Computers and Fluids 181 (2019) 188–207

Contents lists available at ScienceDirect

Computers and Fluids

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

Implementation of an implicit pressure–velocity coupling for the Eulerian multi-fluid model

Gabriel G.S. Ferreira[a], Paulo L.C. Lage[a,*], Luiz Fernando L.R. Silva[b], Hrvoje Jasak[c,d]

[a] Programa de Engenharia Química, COPPE, Universidade Federal do Rio de Janeiro, P.O. Box 68502, Rio de Janeiro, RJ 21941-972, Brazil
[b] Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ 21941-909, Brazil
[c] Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Ivana Lučića 5, Zagreb, Croatia
[d] Wikki Ltd, 459 Southbank House, SE1 7SJ, London, United Kingdom

ARTICLE INFO

Article history:
Received 29 January 2018
Revised 19 November 2018
Accepted 15 January 2019
Available online 15 January 2019

Keywords:
Pressure–velocity implicit coupling
Eulerian multi-fluid model
Finite volume method
Momentum interpolation consistency
Multiphase flow
CFD

ABSTRACT

An implicitly coupled pressure–velocity transient scheme for the solution of the Eulerian multi-fluid model with any number of phases (MIC) was developed and implemented in OpenFOAM®. The numerical methodology is based on the phase-intensive momentum equations for the phase velocities and a pressure equation using a deferred-correction approach of the Rhie–Chow interpolation. An extension of the compact momentum interpolation method (CMI) to a system with any number of phases was developed. The developed methodology was tested against a segregated multiphase solver (MS) and a steady-state single phase solver, in cases considering up to four phases. The proposed method has proven to be more robust and accurate than the segregated counterpart, being able to converge in cases with large drag coefficients, providing time-step independent steady-state solutions.

1. Introduction

A multiphase flow simulator that embraces all possible flow regimes and number of phases would be of great use for research and development in several industries. However, after more than three decades since the development of the first two-phase CFD models [32], problem-specific modeling and solution strategies are preferred both in segregated and in dispersed flows. The explanations for the lack of generalization fall into two main categories. The first is the physical complexity of the multiphase flows, which commonly leads to scale filtering approximations during the development of the models, leading to physical uncertainties and the need of parameter fitting, reducing its range of applicability. The second is the poor robustness of the solution algorithms, when changes in the value of physical coefficients may lead to convergence difficulties or even failure by divergence of the numerical solution.

Regarding the physical model, one of the main strategies employed for the simulations of multiphase flows is the usage of an Eulerian multi-fluid model [15]. As it is still unfeasible to accurately solve all the scales of the multiphase flow in industrial applications, an averaging procedure is applied to the mass and momentum conservation equations and the resulting interfacial momentum exchange terms are modeled through closure laws, instead of directly calculated from the solution. Then, the physical uncertainties arise mainly from the development and selection of the appropriate closure models [15] for the inter-phase transfer laws, stress tensor, turbulence, etc.

However, even for cases where the closure laws yield a reasonable physical description of the system, it may be possible that the numerical procedure is not robust enough, leading to wrong, or even non-physical results. For instance, there exist several pressure-based algorithms to solve the pressure–velocity coupling in incompressible single phase flows and each algorithm can be potentially extended to a multiphase flow model [23]. As each formulation has its own advantages and drawbacks, it is very difficult to develop a single methodology that applies to all cases.

Most modern solution algorithms are based on a collocated variable arrangement, where both the pressure and velocity are stored in the center of the computational mesh cells. However, during the finite volume discretization, it is necessary to compute the volumetric fluxes at the cell faces, and the application of an inappropriate numerical discretization scheme may lead to decoupling of the pressure and velocity fields, leading to non-physical pressure oscillations. This problem was solved by Rhie and Chow [27] for steady single phase incompressible flows. They introduced a special interpolation scheme for the evaluation of the volumet-

* Corresponding author.
E-mail address: paulo@peq.coppe.ufrj.br (P.L.C. Lage).

https://doi.org/10.1016/j.compfluid.2019.01.018
0045-7930/

Nomenclature

$(\cdot)_f$	cell to face interpolation operator
A	cross section area divided by particle volume (m^{-1})
$\mathbf{A}$	main diagonal submatrix of the linear system obtained from discretization of part of the momentum conservation equation (s^{-1})
C	drag coefficient
Co	Courant number
d	dispersed phase diameter (m)
D_P	pressure equation coefficient (kg^{-1} s m^3)
E	error
$\mathbf{F}_B$	body force vector (m s^{-2})
$\mathbf{g}$	gravitational acceleration (m s^{-2})
$\mathbf{H}$	off-diagonal part of the linear system obtained from discretization of part of the momentum conservation equation, $\boldsymbol{\Upsilon}_S - (\boldsymbol{\Upsilon}_A - \mathbf{A})\mathbf{u}$ (m s^{-2})
K	generalized drag coefficient (kg $m^{-3}s^{-1}$)
$\mathbf{M}$	interfacial momentum transfer term (kg m^{-2} s^{-2})
n	number of control volumes in the mesh
N	number of iterations
p	pressure (kg m^{-1} s^{-2})
P	number of phases
Q	number of parallel processes
r	phase fraction
Re	Reynolds number
S	speedup
$\mathbf{S}$	face area vector (m^2)
t	time (s)
u	velocity component (m s^{-1})
$\mathbf{u}$	velocity vector (m s^{-1})
$\mathbf{x}$	position vector (m)

Greek letters

Δ	variation
ϵ_t	transient deviation
η	parallelism efficiency
Θ	ratio of numerical coefficients
λ	tolerance
ν	kinematic viscosity (m^2 s^{-1})
ρ	density (kg m^{-3})
$\boldsymbol{\tau}$	viscous stress tensor per unit mass (m^2 s^{-2})
$\boldsymbol{\Upsilon}$	numerical coefficients of the linear system obtained from the discretization of part of the momentum conservation equation (m s^{-2})
ϕ	volumetric flux at the cell faces (m^3 s^{-1})

Subscripts

α	relative to phase α
β	relative to phase β
χ	relative to phase or variable χ
A	indicates the coefficient matrix of a discretized linear system
c	continuous phase
d	dispersed phase
D	relative to the drag term
f	face value
F_B	relative to the body force term
inner	inner iterations
m	relative to the mixture
outer	outer iterations
pU	relative to the pressure–velocity inner loop
x	x component
y	y component
r	relative quantity
S	indicates the source vector of a discretized linear system
T	relative to the temporal term

Superscripts

*	modified
$\hat{}$	pseudo (velocity or flux)
abs	absolute
C	correction
comp	computational
k	value at present (k) iteration
$k-1$	value at previous ($k-1$) iteration
rel	relative
$t-1$	at the previous time instant
T	transpose
tot	total

Abbreviations

AAMG	Agglomerative algebraic multigrid
BFS	Backward-facing step
CFD	Computational fluid dynamics
CMI	Compact momentum interpolation
GAMG	Generalized algebraic multigrid
HC	Horizontal channel
IPSA	Inter-Phase Slip Algorithm
MIC	Multiphase implicitly coupled
MS	Multiphase segregated
OpenFOAM	Open source field operation and manipulation
PEA	Partial Elimination Algorithm
pUC	Pressure–velocity implicitly coupled
PISO	Pressure-implicit with splitting of operator
SAMG	Selective algebraic multigrid
SIMPLE	Semi-implicit method for pressure-linked equations

ric face fluxes, applying a correction for the pressure gradient term that is commonly referred as Rhie–Chow interpolation.

Later, this interpolation method was extended and modified for several other applications. It was shown by Majumdar [21] and Miller and Schmidt [22] that the results obtained using the Rhie–Chow interpolation are dependent on the value of the relaxation factors, and they both proposed different solutions for the problem. Choi [2] verified that the direct extension of the Rhie–Chow interpolation to transient problems yielded time-step size dependent solutions, proposing a modified momentum interpolation method to remove this dependency. However, it was later shown by Kawaguchi et al. [19] that the approach proposed by Choi [2] was still dependent on the time-step size, being the first to propose a momentum interpolation method that is both independent on the time-step size and relaxation factor values. However, before the analysis performed by Kawaguchi et al. [19], Shen et al. [30] reported an improved Rhie–Chow interpolation method for unsteady flows that can be considered the first to provide solutions that are truly independent from the time-step size.

More recently, Cubero and Fueyo [4] developed the so-called compact momentum interpolation procedure, an alternative momentum interpolation approach that yielded converged solutions that were also independent of the time-step and relaxation factor values. This methodology is based on a special interpolation practice for the numerical coefficients that arise from the discretization of the temporal term, leading to a new correction formula for the volumetric face fluxes that considers not only the pressure gradient, but also the contributions from the time discretization and relaxation schemes. The main advantages of the approach of

Cubero and Fueyo [4] over the one presented by Kawaguchi et al. [19] is the possibility to treat both linear and inertial relaxation factors and the easy generalization of the temporal correction term to higher-order time discretization schemes.

Another key aspect in the solution of the Navier–Stokes equations is the choice of an adequate solution algorithm for the system composed by the momentum and continuity equations. For years, the most popular algorithms were those based on the SIMPLE procedure [25]. These methods are commonly referred as segregated algorithms, being based on the separate solution of the momentum and pressure equations that are iterated until convergence. However, as pointed out by Darwish et al. [8], with the increasing availability of computer memory and due to scalability problems in the segregated methods, the implicitly coupled methods have gained renewed interest. Darwish et al. [8] implemented an implicitly coupled scheme for the solution of steady single phase flows, showing that the CPU times of the coupled approach are substantially smaller than those of the segregated one, with larger speedups for finer meshes.

When dealing with multiphase flows, care must be taken not only with the pressure–velocity solution algorithm, but also with the interfacial momentum coupling method. A common practice is to consider also a Rhie–Chow-like correction for the drag and the body force terms [23]. Cubero et al.[5] have extended their compact momentum interpolation formulation for unsteady two-phase flows by using a special interpolation practice for the drag coefficient, which included a drag correction for the face flux equations, resulting in the compact momentum interpolation method (CMI). They have shown that the absence of the drag correction led to spurious relative velocities, that created non-physical oscillations in the volumetric phase fraction fields. Their method was based on the most popular solution algorithm for two-phase flows, the Inter-Phase Slip Algorithm (IPSA), which is a generalization of the single-phase SIMPLE method for unsteady two-phase flows [32]. However, the IPSA method has some drawbacks, as it converges very slowly when the interfacial momentum transfer term is dominant. This problem was solved by the development of the Partial Elimination Algorithm (PEA) [33], which creates an explicit approximation for the phase velocity that can be substituted into the other phase momentum equation to enhance the coupling between the phases. However, the PEA method is not easily generalized to a many phase system and it can only be applied to decouple one phase pair at a time [24]. A remedy for this issue is to use an implicitly coupled method to solve all the momentum equations simultaneously, also coupling the inter-phase momentum transfer terms implicitly. Darwish and Moukalled [6] have implemented a steady two-phase implicitly coupled solver considering the simultaneous solution of the pressure and the phase velocities, obtaining speedups between 1.3 and 4.6, in comparison to the segregated solution approach. Usually, the smaller the particles, the stronger is the coupling between the phases. Thus, several applications that deals with small particles present critical interfacial momentum coupling issues, such as gas-solid particle flows [35] and emulsion flows [9].

In this work we developed a solver for the solution of the unsteady multi-fluid model with any number of phases. This was achieved by combining a generalization of the CMI method [5] to a many phase system with the implicitly coupled solution for the pressure and all phase velocity fields. The CMI is used on the development of a volumetric face flux equation that is used to develop the multiphase pressure equation, that considers the pressure, temporal, drag and body force corrections. The methodology was implemented using the block-coupled matrix structure implemented in `foam-extend`, a fork of the OpenFOAM® software. The developed code is tested and verified against a steady implicitly coupled single phase and a multiphase segregated solver, being the latter developed by extending the conventional segregated code, `twoPhaseEulerFoam`, that is available in `foam-extend`. The parallel scalabilities of both the coupled and segregated multiphase solvers were also evaluated.

2. Eulerian multi-fluid model

2.1. Multi-fluid equations

In this work we use the Eulerian multi-fluid model, which is a generic framework for the treatment of multiphase flows with any number of phases. For a system of P incompressible phases and neglecting mass transfer between phases, the phasic mass and momentum conservation equations reduce to [13]:

$$\frac{\partial (r_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha) = 0, \tag{1}$$

$$\frac{\partial (r_\alpha \rho_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha) = -r_\alpha \nabla p - \nabla \cdot (r_\alpha \rho_\alpha \tau_\alpha) + r_\alpha \rho_\alpha \mathbf{g} + \mathbf{M}_\alpha \tag{2}$$

where r is the volumetric phase fraction, ρ is the density, $\mathbf{u}$ is the velocity, p is pressure shared by all the phases and $\mathbf{g}$ is the gravity field. The stress tensor per unit mass $\boldsymbol{\tau}$ for a laminar flow, assuming a Newtonian functional form, is given as:

$$\tau_\alpha = -\nu_\alpha \left[2\mathbf{D}_\alpha - \frac{2}{3}(\nabla \cdot \mathbf{u}_\alpha)\mathbf{I} \right] \tag{3}$$

$$\mathbf{D}_\alpha = \frac{1}{2}\left[\nabla \mathbf{u}_\alpha + (\nabla \mathbf{u}_\alpha)^t \right], \tag{4}$$

where ν is the kinematic viscosity. The interfacial momentum transfer terms are written in the following form:

$$\mathbf{M}_\alpha = \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^{P} \mathbf{M}_{\alpha,\beta} \tag{5}$$

where $\mathbf{M}_{\alpha,\beta}$ is the momentum exchanged between phases α and β, such that $\mathbf{M}_{\alpha,\beta} = -\mathbf{M}_{\beta,\alpha}$. Usually, these interfacial transfer terms are decomposed in drag, lift and virtual mass forces, among others, but in this work we considered only the drag force, since in the current analysis we were interested in cases where it is the dominant term. Then, $\mathbf{M}_{\alpha,\beta}$ is written as:

$$\mathbf{M}_{\alpha,\beta} = r_\alpha r_\beta K_{\alpha\beta} \mathbf{u}_{r,\alpha\beta} \tag{6}$$

where $K_{\alpha\beta}$ is the generalized drag coefficient and $\mathbf{u}_{r,\alpha\beta} = \mathbf{u}_\beta - \mathbf{u}_\alpha$ is the relative velocity between phases α and β. For cases where phase inversion is possible, Weller [36] proposed the following symmetric model for the generalized drag coefficient:

$$K_{\alpha\beta} = \frac{1}{2}\left(r_\alpha \rho_\beta A_\alpha C_{\alpha,\beta} + r_\alpha \rho_\beta A_\beta C_{\beta,\alpha} \right) \left| \mathbf{u}_{r,\alpha\beta} \right| \tag{7}$$

where $C_{\alpha,\beta}$ is the drag coefficient considering phase α as the dispersed phase and phase β the continuous phase, A_α is the α-phase particle projected area normal to the relative velocity divided by its volume. For a spherical particle with diameter d_α, we have:

$$A_\alpha = \frac{3}{2 d_\alpha} \tag{8}$$

The drag coefficient was calculated using a modified formulation of the Schiller and Naumann [29] correlation:

$$C_{\alpha,\beta} = \max\left[0.44, \frac{24}{Re_{\alpha,\beta}}\left(1 + 0.15 Re_{\alpha,\beta}^{0.687}\right) \right] \tag{9}$$

where the particle Reynolds number is defined by:

$$Re_{\alpha,\beta} = \frac{\rho_\alpha \mathbf{u}_{r,\alpha\beta} d_\alpha}{\mu_\beta} \tag{10}$$

where μ is the dynamic viscosity.

3. Numerical formulation

In this section we describe the numerical formulation used in this work, detailing the two different coupling algorithms considered for the multiphase flow solution.

3.1. Phase intensive formulation

In the current work we applied the phase-intensive formulation for the momentum equations, the same used by Rusche [28] and Silva and Lage [31]. It is obtained by dividing Eq. (2) by ρ_α and r_α and making some rearrangements:

$$\begin{aligned}\frac{\partial \mathbf{u}_\alpha}{\partial t} &+ \nabla\cdot(\mathbf{u}_\alpha\mathbf{u}_\alpha) - \mathbf{u}_\alpha(\nabla\cdot\mathbf{u}_\alpha) - \nabla\cdot\left(\nu_\alpha\frac{\nabla r_\alpha}{r_\alpha}\mathbf{u}_\alpha\right)\\ &+ \mathbf{u}_\alpha\nabla\cdot\left(\nu_\alpha\frac{\nabla r_\alpha}{r_\alpha}\right) - \nabla\cdot(\nu_\alpha\nabla\mathbf{u}_\alpha)\\ &+ \nabla\cdot\tau_\alpha^C + \frac{\nabla r_\alpha}{r_\alpha}\cdot\tau_\alpha^C = \frac{\mathbf{M}_\alpha}{r_\alpha\rho_\alpha} - \frac{1}{\rho_\alpha}\nabla p + \mathbf{g}\end{aligned} \tag{11}$$

where the stress correction term, τ_α^C, is defined by

$$\tau_\alpha^C = -\nu_\alpha\left[(\nabla\mathbf{u}_\alpha)^T - \frac{2}{3}(\nabla\cdot\mathbf{u}_\alpha)\mathbf{I}\right]. \tag{12}$$

Then, in order to stabilize the gravity source term in cases with large buoyant forces, we applied a modified formulation of the gravity and pressure gradient term [28]. The formulation is based on the definition of a modified pressure, given as:

$$p^* = p - \rho_m(\mathbf{g}\cdot\mathbf{x}) \tag{13}$$

where ρ_m is the mixture average density, $\rho_m = \sum_{\alpha=1}^{P} r_\alpha\rho_\alpha$, and $\mathbf{x}$ is the position vector with respect to an inertial reference frame. Using Eq. (13), we can find that [12,28]:

$$-\frac{1}{\rho_\alpha}\nabla p + \mathbf{g} = -\frac{1}{\rho_\alpha}\nabla p^* + \mathbf{F}^*_{B\alpha}, \tag{14}$$

with the modified body force $\mathbf{F}^*_{B\alpha}$ being written as:

$$\mathbf{F}^*_{B\alpha} = -\frac{1}{\rho_\alpha}(\mathbf{g}\cdot\mathbf{x})\nabla\rho_m + \left(1 - \frac{\rho_m}{\rho_\alpha}\right)\mathbf{g} \tag{15}$$

Using Eq. (14), Eq. (11) can be rewritten as:

$$\begin{aligned}\frac{\partial \mathbf{u}_\alpha}{\partial t} &+ \nabla\cdot(\mathbf{u}_\alpha\mathbf{u}_\alpha) - \mathbf{u}_\alpha(\nabla\cdot\mathbf{u}_\alpha) - \nabla\cdot\left(\nu_\alpha\frac{\nabla r_\alpha}{r_\alpha+\delta}\mathbf{u}_\alpha\right)\\ &+ \mathbf{u}_\alpha\nabla\cdot\left(\nu_\alpha\frac{\nabla r_\alpha}{r_\alpha+\delta}\right) - \nabla\cdot(\nu_\alpha\nabla\mathbf{u}_\alpha)\\ &+ \nabla\cdot\tau_\alpha^C + \frac{\nabla r_\alpha}{r_\alpha+\delta}\cdot\tau_\alpha^C = \frac{\mathbf{M}_\alpha}{r_\alpha\rho_\alpha} - \frac{\nabla p^*}{\rho_\alpha}\\ &- \frac{1}{\rho_\alpha}(\mathbf{g}\cdot\mathbf{x})\nabla\rho_m + \left(1 - \frac{\rho_m}{\rho_\alpha}\right)\mathbf{g}\end{aligned} \tag{16}$$

where δ is a positive small parameter introduced to avoid division by zero.

3.2. Phase fraction equations

The phase fraction equation formulation is based on the one introduced by Weller [36]. It was later generalized to a multiphase mixture with any number of phases by Silva and Lage [31] to yield:

$$\frac{\partial r_\alpha}{\partial t} + \nabla\cdot(\bar{\mathbf{u}}r_\alpha) + \nabla\cdot\left(\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P} r_\beta r_\alpha\mathbf{u}_{r,\alpha\beta}\right) = 0 \tag{17}$$

where $\bar{\mathbf{u}}$ is the mixture average velocity, defined by $\bar{\mathbf{u}} = \sum_{\alpha=1}^{P} r_\alpha\mathbf{u}_\alpha$. This formulation has the advantage of ensuring boundedness of r_α, since the terms are written in a conservative form, and a better coupling between the phases is achieved by the presence of the relative velocities [28].

Eq. (17) was discretized in the following form:

$$\left\lfloor\frac{\partial[r_\alpha]}{\partial t}\right\rfloor + \lfloor\nabla\cdot(\bar{\mathbf{u}}[r_\alpha])\rfloor + \left\lfloor\nabla\cdot\left(\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}\mathbf{u}_{r,\alpha\beta}r_\beta[r_\alpha]\right)\right\rfloor = 0 \tag{18}$$

where the terms written as $\lfloor[\psi]\rfloor$ are discretized implicitly with respect to variable ψ. Eq. (18) was sequentially solved for each phase after the solution of the pressure and velocity discretized equations in both the segregated (MS) and implicitly coupled (MIC) solvers.

3.3. Segregated pressure-velocities coupling scheme

In this section we describe the numerical methodology applied for the multiphase segregated (MS) solver.

3.3.1. Multiphase momentum equations

The MS scheme used as a reference implementation in this work is mainly based on the implementation of Silva and Lage [31]. The main differences arise from the consideration of the symmetric drag formulation (Eq. (7)), the usage of the modified pressure (Eq. (13)) and the absence of the lift and virtual mass momentum transfer terms. Despite these differences, we adopted a similar discretization procedure and solution algorithm. The discretization of the left-hand side of Eq. (16) gives [28]:

$$\begin{aligned}\boldsymbol{\Upsilon}_\alpha :=& \left\lfloor\frac{\partial[\mathbf{u}_\alpha]}{\partial t}\right\rfloor + \lfloor\nabla\cdot(\mathbf{u}_\alpha[\mathbf{u}_\alpha])\rfloor - \lfloor(\nabla\cdot\mathbf{u}_\alpha)[\mathbf{u}_\alpha]\rfloor\\ &- \left\lfloor\nabla\cdot\left(\nu_\alpha\frac{\nabla r_\alpha}{r_\alpha+\delta}[\mathbf{u}_\alpha]\right)\right\rfloor\\ &- \lfloor\nabla\cdot(\nu_\alpha\nabla[\mathbf{u}_\alpha])\rfloor + \left\lfloor\frac{\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P} r_\beta K_{\alpha\beta}[\mathbf{u}_\alpha]}{\rho_\alpha}\right\rfloor\\ &+ \left\lfloor\nabla\cdot\left(\nu_\alpha\frac{\nabla r_\alpha}{r_\alpha+\delta}\right)[\mathbf{u}_\alpha]\right\rfloor + \nabla\cdot\tau_\alpha^C + \frac{\nabla r_\alpha}{r_\alpha+\delta}\cdot\tau_\alpha^C\end{aligned} \tag{19}$$

where $\boldsymbol{\Upsilon}_\alpha$ represents the numerical coefficients of the linear system obtained in the discretization. Considering that the linear system of equations are stored in $\boldsymbol{\Upsilon}_\alpha$ in the form

$$(\boldsymbol{\Upsilon}_\alpha)_A\mathbf{u}_\alpha = (\boldsymbol{\Upsilon}_\alpha)_S, \tag{20}$$

the matrix coefficients of $(\boldsymbol{\Upsilon}_\alpha)_A$ and the source term $(\boldsymbol{\Upsilon}_\alpha)_S$ can be used to assemble the following approximation to Eq. (16):

$$\mathbf{A}_\alpha\mathbf{u}_\alpha - \mathbf{H}_\alpha = -\frac{\nabla p^*}{\rho_\alpha} - \frac{(\mathbf{g}\cdot\mathbf{x})\nabla\rho_m}{\rho_\alpha} + \left(1 - \frac{\rho_m}{\rho_\alpha}\right)\mathbf{g} + \frac{\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P} r_\beta K_{\alpha\beta}\mathbf{u}_\beta}{\rho_\alpha} \tag{21}$$

where $\mathbf{A}_\alpha$ represents the diagonal coefficients of $(\boldsymbol{\Upsilon}_\alpha)_A$ and $\mathbf{H}_\alpha$ is calculated explicitly using the current-level of iteration value of $\mathbf{u}_\alpha$ by $\mathbf{H}_\alpha = (\boldsymbol{\Upsilon}_\alpha)_S - (\boldsymbol{\Upsilon}_\alpha)_N\mathbf{u}_\alpha$, being $(\boldsymbol{\Upsilon}_\alpha)_N$ the non-diagonal part of the matrix $(\boldsymbol{\Upsilon}_\alpha)_A$, that is $(\boldsymbol{\Upsilon}_\alpha)_N = (\boldsymbol{\Upsilon}_\alpha)_A - \mathbf{A}_\alpha$.

Then, using Eq. (21) we can write the following explicit approximation for the phase velocities:

$$\mathbf{u}_\alpha = \frac{\mathbf{H}_\alpha}{\mathbf{A}_\alpha} - \frac{\nabla p^*}{\rho_\alpha \mathbf{A}_\alpha} - \frac{(\mathbf{g}\cdot\mathbf{x})\nabla\rho_m}{\rho_\alpha \mathbf{A}_\alpha} + \left(1-\frac{\rho_m}{\rho_\alpha}\right)\frac{\mathbf{g}}{\mathbf{A}_\alpha} + \frac{\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P} r_\beta K_{\alpha\beta}\mathbf{u}_\beta}{\rho_\alpha \mathbf{A}_\alpha}, \tag{22}$$

which is used to derive the face-flux equations, as will be detailed in the next section.

3.3.2. Pressure equation and face flux update

The pressure equation is derived by enforcing the global continuity. The global continuity equation can be expressed as:

$$\nabla\cdot\left(\sum_{\alpha=1}^{P} r_\alpha \mathbf{u}_\alpha\right) = 0 \tag{23}$$

Applying the finite-volume discretization to the divergence operator in Eq. (23), it is rewritten as:

$$\nabla\cdot\left(\sum_{\alpha=1}^{P} r_\alpha \mathbf{u}_\alpha\right) = \sum_f\left(\sum_{\alpha=1}^{P} r_{\alpha f}\mathbf{u}_{\alpha f}\cdot\mathbf{S}_f\right) = \nabla_{\mathbf{D}}\cdot\left(\sum_{\alpha=1}^{P}(r_\alpha)_f\phi_\alpha\right), \tag{24}$$

where $\phi_\alpha = \mathbf{u}_{\alpha f}\cdot\mathbf{S}_f$ is the volumetric face flux, the subscript f indicates that a variable is evaluated at a face of the control volume, whereas $()_f$ represents the face value approximated by an interpolation scheme using the neighbor cell-centered values. The operator $(\nabla_{\mathbf{D}}\cdot)$ represents the discretized divergent operator, being defined for notational convenience.

The face fluxes are obtained using the Rhie and Chow [27] interpolation, rather than by linear interpolation of the velocity predicted by Eq. (22). This enhances the pressure–velocity coupling, since the pressure gradient at the cell faces is calculated based on a reduced stencil using the cell centered values of the pressure at the neighbor control volumes, instead of interpolating the cell-centered pressure gradient values. This reduced stencil formulation for discretization of the gradient at the face centers is written here as ∇_f. Then, the volumetric face fluxes are calculated assuming the following equation:

$$\phi_\alpha = \hat{\phi}_\alpha + \phi_\alpha^* + \phi_\alpha^{\mathrm{ddtCorr}} + \frac{1}{\rho_\alpha(\mathbf{A}_\alpha)_f}\nabla_f p^*\cdot\mathbf{S}_f \tag{25}$$

$$\hat{\phi}_\alpha = \left(\frac{\mathbf{H}_\alpha}{\mathbf{A}_\alpha}\right)_f \tag{26}$$

$$\phi_\alpha^* = -\frac{(\mathbf{g}\cdot\mathbf{x})\nabla_f\rho_m}{\rho_\alpha(\mathbf{A}_\alpha)_f}\cdot\mathbf{S}_f + \left[\left(1-\frac{\rho_m}{\rho_\alpha}\right)_f\frac{\mathbf{g}}{(\mathbf{A}_\alpha)_f}\right]\cdot\mathbf{S}_f + \sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}\frac{(r_\beta)_f(K_{\alpha\beta})_f}{\rho_\alpha(\mathbf{A}_\alpha)_f}\phi_\beta \tag{27}$$

The $\phi_\alpha^{\mathrm{ddtCorr}}$ term is a correction of the volumetric face fluxes associated to the interpolation errors in the explicit part of the discretization of the transient term, similar to the one developed by Choi [2] but with an additional empirical factor. The form of the temporal correction term depends on the time integration scheme employed [4], but for the first-order Euler implicit method it is given as:

$$\phi_\alpha^{\mathrm{ddtCorr}} = \frac{\gamma}{\Delta t}\frac{1}{(\mathbf{A}_\alpha)_f}\left[\phi_\alpha^{t-1} - \left(\mathbf{u}_\alpha^{t-1}\right)_f\cdot\mathbf{S}_f\right] \tag{28}$$

where the empirical factor γ is defined as:

$$\gamma = 1 - \min\left(\frac{|\phi^{t-1} - \left(\mathbf{u}_\alpha^{t-1}\right)_f\cdot\mathbf{S}_f|}{|\phi^{t-1}| + \delta}, 1\right) \tag{29}$$

where the superscript $t-1$ stands for the value of the variable at a previous timestep. Substitution of Eq. (25) into Eq. (24) provides the following pressure equation:

$$\lfloor\nabla_{\mathbf{D}}\cdot\left(D_P\nabla_f[p^*]\cdot\mathbf{S}_f\right)\rfloor = \nabla_{\mathbf{D}}\cdot\left[\sum_{\alpha=1}^{P} r_\alpha\left(\hat{\phi}_\alpha + \phi_\alpha^* + \phi_\alpha^{\mathrm{ddtCorr}}\right)\right] \tag{30}$$

where

$$D_P = \sum_{\alpha=1}^{P}\frac{(r_\alpha)_f}{\rho_\alpha(\mathbf{A}_\alpha)_f} \tag{31}$$

After the solution of the pressure equation, the volumetric fluxes are calculated with Eq. (25) using the updated pressure field, and the cell centered velocities are explicitly updated as:

$$\mathbf{u}_\alpha = \frac{\mathbf{H}_\alpha}{\mathbf{A}_\alpha} + \left[\phi_\alpha^* + \frac{1}{\rho_\alpha(\mathbf{A}_\alpha)_f}\nabla_f p^*\cdot\mathbf{S}_f\right]_{f\to c} \tag{32}$$

where the subscript $f\to c$ denotes a vector field reconstruction at the cell centers from face flux values. A solution algorithm based on the PISO approach of Issa [16] was applied for the pressure–velocity coupling. Further details of the solution algorithm are given in Section 4.5.

3.4. Implicit coupling scheme

In the following we describe the numerical methodology developed for the multiphase implicitly coupled (MIC) solver.

3.4.1. Momentum equations

For the multiphase system with P phases, we had to extend the CMI developed by Cubero et al. [5]. First, we separate the temporal and drag coefficients from the $\mathbf{A}_\alpha$ and $\mathbf{H}_\alpha$ operators of the semi-discretized momentum equation (Eq. (21)), obtaining the following equation:

$$[\mathbf{A}_\alpha + \mathbf{A}_{T\alpha} + \mathbf{A}_{D\alpha}]\mathbf{u}_\alpha - \mathbf{H}_\alpha - \mathbf{H}_{T\alpha} = -\frac{\nabla p^*}{\rho_\alpha} + \mathbf{F}_{B\alpha}^* + \sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}\mathbf{A}_{D\alpha\beta}\mathbf{u}_\beta \tag{33}$$

where the total drag coefficients $\mathbf{A}_{D\alpha}$ and the other phases drag $\mathbf{A}_{D\alpha\beta}$ are defined by:

$$\mathbf{A}_{D\alpha} = \sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}\mathbf{A}_{D\alpha\beta}, \quad \mathbf{A}_{D\alpha\beta} = \frac{r_\beta K_{\alpha\beta}}{\rho_\alpha} \tag{34}$$

Considering a first-order Euler scheme with a fixed time step, the explicit temporal coefficient is given by $\mathbf{H}_{T\alpha} = \mathbf{A}_{T\alpha}\mathbf{u}_\alpha^{t-1}$. Following Cubero et al. [5], we divide both sides of Eq. (33) by $\mathbf{A}_\alpha$ to obtain:

$$(1 + \Theta_{T\alpha} + \Theta_{D\alpha})\mathbf{u}_\alpha = \hat{\mathbf{u}}_\alpha + \sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}\Theta_{D\alpha\beta}\mathbf{u}_\beta + \Theta_{T\alpha}\mathbf{u}_\alpha^{t-1} + \frac{\mathbf{F}_{B\alpha}^*}{\mathbf{A}_\alpha} - \frac{\nabla p^*}{\rho_\alpha\mathbf{A}_\alpha} \tag{35}$$

where the pseudo-velocities $\hat{\mathbf{u}}_\alpha$ are defined by:

$$\hat{\mathbf{u}}_\alpha = \frac{\mathbf{H}_\alpha}{\mathbf{A}_\alpha}, \tag{36}$$

and the coefficient ratios, Θ, as:

$$\Theta_{T\alpha} = \frac{\mathbf{A}_{T\alpha}}{\mathbf{A}_\alpha}, \quad \Theta_{D\alpha} = \frac{\mathbf{A}_{D\alpha}}{\mathbf{A}_\alpha}, \quad \Theta_{D\alpha\beta} = \frac{\mathbf{A}_{D\alpha\beta}}{\mathbf{A}_\alpha} \tag{37}$$

Then, the following approximation for the velocities is obtained:

$$\mathbf{u}_\alpha = \frac{1}{1+\Theta_{T\alpha}+\Theta_{D\alpha}}\left[\hat{\mathbf{u}}_\alpha + \sum_{\substack{\beta=1\\ \beta\neq\alpha}}^{P}\Theta_{D\alpha\beta}\mathbf{u}_\beta + \Theta_{T\alpha}\mathbf{u}_\alpha^{t-1} + \frac{\mathbf{F}^*_{B\alpha}}{\mathbf{A}_\alpha} - \frac{\nabla p^*}{\rho_\alpha \mathbf{A}_\alpha}\right] \tag{38}$$

3.4.2. Momentum interpolation

Following Cubero and Fueyo [4] and Cubero et al. [5], we define the velocities at the cell faces $\mathbf{u}_{\alpha,f}$ by:

$$\mathbf{u}_{\alpha,f} = (\mathbf{u}_\alpha)_f + \langle \mathbf{u}_\alpha \rangle \tag{39}$$

where $(\mathbf{u}_\alpha)_f$ is the linearly interpolated velocity at the faces and $\langle \mathbf{u}_\alpha \rangle$ is the velocity correction term. Depending on how the momentum equation is written, different correction terms can be derived. The derivation of a correction term is obtained by rewriting Eq. (39) as:

$$\langle \mathbf{u}_\alpha \rangle = \mathbf{u}_{\alpha,f} - (\mathbf{u}_\alpha)_f, \tag{40}$$

Then, Eq. (38) can be substituted into the above equation to yield:

$$\begin{aligned}\langle \mathbf{u}_\alpha \rangle = {} & \frac{\hat{\mathbf{u}}_{\alpha,f}}{1+\Theta_{T\alpha,f}+\Theta_{D\alpha,f}} - \left(\frac{\hat{\mathbf{u}}_\alpha}{1+\Theta_{T\alpha}+\Theta_{D\alpha}}\right)_f \\ & + \frac{\sum_{\substack{\beta=1\\ \beta\neq\alpha}}^{P}\Theta_{D\alpha\beta,f}\mathbf{u}_{\beta,f}}{1+\Theta_{T\alpha,f}+\Theta_{D\alpha,f}} - \left(\frac{\sum_{\substack{\beta=1\\ \beta\neq\alpha}}^{P}\Theta_{D\alpha\beta}\mathbf{u}_\beta}{1+\Theta_{T\alpha}+\Theta_{D\alpha}}\right)_f \\ & + \frac{\Theta_{T\alpha,f}\mathbf{u}^{t-1}_{\alpha,f}}{1+\Theta_{T\alpha,f}+\Theta_{D\alpha,f}} - \left(\frac{\Theta_{T\alpha}\mathbf{u}^{t-1}_{\alpha}}{1+\Theta_{T\alpha}+\Theta_{D\alpha}}\right)_f \\ & + \frac{\mathbf{F}^*_{B\alpha,f}}{(1+\Theta_{T\alpha,f}+\Theta_{D\alpha,f})\mathbf{A}_{\alpha,f}} - \left(\frac{\mathbf{F}^*_{B\alpha}}{(1+\Theta_{T\alpha}+\Theta_{D\alpha})\mathbf{A}_{\alpha}}\right)_f \\ & - \frac{\nabla p^*_f}{(1+\Theta_{T\alpha,f}+\Theta_{D\alpha,f})\rho_\alpha \mathbf{A}_{\alpha,f}} + \left(\frac{\nabla p^*}{(1+\Theta_{T\alpha}+\Theta_{D\alpha})\rho_\alpha \mathbf{A}_{\alpha}}\right)_f\end{aligned} \tag{41}$$

which is exact and could have been directly applied. However, we applied several approximations in the above equation. As there is no better approximation for the face value of many of the variables in Eq. (41) than the linear interpolation of the cell-centered values, this was assumed for these terms. Hence, the cell face value of the numerical coefficients $\Theta_{T\alpha,f}$, $\Theta_{D\alpha,f}$, $\Theta_{D\alpha\beta,f}$ and $\mathbf{A}_{\alpha,f}$ are approximated as:

$$\begin{aligned}&\Theta_{T\alpha,f} = (\Theta_{T\alpha})_f; \quad \Theta_{D\alpha,f} = (\Theta_{D\alpha})_f; \quad \Theta_{D\alpha\beta,f} = (\Theta_{D\alpha\beta})_f; \\ &\mathbf{A}_{\alpha,f} = (\mathbf{A}_\alpha)_f\end{aligned} \tag{42}$$

The values of the face velocities $\mathbf{u}_{\alpha,f}$ are not actually needed because, in the derivation of pressure equation, only its dot product with the face area vector, $\mathbf{S}_f$, which defines the volumetric face flux, ϕ_α, is necessary. The face pressure and density gradients are calculated from the cell-centered values of the volumes that share the face. Then, $\nabla p^*_f = \nabla_f p^*$ and the modified body force at the faces $\mathbf{F}^*_{B\alpha,f}$ is calculated by:

$$\mathbf{F}^*_{B\alpha,f} = -\frac{1}{\rho_\alpha}(\mathbf{g}\cdot\mathbf{x})_f \nabla_f \rho_m + \left(1-\frac{\rho_m}{\rho_\alpha}\right)_f \mathbf{g}, \tag{43}$$

Besides that, in order to save computational time, it is very common in CFD codes to consider that $(\prod_i \psi_i)_f = \prod_i (\psi_i)_f$. This simplification introduces second order errors that are of little consequence for discretization schemes up to second order accurate. Since it simplifies code implementation and speeds up calculations, it was used in most of the correction terms.

Applying the above simplifications, the correction for the pseudo-velocities, $\hat{\mathbf{u}}_\alpha$, vanishes and Eq. (41) simplifies to:

$$\langle \mathbf{u}_\alpha \rangle = \langle \mathbf{u}_\alpha \rangle_D + \langle \mathbf{u}_\alpha \rangle_T + \langle \mathbf{u}_\alpha \rangle_{F_B} + \langle \mathbf{u}_\alpha \rangle_{\nabla p^*} \tag{44}$$

where $\langle \mathbf{u}_\alpha \rangle_D$, $\langle \mathbf{u}_\alpha \rangle_T$, $\langle \mathbf{u}_\alpha \rangle_{F_B}$ and $\langle \mathbf{u}_\alpha \rangle_{\nabla p^*}$ are the drag, temporal, body force and pressure corrections, respectively, which are given by:

$$\langle \mathbf{u}_\alpha \rangle_D = \frac{\sum_{\substack{\beta=1\\ \beta\neq\alpha}}^{P}(\Theta_{D\alpha\beta})_f\left[\mathbf{u}_{\beta,f} - (\mathbf{u}_\beta)_f\right]}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f} \tag{45}$$

$$\langle \mathbf{u}_\alpha \rangle_T = \frac{(\Theta_{T\alpha})_f\left[\mathbf{u}^{t-1}_{\alpha,f} - (\mathbf{u}^{t-1}_{\alpha})_f\right]}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f} \tag{46}$$

$$\langle \mathbf{u}_\alpha \rangle_{F_B} = \frac{\mathbf{F}^*_{B\alpha,f} - (\mathbf{F}^*_{B\alpha})_f}{\left[1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f\right](\mathbf{A}_\alpha)_f} \tag{47}$$

$$\langle \mathbf{u}_\alpha \rangle_{\nabla p^*} = \frac{(\nabla p^*)_f - \nabla_f p^*}{\left[1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f\right]\rho_\alpha(\mathbf{A}_\alpha)_f} \tag{48}$$

3.4.3. Pressure equation

In order to derive an implicitly coupled scheme, we use the velocity correction deduced in Eq. (44) to evaluate the volumetric face fluxes in the global continuity equation (Eq. (24)). Then, the volumetric face fluxes are written as:

$$\phi_\alpha = \left[(\mathbf{u}_\alpha)_f + \langle \mathbf{u}_\alpha \rangle\right]\cdot\mathbf{S}_f \tag{49}$$

The above relation is substituted in Eq. (24) to yield:

$$\nabla_{\mathbf{D}}\cdot\left(\sum_{\alpha=1}^{P}(r_\alpha)_f\left[(\mathbf{u}_\alpha)_f + \langle \mathbf{u}_\alpha \rangle\right]\cdot\mathbf{S}_f\right) = 0 \tag{50}$$

which, using Eqs. (44)–(48), gives the following equation for the pressure:

$$\begin{aligned}\nabla_{\mathbf{D}}\cdot\left[D_P\nabla_f p^*\cdot\mathbf{S}_f\right] = {} & \nabla_{\mathbf{D}}\cdot\left(\sum_{\alpha=1}^{P}(r_\alpha)_f(\mathbf{u}_\alpha)_f\cdot\mathbf{S}_f\right) \\ & + \nabla_{\mathbf{D}}\cdot\left[D_P(\nabla p^*)_f\cdot\mathbf{S}_f\right] \\ & + \nabla_{\mathbf{D}}\cdot\left[\sum_{\alpha=1}^{P}(r_\alpha)_f\left(\frac{\sum_{\substack{\beta=1\\ \beta\neq\alpha}}^{P}(\Theta_{D\alpha\beta})_f\left[\phi_\beta - (\mathbf{u}_\beta)_f\cdot\mathbf{S}_f\right]}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f}\right.\right. \\ & + \frac{(\Theta_{T\alpha})_f\left[\phi^{t-1}_\alpha - (\mathbf{u}^{t-1}_\alpha)_f\cdot\mathbf{S}_f\right]}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f} \\ & \left.\left. + \frac{\left[\mathbf{F}^*_{B\alpha,f} - (\mathbf{F}^*_{B\alpha})_f\right]\cdot\mathbf{S}_f}{\left[1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f\right](\mathbf{A}_\alpha)_f}\right)\right]\end{aligned} \tag{51}$$

where the pressure equation coefficient is given by:

$$D_P = \sum_{\alpha=1}^{P}\frac{(r_\alpha)_f}{\left[1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f\right]\rho_\alpha(\mathbf{A}_\alpha)_f}. \tag{52}$$

4. Implementation of the implicit coupling

4.1. Block coupled matrix structure

The finite volume discretization of a single transport equation results in a linear system of the following form:

$$\Lambda_A\mathbf{y} = \Lambda_B \tag{53}$$

where Λ_A is the coefficient matrix, $\mathbf{y}$ is a vector of unknowns representing the cell center values for the field that is being solved

for, and Λ_B is a vector of sources. The coefficient matrix is a $n \times n$ square matrix, while $\mathbf{y}$ and Λ_B both have dimension equal to the number of discretization volumes n:

$$\Lambda_A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}; \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix};$$
$$\Lambda_B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n, \end{bmatrix} \tag{54}$$

As the finite volume method formulations are usually based on local approximations using a small stencil of cells, matrix Λ_A is a sparse matrix, that is, most $a_{i,j}$ coefficients are equal to zero.

However, in a multi-variable block-coupled solution, there are d unknowns for each cell value. In this work, we performed the implicitly coupled solution of the momentum and pressure equations for a system composed of P phases. This means that our block coupled matrix has $(3P+1) \times (3P+1)$ coefficients for each cell, given by the three cartesian components of the phase velocities and the pressure equation ($d = 3P+1$).

For a two-phase flow, the variables are the phase velocity fields, $\mathbf{u}_a$ and $\mathbf{u}_b$ and the modified pressure, p^*. In the following, the matrix coefficient arrangement is illustrated for the two-phase flow case, where, for the i-index cell, we can write the sub-vector of unknowns, $\mathbf{y}_i$, the sub-vector of source terms, $\Lambda_{B,i}$, and the sub-matrix of coefficients that relates this cell with the j neighbor cell, $\Lambda_{A,i,j}$, as:

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{u}_{ai} \\ \mathbf{u}_{bi} \\ p^* \end{bmatrix}, \quad \Lambda_{B,i} = \begin{bmatrix} b_i^{\mathbf{u}_a} \\ b_i^{\mathbf{u}_b} \\ b_i^{p^*} \end{bmatrix}, \quad \Lambda_{A,i,j} = \begin{bmatrix} a_{i,j}^{\mathbf{u}_a,\mathbf{u}_a} & a_{i,j}^{\mathbf{u}_a,\mathbf{u}_b} & a_{i,j}^{\mathbf{u}_a,p^*} \\ a_{i,j}^{\mathbf{u}_b,\mathbf{u}_a} & a_{i,j}^{\mathbf{u}_b,\mathbf{u}_b} & a_{i,j}^{\mathbf{u}_b,p^*} \\ a_{i,j}^{p^*,\mathbf{u}_a} & a_{i,j}^{p^*,\mathbf{u}_b} & a_{i,j}^{p^*,p^*} \end{bmatrix} \tag{55}$$

where

$$b_i^{\mathbf{u}_\alpha} = \begin{bmatrix} b_i^{u_{\alpha x}} \\ b_i^{u_{\alpha y}} \\ b_i^{u_{\alpha z}} \end{bmatrix}, \; a_{i,j}^{\mathbf{u}_\alpha,\mathbf{u}_\beta} = \begin{bmatrix} a_{i,j}^{u_{\alpha x},u_{\beta x}} & a_{i,j}^{u_{\alpha x},u_{\beta y}} & a_{i,j}^{u_{\alpha x},u_{\beta z}} \\ a_{i,j}^{u_{\alpha y},u_{\beta x}} & a_{i,j}^{u_{\alpha y},u_{\beta y}} & a_{i,j}^{u_{\alpha y},u_{\beta z}} \\ a_{i,j}^{u_{\alpha z},u_{\beta x}} & a_{i,j}^{u_{\alpha z},u_{\beta y}} & a_{i,j}^{u_{\alpha z},u_{\beta z}} \end{bmatrix},$$
$$a_{i,j}^{\mathbf{u}_\alpha,p^*} = \begin{bmatrix} a_{i,j}^{u_{\alpha x},p^*} \\ a_{i,j}^{u_{\alpha y},p^*} \\ a_{i,j}^{u_{\alpha z},p^*} \end{bmatrix}, \left(a_{i,j}^{p^*,\mathbf{u}_\alpha}\right)^T = \begin{bmatrix} a_{i,j}^{p^*,u_{\alpha x}} \\ a_{i,j}^{p^*,u_{\alpha y}} \\ a_{i,j}^{p^*,u_{\alpha z}} \end{bmatrix} \tag{56}$$

and $a_{i,j}^{\eta,\xi}$ is the coefficient submatrix that represents the influence of the variable ξ at cell j in the variable η at cell i.

The `foam-extend-4.0` version of OpenFOAM®has several facilities for the implementation of block-coupled solvers, which were used in this work to fill the block-coupled matrix with the appropriate coefficients [1,3,17]. The numerical treatment applied to each equation is described in the next section.

4.2. Implicit terms in the momentum equations

As shown in Section 3.1, we employed the phase intensive formulation of the momentum equation. The discretized version for a phase α is given by:

$$\left\lfloor \frac{\partial[\mathbf{u}_\alpha]}{\partial t} \right\rfloor + \lfloor \nabla \cdot (\mathbf{u}_\alpha[\mathbf{u}_\alpha]) \rfloor - \lfloor \mathbf{u}_\alpha (\nabla \cdot [\mathbf{u}_\alpha]) \rfloor$$
$$- \left\lfloor \nabla \cdot \left(\nu_\alpha \frac{\nabla r_\alpha}{r_\alpha + \delta} [\mathbf{u}_\alpha] \right) \right\rfloor - \lfloor \nabla \cdot (\nu_\alpha \nabla [\mathbf{u}_\alpha]) \rfloor$$
$$+ \left\lfloor \nabla \cdot \left(\nu_\alpha \frac{\nabla r_\alpha}{r_\alpha + \delta} \right) [\mathbf{u}_\alpha] \right\rfloor + \left\lfloor \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^{P} \frac{r_\beta K_{\alpha\beta}}{\rho_\alpha} [\mathbf{u}_\alpha] \right\rfloor$$
$$- \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^{P} \left\lfloor \frac{r_\beta K_{\alpha\beta}}{\rho_\alpha} [\mathbf{u}_\beta] \right\rfloor + \left\lfloor \frac{1}{\rho_\alpha} \nabla [p^*] \right\rfloor$$
$$= -\nabla \cdot \tau_\alpha^C + \frac{\nabla r_\alpha}{r_\alpha + \delta} \cdot \tau_\alpha^C + \mathbf{F}_{B\alpha}^* \tag{57}$$

where the terms inside brackets are treated implicitly, including the drag terms from the other phases and the modified pressure gradient. The terms at the right hand side of Eq. (57) are treated explicitly. During matrix filling, for a given phase α, the operators in $[\mathbf{u}_\alpha]$ feed values in $a_{i,j}^{\mathbf{u}_\alpha,\mathbf{u}_\alpha}$ and $b_i^{\mathbf{u}_\alpha}$, while the pressure gradient term stores information into $a_{i,j}^{\mathbf{u}_\alpha,p^*}$ and, depending on the discretization method applied to the pressure gradient, maybe into $b_i^{\mathbf{u}_\alpha}$. The drag term does not depend on neighbor cells and, therefore, feeds values only in the $a_{i,i}^{\mathbf{u}_\alpha,\mathbf{u}_\beta}$ coefficients. All the explicit terms are inserted into the source term $b_i^{\mathbf{u}_\alpha}$. For a multiphase system, the discretization of the phase momentum equations feeds the first $3P$ rows of the submatrices $\Lambda_{A,i,j}$. The last row is fed by the discretized pressure equation, which is described below.

4.3. Implicit terms in the pressure equation

In this work we follow the same approach of Cubero and Fueyo [4] and Darwish et al. [7] to develop a pressure equation for the implicit solution of the pressure–velocity coupling in incompressible flows, which consists of applying the velocity correction equations to the global continuity equation in order to obtain a Poisson-like equation for the pressure. A first step is the reorganization of the velocity divergence term and part of the drag term in Eq. (51), whose details are given in Appendix A. These terms are implicitly discretized. Then, the terms that depend on the volumetric fluxes or on the linear interpolation of the pressure gradient are treated explicitly considering the current iteration values of ϕ and p^*. Taking into account all these discretizations, Eq. (51) can be rewritten as:

$$- \left\lfloor \nabla_{\mathbf{D}} \cdot \left(D_P \nabla_f [p^*] \cdot \mathbf{S}_f \right) \right\rfloor$$
$$+ \sum_{\alpha=1}^{P} \left\lfloor \nabla_{\mathbf{D}} \cdot \left\{ \left[(r_\alpha)_f - \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^{P} \frac{(r_\beta)_f (\Theta_{D\beta\alpha})_f}{1 + (\Theta_{T\beta})_f + (\Theta_{D\beta})_f} \right] ([\mathbf{u}_\alpha])_f \cdot \mathbf{S}_f \right\} \right\rfloor$$
$$= -\nabla_{\mathbf{D}} \cdot \left[D_P (\nabla p^*)_f \cdot \mathbf{S}_f \right] + \Phi_D + \Phi_T + \Phi_{F_B} \tag{58}$$

where

$$\Phi_D = \nabla_{\mathbf{D}} \cdot \left[\sum_{\alpha=1}^{P} (r_\alpha)_f \frac{-\sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^{P} (\Theta_{D\alpha\beta})_f \phi_\beta}{1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f} \right] \tag{59}$$

$$\Phi_T = \nabla_{\mathbf{D}} \cdot \left\{ \sum_{\alpha=1}^{P} (r_\alpha)_f \frac{(\Theta_{T\alpha})_f \left[(\mathbf{u}_\alpha^{t-1})_f \cdot \mathbf{S}_f - \phi_\alpha^{t-1} \right]}{1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f} \right\} \tag{60}$$

$$\Phi_{F_B} = \nabla_{\mathbf{D}} \cdot \left\{ \sum_{\alpha=1}^{P} (r_\alpha)_f \frac{\left[(\mathbf{F}_{B\alpha}^*)_f - \mathbf{F}_{B\alpha,f}^* \right] \cdot \mathbf{S}_f}{\left[1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f \right] (\mathbf{A}_\alpha)_f} \right\} \tag{61}$$

In Eq. (58), the terms on the left hand side are discretized implicitly with respect to the pressure and phase velocities, respectively, whereas those on the right hand side are treated explicitly.

The operator in $[p^*]$ can be recognized as the Laplacian, and it defines the coefficients $a_{i,j}^{p^*,p^*}$ and feeds information into $b_i^{p^*}$. The divergence operator in $[\mathbf{u}_\alpha]$ sets up the coefficients $a_{i,j}^{p^*,\mathbf{u}_\alpha}$ and also feeds values in $b_i^{p^*}$. As usual, the explicit terms are inserted into the source $b_i^{p^*}$. Note that Eq. (51) was multiplied by -1 to provide Eq. (58). This was performed to force all the numerical coefficients in the main diagonal of the block-coupled matrix, $\Lambda_{A,i,j}$, to have the same sign, which is beneficial for the linear system solver.

4.4. Face flux correction equation

After solution of the block-coupled system to a given tolerance, we obtain the new values for $\mathbf{u}_\alpha$ and p^* fields, and the face fluxes ϕ are updated by combining Eqs. (44) and (49) into the following equation:

$$\begin{aligned}\phi_\alpha = (\mathbf{u}_\alpha)_f \cdot \mathbf{S}_f &+ \frac{\sum_{\substack{\beta=1 \\ \beta\neq\alpha}}^{P} (\Theta_{D\alpha\beta})_f \left[\phi_\beta^{k-1} - \left(\mathbf{u}_\beta^k\right)_f \cdot \mathbf{S}_f\right]}{1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f} \\ &+ \frac{(\Theta_{T\alpha})_f \left[\phi_\alpha^{t-1} - (\mathbf{u}_\alpha^{t-1})_f \cdot \mathbf{S}_f\right]}{1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f} \\ &+ \frac{\left[\mathbf{F}_{B\alpha,f}^* - (\mathbf{F}_{B\alpha}^*)_f\right] \cdot \mathbf{S}_f}{\left[1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f\right](\mathbf{A}_\alpha)_f} \\ &+ \frac{\left[(\nabla p^{*k-1})_f - \nabla_f p^{*k}\right] \cdot \mathbf{S}_f}{\left[1 + (\Theta_{T\alpha})_f + (\Theta_{D\alpha})_f\right]\rho_\alpha (\mathbf{A}_\alpha)_f}\end{aligned} \tag{62}$$

where the $k-1$ superscript refers to the variable value at the previous iteration level, being calculated using the same numerical values used in the assembly of the right hand side of the pressure equation (Eq. (58)), while the k superscript refers to the values at the current iteration level, which are obtained after solution of the block-coupled system.

4.5. Solution algorithms and convergence criteria

In order to carry out a fair performance analysis of the MS and MIC solvers, they both employed the same error criteria for the convergence of the solution within a time step. The convergence of the pressure–velocity coupling was evaluated through the maximum error of the pressure and velocity fields between successive inner or outer iterations using a mixed tolerance criterion given by $E_\chi^{\text{mixed}} < 1$, where E_χ^{mixed} is the mixed error that is defined by:

$$E_\chi^{\text{mixed}} = \max\left(\frac{|\chi^k - \chi^{k-1}|}{\lambda_\chi^{\text{abs}} + \lambda_\chi^{\text{rel}}|\chi^k|}\right) \tag{63}$$

where $\lambda_\chi^{\text{abs}}$ and $\lambda_\chi^{\text{rel}}$ are chosen values for the absolute and relative tolerances. The convergence of the phase fraction fields was evaluated using the absolute error between the iterations, E_χ^{abs}, which is defined by:

$$E_\chi^{\text{abs}} = \max\left(|\chi^k - \chi^{k-1}|\right) \tag{64}$$

Then, for each time step, the solution procedure applied for the MIC and MS solvers is the following:

1. Outer loop iterations: For a given maximum number of iterations, $N_{\text{outer}}^{\max}$, or until $E_{\chi,\text{outer}}^{\text{mixed}} < 1$ for the modified pressure and all components of the velocity fields and $E_{r_\alpha}^{\text{abs}} < \lambda_{r_\alpha,\text{outer}}^{\text{abs}}$, $\alpha = 1,\ldots,P$.
   (a) Pressure–velocity coupling iterations: For a given maximum number of iterations, $N_{pU,\text{inner}}^{\max}$, or until $E_{\chi,\text{inner}}^{\text{mixed}} < 1$ for the modified pressure and all components of the velocity fields.
   - For the MIC solver:
     (1) Update the drag coefficients, the momentum equation operators $\mathbf{A}_\alpha$ and $\mathbf{H}_\alpha$, the numerical coefficients defined in Eq. (37) and the explicit correction terms in Eqs. (59)–(61).
     (2) Assembly the block-coupled matrix for $p^* - \mathbf{u}_\alpha$ with the coefficients from the discretization of Eqs. (57) and (58).
     (3) Solve the block-coupled linear system for a given tolerance.
     (4) Calculate the face fluxes using Eq. (62).
   - For the MS solver:
     (1) Update the drag coefficients, the matrix operators $\mathbf{A}_\alpha$ and $\mathbf{H}_\alpha$ and assemble the pressure equation.
     (2) Update the pressure field solving the pressure equation (Eq. (30)).
     (3) Use Eqs. (25) and (32), to correct the volumetric fluxes and the velocities, respectively.
   (b) For a given maximum number of iterations, $N_{r_\alpha,\text{inner}}^{\max}$, or until $E_{r_\alpha,\text{inner}}^{\text{abs}} < \lambda_{r_\alpha,\text{inner}}^{\text{abs}}$, $\alpha = 1,\ldots,P$, solve the phase fraction equations (Eq. (18)).
2. Advance timestep.

In order to evaluate the convergence performance of both solvers, for each time step we stored the number of outer iterations N_{outer} and the total number of pressure–velocity inner iterations used by the algorithms, $N_{\text{pU,inner}}^{\text{tot}} = \sum_{i=1}^{N_{\text{outer}}} N_{\text{pU,inner},i}$.

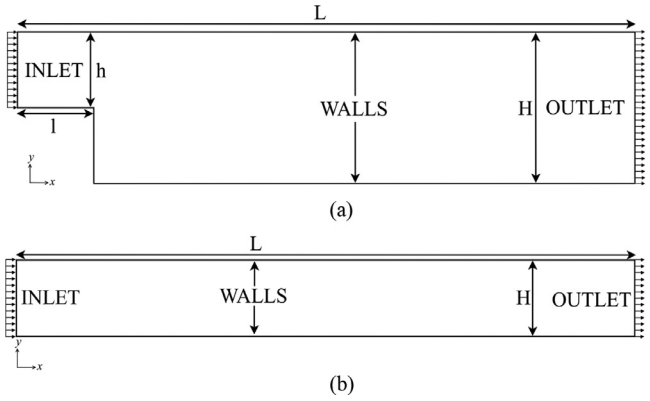


Fig. 1. Test cases: (a) backward-facing step geometry and (b) horizontal channel.

5. Simulation conditions

5.1. Test cases

The MIC and MS methods were compared in two-dimensional test cases considering two different geometries: a backward-facing step and a horizontal channel. The conditions considered in each test case are detailed below.

5.1.1. The backward-facing step

This simple geometry was chosen because it has a relatively complex flow pattern even in laminar flow conditions. The considered geometry, shown in Fig. 1(a), is the same tested by Silva and Lage [31] using $L = 11H$, $l = H$, $h = H/2$ and $H = 0.01$ m. Two groups of test cases, BFS1 and BFS2, were performed considering this geometry.

The BFS1 test cases were used to compare the results obtained using the two coupling methods (MS and MIC) with those obtained by the single phase steady-state implicitly coupled solver embedded in the `foam-extend` distribution, the `pUCoupledFoam` (pUC). These tests were performed in order to verify the temporal

Table 1
Phase properties considered in the horizontal channel cases.

Phases		Properties		
		ρ [kg m^{-3}]	ν [m^2/s]	d [m]
Continuous	c	1000	10^{-5}	10^{-4}
Dispersed	A	2000	10^{-4}	10^{-4}
	B	50	10^{-4}	10^{-4}
	C	2000	10^{-5}	10^{-4}
	D	2000	10^{-4}	10^{-5}

consistency of the momentum interpolation, as it is expected that the steady-state solution of a consistent transient method should be the same as the converged solution of a steady-state solver [4]. For this case we considered a parabolic velocity profile at the inlet with an average value of 0.5 m/s and a kinematic viscosity $\nu = 10^{-5}$ m^2/s, resulting in a Reynolds number of 500, and the gravitational force field was neglected. The multiphase solver was forced to simulate a single phase flow case considering a two-phase system with the same properties for both phases. The inlet phase fractions were 0.2 and 0.8 for the two phases, both with a diameter of 10^{-3} m. A mesh convergence analysis was performed considering uniform cartesian meshes with 4200 (M1), 8820 (M2) and 16,800 (M3) cells, for which (Δx, Δy) are equal to (5×10^{-4} m, 5×10^{-4} m), (3.33×10^{-4} m, 3.571×10^{-4} m) and (2.5×10^{-4} m, 2.5×10^{-4} m), respectively.

The BFS2 test cases were carried out to verify the extension of the numerical methodology to multiphase flows in both solvers (MS and MIC). These test cases considered a two-phase flow with a dispersed phase, a, and a continuous phase, b, with $r_a = 0.1$ and $r_b = 0.9$ at the inlet. Different densities were considered for the phases, with $\rho_b = 900$ kg/m^3 and $\rho_a = 1000$ kg/m^3. We also assumed that $r_a = 0$ at $t = 0$ for the whole domain, forcing density gradients to exist in the beginning of the simulation. The results of these two-phase simulations were compared to those from the flow simulation of a three-phase system with two identical dispersed phases, $a1$ and $a2$, with half the original phase fraction ($r_{a1} = r_{a2} = 0.05$ at the inlet). It should be noted that, in order to make both cases comparable, we had to neglect the drag between the dispersed phases $a1$ and $a2$. This analysis was carried out only for the coarsest mesh M1 in order to enlarge the deviations between the two numerical methodologies. The temporal consistency of the applied momentum interpolation methods was also investigated for this case by considering two different values for the Courant number (0.2 and 0.4).

5.1.2. The horizontal channel

The flow in a horizontal channel of phases with different densities leads to inhomogeneities in the phase fraction fields, being an interesting test case to compare the performance of the multiphase flow solvers. We compared the performance of the MIC and MS solvers considering continuous and dispersed phases with different properties, which are shown in Table 1. The continuous phase is referred below simply as phase c. The dispersed phase used as reference was phase A and the other dispersed phases were selected in order to verify the performance of the MIC and MS approaches for different values of density (B), viscosity (C) and diameter (D) of the dispersed phase.

The horizontal channel geometry consists of a rectangular two-dimensional channel with a length $L = 1.8$ m and a height $H = 0.025$ m and is shown in Fig. 1(b). Gravity was aligned with the y-axis pointing downwards with a magnitude of 9.8 m/s^2. A mesh convergence analysis was performed considering four $n_x \times n_y$ meshes, and the time steps were selected in order to keep the maximum Co number in the mesh below 0.3: mesh 1 (M1) with $n_x = 250$ and $n_y = 10$ (total of 2500 volumes, with $\Delta t = 0.01$ s), mesh 2 (M2) with $n_x = 500$ and $n_y = 20$ (total of 10,000 volumes, with $\Delta t = 0.005$ s), mesh 3 (M3) with $n_x = 1000$ and $n_y = 40$ (total of 40,000 volumes, with $\Delta t = 0.0025$ s) and mesh 4 (M4) with $n_x = 2000$ and $n_y = 80$ (total of 160,000 volumes, with $\Delta t = 0.00125$ s). A uniform velocity profile of 0.1 m/s was assumed at the inlet, resulting in a hydrodynamic residence time of 18 s. The Reynolds number calculated based on the properties of the continuous phase is 50. A total of five test cases were considered, whose inlet conditions are summarized in Table 2. The initial conditions for each case were generated from a 30 s simulation without considering the gravity force in order to develop the velocity and pressure fields. These preliminary simulations were started using uniform fields with values equal to the inlet conditions for the velocities and phase fractions and a zeroed value field for the modified pressure.

Table 2
Inlet conditions for the different test cases in the horizontal channel geometry.

Case	Dispersed phases	Phase fractions	u_{IN} [m/s]
HC1	A	$r_{dA} = 0.2$	
HC2	B	$r_{dB} = 0.2$	
HC3	C	$r_{dC} = 0.2$	
HC4	D	$r_{dD} = 0.2$	0.1
HC5	B, C, D	$r_{dB} = 0.1$ $r_{dC} = 0.05$ $r_{dD} = 0.05$	

5.2. Numerical procedure

In all the cases, the numerical discretization schemes were similar. The first-order implicit Euler method was used for temporal discretization. Laplacians, gradients and cell to face interpolations were calculated considering a linear approximation. The implicitly discretized velocity divergence in the pressure equation was also discretized considering a linear, second-order approximation. The upwind scheme was used for the momentum and phase fraction advection. When using the MIC solver, the block-coupled linear system was solved by clustering algebraic multigrid methods (AMG) [14,26]. Both the Block-Selective Algebraic Multigrid (SAMG) and aggregative (AAMG) methods were employed, being the former recently implemented by Uroić and Jasak [34] in `foam-extend`. The ILUC0 smoother described in [34] was applied for both methods. For the MS solver, a generalized AMG (GAMG) method with geometric pairing and a simplified diagonal-based Cholesky (DIC) smoother was applied on the solution of the pressure equation. The iterations of these linear solvers were controlled by specifying absolute and relative tolerances for the residual norm of the normalized linear system [18]. The relative tolerance is optional and simply defines the minimum ratio between the final and initial values of this residual norm. A relative tolerance of 10^{-2} and an absolute tolerance of 10^{-9} were employed for the implicitly coupled solver, whereas an absolute tolerance of 10^{-10} was used for the segregated solver. The phase fraction equations were solved using a preconditioned bi-conjugate gradient method (PBiCG [10]) with a simplified diagonal-based incomplete LU preconditioner [20] with an absolute tolerance of 10^{-12} for the normalized residual of the linear system.

A maximum of six external corrections ($N^{\max}_{\text{outer}} = 6$) and fifty pressure–velocity coupling corrections ($N^{\max}_{pU,\text{inner}} = 50$) were used for both the MIC and MS methods. The convergence of the pressure–velocity correction loop was controlled by setting absolute tolerance values of $\lambda^{\text{abs}}_{p^*,\text{inner}} = 4.5 \times 10^{-2}$ Pa and $\lambda^{\text{abs}}_{u_\alpha,\text{inner}} = 2 \times 10^{-6}$ m/s, and a relative tolerance of $\lambda^{\text{rel}}_{p^*,\text{inner}} = \lambda^{\text{rel}}_{u_\alpha,\text{inner}} = 10^{-5}$ for both variables. An absolute tolerance of $\lambda^{\text{abs}}_{r_\alpha,\text{inner}} = 10^{-5}$ was applied to the inner loops in the solution of the phase fraction fields.

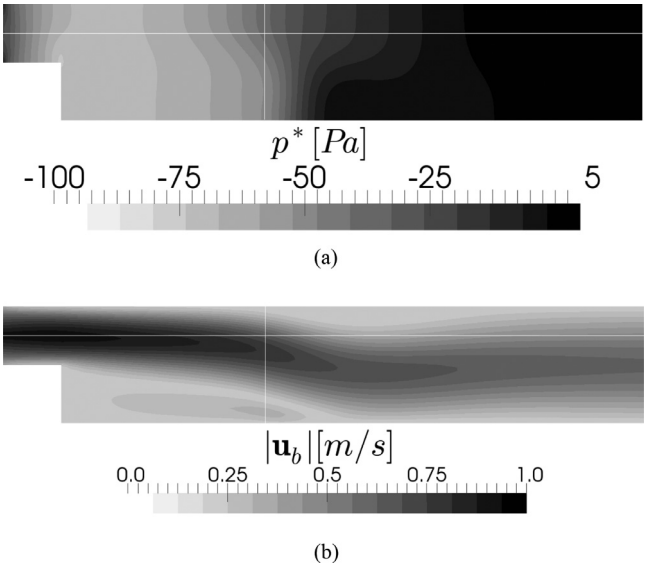


Fig. 2. Pressure (a) and velocity magnitude (b) fields for the flow over the backward-facing step and the lines used for sampling. The results shown are for the MIC solver with mesh M3, mimicking a single phase flow by considering two identical phases.

For the convergence of the outer loop the absolute tolerance values for pressure and velocity were relaxed to $\lambda^{abs}_{p^*,outer} = 4.5 \times 10^{-1}$ Pa and $\lambda^{abs}_{u_\alpha,outer} = 2 \times 10^{-5}$ m/s, respectively, a relative tolerance of $\lambda^{rel}_{p^*,outer} = \lambda^{rel}_{u_\alpha,outer} = 10^{-5}$ was used for both variables and an absolute tolerance of $\lambda^{abs}_{r_\alpha,outer} = 10^{-4}$ was employed for the phase fractions.

The simulations were run in an Intel Xeon X5675 with 32 physical cores running at 3.07 GHz CPUs and the CentOS 6.6 operating system with the Open MPI package [11] version 1.6.5 installed for the parallel runs. The parallelism efficiency, η, and the speedup with respect to the serial case, S, were calculated using the execution time, t^{comp}, of the master processor for each case, being given by:

$$S = \frac{t^{comp}_{serial}}{t^{comp}_{parallel}}; \qquad \eta = \frac{S}{Q} \tag{65}$$

where Q is the number of processes in the parallel run.

In order to detect the approach to a steady-state solution, a transient deviation $\epsilon_{t\chi}$ was defined by comparing the fields of a generic variable χ at two time instants separated by 100 time steps, that is:

$$\epsilon_{t\chi} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\chi_i(t) - \chi_i(t - 100\Delta t))^2} \tag{66}$$

where n is the number of control volumes.

6. Results

6.1. Flow over a backward-facing step

The steady-state results for the modified pressure and the continuous phase velocity magnitude fields are shown in Fig. 2, and were obtained with mesh M3 and MIC solver for the two-phase flow of two identical phases, mimicking the single phase flow (BFS1 test cases). Fig. 2 also shows the vertical and horizontal lines, located at $x = 0.045$ m from the inlet and at $y = 0.0075$ m from the bottom wall, respectively, used for sampling and comparing the results for the different meshes and numerical methodologies. Fig. 3 shows the steady-state values for the pressure at the horizontal and vertical lines and the x and y-component of the velocity at the vertical line for the BFS1 test cases using the three meshes. It shows that the results obtained with the MIC solver have a very good agreement with those generated by the pUC solver, with a visually exact superposition of the profiles. On the other hand, we observed that the MS algorithm show some deviations, which are attributed to the differences in the numerical formulation. As the mesh is refined, the difference between the methodologies also diminishes, suggesting that they should yield the same results on a sufficiently fine mesh.

Fig. 4 show the dispersed phase fraction, pressure and velocity profiles along the vertical line for the BFS2 cases. The profiles of the b-phase fraction at $t = 0.1$ s and $t = 0.4$ s are displayed in Fig. 4(a) and (b), respectively. They show that, despite the different results between the MS and MIC methodologies, both of them provide the same results when the two- and three-phase cases are compared, showing that the generalization of the numerical methodology to any number of phases was performed correctly. The steady-state profiles of pressure and y-component of the velocity are shown in Fig. 4(c) and (d), respectively. It can also be observed that the results obtained with the $a - b$ or $a1 - a2 - b$ systems were the same in both formulations. The influence of the time-step size on the steady-state solution obtained by the MS and MIC solvers was also verified for the BFS2 cases. Fig. 5(a) and (b) show, respectively, the steady-state profiles of the pressure and the y-component of the velocity on the vertical line for the simulations considering Courant number values of 0.2 and 0.4. As expected, the MS methodology failed in achieving time-step independent results since it does not apply a temporally consistent momentum interpolation technique. On the other hand, the MIC formulation provided the same results for the two employed Courant numbers.

6.2. Flow in a horizontal channel

First, the parallel scalability of both MIC and MS approaches were evaluated. Table 3 displays the execution time, the speedup and the parallelism efficiency considering the serial and parallel simulations divided in up to ten processors. This analysis was performed for case HC1 with M3 considering a transient simulation of 30 s. The comparison of the execution times spent with the AAMG and SAMG shows that the later is from 10 to 25 times faster than the former. As discussed by Uroić and Jasak [34], the convergence of the AAMG method stalls after few iterations. For this reason, the number of iterations performed by the linear solver were always equal to the maximum allowed (10) without achieving the prescribed tolerance, resulting in more pressure-velocities iterations in the inner loop to achieve convergence. On the other hand, the SAMG method is capable of reaching the prescribed tolerance in few iterations, resulting in less iterations also in the pressure-velocities coupling loop. The results for the MS solver are also shown in Table 3, revealing that the MS approach was from 5 to 10 times faster than the MIC with SAMG. On the other hand, the MIC method shows a better parallel scalability, as its efficiency is close to 68% for the case with 10 parallel processes, while the MS efficiency was below 35%. This suggests that the MIC solver with the SAMG linear solver may eventually become faster than MS solver for very large meshes.

In order to compare the results obtained with the MIC and MS methods, the HC test cases were run for a total of 500 s in order to achieve a steady-state solution. Due to the better performance of the SAMG linear solver when compared to the AAMG method, the former was used in the simulations with the MIC solver to generate the results given in the following.

The vertical lines used for sampling the velocity and phase fraction fields were located at $x = 1$ m and $x = 1.65$ m, respectively,

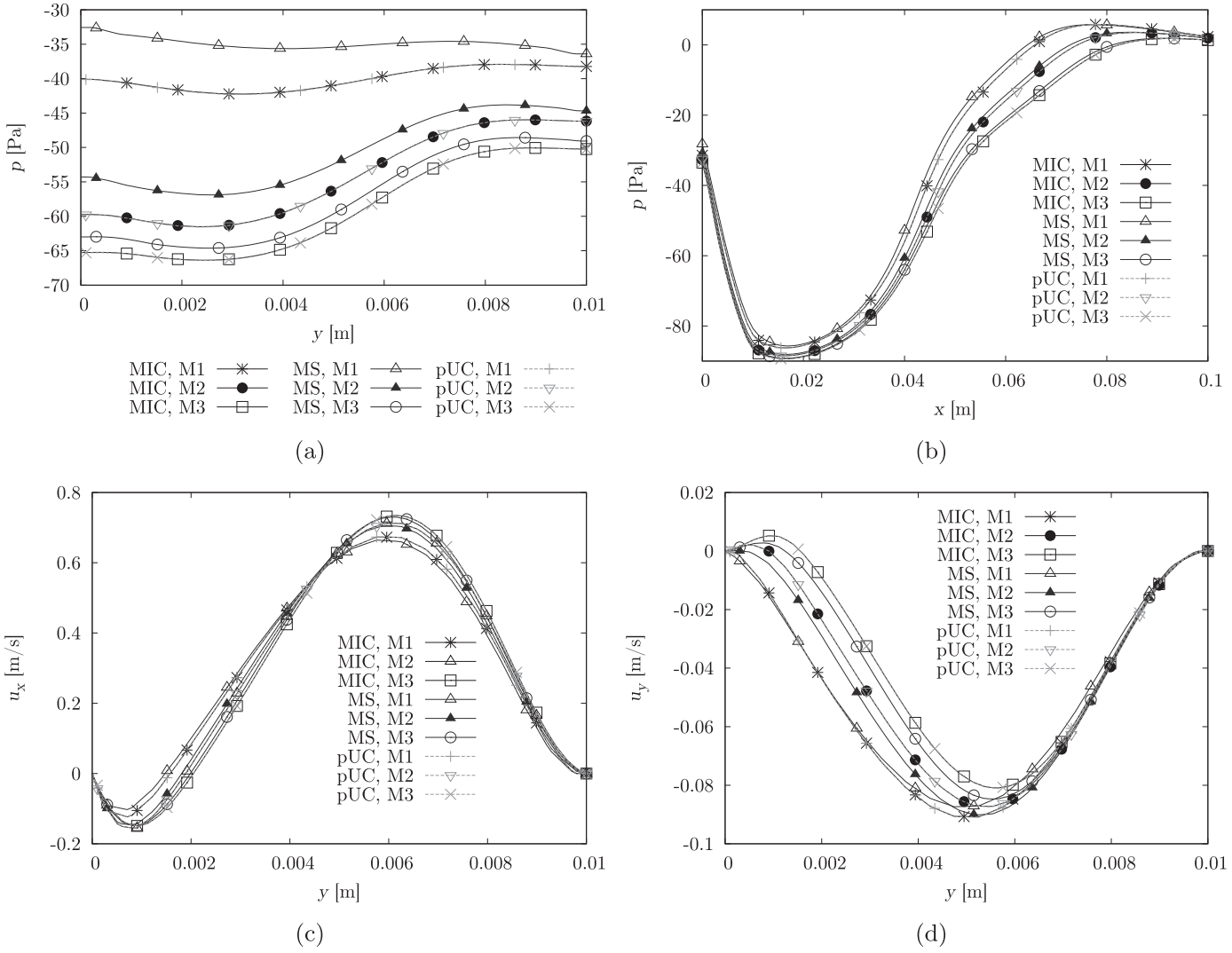


Fig. 3. Verification of the multiphase methodologies against the single phase steady-state coupled solver for several meshes: (a) the pressure profile along the vertical line, (b) the pressure profiles for the horizontal line, (c) the *x*- and (d) *y*-component velocity profiles along the vertical line.

Table 3
Comparison of the parallelism efficiency and computational times spent with the MIC using different solution methods and comparison with the MS solver for several number of processes.

Q	MIC						MS		
	AAMG			SAMG			GAMG		
	t^{comp} [s]	S	η	t^{comp} [s]	S	η	t^{comp} [s]	S	η
1	204,140	–	–	13,431	–	–	1434	–	–
2	146,540	1.4	69.7%	7169	1.9	93.7%	855	1.7	83.9%
4	98,677	2.1	51.7%	3673	3.7	91.4%	554	2.6	64.7%
8	55,403	3.7	46.1%	2286	5.9	73.4%	442	3.2	40.6%
10	48,850	4.2	41.8%	1980	6.8	67.8%	424	3.4	33.9%

and the horizontal line used for sampling the modified pressure was located at $y = 0.0125$ m, as displayed in Fig. 6. The results obtained in the case HC1 are shown in Fig. 7. The velocity and pressure profiles after 500 s of simulation are displayed in Fig. 7(a) and (b), respectively, and show that the results obtained with the MIC and MS solvers are similar regardless of the mesh spacing for this case. The dispersed phase fraction profile is shown in Fig. 7(c). It is also observed that the results obtained with the M3 mesh are very similar to those using the M4 mesh. Hence, given the large computational cost of the simulations with M4, M3 was the finest mesh used in the cases HC2, HC3, HC4 and HC5. As expected, Fig. 7(c) shows that the denser dispersed phase deposits at the bottom of the channel. As the mesh refinement is increased, the phase fraction gradients increase and the phase fraction at the vicinity of the bottom wall reaches a unitary value, while the phase fraction near the top wall approaches zero. Despite these steep phase fraction gradients, no convergence issues were observed for the MIC and MS approaches in the HC1 cases.

The maximum transient deviations in the velocity, modified pressure and phase fraction fields obtained with the M3 mesh are displayed in Fig. 7(d). In the first 100 s of simulation, the transient deviations of both methods have the same order of magnitude.

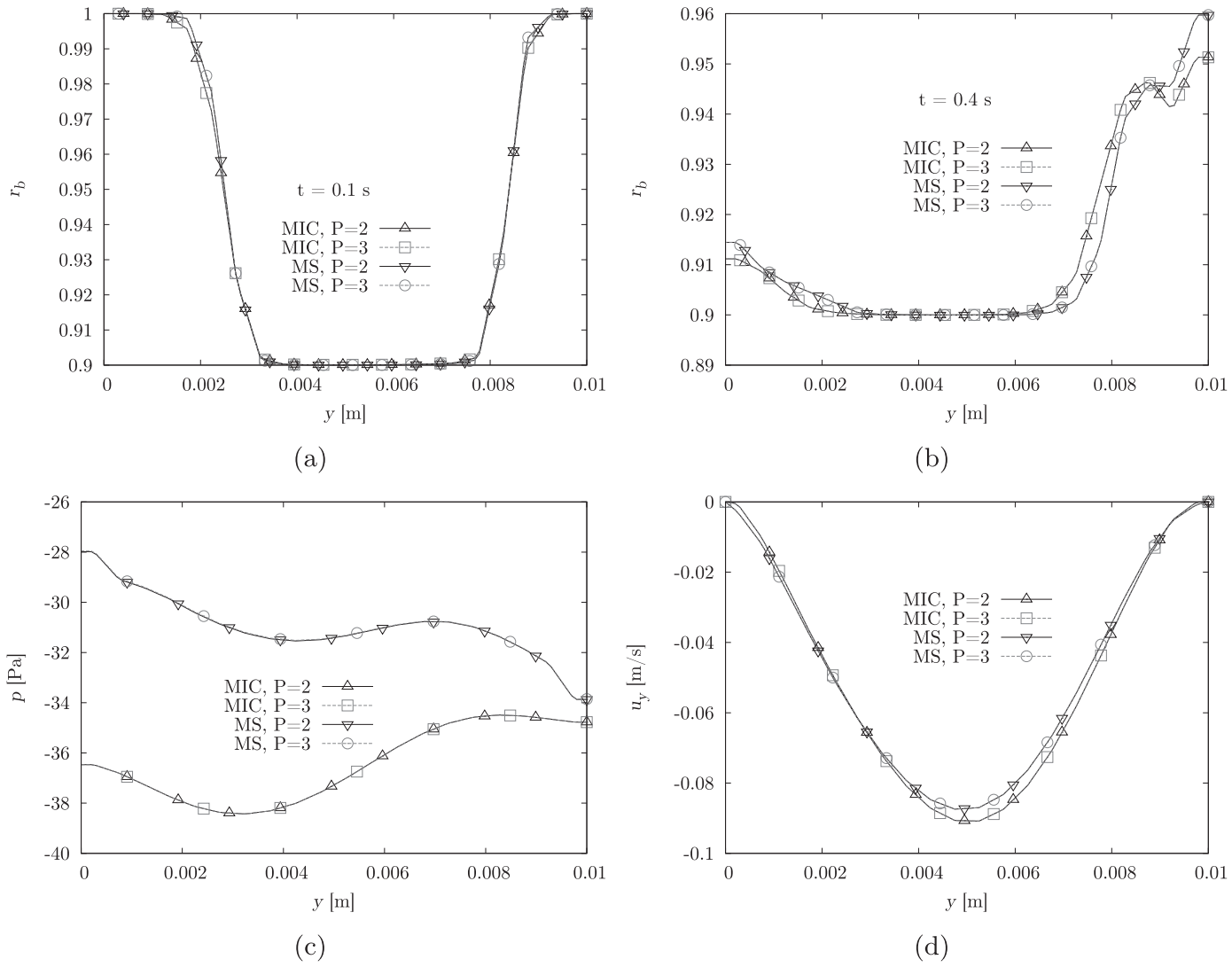


Fig. 4. Multiphase flow simulation considering different number of phases and numerical formulations (mesh M1): b-phase fraction at (a) $t = 0.1$ s and (b) $t = 0.4$ s and the steady state profiles for (c) the pressure and (d) the y-component of the velocity along the vertical line.

Then, after 100 s, rapid declines are noticed in the transient deviations of the velocity and phase fraction in the MIC solver, which become negligible after 200 s. This faster convergence to steady state may be credited to the better coupling between the phase velocities in the MIC solver. The number of inner and outer corrections performed by both solvers in each time step using the M3 mesh is shown in Fig. 7(e) and (f), respectively. During most of the simulation time both methods performed only one outer iteration per time step. Thus, these figures show only the beginning of the simulation, where the number of iterations performed by both methods were different. Except for the first time step, where the maximum allowed number of pressure–velocity coupling iterations (50) was employed, the MS method performed only one inner iteration per time step. On the other hand, the MIC solver executed several iterations in the first second of simulation. This explains the lower computational cost of the MS solver when compared to the MIC solver, as shown in Table 3. Not only the cost per iteration of the MS solver was lower, but also the number of inner iterations required to achieve the convergence within the time step was smaller. However, this is a case were both methods yielded a physically sound solution. There are simulation conditions where the MS solver fails and the MIC method must be employed, as those for the cases HC4 and HC5, which are analyzed below.

The test cases HC2 were designed to evaluate the performance of the MIC and MS methods when exists a large density ratio between the dispersed and continuous phases ($\rho_c/\rho_{dB} = 20$). The corresponding results are shown in Fig. 8. For this case, the MS solver with M2 diverged at $t = 97$ s and the steady state results could not be compared for this mesh. However, the velocity and pressure profiles shown in Fig. 8(a) and (b) reveal a good mesh convergence between M2 and M3 using MIC. The results obtained with the MIC and MS solvers were similar for M1 and M3.

The dispersed phase fraction profiles for the HC2 simulations are displayed in Fig. 8(c). As expected, the phase fraction of the lighter dispersed phase B is higher near the top of the channel. As in the previous case, the phase fraction gradient becomes steeper as the mesh is refined. Observing the transient deviations in Fig. 8(d) for the M3 mesh, it is noticed that both MIC and MS solvers achieved state state solutions for which the pressure fields still have some small fluctuations ($\epsilon_{tp*} \sim 10^{-2}$). These fluctuations are due to the large density gradients that emerge in this case. The comparisons in the number of inner and outer loop iterations executed by both solvers for the M3 mesh are shown in Fig. 8(e) and (f), respectively. They reveal that the convergence with the MIC for this case is more costly than in the base case HC1. Both MIC and MS methods still perform only one iteration per time step for most

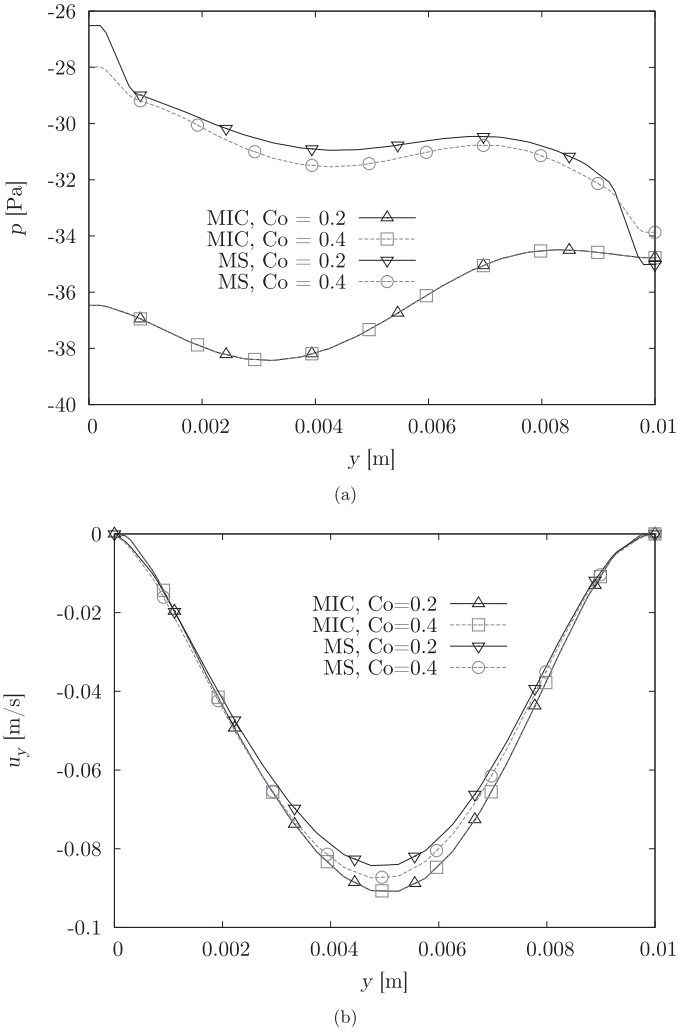


Fig. 5. Verification of the time-step dependency of the multiphase methodologies: profiles of (a) the pressure along the horizontal line and (b) the y-component of the velocity along the vertical line for different Courant numbers using the MS and MIC methodologies.

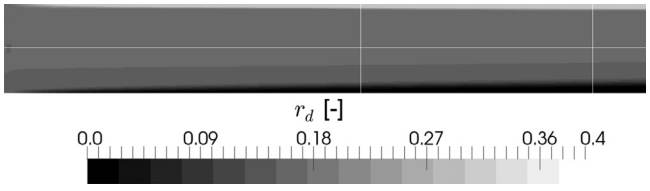


Fig. 6. Lines used for sampling the data in the horizontal case channel. The dispersed fraction field shown is for the HC2 case using the MIC solver.

of the simulated time, but MIC performs 3 inner iterations for almost 75 s of simulation, while MS performs more than one inner iteration only in the first 0.06 s of simulation.

The test case HC3 evaluates the effect of a different value for the dispersed phase viscosity in the performance of the two solvers. The obtained results are shown in Fig. 9. As in the previous case, the velocity and pressure profiles have already achieved mesh convergence for the M3 mesh, as shown in Fig. 9(a) and (b). The dispersed phase fraction profile, shown in Fig. 9(c), is very similar to that of case HC1, with the heavier dispersed phase C depositing at the bottom wall, and the phase fraction gradients increasing as the mesh is refined. However, the main difference in these two test cases is the behavior of the transient deviations, shown in Fig. 9(c) for the M3 mesh. When the steady state solution was achieved, the transient deviations using the MS solver are about five orders of magnitude larger than those using the MIC solver. This can be explained by the fact that a lower dispersed phase viscosity creates larger relative velocities in the streamwise direction and, thus, the continuous and dispersed phase velocity profiles have different development. The better coupling between the phases in the MIC solver yields a smoother convergence to the steady state solution. The number of inner and outer iterations are shown in Fig. 9(e) and (f), respectively, for the M3 mesh. As in the previous cases, the MS solver required more iterations in the first few time steps, while the MIC executed more inner iterations during the first 7 s of simulation.

Fig. 10 shows the results for test case HC4, which was designed for evaluating both solvers in the simulation of a dispersed phase with small particles (or drops). Fig. 10(a) and (b) show, respectively, the velocity and pressure profiles obtained from the MIC and MS methods. The results from the MIC method have mesh convergence but the results obtained from the MS method do not, which are very different for the three meshes. The MS method provides a velocity profile that tends to be flat instead of parabolic, as expected. This happened due to numerical limitations of the MS solver when dealing with the large drag coefficients that occur for this case, as a result of the small value of the diameter of the dispersed phase D. It must be pointed out that the tolerance criteria specified for controlling the inner and outer loop iterations were fullfilled for all cases in all time steps. Therefore, the MS solver convergence has stalled. The dispersed phase fraction profile is shown in Fig. 10(c), which shows that the phase fraction gradients are less steep than the previous cases due to the smaller settling velocity caused by the decrease in the diameter of the dispersed phase. As expected, the different velocity fields provided by the MS and MIC solvers led to different phase fraction profiles. Fig. 10(d) shows the transient deviations, which are up to eight orders of magnitude larger for the MS method results when compared to those obtained by the MIC method. However, the number of inner and outer iterations in both the MIC and MS solvers, shown in Fig. 10(e) and (f), respectively, are the same, that is, only one iteration per time step, except at the beginning of the simulations.

The results of the test case HC5 are shown in Figs. 11 and 12. These tests were performed in order to evaluate the MIC and MS solvers in a multiphase case with several dispersed phases including large density differences and small particle sizes. Fig. 11(a) and (b) show, respectively, the velocity and pressure profiles, which are different for the MIC and MS methods, similarly to the HC4 case. This is related to the presence of the small-diameter dispersed phase D and the numerical limitations of the MS solver to deal with large drag coefficients. However, for the HC5 case, the MS solver results get closer to those obtained with the MIC solver as the mesh is refined. This may be credited to the smaller fraction of phase D in this test case when compared to HC4 case. The phase fraction profiles of the continuous and dispersed phases C, D and E are shown in Fig. 11(c)–(f), respectively. The results for both methods agree with what was qualitatively expected, with the lighter dispersed phase B concentrating near the top wall of the channel whereas the heavier dispersed phase C concentrated at the bottom of the channel and the fraction of phase D followed the qualitative pattern of the continuous phase fraction because of the small settling rate of its particles.

The transient deviations for the M3 mesh are shown in Fig. 12(a). Their values for the MIC solver are seven orders of magnitude lower than those for the MS solver, indicating that, despite the long simulation time, the latter did not converged to the steady-state solution. The inner and outer iterations executed by the MIC and MS solvers are shown in Fig. 12(b) and (c), respec-

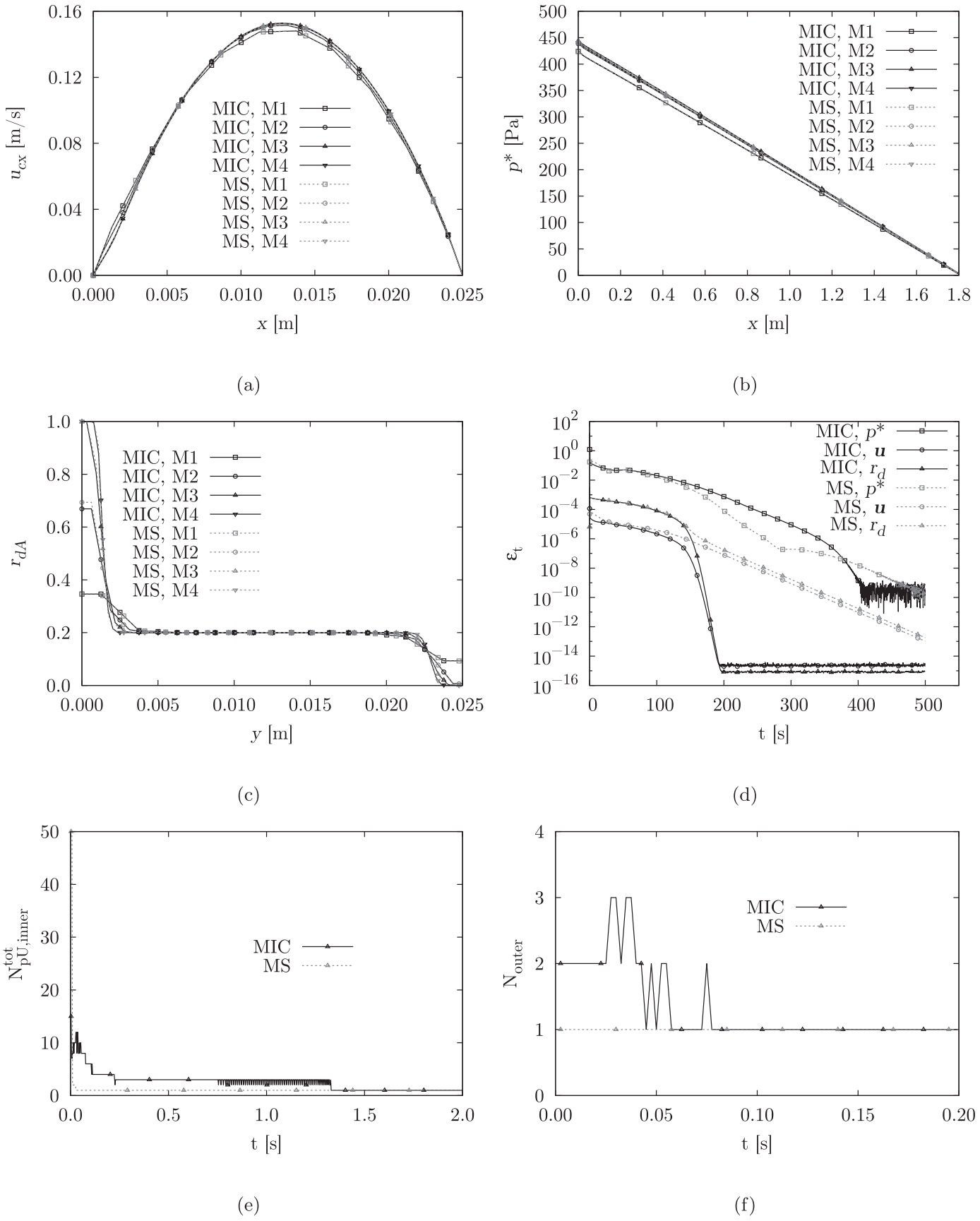
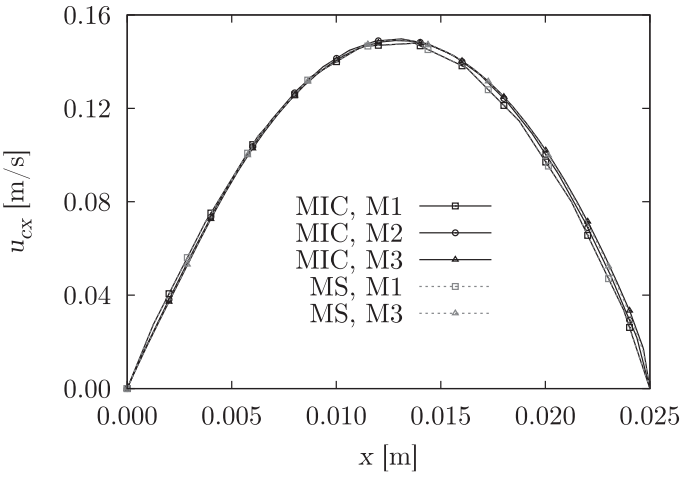
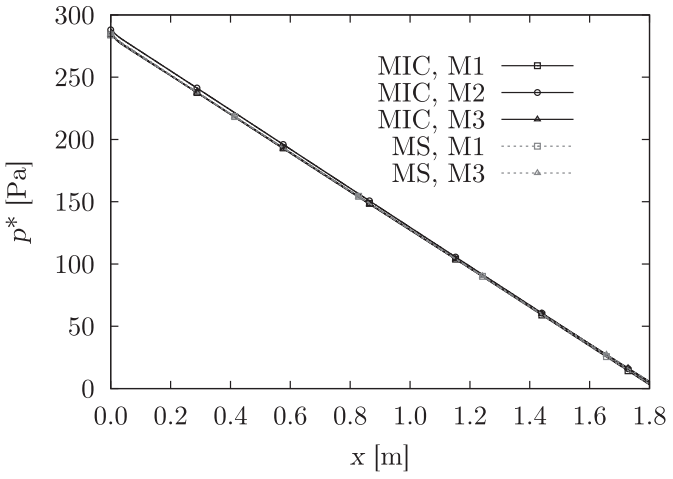


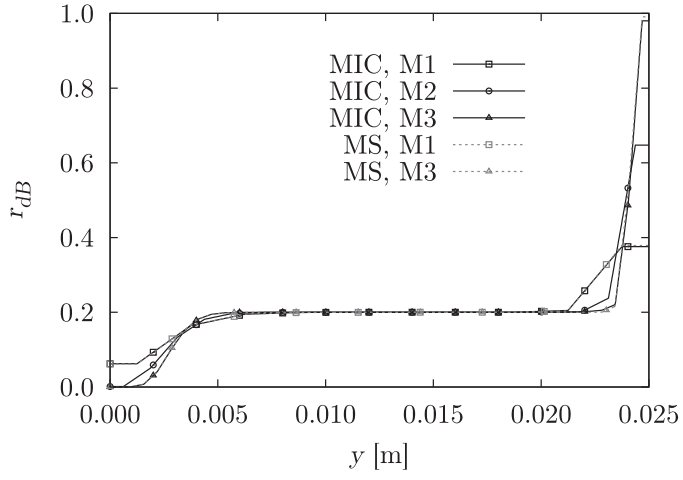
Fig. 7. Results for the case HC1: (a) continuous phase velocity, (b) modified pressure, (c) dispersed phase fraction, (d) transient deviations, (e) total number of inner iterations and (f) number of outer iterations within the timestep.



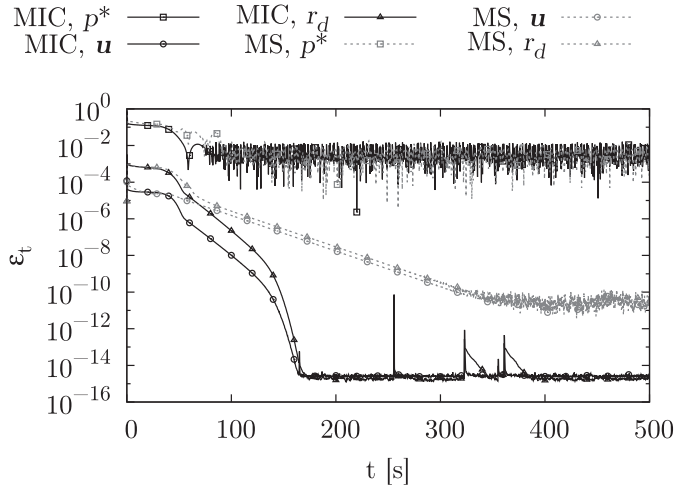
(a)



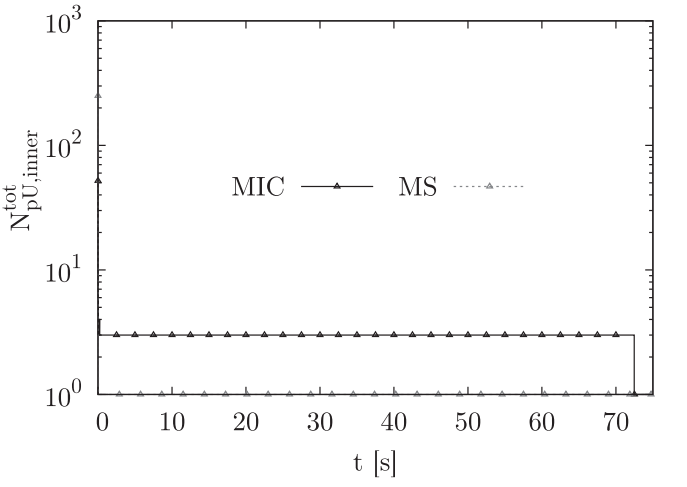
(b)



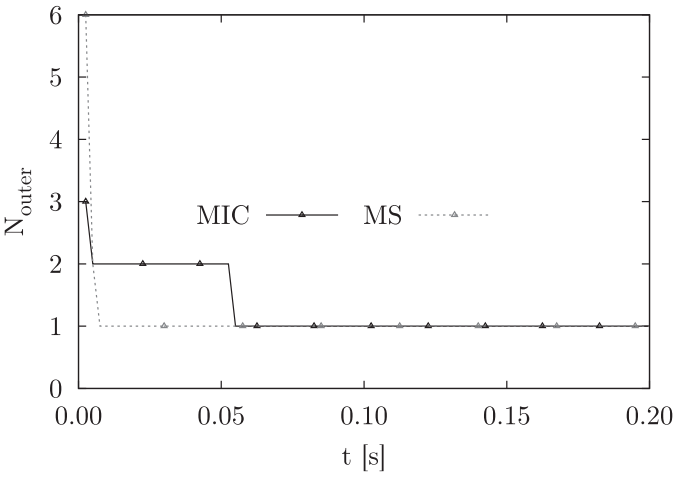
(c)



(d)

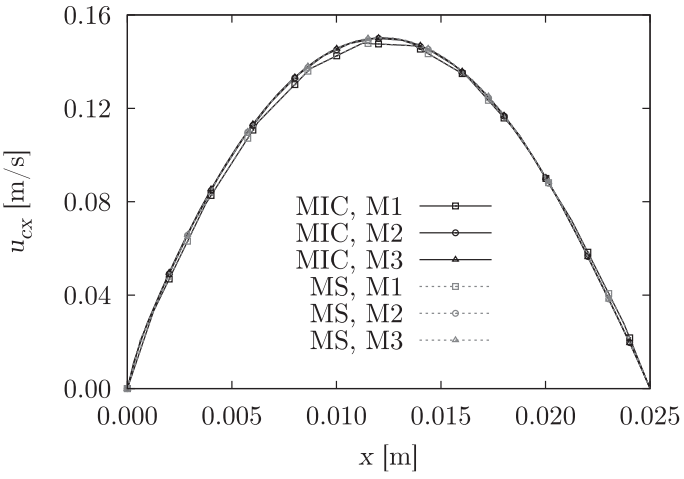


(e)

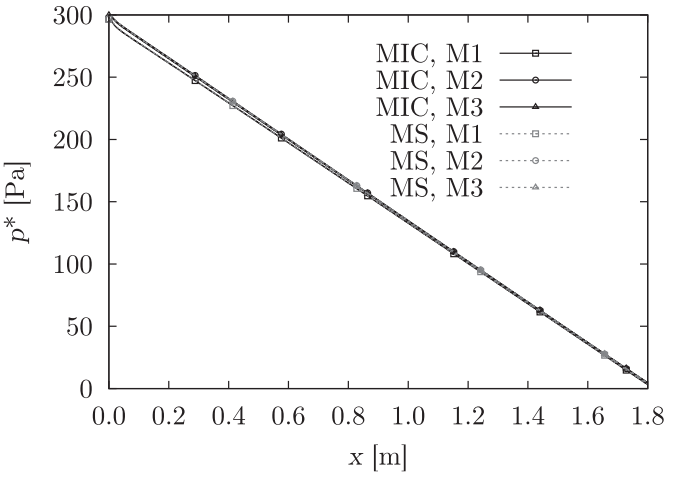


(f)

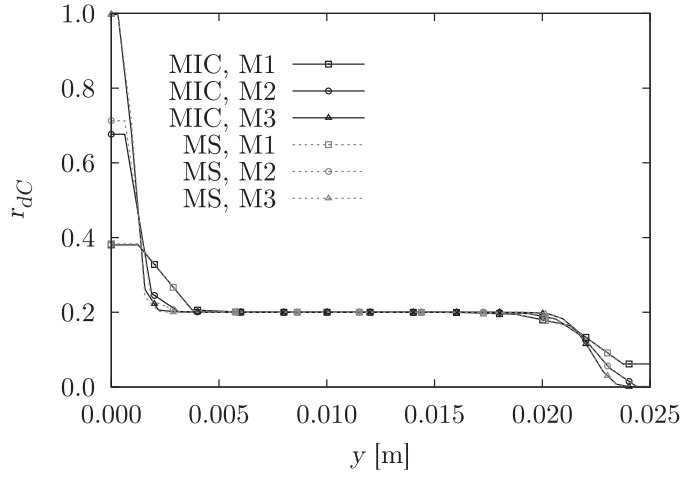
Fig. 8. Results for the case HC2: (a) continuous phase velocity, (b) modified pressure, (c) dispersed phase fraction, (d) transient deviations, (e) total number of inner iterations and (f) number of outer iterations within the timestep.



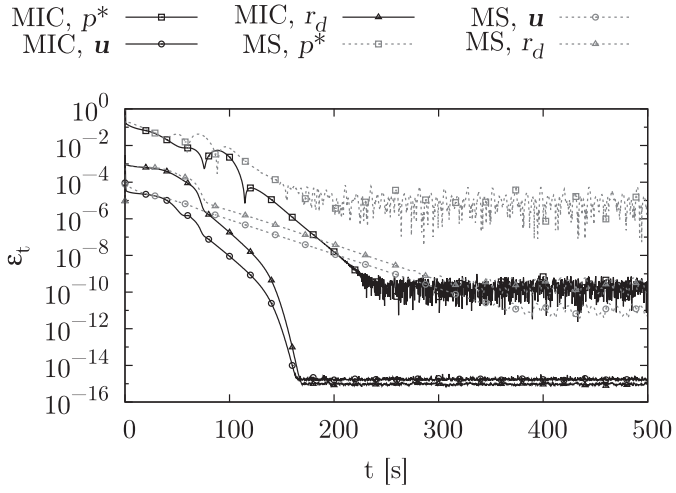
(a)



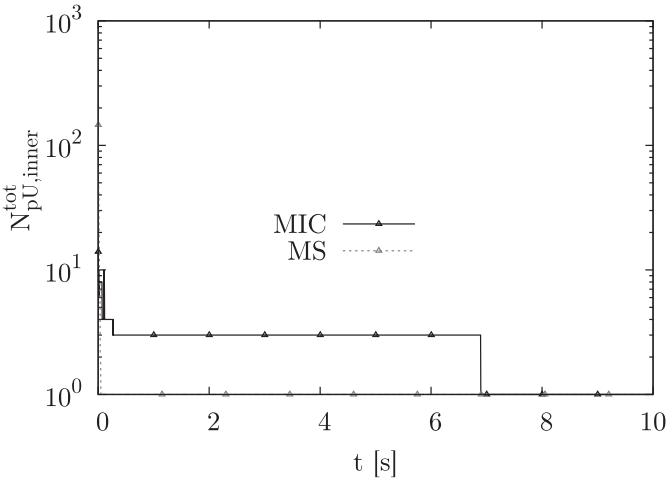
(b)



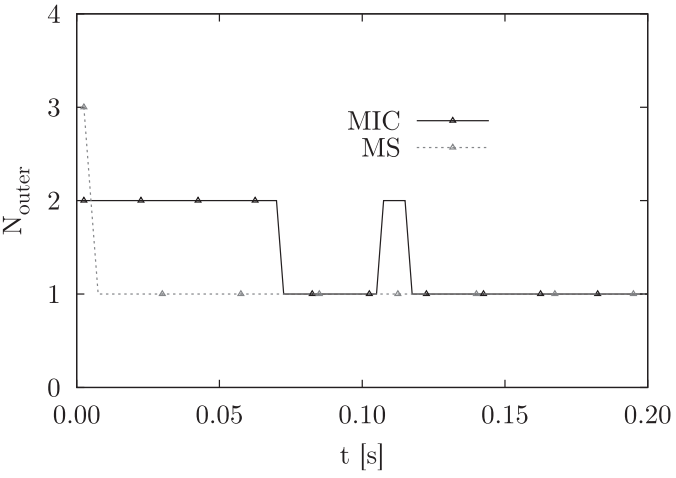
(c)



(d)

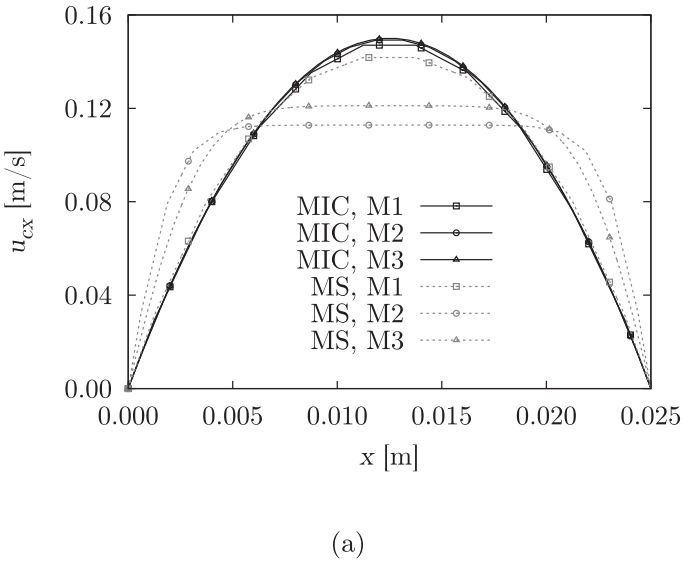


(e)

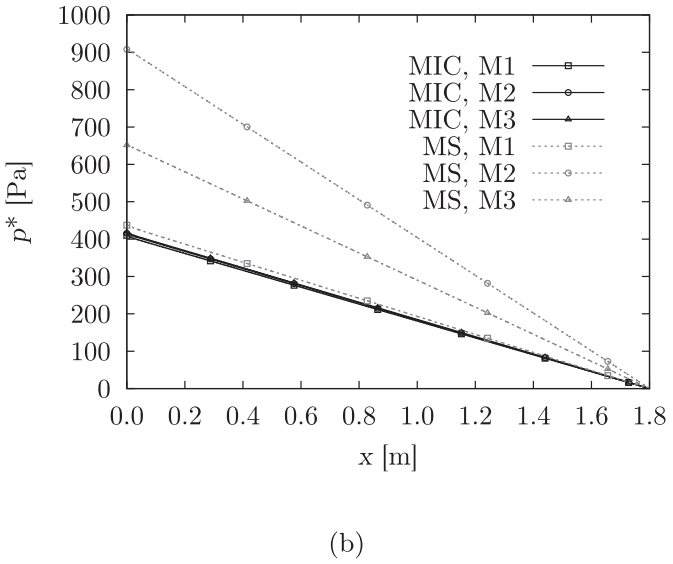


(f)

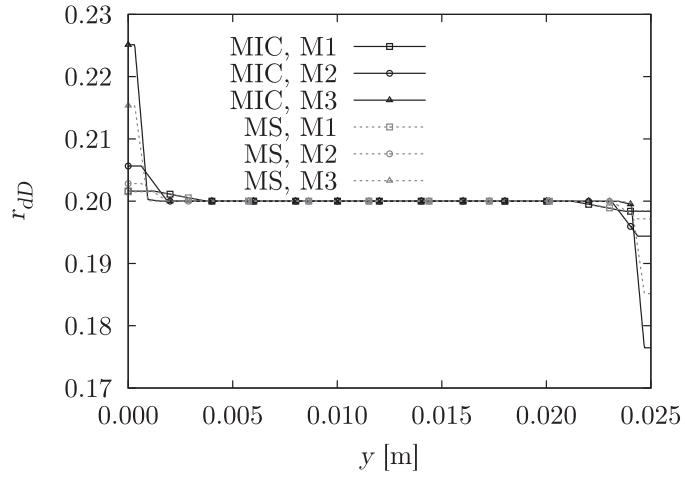
Fig. 9. Results for the case HC3: (a) continuous phase velocity, (b) modified pressure, (c) dispersed phase fraction, (d) transient deviations, (e) total number of inner iterations and (f) number of outer iterations within the timestep.



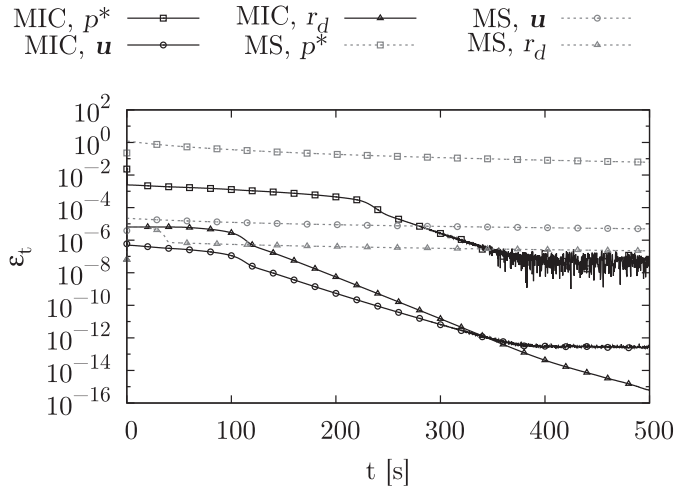
(a)



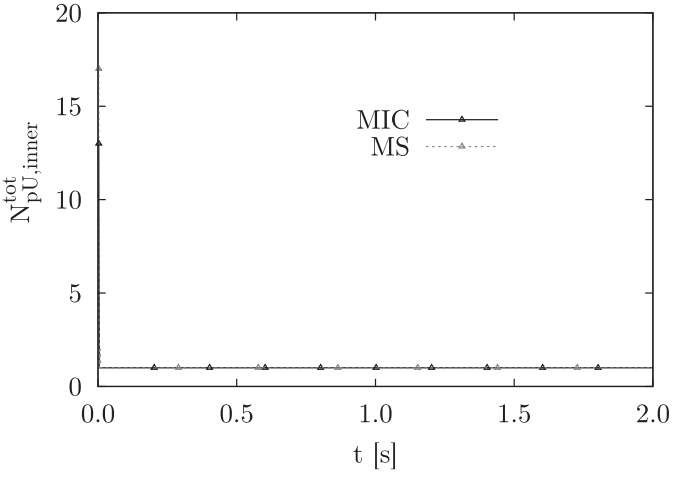
(b)



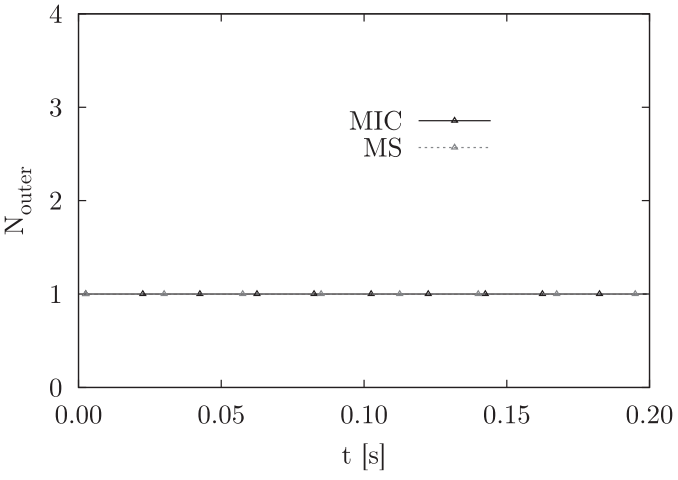
(c)



(d)



(e)



(f)

Fig. 10. Results for the case HC4: (a) continuous phase velocity, (b) modified pressure, (c) dispersed phase fraction, (d) transient deviations, (e) total number of inner iterations and (f) number of outer iterations within the timestep.

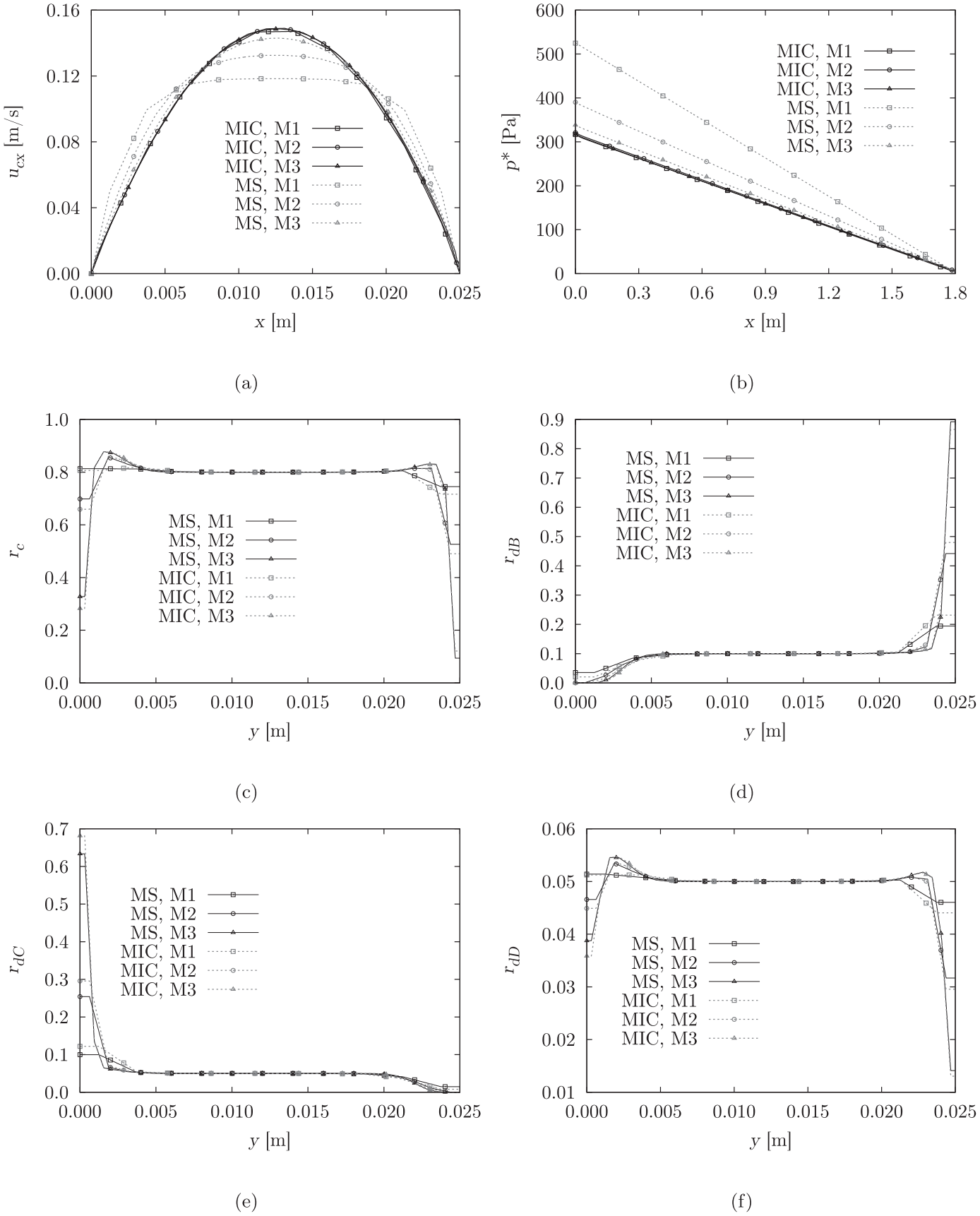


Fig. 11. Results for the case HC5: (a) continuous phase velocity, (b) modified pressure, (c) continuous phase fraction and fraction of dispersed phases (d) B, (e) C and (f) D.

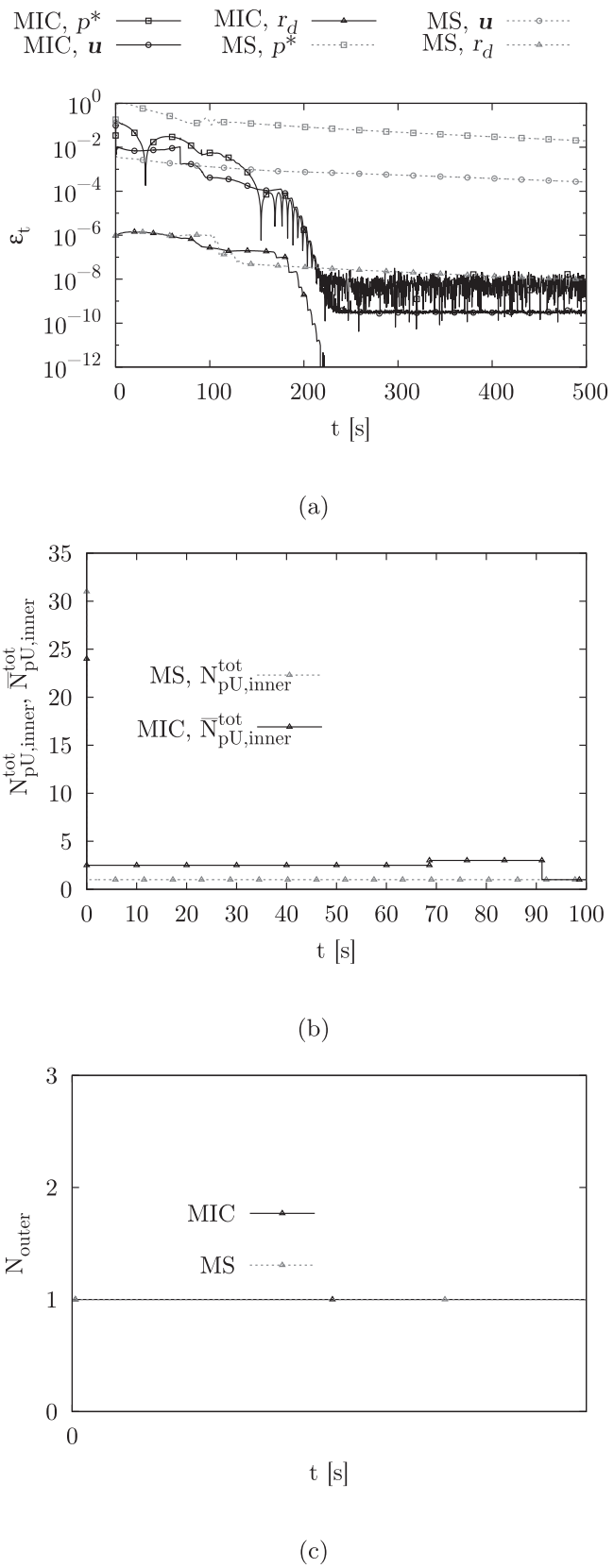


Fig. 12. Results for the case HC5: (a) transient errors, (b) total number of inner iterations and (c) number of outer iterations within the timestep for the case HC5 with M3.

tively. Both solvers performed only one outer iteration during the whole simulation. The MIC solver executed from 1 to 4 inner iterations, with an average of 2.5 iterations, in the interval between 0.0025 s and 68 s of the simulation, hence the number of iterations plotted is the average of the total number of inner iterations, $\overline{N}^{tot}_{pU,inner}$, for this interval. After approximately 90 s, just one inner iteration was executed until the end of simulation. On the other hand, the MS solver performed several iterations only in the first time step, using only one iteration for the rest of the simulation. Therefore, the MS solver runned much faster, but yielded a wrong solution.

7. Summary and conclusions

The implementation and verification of an implicitly coupled solver (MIC) for the multi-fluid model was performed. The computational code was implemented in C++ using the libraries from `foam-extend`, a fork of the open-source CFD toolbox OpenFOAM®. The numerical methodology developed in this work is based on the simultaneous solution of the momentum and pressure equations, being extended for systems with any number of phases.

We performed a thorough verification of the code, comparing it with a steady-state single phase flow solver and with an extended multiphase version (MS) of the segregated solver for the two-fluid Eulerian model implemented in foam-extend, the `twoPhaseEulerFoam` solver.

The Aggregative AMG and the Block-Selective AMG [34] methods were tested for the solution of the block-coupled matrix in the MIC solver. The SAMG has proven to be more effective on the solution of the problem, reaching the prescribed tolerance with few linear solver iterations. On the other hand, the AAMG solver showed a poor convergence behavior, performing the maximum allowed number of linear system iterations and requiring a large number of pressure–velocity coupling iterations to reach the desired accuracy.

The parallel scalabilities of the MS and MIC solvers were evaluated for the solution of a two-phase flow in a two-dimensional horizontal channel using 1–10 processors. For this case, the MS method has shown to be from 4.6 to 9.3 times faster than the MIC solver. However, its parallel efficiency was half of the one obtained for the MIC solver for the 10-processor runs.

The analysis of the convergence to the steady-state solution in several test cases showed that the MIC method was more robust that the MS solver. Considering cases with similar convergence criteria, the MIC method was able to reach steady-state solutions with smaller errors. The main advantage of the MIC solver over the MS solver is its capability of achieving convergence even when the drag coefficient is large. For these cases, the segregated pressure–velocity coupling used in the MS solver fails, providing flat velocity profiles and wrong pressure fields. These wrong solutions provided by the MS solver occur irrespective of the convergence criteria used for controlling the inner and outer iteration loops.

Acknowledgments

The authors thanks PETROBRAS contract 016/00097-2 (SAP 4600533021, Jur 0050.0102155.16.9); for the financial support. Paulo L.C. Lage acknowledges the financial support from CNPq, grants nos. 456905/2014-6 and 305265/2015-6, and from FAPERJ, grant no. E-26/111.361/2010. Luiz Fernando L.R. Silva acknowledges the financial support from FAPERJ, grant no. E-26/110.150/2014. Gabriel G.S. Ferreira acknowledges the financial support from CNPq, grant no. 140292/2013-6 and from CAPES Finance code 001 – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of Education of Brazil.

Appendix A. Rearrangement of the velocity divergence and drag terms

First, Eq. (51) is rewritten with the linearly interpolated velocity term of the drag correction moved into the velocity divergence term, yielding:

$$\nabla_{\mathbf{D}}\cdot\left[D_{Pf}\nabla_f p^*\cdot\mathbf{S}_f\right]=\nabla_{\mathbf{D}}\cdot(\Xi)+\nabla_{\mathbf{D}}\cdot\left[D_{Pf}(\nabla p^*)_f\cdot\mathbf{S}_f\right]$$
$$+\nabla_{\mathbf{D}}\cdot\left[\sum_{\alpha=1}^{P}(r_\alpha)_f\left(\frac{\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}(\Theta_{D\alpha\beta})_f\phi_\beta}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f}+\frac{(\Theta_{T\alpha})_f\left[\phi_\alpha^{t-1}-(\mathbf{u}_\alpha^{t-1})_f\cdot\mathbf{S}_f\right]}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f}\right.\right.$$
$$\left.\left.+\frac{\left[\mathbf{F}^*_{B\alpha,f}-(\mathbf{F}^*_{B\alpha})_f\right]\cdot\mathbf{S}_f}{\left[1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f\right](\mathbf{A}_\alpha)_f}\right)\right] \tag{A.1}$$

where

$$\Xi=\sum_{\alpha=1}^{P}(r_\alpha)_f\left[(\mathbf{u}_\alpha)_f\cdot\mathbf{S}_f-\frac{\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}(\Theta_{D\alpha\beta})_f(\mathbf{u}_\beta)_f\cdot\mathbf{S}_f}{1+(\Theta_{T\alpha})_f+(\Theta_{D\alpha})_f}\right] \tag{A.2}$$

In order to simplify the deduction, consider a system with $P=3$ phases *a, b* and *c*. Then, we can write Eq. (A.2) as:

$$\Xi=(r_a)_f\left[(\mathbf{u}_a)_f\cdot\mathbf{S}_f-\frac{(\Theta_{Dab})_f(\mathbf{u}_b)_f\cdot\mathbf{S}_f}{1+(\Theta_{Ta})_f+(\Theta_{Da})_f}-\frac{(\Theta_{Dac})_f(\mathbf{u}_c)_f\cdot\mathbf{S}_f}{1+(\Theta_{Ta})_f+(\Theta_{Da})_f}\right]$$
$$+(r_b)_f\left[(\mathbf{u}_b)_f\cdot\mathbf{S}_f-\frac{(\Theta_{Dba})_f(\mathbf{u}_a)_f\cdot\mathbf{S}_f}{1+(\Theta_{Tb})_f+(\Theta_{Db})_f}-\frac{(\Theta_{Dbc})_f(\mathbf{u}_c)_f\cdot\mathbf{S}_f}{1+(\Theta_{Tb})_f+(\Theta_{Db})_f}\right]$$
$$+(r_c)_f\left[(\mathbf{u}_c)_f\cdot\mathbf{S}_f-\frac{(\Theta_{Dca})_f(\mathbf{u}_a)_f\cdot\mathbf{S}_f}{1+(\Theta_{Tc})_f+(\Theta_{Dc})_f}-\frac{(\Theta_{Dcb})_f(\mathbf{u}_b)_f\cdot\mathbf{S}_f}{1+(\Theta_{Tc})_f+(\Theta_{Dc})_f}\right] \tag{A.3}$$

By grouping the terms associated to each phase velocity, we get:

$$\Xi=\left[(r_a)_f-\frac{(r_b)_f(\Theta_{Dba})_f}{1+(\Theta_{Tb})_f+(\Theta_{Db})_f}-\frac{(r_c)_f(\Theta_{Dca})_f}{1+(\Theta_{Tc})_f+(\Theta_{Dc})_f}\right](\mathbf{u}_a)_f\cdot\mathbf{S}_f$$
$$+\left[(r_b)_f-\frac{(r_a)_f(\Theta_{Dab})_f}{1+(\Theta_{Ta})_f+(\Theta_{Da})_f}-\frac{(r_c)_f(\Theta_{Dcb})_f}{1+(\Theta_{Tc})_f+(\Theta_{Dc})_f}\right](\mathbf{u}_b)_f\cdot\mathbf{S}_f$$
$$+\left[(r_c)_f-\frac{(r_a)_f(\Theta_{Dca})_f}{1+(\Theta_{Ta})_f+(\Theta_{Da})_f}-\frac{(r_b)_f(\Theta_{Dcb})_f}{1+(\Theta_{Tb})_f+(\Theta_{Db})_f}\right](\mathbf{u}_c)_f\cdot\mathbf{S}_f$$
$$=\sum_{\alpha=1}^{P}\left[(r_\alpha)_f-\sum_{\substack{\beta=1\\\beta\neq\alpha}}^{P}\frac{(r_\beta)_f(\Theta_{D\beta\alpha})_f}{1+(\Theta_{T\beta})_f+(\Theta_{D\beta})_f}\right](\mathbf{u}_\alpha)_f\cdot\mathbf{S}_f \tag{A.4}$$

This result can be straightforwardly extended to a system with any number of phases, as considered in Eq. (58).

References

[1] Cardiff P, Tuković Z, Jasak H, Ivanković A. A block-coupled finite volume methodology for linear elasticity and unstructured meshes. Comput Struct 2016;175:100–22.
[2] Choi SK. Note on the use of momentum interpolation method for unsteady flows. Numer Heat Transfer Part A 1999;36:545–50.
[3] Clifford I, Jasak H. The application of a multi-physics toolkit to spatial reactor dynamics. International conference on mathematics, computational methods & reactor physics. Saratoga Springs, New York: Curran Associates; 2009.
[4] Cubero A, Fueyo N. A compact momentum interpolation procedure for unsteady flows and relaxation. Numer Heat Transfer Part B 2007;52:507–29. June 2014
[5] Cubero A, Sánchez-insa A, Fueyo N. A consistent momentum interpolation method for steady and unsteady multiphase flows. Comput Chem Eng 2014;62:96–107.
[6] Darwish M, Moukalled F. A coupled pressure-based finite-volume solver for incompressible two-phase flow. Numer Heat Transfer Part B 2015;67:47–74.
[7] Darwish M, Sraj I, Moukalled F. A coupled incompressible flow solver on structured grids. Numer Heat Transfer Part B 2007;52:37–41.
[8] Darwish M, Sraj I, Moukalled F. A coupled finite volume solver for the solution of incompressible flows on unstructured grids. J Comput Phys 2009;228(1):180–201.
[9] Favero JL, Silva LFLR, Lage PLC. Modeling and simulation of mixing in water-in-oil emulsion flow through a valve-like element using a population balance model. Comput Chem Eng 2015;75:155–70.
[10] Fletcher R. Conjugate gradient methods for indefinite systems. In: Watson GA, editor. Numerical analysis: proceedings of the Dundee conference on numerical analysis, 1975. Berlin, Heidelberg: Springer Berlin Heidelberg; 1976. p. 73–89.
[11] Gabriel E, Fagg GE, Bosilca G, Angskun T, Dongarra JJ, Squyres JM, et al. Open MPI: goals, concept, and design of a next generation MPI implementation. In: Proceedings, 11th European PVM/MPI users' group meeting, Budapest, Hungary; 2004. p. 97–104.
[12] Gu CY. Computation of flows with large body forces. In: Taylor C, Chin JH, editors. Numerical methods in laminar and turbulent flow. Swansea: Pineridge Press; 1991. p. 294–305.
[13] Hill DP. The computer simulation of dispersed two-phase flows. Imperial College of Science, Technology and Medicine, University of London; 1998. Ph.D. thesis.
[14] Hutchinson B, Galpin P, Raithby G. Application of additive correction multigrid to the coupled fluid flow equations. Numer Heat Transfer 1988;13:133147.
[15] Ishii M, Hibiki T. Thermo-fluid dynamics of two-phase flow. 1st ed. Nova York: Springer; 2006.
[16] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. J Comput Phys 1986;62(1):40–65. doi:10.1016/0021-9991(86)90099-9.
[17] Jareteg K, Vukcevik V, Jasak H. PUCoupledFoam - an open source coupled incompressible pressure-velocity solver based on foam-extend; 2014.
[18] Jasak H. Error analysis and estimation for the finite volume method with applications to fluid flows. Imperial College of Science, Technology and Medicine, University of London; 1996. Ph.D. thesis.
[19] Kawaguchi Y, Tao W-Q, Ozoe H. Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method. Numer Heat Transfer Part B 2002;41(1):85–94.
[20] Lee J, Zhang J, Lu C-C. Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems. J Comput Phys 2003;185:158–75.
[21] Majumdar S. Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids. Numer Heat Transfer 1988;14:125–32.
[22] Miller T, Schmidt F. Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a nonstaggered grid system. Numer Heat Transfer 1988;14:213–33.
[23] Moukalled F, Darwish M. Pressure based algorithms for single-fluid and multi-fluid flows. In: Minkowycz WM, Sparrow EM, Murthy JY, editors. Handbook of numerical heat transfer. John Wiley & Sons, Inc.; 2009. p. 345–67.
[24] Passalacqua A, Fox RO. Implementation of an iterative solution procedure for multi-fluid gasparticle flow models on unstructured grids. Powder Technol 2011;213(1–3):174–87.
[25] Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. Int J Heat Mass Transfer 1972;15:1787–806.
[26] Raw M. A coupled algebraic multigrid method for the 3D Navier-Stokes equations. In: Hackbusch W, Wittum G, editors. Fast solvers for flow problems. Wiesbaden: Vieweg+Teubner Verlag; 1995. p. 204–15.
[27] Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J 1983;21(11):1525–32.
[28] Rusche H. Computational fluid dynamics of dispersed two-phase flows at high phase fractions. Imperial College of Science, Technology and Medicine, University of London; 2002. Ph.D. thesis.
[29] Schiller L, Naumann AZ. Über die grundlegenden berechnungen bei der schwerkraftaufbereitung. Ver Deut Ing 1933;77:318–20.
[30] Shen WZ, Michelsen J, Sorensen JN. Improved Rhie-Chow interpolation for unsteady flow computations. AIAA J 2001;39:2406–9.
[31] Silva LFLR, Lage PLC. Development and implementation of a polydispersed multiphase flow model in OpenFOAM. Comput Chem Eng 2011;35(12):2653–66.
[32] Spalding D. Numerical computation of multi-phase flow and heat transfer. In: Taylor C, Morgan K, editors. Recent advances in numerical methods in fluid mechanics. Swansea: Pineridge Press; 1980. p. 139–68.
[33] Spalding D. Developments of the IPSA procedure for numerical computation of multi-phase flow phenomena with interphase slip, unequal temperatures, etc. In: Shih TM, editor. Numerical methodologies in heat transfer. Washington: Hemisphere Publishing Corp.; 1983. p. 421–36.
[34] Uroić T, Jasak H. Block-selective algebraic multigrid for implicitly coupled pressure-velocity system. Comput Fluids 2018;167:100–10.
[35] Venier CM, Damian SM, Nigro NM. Numerical aspects of Eulerian gas-particles flow formulations. Comput Fluids 2016;133:151–69.
[36] Weller HG. Derivation, modelling and solution of the conditionally averaged two-phase flow equations. Technical report. United Kingdom: OpenCFD Ltd.; 2005.