CrossMark

A Semi-implicit Treatment of Porous Media in Steady-State CFD

Andreas Domaingo[1] · **Daniel Langmayr[2,3]** ·
Bence Somogyi[3] · **Raimund Almbauer[4]**

Abstract There are many situations in computational fluid dynamics which require the definition of source terms in the Navier–Stokes equations. These source terms not only allow to model the physics of interest but also have a strong impact on the reliability, stability, and convergence of the numerics involved. Therefore, sophisticated numerical approaches exist for the description of such source terms. In this paper, we focus on the source terms present in the Navier–Stokes or Euler equations due to porous media—in particular the Darcy–Forchheimer equation. We introduce a method for the numerical treatment of the source term which is independent of the spatial discretization and based on linearization. In this description, the source term is treated in a fully implicit way whereas the other flow variables can be computed in an implicit or explicit manner. This leads to a more robust description in comparison with a fully explicit approach. The method is well suited to be combined with coarse-grid-CFD on Cartesian grids, which makes it especially favorable for accelerated solution of coupled 1D–3D problems. To demonstrate the applicability and robustness of the proposed method, a proof-of-concept example in 1D, as well as more complex examples in 2D and 3D, is presented.

✉ Andreas Domaingo
andreas.domaingo@v2c2.at

Daniel Langmayr
daniel.langmayr@andritz.com

Bence Somogyi
bence.somogyi@ktm.at

Raimund Almbauer
raimund.almbauer@tugraz.at

[1] VIRTUAL VEHICLE Research Center, Inffeldgasse 21a, 8010 Graz, Austria

[2] ANDRITZ HYDRO GmbH, Dr.-Karl-Widdmann-Strasse 5, 8160 Weiz, Austria

[3] Present Address: KTM AG, Stallhofnerstrasse 3, 5230 Mattighofen, Austria

[4] Graz University of Technology, Inffeldgasse 25b, 8010 Graz, Austria

⚫ Springer

1 Introduction

Source terms are frequently encountered in computational fluid dynamics. They are necessary in situations, where processes affect the flow which are not resolved by the Navier–Stokes equations or the applied numerical grid. Examples of such processes are chemical reactions, turbulence, geometry simplifications in porous media, body force terms like buoyancy, etc. Such source terms not only affect the physics of the flow solution, but also have a large effect on the applied numerical scheme in terms of reliability, stability, and convergence behavior. One reason for an undesirable blow up due to source terms is that in the course of integration, the solution may come into regimes where the source terms exhibit large unrealistic values dominating the actual physics. Therefore, special care has to be taken with regard to source terms in order to guarantee solver robustness.

Numerous different approaches have been proposed to diminish the disadvantageous effects of source terms. In Yu and Chang (1997), a conservative interpolation of source terms in space–time is presented. A robust solution for conservation laws with source terms by means of a wave-propagation algorithm is discussed in LeVeque (1998). A discretization of the source terms by taking the natural upwind direction into account is considered in Vázquez-Cendón (1999) for the shallow water equations. A technique for balancing flux gradients and source terms is investigated in Hubbard and Garcia-Navarrot (2000). Implicit integration schemes of different order are evaluated in Mulder (1985) for the one-dimensional Euler equation. Other methods concentrate on solution-limited time stepping (Lian et al. 2009, 2010). In the latter method, the CFL number is reduced locally, to ensure the change in each cell to be below a specified tolerance thereby enhancing code reliability. Usually, as an initial step, the choice of implicit or explicit treatment has to be made (Piscaglia et al. 2009). This can be done based on the Jacobian of the source term or on the basis of individual terms (Merci et al. 2000). Generally speaking, positive source Jacobian eigenvalues are better handled by explicit algorithms, whereas negative eigenvalues improve the convergence of implicit approaches.

In this paper, we present an approach to efficiently model the influence of porous media resolved by source terms for steady-state CFD calculations. This is a common approach for investigations in which the porous medium cannot be fully resolved by the numerical grid (Hayes et al. 2008; Aslan et al. 2014; Hörmann et al. 2005). A well-known application is, e.g., the air flow in a vehicles underhood compartment or in HVAC ducts in the presence of heat exchangers (Janssen et al. 2015), where only the influence of the heat exchanger on the overall pressure loss of the system is of interest. In this case, the resistance, i.e., the local pressure loss, can be expressed by a suitable source term.

Our emphasis is on methods, in which the solution is found by quasi-time relaxation to the equilibrium state. Such a method has been proposed by the authors in Langmayr et al. (2013) for a calibrated coarse grid method (CCGM). The main goal of this approach is to reproduce the average results of a CFD simulation on a very fine grid with an under-resolved solution of the Euler or Navier–Stokes equations. This is established via appropriate averaging and source terms. The relaxation is controlled by a global relaxation time step, which is small enough to ensure a suitably small CFL number in the entire computational domain even for an explicit treatment of convection and diffusion terms. Pressure is treated semi-implicitly with SIMPLE (Patankar 1980). Additionally, source terms arising from porous media are treated implicitly, leading to local time steps to ensure stability, i.e., for each cell in the computational grid an individual relaxation time step is applied.

Therefore, the paper is organized as follows: First, we present the transport equations for the type of problem of interest in simplified form. Emphasis is put on the modeling of the source terms arising due to the presence of porous media and their influence on pressure–velocity coupling, while the details of discretization for convection and diffusion are not considered. The developed approach leads to a non-constant system matrix for the pressure equation. Therefore, we outline a procedure for the fast recalculation of this matrix in each iteration step in the next section. Finally, we provide a 1D proof-of-concept example, and more sophisticated 2D and 3D examples.

2 Theoretical Background

In this section, we review the basic equations describing the problem to be solved, and we present the necessary steps to derive the model equations.

2.1 Governing Equations

Our main goal is to describe subsonic air flow in the presence of porous media by means of coarse grid CFD. A typical example is the air flow through a vehicle's underhood compartment, where the different heat exchangers are modeled as porous media. Such a flow is governed by the Navier–Stokes equations

$$\nabla \cdot \rho \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\rho \mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} + \boldsymbol{\mathcal{S}}, \tag{2}$$

where $\mathbf{u}$ and p, respectively, denote the independent flow-field variable velocity and pressure. The air density is given by ρ and the viscosity by μ. Additional body forces like the resistance of the porous media are given by the source term $\boldsymbol{\mathcal{S}}$. It should be noted that this source terms account for geometric details which are not resolved by the numerical grid. In this approach, the flow within the porous medium is split into a free-stream part (with fully open geometry) and additional resistance forces (which account for flow area reduction and friction). Therefore, the source must be considered additionally to the convection and diffusion operators in this approach.

In vehicle aerodynamics, the heat exchanger resistance is commonly modeled by a suitable Darcy–Forchheimer law (Whitaker 1996; Rohsenow et al. 1998; Dukhan et al. 2014; Janssen et al. 2015). In our case, we follow the formulation as it is used, e.g., in OpenFOAM (Hafsteinsson 2009), which is given by

$$\mathcal{S}_i(u_i) = -\left(\mu D + \frac{1}{2}\rho|u_i|F\right)u_i. \tag{3}$$

Here, the Darcy constant D accounts for viscous losses (and corresponds to the inverse permeability) and the Forchheimer constant F for losses due to turbulent effects in the porous medium.

Generally for porous media, the sign of the source term is always opposite to the sign of the local velocity, and the derivative w.r.t. velocity is always negative, which is a crucial condition for a robust solution (Patankar 1981). Thus, an implicit treatment of the porous medium is always preferable (Merci et al. 2000).

2.2 Discretized Equations

When developing our discretized model equations from the Navier–Stokes equations, our main goal is to obtain a steady-state solution for the momentum equation Eq. (2). In our approach, the system relaxes from some initial state toward equilibrium by applying simple forward Euler time stepping

$$\frac{u_i^{(n+1)} - u_i^{(n)}}{\Delta t} = \mathcal{C}_i + \mathcal{D}_i + \mathcal{P}_i + \mathcal{S}_i, \tag{4}$$

where n denotes the iteration step for a relaxation parameter Δt (fictitious time step). Furthermore, $\mathcal{C}_i$, $\mathcal{D}_i$ and $\mathcal{P}_i$ denote the discretized forms of convection, diffusion, and the pressure gradient, respectively.

To account for mass conservation, Eq. (1), we apply the SIMPLE algorithm (Patankar 1980) to establish the velocity–pressure coupling. Basically, the update procedure for velocity is then given by

$$u_i^{(n+1)} = u_i^{(n)} + \Delta t \left[\mathcal{C}_i^{(n)} + \mathcal{D}_i^{(n)} + \mathcal{P}_i^{(n+1)} + (1 - \Psi)\mathcal{S}_i^{(n)} + \Psi \mathcal{S}_i^{(n+1)} \right], \tag{5}$$

where the superscripts (n) and $(n+1)$ denote, for which iteration step the discretization operators should be evaluated. Please note that the implicit treatment of the pressure via $\mathcal{P}_i^{(n+1)}$ is part of the SIMPLE strategy to ensure mass conservation, while the blending factor Ψ makes the source term implicit.

For the further treatment of the model equations, we make the following assumptions, which are well fulfilled by the Darcy–Forchheimer law Eq. (3):

- The source term depends explicitly on the velocity, i.e., an analytical dependence of the source term on velocity is provided: $\mathcal{S}_i = \mathcal{S}_i(u_i)$.
- The source term is sufficiently smooth to be linearized.

By virtue of these assumptions, the implicit contribution of the source term can be written in linearized form as

$$\mathcal{S}_i^{(n+1)} \approx \mathcal{S}_i^{(n)} + \left(\frac{d\mathcal{S}_i}{du_i} \right)^{(n)} \left(u_i^{(n+1)} - u_i^{(n)} \right)$$
$$:= \mathcal{S}_i^{(n)} + \mathcal{S}_i'^{(n)} \left(u_i^{(n+1)} - u_i^{(n)} \right). \tag{6}$$

Furthermore, we consider only linear discretizations of the pressure gradient. This allows to split the implicit pressure contribution into a contribution from the previous iteration and a corrective contribution, such that

$$p^{(n+1)} := p^{(n)} + \Delta p^{(n)} \implies \mathcal{P}\left(p^{(n+1)} \right) = \mathcal{P}\left(p^{(n)} \right) + \mathcal{P}\left(\Delta p^{(n)} \right)$$
$$:= \mathcal{P}^{(n)} + \Delta \mathcal{P}^{(n)}. \tag{7}$$

By inserting the simplifications Eqs. (6) and (7) into the update rule Eq. (5) and introducing an intermediate velocity u_i^*, we obtain

$$u_i^{(n+1)} = \left\{ u_i^{(n)} + \Delta t \left[\mathcal{C}_i^{(n)} + \mathcal{D}_i^{(n)} + \mathcal{P}_i^{(n)} + \mathcal{S}_i^{(n)} + \Psi \mathcal{S}_i'^{(n)} \left(u_i^* - u_i^{(n)} \right) \right] \right\}$$
$$+ \Delta t \left[\Delta \mathcal{P}_i^{(n)} + \Psi \mathcal{S}_i'^{(n)} \left(u_i^{(n+1)} - u_i^* \right) \right]. \tag{8}$$

To fix the value of the yet unknown intermediate velocity, we demand that it is equal to the term within curly braces in the above equation, yielding the final update rule

$$\Delta t^* := \frac{\Delta t}{1 - \Delta t \Psi \mathcal{S}_i'^{(n)}} \tag{9}$$

$$u_i^* = u_i^{(n)} + \Delta t^* \left[\mathcal{C}_i^{(n)} + \mathcal{D}_i^{(n)} + \mathcal{P}_i^{(n)} + \mathcal{S}_i^{(n)} \right] \tag{10}$$

$$u_i^{(n+1)} = u_i^* + \Delta t^* \Delta \mathcal{P}_i^{(n)}. \tag{11}$$

This update procedure corresponds to that of the SIMPLE algorithm with explicit source terms, except for the position-dependent time step Δt^*.

The obvious advantage of this approach is that the implicit treatment of the source terms neither increases the complexity nor decreases the speed of the SIMPLE algorithm, as no additional equation solving is required.

2.3 Pressure–Velocity Coupling

To complete the procedure presented in the previous section, a suitable pressure–velocity coupling must be implemented to ensure mass conservation according to the continuity equation Eq. (1). In the discretized form, continuity is given by

$$\mathcal{L}\left(\mathbf{u}^{(n+1)}\right) = \mathcal{L}\left(\mathbf{u}^* + \Delta t^* \Delta \mathcal{P}^{(n)}\right) = 0, \tag{12}$$

where the operator $\mathcal{L}$ denotes the discretized divergence operator. By assuming that the divergence operator $\mathcal{L}$ is linear, mass conservation can be written as

$$\mathcal{L}\left(\Delta t^* \Delta \mathcal{P}^{(n)}\right) = -\mathcal{L}\left(\mathbf{u}^*\right), \tag{13}$$

where the r.h.s. can be calculated after the predictor step from the intermediate u^*. By introduction of the influence function $\mathcal{F}$ with

$$\mathcal{F}\left(\Delta p^{(n)}\right) := \mathcal{L}\left(\Delta t^* \Delta \mathcal{P}^{(n)}\right) = \mathcal{L}\left[\Delta t^* \mathcal{P}\left(\Delta p^{(n)}\right)\right] \tag{14}$$

and the fact that the combined application of divergence and gradient operator can also be written as a matrix operation, the mass conservation takes the final form

$$\mathcal{F}\left(\Delta p^{(n)}\right) = A \Delta p^{(n)} = -\mathcal{L}\left(\mathbf{u}^*\right), \tag{15}$$

where A is the pressure influence matrix. Contrary to the explicit case, this matrix is not constant for all iterations steps, but has to be updated in each step because of the derivative $\mathcal{S}_i'^{(n)}$ which is part of the time step Δt^* in Eq. (12).

For a Cartesian grid with a first-order pressure gradient, the pressure influence matrix has a predominantly diagonal structure with six (3D) and four (2D), respectively, occupied side diagonals (c.f. Fig. 1). Additional matrix elements occur locally next to these "geometry" elements due to the implicit treatment of the porous media. Our target is to place an arbitrary number of porous regions within the computational domain. Moreover, we want to keep the approach general, i.e., not only for Darcy–Forchheimer porosities Eq. (3), but for any law which describes the pressure drop in dependence of velocity sufficiently smooth and monotonic. To this end, we describe the system not directly by the influence matrix A, but we claim to know the different influence functions to describe geometry ($\mathcal{F}_G$) and porous
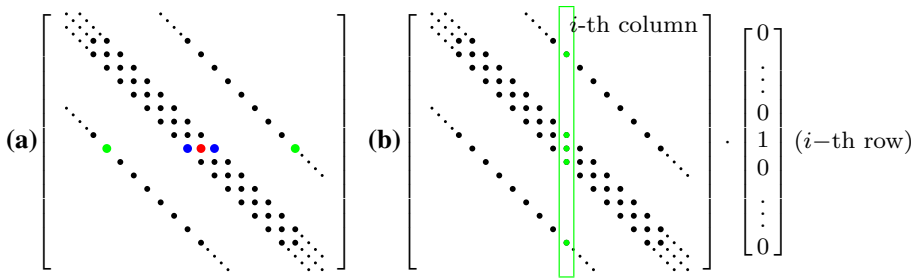
Fig. 1 **a** Typical structure of the pressure influence matrix for a 2D case. The *red dot* indicates a cell, which contains an additional source term due to a porous medium. The original matrix is changed due to source terms in *x*-direction (*blue*), *y*-direction (*green*) and both of them (*red*). **b** Projection of *i*-th column of influence matrix by suitable test vector

media ($\mathcal{F}_{P,i}$), where the latter can be derived from the pressure loss law. Due to this splitting, the total influence function reads

$$\mathcal{F}\left(\Delta p^{(n)}\right) = \mathcal{F}_G\left(\Delta p^{(n)}\right) + \sum_i \mathcal{F}_{P,i}\left(\Delta p^{(n)}\right). \qquad (16)$$

To determine the influence matrix automatically for an arbitrary number of influence functions, the following testing technique is applied: First, a series of pressure test vectors is defined by

$$\Delta \hat{p}_\ell := \delta_{\ell m}, \quad \ell, m = 1, \ldots, M, \qquad (17)$$

where M is the number of unknown pressure values in general the number of cells in the numerical grid and the Kronecker delta $\delta_{\ell m}$ is defined by

$$\delta_{\ell m} = \begin{cases} 1 & \text{if } \ell = m \\ 0 & \text{else.} \end{cases} \qquad (18)$$

Then, these test vectors are subsequently applied to the operator $\mathcal{F}$ to give the columns of the influence matrix by virtue of Eq. (13), c.f. Fig. 1, as

$$\mathcal{F}\left(\Delta \hat{p}_\ell\right) = A_{im}\delta_{\ell m} = A_{i\ell}. \qquad (19)$$

This method does not represent a new way of how velocity and pressure are coupled (this is still according to the SIMPLE algorithm). Many methods for this purpose have been proposed and intensively studied in the literature (Patankar 1980; Ferziger and Perić 2002; Raithby and Schneider 1979; Doormaal and Raithby 1984). The approach considered here is a formal framework, which can be applied to velocity–pressure coupling algorithms which perform a predictor step for the velocity equation and a corrector step for pressure equation to fulfill mass conservation:

1. Predictor step: $u^{(n)} \to u^*$ with mass imbalance Δm.
2. Corrector step: $A\Delta p = \Delta m$.

The necessary influence matrix A is not assembled directly, but all the details about the used discretization scheme are contained in the influence operator $\mathcal{F}$.

In this way, an easy and straight forward determination of the influence matrix is possible for any combination of applied source terms. The price to pay is that the operator $\mathcal{F}$ has to be evaluated M times per iteration step.

2.4 Accelerated Pressure–Velocity Coupling

To overcome the drawback mentioned in the last section, the fact that usually finite stencils are applied for the discretized evaluation of divergence $\mathcal{L}$ and gradient $\mathcal{P}$ can be exploited to reduce the number of required test vectors.

To this end, the single columns of the pressure influence matrix will be denoted by $\mathbf{a}_j$. The "structure" of the influence matrix can be described by the vectors $\mathbf{a}_j^0$ with

$$A^0 = \left\{\mathbf{a}_j^0\right\}_m = a_{j,m}^0 := 1 - \left|\mathrm{sgn}\left(a_{j,m}\right)\right|, \tag{20}$$

where the sign function sgn is defined by

$$\mathrm{sgn}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0, \end{cases} \tag{21}$$

so that all non-empty matrix elements are marked by a 1, as shown in Fig. 2. As the structure of the matrix is constant during calculation, and only the values change, these vectors can be determined in a preliminary step. Then, it is possible to find sets of columns denoted by $\mathcal{K}_i = \kappa_{i,l}$, such that the corresponding stencils of each set are completely independent, which can be expressed by

$$\mathbf{a}_{l_1}^0 \cdot \mathbf{a}_{l_2}^0 = \delta_{l_1,l_2} \quad \forall (l_1, l_2) \in \mathcal{K}_i. \tag{22}$$

These sets can be chosen such that

- $\bigcup_i (\mathcal{K}_i) = \{1, \ldots, M\}$,
- $\bigcap(\mathcal{K}_j, \mathcal{K}_k) = \emptyset$ for $j \neq k$,

i.e., the sets span the entire range of admissible columns without selecting any column twice.

Once these sets have been defined, new pressure test vectors can be constructed by

$$\Delta p_i^* := \sum_l \Delta \hat{p}_{\kappa_{i,l}}. \tag{23}$$

Since the stencils in the selected columns $\kappa_{i,l}$ are completely independent, a unique mapping can be established to reconstruct the influence matrix columns $\mathbf{a}_{\kappa_{i,l}}$ from the testing operation $\mathcal{F}(\Delta \hat{p}_i^*)$. Namely, all rows, for which the matrix structure vector a_j^0, $j \in \mathcal{K}_i$ are nonzero, belong to column j of the influence matrix.

The number of independent sets $\mathcal{K}$ depends on the used stencil for the discretization of the pressure gradient. Therefore, a constant number of test vectors Δp_i^* are necessary to obtain the pressure influence matrix, regardless of the number of cells in the numerical grid. For our typical application example with coarse grids with 10^3 to 10^4 cells, the number of test vectors is reduced to less than 20. Due to this fact, a tremendous speed-up of the calculation is possible, keeping in mind that the influence matrix must be reconstructed in every iteration step.

3 Numerical Results

In the following, we present two application examples of the developed model equations to proof the concept and applicability.
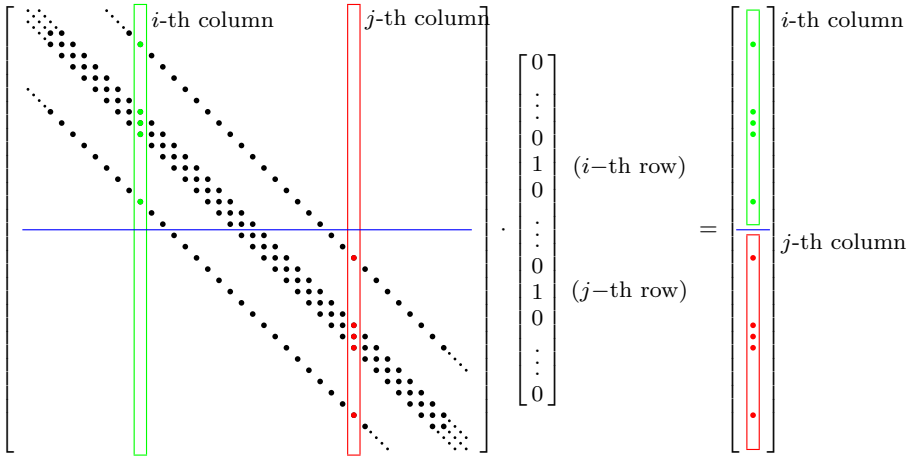
Fig. 2 Pressure influence structure matrix A^0. The i-th (*green*, a_i^0) and j-th (*red*, a_j^0) column don't have any nonzero elements in the same row, i.e., their stencils don't overlap: $a_i^0 \cdot a_j^0 = 0$

3.1 1D Flow Through a Pipe

As a proof-of-concept calculation, the laminar flow through a pipe is considered. On both sides of the pipe, a fixed pressure of 0 Pa is prescribed, while zero gradient conditions for velocity are applied. To drive the flow, a constant body force is given in the front region of the pipe ($150 \le x \le 250$ mm) in terms of a dimensionless Cp-value. In the rear region ($550 \le x \le 850$ mm), a porous medium with Darcy–Forchheimer law is positioned (cf. Fig. 3; Table 1).

According to the standard definition of the pressure coefficient (C_p), the pressure increase in the front region of the pipe is given by

$$\Delta p = \frac{1}{2}\rho U_\infty^2 C_p = 625.93\,\text{Pa}. \tag{24}$$
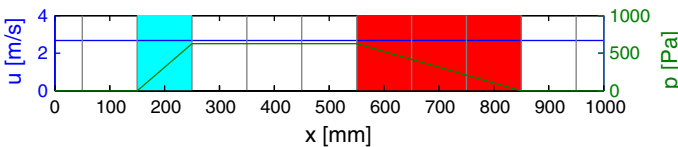
Fig. 3 Geometry for proof-of-concept calculation including regions with constant (*cyan*) and velocity-dependent (*red*, Darcy–Forchheimer) body forces. The analytical results are indicated for velocity (*blue*) and pressure (*green*), and the coarse grid resolution is indicated by *gray lines*

Table 1 Parameters for proof-of-concept calculation

C_p [1]	U_∞ (km/h)	ρ (kg/m^3)	μ (Pa s)	D (1/m^2)	F (1/m)
0.8	130	1.2	1.8×10^{-5}	3×10^7	150

By rewriting the source term for the Darcy–Forchheimer law, Eq. (3), in terms of the pressure drop over the porous region thickness $L = 0.3$ m, the equilibrium flow velocity u is given by

$$\Delta p = L \left(\mu D + \frac{1}{2} \rho F u \right) u \Rightarrow u = 2.67 \, \text{m/s}. \tag{25}$$

To demonstrate the effect of the implicit source term treatment independently of the CFL condition, the convection term is not considered for the present example, as the flow velocity is constant due to mass conservation, anyway.

The iteration history for different relaxation time steps is given in Fig. 4 for the velocity and in Fig. 5 for the residual du/dt. It can be observed that by application of the implicit source terms the correct solution can be obtained over a wide range of admissible relaxation time steps. A fully explicit scheme provides a correct solution only for small time steps, shows oscillatory behavior for medium size time steps, and does not converge at all for large time steps. For certain medium size time steps, a blending with factor $\Psi_s = 0.5$ provides a faster convergence than the fully implicit treatment, but for large time steps the fully implicit treatment is superior.

Consequently, in terms of robustness of the method, a fully implicit treatment with $\Psi_s = 1$ is the method of choice, which will be applied for the further test cases.

3.2 2D Flow Through a Channel

As a second example, the flow through a 2D channel partially blocked by a porous medium is considered. The problem dimensions as well as boundary conditions are shown in Fig. 6. It should be noted that a particular coarse grid is chosen for this demonstration example to investigate the application of the semi-implicit scheme in the framework of the CCGM (Langmayr et al. 2013). The porosity coefficients in longitudinal x-direction are given by
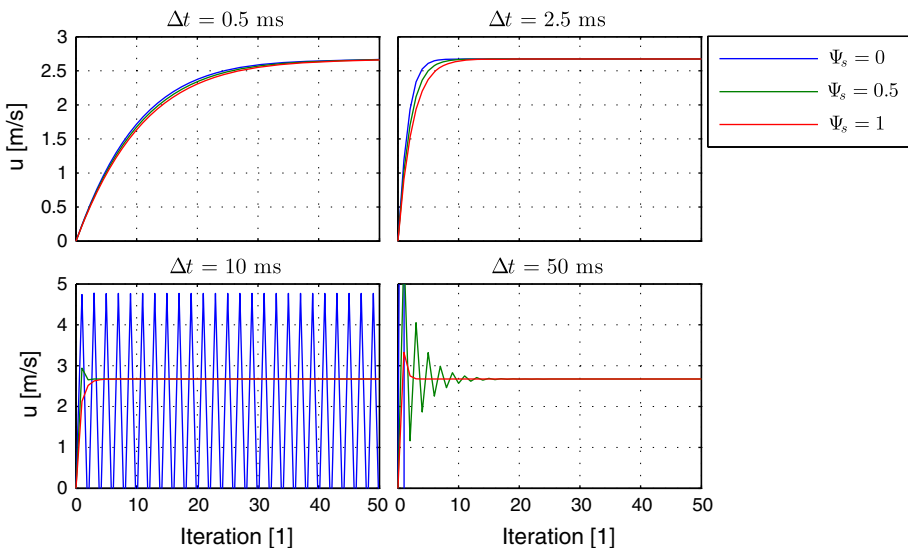

Fig. 4 Iteration history of velocity in simple channel for different relaxation time steps for explicit (*blue*), mixed (*green*) and implicit scheme (*red*)
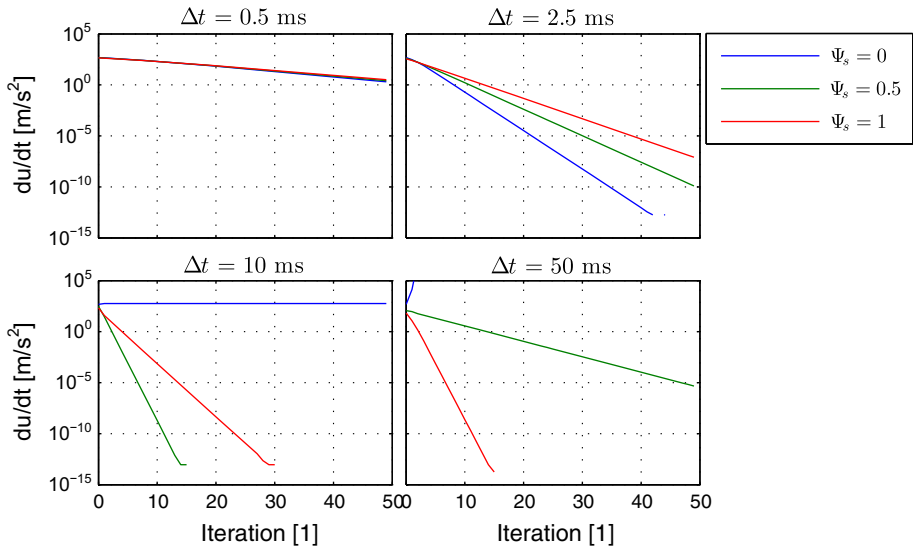
Fig. 5 Iteration history of residuals in simple channel for different time steps for explicit (*blue*), mixed (*green*) and implicit scheme (*red*)
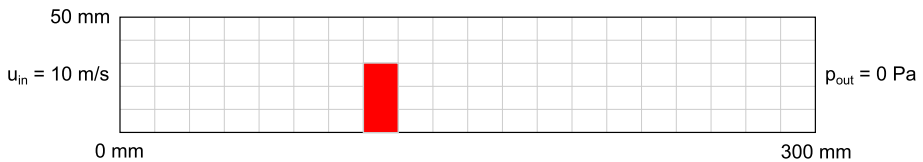

Fig. 6 Geometry for 2D channel flow with Darcy–Forchheimer porous medium in red. The coarse grid resolution is indicated by *gray lines*

- Darcy coefficient $D = 3 \times 10^7$ [$1/m^2$]
- Forchheimer coefficient $F = 150$ [$1/m$]

In the present case, the porous medium should represent the strongly anisotropic character encountered, e.g., in micro-channel heat exchangers. It is common practice to model the Darcy–Forchheimer coefficients in the lateral direction, by considering an additional factor f_{lat} compared to the longitudinal direction. Usually, this factor is in the range of 10^2–10^4 (Hafsteinsson 2009).

The first tests are performed for the weak anisotropic case with $f_{lat} = 100$. The pressure distribution and velocities for this case are shown in Fig. 7, with an average pressure at the inlet of $\bar{p}_{in} = 89$ Pa. Convergence tests for the explicit and implicit treatment of the source terms are conducted by performing calculations for different effective relaxation steps Δt. The basic step size is given by $\Delta t_0 = 10^{-4}$ s and the further steps by a series with $\Delta t_i = \Delta t_0 * 2^i$ s. The explicit case shows a stable convergence behavior for Δt_0 to Δt_3. With an implicit treatment of the source terms, step sizes up to Δt_6 are admissible. Detailed convergence curves are shown in Fig. 8, where the residuals for iteration step n are defined as
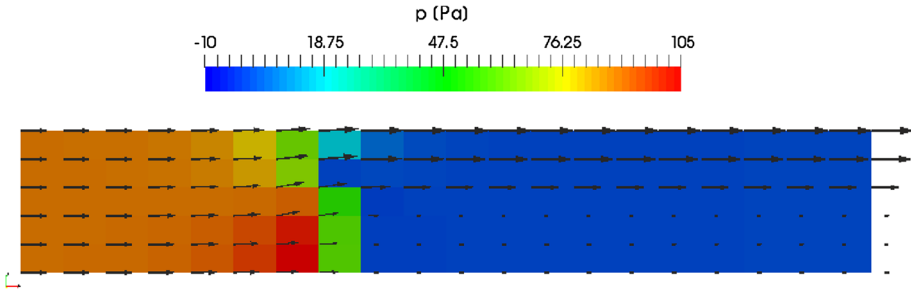
Fig. 7 Pressure distribution in channel with velocity at cell centers. The velocity vectors at the inlet on the *left side* correspond to 10 m/s
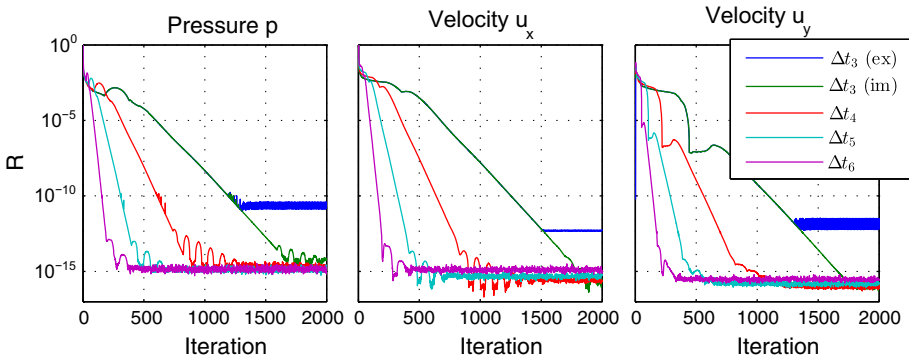

Fig. 8 Residuals for anisotropic porous medium with $f_{lat} = 100$ in dependency of effective relaxation step $\Delta t_i = 2^i \Delta t_0$. For the smallest considered time step Δt_3 explicit (ex) as well as implicit (im) calculations were performed

$$R_p(n) := \frac{\max\left(\left|p^{(n+1)} - p^{(n)}\right|\right)}{\bar{p}_{in}} \tag{26}$$

$$R_{u_d}(n) := \frac{\max\left(\left|u_d^{(n+1)} - u_d^{(n)}\right|\right)}{u_{in}}, \quad d = x, z. \tag{27}$$

The residuals for pressure and longitudinal velocity component show a predominantly exponential decay rate as theoretically expected. For the vertical velocity component, additional kinks are visible due to the source terms during the first iteration phase.

If the relaxation time step is small, so that also an explicit treatment is possible, the explicit and implicit scheme show the same convergence behavior in the initial iteration history, which was already observed for the 1D proof-of-concept calculation. However, the explicit scheme is not able to converge to the same precision as the implicit scheme. For example, the pressure residual displays saturation at about $R_p = 10^{-10}$ (c.f. Fig. 8).

Considering the exponential decay $R = \exp(-\beta_i n)$ with the decay rate $\beta_i := \beta \Delta t_i$, the ratio $r_{\beta,i} := \beta_i/\beta_1$ should correspond to $r_{\beta,i} \equiv i$ in the case of constant system matrices. A comparison between the ideal case and the decay coefficients extracted from the residuals in Fig. 8 is shown in Fig. 9. Although in our case the system matrices are not constant due to the local time step, Eq. (9), a decay behavior as predicted is observed. For the special application example, even a better convergence than in theory is achieved.
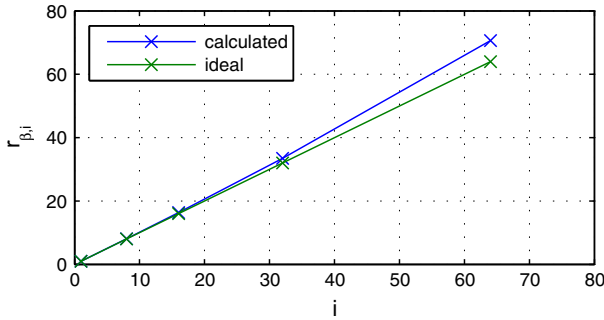
Fig. 9 Relative decay coefficient $r_{\beta,i}$ in dependency of the effective relative step size i

The execution wall times for the different cases are listed in Table 2. All calculations were performed for 2000 iterations on a laptop with Intel Core i7 CPU at 2.13 GHz. The used program code was in prototype state without any optimizations performed. The time difference between explicit and semi-implicit treatment of the source terms is about 2 %. With the largest possible relaxation step with semi-implicit scheme, a speed-up of about three can be achieved compared to the largest admissible step size with an explicit scheme (not taking into account that convergence is reached much faster and less than 2000 iterations would be required). If only 750 iterations are considered, the gain in speed is more than five.

Finally, the influence of the factor f_{lat} to model anisotropy is investigated. Figure 10 shows the relative vertical velocity component $w_{rel} := w/u_{in}$ in the topmost and bottommost velocity cell within the porous region. In the ideal case, the vertical velocity component should converge to 0, as the diameter of the micro channels goes to 0. For the considered case, the relative vertical velocity is about 3 % for $f_{\text{lat}} = 100$ and less than 1 % for $f_{\text{lat}} = 1000$ without the need to introduce additional baffles. For this degree of anisotropy, the calculation is stable for effective step sizes as large as Δt_6.

3.3 3D Flow Through a Channel

The final example treats the flow through a 3D channel partially blocked by a porous medium. The basic geometry is similar to the front region of the Pitz-Daily backward facing step (Pitz and Daily 1983). Additionally, a porous medium is placed in the upper half of the channel, c.f. Fig. 11. The boundary conditions and Darcy–Forchheimer coefficients are the same as for example 2. The factor for lateral forces was 1000 if not noted otherwise. Again, it should be noted that a particular coarse grid is chosen for this demonstration example to investigate the application of the semi-implicit scheme in the framework of the CCGM (Langmayr et al. 2013).

A comparison of the pressure distribution and velocities in the channel calculated with OpenFOAM and the proposed method is shown in Fig. 12 for different cut planes. Line plots

Table 2 Required calculation times for 2000 iterations for the different relaxation time steps

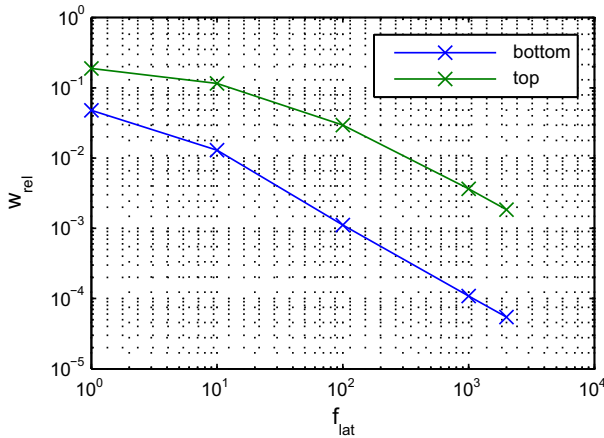| Case (relaxation time step)	Δt_3 (ex)	Δt_3 (im)	Δt_4	Δt_5	Δt_6
Wall time (s)	95.2	97.2	64.0	51.6	34.5

Fig. 10 Vertical velocity relative to inlet velocity within porous medium in dependency of anisotropy parameter f_{lat}
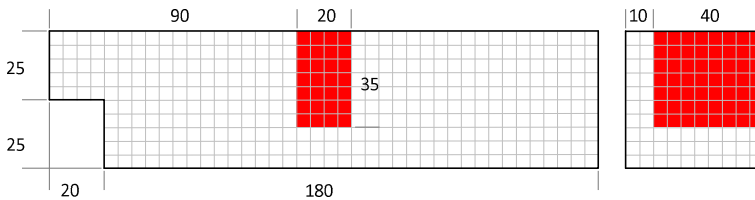

Fig. 11 Geometry for 3D channel flow with Darcy–Forchheimer porous medium in *red*. The coarse grid resolution is indicated by *gray lines*. Length is given in millimeter

of the in x-direction at $y = 0$ mm for different values of z are shown in Fig. 13. There is a good match of the results between the inlet and the entry plane of the porous medium. In the first cell row of the porous medium, OpenFOAM predicts higher pressures than our method and in the last cell row lower pressures. In the region behind the porous medium, the results obtained OpenFOAM show a small pressure increase, which seems to be unphysical at this position. The same is not the case for the proposed method.

4 Conclusions

In this paper, we presented a method for a semi-implicit modeling of porous media in the steady-state Navier–Stokes equations. This method corresponds to applying a suitable local time step in porous regions in the framework of time-explicit relaxation methods. Therefore, only the pressure equation has to be solved, which allows for a fast calculation. The applicability of this method was tested by three different cases: In the first case, the flow was driven by a body force and in the other ones by external boundary conditions. Since the semi-implicit treatment allows for a much larger relaxation time step, about ten times less iterations are necessary compared to the explicit treatment, to achieve the same residuals. This corresponds to a speedup of five in terms of wall time.

Furthermore, an automatic procedure to identify the source terms was presented and applied to the test cases. This procedure allows to "test" the influence of the porous medium
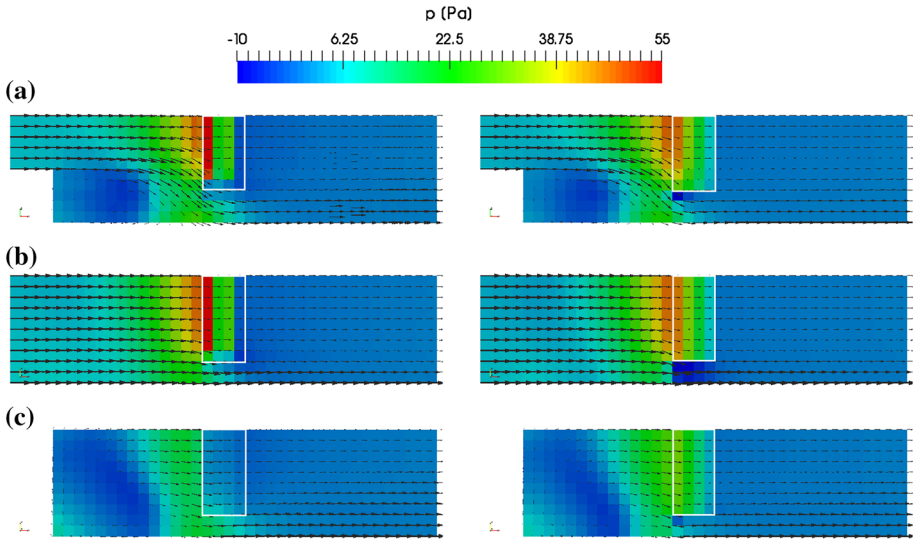
Fig. 12 Comparison of pressure distribution calculated with OpenFOAM (*left*) and presented method (*right*) for cut planes at $y = 0$ mm (**a**), $z = 10$ mm (**b**) and $z = -10$ mm (**c**). Velocity is indicated by *black arrows* which correspond to 10 m/s at the inlet on the *left side*. The position of the porous medium by the *white rectangle*
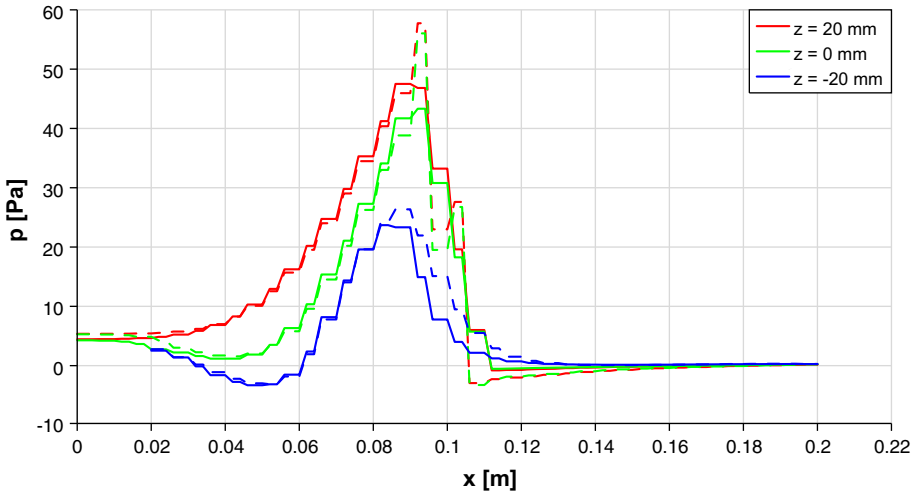

Fig. 13 Comparison of pressure in in plane $y = 0$ at different heights z between current approach (*solid*) and OpenFOAM (*dashed*)

on the flow structure. The advantage is that in this way also porous media can be considered, for which no analytic characteristic curve is available—provided that the characteristics is sufficiently smooth and monotonic. This method to determine the pressure influence matrix runs with a constant number of test operations independently on the used grid size. Therefore, it has hardly any influence on the calculation time for usual calculation setups, where the free-stream volume is large compared to the volume occupied by the porous medium.

Due to the observed properties, the proposed semi-implicit approach is a well-suited method for accelerating calibrated coarse grid methods (CCGM). As a consequence, the semi-implicit treatment of source terms arising in the presence of porous media is a promising tool for accelerated 1D–3D coupling in vehicle underhood flow calculation.

References

Aslan, A.R., Anbarci, K., Açikgöz, M.B., Icke, O., Aslan, O., Dinc, C.: Numerical Modelling and Prediction of the effect of cooling drag on the total vehicle drag. In: Proceedings of the Eighth International Conference on Computational Fluid Dynamics, Chengdu, China, ICCFD8-2014-0427 (2014)

Dukhan, N., Bağcı, Ö., Özdemir, M.: Experimental flow in various porous media and reconciliation of Forchheimer and Ergun relations. Exp. Therm. Fluid Sci. **57**, 425–433 (2014)

Ferziger, J.H., Perić, M.: Computational Methods for Fluid Dynamics, 3 rev. edn. Springer, Berlin (2002)

Hafsteinsson, H.E.: Porous Media in OpenFOAM. Chalmers University of Technology, Gothenburg (2009)

Hayes, A.M., Khan, J.A., Shaaban, A.H., Spearing, I.G.: The thermal modeling of a matrix heat exchanger using a porous medium and the thermal non-equilibrium model. Int. J. Therm. Sci. **47**, 1306–1315 (2008)

Hörmann, T., Lechner, B., Puntigam, W., Moshammer, T., Almbauer, R.A.: Numerical and experimental investigation of flow and temperature fields around automotive cooling systems. SAE Technical Paper 2005–01-2006 (2005)

Hubbard, M.E., Garcia-Navarrot, P.: Flux difference splitting and the balancing of source terms and flux gradients. J. Comput. Phys. **165**, 89–125 (2000)

Janssen, C., Effenberger, M., Gneiting, R., Kleber, A., Larpent, S., Stauch, R.: Improving 1D simulations of thermal management systems with embedded 3D CFD. In: Proceedings of 15. Internationales Stuttgarter Symposium, pp. 711–724. Springer Fachmedien Wiesbaden (2015)

Langmayr, D., Almbauer, R.A., Peller, N., Lichtenberger, A.: Calibrated coarse grid-finite volume method for the fast calculation of the underhood flow of a vehicle. J. Fluids Eng. **135**(10) 104502–104502-5 (2013)

LeVeque, R.J.: Balancing Source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. J. Comput. Phys. **146**, 346–365 (1998)

Lian, C., Xia, G., Merkle, C.L.: Solution-limited time stepping to enhance reliability in CFD applications. J. Comput. Phys. **228**, 4836–4857 (2009)

Lian, C., Xia, G., Merkle, C.L.: Impact of source terms on reliability of CFD algorithms. Comput. Fluids **39**, 1909–1922 (2010)

Merci, B., Steelant, J., Vierendeels, J., Riemslagh, K., Dick, E.: Computational treatment of source terms in two-equation turbulence models. AIAA J. **38**(11), 2085–2093 (2000)

Mulder, W.A.: Experiments with implicit upwind methods for the Euler equations. J. Comput. Phys. **59**, 232–246 (1985)

Patankar, S.H.: Numerical Heat Transfer and Fluid Flow. McGraw-Hill, Washington (1980)

Patankar, S.H.: A calculation procedure for two-dimensional elliptic situations. Numer. Heat Transf. **4**, 409–425 (1981)

Piscaglia, F., Montorfano, A., Onorati, A.: Development of a multi-dimensional parallel solver for full-scale DPF modeling in OpenFOAM. SAE Technical Paper 2009–01-1965 (2009)

Pitz, R.W., Daily, J.W.: Combustion in a turbulent mixing layer formed at a rearward-facing step. AIAA J. **21**(11), 1565–1570 (1983)

Raithby, G.D., Schneider, G.E.: Numerical solution of problems in incompressible fluid flow: treatment of the velocity-pressure coupling. Numer. Heat Transf. **2**, 417–440 (1979)

Rohsenow, W.M., Hartnett, J.P., Cho, Y.I. (eds.): Handbook of heat transfer. McGraw-Hill, Boston (1998)

Van Doormaal, J.P., Raithby, G.D.: Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numer. Heat Transf. **7**, 147–163 (1984)

Vázquez-Cendón, M.E.: Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. J. Comput. Phys. **148**, 497–526 (1999)

Whitaker, S.: The Forchheimer equation: a theoretical development. Transp. Porous Med. **25**, 27–61 (1996)

Yu, S.T., Chang, S.C.: Treatments of stiff source terms in conservation laws by the method of space-time conservation element and solution element. AIAA 97–0435 (1997)