Journal of Computational Physics 306 (2016) 199–215

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Improved PISO algorithms for modeling density varying flow in conjugate fluid–porous domains

M. Nordlund [a,*], M. Stanic [b], A.K. Kuczaj [a,b], E.M.A. Frederix [b], B.J. Geurts [b,c]

[a] Philip Morris International R&D, Philip Morris Products S.A., Quai Jeanrenaud 5, 2000 Neuchâtel, Switzerland
[b] Multiscale Modeling and Simulation, Faculty EEMCS, J.M. Burgers Center, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[c] Anisotropic Turbulence, Fluid Dynamics Laboratory, Faculty of Applied Physics, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 11 June 2015
Received in revised form 6 November 2015
Accepted 12 November 2015
Available online 21 November 2015

Keywords:
Porous
PISO
Rhie–Chow
Spurious oscillations
Collocated
Segregated
Finite volume method

ABSTRACT

Two modified segregated PISO algorithms are proposed, which are constructed to avoid the development of spurious oscillations in the computed flow near sharp interfaces of conjugate fluid–porous domains. The new collocated finite volume algorithms modify the Rhie–Chow interpolation to maintain a correct pressure–velocity coupling when large discontinuous momentum sources associated with jumps in the local permeability and porosity are present. The Re-Distributed Resistivity (RDR) algorithm is based on spreading flow resistivity over the grid cells neighboring a discontinuity in material properties of the porous medium. The Face Consistent Pressure (FCP) approach derives an auxiliary pressure value at the fluid–porous interface using momentum balance around the interface. Such derived pressure correction is designed to avoid spurious oscillations as would otherwise arise with a strictly central discretization. The proposed algorithms are successfully compared against published data for the velocity and pressure for two reference cases of viscous flow. The robustness of the proposed algorithms is additionally demonstrated for strongly reduced viscosity, i.e., higher Reynolds number flows and low Darcy numbers, i.e., low permeability of the porous regions in the domain, for which solutions without unphysical oscillations are computed. Both RDR and FCP are found to accurately represent porous media flow near discontinuities in material properties on structured grids.

1. Introduction

Fluid flow through porous media is fundamental to many natural and industrial processes, such as groundwater flows, filtration, and chemical and biomass processing [1–5]. In order to efficiently simulate these processes and predict their characteristic properties, robust and accurate numerical models are of high importance. While the equations governing the flow and heat transfer in porous media are readily specified using the method of volume averaging [6] in conjunction with suitable closure models, it often remains challenging to obtain physically acceptable numerical solutions in the vicinity of fluid–porous interfaces. Without special care in the algorithm development, discontinuities in the material properties, particularly the resistivity, at the interface may yield spurious oscillations in the solution. This is especially true for high

* Corresponding author.
E-mail address: markus.nordlund@pmi.com (M. Nordlund).

http://dx.doi.org/10.1016/j.jcp.2015.11.035
0021-9991/

Reynolds (Re) number flows at low Darcy (Da) number, for which the jump in flow resistivity is high, while damping effects of viscosity are rather low. The occurrence of spurious oscillations is particularly pronounced when segregated algorithms are applied, in which the velocity and pressure equations are solved separately and an iterative solution process is required. Numerical schemes that avoid spurious oscillations at sharp interfaces were proposed by [7] and [8] for both structured and unstructured grids using a collocated variable, finite volume block-coupled solver. In this case pressure and velocity are solved simultaneously. These rather complex schemes require special, local treatment of the fluid–porous interfaces. In this paper we turn to segregated solvers, which are more commonly adopted in Computational Fluid Dynamics (CFD) studies and present alternative treatments that are equally well suited to alleviate the problem of spurious oscillations. We compare the RDR [9] and FCP methods in the context of the Pressure Implicit with Splitting of Operators (PISO) algorithm and assess their relative performance in some detail for several test cases of increasing complexity. It will be shown that both RDR and FCP approaches are effective in eliminating spurious oscillations.

A correction to the Rhie–Chow interpolation in case the fluid is subject to large body forces was suggested by [10] and a force field discretization rule for the volume-of-fluid method was suggested by [11] as remedies for the occurring spurious velocity oscillations. These correction schemes may be applied for explicit treatment of the flow resistance terms present for porous media flow. However, due to the stiffness of the equations such explicit treatment suffers from very small time steps which render the method less effective.

In this paper two modified Rhie–Chow/PISO algorithms are proposed, which are constructed to avoid the development of spurious oscillations in the solution when large discontinuous momentum sources are present. The RDR algorithm is based on a redistribution of the discontinuous flow resistivity over grid cells adjacent to the discontinuity in the material properties, in a similar way as was proposed by [12] for discontinuous body force fields. In order not to create time step restrictions for high Re, low Da numbers, for which the resistance is high, this term is treated implicitly. The FCP algorithm uses an approach similar to that presented by Oxtoby et al. [11], where interface pressure values are derived by ensuring a force balance based on the momentum equations on both sides of the fluid–porous interface. This is aimed at eliminating spurious oscillations [12]. Differently from Oxtoby et al. [11] the FCP method treats all terms, except pressure, implicitly, thus relaxing the time step constraint imposed by high Re/low Da numbers.

Throughout this paper we adopt a numerical formulation, which solves the equations directly for the pressure field, rather than the density field as is common for compressible flows. The selected formulation is computationally cheaper than the density-based formulation for density varying flows, and does not require resolution of high frequency pressure fluctuations (i.e., sound propagation) [13,14]. The methods are implemented in the open source OpenFOAM® environment, which is based on collocated grid discretization, storing all variable fields at cell centers [15].

The proposed algorithms are successfully compared with published data for the velocity and pressure fields for incompressible, isothermal flow through a porous plug [7] and for flow parallel to a porous region, i.e., the so-called Beavers–Joseph problem [16]. These problems are used as benchmarks, since their particular geometry is known to cause oscillatory solutions in case standard interpolation methods are used for the resistivity term. The so-called porous resistance term in the momentum equation, which is the major source of the discontinuity in the domain, is a function of velocity and permeability. The two relevant flow parameters are Re and Da numbers. At fixed length-scale and fluid properties, high velocity corresponds to high Re number and low permeability corresponds to low Da number. An increase/decrease of these parameters is directly influencing the magnitude of the porous resistance term. The robustness of the proposed algorithms is demonstrated for both high Re number flows and low Da numbers, which constitutes the most challenging situation. RDR is found to successfully remove spurious oscillations compared to the reference model which adopts central discretization, for all tested values of Re and Da numbers. FCP is found to be adequate for most cases involving lower Re and higher Da numbers, while for higher Re and lower Da numbers some oscillations can still be seen in the solution.

Spurious numerical oscillations of velocity or any other variable can have a big impact on delicate physical processes such as aerosol formation and evolution. Even slight variations in thermodynamic variables can have a large influence on, e.g., the nucleation rate, which is at the core of the formation of aerosol droplets. Such strong dependencies on rather subtle changes in the local conditions emphasizes the need for an accurate treatment of fluid–porous interfaces where oscillations may occur. This paper provides simple and effective numerical methods to suppress the aforementioned oscillations also in case one relies on segregated solvers with a collocated arrangement of variables.

The organization of this paper is as follows. In Section 2, the governing equations for weakly density varying flow and heat transfer in conjugate fluid–porous domains are introduced. In Section 3 the finite volume discretization is described and an analysis of the discretized equations for large, discontinuous source terms is given. Thereafter, Re-Distributed Resistivity (RDR) and Face Consistent Pressure (FCP) algorithms are put forward in Sections 4 and 5 followed by their validation in Section 6 and conclusions in Section 7.

2. Governing equations

Fluid flow and conjugate heat transfer in fluid–porous domains can be described by the volume-averaged mass, momentum and energy conservation equations [6]. Imagine a volume V, occupied by two constituents α (fluid) and β (solid), such that $V = V_\alpha + V_\beta$. Then the extrinsic, or superficial, average and the intrinsic volume average of a fluid property φ are defined as: $\langle\varphi_\alpha\rangle = \frac{1}{V}\int_{V_\alpha}\varphi_\alpha dV$ and $\langle\varphi_\alpha\rangle^\alpha = \frac{1}{V_\alpha}\int_{V_\alpha}\varphi_\alpha dV$, respectively [6]. The $\langle\rangle$ brackets imply volume averaging over volume V and for intrinsic properties the exponent above the brackets $\langle\rangle^\alpha$ or $\langle\rangle^\beta$ implies volume averaging only over

volume V_α or V_β, respectively. The relationship between intrinsic and extrinsic volume averaged properties is defined as $\langle\varphi_\alpha\rangle^\alpha = \phi\langle\varphi_\alpha\rangle$, where $\phi = V_\alpha/V$ is the porosity of the porous medium.

Following the derivations in [6] and the closure modeling in [17] and [18], the equations governing the flow of fluid and heat in isotropic, heterogeneous porous media, consisting of a fluid phase α and a solid phase β are presented. The mass conservation for fluid flow is

$$\partial_t(\phi\langle\rho_\alpha\rangle^\alpha) + \partial_i(\langle\rho_\alpha\rangle^\alpha\langle u_i\rangle) = 0 \tag{1}$$

where t is the time, ρ the density and u_i the velocity in the x_i direction. Because only weak density variations are considered, the mass dispersion term originating from the volume averaging procedure is neglected. The operator ∂_t is the temporal partial derivative and ∂_i is the partial derivative with respect to the spatial coordinate x_i. We adopt a mixed formulation in which the extrinsically averaged velocity field $\langle u_i\rangle$ transports the intrinsically averaged fluid mass density $\langle\rho_\alpha\rangle^\alpha$ in the volume V. The momentum equation for fluid flow is

$$\partial_t(\langle\rho_\alpha\rangle^\alpha\langle u_i\rangle) + \partial_j(\phi^{-1}\langle\rho_\alpha\rangle^\alpha\langle u_j\rangle\langle u_i\rangle) = -\phi\partial_i\langle p_\alpha\rangle^\alpha + \partial_j\langle\tau_{ij}\rangle + \phi\langle\zeta_i\rangle^\alpha - \phi D\langle u_i\rangle \tag{2}$$

with the volume-averaged rate of strain tensor

$$\langle\tau_{ij}\rangle = \langle\mu_\alpha\rangle^\alpha(\partial_j\langle u_i\rangle + \partial_i\langle u_j\rangle) - (\frac{2}{3}\langle\mu_\alpha\rangle^\alpha - \langle\kappa_\alpha\rangle^\alpha)\delta_{ij}\partial_k\langle u_k\rangle \tag{3}$$

In Eqs. (2) and (3) p is the pressure, ζ_i a momentum body source including also higher-order perturbation terms originating from the volume averaging procedure. The dynamic viscosity is denoted μ, δ_{ij} is the Kronecker delta tensor and κ the dilatational viscosity. The key term for this paper is the isotropic porous resistance term $\phi D\langle u_i\rangle$, the last term in Eq. (2), which takes a finite value in the porous media and zero otherwise. This porous resistance term is the source of discontinuity in the domain, and thus the primary cause of spurious oscillations, as later will be described in detail. It is defined as a scalar $D = \langle\mu_\alpha\rangle^\alpha(K^{-1} + K^{-1}F)$, where K is the isotropic permeability scalar and $F = \frac{\langle\rho_\alpha\rangle^\alpha}{\langle\mu_\alpha\rangle^\alpha}K^{1/2}|\langle u_j\rangle|c_E$ the non-Darcy resistivity, in which c_E is the form drag coefficient, that together with the permeability accounts for the microscopic structure of the porous media.

The temperature equations for the fluid and solid constituents are respectively:

$$\begin{aligned}&\langle c_{p,\alpha}\rangle^\alpha\left(\partial_t(\phi\langle\rho_\alpha\rangle^\alpha\langle T_\alpha\rangle^\alpha) + \partial_i(\langle\rho_\alpha\rangle^\alpha\langle u_i\rangle\langle T_\alpha\rangle^\alpha)\right)\\ &\quad= \partial_i(\lambda_\alpha^e\partial_i\langle T_\alpha\rangle^\alpha) + h_{\alpha\beta}s_{\alpha\beta}(\langle T_\beta\rangle^\beta - \langle T_\alpha\rangle^\alpha) + \partial_t(\phi\langle p_\alpha\rangle^\alpha) + \langle u_i\rangle\partial_i\langle p_\alpha\rangle^\alpha + \hat{\Lambda}_\alpha\end{aligned} \tag{4}$$

$$\langle c_{p,\beta}\rangle^\beta\langle\rho_\beta\rangle^\beta\partial_t((1-\phi)\langle T_\beta\rangle^\beta) = \partial_i(\lambda_\beta^e\partial_i\langle T_\beta\rangle^\beta) - h_{\alpha\beta}s_{\alpha\beta}(\langle T_\beta\rangle^\beta - \langle T_\alpha\rangle^\alpha) \tag{5}$$

where $c_{p,m}$ is the specific heat at constant pressure, T_m the temperature, λ_m^e the effective thermal conductivity for the phase $m \in \{\alpha, \beta\}$, including both the stagnant effective conductivity and tortuosity effects. The interfacial heat transfer coefficient is denoted by $h_{\alpha\beta}$ and $s_{\alpha\beta}$ is the specific surface area (surface available per unit of porous media volume). Thermal dispersion is denoted $\hat{\Lambda}_\alpha$. In Eqs. (2) to (5) the closure terms suggested in [17,18] were adopted and local thermal non-equilibrium between the two phases was assumed.

This is to illustrate that the so-called one-domain approach [19] is taken, solving the described equations for the entire fluid–porous domain, rather than splitting the domain into parts which are governed by multiple sets of equations and imposing interfacial boundary conditions in places where they merge. In the next section the discretization of the volume-averaged governing equations for the fluid flow in conjugate fluid–porous domains is presented.

3. Finite volume discretization and modified Rhie–Chow interpolation

Application of the Finite Volume (FV) method requires flux evaluation at cell faces, and interpolation of cell centered variables onto the cell faces. Segregated algorithms for FV methods, such as PISO [20], are based on either iterative or stepwise procedures for obtaining the solution. PISO consists of two main steps: the predictor and the corrector step. Both of these steps require evaluation of pressure gradients at cell faces for calculating pressure gradient values. An incorrect evaluation of these pressure gradients and pressure face values leads to pressure–velocity decoupling or the 'checker-board' pressure pattern, which is a well known problem on collocated grids [21,15]. In 1983, Rhie and Chow proposed a method which maintains the pressure–velocity coupling on collocated grids even when using segregated solvers [21]. This method is today known as the Rhie–Chow interpolation and will be described in some detail in this section to provide a point of reference for understanding the modifications in RDR and FCP as presented momentarily.

The Rhie–Chow interpolation was developed assuming governing equations without discontinuous body forces acting on the flow. As shown in Section 2, this is not the case when porous media are present in part of the domain. When body force discontinuities are present, in particular in the porous resistance term $\phi D\langle u_i\rangle$, Rhie–Chow interpolation fails to maintain the pressure–velocity coupling resulting in 'velocity ripples', as was discussed earlier by Mencinger et al. [12]. The exact mechanism behind its failure will also be presented here. In this section we will present a modified version of the Rhie–Chow interpolation which does preserve pressure–velocity coupling even when large body force discontinuities are present in the flow. In order to discretize the governing equations using a collocated FV method, the equations are

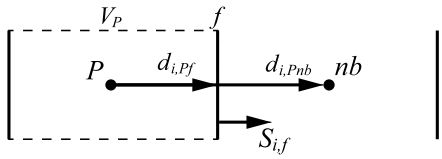

Fig. 1. Example cell of a collocated grid for FV analysis. The owner cell is denoted by P, the neighbor cell by nb. The vector $d_{i,Pnb}$ is the distance vector between the node P and the node nb. The common face of cells P and nb is denoted by f. The surface normal vector of the face f is denoted $S_{i,f}$; this vector has a magnitude equal to the surface area of the face f. The vector between the cell center P and face center is denoted with $d_{i,Pf}$. The index i stands for the ith coordinate direction.

integrated over a control volume V_P centered around a node P and bounded by N faces with surface area vector $S_{i,f}$ and face centers f, as shown in Fig. 1.

Applying Gauss' divergence theorem and using Einstein's tensor notation, the semi-discretized form of the volume-averaged mass, momentum and energy equations for a collocated variable discretization yields:

$$V_P \partial_t(\phi_P \rho_P) + \sum_f (\rho u_i)_f S_{i,f} = 0 \tag{6}$$

$$V_P \partial_t(\rho_P u_{i,P}) + \sum_f (\phi^{-1} \rho u_j)_f u_{i,f} S_{j,f} = -\phi_P \sum_f p_f S_{i,f} + \sum_f \tau_{ij} S_{j,f} + \phi_P \zeta_{i,P} V_P - \phi_P D_P u_{i,P} V_P \tag{7}$$

$$\begin{aligned} V_P c_{p,P}[\partial_t(\phi_P \rho_P T_P) + \sum_f (\rho u_i)_f T_f S_{i,f}] \\ = \sum_f \lambda^e_f (\partial_i T)_f S_{i,f} + V_P h_{\alpha\beta,P} s_{\alpha\beta,P}(T_{\beta,P} - T_P) + V_P \partial_t(\phi_P p_P) + u_{i,P} \sum_f p_f S_{i,f} + V_P \hat{\Lambda}_P \end{aligned} \tag{8}$$

$$V_P (c_{p,\beta} \rho_\beta)_P \partial_t((1-\phi_P) T_{\beta,P}) = \sum_f \lambda^e_{\beta,f} (\partial_i T_\beta)_f S_{i,f} - V_P h_{\alpha\beta,P} s_{\alpha\beta,P}(T_{\beta,P} - T_P) \tag{9}$$

Here and in the sequel the spatial averaging operators $\langle\rangle^\alpha$ and $\langle\rangle$ and the subscript α for the fluid phase are dropped for brevity, such that, i.e., $\langle T_\alpha \rangle^\alpha_P = T_P$ and $\langle T_\beta \rangle^\beta_P = T_{\beta,P}$. Note that u_i refers to the superficial velocity $\langle u_i \rangle$ and the other variables are intrinsic.

In the discretized equations above, some of the properties and variables are required at the cell centers and some at the face centers. In a collocated variable representation, the properties and variables are all stored in the cell-centers and need to be interpolated by interpolation schemes to the face centers. Applying discretization and interpolation schemes for the terms and variables in (7), dividing by V_P and replacing the discretized pressure gradient with its non-discretized form for ease of notation, the semi-discretized momentum equation for the node P can be written as:

$$A_P u_{i,P} = H_{i,P} - \phi_P (\partial_i p)_P \tag{10}$$

A_P is a scalar containing all the central coefficients, including implicit time derivative contributions, and $H_{i,P}$ is a vector containing contributions from the neighboring cells, as well as time derivative contributions. A_P and $H_{i,P}$ are left unspecified as their exact discretized form depends on the choice of discretization. More details on the discretization of A_P and $H_{i,P}$ can be found in [20,22]. The choices of the discretization schemes of this particular work are specified in Section 6. Dividing by A_P results in the following equation:

$$u_{i,P} = A_P^{-1} H_{i,P} - A_P^{-1} \phi_P (\partial_i p)_P \tag{11}$$

giving us the expression for the velocity at cell P. Eq. (11) is used in both the *predictor* and *corrector* steps of the PISO algorithm [20]. In the predictor step it is used to find the initial solution of the velocity prior to entering the corrector loop, where the same equation is used to update the velocity field with the improved pressure field.

3.1. Discretized pressure equation and Rhie–Chow interpolation

In the chosen formulation there is no explicit equation for the density in the conservation equations. The primary reason for choosing this formulation is computational efficiency, as density-based solvers tend to require significantly smaller time steps and resolve pressure field details, such as sound wave propagation [14], that are not of importance to our application. In the selected formulation, the pressure equation is obtained by combining momentum and mass conservation. Once the pressure field is solved for, we obtain the density field through the use of the equation-of-state for which we adopt the ideal gas law.

To derive the equation for the pressure Eq. (11) is multiplied by ρ and thereafter inserted into the mass conservation equation (6). Furthermore, replacing ρ in the time derivative with the equation of state:

$$\rho = \psi p \tag{12}$$

where ψ is the compressibility factor, results in the following pressure equation:

$$V_P \partial_t(\phi_P \psi_P p_P) + \sum_f (\rho A^{-1} H_i)_f S_{i,f} = \sum_f (\rho A^{-1} \phi \partial_i p)_f S_{i,f} \tag{13}$$

In Eq. (13), $(A^{-1}\phi\partial_i p)_f$ needs to be specified using an appropriate interpolation rule based on cell-centered values of the variables. One way to determine $(\rho A^{-1}\phi\partial_i p)_f$ is to interpolate $A_P^{-1}\phi_P(\partial_i p)_P$ and its neighbor value $A_{nb}^{-1}\phi_{nb}(\partial_i p)_{nb}$ onto the cell face f using linear interpolation defined as:

$$\Gamma_f = (1-r)\Gamma_P + r\Gamma_{nb} \tag{14}$$

where $r = |d_{i,Pf}|/|d_{i,Pnb}|$, and Γ is the property to be interpolated. However, if this would be adopted on a uniform, collocated grid the pressure–velocity decoupling would be inevitable as the contribution of the central pressure value (node P) would cancel out in the pressure gradient calculation for cell P [15]. It is at this point that Rhie and Chow [21] proposed that instead of directly interpolate this term onto the faces it is approximated as $(\rho A^{-1}\phi\partial_i p)_f = (\rho A^{-1}\phi)_f(\partial_i p)_f$, where $(\partial_i p)_f$ is calculated directly as:

$$(\partial_i p)_f = \hat{n}_{i,f}\frac{p_{nb} - p_P}{|d_{i,Pnb}|} \tag{15}$$

with $\hat{n}_{i,f}$ being the unit normal vector. Adopting the method of Rhie and Chow, the pressure equation takes the following form:

$$V_P \partial_t(\phi_P \psi_P p_P) + \sum_f (\rho A^{-1} H_i)_f S_{i,f} = \sum_f (\rho A^{-1}\phi)_f (\partial_i p)_f S_{i,f} \tag{16}$$

As it is well known, this approach ensures pressure–velocity coupling and avoids the unphysical 'checker-board' flow pattern. Anticipating further modifications in the next two sections we remark that this Rhie–Chow approximation is only practical in case the governing equations do not contain discontinuous body forces. Modifications that cope also with discontinuous body forces will be formulated in the RDR and FCP methods.

In the discretization of the governing equations, the mass flux over the cell faces plays an important role and needs to be regularly updated within the algorithm. The expression for the update of mass flux follows from the pressure equation (16):

$$(\rho u_i)_f S_{i,f} = (\rho A^{-1} H_i)_f S_{i,f} - (\rho A^{-1}\phi)_f (\partial_i p)_f S_{i,f} \tag{17}$$

where, again, the pressure gradient at the face is directly computed using the Rhie–Chow assumption from Eq. (15).

3.2. Modified Rhie–Chow interpolation for discontinuous body forces

This section presents a modified Rhie–Chow interpolation method making it suitable for domains that incorporate body force discontinuities. We show the proposed modifications, starting from the momentum equation and derive novel pressure and corrector step equations. For easier notation, we revert the discretized pressure term $\sum_f p_f S_{i,f}$ from Eq. (7) back to its non-discretized form $(\partial_i p)_P$.

To avoid interpolation of the discontinuous ϕ from the cell-centers to the face centers in the Laplacian term $(\sum_f (\rho A^{-1}\phi)_f (\partial_i p)_f S_{i,f})$ of the pressure equation (13), the discretized momentum equation (16) is divided by ϕ_P. Further division by V_P yields:

$$\frac{1}{\phi_P}\partial_t(\rho_P u_{i,P}) + \frac{1}{\phi_P V_P}\sum_f (\phi^{-1}\rho u_j)_f u_{i,f} S_{j,f} = -(\partial_i p)_P + \frac{1}{\phi_P V_P}\sum_f \tau_{ij,f} S_{j,f} + \zeta_{i,P} - D_P u_{i,P} \tag{18}$$

Instead of discretizing and inserting $D_P u_{i,P}$ as a coefficient into A_P and $H_{i,P}$, as was done in (10), $D_P u_{i,P}$ is kept as a separate term, independent of A_P and $H_{i,P}$. The rest of the terms, except the pressure gradient term, are discretized according to:

$$A_P u_{i,P} - H_{i,P} = \frac{1}{\phi_P}\partial_t(\rho_P u_{i,P}) + \frac{1}{\phi_P^2 V_P}\sum_f (\rho u_j)_f u_{i,f} S_{j,f} - \frac{1}{\phi_P V_P}\sum_f \tau_{ij,f} S_{j,f} - \zeta_{i,P} \tag{19}$$

Note that the reciprocal porosity in the convective term of (19) has been moved outside of the sum to avoid interpolation of the discontinuous porosity to the cell-faces. This approximation is relevant only at the interface, and is generally small in character since porosity values are bounded between 0 and 1. The resulting semi-discretized momentum equation takes the form:

$$A_P u_{i,P} + D_P u_{i,P} = H_{i,P} - (\partial_i p)_P \tag{20}$$

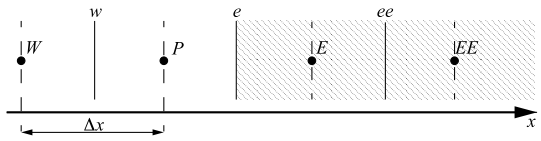

Fig. 2. Example cell in the vicinity of the fluid–porous interface. The owner cell is denoted by P, the east neighbor cell by E and the west neighbor by W. The local spatial step Δx is assumed uniform for ease of analysis. Cell faces from west to east are denoted by w, e and ee, respectively. The hatched areas symbolize the porous region.

Dividing it first by A_P and then collecting the velocity terms, the cell-centered velocity becomes:

$$u_{i,P} = B_P A_P^{-1}(H_{i,P} - (\partial_i p)_P) \tag{21}$$

where $B_P = (1 + D_P A_P^{-1})^{-1}$. We emphasize that only the matrix B_P contains the porous resistivity D_P. This is important to maintain a strong pressure–velocity coupling throughout the algorithm. In order to avoid oscillations near the discontinuity and to interpolate the discontinuous variables consistently to the interface in the face mass flux expression, B is interpolated separately to the faces resulting in the following modified Rhie–Chow interpolation:

$$(\rho u_i)_f S_{i,f} = B_f[(\rho A^{-1} H_i)_f S_{i,f} - (\rho A^{-1})_f (\partial_i p)_f S_{i,f}] \tag{22}$$

where $B_f = (1 + D_f (A^{-1})_f)^{-1}$. The terms $(\rho A^{-1} H_i)_f$, $(\rho A^{-1})_f$, D_f and $(A^{-1})_f$ are found using the linear interpolation in (14), and $(\partial_i p)_f$ is discretized using (15). Note that the expression in the square brackets in Eq. (22) is in fact the original Rhie–Chow [21] expression for mass flux, Eq. (17). We keep the original Rhie–Chow method for the continuous or slowly varying variables (density and velocity), and isolate the contribution from the large discontinuity in the matrix B_f.

In order to obtain the modified pressure equation, we insert Eq. (22) into the discretized mass conservation equation (6) and replace ρ_P in the time derivative with (12) to find:

$$V_P \partial_t(\phi_P \psi_P p_P) + \sum_f B_f (\rho A^{-1} H_i)_f S_{i,f} = \sum_f B_f (\rho A^{-1})_f (\partial_i p)_f S_{i,f} \tag{23}$$

The modified velocity definition Eq. (21) and the modified Rhie–Chow interpolation for the mass flux are key steps in accounting for the discontinuous resistivity contribution. This modified Rhie–Chow interpolation constitutes a new simulation method that in principle incorporates discontinuities in D. As will be shown in numerical illustrations in Section 6, this particular modification by itself does not eliminate velocity oscillations to a sufficient degree. This drives us toward the construction of further adaptations that more effectively handle such spurious oscillations, which, nevertheless, follow the pattern as laid down in Eq. (21). The first option for further modifications addresses the definition of the matrix B_P – this option will be referred to as the RDR approach. Alternatively, a second option presents itself through an adaptation of the pressure gradient in Eq. (21) – this option will be referred to as the FCP approach, closely following Oxtoby et al. [11]. In the next two sections we will elaborate the RDR and FCP options respectively before coming to a detailed assessment of the original Rhie–Chow approach Eq. (11), the modified Rhie–Chow approach Eq. (21) and the further RDR and FCP modifications in Section 6.

3.3. Pressure–velocity decoupling for discontinuous body forces

The Rhie–Chow interpolation and the PISO algorithm were proposed for single-phase fluid flows, for which the material properties are smooth and continuously varying in response to temperature and density changes. The Rhie–Chow interpolation as described previously has been shown to result in spurious pressure and velocity oscillations in the vicinity of discontinuities in porous properties or in general large source terms in the governing equations [7,8,10,12]. There are two reasons why Rhie–Chow interpolation fails and we will dedicate some space to clarify these. For the purpose of clarification, we concentrate on 1D example grid in the vicinity of a fluid–porous interface, shown in Fig. 2.

The first reason for the failure of Rhie–Chow interpolation in the presence of discontinuities is that the Rhie–Chow approximation $(\rho A^{-1}\phi \partial_i p)_f = (\rho A^{-1}\phi)_f (\partial_i p)_f$, is inaccurate near faces where large body force discontinuities arise. This can be easily verified by writing out the above approximation in its discrete form for the grid shown in Fig. 2:

$$\frac{1}{2}\Big[\rho_P A_P^{-1}\phi_P \frac{p_e - p_w}{\Delta x} + \rho_E A_E^{-1}\phi_E \frac{p_{ee} - p_e}{\Delta x}\Big] = \frac{1}{2}\Big[\rho_P A_P^{-1}\phi_P + \rho_E A_E^{-1}\phi_E\Big]\Big(\frac{p_E - p_P}{\Delta x}\Big) \tag{24}$$

with the local spatial step $\Delta x = const$, and face values of pressure p_w, p_e and p_{ee} being interpolated using Eq. (14). Expanding the left hand side of Eq. (24) using the interpolation Eq. (14) and some algebra we get that

$$\begin{aligned}
&\frac{1}{2}\Big[\rho_P A_P^{-1}\phi_P \frac{p_e - p_w}{\Delta x} + \rho_E A_E^{-1}\phi_E \frac{p_{ee} - p_e}{\Delta x}\Big] \\
&= \frac{1}{2}\Big[\rho_P A_P^{-1}\phi_P + \rho_E A_E^{-1}\phi_E\Big]\Big(\frac{p_E - p_P}{\Delta x}\Big) + \frac{1}{2}\Big[\rho_P A_P^{-1}\phi_P \frac{1}{2}\Big(\frac{p_P - p_W}{\Delta x} - \frac{p_E - p_P}{\Delta x}\Big) \\
&\quad - \rho_E A_E^{-1}\phi_E \frac{1}{2}\Big(\frac{p_E - p_P}{\Delta x} - \frac{p_{EE} - p_E}{\Delta x}\Big)\Big]
\end{aligned} \tag{25}$$

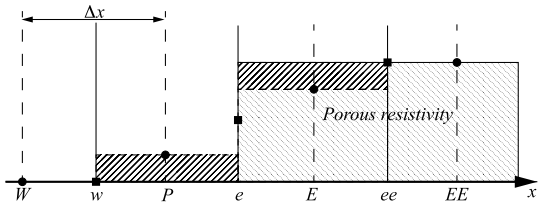

Fig. 3. Example grid showing visually the value of the porous resistance across the interface. The owner cell is denoted by P and the neighbor cell by E. Cell faces are denoted w, e and ee. Δx is the local spatial step of the grid, assumed constant for simplicity. The texturized grey area is representing the value of the porous resistance term $\phi D\langle u_i\rangle$ from Eq. (2). The solid lines mark the state of porous resistance value prior to using any additional interface treatment methods, while the dashed lines combined with right-slanted hatching pattern are porous resistance values after the application of the RDR method.

where the last two terms on the right hand side of Eq. (25) can be thought of as 'correction' terms relative to the original approximation Eq. (24). In fact, the first term on the right hand side corresponds directly to Eq. (24) while the last two terms list additional contributions. These additional contributions were written in terms of the difference between the left and right sided derivative of the pressure at the cell faces, which can be shown to reduce to zero in case of a differentiable pressure field. However, in the case of discontinuous body forces, i.e., a kink in the pressure field indicative of its non-differentiability at the interface, these correction terms are not necessarily small and alternative approaches such as RDR and FCP are essential.

The second reason for the failure of the Rhie–Chow version of PISO is what Mencinger et al. [12] refer to as *imbalance* of the cell pressure gradient and cell body force term. We will describe this *imbalance* in some more detail, thereby deriving the basis for the RDR method.

Near an interface of discontinuity we equate the pressure gradient term $\phi\partial_i p$ and the porous resistance term $\phi D\langle u_i\rangle$ from the discretized momentum equation, Eq. (7), for cell P in Fig. 3

$$\phi_P \sum_f p_f S_{i,f} = -\phi_P D_P u_{i,P} V_P \tag{26}$$

In the chosen 1D setting $V_P = \Delta x$. Dividing by ϕ_P and Δx, and carrying out the sum we arrive at

$$\frac{p_e - p_w}{\Delta x} = -D_P u_{i,P} \tag{27}$$

We use linear interpolation to obtain values of the pressure at the faces w and e.

$$\frac{p_E + p_P - p_P - p_W}{2\Delta x} = \frac{p_E - p_W}{2\Delta x} = -D_P u_{i,P} \tag{28}$$

Expanding Eq. (28) relative to the face value at face e we find

$$\frac{(p_E - p_e) + (p_e - p_W)}{2\Delta x} = -D_P u_{i,P} \tag{29}$$

which can suggestively be rewritten in terms of the weighted sum of two pressure derivatives

$$\frac{1}{2}\left[\frac{1}{2}\frac{2(p_E - p_e)}{\Delta x} + \frac{3}{2}\frac{2(p_e - p_W)}{3\Delta x}\right] = -D_P u_{i,P} \tag{30}$$

or simplified

$$\frac{1}{4}\left[\frac{(p_E - p_e)}{\frac{1}{2}\Delta x}\right] + \frac{3}{4}\left[\frac{(p_e - p_W)}{\frac{3}{2}\Delta x}\right] = -D_P u_{i,P} \tag{31}$$

The formulation is in terms of the intrinsic velocity for which we approximate $u_{i,P} = u_{i,W} = u_{i,E} = u_i$. For each of the pressure gradients in (31) we can identify a corresponding resistivity contribution. This can be expressed as

$$-\frac{1}{4}D_{Ee}u_i - \frac{3}{4}D_{We}u_i = -D_P u_i \tag{32}$$

where D_{Ee} and D_{We} are porous resistivity values associated with the regions between point E and face e and between point W and face e, respectively. In the setting as presented in Fig. 3 we observe that $D_{We} = 0$, since this region is pure fluid, while $D_{Ee} = D_E$ since this region is located within the uniform porous medium. By the same rationale $D_P = 0$. Inserting these values into Eq. (32) we notice

$$-\frac{1}{4}D_E u_i - \frac{3}{4}\cdot 0 \cdot u_i \neq 0 \cdot (-u_i) \tag{33}$$

This is what Mencinger et al. [12] refer to as *imbalance* between the pressure gradient and the body force. There is an easy remedy to this problem which consists of re-defining D_P for which we proceed as follows. Although physically the porous resistance in cell P is zero, Eq. (33) shows that a force *balance* would be maintained if we would assign D_P a new value $D_P^{rd} = \frac{1}{4} D_E$. With a similar argument applied to the cell centered around E it can be concluded that the value of D_E should be modified to $D_E^{rd} = \frac{3}{4} D_E$.

This discretization results in a *redistribution* of porous resistance between cell E and cell P, which is why the method is called Re-Distributed Resistivity (RDR). A graphical depiction of porous resistance values associated with the utilization of the RDR method is denoted by dashed lines and right-slanted hatching in Fig. 3.

Apart from the porous resistance redistribution, where linear interpolation of the pressure from cell centers to faces is applied and discontinuities are handled by redistributing the resistivity, an alternative 'accounting' of discontinuities in the body force term can be achieved by assigning alternative values to the pressure at the interface p_e. This has direct influence on the pressure gradients in cells P and E. Deriving such an auxiliary value for p_e is the essence of the FCP method, which is obtained closely following Oxtoby et al. [11]. Technical details of the implementation of both RDR and FCP methods will be covered in detail in Sections 4 and 5.

4. Re-distributed resistivity algorithm

Due to the presence of the body force discontinuity, represented by the porous resistance term $\phi D \langle u_i \rangle$ in Eq. (2), the pressure at the fluid–porous interface based on the standard central discretization is such that the velocity solution adjusts with velocity oscillations. These oscillations have been explored before in [12,10]. In order to restore the balance between the pressure gradient and the body force, we propose to adapt the resistivity in the cells adjacent to the interface. This is the basic principle behind the novel Re-Distributed Resistivity (RDR) algorithm [9] which will be presented below.

4.1. Flow resistivity redistribution

To avoid the generation of spurious velocity oscillations in the vicinity of the discontinuity, when solving the discretized momentum equation, Eq. (20) or correcting the velocity with Eq. (21), a balance between the discontinuous porous resistance field and the cell-centered pressure gradient is required [12]. Because of the body force discontinuity at the fluid–porous interface, the interface pressure value p_f is erroneously determined by linear interpolation using Eq. (14) as was shown in Section 3.3. These pressure gradients in the two cells neighboring the discontinuity are no longer in balance with the cell-centered flow resistance in these cells. Because of this, the cell-centered resistances are modified to balance the cell-centered pressure gradients calculated from the linearly interpolated face pressures, as shown in Section 3.3.

In order to determine the cells requiring redistribution of porous resistivity, a cell indicator function is defined as:

$$\Omega_P = \frac{\sum_f |(\Delta\phi)_f|}{\max(\sum_f |(\Delta\phi)_f|, \xi)} \tag{34}$$

where $(\Delta\phi)_f = \phi_{nb} - \phi_P$ and ξ is a small number on the order of 10^{-15} to avoid division by zero if $\sum_f |(\Delta\phi)_f| = 0$. Ω_P has the value 1 for cells requiring redistribution of resistivity and 0 otherwise. Another indicator function θ_f is defined as:

$$\theta_f = \frac{|(\Delta(\Omega\phi))_f|}{\max(|(\Delta(\Omega\phi))_f|, \xi)} \tag{35}$$

where $\Delta(\Omega\phi)_f = \Omega_{nb}\phi_{nb} - \Omega_P\phi_P$, in order to determine the faces required for the calculation of the redistributed flow resistivity in the cells where $\Omega_P = 1$. θ_f has the value 1 for the fluid–porous interface faces and the faces of the cells adjacent to a discontinuity and 0 otherwise. The redistributed resistivity D_P^{rd} is computed using inverse distance weighting for the required faces according to:

$$D_P^{rd} = (1 - \Omega_P) D_P + \frac{\Omega_P}{\sum_f \omega_f \theta_f} \sum_f \omega_f \theta_f D_f \tag{36}$$

where $\omega_f = 1/|d_{i,Pf}|$ and D_f is interpolated to the face centers by the linear interpolation in (14). The graphical depiction of this porous resistance redistribution process is shown in Fig. 4. The redistributed resistivity (36) is based on a 'double' interpolation in cells with discontinuities, i.e., first linear interpolation is used to estimate the face values D_f, after which inverse distance weighting is used to complete the re-distribution. This approach is similar to that proposed by Raeini et al. (2012) [23] for smoothing the interface representation in a volume-of-fluid method, by using 'double interpolation' and Gaussian smoothing to define the new interface normals.

The modified discretized momentum equation with the redistributed resistivity yields:

$$\frac{1}{\phi_P}\partial_t(\rho_P u_{i,P}) + \frac{1}{\phi_P^2 V_P}\sum_f (\rho u_j)_f u_{i,f} S_{j,f} = -\frac{1}{V_P}\sum_f p_f S_{i,f} + \frac{1}{\phi_P V_P}\sum_f \tau_{ij,f} S_{j,f} + \zeta_{i,P} - D_P^{rd} u_{i,P} \tag{37}$$

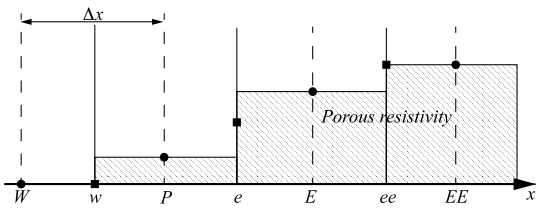

Fig. 4. Graphical depiction showing the state of variables after the redistribution of the porous resistance has been carried out. The new state of the porous resistance field compensates for over or underestimates of the pressure gradients in the cells adjacent to the interface and force *balance* is restored, retaining the correct velocity field.

and the expression for the velocity correction reads:

$$u_{i,P} = B_P^{rd} A_P^{-1} (H_{i,P} - (\partial_i p)_P) \tag{38}$$

where $B_P^{rd} = (1 + D_P^{rd} A_P^{-1})^{-1}$. The face velocity $u_{i,f}$ in the convective term can be discretized by any suitable scheme (e.g. linear) and the mass flux $(\rho u_i)_f S_{i,f}$ is calculated from Eq. (22). We emphasize that relative to the first Rhie–Chow modification Eq. (21), the RDR approach can be interpreted by a change in B_P only through a redistribution of the resistivity D in the direct vicinity of the fluid–porous interface or porous discontinuity.

4.2. Re-distributed resistivity algorithm

The original PISO algorithm [24] is a stepwise technique for the solution of the implicitly discretized time-dependent flow equations in a segregated manner. The solution is accomplished at each time step through a sequential predictor–corrector process by which the different dependent variables are updated individually. Following the notation in [24], variables with asterisks are intermediate step solutions. The RDR algorithm scheme is shown in Fig. 5.

The RDR algorithm differs from the original PISO algorithm [24] in that *(i)* the equations are differently formulated, *(ii)* the modified Rhie–Chow interpolation is used and *(iii)* an extra step prior to the predictor step is introduced, where we execute the porous resistance redistribution in accordance with Eq. (36). No significant computational cost was noticed because of this extra step. Subsequently, we continue the use of the B_P^{rd} matrix in the first and second corrector steps in order to keep the method consistent. The original PISO algorithm [24] was intended to have only a single pass through the predictor and two corrector steps. In order to improve the precision of the algorithm and to control conservation errors coming from the segregated approach, we have added a convergence condition at the end of the algorithm as proposed by [25]. This condition is based on the residual value obtained by inserting the latest pressure solution p^{**} into the pressure equation of the first corrector step. Once the residual value goes down below a pre-set value, e.g. 10^{-12}, the solution is considered as converged. The conservation errors can therefore be controlled to an arbitrarily precision by specifying the pre-set value.

In the next section we will present the FCP method, which achieves similar results as RDR by modifying the value of the interface pressure instead of the resistivity.

5. Face Consistent Pressure PISO algorithm

In the previous section we presented the RDR method which aims to establish a balance between the pressure gradient and the body force term by redistributing the body force term between the cells adjacent to the interface. Considering Eq. (21) an alternative approach presents itself: if the pressure at the interface would be adapted, the balance between the pressure gradient and the body force terms could be achieved as well. This approach was previously explored by [7] and [11], where special face pressure interpolation schemes were introduced to accurately estimate the interface pressure. Since such interface pressure values are derived to account for the influence of the local jump in the body force across the interface, the pressure at the interface is referred to as *consistent* with the resistivity jump. Schemes in [7] and [11] require either explicit corrections based on the mass flux or extrapolations of the cell-centered pressure from either side of the interface.

In this section we present the Face Consistent Pressure (FCP) algorithm, which utilizes auxiliary pressure values at the interface in order to prevent spurious velocity oscillations. The method builds on the work of Oxtoby et al. [11], but is different in three key ways: FCP solves for extrinsic velocity field, FCP treats all terms, except pressure gradient, implicitly, and FCP incorporates the modified Rhie–Chow interpolation, Eq. (21), presented in Section 3.2. The FCP method calculates the interface pressure value which is consistent with the adjacent flow resistance. This results in a pressure field that by default satisfies the balance between the discontinuous flow resistance field and the cell-centered pressure gradient [12] in cells adjacent to the interface at steady-state conditions. The derivation of the equations and the functioning of the modified algorithm are presented below.

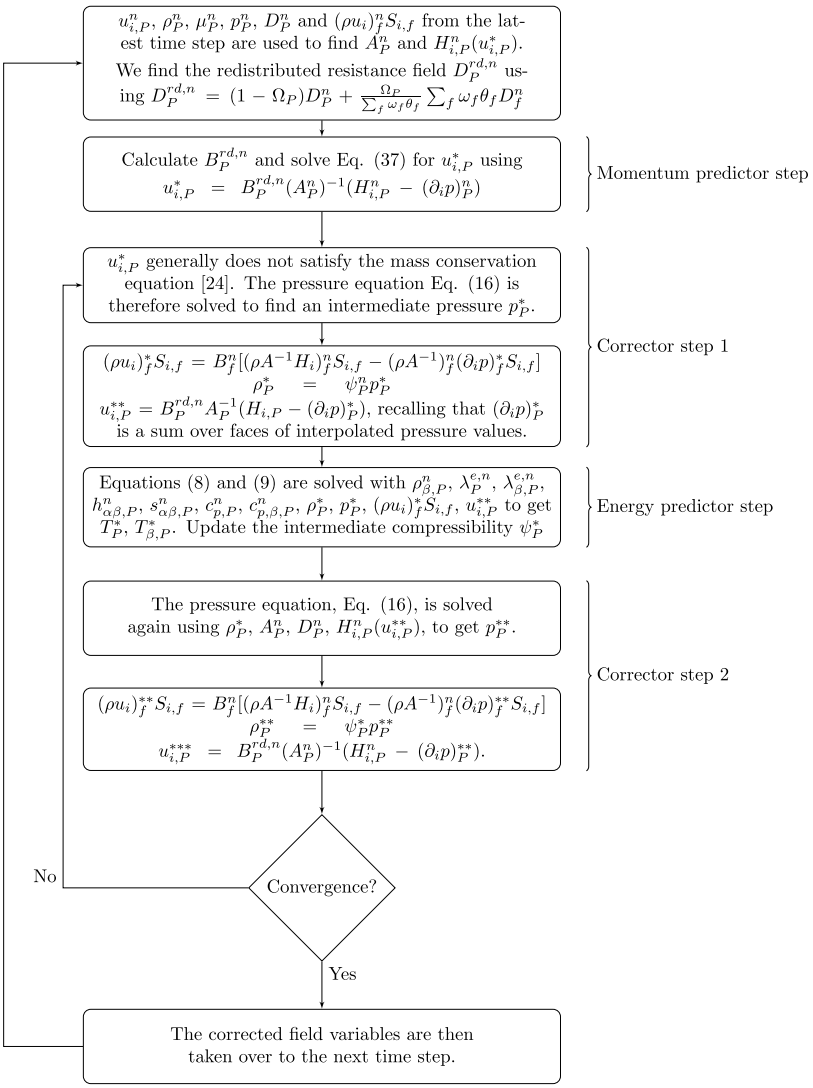

Fig. 5. RDR algorithm. The major difference with respect to the original PISO [24] is that the time step begins with redistributing the porous resistance values. The convergence condition is based on the residual value obtained by inserting the latest pressure solution p^{**} into the pressure equation of the first corrector step [25].

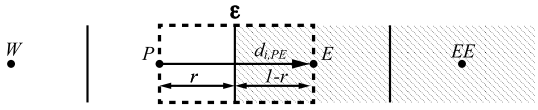

Fig. 6. Control volume around the interface (dashed line) which is used for derivation of the pressure interpolation at the interface. The major difference with respect to the generic cell representation of Fig. 1 is that we now concentrate on the immediate vicinity of the fluid–porous interface. To differentiate the interface faces from the other faces denoted with f, we denote the interface values with index ε. The control volume used for derivation of auxiliary interface pressure value is marked with dashed lines. The vector connecting the two cell centers is denoted $d_{i,PE}$ and the distance fractions r and $(1-r)$ are also marked.

5.1. Derivation of the consistent interface pressure

Let us analyze the momentum balance of a fictional control volume around the interface as shown in Fig. 6. Cell center P is in the pure fluid domain and cell center E is in the porous domain. In order to differentiate between the specially derived interface pressure and the other face values (calculated by linear interpolation), we denote the interface values with the index ε. The interface ε is positioned between points P and E at distance fraction r away from point P and $(1-r)$ away from point E. The distance between the two points is denoted by $|d_{i,PE}|$.

Following the combined notation of Fig. 6 and Section 3, we first write out the mass conservation expressions for each half of the control volume:

$$(\partial_t(\phi\rho))_P + \frac{\hat{n}_i(\rho u_i)_\varepsilon - \hat{n}_i(\rho u_i)_P}{r|d_{i,PE}|} = 0 \tag{39}$$

and

$$(\partial_t(\phi\rho))_E + \frac{\hat{n}_i(\rho u_i)_E - \hat{n}_i(\rho u_i)_\varepsilon}{(1-r)|d_{i,PE}|} = 0 \tag{40}$$

By combining Eqs. (39) and (40) we get:

$$r|d_{i,PE}|(\partial_t(\phi\rho))_P + (1-r)|d_{i,PE}|(\partial_t(\phi\rho))_E = \hat{n}_i(\rho u_i)_P - \hat{n}_i(\rho u_i)_E \tag{41}$$

The momentum fluxes for the left and right half of the fictional control volume can be expressed on the basis of Eq. (21)

$$(\rho u_i)_P = (B\rho A^{-1}H_i)_P - (B\rho A^{-1})_P \frac{p_\varepsilon - p_P}{r|d_{i,PE}|}\hat{n}_i \tag{42}$$

and

$$(\rho u_i)_E = (B\rho A^{-1}H_i)_E - (B\rho A^{-1})_E \frac{p_E - p_\varepsilon}{(1-r)|d_{i,PE}|}\hat{n}_i \tag{43}$$

where the pressure gradient is approximated using the auxiliary pressure value p_ε. Inserting Eqs. (42) and (43) into Eq. (41), we can solve for the consistent pressure at the interface:

$$p_\varepsilon = (B\rho A^{-1})_\varepsilon^{-1}[(B\rho A^{-1}p)_\varepsilon - r(1-r)|d_{i,PE}|^2((\partial_i(B\rho A^{-1}H_i))_\varepsilon + (\partial_t(\phi\rho))_\varepsilon)] \tag{44}$$

with interpolated terms being:

$$(B\rho A^{-1})_\varepsilon = (1-r)(B\rho A^{-1})_P + r(B\rho A^{-1})_E \tag{45}$$

$$(\partial_i(B\rho A^{-1}H_i))_\varepsilon = |d_{i,PE}|^{-1}(\hat{n}_i(B\rho A^{-1}H_i)_P - \hat{n}_i(B\rho A^{-1}H_i)_E) \tag{46}$$

$$(\partial_t(\phi\rho))_\varepsilon = r(\partial_t(\phi\rho))_P + (1-r)(\partial_t(\phi\rho))_E \tag{47}$$

and

$$(B\rho A^{-1}p)_\varepsilon = (1-r)(B\rho A^{-1}p)_P + r(B\rho A^{-1}p)_E \tag{48}$$

We refer to p_ε as consistent pressure because its value is taking into consideration the local jump in porous resistance through the coefficient matrix B.

In order to obtain the expression for the final pressure field, we utilize the same masking function θ_f as defined in Section 4.1 to identify the porous fluid interface. The expression of the new face pressure field denoted Π_f then becomes:

$$\Pi_f = (1-\theta_f)p_f + \theta_f p_\varepsilon \tag{49}$$

Note that because of the use of the masking function, the newly devised face consistent pressure correction (second term in Eq. (49)) will be applied only at the interface. The other face values of pressure (p_f) in the domain will be calculated according to the interpolation scheme of choice (e.g. linear, upwind etc.). This can be altered so that the face consistent pressure interpolation would be applied throughout the domain $\Pi_f = p_\varepsilon$, regardless of the interface presence. We refer to this approach as the Global-FCP method, since the interpolation of Eq. (44) is applied to all the faces in the domain. We do not pursue this approach in this paper and leave its analysis for future work.

By using the newly devised pressure Π_f we can now form a pressure gradient term in the momentum equation (Eq. (18)) to get:

$$\frac{1}{\phi_P}\partial_t(\rho_P u_{i,P}) + \frac{1}{\phi_P V_P}\sum_f(\phi^{-1}\rho u_j)_f u_{i,f} S_{j,f} = -\sum_f \Pi_f S_{i,f} + \frac{1}{\phi_P V_P}\sum_f \tau_{ij,f} S_{j,f} + \zeta_{i,P} - D_P u_{i,P} \tag{50}$$

By creating this consistent value of face pressure Π_f at the interface, we aim at avoiding the force imbalance in the cells adjacent to the interface.

A correction of the pressure value at the interface has been derived by analyzing the momentum balance in the fictitious control volume around the interface. This corrected pressure incorporates the influence of the porous resistance values adjacent to the fluid–porous interface. Therefore, the pressure gradients at points P and E satisfy the force balance between the discontinuous flow resistance field and the cell-centered pressure gradient [12]. This force balance aims to remove the spurious oscillations in the velocity field.

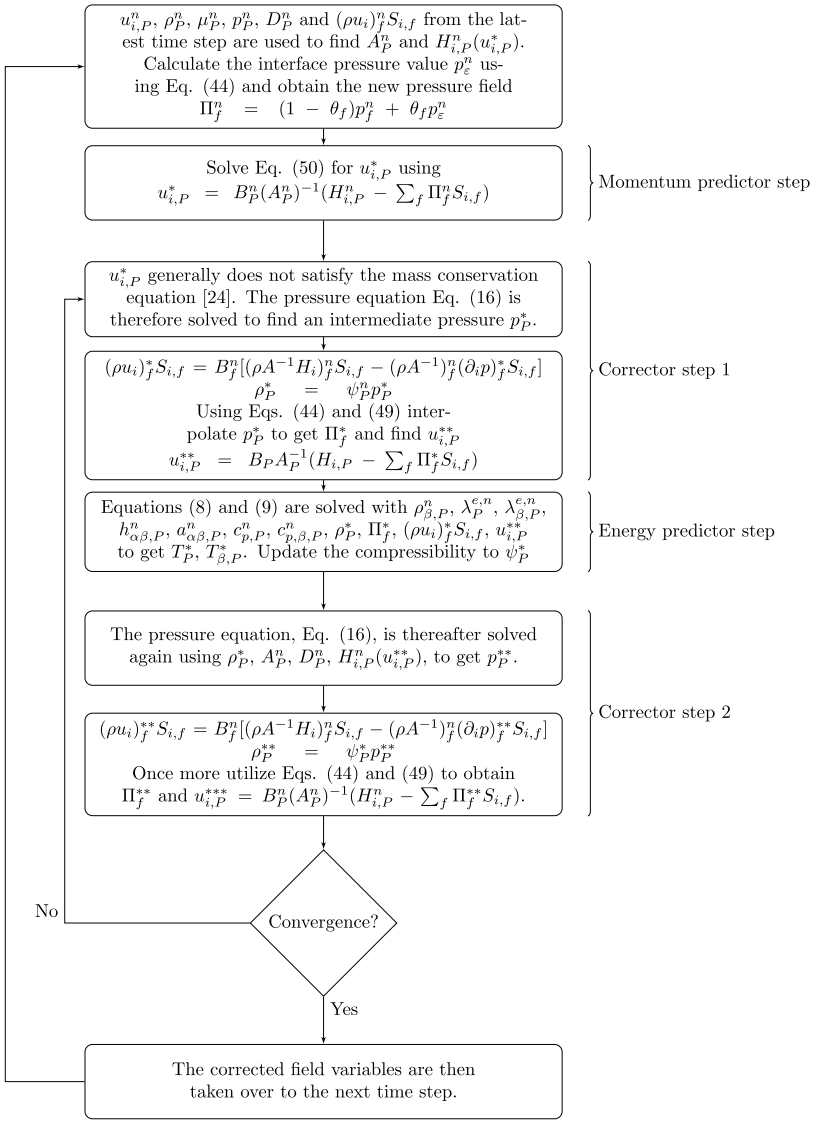

Fig. 7. FCP algorithm. The difference with respect to the RDR algorithm is that instead of redistributing porous resistivity, the face pressure field p_f is updated to Π_f using Eq. (49) throughout the algorithm.

5.2. Face Consistent Pressure algorithm

The FCP algorithm is very similar to the original PISO algorithm proposed by Issa et al. [24]. The main differences are *(i)* the differently formulated equations, *(ii)* the modified Rhie–Chow interpolation and *(iii)* the use of a modified pressure (Eq. (49)) in the momentum predictor and corrector steps throughout the algorithm. Following [24], the FCP algorithm takes the form shown in Fig. 7.

The FCP algorithm uses the same steps as the RDR algorithm, except that instead of updating B_P to B^{rd}_P in the predictor and corrector steps, we update the face pressure field p_f to Π_f. The convergence condition at the end is the same as for the RDR algorithm to control conservation errors coming from the segregated approach.

The proposed RDR, FCP and the original PISO algorithms described in Sections 4, 5 and 3, respectively, have been implemented using the OpenFOAM® open source computational fluid dynamics C++ library (version 2.2.0).

6. Porous plug and Beavers–Joseph test cases

The case for flow perpendicular to a porous region [26] and the Beavers–Joseph problem [16] for flow parallel to a porous region are considered, in order to demonstrate the accuracy and robustness of the proposed algorithms. The respective geometries of these test cases are shown in Fig. 8. In these illustrations we assume incompressible and isothermal flow and

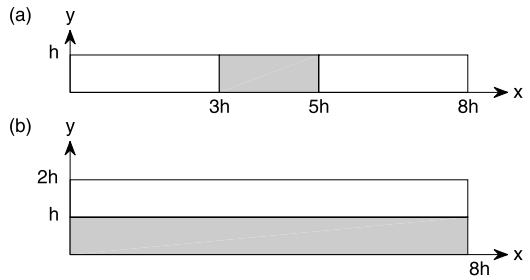

Fig. 8. Geometries for the (a) porous plug and (b) Beavers–Joseph cases.

an isotropic porous medium even though the algorithms are implemented in a compressible formulation. Exploration of compressibility and heat transfer are considered in future work.

The first case is set in a 2D channel, of height h and length $8h$, in which an isotropic porous plug has been installed between $3h$ and $5h$ [26]. The inlet velocity boundary condition is a fully developed parabolic profile, with mean velocity U. The Reynolds number takes the values $Re = \{1, 100, 1000\}$ and is defined as $Re = \frac{\rho U h}{\mu}$. The Darcy number of the flow is defined as $Da = K/h^2$, where K is the porous medium permeability, and for testing purposes we chose $Da = \{10^{-2}, 10^{-3}, 10^{-7}\}$. These values of Re and Da numbers correspond to low and moderate amplitudes of the porous resistance terms. The porosity of the porous plug is kept constant at $\phi = 0.7$. The outlet boundary condition for velocity is zero gradient or $\hat{n}_j \partial_j u_i = 0_i$ where $\hat{n}_j$ is the normal vector to the outlet and 0_i is a zero vector. The pressure value is specified as constant at the outlet, while at the inlet we require the zero gradient value of pressure or $\hat{n}_i \partial_i p = 0$. For channel walls, no-slip boundary conditions ($u_i = 0$) were applied. The grid is uniform, structured and its resolution is equivalent to the grid resolution used in [7] for better comparison. The geometry is presented in Fig. 8(a).

The second case we consider was originally proposed by Beavers and Joseph [16], and is commonly used in the literature as a validation case. We adopt a slightly modified version following [26] and [7]. The domain is a 2D channel of height $2h$ and length $8h$. Half of the channel height h is filled with isotropic porous material. For the inlet velocity boundary condition a uniform velocity of magnitude U is adopted at $Re = 1$ so that the flow has time to fully develop before reaching the end of the domain. The tested Darcy numbers are $Da = \{10^{-2}, 10^{-3}\}$, and the porosity is kept constant for all cases $\phi = 0.7$. The outlet boundary condition for velocity is zero gradient or $\hat{n}_j \partial_j u_i = 0_i$ where $\hat{n}_j$ is the normal vector to the outlet. The pressure value is specified as constant at the outlet, while at the inlet we require the zero gradient value of pressure or $\hat{n}_i \partial_i p = 0$. The choice of the grid resolution was motivated by the work of [7] for better comparison. Walls of the channels have an imposed no-slip boundary condition for the velocity or $u_i = 0$. The geometry is shown in Fig. 8(b).

The time derivative is discretized by a second-order implicit backward differencing scheme and the convective term by the second-order linear upwind differencing (LUD) scheme [27]. The discretized equations for each coordinate direction are solved using standard solvers and utilities available in OpenFOAM® such as: the smooth solver with a Gauss–Seidel smoother down to a tolerance 10^{-11} for velocity components, and the pressure equations are solved with a Preconditioned Conjugate Gradient (PCG) solver adopting the Faster Diagonal Incomplete-Cholesky (FDIC) preconditioner down to a tolerance of 10^{-12}.

The proposed RDR and FCP methods differ from a standard central discretization wherever there are variations or discontinuities in the porosity and consequently in the resistivity term. Central discretization methods have already proven their suitability to simulate flow in porous media with smoothly varying porosity [28]. Therefore, we restrict here to discontinuities in the porosity near fluid–porous interfaces, where spurious oscillations may arise. The two test cases considered are designed to address these challenges close to fluid–porous interfaces, where discontinuities are most pronounced. The proposed algorithms are however also suitable for general heterogeneous fluid–porous media, e.g., where the porosity and resistivity are smooth functions of the position in the domain. We validate the new method for segregated solvers against numerical results from literature based on fully block-coupled numerical methods [8]. Validation is done for 2D geometries, on structured Cartesian grids, because no standard validation case based on 3D geometries are available. The robustness and applicability of the algorithms for complex 3D geometries and general unstructured meshes will be pursued further in future research.

6.1. Porous plug case

In Fig. 9, the velocity magnitude along the centerline of the porous plug geometry located at height $0.5h$ is presented. Since the velocity inlet is specified as fully developed parabolic profile with mean superficial velocity U, the peak value at the centerline is $1.5U$. Once the flow reaches the porous plug, we register a dip in the centerline velocity. This may appear counter-intuitive, as the porous plug constricts the flow and acceleration is expected. However, what happens is that the velocity profile in the y-direction changes once the fluid enters the porous plug, going from parabolic to a flat profile, thereby maintaining conservation of mass even though the centerline velocity is much reduced. After the fluid exits the porous plug, the velocity profile in y-direction is seen to rapidly recover back to the Poiseuille profile.

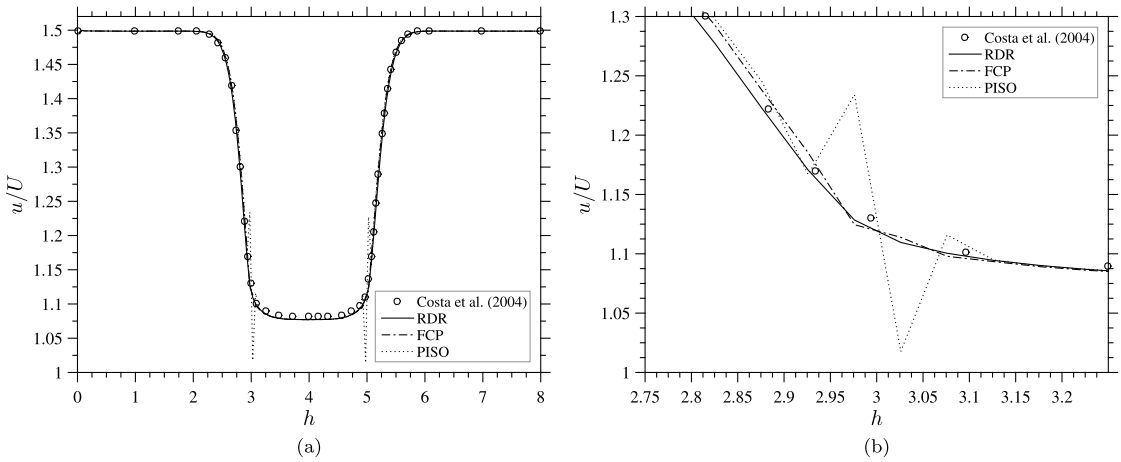

Fig. 9. Centerline velocity for porous plug problem for $Re = 1$ and $Da = 10^{-3}$.

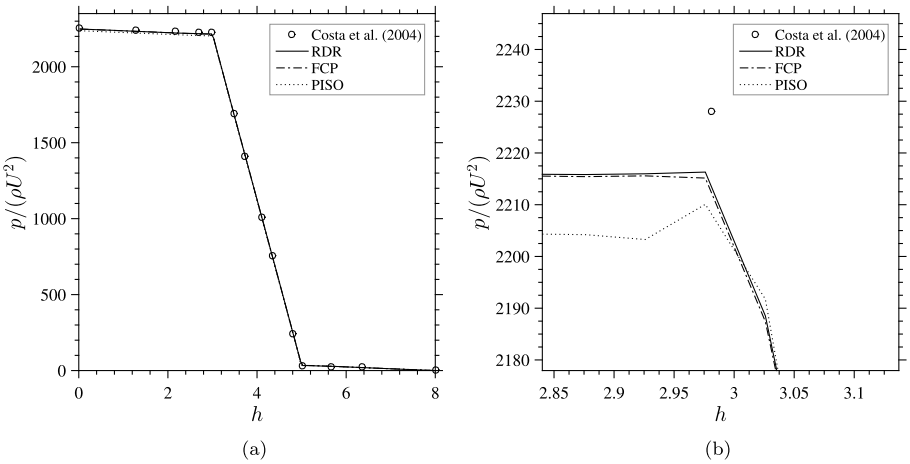

Fig. 10. Figure (a) showing centerline pressure for the porous plug problem at $Re = 1$ and $Da = 10^{-3}$. Figure (b) is zoomed in around the interface, where the pressure gradient is discontinuous.

It can be seen in Fig. 9(a) that both the RDR and FCP algorithms provide an oscillation-free solution for the reference case ($Re = 1$, $Da = 10^{-3}$), while the solution obtained by the original PISO algorithm exhibits substantial oscillations. This is emphasized in Fig. 9(b), where we see a zoomed-in area around the interface. We also see that both the RDR and FCP algorithms are in good agreement with the reference data [26]. At higher Re and/or lower Da the spurious oscillations were found to become even more pronounced.

In Fig. 10(a) we present the pressure field along the centerline of the geometry, corresponding to the velocity solution shown in Fig. 9. The pressure gradient in the fluid area upstream of the porous plug is low as the flow does not require a high pressure gradient to maintain the required mass flow rate. Once the fluid reaches the porous plug at $3h$, we see a discontinuity in the pressure gradient and a much stronger forcing of the flow, i.e. a much steeper negative pressure gradient, is needed in the porous region to keep the mass-flow rate constant. The pressure gradient rapidly recovers its value that was also observed ahead of the plug after exiting the porous plug at $5h$.

All solutions seem to match the reference data of [26]. When we zoom around the interface, Fig. 10(b), we see that the original PISO algorithm is producing small pressure variations, which are linked to the velocity oscillations. We also see that RDR and FCP slightly underestimate the value of the pressure with respect to the reference case, but show no unphysical oscillations. The slight underestimation by about 0.5% is a measure for the remaining discretization error at the spatial resolution that was adopted.

Next, we compare the RDR and FCP algorithms for different flow and porosity conditions and compare velocity solutions for four different cases at $Re = \{100, 1000\}$ and $Da = \{10^{-3}, 10^{-7}\}$. Fig. 11 shows centerline velocity solutions of the RDR and FCP methods for these four cases. These solutions look identical, with the only noticeable difference being a slightly smoother velocity profile of the FCP solution at the porous plug inlet ($h = 3$) at $Da = 10^{-7}$. The original PISO algorithm failed at producing accurate results for these demanding conditions. The discrepancies in Fig. 11 become obvious once we concentrate on the area around the interface, which are shown in Figs. 11(c) and 11(d).

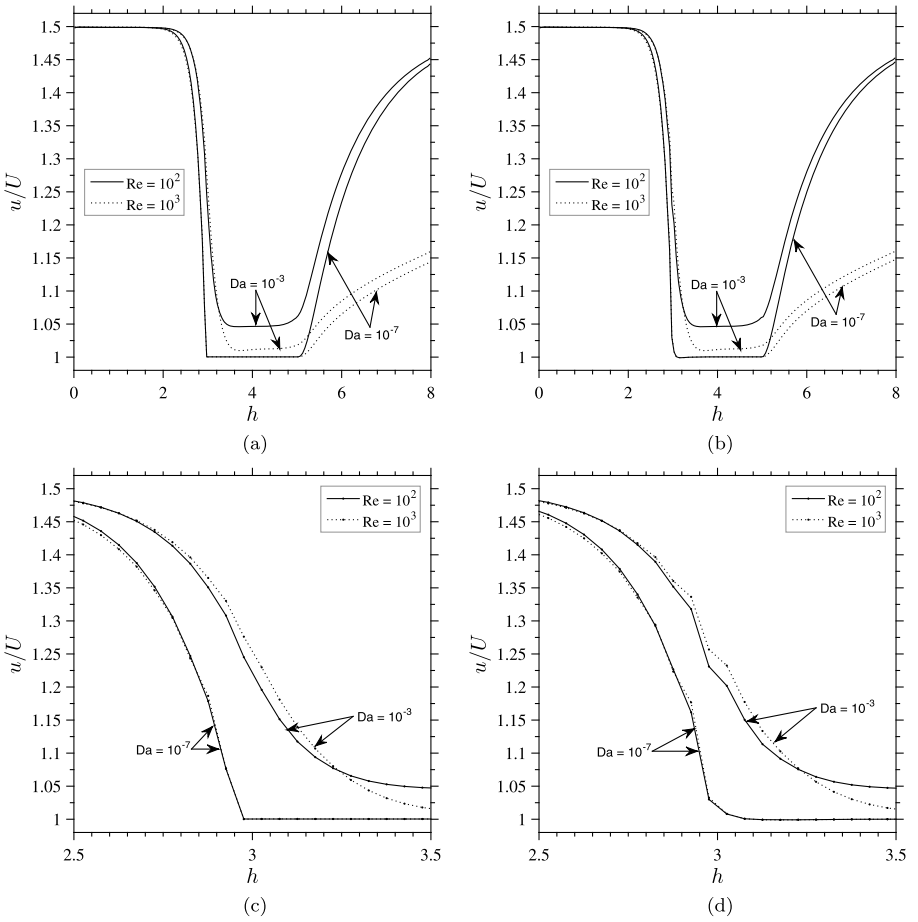

Fig. 11. Centerline velocities for the porous plug problem at $Re = \{100, 1000\}$ and $Da = \{10^{-3}, 10^{-7}\}$. The Figures (a) and (c) show the RDR solutions, while the FCP solutions are presented in the Figures (b) and (d). Notice the small-amplitude oscillations in the FCP solutions in Figure (d), particularly for cases with $Da = 10^{-3}$.

From Figs. 11(c) and 11(d) it can be immediately concluded that RDR produces smoother solutions.

The convergence of both algorithms has been tested to confirm correct implementation of the methods and to quantify the truncation error coming from spatial discretization. Fig. 12(a) shows the l^2 norm plot of both the RDR and FCP solutions for the porous plug case. The reference resolution case had a four times higher resolution than the highest plotted point in Fig. 12(a). Fig. 12(b) qualitatively demonstrates the convergence, while also showing good agreement with the reference data of Costa et al. [26]. The convergence rate for both algorithms is roughly of the first order.

As a remark, we mention that in addition to the original PISO, RDR and FCP algorithms, we also investigated the possibility of using only a modified Rhie–Chow PISO algorithm, without the RDR or FCP extensions. This version of the code produced smooth velocity solution for the $Re = 1$ and $Da = 10^{-3}$ case, but at more aggressive parameters, e.g. $Re > 100$, the spurious oscillations occurred. This confirms that making Rhie–Chow interpolation consistent is not enough to remove the spurious oscillations in general, as suggested in Section 3.2.

We used a porous plug case [26] to compare the performance of the newly developed RDR and FCP algorithms. We find that for low *Re*/high *Da* numbers, both RDR and FCP algorithms give satisfactory solutions. Once pushed to higher *Re*/lower *Da* numbers, the RDR algorithm maintains the smoothness of the velocity solution, while the FCP solutions experience some oscillations around the interface.

6.2. Beavers–Joseph case

For the Beavers–Joseph case [26,7,16] we present the results at $Re = 1$ and $Da = \{10^{-2}, 10^{-3}\}$ for which a comparison with a reference solution from literature is shown in Fig. 13.

All algorithms are seen to perform equally well (nearly overlapping), compared to the reference data [26]. As can be seen, the original PISO algorithm also provides oscillation free results. This is because the pressure gradient over the porous interface is close to zero, leading to a negligible flow in the y-direction, i.e., normal to the interface between the fluid and the porous medium. As a direct consequence, the spurious oscillations are not to be expected. Therefore we limit ourselves to present results only for a few selected parameters.

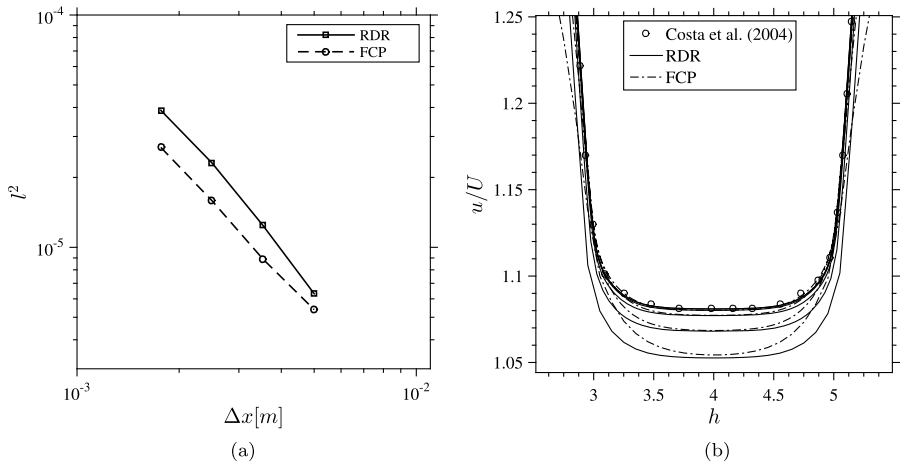

Fig. 12. Figure (a) showing the l^2 norm of the difference between the numerical solution computed at Δx and a fine-scale reference solution. The four points in Figure (a) are from meshes of 800, 3200, 12 800 and 51 200 cells. The reference case mesh contained 204 800 cells. Figure (b) is qualitatively showing convergence, while demonstrating good agreement with the literature data of Costa et al. [26].

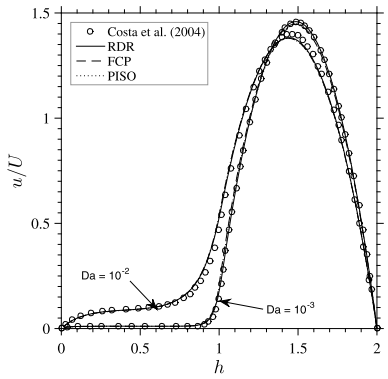

Fig. 13. Fully developed velocity profiles at the end of the domain for the Beavers–Joseph problem ($Re = 1$ for $Da = 10^{-2}$ and $Da = 10^{-3}$).

7. Conclusions

In this paper we presented the governing equations for volume-averaged weakly density varying flow through porous media, with non-equilibrium heat transfer [6,8,18]. As a point of reference, the original PISO algorithm [24] for segregated solvers on collocated grids was presented, whose key element is the utilization of the Rhie–Chow interpolation [21] and a predictor–corrector scheme. By using the Rhie–Chow interpolation, the original PISO algorithm for collocated grids maintains the pressure–velocity coupling and avoids 'checker-board' patterns in the pressure and velocity fields when the flow fields are smooth. However, the standard Rhie–Chow interpolation is argued to fail when large body force discontinuities are present in the domain [12,10], resulting in spurious oscillations in the velocity solution. In conjugate fluid–porous domains these jumps in body forces correspond to jumps in the porous resistance and porosity across the fluid–porous interface. In Section 3.3 we elaborated the reasons behind this failure and in Section 3.2 a modification of the Rhie–Chow method was introduced that takes the resistivity jumps explicitly into account. The incorporation of resistivity jumps in the Rhie–Chow interpolation was found to be insufficient to remove the velocity oscillations. Therefore, two modified algorithms have been proposed: the RDR and the FCP algorithms. Sections 4 and 5 are dealing with the numerical details of the proposed algorithms. While the RDR method addresses the problem through a redistribution of resistivity in cells adjacent to the interface, while using linear interpolation of the pressure to the faces, the FCP method finds an alternative pressure estimate at the cell interface to counteract the problem. The obtained results from the algorithms were compared to literature data for incompressible and isothermal flow parallel and perpendicular to a porous region and were found to agree well with the reference velocity and pressure data for all cases. The results were also compared to data generated by the original PISO algorithm [24], which was shown to generate undesired pressure–velocity decoupling with resulting spurious oscillations in the velocity in the vicinity of the discontinuity, as expected. Both the RDR and FCP algorithms generate smooth, non-oscillatory results for cases with moderately low *Re*/high *Da* numbers, while at higher *Re*/lower *Da* numbers the RDR algorithm produces smooth and physically more convincing solutions than the FCP method.

Ongoing work focuses on further exploration of operational parameters for both algorithms on both structured and unstructured meshes, as well as an assessment of the compressible capability, including non-equilibrium heat transfer.

Acknowledgement

The research described in this paper was funded by Philip Morris Products S.A., Switzerland.

References

[1] Y. Peng, J. Richardson, Properties of ceramic foam catalyst supports: one-dimensional and two-dimensional heat transfer correlations, Appl. Catal. A, Gen. 266 (2004) 235–244.
[2] N. Yucel, R.T. Guven, Forced-convection cooling enhancement of heated elements in a parallel-plate channels using porous inserts, Numer. Heat Transf., Part A, Appl. 51 (2007) 293–312.
[3] A.-R. Khaled, K. Vafai, The role of porous media in modeling flow and heat transfer in biological tissues, Int. J. Heat Mass Transf. 46 (2003) 4989–5003.
[4] C.T. Simmons, T.R. Fenstemaker, J.M. Sharp Jr., Variable-density groundwater flow and solute transport in heterogeneous porous media: approaches, resolutions and future challenges, in: Practical Applications of Coupled Process Models in Subsurface Environments, J. Contam. Hydrol. 52 (2001) 245–275.
[5] F.J. Molz, M.A. Widdowson, L.D. Benefield, Simulation of microbial growth dynamics coupled to nutrient and oxygen transport in porous media, Water Resour. Res. 22 (1986) 1207–1216.
[6] S. Whitaker, The Method of Volume Averaging, Kluwer Academic Publishers, 1999.
[7] L. Betchen, A.G. Straatman, B.E. Thompson, A nonequilibrium finite-volume model for conjugate fluid/porous/solid domains, Numer. Heat Transf., Part A, Appl. 49 (2006) 543–565.
[8] C.T. DeGroot, A.G. Straatman, A finite-volume model for fluid flow and nonequilibrium heat transfer in conjugate fluid–porous domains using general unstructured grids, Numer. Heat Transf., Part B, Fundam. 60 (2011) 252–277.
[9] M. Nordlund, A. Kuczaj, M. Stanic, E. Frederix, B. Geurts, Modified Rhie–Chow/PISO algorithm for collocated variable finite porous media flow solvers, in: 5th International Conference on Porous Media and Their Applications in Science, Engineering and Industry, in: ECI Symposium Series, vol. 40, 2014.
[10] C. Gu, Computation of flows with large body forces, in: J.C.E.C. Taylor (Ed.), Numerical Methods in Laminar and Turbulent Flow, Pineridge Press, Swansea, 1991, pp. 294–305.
[11] O. Oxtoby, J. Heyns, R. Suliman, A finite-volume solver for two-fluid flow in heterogeneous porous media based on OpenFOAM, in: Proceedings of Open Source CFD International Conference, 2013.
[12] J. Mencinger, I. Žun, On the finite volume discretization of discontinuous body force field on collocated grid: application to VOF method, J. Comput. Phys. 221 (2007) 524–538.
[13] B. Muller, Computation of Compressible Low Mach Number Flow, 1996.
[14] J.B. Bell, A.J. Aspden, M.S. Day, M.J. Lijewski, Numerical simulation of low Mach number reacting flows, J. Phys. Conf. Ser. 78 (2007) 012004.
[15] H. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education Limited, 2007, http://books.google.nl/books?id=RvBZ-UMpGzIC.
[16] G.S. Beavers, D.D. Joseph, Boundary conditions at a naturally permeable wall, J. Fluid Mech. 30 (1967) 197–207.
[17] C.T. DeGroot, A.G. Straatman, Numerical results for the effective flow and thermal properties of idealized graphite foam, J. Heat Transf. 134 (2012), 042603.
[18] K. Vafai, C. Tien, Boundary and inertia effects on flow and heat transfer in porous media, Int. J. Heat Mass Transf. 24 (1981) 195–203.
[19] P. Yu, T.S. Lee, Y. Zeng, H.T. Low, A numerical method for flows in porous and homogenous fluid domains coupled at the interface by stress jump, Int. J. Numer. Methods Fluids 53 (2007) 1755–1775.
[20] R. Issa, B. Ahmadi-Befrui, K. Beshay, A. Gosman, Solution of the implicitly discretised reacting flow equations by operator-splitting, J. Comput. Phys. 93 (1991) 388–410.
[21] C. Rhie, W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[22] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows, PhD thesis, Imperial College, University of London, 1996.
[23] A. Raeini, M. Blunt, B. Bijeljic, Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method, J. Comput. Phys. 231 (2012) 5653–5668.
[24] R. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1986) 40–65.
[25] E. Frederix, M. Stanic, A. Kuczaj, M. Nordlund, B. Geurts, Extension of the compressible {PISO} algorithm to single-species aerosol formation and transport, Int. J. Multiph. Flow 74 (2015) 184–194.
[26] V.A.F. Costa, L.A. Oliveira, B.R. Baliga, A.C.M. Sousa, Simulation of coupled flows in adjacent porous and open domains using a control-volume finite-element method, Numer. Heat Transf., Part A, Appl. 45 (2004) 675–697.
[27] R. Warming, R. Beam, Upwind second-order difference schemes and applications in aerodynamic flows, AIAA J. 14 (1976) 1241–1249.
[28] R. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.