On the use of the method of manufactured solutions for the verification of CFD codes for the volume-averaged Navier-Stokes equations

Bruno Blais[a], François Bertrand[a,*]

[a]*Research Unit for Industrial Flows Processes (URPEI), Department of Chemical Engineering, École Polytechique de Montréal, P.O. Box 6079, Stn Centre-Ville, Montréal,QC, Canada, H3C 3A7*

Abstract

The volume-averaged Navier-Stokes (VANS) equations are a key constituent of numerous models used to study complex problems such as flows in porous medias or containing multiple phases (e.g solid-liquid flows). These equations solve the mesoscopic scale of the flow without taking into account explicitly each individual solid particles, therefore greatly reducing computational cost. However, due to a lack of analytical solutions, the models using the VANS equations are generally validated directly against experimental data or empirical correlations. In this work, a framework to design analytical solutions and verify codes that solve the VANS equations by applying the method of manufactured solutions is presented for the first time. Three test cases of increasing complexity are designed with this method and are applied to assess the accuracy and order of convergence of a finite volume solver used in a framework combining computational fluid dynamics (CFD) and the discrete element method (CFD-DEM). The generic verification framework developed in this work is valid for any mesh-based numerical method in both proprietary or open-source codes that solve any form of the VANS equations such as those used in two-fluid, CFD-DEM, multiphase particle-in-cell and porous media modeling.

Keywords: Computational Fluid Dynamics; Multiphase flows; Code

*Corresponding author
Email address: francois.bertrand@polymtl.com (François Bertrand)

Verification; Method of Manufactured Solutions; Volume Averaged Navier-Stokes; CFD-DEM.

1. Introduction

It is well established that the Navier-Stokes equations govern the evolution of the velocity and pressure of the incompressible flow of an isothermal Newtonian fluid. However, this is not true for two phase flows for which a generic set of governing equations has yet to be established despite the extensive literature on the subject [1, 2, 3, 4, 5, 6]. The intrinsic multiscale nature of multiphase flows is another issue that renders their modeling very challenging. For instance, in the case of particle-laden flow, particle-particle collisions and solid-fluid interactions taking place at the microscopic scale can affect the flow at the mesoscopic and macroscopic scales [7]. This multiscale phenomenon complexifies the coupling between the phases and, when relevant, turbulence modeling. The resulting multiphase problem is hard to tackle with a single generic approach. For example, in solid-liquid mixing, it becomes computationally intractable to resolve fluid flow around more than a few thousand particles in a stirred tank. Therefore, different families of models have been designed, which can resolve the scales of interest with varying precision. These modeling scales can be referred to as macro, meso and micro [7].

The use of averaged equations for the description of the fluid phase consists in a meso approach that allows the investigation of multiphase flows at industrial and lab scales, such as the concentrated particle-laden flows in fluidized bed [8, 9] and mixing systems [10].

Various formulations of the governing averaged equations have been developed, as in [11, 5, 3]. In this paper, we focus on the case of a solid phase dispersed in another continuous phase, within the context of our work on solid-liquid mixing in both laminar and turbulent regimes.

For dispersed multiphase flows, the volume averaged Navier-Stokes equations (VANS) are a key constituent in more than one model. They are at the

foundation of the well-known two-fluid model in which both the fluid and the dispersed phases are considered to be governed by continuum fluid mechanics. This two-fluid model has been used to study a large number of solid-fluid processes, for instance in fluidized beds [12] and solid-liquid mixing [13]. The VANS equations have also been used to simulate porous media flows [14]. Furthermore, these equations are inherent to Euler-Lagrange models such as the Unresolved CFD-DEM, which combines the discrete element method (DEM) [15, 16] for the solid particles and the volume averaged Navier-Stokes equation for the fluid phase [17]. This approach combining CFD and DEM is particularly promising for the study of solid-liquid mixing due to its capacity to predict maximum packing fraction [15] and its suitability over the entire range of granular regimes (quasi-static to fast) [18].

However, work is still needed to understand the limitations of current CFD-DEM models. For instance, it remains unclear how DEM parameters such as the rolling friction and coefficient of restitution, or the different solid-fluid interaction forces (Saffman, Magnus, etc.) affect the flow at the macroscopic scale of the process. While this has been partly addressed by [19] for the solid-liquid interactions in stirred-tanks and [20] for the impact of rolling friction in fluidized beds, these reports are limited in scope as they pertain to specific flow situations.

More particularly, the past decade has seen an increase in the number of techniques used to solve the VANS equations for the continuous phase in CFD-DEM based models. While a large body of work has solved the fluid phase by classical finite volume [17, 9, 21, 22] or finite difference methods [23], alternative approaches have been proposed, based on Smoothed Particle Dynamics (SPH) [24] or the Lattice Boltzmann Method (LBM) [25].

CFDEM, an open source CFD-DEM framework [26], combines the finite volume method for the continuous fluid phase with the DEM for the particles. More precisely, this framework introduced by Kloss *et al.* [27] comprises the finite volume library Open∇FOAM [28] with the DEM code LIGGGHTS [29, 30] based on the molecular dynamic software LAMMPS [31]. It is highly interest-

ing due to its open source nature and the large user community surrounding Open∇FOAM, LAMMPS and LIGGGHTS. Furthermore, the platform is fully parallel, allowing it to handle relatively large problems.

The development of any simulation model, including those based on the CFD-DEM paradigm, should comprise a verification and validation step in order to assess its accuracy and performance. The definitions given by the American Institute of Aeronautics and Astronautics (AIAA) [32] and the American Society of Mechanical Engineers (ASME) [33] for these two concepts are :

- *Verification* : The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model.

- *Validation* : The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

Consequently, a verification procedure should establish that a numerical model for a set of partial differential equations with its associated boundary conditions converges towards the desired analytical solution at the asymptotic convergence rate prescribed by the mathematical analysis of the space and time discretizations used in the numerical scheme. On the other hand, validation refers to the comparison between experimental results and the solution obtained by the model. Therefore, one should first verify that a numerical implementation of a model is coherent before comparing it with experimental results. Failure to do so can lead to confusion as one is then unable to distinguish if the disagreement between experimental results and a model arise from fundamental inadequacies in the model or if it is solely due to coding errors, an incorrect discretization or the improper convergence of the underlying numerical scheme. We refer the readers to the seminal books by Roache [34] and Oberkampft and Roy [35] for a thorough discussion on the necessity of verification procedures in computational science and scientific computing in general.

Numerous validation studies have been reported for CFD-DEM models on a large variety of topics such as liquid fluidized beds [9], gas fluidized beds [23], rotor granulators [36], pneumatic conveying [37]. However, numerical verification work has been limited. Garg *et al.* [22] have carried out an extensive verification of the DEM resolution and solid-fluid coupling of the MFIX CFD-DEM software. Other analyzes have been carried out on mass conservation [38] and on the algorithm used to project DEM particles onto a CFD mesh [39]. However, to our knowledge, a systematic numerical verification procedure for the VANS equations has not been proposed yet. This is mainly due to the lack of analytical solutions for the VANS equations. Although solutions are available for two-fluid models through the use of analytical Riemann solvers and wave ordering, as for example in [40], these solutions are limited to the context of inviscid compressible flows. One can refer to the book of Toro [41] for more details on the Riemann solution of Euler systems of equations.

An alternative method for the verification of CFD models is the so-called Method of Manufactured Solutions (MMS), which can design analytical solutions for a set of partial differential equations [34, 35]. This method has been used with great success for the verification of wall-bounded turbulent flow [42], eddy-viscosity models [43] and Reynolds Averaged Navier-Stokes equations in turbulence modeling [44].

In this work, the lack of analytical solutions to the VANS equations is remedied by extending the method of manufactured solutions and using it to develop a step-by-step verification procedure for CFD codes solving these equations. Although this verification is carried out in the context of a pressure implicit with splitting of operators (PISO) solver for the VANS equations, the methodology developed in the present work is generic and can readily be applied to any mesh-based numerical scheme whether it is based on a finite difference, finite volume, finite element or lattice Boltzmann formulation.

The VANS equations are first recalled in the context of solid-liquid flow problems. The method of manufactured solutions applied to such multiphase flows is then introduced and used to design three test cases, which then serve to

verify the accuracy of the PISO scheme. Finally, some comments are given on the capacity of the procedure to identify formulation errors within a CFD code and the mass conservation properties of the PISO scheme in the context of the VANS equations.

2. Volume Averaged Navier-Stokes Equations

There exist a number of forms for the VANS equations that have been proposed in the literature for solid-liquid flows. The main differences between these forms lie principally in the treatment of the stress tensor of the fluid and the solid-fluid coupling, as thoroughly discussed by Zhou *et al.* [17] for the two-fluid and the CFD-DEM models.

Following the notation of these authors, set I (original model B) of the VANS equations, which is based on local averaging [5], is used in the current work. We refer the reader to the article of Zhou *et al.* [17] for an in-depth description of the origin of the model and its comparison to set II (model A) and set III (simplified model B).

This formulation, which has shown its suitability for solid-liquid flows [17], will be simply referred to as the VANS equations in the remainder of this paper.

The incompressible VANS equations are:

$$\frac{\partial \epsilon_f}{\partial t} + \nabla \cdot (\epsilon_f \boldsymbol{u}) = 0 \tag{1}$$

$$\frac{\partial (\rho_f \epsilon_f \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho_f \epsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \epsilon_f \boldsymbol{g} - \boldsymbol{F}_{pf} \tag{2}$$

where ϵ_f is the void fraction, ρ_f the density of the fluid, p the pressure, $\boldsymbol{u}$ the velocity and $\boldsymbol{g}$ the gravity. The viscous stress tensor $\boldsymbol{\tau}$ is defined as:

$$\boldsymbol{\tau} = \mu \left((\nabla \boldsymbol{u}) + (\nabla \boldsymbol{u})^T - \frac{2}{3} (\nabla \cdot \boldsymbol{u}) \boldsymbol{\delta}_k \right) \tag{3}$$

where μ is the dynamic viscosity and $\boldsymbol{\delta}_k$ the identity tensor. The volumetric

particle-fluid interaction term $\boldsymbol{F}_{pf}$ can be broken down into a sum of forces:

$$\boldsymbol{F}_{pf} = \frac{1}{\Delta V} \sum_{i}^{n_p} f_{pf,i} \tag{4}$$

$$\boldsymbol{f}_{pf,i} = \boldsymbol{f}_{d,i} + \boldsymbol{f}_{\nabla p,i} + \boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau},i} + \boldsymbol{f}_{\mathrm{vm},i} + \boldsymbol{f}_{B,i} + \boldsymbol{f}_{\mathrm{Saff},i} + \boldsymbol{f}_{\mathrm{Mag},i} \tag{5}$$

where n_p is the number of particles and $\boldsymbol{f}_{pf,i}$ is the sum of all fluid-solid interaction forces involving particle i: drag ($\boldsymbol{f}_{d,i}$), pressure gradient ($\boldsymbol{f}_{\nabla p,i}$), viscous stress gradient ($\boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau},i}$), virtual mass ($\boldsymbol{f}_{\mathrm{vm},i}$), Basset force ($\boldsymbol{f}_{B,i}$), Saffman lift ($\boldsymbol{f}_{\mathrm{Saff},i}$) and Magnus lift ($\boldsymbol{f}_{\mathrm{Mag},i}$). In practice, this term then requires information from the DEM part. For this reason, it is omitted in this work as we are interested in the verification of the codes that solve the VANS equations and not in the coupling between solid and fluid.

It is important to note that the velocity and void fraction resulting from these equations are not seperately divergence free, which means that all terms of the stress tensor are a priori non-zero. In particular, this may lead to the appearance of normal stresses even if the fluid is Newtonian and the flow is incompressible.

3. Numerical solution of the VANS equations

As previously noted in the introduction, numerous methods can be used to solve the VANS equations. In the present work, a finite volume method based on a PISO pressure predictor-corrector scheme [45] was implemented using the Open∇FOAM library. The key idea is to solve the momentum and pressure equations separately, and to use the pressure in order to ensure mass conservation by correcting iteratively the predicted velocity. Although alternative predictor-corrector schemes such as SIMPLE [46] would also be adequate, the choice of the PISO scheme is motivated by its good performance for the simulation of transient flows. One can refer to the book by Versteeg and Malasekera [47] for a comprehensive overview of some of the pressure-corrector schemes available in the literature. For a more generic presentation of the cell-centered

finite volume formulations available in Open∇FOAM, the reader is referred to the work of Weller *et al.* [48] and Jasak *et al.* [49].

In this section, a modified PISO algorithm that allows for a resolution of the VANS equations in conservative form is presented. This scheme was originally used in the work of Kloss *et al.* [27], although it was not described. It is detailed and extended upon in the present work.

A notation close to the one used in the book by Ferziger and Perić [50] is followed as it is similar to the Open∇FOAM formalism.

The VANS equations alongside generic mass and momentum source terms $H(\boldsymbol{x}, t)$ and $\boldsymbol{G}(\boldsymbol{x}, t)$ are given by:

$$\frac{\partial \epsilon_f}{\partial t} + \nabla \cdot (\epsilon_f \boldsymbol{u}) = H \tag{6}$$

$$\frac{\partial (\rho_f \epsilon_f \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho_f \epsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{G} \tag{7}$$

where the void fraction ϵ_f is calculated via a projection of the particle positions and radii onto the mesh. Therefore, ϵ_f is constant through the whole time step m.

The pressure predictor-corrector scheme begins with the solution of a predictor step for velocity $\boldsymbol{u}^{\mathrm{m}^*}$ using the pressure and velocity at time step $m-1$ (or the initial condition when $m = 1$):

$$A_{\mathrm{i}} \boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^*} + \sum_{\mathrm{j}} A_{\mathrm{j}} \boldsymbol{u}_{\mathrm{j}}^{\mathrm{m}^*} = Q_{\boldsymbol{u},\epsilon_f}^{\mathrm{m}-1} - \left(\frac{\delta p^{\mathrm{m}-1}}{\delta x} \right) + \boldsymbol{G} \tag{8}$$

The content of A and $Q_{\boldsymbol{u},\epsilon_f}^{\mathrm{m}-1}$ can be deduced from (7). The indices i and j refer to the cell centroids and to the neighboring cells, respectively. The pressure term is given explicitely and the symbolic derivative represents the centered spatial discretization scheme used in the present work. For an exhaustive presentation of this classical discretization scheme along with its truncation error analysis, we refer the reader to the book by Ferziger and Perić [50]. The resulting velocity $\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^*}$ does not respect (6), hence the need for a pressure corrector step.

First, a correction is applied to prevent velocity-pressure decoupling and

checkboard phenomenon following the work of Rhie and Chow [51]:

$$\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{**}} = \frac{Q_{\boldsymbol{u},\epsilon_f}^{\mathrm{m}^{*}-1} - \sum_{\mathrm{j}} A_{\mathrm{j}} \boldsymbol{u}_{\mathrm{j}}^{\mathrm{m}^{*}}}{A_{\mathrm{i}}} \tag{9}$$

This velocity and the cell-centered void fraction ϵ_f are used to calculate new mass fluxes at the cell faces via linear interpolation:

$$\phi_{\mathrm{F}}^{\mathrm{m}^{**}} = \langle \boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{**}} \epsilon_{f,\mathrm{i}} \rangle_{\mathrm{F}} \cdot \boldsymbol{S}_{\mathrm{F}} \tag{10}$$

where the $\langle . \rangle_{\mathrm{F}}$ operator denotes the face interpolation of a variable, the value of which is known at the cell centroids, and where $\boldsymbol{S}_{\mathrm{F}}$ is the surface normal vector term.

A similar interpolation is performed for the interpolation of the velocity coefficient A and the void fraction, which are known only at the cell centroids.

The following pressure correction correction equation is then applied:

$$\begin{aligned} \sum_{\mathrm{F}} \langle \frac{\epsilon_{f,\mathrm{i}}}{A_{\mathrm{i}}} \rangle_{\mathrm{F}} \boldsymbol{S}_{\mathrm{F}} \cdot \nabla p^{\mathrm{m}^{**}} = & \sum_{\mathrm{F}} \phi_{\mathrm{F}}^{\mathrm{m}^{**}} + \sum_{\mathrm{F}} \langle \frac{\epsilon_{f,\mathrm{i}}}{A_{\mathrm{i}}} \rangle_{\mathrm{F}} \langle \boldsymbol{G} \rangle_{\mathrm{F}} \cdot \boldsymbol{S}_{\mathrm{F}} \\ & + \frac{\delta \epsilon_{f,\mathrm{i}}}{\delta t} - H \end{aligned} \tag{11}$$

where again the symbolic time derivative is used to represent the discretization scheme used for the time derivative of the void fraction. In the present work, a second-order Crank-Nicholson scheme is used.

Finally, the velocity is corrected:

$$\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{***}} = \boldsymbol{u}^{\mathrm{m}^{**}} + \frac{1}{A_{\mathrm{i}}} \left(-\frac{\delta p^{\mathrm{m}^{**}}}{\delta x} + \boldsymbol{G} \right) \tag{12}$$

If Neumann or Robin boundary conditions are imposed in the domain, they are updated using the current value of the velocity and pressure.

This corrector step is generally carried out twice as in the original PISO method. This VANS PISO scheme has very good mass conservation properties because the equations are solved in their conservative formulation. We will come back to this in the results section.

In this work, the second-order centered scheme was used for cell face interpolations and gradients calculations. A second-order Crank-Nicholson scheme was used for time integration.

The block diagram in Figure 1 summarizes the VANS PISO scheme used in this work.

4. Verification of the Volume Averaged Navier-Stokes equations using the Method of Manufactured Solutions

The method of manufactured solutions (MMS) is a generic approach that allows one to build analytical solutions to given partial differential equations (PDE) [34, 35]. The complexity of the analytical solution can be chosen arbitrarily, allowing one to design a test case for which all the terms in the PDE are of the same order of magnitude. In this section, we show how this approach can be applied to develop analytical solutions to the VANS equations in order to create a rigorous verification procedure for a code that solves these equations. For a thorough presentation of MMS in a more general scientific computing context, we refer the reader to the books by Roache [34] and Oberkampft and Roy[35].

The MMS procedure is straightforward. First, we consider the VANS equations (Eqs. (6) and (7)) in the absence of solid-fluid interactions ($\boldsymbol{F}_{pf} = 0$). We choose a velocity field $\boldsymbol{u}$, a void fraction ϵ_f and a pressure p, and build a vector of manufactured variables $\boldsymbol{s}_{\mathrm{M}} = [\boldsymbol{u}^T, \epsilon_f, p]^T$ that satisfy the continuity equation (6).

For cases where the void fraction is time independent, it is preferable to chose a velocity and a void sraction that are intrinsically mass conservative, which means $H = 0$, as experience has shown us that this leads to a system that is more stable and closer to the real context of application of the VANS equations. In all cases, $\boldsymbol{s}_{\mathrm{M}}$ is not a solution of the complete VANS equations since it does not satisfy the momentum conservation (7). To do so, the following momentum source term $\boldsymbol{G}$ is added to the latter equation:

$$\boldsymbol{G}(\boldsymbol{s}) = \frac{\partial \left(\rho_f \epsilon_f \boldsymbol{u}\right)}{\partial t} + \nabla \cdot \left(\rho_f \epsilon_f \boldsymbol{u} \otimes \boldsymbol{u}\right) + \nabla p - \nabla \cdot \boldsymbol{\tau} \tag{13}$$

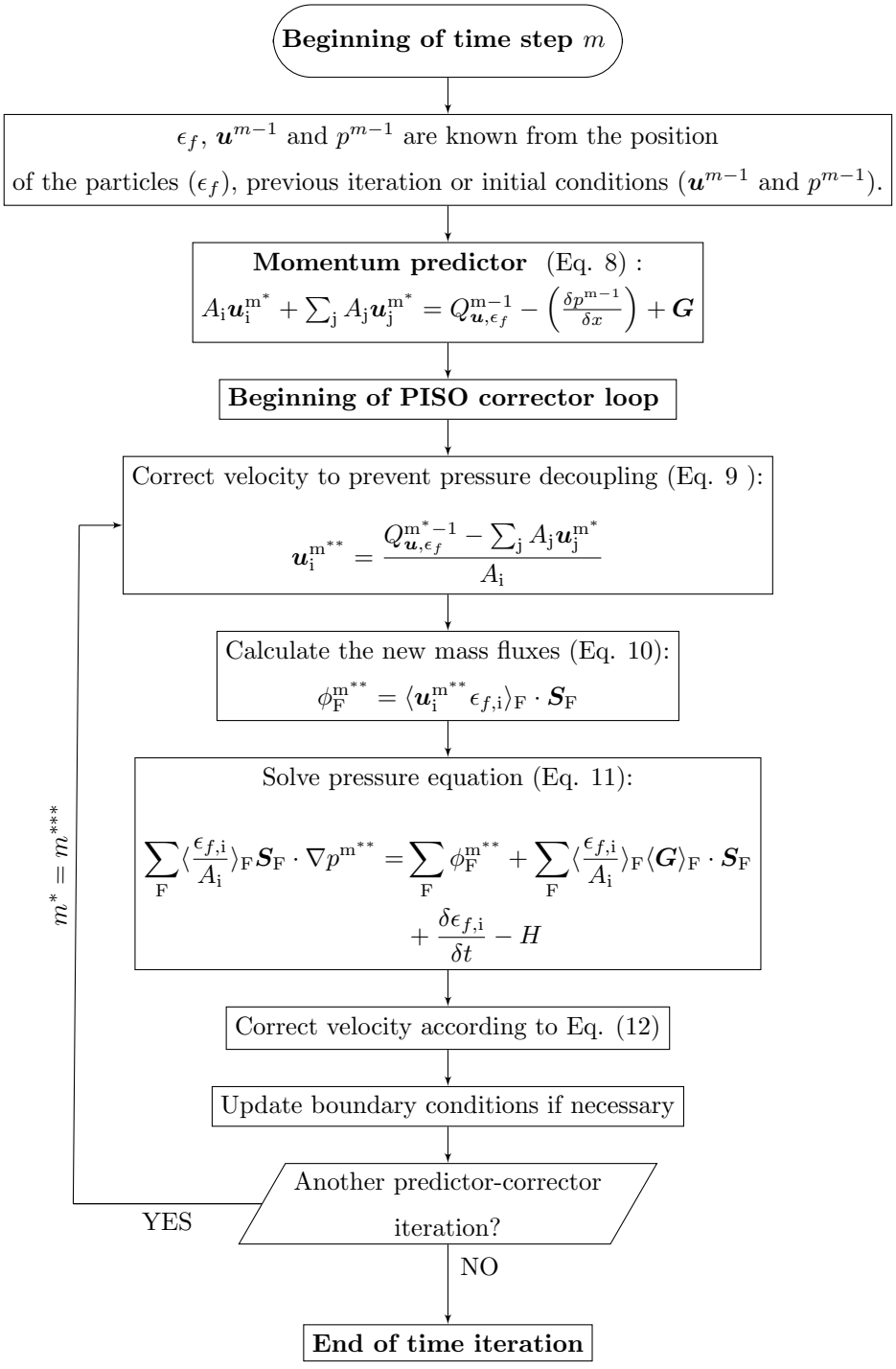

Figure 1: Flow chart for the VANS PISO scheme

where

$$\boldsymbol{\tau} = \mu \left((\nabla \boldsymbol{u}) + (\nabla \boldsymbol{u})^T - \frac{2}{3} (\nabla \cdot \boldsymbol{u}) \boldsymbol{\delta}_k \right) \tag{14}$$

With this definition of $\boldsymbol{G}$, the manufactured solution is an analytical solution of the VANS equations.

This solution can be used to assess the accuracy of the scheme described in Figure 1 by monitoring the decrease of the Euclidean norm of the error ($\|e\|_2$) with respect to the mesh size (Δx). On a Cartesian, homogenous and regular structured meshes, this error is defined for a variable $\boldsymbol{\xi}$ as [35]:

$$\|e_{\boldsymbol{\xi}}\|_2 = \sqrt{\frac{1}{N} \sum_i^N \|\boldsymbol{\xi}_{\mathrm{M},i} - \boldsymbol{\xi}_i\|^2} \tag{15}$$

where N is the number of mesh cells, $\boldsymbol{\xi}_{\mathrm{M}}$ the manufactured solution and $\boldsymbol{\xi}$ the numerical solution. The order of convergence obtained via simulations can be compared with the theoretical order of convergence of the scheme used for the solution of the VANS equations.

In the case of transient problems and if the simulations are carried out with a constant Courant-Friedrichs-Lewy number ($\mathrm{CFL} = \frac{\Delta t |\boldsymbol{u}|}{\Delta x}$), it follows that:

$$\|e_{\boldsymbol{\xi}}\|_2 \propto \alpha (\Delta x)^n + \beta \frac{\mathrm{CFL}^k}{|\boldsymbol{u}|^k} (\Delta x)^k \tag{16}$$

where α and β are unknown numerical constants, and where n and k are the theoretical orders of convergence of the space and time discretization used. If the orders of convergence for both the space and time discretizations are equal ($n = k$) as is the case in the present work where a linear space discretization is combined with a Crank-Nicolson time integration scheme, the error reduces to:

$$\|e_{\boldsymbol{\xi}}\|_2 \propto \gamma (\Delta x)^n \tag{17}$$

with γ an unknown numerical constant.

If the numerical model is consistent, the order of convergence measured via simulations should be the one predicted by theoretical analysis. Disagreement between the measured and the prescribed orders of convergence would imply

that there is a formulation error, a coding mistake or a discretization inconsistency. This approach is particularly powerful in the context of pressure-corrector schemes as inconsistencies in iterative approaches are hard to identify.

The source terms H and $\boldsymbol{G}$ can be calculated using a symbolic manipulator and directly inserted in the code. In the present work, Mathematica 8 [52] was used and the output was adapted to the OpenFOAM C++ syntax using regular expressions and Python 2.7.

The manufactured solution should be sufficiently differentiable, at least twice, to ensure that all members of the underlying discretized equations are non-zero. Therefore, the velocity, pressure and void fraction should preferably be polynomial, trigonometric or exponential functions. Furthermore, this method is most efficient when the manufactured solution generates terms in the PDE that are of the same order of magnitude. This prevents errors in one term of the equations from being damped by the stronger magnitude of others terms and ensure that all terms are significant in the calculation of the error. In the case of the VANS equations, this can be achieved by using $\mathrm{Re} = 1$. The values of the variables should also remain consistent with the physics of the equations. For instance, we should have $\epsilon_f \in]0, 1]$.

In the case of unresolved CFD-DEM, the void fraction ϵ_f is accounted for by projecting the particle positions and radii onto the CFD mesh. In this case, ϵ_f must be manufactured like $\boldsymbol{u}$ and p, although it is not a variable that is solved for. However, even if the void fraction is known analytically on the entire domain, its value should only be specified at the cell centroids and the boundary surfaces in the same manner as the other state variables. Failure to do so may lead to inconsistencies in the surface interpolation. Finally, the source term $\boldsymbol{G}$ and its divergence $\nabla \cdot \boldsymbol{G}$ are present in the momentum predictor and pressure correction equations, respectively. Although $\nabla \cdot \boldsymbol{G}$ is known by construction, it is calculated explicitly within the CFD scheme in order to maintain consistency. These are subtle details that can have an impact on the conclusions drawn from the order of convergence analysis.

5. Simulation and test case set-up

All the test cases designed involve either Dirichlet or periodic boundary conditions. Three different tests of increasing complexity are proposed, the last of which fully investigates all aspects of the VANS equations. The test cases are designed in 2D for simplicity and speed, and permutation of coordinates allows for a verification of all coordinates. The extension to 3D is straightforward.

All the simulations were carried out in transient regime using the VANS PISO predictor-corrector scheme described in Section 3, with a CFL of 0.2 (see Eq. (17)). Convergence to the steady-state solution was reached when the velocity and pressure residuals were inferior to 10^{-8}. The simulations were carried out on a large number of meshes (over 15) which involved from 400 to 250000 cells. Then, the order of convergence was calculated via a linear least squares regression of the Euclidean norm of the error over the entire range of meshes.

The domain of all three simulations is $\Omega = [-1, 1] \times [-1, 1]$. Since the manufactured solutions are periodic in nature, this corresponds to a full period of the velocity and pressure fields.

The three test cases are presented next along with the corresponding graphs of the Euclidean error between the analytical solution and the numerical solution obtained with the VANS PISO scheme for different mesh sizes. The results are analyzed in Section 6.

5.1. Case 1 : steady-state divergence-free flow problem

The first manufactured case is defined as:

$$\boldsymbol{u} = 2 \begin{bmatrix} -\left(\sin\left(\pi x\right)\right)^2 \sin\left(\pi y\right) \cos\left(\pi y\right) \\ \sin\left(\pi x\right) \cos\left(\pi x\right) \left(\sin\left(\pi y\right)\right)^2 \\ 0 \end{bmatrix} \tag{18}$$

$$p = \sin\left(\pi x\right) \sin\left(\pi y\right) \tag{19}$$

$$\epsilon_f = \frac{1}{2} + \frac{1}{4} \sin\left(\pi x\right) \sin\left(\pi y\right) \tag{20}$$

This case requires only a momentum source term ($\boldsymbol{G}$) in (7). This source term is calculated using (13) and is given by the following expression :

$$\begin{aligned} G_x =& \pi \sin(\pi y) \left(\cos(\pi x) \left(\sin^3(\pi x) \sin(\pi y)(\sin(\pi x) \sin(\pi y) + 2) + 1\right)\right) \\ &+ \pi \sin(\pi y) \left(4\pi\mu \cos^2(\pi x) \cos(\pi y)\right) \\ &- 12\pi^2 \sin(\pi y)\mu \sin^2(\pi x) \cos(\pi y)) \end{aligned} \tag{21}$$

$$\begin{aligned} G_y =& \pi \sin(\pi x) \cos(\pi y)(\sin^3(\pi x) \sin^3(\pi y)(\sin(\pi x) \sin(\pi y) + 2) + 1) \\ &+ \frac{1}{8}\pi \sin(2\pi x)(\sin(2\pi x) \sin(2\pi y) \sin^2(\pi y)(\sin(\pi x) \sin(\pi y) + 2) \\ &+ 16\pi\mu(1 - 2\cos(2\pi y))) \end{aligned} \tag{22}$$

In this case, the velocity, pressure and void fraction are all steady and sufficiently differentiable fields. Furthermore, the velocity field is divergence free. Therefore, the normal stresses in the viscous stress tensor are all zero.

The u and v components of $\boldsymbol{u}$, the pressure and the void fraction are shown in Figure 2.

The graph of the Euclidean norm of the error as a function of the mesh size is given in Figure 3.

5.2. Case 2 : steady-state non divergence-free flow problem

The second manufactured solution is defined as :

$$\boldsymbol{u} = \frac{1}{e} \begin{bmatrix} e^{\sin(\pi x) \sin(\pi y)} \\ e^{\sin(\pi x) \sin(\pi y)} \\ 0 \end{bmatrix} \tag{23}$$

$$p = \frac{1}{2} + \frac{1}{2} \sin(\pi x) \sin(\pi y) \tag{24}$$

$$\epsilon = \frac{1}{e} e^{-\sin(\pi x) \sin(\pi y)} \tag{25}$$

Figure 4 presents contour plots of the elements of this manufactured solution. Contrary to case 1, the divergence of the velocity field is now non-zero, although mass conservation (6) is still satisfied. Therefore, all the component of the full viscous stress tensor for a compressible flow are present in the VANS equations.

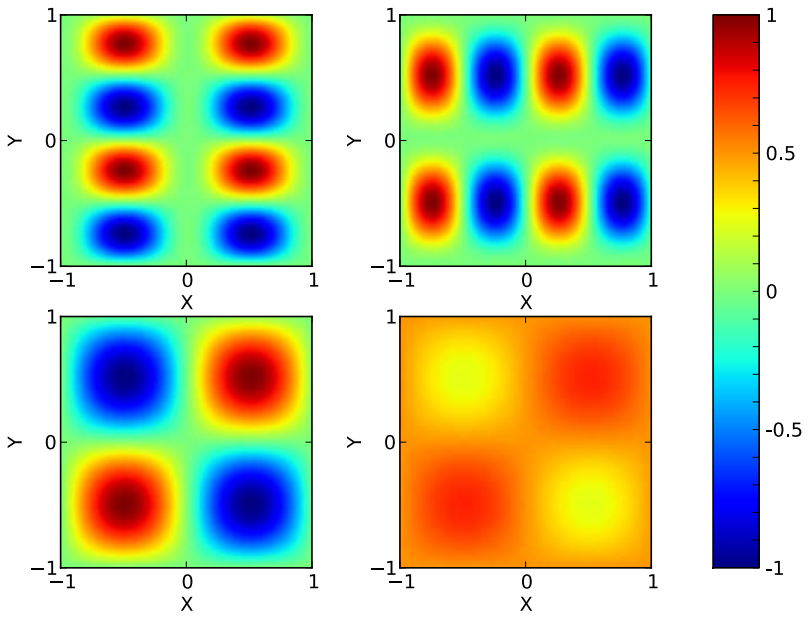

Figure 2: Analytical solution for case 1. Top left panel : u component of velocity — Top right panel : v component of velocity — Bottom left panel : pressure — Bottom right panel : void fraction

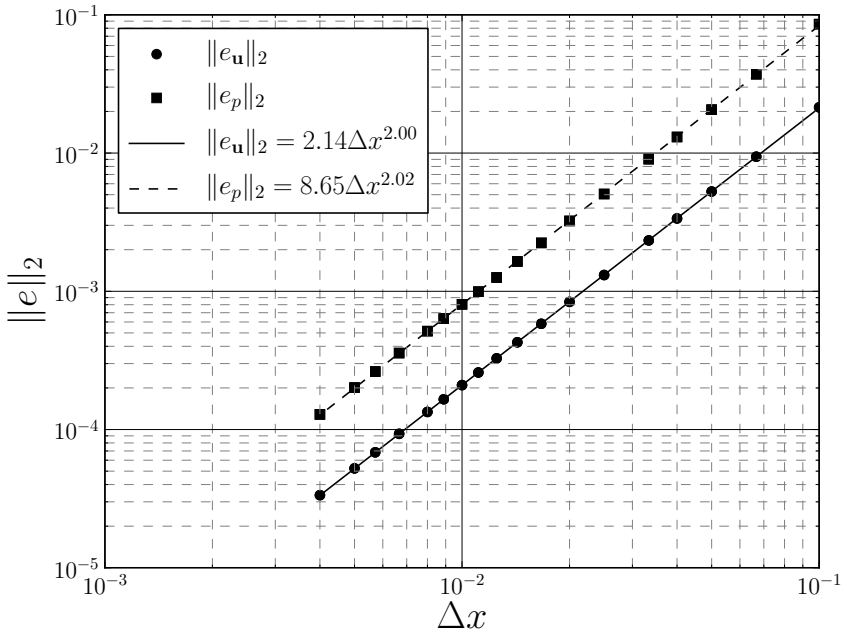

Figure 3: Graph of the Euclidean norm of the error for $\boldsymbol{u}$ and p with respect to mesh size for case 1.

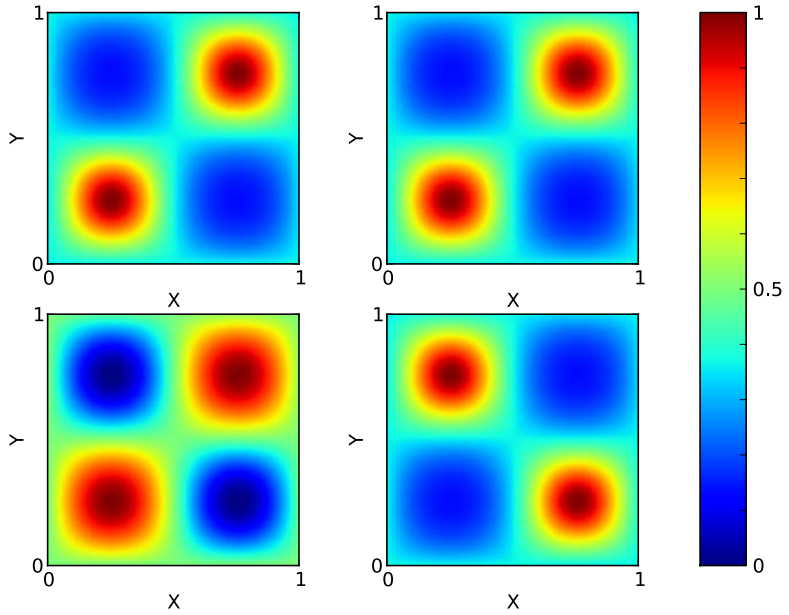

Figure 4: Analytical solution for case 2. Top left panel : u component of velocity — Top right panel : v component of velocity — Bottom left panel : pressure — Bottom right panel : void fraction

For this case, only a momentum source ($\boldsymbol{G}$) calculated using (13) is necessary. Due to its length, this term is not given here.

Figure 5 displays the graph of the Euclidean norm of the error as function of the mesh size.

5.3. Case 3 : unsteady non divergence-free flow problem

This test case is the unsteady extension of case 2, for which the solution is defined as :

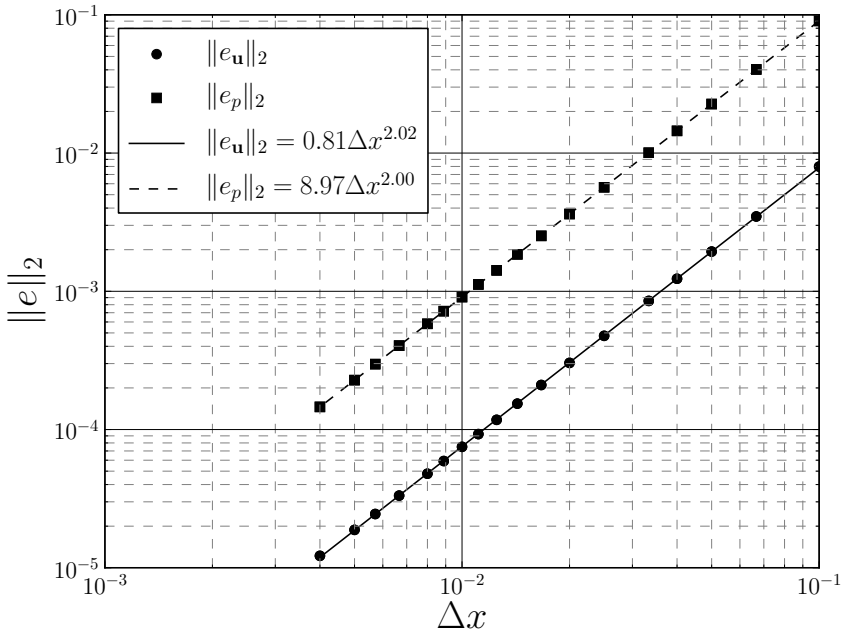

Figure 5: Graph of the Euclidean norm of the error for $\boldsymbol{u}$ and p with respect to mesh size for case 2.

$$\boldsymbol{u} = \frac{\cos(Tt)}{e} \begin{bmatrix} e^{\sin(\pi x)\sin(\pi y)} \\ e^{\sin(\pi x)\sin(\pi y)} \\ 0 \end{bmatrix} \tag{26}$$

$$p = \frac{1}{2} + \frac{1}{2}\cos(Tt)\sin(\pi x)\sin(\pi y) \tag{27}$$

$$\epsilon_f = \frac{1 - 0.1\cos(Tt)}{e} e^{-\sin(\pi x)\sin(\pi y)} \tag{28}$$

where T is the frequency of the velocity field, which is chosen here to be equal to 2π.

Figure 6 presents contour plots at time $t = 2$s. This test case is the most complete of the three since it is unsteady and the velocity field non-divergence free. Note that, since the void fraction is also unsteady, mass (H) and momentum ($\boldsymbol{G}$) source terms are required in (6) and (7), respectively.

Figure 7 displays the graph of the Euclidean norm of the error as a function of the mesh size at time $t = 2$s.

6. Discussion

The graphs displayed in Figures 3, 5 and 7 show that the velocity and the pressure solved by the VANS PISO scheme exhibit second-order convergence in both time and space. Since the time integration and the space discretization schemes for the calculation of the face fluxes and gradients are second-order accurate, these results highlight the fact that the VANS PISO approach within the CFDEM framework preserves this second-order accuracy. These results are in agreement with the theoretical development of the PISO approach [45]

Furthermore, the scheme has very good properties in terms of mass conservation. Indeed, for all simulations, the maximal local mass losses within one time iteration were observed to be of the order of 10^{-8}, whereas total mass conservation was ensured up to 10^{-10}. Even better mass conservation could be reached by the reducing the residual tolerance in the iterative solver or increasing the number of PISO loops.

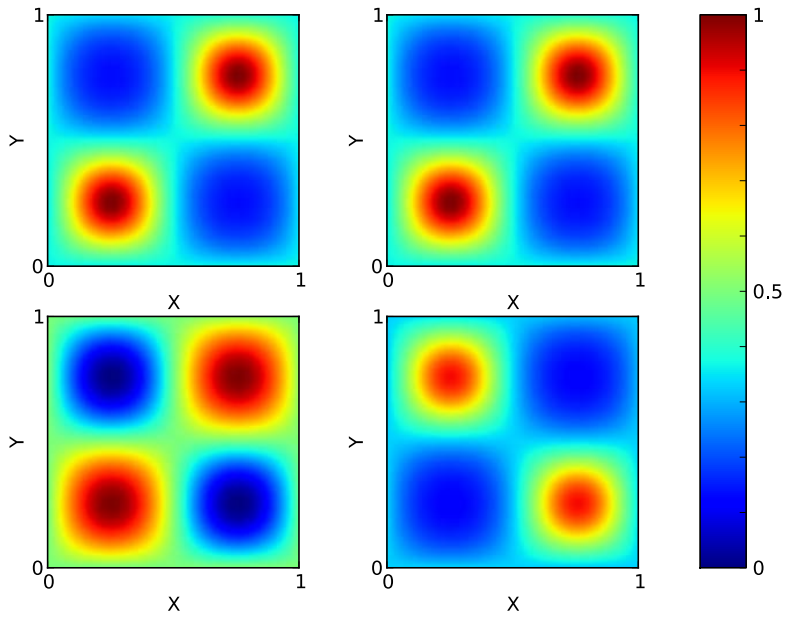

Figure 6: Analytical solution for case 3 at time $t = 2$s. Top left panel : u component of velocity — Top right panel : v component of velocity — Bottom left panel : pressure — Bottom right panel : void fraction

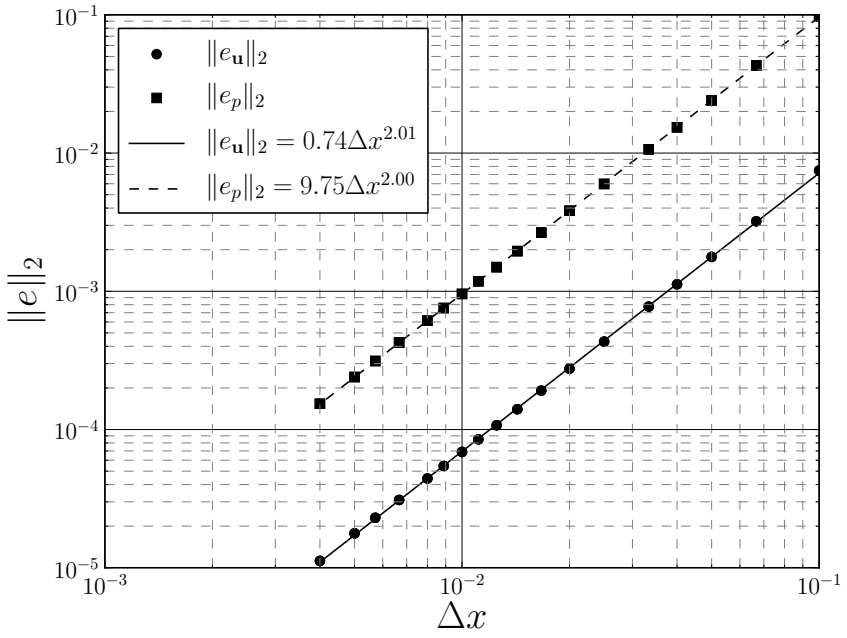

Figure 7: Graph of the Euclidean norm of the error for $\boldsymbol{u}$ and p with respect to mesh size for case 3.

One may notice in Figures 3, 5 and 7 that a second-order convergence rate is reached even for relatively coarse meshes for both pressure and velocity. Indeed, for all regressions, the R^2 coefficient was never lower than 0.9999. However, it is important to recall that a convergence rate is asymptotic by definition, which means that it is not always reached for coarse meshes. One should therefore remain careful and analyze the asymptotic rate of convergence, as the order of convergence for very coarse meshes can be different than the one predicted by the theoretical analysis, hinting a priori at an inconsistent scheme. This is clearly not the case for the meshes used in Figures 3, 5 and 7, but it has been observed for larger mesh sizes ($\Delta x > 0.1$) and for other types of boundary conditions.

One of the advantages of the combination of the MMS with the order of convergence analysis lies in its quantitative aspect. Indeed, one only needs to compare the order of convergence obtained with the theoretical order of convergence related to the space and time discretization schemes used, to ensure that the code implementation and the overall iterative scheme are consistent with the model equations. This is less error-prone than simple visual observation for which local errors can be difficult to pinpoint. The method is sensitive as a single error in a mesh cell, a term in the equations or the boundary conditions, may be sufficient to reduce the order of convergence. During the course of the verification carried out in this work, formulation errors in the VANS PISO scheme implemented in the CFDEM framework were identified and corrected using this approach. We would also like to stress the need for more than one single test case. Indeed, test cases can be designed to investigate a specific portion of the code or of the PDE. For example, case 2 is different from case 1 as it brings into play the full viscous stress tensor in the VANS equations. In particular, it allowed us to identify errors in the implementation of components of this tensor related to the occurent of a non-divergence free velocity.

7. Conclusion

The volume averaged Navier-Stokes equations can serve to model multiphase or porous medias flows in various applications. Indeed, they are part of numerous models in which they can describe all phases, such as in two-fluid models, or only the suspending fluid, as is the case in unresolved CFD-DEM.

The open source CFDEM platform combining LIGGGHTS and Open∇FOAM is highly interesting because of its open source character and the large community surrounding these two software tools. However, this platform is relatively new and requires thorough verification. In this article, it was shown that a PISO VANS scheme used within this coupling strategy is second-order accurate in both time and space, by applying the Method of Manufactured Solutions to the VANS equations. This allowed the verification of the implementation of these model equations.

It is important to mention that the approach proposed in this work is general and can be applied to any volume averaged formulation, be it for a two-fluid model, flow in porous media or a Euler-Lagrange model that relies on the finite volume, finite element or lattice Boltzmann method, as well as any other method based on a Eulerian description of the volume averaged phases. Furthermore, the step-by-step methodology using distinctive test cases allows one to easily identify errors that are linked to the incorrect discretization of a specific term of the VANS equations.

Finally, the methodology introduced here could also be extended to problems where the VANS equations must include a turbulence model, by manufacturing a solution that accounts for this turbulence model. In all situations, designing verification test cases via MMS allows for a clear and quantitative verification of a code before the corresponding model is validated by comparison with experimental data.

8. Acknowledgements

The financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged. In particular, Bruno Blais is thankful for the NSERC Vanier Scholarship. The authors also acknowledge technical support and computing time provided by Compute Canada, as well as Prof. Dominique Pelletier from Polytechnique Montreal for fruitful discussions on the method of manufactured solutions and its application to predictor-corrector schemes in the context of the finite volume method. Finally, we are grateful to DI Dr. Christoph Goniva from DCS computing for discussions on the formulation of the VANS equations within the CFDEM framework.

References

[1] J. Boure, J. Delhaye, General equations and two-phase flow modeling, Handbook of Multiphase Systems (1982) 1–36.

[2] D. Drew, Mathematical modeling of two-phase flow, Annual review of fluid mechanics 15 (1) (1983) 261–291.

[3] D. Gidaspow, Multiphase flow and fluidization: continuum and kinetic theory descriptions, Academic press, 1994.

[4] C. Crowe, J. D. Schwartzkopf, M. Sommerfeld, Y. Tsuji, Multiphase flows with droplets and particles, CRC Press, 2012.

[5] T. Anderson, R. Jackson, A fluid mechanical description of fluidized beds, Industrial & Engineering Chemistry Fundamentals 6 (1967) 527–539.

[6] A. Prosperetti, G. Tryggvason, Computational methods for multiphase Flow, Cambridge University Press, 2007.

[7] Y. Tsuji, Multi-scale modeling of dense phase gas-particle flow, Chemical Engineering Science 62 (13) (2007) 3410–3418. doi:DOI10.1016/j.ces.2006.12.090.

[8] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, A. B. Yu, Discrete particle simulation of particulate systems: A review of major applications and findings, Chemical Engineering Science 63 (23) (2008) 5728–5770. doi:DOI10.1016/j.ces.2008.08.006.

[9] A. Di Renzo, F. Cello, F. P. Di Maio, Simulation of the layer inversion phenomenon in binary liquid-fluidized beds by DEM-CFD with a drag law for polydisperse systems, Chemical Engineering Science 66 (13) (2011) 2945–2958. doi:DOI10.1016/j.ces.2011.03.035.

[10] T. Shao, Y. Hu, W. Wang, Y. Jin, Y. Cheng, Simulation of solid suspension in a stirred tank using CFD-DEM coupled approach, Chinese Journal of Chemical Engineering 21 (10) (2013) 1069–1081. doi:10.1016/s1004-9541(13)60580-7.

[11] M. Ishii, Themo-Fluid Dynamics Theory of Two-Phase Flow, 1975.

[12] J. Wang, M. A. van der Hoef, J. A. M. Kuipers, Why the two-fluid model fails to predict the bed expansion characteristics of Geldart a particles in gas-fluidized beds: A tentative answer, Chemical Engineering Science 64 (3) (2009) 622–625. doi:10.1016/j.ces.2008.09.028.

[13] A. Tamburini, A. Brucato, A. Cipollina, G. Micale, M. Ciofalo, Cfd predictions of sufficient suspension conditions in solid-liquid agitated tanks, International Journal of Nonlinear Sciences and Numerical Simulation 13 (6) (2012) 427–443. doi:DOI10.1515/ijnsns-2012-0027.

[14] R. E. Hayes, A. Afacan, B. Boulanger, An equation of motion for an incompressible Newtonian fluid in a packed bed, Transport in Porous Media 18 (2) (1995) 185–198. doi:10.1007/BF01064677.

[15] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, A. B. Yu, Discrete particle simulation of particulate systems: Theoretical developments, Chemical Engineering Science 62 (13) (2007) 3378–3396. doi:DOI10.1016/j.ces.2006.12.089.

[16] F. Bertrand, L. A. Leclaire, G. Levecque, DEM-based models for the mixing of granular materials, Chemical Engineering Science 60 (8-9) (2005) 2517–2531. doi:DOI10.1016/j.ces.2004.11.048.

[17] Z. Y. Zhou, S. B. Kuang, K. W. Chu, A. B. Yu, Discrete particle simulation of particle-fluid flow: model formulations and their applicability, Journal of Fluid Mechanics 661 (2010) 482–510. doi:Doi10.1017/S002211201000306x.

[18] X. Chen, J. Wang, A comparison of two-fluid model, dense discrete particle model and CFD-DEM method for modeling impinging gassolid flows, Powder Technology 254 (2014) 94–102. doi:10.1016/j.powtec.2013.12.056.

[19] J. J. Derksen, Numerical simulation of solids suspension in a stirred tank, Aiche Journal 49 (11) (2003) 2700–2714. doi:DOI10.1002/aic.690491104.

[20] C. Goniva, C. Kloss, N. G. Deen, J. A. M. Kuipers, S. Pirker, Influence of rolling friction on single spout fluidized bed simulation, Particuology 10 (5) (2012) 582–591. doi:10.1016/j.partic.2012.05.002.

[21] K. D. Kafui, C. Thornton, M. J. Adams, Discrete particle-continuum fluid modelling of gas-solid fluidised beds, Chemical Engineering Science 57 (13) (2002) 2395–2410. doi:PiiS0009-2509(02)00140-9Doi10.1016/S0009-2509(02)00140-9.

[22] R. Garg, J. Galvin, T. Li, S. Pannala, Open-source MFIX-DEM software for gassolids flows: Part Iverification studies, Powder Technology 220 (2012) 122–137.

[23] P. Pepiot, O. Desjardins, Numerical analysis of the dynamics of two- and three-dimensional fluidized bed reactors using an Euler-Lagrange approach, Powder Technology 220 (2012) 104–121. doi:DOI10.1016/j.powtec.2011.09.021.

[24] M. Robinson, S. Luding, M. Ramaioli, Grain sedimentation with SPH-DEM and its validation, in: AIP Conference Proceedings, Vol. 1542, p. 1079.

[25] F. Song, W. Wang, J. Li, A lattice Boltzmann method for particle-fluid two-phase flow, Chemical Engineering Science.

[26] CFDEM, CFDEM- Open Source CFD, DEM and CFD, URL : http://www.cfdem.com, 2014.

[27] C. Kloss, C. Goniva, A. Hager, S. Amberger, S. Pirker, Models, algorithms and validation for opensource DEM and CFDDEM, Progress in Computational Fluid Dynamics, an International Journal 12 (2) (2012) 140–152. doi:10.1504/pcfd.2012.047457.

[28] OpenCFD, OpenFOAM - The Open Source CFD Toolbox, URL : http://www.openfoam.com, 2014.

[29] LIGGGHTS, LAMMPS Improved for General Granular and Granular Heat Transfer Simulations, URL : hhtp://www.liggghts.com, 2014.

[30] C. Kloss, C. Goniva, M. The Minerals, S. Materials, LIGGGHTS Open Source Discrete Element Simulations of Granular Materials Based on LAMMPS, John Wiley & Sons, Inc., 2011, pp. 781–788, (TMS).

[31] S. Plimpton, P. Crozier, A. Thompson, LAMMPS - large-scale atomic/molecular massively parallel simulator, Sandia National Laboratories.

[32] American Institute of Aeronautics and Astronautics, AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations, American Institute of Aeronautics & Astronautics, 1998.

[33] American Society of Mechanical Engineers, Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer, American Society of Mechanical Engineers, 2009.

[34] P. Roache, Verification and Validation in Computational Science and Engineering, Hermosa Publishers, 1998.

[35] W. Oberkampf, C. Roy, Verification and Validation in Scientific Computing, Cambridge University Press, 2010.

[36] J. Neuwirth, S. Antonyuk, S. Heinrich, M. Jacob, CFDDEM study and direct measurement of the granular flow in a rotor granulator, Chemical Engineering Science 86 (2013) 151–163. doi:10.1016/j.ces.2012.07.005.

[37] S. B. Kuang, A. B. Yu, Micromechanic modeling and analysis of the flow regimes in horizontal pneumatic conveying, AIChE Journal 57 (10) (2011) 2708–2725. doi:Doi10.1002/Aic.12480.

[38] C. L. Wu, K. Nandakumar, A. S. Berrouk, H. Kruggel-Emden, Enforcing mass conservation in DPM-CFD models of dense particulate flows, Chemical Engineering Journal 174 (1) (2011) 475–481. doi:10.1016/j.cej.2011.08.033.

[39] J. Marshall, K. Sala, Comparison of methods for computing the concentration field of a particulate flow, International Journal of Multiphase Flow.

[40] J.-M. Hérard, O. Hurisse, A fractional step method to compute a class of compressible gasliquid flows, Computers & Fluids 55 (0) (2012) 57–69. doi:http://dx.doi.org/10.1016/j.compfluid.2011.11.001.

[41] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics : A Practical Introduction, Springer-Verlag, Berlin, 2009.

[42] L. Eça, M. Hoekstra, A. Hay, D. Pelletier, A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow, International Journal of Computational Fluid Dynamics 21 (3-4) (2007) 175–188. doi:10.1080/10618560701553436.

[43] L. Eça, M. Hoekstra, A. Hay, D. Pelletier, On the construction of manufactured solutions for one and two-equation eddy-viscosity models, International Journal for Numerical Methods in Fluids 54 (2) (2007) 119–154. doi:10.1002/fld.1387.

[44] L. Eça, M. Hoekstra, A. Hay, D. Pelletier, Verification of RANS solvers with manufactured solutions, Engineering with Computers 23 (4) (2007) 253–270. doi:10.1007/s00366-007-0067-9.

[45] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational physics 62 (1) (1986) 40–65.

[46] S. V. Patankar, D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer 15 (10) (1972) 1787–1806.

[47] H. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education Limited, 2007.

[48] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in Physics 12 (6) (1998) 620–631. doi:doi:http://dx.doi.org/10.1063/1.168744.

[49] H. Jasak, A. Jemcov, Z. Tukovic, Openfoam: A C++ library for complex physics simulations, in: International workshop on coupled methods in numerical dynamics, 2007, pp. 1–20.

[50] J. H. Ferziger, M. Peri, Computational methods for fluid dynamics, Vol. 3, Springer Berlin, 1996.

[51] F. P. Krrholm, Rhie-chow interpolation in openfoam, Appendix from Numerical Modelling of Diesel Spray Injection and Turbulence Interaction at Chalmers University. http://www. tfd. chalmers. se/hani/kurser/OS CFD.

[52] W. Research, Mathematica, version 8.0 Edition, Wolfram Research, Champaign, Illinois, 2010.