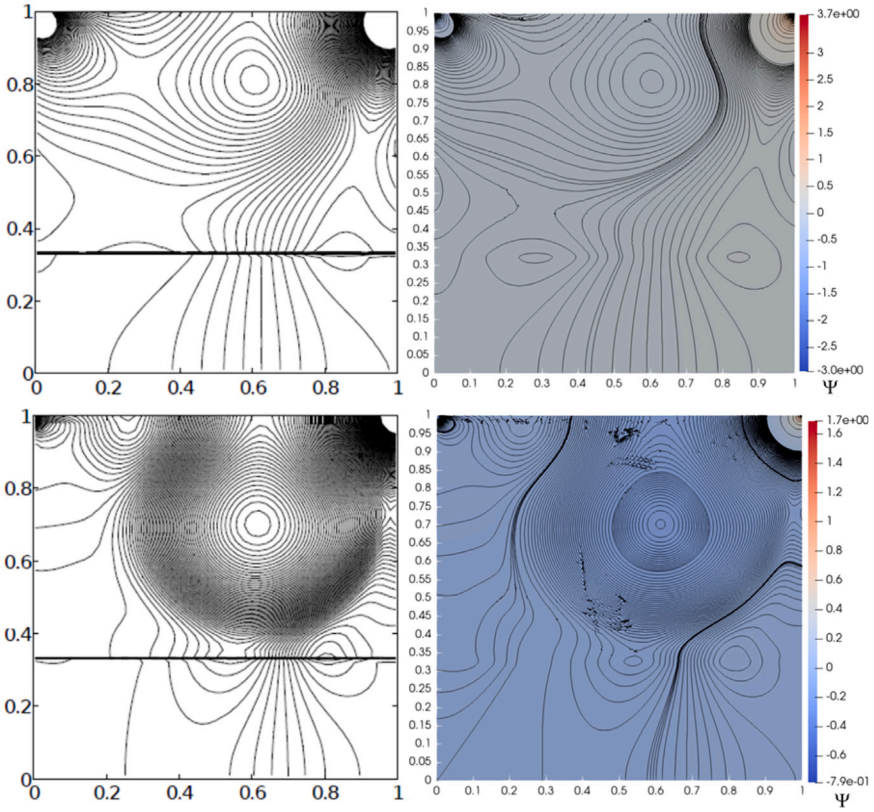


Fig. 14. Test4. Comparison between the contours of the (kinematic) pressure computed by the ODA solver and the averaged DNS results [6]. Left column: averaged DNS results, right column: ODA results. Top row: $Re = 100$, bottom row: $Re = 1000$ (kinematic pressure of the proposed model in $\mathrm{m}^2/\mathrm{s}^2$).

Fig. 20(b) we compare the time evolution of the fluxes crossing the boundary of the block for the case of $Re \simeq 160$. Again, ODA and TDA-MPFA solvers predict similar trends, with significant inflow crossing the top side of the obstacle, coming from the channel above the porous block, and a smaller amount of inflow through the downstream side of the block, due to the anisotropic effects. The results of the TDA-TPFA scheme do not match those of the two previous solvers, since again the anisotropic effects within Ω_{pm} are not properly captured.

5. Conclusions

A mesoscale ODA solver has been presented for the simulation of transfer processes between a free fluid and an anisotropic porous media. The governing equations are given by the Navier–Stokes–Brinkman equations together the continuity equation, assuming incompressible fluids. A fractional time step procedure is applied, by solving a prediction and two corrector steps within each time step. The numerical features and the associated advantages of the algorithm steps are presented and discussed. The numerical flux discretization strategy adopted in the corrector steps can be regarded as a Two-Point-Flux-Approximation (TPFA) scheme, but, unlike the standard TPFA schemes, the presented model correctly retains the anisotropy effects of the porous medium without the **K**-orthogonality grid condition. This is proved by means of several tests. Very good agreement is obtained with both reference analytical and averaged pore-scale solutions. The proposed solver overcomes the restrictions of most other ODA solvers at the mesoscale that were recently presented in the literature, such as low Reynolds numbers, 1D flow or linearization of the convective inertial terms. In some of the numerical applications we discuss the discrepancies between the solutions provided by the present solver and a macroscopic-scale ODA algorithm, where a set of *penalized* Navier–Stokes equations is solved. Compared with the reference solutions, the results of this *penalized* algorithm show significant differences throughout the whole domain, not only close to the Ω_{pm}/Ω_{ff} interface.

There is a current research activity aimed at the extension of the proposed numerical methodology to 3D domains discretized in space by unstructured meshes.

Funding

This work was supported by the German Research Foundation (DFG) for supporting this work by funding SFB 1313, Project Number 327154368, Research Project A02, and by funding SimTech via Germany's Excellence Strategy (EXC 2075 – 390740016).

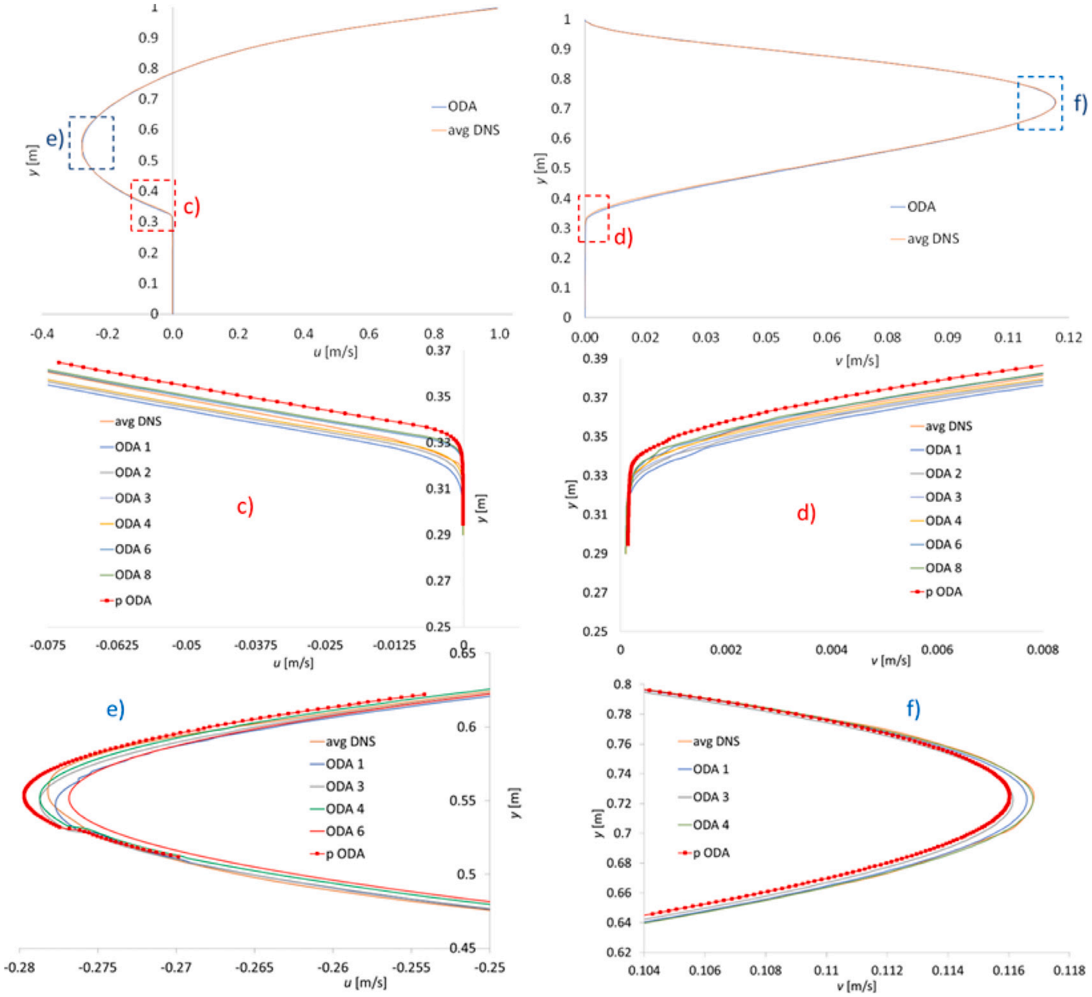


Fig. 15. Test4. Velocity components computed by the ODA solver and the averaged DNS results [6] at $x = 0.5$ m, $Re = 100$. (a) u; (b) v; (c) u, zoom in the peak value region; (d) v, zoom in the peak value region; (e) u, zoom in the transition layer region; (f) v, zoom in the transition layer region.

CRediT authorship contribution statement

Costanza Aricò: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Rainer Helmig:** Conceptualization, Formal analysis, Methodology, Project administration, Funding acquisition, Resources, Supervision, Writing – original draft, Writing – review & editing. **Daniele Puleo:** Visualization. **Martin Schneider:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology,Funding acquisition, Project administration, Resources, Writing – original draft, Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

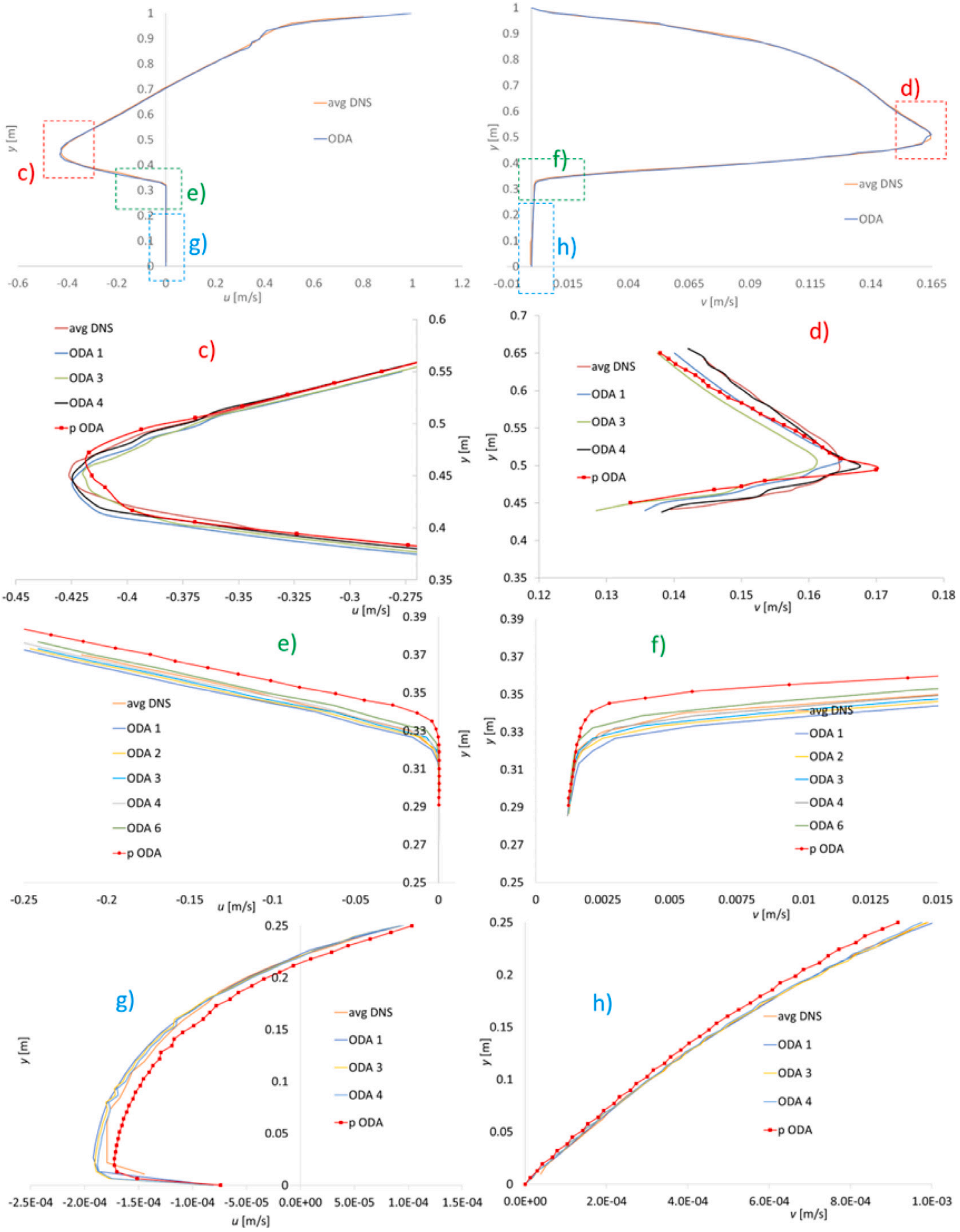


Fig. 16. Test4. Velocity components computed by the ODA solver and the averaged DNS results [6] at $x = 0.5$ m, $Re = 1000$. (a) u; (b) v; (c) u, zoom in the peak value region; (d) v, zoom in the peak value region; (e) u, zoom in the transition layer region; (f) v, zoom in the transition layer region; (g) u, zoom in the bulk Ω_{pm} region; (h) v, zoom in the bulk Ω_{pm} region.

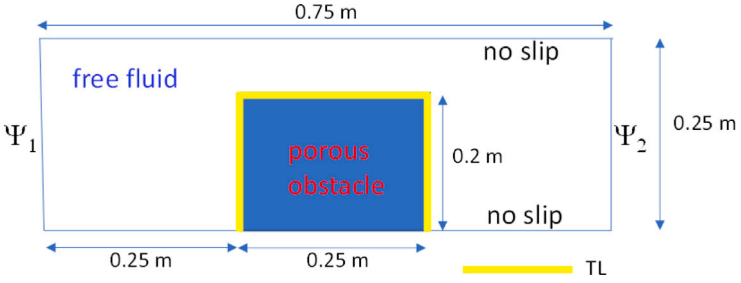


Fig. 17. Test 5. Definition sketch ($Re \ll 1$ case) and assigned BCs.

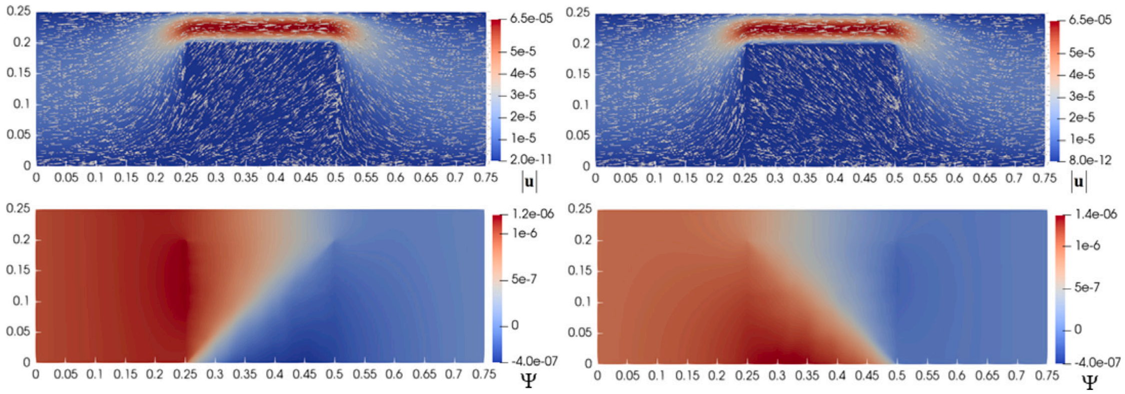


Fig. 18. Test 5. Velocity and kinematic pressure fields computed by the ODA solver for the case of $Re \ll 1$. Left column: $\alpha = -\pi/4$, right column: $\alpha = \pi/4$. Top row: $\mathbf{u}$, bottom row: Ψ (velocity in m/s, kinematic pressure in m^2/s^2).

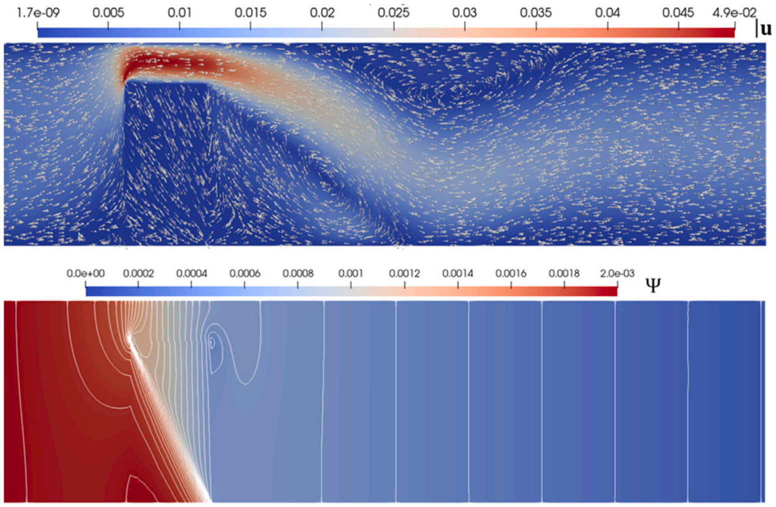
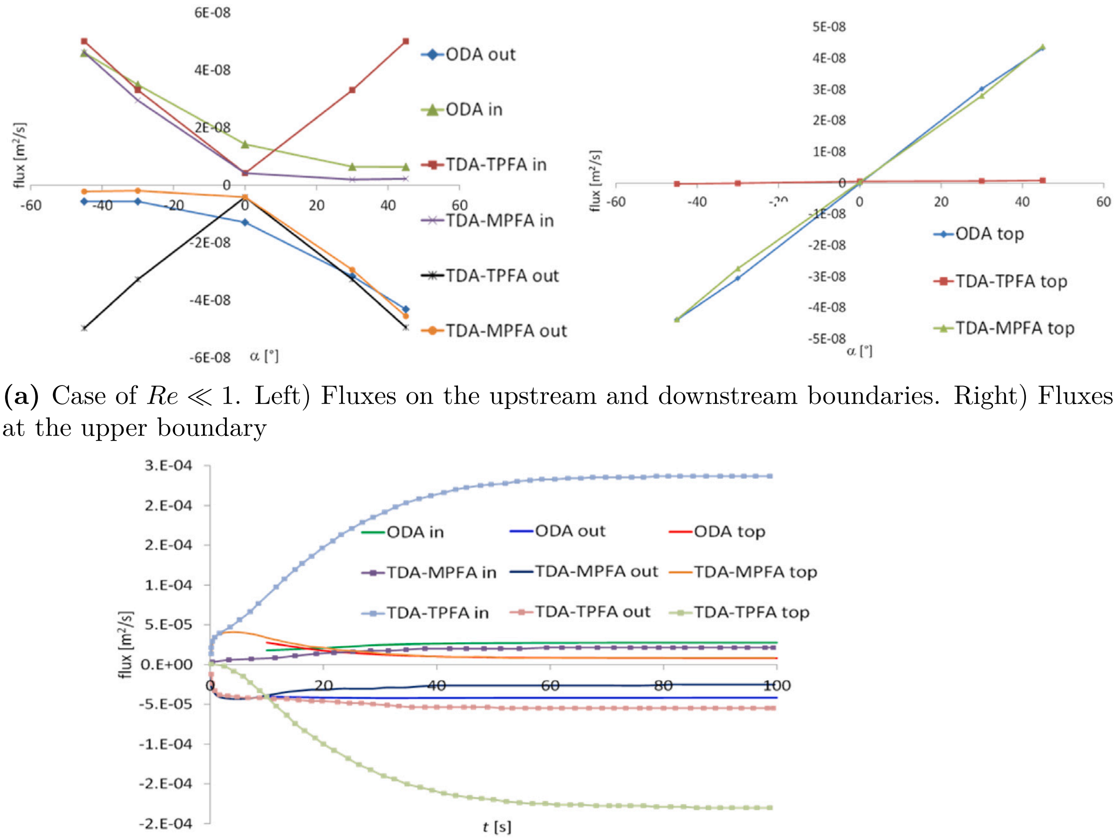


Fig. 19. Test 5. Velocity and kinematic pressure fields computed by the ODA solver, case $Re \simeq 160$, $\alpha = \pi/4$. To improve visualization, the domain is scaled by a factor of 2 along the vertical direction (velocity in m/s, kinematic pressure in m^2/s^2).



(a) Case of $Re \ll 1$. Left) Fluxes on the upstream and downstream boundaries. Right) Fluxes at the upper boundary

(b) Case of $Re \simeq 160$. Time evolution of the fluxes on the upstream, downstream and upper boundaries

Fig. 20. Test 5. Computed fluxes crossing the boundary of Ω_{pm}. Comparison between the ODA, TDA-MPFA and TDA-TPFA solvers (results of TDA-MPFA and TDA-TPFA from [53])

Appendix A. Properties of the RT0 space functions

Any function $\mathbf{u}_e \in \Re_e$, where $\Re_e$ is the lowest-order Raviart–Thomas (**RT0**) space function [16], is written as

$$\mathbf{u}_e(\mathbf{x}) = \sum_{j=1}^{3} Q_j^e \, \mathbf{w}_j^e \quad \text{with} \quad \mathbf{w}_j^e = \frac{\left(\mathbf{x} - \mathbf{x}_j\right)}{2A_e} \qquad e = 1, \dots, N_T, \tag{A.1}$$

where $\mathbf{w}_j^e$ is the jth space function of $\Re_e$, $\mathbf{x}_j$ is the coordinate vector of node j in triangle e, opposite to side j, A_e is the area of triangle e and Q_j^e is the flux crossing side j, positive outward. The properties of $\Re_e$ are

$$\nabla \cdot \mathbf{u}_e \text{ is constant inside each triangle } e \tag{A.2a}$$

$$\mathbf{u}_e \cdot \mathbf{n}_j \text{ is constant for each side } j \in e, \tag{A.2b}$$

and $\mathbf{n}_j$ is the unit vector orthogonal to side j, pointing outward. According to Eq. (A.1), the velocity components are piecewise linear inside each triangle e, and due to Eq. (A.2a), $\mathbf{u}_e$ is piecewise constant within e if $\sum_{j=1}^{3} Q_j^e = 0$. If this condition is satisfied, $\nabla \cdot \mathbf{u}_e = 0 \ \forall \mathbf{x} \in e$, $\forall e \in \Omega_T$, and, if the fluxes of two neighboring triangles are equal in value and opposite in sign along the common side, both local and global mass continuity are preserved.

Appendix B. Details of the numerical procedure applied for the CP1

We iteratively solve the two systems (18) for the components of $\Delta\tilde{\mathbf{u}}_e$ unknowns, $\Delta\tilde{u}_e$ and $\Delta\tilde{v}_e$, respectively, as described in Eq. (B.1)

$$\begin{cases} i = 0 \qquad \text{! initialize the loop counter} \\ \begin{cases} \bar{\mathbf{M}}_{x,e}\begin{pmatrix} \Delta\tilde{u}_e^i \\ 0 \end{pmatrix} A_e = \nu \sum_{j=1}^3 \frac{\Delta\tilde{u}_e^i - \Delta\tilde{u}_{ep}^i}{d_{e,ep}} l_j^e - \nu \sum_{j=1}^3 \frac{\Delta\breve{u}_e - \Delta\breve{u}_{ep}}{d_{e,ep}} l_j^e \\ \bar{\mathbf{M}}_{y,e}\begin{pmatrix} 0 \\ \Delta\tilde{v}_e^i \end{pmatrix} A_e = \nu \sum_{j=1}^3 \frac{\Delta\tilde{v}_e^i - \Delta\tilde{v}_{ep}^i}{d_{e,ep}} l_j^e - \nu \sum_{j=1}^3 \frac{\Delta\breve{v}_e - \Delta\breve{v}_{ep}}{d_{e,ep}} l_j^e \end{cases} \\ \begin{cases} \text{do while } (err > toll) \\ \begin{cases} \bar{\mathbf{M}}_{x,e}\begin{pmatrix} \Delta\tilde{u}_e^{i+1} \\ \Delta\tilde{v}_e^i \end{pmatrix} A_e = \nu \sum_{j=1}^3 \frac{\Delta\tilde{u}_e^{i+1} - \Delta\tilde{u}_{ep}^{i+1}}{d_{e,ep}} l_j^e - \nu \sum_{j=1}^3 \frac{\Delta\breve{u}_e - \Delta\breve{u}_{ep}}{d_{e,ep}} l_j^e \\ \bar{\mathbf{M}}_{y,e}\begin{pmatrix} \Delta\tilde{u}_e^i \\ \Delta\tilde{v}_e^{i+1} \end{pmatrix} A_e = \nu \sum_{j=1}^3 \frac{\Delta\tilde{v}_e^{i+1} - \Delta\tilde{v}_{ep}^{i+1}}{d_{e,ep}} l_j^e - \nu \sum_{j=1}^3 \frac{\Delta\breve{v}_e - \Delta\breve{v}_{ep}}{d_{e,ep}} l_j^e \end{cases} \\ i = i + 1 \qquad \text{! update the loop counter} \\ \text{end do} \end{cases} \\ , \end{cases} \tag{B.1a}$$

$$\text{with } \bar{\mathbf{M}}_{x,e} = \begin{pmatrix} \bar{M}_{1,1}^e \\ \bar{M}_{1,2}^e \end{pmatrix}^T \qquad \bar{\mathbf{M}}_{y,e} = \begin{pmatrix} \bar{M}_{2,1}^e \\ \bar{M}_{2,2}^e \end{pmatrix}^T, \tag{B.1b}$$

$$err = \min(err_x, err_y) \text{ with } \begin{cases} err_x = \frac{\sqrt{\left(\|\Delta\tilde{u}_e^{i+1} - \Delta\tilde{u}_e^i\|\right)}}{\sqrt{\|\Delta\tilde{u}_e^i\|}} \\ err_y = \frac{\sqrt{\left(\|\Delta\tilde{v}_e^{i+1} - \Delta\tilde{v}_e^i\|\right)}}{\sqrt{\|\Delta\tilde{v}_e^i\|}}, \end{cases} \tag{B.1c}$$

where i is the counter of the iterations in the iterative procedure and $1d-04 \le toll \le 1d-03$.

The diagonal and off-diagonal matrix coefficients of system (21), denoted as $\mathcal{A}_{e,e}^{x(y),CP1}$ and $\mathcal{A}_{e,ep}^{x(y),CP1}$, as well as the coefficients of the source term vectors, denoted as $\mathcal{B}_e^{x(y),CP1}$, are given in Eq. (B.2)

$$\begin{aligned} \mathcal{A}_{e,e}^{x,CP1} &= \bar{M}_{1,1}^e A_e + \nu \sum_{j=1}^3 \frac{l_j^e}{d_{e,ep}}, \qquad \mathcal{A}_{e,e}^{y,CP1} = \bar{M}_{2,2}^e A_e + \nu \sum_{j=1}^3 \frac{l_j^e}{d_{e,ep}}, \\ \mathcal{A}_{e,ep}^{x(y),CP1} &= -\nu \frac{l_j^e}{d_{e,ep}}, \\ \mathcal{B}_e^{x,CP1} &= \nu \sum_{j=1}^3 \frac{\Delta\breve{u}_e - \Delta\breve{u}_{ep}}{d_{e,ep}} l_j^e - \bar{M}_{1,2}^e A_e \Delta\tilde{v}_e^i, \\ \mathcal{B}_e^{y,CP1} &= \nu \sum_{j=1}^3 \frac{\Delta\breve{v}_e - \Delta\breve{v}_{ep}}{d_{e,ep}} l_j^e - \bar{M}_{2,1}^e A_e \Delta\tilde{u}_e^i, \end{aligned} \tag{B.2}$$

If the mesh satisfies the Delaunay property, $d_{e,ep}$ are positive (see its definition after Eq. (15)), and the matrices are also positive-definite, so that the $\mathcal{M}$-matrix property is guaranteed [40].

Appendix C. Details of the numerical procedure applied for the CP2

Starting from Eq. (C.1),

$$\left(\Xi_j \nabla\eta\right) \cdot \mathbf{n}_j = \left(\Xi_j \mathbf{n}_j\right) \cdot \nabla\eta, \tag{C.1}$$

applying a co-normal decomposition, we obtain (see also Fig. 3)

$$\Xi_j \mathbf{n}_j = \mathbf{d}_j \quad \text{with} \quad \mathbf{d}_j = \mathbf{d}_{j,n} + \mathbf{d}_{j,\tau} \quad \text{and} \quad \mathbf{d}_{j,\tau} = \mathbf{d}_{j,\tau_{n_1}} + \mathbf{d}_{j,\tau_{n_2}}, \tag{C.2}$$

where the vectors $\mathbf{d}_{j,n}$ and $\mathbf{d}_{j,\tau}$ are parallel to the directions $\mathbf{n}_j$ and $\boldsymbol{\tau}_j$, respectively, with $\boldsymbol{\tau}_j$ the unit vector tangential to side l_j^e. Let $\mathbf{n}_l$ and $\mathbf{n}_m$ (with $l = 1, 2$ and $m = 3, 4$), be the unit vector orthogonal to the sides shared by cells e and ep_l, and by cells ep and

ep_m, respectively, pointing outwards from e and ep, respectively (see Fig. 3). According to the last relation in Eq. (C.2), vector $\mathbf{d}_{j,\tau}$ is decomposed along the $\mathbf{n}_1$ and $\mathbf{n}_2$ directions.

Starting from Eq. (C.2), we discretize the dot product $\mathbf{d}_j \cdot \nabla\eta$ in Eq. (26) as

$$\mathbf{d}_j \cdot \nabla\eta = \left(\mathbf{d}_{j,n} + \mathbf{d}_{j,\tau}\right) \cdot \nabla\eta \qquad \text{with} \tag{C.3a}$$

$$\mathbf{d}_{j,n} \cdot \nabla\eta = \frac{1}{2}\left(\frac{\eta_{ep}-\eta_e}{d_{e,ep}} - \frac{\eta_e-\eta_{ep}}{d_{e,ep}}\right) d_{j,n} \qquad \text{and} \tag{C.3b}$$

$$\begin{gathered}\mathbf{d}_{j,\tau} \cdot \nabla\eta = \frac{1}{2}\left(\frac{\eta_{ep_1}-\eta_e}{d_{e,ep_1}} d_{\tau_{n_1}}\alpha_1 + \frac{\eta_{ep_2}-\eta_e}{d_{e,ep_2}} d_{\tau_{n_2}}\alpha_2\right) - \\ \frac{1}{2}\left(\frac{\eta_{ep_3}-\eta_{ep}}{d_{ep,ep_3}} d_{\tau_{n_3}}\alpha_3 + \frac{\eta_{ep_4}-\eta_{ep}}{d_{ep,ep_4}} d_{\tau_{n_4}}\alpha_4\right)\end{gathered}, \tag{C.3c}$$

where, with the help of Fig. 3, $d_{j,n} = \mathbf{d}_j \cdot \mathbf{n}_j$, $d_{\tau_{n_{l(m)}}} = \mathbf{d}_{j,\tau} \cdot \mathbf{n}_{l(m)}$, η_e, η_{ep}, $\eta_{ep_{l(m)}}$ are the values of η in the circumcenters of cells e, ep, and $ep_{l(m)}$, respectively, $d_{e,ep}$, d_{e,ep_l} and d_{e,ep_m} are the distances with a sign of the circumcenters of cells e and ep, e and ep_l, ep and ep_m respectively, and coefficient $\alpha_{l(m)} = 1$ if $\mathbf{d}_{j,\tau} \cdot \mathbf{n}_{l(m)} > 0$ otherwise $\alpha_{l(m)} = -1$.

According to Eqs.(C.1) to (C.3), Eq. (26) becomes

$$\begin{gathered}\sum_{j=1}^{3}\left(\overline{Fl}_j^e\right) = \sum_{j=1}^{3}\left(\frac{\eta_e-\eta_{ep}}{d_{e,ep}} d_{j,n}\right) l_j^e + \\ \frac{1}{2}\sum_{j=1}^{3}\left(\sum_{l=1,2}\frac{\eta_e-\eta_{ep_l}}{d_{e,ep_l}} d_{\tau_{n_l}}\alpha_l + \frac{1}{2}\sum_{m=3,4}\frac{\eta_{ep_m}-\eta_{ep}}{d_{ep,ep_m}} d_{\tau_{n_m}}\alpha_m\right) l_j^e\end{gathered}, \tag{C.4}$$

and Eq. (C.4) form a system to be solved for the η unknowns.

The diagonal and off-diagonal matrix coefficients of the systems (28)–(29a), denoted as $\mathcal{A}_{e,e}^{CP2}$ and $\mathcal{A}_{e,ep}^{CP2}$, are

$$\mathcal{A}_{e,e}^{CP2} = \sum_{j=1}^{3}\frac{l_j^e}{d_{e,ep}} d_{j,n} \qquad \mathcal{A}_{e,ep}^{CP2} = -\frac{l_j^e}{d_{e,ep}} d_{j,n}, \tag{C.5}$$

and the coefficients of the source term vectors of systems (28)–(29a), denoted as $\mathcal{B}_{1,e}^{CSP}$ and $\mathcal{B}_{2,e}^{CP2}$, respectively, are

$$\begin{gathered}\mathcal{B}_{1,e}^{CP2} = \sum_{j=1}^{3}\left(\bar{F}l_j^e\right), \\ \mathcal{B}_{2,e}^{CP2} = \sum_{j=1}^{3}\left(\bar{F}l_j^e\right) - \\ \frac{1}{2}\sum_{j=1}^{3}\left(\sum_{l=1,2}\frac{\tilde{\eta}_e-\tilde{\eta}_{ep_l}}{d_{e,ep_l}} d_{\tau_{n_l}}\alpha_l + \sum_{m=3,4}\frac{\tilde{\eta}_{ep}-\tilde{\eta}_{ep_m}}{d_{ep,ep_m}} d_{\tau_{n_m}}\alpha_m\right) l_j^e.\end{gathered} \tag{C.6}$$

References

[1] M. Chandesris, D. Jamet, Boundary conditions at a fluid–porous interface: An a priori estimation of the stress jump coefficients, Int. J. Heat Mass Transfer 50 (17) (2007) 3422–3436, http://dx.doi.org/10.1016/j.ijheatmasstransfer.2007.01.053.

[2] K. Mosthaf, K. Baber, B. Flemisch, R. Helmig, A. Leijnse, I. Rybak, B. Wohlmuth, A coupling concept for two-phase compositional porous-medium and single-phase compositional free flow, Water Resour. Res. 47 (10) (2011) http://dx.doi.org/10.1029/2011WR010685.

[3] G. Beavers, D.D. Joseph, Boundary conditions at a naturally permeable wall, J. Fluid Mech. 30 (1) (1967) 197–207, http://dx.doi.org/10.1017/S0022112067001375.

[4] J.A. Ochoa-Tapia, S. Whitaker, Momentum transfer at the boundary between a porous medium and a homogeneous fluid—I. Theoretical development, Int. J. Heat Mass Transfer 38 (14) (1995) 2635–2646, http://dx.doi.org/10.1016/0017-9310(94)00346-W.

[5] J.A. Ochoa-Tapia, S. Whitaker, Momentum transfer at the boundary between a porous medium and a homogeneous fluid—II. Comparison with experiment, Int. J. Heat Mass Transfer 38 (14) (1995) 2647–2655, http://dx.doi.org/10.1016/0017-9310(94)00347-X.

[6] G.A. Zampogna, A. Bottaro, Fluid flow over and through a regular bundle of rigid fibres, J. Fluid Mech. 792 (2016) 5–35, http://dx.doi.org/10.1017/jfm.2016.66.

[7] U. Lācis, S. Bagheri, A framework for computing effective boundary conditions at the interface between free fluid and a porous medium, J. Fluid Mech. 812 (2017) 866–889, http://dx.doi.org/10.1017/jfm.2016.838.

[8] I. Rybak, C. Schwarzmeier, E. Eggenweiler, U. Rüde, Validation and calibration of coupled porous-medium and free-flow problems using pore-scale resolved model, Comput. Geosci. 25 (2021) 621–635, http://dx.doi.org/10.1007/s10596-020-09994-x.

[9] M. Discacciati, A. Quarteroni, Analysis of a domain decomposition method for the coupling of Stokes and Darcy equations, in: F. Brezzi, A. Buffa, S. Corsaro, A. Murli (Eds.), Numerical Mathematics and Advanced Applications, Springer Milan, Milano, 2003, pp. 3–20.

[10] B. Rivière, I. Yotov, Locally Conservative Coupling of Stokes and Darcy Flows, SIAM J. Numer. Anal. 42 (5) (2005) 1959–1977, http://dx.doi.org/10.1137/S0036142903427640.

[11] W.J. Layton, F. Schieweck, I. Yotov, Coupling fluid flow with porous media flow, SIAM J. Numer. Anal. 40 (6) (2002) 2195–2218, http://dx.doi.org/10.1137/S0036142901392766.

[12] D. Vassilev, C. Wang, I. Yotov, Domain decomposition for coupled Stokes and Darcy flows, Comput. Methods Appl. Mech. Engrg. 268 (2014) 264–283, http://dx.doi.org/10.1016/j.cma.2013.09.009, URL https://www.sciencedirect.com/science/article/pii/S0045782513002363.

[13] V. Girault, D. Vassilev, I. Yotov, Mortar multiscale finite element methods for Stokes–Darcy flows, Numer. Math. 127 (1) (2014) 93–165, http://dx.doi.org/10.1007/s00211-013-0583-z.

[14] B. Ganis, D. Vassilev, C. Wang, I. Yotov, A multiscale flux basis for mortar mixed discretizations of Stokes–Darcy flows, Comput. Methods Appl. Mech. Engrg. 313 (2017) 259–278, http://dx.doi.org/10.1016/j.cma.2016.09.037, URL https://www.sciencedirect.com/science/article/pii/S004578251630874X.

[15] G. Kanschat, B. Rivière, A strongly conservative finite element method for the coupling of Stokes and Darcy flow, J. Comput. Phys. 229 (17) (2010) 5933–5943, http://dx.doi.org/10.1016/j.jcp.2010.04.021.

[16] P.A. Raviart, J.M. Thomas, A mixed finite element method for 2-nd order elliptic problems, in: I. Galligani, E. Magenes (Eds.), Mathematical Aspects of Finite Element Methods, Springer Berlin Heidelberg, 1977, pp. 292–315.

[17] S. Badia, R. Codina, Stabilized continuous and discontinuous Galerkin techniques for Darcy flow, Comput. Methods Appl. Mech. Engrg. 199 (25) (2010) 1654–1667, http://dx.doi.org/10.1016/j.cma.2010.01.015.

[18] B. Goyeau, D. Lhuillier, D. Gobin, M. Velarde, Momentum transport at a fluid–porous interface, Int. J. Heat Mass Transfer 46 (21) (2003) 4071–4081, http://dx.doi.org/10.1016/S0017-9310(03)00241-2.

[19] M.L. Bars, M.G. Worster, Interfacial conditions between a pure fluid and a porous medium: implications for binary alloy solidification, J. Fluid Mech. 550 (2006) 149–173, http://dx.doi.org/10.1017/S0022112005007998.

[20] W. Gray, A derivation of the equations for multi-phase transport, Chem. Eng. Sci. 30 (2) (1975) 229–233, http://dx.doi.org/10.1016/0009-2509(75)80010-8.

[21] S. Whitaker, Flow in porous media. I: A theoretical derivation of Darcy's law, Transp. Porous Media 1 (1986) 3–25, http://dx.doi.org/10.1007/BF01036523.

[22] K. Tao, J. Yao, Z. Huang, Analysis of the laminar flow in a transition layer with variable permeability between a free-fluid and a porous medium, Acta Mech. 224 (2013) 1943–1955, http://dx.doi.org/10.1007/s00707-013-0852-z.

[23] H. Chen, X.-P. Wang, A one-domain approach for modeling and simulation of free fluid over a porous medium, J. Comput. Phys. 259 (2014) 650–671, http://dx.doi.org/10.1016/j.jcp.2013.12.008.

[24] F.J. Valdés-Parada, D. Lasseux, A novel one-domain approach for modeling flow in a fluid-porous system including inertia and slip effects, Phys. Fluids 33 (2) (2021) http://dx.doi.org/10.1063/5.0036812, 022106.

[25] F.J. Valdés-Parada, D. Lasseux, Flow near porous media boundaries including inertia and slip: A one-domain approach, Phys. Fluids 33 (7) (2021) http://dx.doi.org/10.1063/5.0056345, 073612.

[26] K. Khadra, P. Angot, S. Parneix, J.-P. Caltagirone, Fictitious domain approach for numerical modelling of Navier–Stokes equations, Internat. J. Numer. Methods Fluids 34 (8) (2000) 651–684, http://dx.doi.org/10.1002/1097-0363(20001230)34:8<651::AID-FLD61>3.0.CO;2-D.

[27] C.-H. Bruneau, I. Mortazavi, Passive control of the flow around a square cylinder using porous media, Internat. J. Numer. Methods Fluids 46 (4) (2004) 415–433, http://dx.doi.org/10.1002/fld.756.

[28] C.-H. Bruneau, I. Mortazavi, Numerical modelling and passive flow control using porous media, Comput. & Fluids 37 (5) (2008) 488–498, http://dx.doi.org/10.1016/j.compfluid.2007.07.001, Special Issue Dedicated to Professor M.M. Hafez on the Occasion of his 60th Birthday.

[29] X. Xie, J. Xu, G. Xue, Uniformly-stable finite element methods for Darcy-Stokes-Brinkman models, J. Comput. Math. 26 (3) (2008) 437–455.

[30] F. Cimolin, M. Discacciati, Navier–Stokes/Forchheimer models for filtration through porous media, Appl. Numer. Math. 72 (2013) 205–224, http://dx.doi.org/10.1016/j.apnum.2013.07.001.

[31] A. Parasyris, M. Discacciati, D.B. Das, Mathematical and numerical modelling of a circular cross-flow filtration module, Appl. Math. Model. 80 (2020) 84–98, http://dx.doi.org/10.1016/j.apm.2019.11.016.

[32] L.B.A. Nillama, J. Yang, L. Yang, An explicit stabilised finite element method for Navier-Stokes-Brinkman equations, J. Comput. Phys. 457 (2022) 111033, http://dx.doi.org/10.1016/j.jcp.2022.111033.

[33] C. Aricò, T. Tucciarelli, The MAST FV/FE scheme for the simulation of two-dimensional thermohaline processes in variable-density saturated porous media, J. Comput. Phys. 228 (4) (2009) 1234–1274, http://dx.doi.org/10.1016/j.jcp.2008.10.015.

[34] C. Aricò, M. Sinagra, T. Tucciarelli, The MAST-edge centred lumped scheme for the flow simulation in variably saturated heterogeneous porous media, J. Comput. Phys. 231 (4) (2012) 1234–1274, http://dx.doi.org/10.1016/j.jcp.2011.10.012.

[35] C. Aricò, M. Sinagra, T. Tucciarelli, Anisotropic potential of velocity fields in real fluids: Application to the MAST solution of shallow water equations, Adv. Water Resour. 62 (2013) 13–36, http://dx.doi.org/10.1016/j.advwatres.2013.09.010.

[36] C. Aricò, M. Sinagra, C. Picone, T. Tucciarelli, MAST-RT0 solution of the incompressible Navier–Stokes equations in 3D complex domains, Eng. Appl. Comput. Fluid Mech. 15 (1) (2021) 53–93, http://dx.doi.org/10.1080/19942060.2020.1860830.

[37] C. Aricò, M. Sinagra, Z. Driss, T. Tucciarelli, A new solver for incompressible non-isothermal flows in natural and mixed convection over unstructured grids, Appl. Math. Model. 103 (2022) 445–474, http://dx.doi.org/10.1016/j.apm.2021.10.042.

[38] C. Aricò, T. Tucciarelli, Monotonic solution of heterogeneous anisotropic diffusion problems, J. Comput. Phys. 252 (2013) 219–249, http://dx.doi.org/10.1016/j.jcp.2013.06.017.

[39] S. Perron, S. Boivin, J.-M. Hérard, A finite volume method to solve the 3D Navier–Stokes equations on unstructured collocated meshes, Comput. & Fluids 33 (10) (2004) 1305–1333, http://dx.doi.org/10.1016/j.compfluid.2003.10.006.

[40] F. Letniowski, Three-dimensional delaunay triangulations for finite element approximations to a second-order diffusion operator, SIAM J. Sci. Stat. Comput. 13 (3) (1992) 765–770, http://dx.doi.org/10.1137/0913045.

[41] A. Younes, P. Ackerer, F. Lehmann, A new mass lumping scheme for the mixed hybrid finite element method, Internat. J. Numer. Methods Engrg. 67 (1) (2006) 89–107, http://dx.doi.org/10.1002/nme.1628.

[42] J. Lambert, Computer solution of ordinary differential equations, Comput. J. 19 (2) (1976) 155, http://dx.doi.org/10.1093/comjnl/19.2.155.

[43] C.-J. Lin, J.J. Moré, Incomplete Cholesky factorizations with limited memory, SIAM J. Sci. Comput. 21 (1) (1999) 24–45, http://dx.doi.org/10.1137/S1064827597327334.

[44] J.A. Scott, M. Tůma, HSL_MI28: An efficient and robust limited-memory incomplete Cholesky factorization code, ACM Trans. Math. Software 40 (4) (2014) http://dx.doi.org/10.1145/2617555.

[45] R.H. Magnus, S. Eduard, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1952) 409–435.

[46] J.J. Dongarra, I.S. Duff., D.C. Sorensen, H.V.D. Vorst, Solving Linear Systems on Vector and Shared Memory Computers, Society for Industrial and Applied Mathematics, USA, 1990.

[47] T. Koch, D. Gläser, K. Weishaupt, S. Ackermann, M. Beck, B. Becker, S. Burbulla, H. Class, E. Coltman, S. Emmert, T. Fetzer, C. Grüninger, K. Heck, J. Hommel, T. Kurz, M. Lipp, F. Mohammadi, S. Scherrer, M. Schneider, G. Seitz, L. Stadler, M. Utz, F. Weinhardt, B. Flemisch, DuMux 3 – an open-source simulator for solving flow and transport problems in porous media with a focus on model coupling, Comput. Math. Appl. 81 (2021) 423–443, http://dx.doi.org/10.1016/j.camwa.2020.02.012, Development and Application of Open-source Software for Problems with Numerical PDEs.

[48] J. Ahrens, B. Geveci, C. Law, ParaView: An end-user tool for large data visualization, in: Visualization Handbook, Elesvier, 2005, ISBN 978-0123875822.

[49] D.A. Nield, A.V. Kuznetsov, The effect of a transition layer between a fluid and a porous medium: shear flow in a channel, Transp. Porous Media 78 (2009) 477–487, http://dx.doi.org/10.1007/s11242-009-9342-0.

[50] Wolfram Research, Inc., Mathematica, version 13.2, 2022, Champaign, IL, URL https://www.wolfram.com/mathematica.

[51] F. Mohammadi, E. Eggenweiler, B. Flemisch, S. Oladyshkin, I. Rybak, M. Schneider, K. Weishaupt, A surrogate-assisted uncertainty-aware Bayesian validation framework and its application to coupling free flow and porous-medium flow, Comput. Geosci. 27 (2023) 663–686, http://dx.doi.org/10.1007/s10596-023-10228-z.

[52] K. Yazdchi, S. Srivastava, S. Luding, Microstructural effects on the permeability of periodic fibrous porous media, Int. J. Multiph. Flow 37 (8) (2011) 956–966, http://dx.doi.org/10.1016/j.ijmultiphaseflow.2011.05.003.

[53] M. Schneider, K. Weishaupt, D. Gläser, W.M. Boon, R. Helmig, Coupling staggered-grid and MPFA finite volume methods for free flow/porous-medium flow problems, J. Comput. Phys. 401 (2020) 109012, http://dx.doi.org/10.1016/j.jcp.2019.109012.