Highly scalable DNS solver for turbulent bubble-laden channel flow

Citation for published version (APA):

Cifani, P., Kuerten, J. G. M., & Geurts, B. J. (2018). Highly scalable DNS solver for turbulent bubble-laden channel flow. *Computers & Fluids*, 172, 67-83. <https://doi.org/10.1016/j.compfluid.2018.06.008>

DOI:

[10.1016/j.compfluid.2018.06.008](https://doi.org/10.1016/j.compfluid.2018.06.008)

Document status and date:

Published: 30/08/2018

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Highly scalable DNS solver for turbulent bubble-laden channel flow

P. Cifani^{a,*}, J.G.M. Kuerten^{c,b}, B.J. Geurts^{b,d}

^a*Institute for Mathematics and Computer Science, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands*

^b*Multiscale Modeling and Simulation, Faculty EEMCS, University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands*

^c*Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands*

^d*Multiscale Physics of Energy Systems, Center for Computational Energy Research, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands*

Abstract

We present an efficient and highly scalable solver for direct numerical simulation (DNS) of dispersed gas-liquid flow, containing a large number of deformable bubbles. We apply this to $O(10^4)$ bubbles in a turbulent flow. This was accomplished by a well-considered combination of state-of-the-art numerical methods, as well as fast and scalable numerical algorithms that have their origin in single-phase and two-phase flows. The features key-elements of the algorithm of curvature computation, bubble collision and variable-coefficient Poisson equation solver are presented. Resolution requirements for an accurate advection of a rising bubble have been established by comparison with the (Hysing, 2009) benchmark. The Generalized Height Function (GHF) method, was adopted for the curvature computations. We observed agreement with theoretical convergence rates, as well as a significant reduction of spurious velocities when employing GHF, compared to a finite difference approach. The main interaction mechanisms between bubbles and walls of the domain were analyzed, showing second order convergence of the underlying numerical methods.

A detailed analysis of the parallel performance of the Navier-Stokes (NS)

*Corresponding author. Tel: +31 50 36 33939
Email address: p.cifani@rug.nl (P. Cifani)

solver and the solver for the gas volume fraction was carried out. The analysis revealed close to linear scaling up to ≈ 18000 cores on a computational grid with 1 billion cells for the NS solver, and an ideal scaling of the gas volume fraction solver up to $O(10^3)$ bubbles. Beyond that, acceptable overhead for up to $O(10^4)$ bubbles was found.

A simulation of a downflow configuration of a turbulent channel loaded with a total of 10000 bubbles illustrates the computational capabilities. First and second order statistics of the velocity field were computed as well as the profiles of the average gas volume fraction field in the statistically steady state. In the case considered, a 47% increase of the wall shear stress was observed, brought about by turbulence modification arising from the embedded bubbles. The interaction between turbulence and bubbles at high volume fraction resulted in a strong attenuation of the rms of the velocity fluctuations in a wide region of the core of the channel.

Keywords: Bubbly flow, Channel flow, DNS, VOF, openMP-MPI

1. Introduction

This study deals with turbulent bubble-laden channel flows. Under suitable flow conditions, the dynamics of such complex systems can be studied by Direct Numerical Simulation (DNS) where all the relevant details of the flow are re-
5 solved. This paper describes the development of a new solver for such dispersed multiphase flow, which allows for very large numbers of bubbles to be simulated in full detail. The new solver provides a powerful tool for better understanding the physics that governs the problem and it has been made available as open-source software.

10 Turbulence is characterized by a wide range of scales of motion expressed through an energy cascade mechanism [1]. All scales need to be accurately resolved by the computational grid and time-step. A fundamental parameter for single-phase flows is the Reynolds number (Re), defined as the ratio between the magnitudes of convection and diffusion. As Re increases the smallest tem-

15 poral and spatial scales decrease very rapidly. At the same time, the presence
 of a second phase adds even more scales to the spectrum, next to those already
 introduced by turbulence alone. For example, a distorted gas-liquid interface
 combined with surface tension leads to capillary waves with their own dynam-
 ics. Likewise, a very thin liquid film between two colliding bubbles is formed
 20 with characteristic dimension orders of magnitude smaller than the scales of
 the surrounding turbulent flow. Equally relevant, the presence of bubbles in
 turbulent flows significantly changes the flow statistics by the introduction of
 higher velocity fluctuations induced by the wakes generated by these bubbles.
 These and similar mechanisms combined lead to an enormous computational
 25 challenge. Hence, in order to be able to simulate systems of practical merit, it
 is essential to design efficient and accurate numerical algorithms that scale well
 on a large number of processors.

A comprehensive review of DNS of bubbly flows is presented in [2]. In the
 pioneering work [3] a DNS of turbulent downflow in a channel, with fully de-
 30 formable bubbles, is carried out for different gas volume fractions. Several first
 order and second order flow statistics are computed and compared with single-
 phase channel flow. Some key observations are as follows. Migration of bubbles
 toward the core of the channel was observed and attributed to the lift force.
 This leaves a bubble-free wall boundary layer. In addition, with an imposed
 35 shear stress at the walls of the channel, the addition of bubbles significantly
 reduces the mass flow rate. It was also found that the wakes generated by the
 rising bubbles significantly increase the velocity fluctuations in the core of the
 channel. The companion papers [4, 5] focus on the effect of bubble size and bub-
 ble deformability. Since the appearance of these seminal papers, the interest in
 40 DNS of turbulent bubbly flows has grown considerably. More recently, in [6, 7] a
 channel with an up-ward flow configuration was investigated for different initial
 bubble distributions. Given the high computational requirements of such simu-
 lations, numerical methods tailored to solve simplified physical approximations
 have been developed. In [8] the bubbles are approximated as rigid spherical par-
 45 ticles and the mass density is assumed to be uniform in the governing equations.

A similar approach was developed in [9] where the bubble deformability is approximated by a phenomenological model. In this paper we will steer away from any such simplifications and create a reference computational model accurately accounting for all physical mechanisms.

50 The ever-increasing computational power makes it now feasible to perform DNS at relatively high Re and for a large number of bubbles. Currently, these simulations are limited to hundreds of bubbles [2]. The combination of methods presented in this paper allows increasing this bubble limit by two orders of magnitude. For fully resolved bubbles the main computational cost is related to
55 the advection algorithm which solves for the transport of an indicator function advected by the underlying velocity field. Mathematically, this is equivalent to solving for a purely hyperbolic equation [10]. Several methods to approach this problem are found in literature: the volume of fluid (VOF) method [11], the front-tracking method [12], the level-set (LS) method [13], the local front
60 reconstruction method (LFRM) [14] as well as various combinations of the aforementioned methods among which the VOF-LS method [15]. In the present work, the VOF method has been adopted since it constitutes a widely used and well-established technique which has been shown to converge to the theoretical order of accuracy with grid refinement [16, 17, 18, 19] and, at the same time, delivers
65 a sharp gas-liquid interface by construction. We stress that the basic framework developed here is largely independent of the underlying advection scheme and hence also applicable to other interface capturing methods.

The VOF algorithm requires a number of main operations: reconstruction of the interface shape [20, 21, 22], estimation of the indicator function fluxes [23, 16,
70 21, 17] and computation of the interface curvature and surface tension [24, 25, 26]. The number of operations involved grows proportionally with the number of bubbles N_b . For large N_b the VOF advection becomes the most expensive part of the solver. Furthermore, in order to represent a realistic physical interaction between bubbles and to avoid so-called numerical coalescence [27], the bubbles
75 have to be tracked individually and typically on a local grid associated with each bubble. This introduces an additional and rather large amount of data transfer

between the velocity and volume fraction grids which needs to be carefully designed to achieve competitive parallel efficiency.

The goal of this study is to document the development of an accurate and
80 massively scalable numerical code able to simulate turbulent channel flow at an
unprecedented high number of bubbles ($N_b = O(10^4)$). A detailed description
of the included algorithms will be given, analyzing grid convergence rates and
related computational costs. This provides a framework for future developments
of numerical methods for even more complex bubbly flows. To this end, the code
85 has been made publicly available [28].

The simple geometry of the channel combined with periodicity in the stream-
wise and the span-wise directions allows to represent the averaged quantities as
one-dimensional profiles along the wall normal direction. This facilitates a direct
comparison with the well-established turbulent statistics of single-phase chan-
90 nel flow [29, 30, 31, 32]. The developed code can be used to generate a unique
database from which effects of a large number of dispersed bubbles on turbu-
lence and collective bubble dynamics can be quantified. An example of such
a simulation is given in this paper, where a turbulent downflow configuration
with 10^4 bubbles is investigated.

95 The paper is organized as follows. In Section 2 the mathematical formulation
of the governing equations is provided. Section 3 details the numerical algorithm
developed for the solution of the variable-coefficient Navier-Stokes equations and
the advection of the volume fraction field. For the former set of equations we
employ a staggered, energy conserving Navier-Stokes solver for wall-bounded
100 turbulence [33]. A modified Poisson equation is employed, as derived in [34],
which allows for the use of fast direct solvers.

The volume fraction algorithm is tested in Section 4 through a number of
steps. While the interface reconstruction algorithm is essentially the same as
used in [19], here we focus on the analysis of the Generalised Height Function
105 method (GHF) [35] for the computation of the curvature. The case of a 2D
rising bubble, proposed as benchmark in [36] is used to establish resolution
requirements for accurately resolving bubble shapes. The theoretically expected

convergence rate of the method is observed in the simulations. Additionally, the order of accuracy of the GHF method is tested for a spherical bubble. Finally, a
110 grid refinement study is conducted for head-on collision and collision against a wall, as these are fundamental mechanisms taking place in bubble-laden channel flow.

In Section 5 the main parallelisation techniques are described and scalability tests are conducted showing efficient parallelization as function of the
115 number of cores and the number of bubbles. The computational domain of the Navier-Stokes equations is decomposed into pencils, as developed in [37]. The communication of the VOF solver relies on Master-Slaves communication and is based on [27, 38], which we here further optimise for hybrid openMP-MPI parallelisation standards [39, 40]. Finally, an application to turbulent bubble-
120 laden channel flow at bulk Reynolds number $Re = 3780$ is shown in Section 6. A summary of the main findings and an overview of possible future developments are provided in Section 7.

2. Mathematical model

The mathematical model used for resolved bubble-laden flows is based on the so called “one-fluid formulation” [10] of the governing equations. A single set of equations is solved for the whole field where differences in the material properties and interfacial terms (e.g., surface tension) are accounted for with the help of a marker function f . In the presence of more than one bubble, an individual marker function f_i is assigned to each bubble, with $f_i = 1$ wherever bubble i is located and $f_i = 0$ elsewhere. Given N bubbles, we advect

$$\frac{\partial f_i}{\partial t} + \mathbf{u} \cdot \nabla f_i = 0 \quad i = 1, \dots, N \quad (1)$$

This formulation, referred to as multiple-marker formulation [27], introduces
125 a convenient numerical setting to account for bubble-bubble interaction (see Section 3.3).

The non-dimensional incompressible Navier-Stokes equation and incompressible continuity equation are used for the simulation of turbulent flows:

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) \right] = -\nabla p + \frac{1}{\text{Fr}^2} \rho \hat{\mathbf{g}} + \frac{1}{\text{Re}} \nabla \cdot (2\mu \mathbf{D}) + \frac{1}{\text{We}} k \mathbf{n} \delta(n), \quad (2)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3)$$

130 where $\mathbf{u}$ is the velocity, p is the pressure, t the time, k is the curvature, $\mathbf{n}$ is the normal vector to the interface, $\mathbf{D}$ the deformation tensor, $\text{Fr} = u/\sqrt{gL}$ the Froude number with L a reference length, $\text{Re} = UL/\nu_l$ the Reynolds number with U a reference velocity, $\text{We} = LU^2\rho_l/\sigma$ the Weber number and ρ and μ are respectively the non-dimensional mass density and dynamic viscosity defined as

135

$$\rho = 1 \cdot (1 - f) + \frac{\rho_g}{\rho_l} f \quad (4)$$

$$\mu = 1 \cdot (1 - f) + \frac{\mu_g}{\mu_l} f \quad (5)$$

where ρ_g and ρ_l are the mass density of the gas and liquid phases, respectively, and f is the global volume fraction field derived from the individual markers f_i as $f = \max(f_i)$ [27]. In the next section the attention is turned to solving the set of governing equations numerically.

140

3. Numerical algorithm

In the following subsections we detail the main components of the computational model and discuss the literature context of these algorithms.

3.1. NS equations discretisation

145

The time-marching of Eqs. (2,3) is carried out with the fractional-step method described in [41]. A third order low-storage Runge-Kutta scheme [42] is used for the convection and diffusion terms of Eq. (2), while for the surface tension term the Crank-Nicolson scheme [42] is used. Each time step, going

from t^n to t^{n+1} , is composed of 3 stages. The time levels of the stages are
150 denoted by the index $s = [0, 3]$, where $s = 0$ corresponds to time level t^n and
 $s = 3$ corresponds to time level t^{n+1} . The Runge-Kutta coefficients γ_s and ξ_s
are the ones used in [43, 44] and here reported for completeness: $\gamma = [\frac{8}{15}, \frac{5}{12}, \frac{3}{4}]$,
 $\xi = [0, -\frac{17}{60}, -\frac{5}{12}]$.

In each stage, we first solve numerically for (1) using the algorithm described
and validated in [19], that employs a piece-wise linear reconstruction of the
interface and a geometric evaluation of the volume fraction fluxes. This equation
is solved at the beginning of each stage delivering the solution for f at stage
 t^{s+1} from which the viscosity and the mass density field, as well as the surface
tension term are updated (see section 3.3, 3.4). At this point, a provisional
velocity u^* can be found from

$$\frac{\mathbf{u}^* - \mathbf{u}^s}{\Delta t} + (C^s - V^s)\gamma_s \mathbf{u}^s - (C^{s-1} - V^{s-1})\xi_s \mathbf{u}^{s-1} =$$

$$\alpha_s \left[-\frac{Gp^s}{\rho^{s+1/2}} + \left(\frac{k\mathbf{n}\delta(n)}{\rho\text{We}} \right)^{s+1/2} + \frac{\hat{\mathbf{g}}}{\text{Fr}^2} \right] \quad (6)$$

where $\alpha_s = \gamma_s + \xi_s$, C^s is the discrete convection operator $u \cdot \nabla$ and V^s is the
155 discrete diffusion operator $\frac{1}{\rho^s \text{Re}} \nabla \cdot (2\mu^s \nabla)$. The surface tension term as well
as the mass density $\rho^{s+1/2}$ are computed from the volume fraction field $f^{s+1/2}$,
obtained by linear interpolation between f^s and f^{s+1} . The continuity constraint
(3) requires the solution of the discrete Poisson equation:

$$D \left(\frac{1}{\rho^{s+1/2}} Gp^{s+1} \right) = D \left(\frac{\mathbf{u}^*}{\alpha_s \Delta t} \right) \quad (7)$$

where D is the discrete divergence operator $\nabla \cdot$ and G is the discrete gradient
160 operator ∇ . Eq. 7 involves the solution of a Poisson equation with discontinuous
coefficients. This has been shown to lead to performance problems for standard
Poisson solvers [45]. A modified constant-coefficient Poisson equation can be
derived instead [34] which allows for the implementation of fast Poisson solvers.
This procedure, also adopted in this work, is detailed in Section 3.2. The velocity
165 and the pressure field are then corrected by the following relation:

$$u^{s+1} = u^* - \alpha_s \Delta t \left(\frac{1}{\rho^{s+1/2}} Gp^{s+1} \right) \quad (8)$$

The velocity field u^{s+1} , resulting from correction (8) is numerically divergence free. The velocity at the final stage, obtained from (8), will be the velocity at the new time level $\mathbf{u}^{n+1}$.

The domain is discretised on a three-dimensional Cartesian grid. For wall-
170 bounded turbulent flows the grid spacing is uniform along the stream-wise direction x and the span-wise direction z . Along the wall normal coordinate y , depending on the grid resolution, the spacing can be uniform or refined, e.g., according to the hyperbolic tangent profile described in [33].

A staggered arrangement of the variables is used for the spatial discretisation,
175 where the velocity components are defined at the cell faces while the pressure field and the volume fraction field are defined at the cell centres. This guarantees a strong coupling between pressure and velocity. The volume fraction field is defined at the cell centres so that the advection velocity of the volume fraction fluxes is readily available at the cell faces avoiding interpolations.

The convective term in the momentum equation (6) is discretised in divergence form which follows from the finite volume approach. This requires the
180 interpolation of the velocities to the faces of the control volumes for which a simple weighted average with weights $\frac{1}{2}$ is used for all the derivatives except $\partial(uv)/\partial x$ and $\partial(wv)/\partial z$, for which a finite volume based interpolation is used, as described in [33]. This leads to an energy-conserving convective scheme for
185 uniform and refined grids along the wall normal direction. For two phase flows, the QUICK [46] interpolation scheme of the convective term has additionally been implemented to avoid the presence of unphysical oscillations for increasing Reynolds numbers, as noted by [3].

The diffusion term is discretised by a central second order finite difference
190 scheme for all the derivatives. The values of the viscosity needed at the cell faces are linearly interpolated from the cell centres. The discrete pressure gradient G is a central finite difference operator (two-points) between the nodes neighbouring the cell faces. For example, the x component of the pressure gradient term
195 reads: $(p_{i+1,j,k} - p_{i,j,k})/(\rho_{i+1/2,j,k} h_{c,i})$ where i, j, k identifies a control volume, $h_{c,i}$ is the grid spacing between nodes $i+1$ and i and $\rho_{i+1/2,j,k}$ is the mass den-

sity at cell face $i + 1/2$, computed by linear interpolation from the cell centre values. The discretisation of the surface tension term is more involved and will be described in section 3.4.

200 The discrete divergence operator D acts as a finite difference operator between face values. For example, the term $D_x(\mathbf{u}^*)$ is equal to $(u_{i,j,k}^* - u_{i-1,j,k}^*)/h_{f,i}$ where $h_{f,i}$ is the grid spacing between faces i and $i - 1$. This simple and compact scheme leads to a second order accurate discretisation of the Poisson equation 7 on a uniform grid.

205 The explicit integration of the convective and the diffusion term leads to time step restrictions in order to guarantee stability of the numerical integration. To avoid the solution of a non-linear system the convective term is commonly treated explicitly. This gives a CFL [47] time step restriction $\Delta t_{u\nabla u} \propto \frac{\Delta x}{u}$, which is also applicable for the stability of the discrete volume fraction equation.

210 It is arguable, however, whether it is convenient to treat the diffusion term explicitly since it could lead to a severe restriction on the time step. The viscous term requires a time step $\Delta t_\nu \propto \text{Re}\Delta x^2$. To relate the grid size Δx to the physics of the problem we use the results of homogeneous turbulence theory, for which the minimum dynamically relevant length scale of the flow is given by
 215 $\Delta \eta \approx L\text{Re}^{-3/4}$ with L a characteristic length of the problem. Having solved the problem in dimensionless form, we have by definition the dimensionless velocity $|u| \approx \mathcal{O}(1)$, which leads to $\Delta t_{u\nabla u} \propto \text{Re}^{-3/4}$ while the viscous time step is $\Delta t_\nu \propto \text{Re}^{-1/2}$. For high Re the most limiting restriction is then $\Delta t_{u\nabla u}$ which justifies the explicit treatment of the viscous term. This may not be true in the
 220 proximity of a wall where the flow obeys a different scaling law [1]. However, this was not an issue at the Reynolds numbers of the cases studied in the present work.

For two-phase flows there are other contributions that play a role: a small CFL number may be required for the advection algorithm to be sufficiently
 225 accurate; the presence of the surface tension term introduces another physical scale to the flow given by the capillary waves. The advection algorithm is accurate for CFL numbers close to 1. Moreover, the surface tension term leads

to a time step restriction $\Delta t_\sigma \propto \left[\frac{\tilde{\rho} \Delta x^3}{2\pi\sigma} \right]^{1/2}$ [48] which translates into $\Delta t_\sigma \propto \text{Re}^{-9/8}$, being the most restrictive limitation on the time marching algorithm at sufficiently high Re . It should be noted that, for the discretisation used in the present work, the surface tension term is not an explicit function of the fundamental variable $\mathbf{u}$ but only indirectly via the volume fraction fields f_i . Hence, in the classical Fourier analysis of (6) it appears as an independent source term that does not influence the amplification factor. In fact, the actual Δt can be orders of magnitude higher than Δt_σ , as shown in [49]. However, if one wants to accurately resolve the capillary waves, the time step should be around the value prescribed by Δt_σ , in which case the use of a fully explicit time integration of the convective and viscous terms is the most efficient choice. When $\Delta t < \Delta t_\sigma$ the benefit of an enlarged numerical stability region provided by the *RK3* scheme can however be outweighed by the tighter physical restriction Δt_σ so that it can be more efficient to use an alternative time integration scheme which does not require multiple stages. For this reason, a second order Adams-Bashforth (AB2) [42] scheme is also implemented. This can be simply obtained by reducing the number of stages from 3 to 1 and setting $\gamma_s = 1.5$, $\xi_s = -0.5$.

3.2. Fast Poisson solver for channel flows

The solution of the Poisson equation (7) is well known to be the most time consuming part of the Navier-Stokes solver. Furthermore a standard multigrid based Poisson solver [50] deteriorates significantly for larger discontinuities in the coefficients. This has been a topic of research in the past years and several improvements have been suggested, some of which are summarised in [45]. These methods mainly focus on better interpolation operators for the transfer of data from fine to coarse grids (coarsening operators) and vice-versa (prolongation operators) in the presence of discontinuities. While this is certainly a viable and effective option for problems with general boundary conditions, in the case of periodic channel flow it is worth investigating whether Eq. (7) can be transformed into a constant-coefficient Poisson equation. This would make it possible to use efficient and well-established direct solvers for constant-coefficient Poisson

matrices.

The idea, developed by [34], and also applied by [51], is to split the density
 260 weighted pressure gradient into two terms:

$$\frac{1}{\rho^{n+1}} \nabla p^{n+1} = \frac{1}{\rho_0} \nabla p^{n+1} + \left(\frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla \hat{p} \quad (9)$$

where ρ_0 is constant and the pressure gradient $\nabla \hat{p}$ is an approximation of ∇p^{n+1}
 based on the pressure gradient at previous time levels. To simplify the notation,
 here we assume to use the Adams-Bashforth time integration scheme, but the
 same considerations apply to the RK3 scheme. If $\nabla \hat{p} = \nabla p^{n+1}$, Eq. (9) is
 265 trivially satisfied. Substitution of (9) into (7) leads to:

$$\nabla^2 p^{n+1} = \nabla \cdot \left[\left(1 - \frac{\rho_0}{\rho^{n+1/2}} \right) \nabla \hat{p} \right] + \nabla \cdot \left(\rho_0 \frac{\mathbf{u}^*}{\Delta t} \right) \quad (10)$$

The first approximation, $\nabla \hat{p} = \nabla p^n$, leads to large errors even for short simu-
 lations of rising bubbles in a laminar flow, as shown in [51]. In the same paper
 the linear extrapolation

$$\nabla \hat{p} = 2\nabla p^n - \nabla p^{n-1} \quad (11)$$

is shown to deliver accurate results for a number of cases of falling droplets and
 270 rising bubbles in a wide range of parameters. For this reason interpolation (11)
 has also been adopted here.

The choice of ρ_0 is related to the numerical stability of (10). A formal Fourier
 stability analysis is not easily applicable given the presence of the discontinuous
 coefficients $\rho_0/\rho^{n+1/2}$. However, if one considers the following simplified equa-
 275 tion: only the first term on the right-hand side is relevant for stability since
 it explicitly contains the variable p ; $\hat{p}$ is approximated by p^n ; the mass density
 $\rho^{n+1/2}$ is assumed to be constant with values between ρ_g and ρ_l ; then the ampli-
 fication factor will be $(1 - \rho_0/\rho^{n+1/2})$, which in absolute value is lower or equal
 than 1 when $\rho_0 \leq 2\rho^{n+1/2}$. Setting for example $\rho_0 = \rho_g$ satisfies this simplified
 280 numerical stability criterion, and this is also the choice made in [34, 51]. Even
 though this result is oversimplified, no numerical instability was found in the
 parameter space investigated in this study.

Once the pressure at the new time level $n+1$ is known, the velocity correction is applied to project the velocity field onto the space of divergence free vector fields. Substituting (9) into (8) leads to the following velocity correction:

$$u^{n+1} = u^* - \Delta t \left[\frac{1}{\rho_0} \nabla p^{n+1} + \left(\frac{1}{\rho^{n+1/2}} - \frac{1}{\rho_0} \nabla \hat{p} \right) \right] \quad (12)$$

Summarising, the only modifications applied to the numerical algorithm presented in section 3.1 are the use of (10) instead of (7) and the use of (12) instead of (8).

The major advantage of solving (10) is that the Poisson matrix reduces to a standard constant coefficients matrix, for which several efficient solvers, iterative and direct, are available from single-phase flow studies. In the particular case of channel flow, the periodicity in x and z directions can be exploited, resorting to a Fast-Fourier-Transform-based (FFT-based) Poisson solver. Applying the Fourier Transform in x and z directions, Eq. (10) reduces to:

$$\left(\frac{\partial^2}{\partial y^2} + \lambda_x + \lambda_z \right) p(k_x, y, k_z) = F(q) \quad (13)$$

where λ_x and λ_y are the eigenvalues:

$$\begin{aligned} \lambda_x &= 2 \left(\frac{\cos(2\pi k_x / n_x) - 1}{\Delta_x^2} \right) & k_x &= 0, \dots, n_x - 1 \\ \lambda_z &= 2 \left(\frac{\cos(2\pi k_z / n_z) - 1}{\Delta_z^2} \right) & k_z &= 0, \dots, n_z - 1 \end{aligned}$$

with k_x , k_z the wave numbers, q the right-hand side of (10) and F the two-dimensional Fourier transform operator. The pressure field in the spectral domain $p(k_x, y, k_z)$ can then be found by solving a tri-diagonal system arising from a second order finite difference discretisation of $\partial^2 / \partial y^2$. This algorithm is a well established technique [52] and has been adopted in this work.

3.3. Multiple-marker formulation

The multiple-marker formulation [27] is used in the context of the VOF method in order to handle collisions between bubbles. When a single marker function f is used, bubbles will always merge if one grid cell contains a part

of two bubbles. This is known as numerical coalescence [27]. By using a different marker function for each bubble we can simulate collisions in which the colliding bubbles remain separated, allowing a meaningful bubble interaction event. The numerical formulation implemented is the one described in [27], which for convenience is also described here. In this paper we do not include physical coalescence in the dynamics but focus on collisions among bubbles and between bubbles and walls. Given N bubbles, for each bubble $i = 1, \dots, N$, Eq. (1) is solved using a linear geometrical reconstruction algorithm as explained in details in [19]. A common volume fraction field f is then built as

$$f = \max_{i=1, \dots, N} f_i \quad (14)$$

which implies $0 \leq f \leq 1$. The mass density and the viscosity field are updated via (4) and (5) using (14). The total surface tension term f_σ is taken to be the sum of all the contributions from each bubble:

$$\mathbf{f}_\sigma = \sum_i^N \sigma k_i \delta(n_i) \mathbf{n}_i \quad (15)$$

As the number of bubbles increases, the solution of N equations of the type (1) on the whole domain becomes computationally expensive. A more efficient approach was proposed in [27, 38], and has been adopted in the present work. Each marker function f_i is discretised and solved on a local rectangular box that moves along with the bubble. The underlying velocity field is transferred to this box grid in order to compute the volume fraction fluxes (see also section 5). Only the velocity on the positions relevant for the term $\nabla \cdot (f\mathbf{u})$ is copied. For cells far away from the interface Eq. (1) reduces to $f_i^n = f_i^{n+1}$ for which the computation of the fluxes is not needed since they are trivially equal to zero. The net volume fraction flux will be different from zero only on interface cells and their neighbours. The size of the velocity field of the box is equal to the size (number of cells) of the bubble plus a halo region of 2 computational cells in each direction. This, and the condition $\text{CFL} < 1$, ensures that all cells for which $f^n \neq f^{n+1}$ are within the box associated with each bubble. Similarly, the determining factor for the box size is the length of the stencil required for

the computation of the curvature (see section 3.4) for which a halo region of 3 cells around the bubble shape is used. This approach guarantees a substantial memory saving and reduced computational time.

3.4. Discretisation of surface tension

335 The surface tension term present in (6) is modelled by the Continuous-Surface-Force (CSF) approach. This approach suffers from spurious currents. The relative importance of the surface tension with respect to the viscosity is described by the Laplace number (La) defined as

$$\text{La} = \frac{\sigma \rho D}{\mu^2} \quad (16)$$

where D is the bubble diameter. The Laplace number is related to We and Re 340 by

$$\text{La} = \frac{\text{Re}^2}{\text{We}} \quad (17)$$

Increasing Re , other non-dimensional parameters being kept constant, La increases. Turbulent flows are characterised by high Laplace numbers. This means that the effect of surface tension is increasingly larger than the damping induced by the viscosity, which in turn leads to a large growth of the spurious currents 345 as time evolves. Ultimately, this numerical problem completely deteriorates the physical solution. An effective way of reducing the magnitude of the spurious currents is to accurately compute the curvature k [53], which becomes the essence at high Reynolds numbers. To this end, the height function method (HF) [54, 24] has been implemented. This method consists of computing a set 350 of heights around each interface cell, calculated by summing the volume fraction field along the direction most aligned with the interface normal vector $\mathbf{n}$. For example if $|n_z| > |n_y| > |n_x|$ a 3×3 stencil is constructed in the $x - y$ plane around the interface cell i, j, k ; for each cell in the stencil a height function $h_{i,j}$ can be found as

$$h_{i,j} = \sum_k f_{i,j,k} \Delta z_k \quad (18)$$

355 The sum over k has no a priori limitation but it can extend over the whole bubble domain. However, to optimise the computational complexity, not all

possible values of k are computed but only those necessary to obtain a valid height function, as explained in [35]. Once all the 9 heights are known the curvature can be computed by standard central differencing schemes. The major
 360 advantage with respect to computing the curvature by discretising [48]

$$k = \nabla \cdot (\hat{n}) \quad (19)$$

with

$$\hat{n} = \frac{\nabla \tilde{f}}{|\nabla \tilde{f}|} \quad (20)$$

where $\tilde{f}$ is a smoothed version of f , is that the reconstructed height function field is smooth and hence well-defined on a finite grid, while the gradients of the volume fraction field itself have the inherited problem of derivatives across
 365 a sharp transition region which may cause large numerical errors and lack of convergence. On the contrary, the HF method has been shown to be second-order accurate with grid refinement [24] (see also section 4.2). Furthermore, no regularisation of f is needed, resulting in a more accurate representation of the curvature near sharp edges.

370 The HF method fails at particularly low resolution of bubble shape details [35]. In the interface cells where this happens, a paraboloid-fitted curvature is computed instead. This hierarchical algorithm, called generalised height function (GHF) was developed in [35] and adopted in this work. A quantitative study on the accuracy and convergence rate of the GHF method, as well as a
 375 comparison with curvature values derived from a smoothed volume fraction field is carried out in Section 4. In the same section the magnitude of the spurious currents for the classical standing bubble problem [55] is analysed.

4. Accuracy analysis

The advection algorithm for the volume fraction used in this paper has been
 380 extensively validated in [19]. Here, we focus on the accuracy of the newly implemented GHF method, as a fundamental unit of the multiphase solver. First, in

subsection 4.1 we present grid refinement results for a rising bubble in 2D, while subsection 4.2 is devoted to a spherical bubble. Furthermore, given the main interest of this work which is in wall-bounded flows loaded with a large number of bubbles, bubble collisions and collisions against a wall represent relevant mechanisms. Subsection 4.3.1 and subsection 4.3.2 will focus on an head-on collision and a so-called grazing collision with a stationary wall. It is shown that the GHF method provides results that are considerably more accurate than those of a finite difference approach, making it suitable for application to bubble dynamics.

4.1. Application of the GHF method to a rising bubble in 2D

In order to establish the accuracy of the GHF method for the simulation of the dynamics of a bubble in an external flow we consider the well-known case of a rising bubble in two spatial dimensions. This case was also published in [19]. The most direct assessment of the method is obtained from a grid refinement study in which the accuracy can be directly related to the required computational effort.

In Figure 1 predicted bubble shapes for flow conditions as also adopted in [19] are compared with the reference provided by [36]. The simulations were initialised from a bubble of circular shape which gradually develops into the presented shapes. We observe that a resolution of 10×20 for the simulation domain of 1×2 is clearly insufficient to capture the shape with more than a qualitative agreement. However, for all higher resolutions we do observe a close agreement with the high-resolution reference data. The resolution of 20×40 corresponds to about 10 grid cells per bubble diameter, and results in a shape which is already closely comparable to [36].

To quantify the convergence of the GHF method upon grid refinement we present the L_1 error of the circularity with respect to high-resolution reference data in Figure 2, as defined in [36]. For all grids adopted here, including the coarsest of 10×20 we notice a clear correspondence with second order convergence as shown by the dashed line. This second order convergence is clearly

Figure 1: Contour plot of the volume fraction $f = 0.5$ at the final simulation time $t = 3$ of a rising bubble. Different grid resolutions are shown as described in the legend. The predictions are in good agreement with the reference results (black line) provided in the benchmark [36].

superior to the convergence rate established in [19], and is obtained without any additional smoothing of the near-interface region. The GHF method is computationally more involved than basic finite difference approaches. However, its robust convergence behaviour and the fact that no additional smoothing is
415 needed are clearly in favour of the GHF method.

The application of the GHF method to a rising bubble in 2D yields a first demonstration of the quality of this method. In the next subsection we turn our attention to a spherical bubble in 3D.

4.2. Comparison of the GHF method with a smoothed finite difference approach for a spherical bubble in 3D

In this subsection we focus on the computation of the curvature of a spherical bubble and compare the GHF method with a ‘smoothed finite difference method’, i.e., a standard finite difference method applied to the discretisation
425 of (19) in which an operator is applied to f as described in [19].

Figure 2: L_1 norm of the error in the circularity obtained for the case of a rising bubble. The reference circularity values are those from the finest grid presented in [36]. The dashed line indicates second order convergence.

In Figure 3 we present the approximation of the curvature of a spherical bubble on a grid of $64 \times 64 \times 64$ cells. The exact value of 4 is seen to be well approximated by the GHF method. Not only is this value properly computed, also the numerically obtained value is found to be highly uniform across the interface. This is in marked contrast with the results obtained with the smoothed
430 finite difference method which not only yields a considerable error, but also displays significant variation of the predicted values over the sphere's interface.

The convergence of the curvature predictions is quantitatively summarised in Figure 4, where the GHF method is compared with two options for the smoothed
435 finite difference approach. The option in which the smoothing length is taken equal to a fixed number of grid cells, equal to 3, is found not to converge at all. Adopting a fixed smoothing length as well as when using the GHF method without any additional smoothing shows second order convergence upon grid refinement. This corresponds fully to the theoretical expectations. On the
440 finest grid the curvature delivered by the GHF method is about 2 orders of magnitude more accurate than the one obtained by adopting a fixed smoothing length and standard finite differencing.

Figure 3: Curvature values on a mid plane of a spherical bubble using: the GHF algorithm (left figure); finite differences derivative Eq. (19) (right figure). The exact value equals 4 in this case.

Figure 4: L_1 norm of the curvature field for a spherical bubble. The line with plusses is the result using Eq. (19) with a variable smoothing length of 3 computational cells; the line with triangles is the result using Eq. (19) with a fixed smoothing length; the line with open circles is the result using the GHF method; the dashed line is the second order convergence slope.

Figure 5: Spurious currents vector field after one time-step for the simulation of a standing spherical bubble under zero gravity. In the left figure the curvature is computed by the GHF method; in the right figure the curvature is computed by equation (19). The velocity field obtained employing the GHF method is approximately 2 orders of magnitude smaller and practically not visible on the same scale. The maximum velocity magnitude in the left picture is about 10^{-5} .

The ultimate consequence of errors in the computation of the curvature is concisely illustrated in Figure 5. On the scale of the comparison the GHF
 445 method appears entirely without spurious velocity effects. Quantitatively, we observed that the spurious velocity induced by the GHF method is two orders of magnitude smaller than the velocities produced by the smoothed finite difference approach.

Based on the numerical assessment of the GHF and smoothed finite differ-
 450 ence methods in this section the GHF approach is clearly favoured and will be adopted throughout the rest of this paper. In the next section we more closely investigate the capturing of bubble-bubble and bubble-wall collisions by the GHF approach.

4.3. Simulation of head on collisions and interaction with a wall

455 The presence of bubbles dispersed in a flow has several effects. First of all, the bubbles take up part of the volume and by moving and interacting with the surrounding fluid the presence of the bubbles influences the overall dynamics. This has many consequences for the effective flow properties of the

total multiphase system. In case of sufficiently high bubble volume fractions the
460 occurrence of collisions between bubbles and between bubbles and walls of the
flow domain becomes relevant for the overall flow dynamics. In this subsection
we present the capabilities of the GHF method, combined with the multi-marker
formulation, in capturing bubble collisions.

For computational convenience, the analysis of collisions is conducted on
465 droplets in air. Two spherical droplets are initially placed at a certain distance
from each other and are given a uniform and opposite velocity. The higher
inertia of the liquid region with respect to the surrounding gas quickly drives
the droplets toward each other. This droplet collision setting is often used in
experiments [56]. From a numerical point of view this is fully equivalent to a
470 bubble-bubble collision, where the volume fraction field is simply switched from
1 to 0 in the liquid phase, and from 0 to 1 in the gas phase.

We focus on a particular physical setting for the simulation of the droplet-
droplet and droplet-wall interactions. The parameters are collected in Table 1,
and seen to correspond closely to the physical situation of water droplets in air
475 under standard conditions. The Weber number is chosen in such a way that the
droplets do not break up, and at the same time the collision is strong enough
to lead to significant deformation of the droplets. In the actual case of bubble-
bubble interaction in channel flow, the lower inertia of the gas phase and the
fact that the bubbles move on average along the stream-wise direction results
480 generally in a more gentle collision with marked less deformation. Hence, the
tests proposed in this section are numerically much more challenging and also
relevant for the milder case of bubble-bubble or bubble-wall interaction.

We first turn our attention to droplet-droplet collisions in Subsection 4.3.1
and subsequently describe droplet-wall interactions in Subsection 4.3.2.

Table 1: Dimensionless parameters - droplet collision

Re	We	ρ_l/ρ_g	μ_l/μ_g
20	4.17	1000	100

485 4.3.1. Droplet-droplet collision

A qualitative impression of the progressing collision is presented by a cut through the center plane $z = 0$ in Figure 6. A smooth sequence of physically recognisable droplet shapes is found in the visualisations of the collision. In fact, the multiple marker approach enables the identification of the interfaces
 490 of each of the droplets separately by the contour $f = 0.5$. We selected a large approaching velocity U_0 to create a demanding test case. In this situation there is a slight overlap between the two droplets found to be limited to approximately 1 grid cell. Moreover, we verified that the individual volumes of the two bubbles are conserved to high accuracy throughout the entire collision event. Relative
 495 variations with respect to the initial volume of less than $1 \cdot 10^{-5}$ were observed at the selected spatial resolution of $64 \times 48 \times 48$ grid points.

The induced velocity field inside and outside the droplets is included in two ways. First, we show the vortical structure of the velocity field with four clockwise/anti-clockwise rotating vortices in the right/left top and bottom parts
 500 of the domain. Specifically, we plot the (u, v) components as velocity vectors in the plane. In addition, contours of the u component are shown in colour as well. The initial phase where the droplets approach (top left figure) displays a clear ‘squeezing out’ of the air between the droplets, ejected toward the top and bottom, which is particularly large just to the north and the south of the region
 505 of closest proximity. This is combined with a positive u in the left of the domain and a negative u in the right part shown in colour. Somewhat later (top right figure) we observe a situation with highly deformed droplets that already begin to separate. A large, very flat interface between the droplets has formed and the vortical structures have all switched rotation direction compared to the initial
 510 approaching phase. The separation velocity is seen to be largest along the x direction near $y = 1.5$. The two droplets gradually separate (bottom left and bottom right figures) showing a retracting flat interface between the droplets and a particular shape that has formed on both droplets just after the close interaction between the droplets has ended.

Figure 6: Three-dimensional simulation of a droplet-droplet collision, showing the velocity field vectors on a mid plane for different simulation times of a droplet collision: $t = 0.2$, top-left figure; $t = 1$, top-right figure; $t = 1.2$, bottom-left figure; $t = 2$, bottom-right figure. Only one velocity vector every 4 grid cells is shown for the sake of clarity. On the background, the computed droplet shape is reported in black together with the contourplot of the x component of the velocity

515 The curvature as computed by the GHF method is illustrated in Figure 7.
 We observe a very sharp representation of the interface with non-zero values of
 the curvature only at the interface cells, i.e., in grid cells directly adjacent to
 the $f = 0.5$ contours of the volume fraction fields associated with the droplets.
 Large variations of the curvature are found along the droplet interfaces, which
 520 are well captured by the method. No signs of numerically induced wiggles in
 the curvature along the contours are observed.

Figure 7: Curvature value computed using the GHF method, for the simulation times: $t = 0.2$, left figure; $t = 1$, right figure.

The convergence of the droplet shapes with grid refinement is illustrated in Figure 8. We compare findings on four different grids that are systematically refined by a factor of two in each coordinate direction. While even the coarsest grid
 525 displays clearly recognisable droplet shapes at the point of high deformation, the strength of the GHF method is best appreciated by the rapid convergence of the droplet shapes with grid refinement. Results on the two finest grids correspond very closely with each other, and display a nearly perfectly overlapping contour of both droplet interfaces. This high level of correspondence allows to
 530 quantify the order of convergence, as shown in Figure 9. The L_1 norm of the difference between the volume fraction field on a given grid g_0, g_1, g_2 compared to the result on the finest grid g_3 shows a clear second order convergence, as is expected theoretically.

Figure 8: Contourplot $f = 0.5$ on a mid-plane of the collision test at $t = 1$ for different grid resolutions: $g_0 = 32 \times 24 \times 24$; $g_1 = 64 \times 48 \times 48$; $g_2 = 128 \times 96 \times 96$; $g_3 = 256 \times 192 \times 192$.

Figure 9: L_1 norm of the volume fraction field error as a function of the grid refinement (black line). The second order line is reported for comparison (dashed line).

4.3.2. Droplet-wall collision

535 We proceed with a description of the capturing of a droplet-wall collision by the GHF method. For that purpose we introduce a spherical droplet close to the channel wall and initialise the velocity inside the droplet such that it approaches the wall under an angle. A physically relevant simulation of the droplet-wall interaction requires the introduction of a contact angle model. Numerically, 540 this translates to an imposed interface slope for those interface cells adjacent to the wall. A detailed description of a contact angle model for the VOF method is given in [57]. While this is certainly a possible approach, in the turbulent channel bubbly flows, imposing a specific contact angle at the wall has a small effect on the mean properties of the flow, since it only affects the shape in the 545 very proximity of the wall. What is however important, as also noticed in [6], is to avoid an artificial sticking of the bubbles at the wall. This happens when a zero-gradient boundary condition for the volume fraction field is used at the wall. Taking in mind Eq. (20), $\nabla f \cdot n_w = 0$, where n_w is the normal to the wall, leads to an interface normal always orthogonal to the wall surface. This does 550 not enable bouncing, causing an agglomeration of bubbles at the wall, which is not always physical. In order to numerically simulate a bouncing event, a value of $f = 0$ is assigned to the halo region next to the wall. The gradient of the volume fraction, i.e., the normal vector to the interface, will then be non-zero in the cells containing an interface. This prevents the interface component 555 normal to wall from being zero. The method is able to numerically simulate a meaningful bouncing event, as shown below.

The sequence of events that illustrates the collision is shown qualitatively in Figure 10, displaying a characteristic flattening of the part of the interface that is squeezed against the wall during the droplet-wall interaction. The complex 560 vortical structure in the (u, v) velocity field is clearly captured in the simulation together with the asymmetrically deformed droplet shapes that are found in the course of the droplet-wall collision.

The characteristic shape of the droplet at the moment of strong deformation

Figure 10: Velocity field vectors on a mid plane for different simulation times of a droplet collision with a wall: $t = 0.3$, top-left figure; $t = 0.9$, top-right figure; $t = 1.5$, bottom-left figure; $t = 3$, bottom-right figure. The wall corresponds to $y = 2$. Only one velocity vector every 4 grid cells is shown for sake of clarity. On the background, the computed droplet shape is reported in black together with the contourplot of the x component of the velocity

is well captured with refined spatial resolution as shown in Figure 11, and, as
 565 in the case of droplet-droplet collision, we can readily establish a clear second
 order convergence as is summarised in Figure 12.

Figure 11: Contourplot $f = 0.5$ on a mid-plane of the collision with a wall (located at $y = 2$) at $t = 0.9$ for different grid resolutions: $g_0 = 32 \times 16 \times 24$; $g_1 = 64 \times 32 \times 48$; $g_2 = 128 \times 64 \times 96$; $g_3 = 256 \times 128 \times 192$

Figure 12: L_1 norm of the volume fraction field error as a function of the grid refinement (black line). The second order line is shown for comparison purpose (dashed line)

This analysis has revealed a successful simulation of these collision events,
 which increases confidence in the application of this method to a turbulent
 bubble-laden channel flow. Of course, the accuracy that can be reached depends
 570 to a large extent on the spatial resolution. For the extreme collision events

that were investigated here with very high droplet deformations we found good convergence on the g_1 and g_2 grids, which cover the initial droplet diameter with 16 and 32 grid cells approximately. This range of resolutions is maintained in the simulations of turbulent bubble-laden channel flow shown in the Section 6.

575 Having analysed the accuracy of the solver we turn our attention to the parallel implementation and performance in the next section.

5. Parallelisation and computational cost

The parallelisation standard used for the developed solver follows a hybrid MPI-openMP framework [40, 39]. The choice of the data distribution is strongly
580 dependent on the computational grid topology and the algorithms used for the numerical solution of the governing equations. The integration of Eq. (6) is done by a fully explicit method that only involves computations on local stencils without preferential direction of update, suggesting a natural equipartition configuration of the load. The choice of the particular decomposition is dictated
585 by the solution algorithm of the Poisson equation (10).

In the particular case of channel flow, the periodicity in two directions can be exploited, resorting to a Fast-Fourier-Transform-based (FFT-based) Poisson solver. This is a well-established technique, also detailed in [52]. The domain is decomposed into 2D pencils (Figure 13), as presented in [37], which allows for
590 an efficient implementation of the Fourier transforms in the periodic directions. An MPI process is assigned to each pencil. Within a pencil a number of serial FFTs are distributed across threads. This procedure, combined with a fully explicit time integration of the momentum equation, leads to an efficient and also massively scalable Navier-Stokes solver, as will be elaborated and illustrated
595 momentarily.

To define the most efficient starting pencil configuration the bubble distribution in the flow has to be taken into account. More precisely, for a downflow channel the bubbles tend to agglomerate in the core of the channel due to the lift force, while the opposite effect is observed for upflow [58]. This suggests that

Figure 13: Example of x -pencil domain decomposition using a 4×4 processor grid.

an MPI subdomain partition along the wall-normal y -direction is not optimal since some of the subdomains, close to the wall for downflow and near the core of the channel for upflow, will not contribute to the advection of the volume fraction. Hence, a y -pencil decomposition will be set as the starting MPI domain decomposition, which we denote as start-pencil. The steps to perform the fast Poisson solver are as follows (see Section 3.2 for the notation):

1. Transposition from start-pencil to x -pencil. Fourier transform of $q(x, y, z)$ in x , with q the right-hand side of (10).
2. Transposition from x -pencil to z -pencil. Fourier transform of $q(k_x, y, z)$ in z .
3. Transposition from z -pencil to y -pencil. Solution of the linear system of Eq. (13).
4. Transposition from y -pencil to z -pencil. Inverse Fourier transform of $p(k_x, y, k_z)$ in z .
5. Transposition from z -pencil to x -pencil. Inverse Fourier transform of $p(k_x, y, z)$ in x .
6. Transposition from x -pencil to start-pencil

The Fourier transforms are performed with the optimised library FFTW [59, 60], while the solution of the tridiagonal system is obtained by using the library LAPACK [61]. A total of 6 pencil decomposition transpositions are performed

620 for every Poisson equation solution. The data transfer is managed entirely by
the optimised collective communication routines `MPI_ALLTOALLV`.

The data transfer at the MPI process boundaries is managed by exchanging
information through ghost-cells. A halo of cells is built around the subdomain
in which the data belonging to neighbouring processors is stored. In order to
625 optimise the amount of data transferred a variable halo size is used for different
fields consistent with the depth of the related finite differencing operator. The
QUICK interpolation scheme, used in the momentum convection term requires
information from 2 upstream nodes, hence an halo of depth 2 cells will be used at
the boundaries. The other finite differencing schemes only require information
630 on the nearest neighbours, thus a halo region of depth of only 1 cell is used. The
data exchange between the halos is carried out by non-blocking communication
`MPI_ISSEND`, `MPI_Irecv`.

In Fig. 14 the averaged wall clock time per time-step of the Navier-Stokes
solver is shown for two different grid resolutions: grid g_1 with 126 million grid
635 points (red line) and grid g_2 with 1 billion grid points (blue line).

Figure 14: Wall clock time C_t per time-step as a function of the number of cores for grid g_1 (red line) corresponding to about 126 million points, and for grid g_2 (blue line) corresponding to about 1 billion points. The dashed line indicates linear scaling.

The solver in all its complexity, including variable material properties in the gov-

erning equations, shows a consistent scaling close to the ideal linear behaviour at all numbers of cores tested. The highest number of cores (17664) on g_2 has been achieved by a final increase in the number of cores by a factor 1.43 which
640 resulted in a corresponding speed-up of 1.40. Furthermore, a remarkably low computational time (per time-step) of 0.2 seconds is observed at the highest number of cores.

As for the curvature calculation, the GHF method consists of a combination of the HF method and the least squares fitting of a paraboloid from which an
645 approximation of the curvature k can be derived. Figure 15 shows the percentage of interface cells whose curvature is computed by the HF method and those whose curvature is computed from the reconstructed paraboloid, for a spherical bubble as a function of the grid points per bubble diameter n_D .

Figure 15: Percentage of curvature computations in the interface cells realised by the HF method (black) and by the least squares fitting of a paraboloid (red), for a spherical bubble as a function of the number of grid points per bubble diameter n_D .

For well resolved bubble shapes ($\approx n_D = 20$), for about 98% of the interface
650 cells the HF method is performed, while the paraboloid fitting is only applied to the remaining 2%. The latter method is computationally more expensive than the standard HF method, since in addition to the computation of interface centroids or heights from which we fit a paraboloid, it requires the solution of a

655 6×6 linear system for each interface cell. However, only a small percentage of least squares fitting is needed for resolved bubbles.

In addition to the curvature computation, the cost of the advection of a single bubble is given by the sum of the time taken by the reconstruction routines and evaluation of the geometrical fluxes [19]. These contributions involve prevalently operations at the interface cells. Ideally, the total computational cost should
660 then scale as n_D^2 . Figure 16 shows that the actual computational cost scales approximately as $n_D^{2.2}$. The slight overhead is mainly due to searching and labelling of the interface cells that are necessary for the main algorithm to function which involve operations over the bubble volume. Resolved bubble shapes ($n_D \approx 15$) require about $1.5 \cdot 10^{-2}$ seconds per time-step.

Figure 16: Computational time of the advection of a single bubble as a function of the number grid point per diameter. A scaling close to factor 2 is observed (dashed line).

665 The advection of the volume fraction equation (1) requires the computation of N hyperbolic equations, one for each bubble. As the number of bubbles increases, if one would solve the equations over the whole domain the simulation will quickly become too expensive. Given the local nature of these equations, each marker function f_i is instead solved on a local rectangular box that moves
670 along with the bubble. The underlying velocity field is transferred to this box in order to compute the volume fraction fluxes. Only the velocity on the positions relevant for the term $\nabla \cdot (f\mathbf{u})$ is copied. The computed surface tension force is

then transferred back to the momentum grid. The size of the box is optimised to reduce the data communication and it is equal to the size of the bubble plus
675 a halo region of 2 cells which allows for a proper definition of finite differences.

At the initial time of the simulation each box is assigned to a Master MPI process based on the position of the centroid of the enclosed bubble. The bubbles are distributed uniformly in space to balance the load. As time evolves the bubbles may occupy different subdomains than the original Master subdomain.
680 The MPI processes belonging to these subdomains are called Slaves of the box. The velocity field on the cells that a bubble happens to occupy at a certain time t is sent from the Slaves to the Master. The Master receives the velocity field and computes the volume fraction f to the next time-level/stage. Based on the new value of f the Master computes the surface tension term which it
685 then sends back, together with f , to the Slaves.

We extend the procedure described in [38] by implementing the following modifications: first, the boxes are periodically reassigned to a new Master whose domain contains the actual centroids of the bubbles. This reduces the amount of data transfer of the velocity, the volume fraction and the surface tension
690 field, back and forth between Master and Slaves. Figure 17 shows the average computational time taken for the volume fraction advection where no reinitialisation is applied (red line) compared with the case in which the redistribution procedure is performed. The results are scaled by the computational time of the advection of a single bubble and the number of cores used is equal to the
695 maximum number of bubbles introduced in the domain. Ideally, in the absence of parallel communication, the scaled computational time should equal 1 until the situation in which the number of bubbles per core exceeds 1. However, an increase of an order of magnitude is observed at about 200 bubbles for boxes randomly distributed in the domain. On the contrary, close to optimal scaling is
700 maintained with a small overhead when the boxes are periodically reinitialised. By doing so, each Master will handle prevalently boxes that are present in its local subdomain. This procedure appeared to be crucial for the overall parallel performance of the solver.

Figure 17: Computational time of the VOF solver, scaled with the computational time of the advection of a single bubble, as a function of the number of bubbles. The black line indicates the results where reinitialisation of the boxes is applied. The red line indicates the results where no reinitialisation of the boxes is applied.

For good performance, a hybrid distributed-shared memory parallelisation is introduced. The benefit of exploiting shared memory is two-fold: a larger number of bubbles, with respect to a full MPI decomposition, will be handled by a given MPI rank due to the smaller number of subdomains. This reduces the expensive Master-Slaves communication across the MPI ranks. Furthermore, in order to send and receive data to or from the correct MPI process, a shared list has to be exchanged. The list contains information about the location of the bubbles on the Eulerian grid, the size of the boxes built around them and the relative Master and Slaves processes. This additional communication cost can be ideally reduced by a factor equal to the number of threads. Nonetheless, as the bubbles move, some communication will always be present at the subdomain boundaries. This means that, for sufficiently large numbers of cores and bubbles, the time taken by exchanging information about the location of the boxes and their corresponding fields becomes dominant. Fig. 18 shows an extension of the results presented in Figure 17 to a larger number of bubbles for an MPI (black line with triangles) and openMP-MPI parallelisation (black line with circles).

Each bubble is resolved by about 20 grid cells along the diameter. This ensures an accurate computation of the shape and the flow scales [2]. The maximum number of cores is set to 12288, which equals the total number of bubbles tested. The hybrid parallelisation clearly performs better at large numbers of bubbles. We can observe a close to optimal scaling up to around 1000 bubbles and a subsequent close to linear increase of the computational time with the number of bubbles. Going from 1 bubble to 12288 bubbles the computational time increases only by a factor of 16. This corresponds to a simulation time of 0.48 seconds per time-step, which is of the same order of magnitude as the time taken by the Navier-Stokes solver for the same computational grid. Hence, the overall computational time of the two solvers combined is adequate to allow for statistically converged simulation of turbulent bubble-laden channel flows at an unprecedented high number of deformable bubbles ($O(10^4)$) on a shared-distributed memory platform.

Figure 18: Computational time of the VOF solver, scaled with the computational time of the advection of a single bubble, as a function of the number of bubbles, for a fixed number of cores which equals the total number of bubbles. The results of a full distributed parallelisation are reported in the line with triangles; the results of an hybrid parallelisation are reported in the line with squares. Optimal performance with Ct/Ct_1 about 1 is established for the latter parallelisation up to 1000 bubbles, while an almost linear increase arises for larger numbers of bubbles.

6. Bubble-laden turbulent channel flow simulations

735 In this section we apply the developed methods and parallelisation algorithms to a turbulent channel flow loaded with a large number of deformable bubbles. This constitutes an interesting study case for the analysis of turbulence modulation in dispersed gas-liquid flows at high gas concentrations.

In order to make a link with previous results reported in literature, we simulate a downflow configuration under the same operational conditions investigated in [3]. The Reynolds number based on the width of the channel is $Re_{2H} = 3780$, the gravity acceleration is $g = 0.1$, the mass density ratio is $\rho_l/\rho_g = 10$, the dynamic viscosity ratio is $\mu_l/\mu_g = 1$, the initial bubble diameter is $D = 0.25$. In [3] the Eotvos number, $Eo = \frac{\rho_l g D^2}{\sigma}$, and the Morton number, $Mo = \frac{g \mu_l^4}{\rho_l \sigma^3}$, are such that the bubbles stay approximately spherical for all simulation time. For strictly fixed bubble shapes alternative and efficient methods have recently been developed using the Immersed Boundary Method, as for example in [8]. The strength of the methodology applied and further developed in this paper, is to handle bubble deformability. Hence, here we reduce the surface tension coefficient σ , with respect to [3], by a factor 10 corresponding to $Eo = 3.1$ and $Mo = 1.5 \cdot 10^{-7}$, which leads to a certain degree of bubble deformation.

In [3] the so-called ‘minimum channel’ flow was simulated for gas volume fractions between 1% and 6% corresponding to a relatively small number of bubbles. In contrast, here we analyse a full scale turbulent channel flow as used often in literature with a total gas volume fraction of $\alpha_t = 20\%$. The chosen channel allows for a comparison with simulations in a well-established computational domain size [29, 31]. Furthermore, the large number of bubbles that we can include enables the investigation of the turbulence statistics, in great detail provided by DNS, and at a gas volume fraction never reported before in literature.

6.1. Initial conditions

A value of $\alpha_t = 20\%$, having fixed the bubble size, is achieved by loading 10000 initially spherical bubbles into the computational domain of size $8\pi H \times 2H \times 8/3\pi H$, with $H = 1$. This corresponds to a channel two times larger along the stream-wise and the span-wise directions, compared to the channel investigated in [31]. The spatial resolution is given by a uniform grid of $1536 \times 160 \times 512$ computational points. This domain size and grid resolution have been tested first for the single-phase channel flow by performing a simulation of the statistically steady state. The resulting velocity field has then been used as initial condition for the bubbly channel flow simulation. Figure 19 left shows the mean velocity profile of the single-phase flow scaled with the friction velocity u_τ as a function of $y^+ = y/\delta_w$, with $\delta_w = \nu/u_\tau$ the wall unit. Figure 19 right shows the rms of the velocity fluctuations scaled with the friction velocity. The results are compared with the ones obtained by an independent spectral method, well validated and used in [31], at a grid resolution $384 \times 193 \times 192$ and a computational domain size of $4\pi \times 2 \times (4/3)\pi$. A good agreement is found for all quantities, with a maximum relative error less than 2%. This shows that the resolution used is appropriate for analysing the quantities of interest in this paper, namely first and second order statistics.

6.2. Comparison of flow features and statistics

For the computation of the statistics of the two-phase flow it is useful to sample the quantities of interest in the liquid phase and in the gas phase. To this end, a definition of volume fraction average is introduced rather than the average over the whole domain in the following way:

$$\langle q \rangle = \frac{\overline{f \cdot q}}{\overline{f}} \quad (21)$$

where f is the volume fraction, q a generic quantity and $\overline{(\cdot)}$, as for the single-phase flow, indicates an average over time and the homogeneous directions. From (21) a natural definition of the root-mean-square (rms) of the fluctuation q' is derived:

Figure 19: Mean stream-wise velocity normalised by the friction velocity as a function of the wall normal coordinate y^+ (left figure); root-mean-square of the velocity fluctuations normalised by the friction velocity as a function of the wall normal coordinate y^+ (right figure). Solid lines: spectral code [31] with channel size $4\pi \times 2 \times (4/3)\pi$ on a $384 \times 193 \times 192$ grid; dashed lines: developed solver with channel size $8\pi \times 2 \times (8/3)\pi$ on a $1536 \times 160 \times 512$ grid. In the right figure the crosses indicate the rms of u' , the circles indicate the rms of w' and the triangles indicate the rms of v' .

$$\begin{aligned}
\text{rms}(q')_f &= \sqrt{\langle (q - \langle q \rangle)^2 \rangle} \\
&= \sqrt{\langle q^2 \rangle - \langle q \rangle^2} \\
&= \sqrt{\frac{f q^2}{f} - \left(\frac{f q}{f}\right)^2}
\end{aligned}
\tag{22}$$

790 The liquid-phase volume fraction averages can simply be derived by substituting
 $(1 - f)$ instead of f in (21). From this point onward, the static quantities pre-
sented are the liquid-phase volume fraction averages indicated by $\langle \cdot \rangle$, unless
stated otherwise. For single-phase flow f is replaced by unity. Furthermore, for
the two-phase flow all the statistics have been averaged also over the left and
795 the right side of the channel to further enhance the statistical convergence of
the data.

In order to keep the flow developed, the volumetric flow rate is kept constant
throughout the whole simulation time by dynamically adjusting the pressure

gradient. Once the flow reaches a statistically steady state, the average wall
800 shear stress τ_w will balance the total weight of the mixture and the imposed
driving force. This can be derived from the global momentum balance along
the stream-wise direction [3]. Fig. 20 shows the shear stress averaged over the
walls as a function of time.

Figure 20: Total wall averaged shear stress τ_w as a function of time. The dashed line is the
initial value of τ_w equal to approximately $3.6 \cdot 10^{-3}$, corresponding to a turbulent channel
flow without bubbles.

A sharp increase of about 47% from the initial value of $3.6 \cdot 10^{-3}$ is observed
805 up until about $t = 50H/U$, after which τ_w varies slightly around $5.3 \cdot 10^{-3}$,
indicating an approximately steady state. Statistics have been collected from
 $t = 50H/U$ to $t = 550H/U$. Previous studies on bubbly downflow configurations
[3] have shown that when the wall shear stress is imposed, a reduction of the
mass flow rate is observed. The liquid is slowed down by the rising motion
810 of the bubbles. Hence, to keep the flow rate constant, given the same initial
conditions, a higher driving force has to be applied. This will be matched by a
corresponding higher τ_w in the statistically steady state, as shown in Fig. 20.

One of the advantages of VOF is its ability to conserve mass since it is based
on the discretisation of the volume fraction conservation equation (1) in its
815 divergence form. However, it is well known that the geometrical time-splitting

advection algorithm as used here, suffers from overshoots or undershoots of f which lead to a small amount of mass loss [16, 19]. Furthermore, small satellites can form in the high shear regions and at relatively low surface tension. These fragments are reset to $f = 0$. In the final simulation time the relative error in
820 the mass conservation is only about 0.04%. While a redistribution procedure of the volume fraction can significantly improve the mass conservation [38], we consider this error acceptable being negligible with respect to other sources of numerical error, e.g. the discretisation error.

A qualitative impression of the flow is given in Fig. 21. In the left figure
825 a snapshot of the bubble distribution is shown together with the contourplot of the stream-wise velocity u for different horizontal planes. In the right figure contour lines of u are shown on a vertical midplane of the channel. Alternating regions of low and high velocity (light green and red regions) due to the rising bubbles are clearly visible.

A measure of the bubble deformability can be derived from the ratio of
830 the principal second moments of inertia of the bubble shapes. In particular, a deformation indicator r_I is defined as the square root of the maximum and minimum of the eigenvalues of the second order inertia tensor [5]. Figure 22 shows r_I of the bubbles as a function of y at $t = 550H/U$. The average r_I
835 is 1.17 which indicates close to spherical shapes ($r_I = 1$). Only 4.7% of the bubbles are characterised by $r_I > 1.3$ where significant deviations from the spherical shape are observed. This translates in an overall behaviour analogous to that of spherical/ellipsoidal bubbles where the volume fraction distribution is mainly influenced by the lift force [5].

The deformability of the bubbles is further shown in Figure 23 where the
840 average volume fraction field as a function of the wall normal coordinate y is shown. Due to the lift force the bubbles are mainly present in the core region, leaving a comparably bubble-free region near the wall [3, 58]. A peak in the volume fraction around $y = 0.34$ is visible, similarly to what was found in [3]
845 and in the experiments [62, 63].

The mean stream-wise velocity profile is reported in Figure 24, as a function

Figure 21: Instantaneous bubble distribution in the vertical channel (left figure) and contour levels of the stream-wise velocity component on a mid-plane (right figure) of the bubbly channel flow at $Re_{2H} = 3780$. The bubbles are represented by grey objects, which are the computed isosurfaces $f = 0.5$. The channel is loaded with 10000 bubbles, corresponding to $\alpha_t = 20\%$.

Figure 22: Ratio of the square root of the principal moments of inertia of the bubbles as a function of the wall normal coordinate y .

Figure 23: Average volume fraction field as a function of the wall normal coordinate y .

of y (left figure) and y^+ (right figure), and compared with the corresponding single-phase flow (dashed line) at the same Re_{2H} . At the statistically steady state the average velocity is approximately constant, and equal to 0.69 in the core of the channel. As expected, the mean shear becomes zero at equilibrium preventing the lateral migration of the bubbles [58]. The profiles start to diverge at $y^+ = 13$ where no logarithmic layer appears to be present for the bubble laden channel. A small decrease of $\langle u \rangle$ is observed near $y = 0.34$, the location of the volume fraction peak. In the near-wall region the behaviour is qualitatively the same as for the single-phase channel flow, corresponding to the absence of bubbles in the close proximity of the walls.

Figure 24: Average stream-wise velocity as a function of the wall normal coordinate on a linear scale (left figure) and on a logarithmic scale (right figure). The dashed line is the mean velocity profile of the corresponding single-phase flow.

Figure 25 shows the rms of u' (left-top figure), the rms of v' (right-top figure), the rms of w' (left-bottom figure) and the Reynolds shear stress $\langle u'v' \rangle$. The latter quantity is close to zero along the entire channel height [3]. In the wall region we observe a general and significant reduction of the velocity fluctuations, similar to what was reported in [3] for the highest volume fraction. While in [3] an increase of velocity fluctuations with the volume fraction α_t is shown in the core region at modest values of α_t , at a significantly higher α_t we observed the opposite behaviour. The rms profiles for the bubbly flow are almost everywhere lower than the corresponding profiles for the single-phase flow. Furthermore, the

velocity fluctuations in the center of the channel appear to be quite independent of y , resembling homogeneous turbulence, from $y = 0.4$ with values equal to approximately 0.97, 0.6 and 0.66 respectively for the rms of u' , the rms of v' and the rms of w' .

Figure 25: Second order velocity fluctuation statistics normalised by the wall shear velocity as a function of the wall normal coordinate y : rms of u' (left-top figure); rms of v' (right-top figure); rms of w' (left-bottom figure); $\langle u'v' \rangle$ (right-bottom figure). Single-phase flow (dashed lines).

870 The decrease of fluctuations in the boundary layer region, where the volume
fraction is nearly zero, is not yet fully understood. It appears that the bubbles
interact in such a way that the turbulence structures are destroyed. Since the
velocity of the bubbles is lower than that of the liquid they act essentially
as obstacles at which the flow speed reduces to that of the bubbles. In our
875 simulation this appears to lead to a strong reduction of $\langle u'^2 \rangle$. Furthermore,

from the equations for the turbulence fluctuations in the three directions one can show that only the equation for $\langle u'^2 \rangle$ has a production term. From the stream-wise fluctuation the energy is redistributed among the components $\langle v'^2 \rangle$ and $\langle w'^2 \rangle$, hence leading to lower kinetic energy also in these two components [1].

A more detailed and exhaustive investigation of statistics of turbulent bubbly flow will be the subject of future investigations. Here, we demonstrated that statistically converged DNS of bubble-laden flows at high gas volume fractions becomes feasible with the methodology developed in this paper. Furthermore, the regime of high volume fractions constitutes an interesting parameter space that presents similarities but also substantial differences with respect to low gas volume fraction cases. These simulations were run on a Bull system for a total of 800.000 CPU hours.

The main findings and conclusions will be summarised in the next section.

7. Conclusions

The main achievement documented in this paper is the development of an accurate and computationally efficient method for DNS of turbulent bubbly flow. This was accomplished by a well-considered combination of state-of-the-art methods as well as fast and scalable numerical algorithms coming from both the single-phase and the two-phase flow community. Furthermore, the main problematic aspects of the underlying numerical methods, such as the computation of the surface tension, bubble/droplet collisions and the solution procedure of the variable-coefficient Poisson equation were analysed and discussed in detail. The unique feature of this work is the combined analysis of accuracy, computational cost and parallel scalability of the algorithms presented, which will serve as guideline for future numerical investigations of physical systems characterised by an high number of bubbles. Furthermore, a DNS at an unprecedented high number of bubbles and gas volume fraction was carried out and the mean and fluctuating flow statistics were analysed in some detail. This

905 represents a step forward toward resolved numerical simulations of real bubbly flows.

The accuracy tests on a rising bubble conducted in Section 4 show a clear second order convergence of the L_1 norm of the circularity. A close approximation of the bubble shape at the final time can be recovered starting from 10
910 grid points per diameter. The GHF method was shown to converge to second order accuracy with grid refinement and to lead to significantly more accurate curvature values than those obtained from finite differences of the smoothed volume fraction field. This, in turn, reduces the spurious velocities which is a necessary requirement at high Laplace numbers. Furthermore, for well-resolved
915 bubble shapes (15–20 grid points per diameter) the GHF method is able to calculate the curvature with the standard and computationally cheap HF method in about 98% of the interface cells, while for the remaining 2% a more expensive least squares fitting is used. The combination of the implementation of the reconstruction routines developed by [64] (see also [19]) and the GHF method
920 results in an overall computational time to solve a resolved bubble, including the volume fraction advection, of about $1 \cdot 10^{-2}$ seconds per time-step.

The droplet-droplet collision and the collision against a wall, simulated in 4.3, show how the multiple-marker formulation allows for a physical interaction between droplets or bubbles preventing numerical coalescence. The convergence
925 tests revealed close to second order convergence, with a total relative loss of mass less than $1 \cdot 10^{-5}$ during an entire, quite vigorous collision event.

Section 5 focuses on the parallel performance of the Navier-Stokes solver and the VOF solver. The inter-process communication involved in the solution of Eq. (6) and Eq. (13) is given by the exchange of data in the halo regions to ensure
930 a correct computation of the differential operators, and by the transposition of data among the pencils. These operations have been performed by using, respectively, the non-blocking communication pair `MPI_ISSEND`, `MPI_IRecv` and the optimised `MPI_ALLTOALLV` routine for all the necessary transpositions $x \leftrightarrow y, x \leftrightarrow z, y \leftrightarrow z$. Close to linear scaling is observed up to 17664 cores on a 1
935 billion grid, with a computational time equal to 0.2 seconds per time-step.

As for the VOF solver, the communication cost is due to the exchange of the shared list for the bookkeeping of the bubbles and in the transfer of the relevant fields with the NS solver. The latter has been largely minimised by periodically reassigning the boxes containing the bubbles to a new set of MPI Masters based on their local position. Furthermore, multi-threading has been introduced in order to minimise the computationally expensive Master-Slaves communication. These last two steps are essential for efficiency and constitute an improvement with respect to the original full MPI algorithm proposed in [38]. Close to ideal scaling was found up to $O(1000)$ bubbles with an acceptable overhead of a factor 16 when further increasing to about 12000 bubbles. At this number of bubbles, the total cost of the VOF solver is 0.48 second per time-step. In order to achieve statistically converged simulations we need a sampling over about 500 dimensionless time units, with a typical time-step of 10^{-3} . Assuming, for example, to simulate $O(10^4)$ bubbles on a 1 billion grid using 10^4 cores, from Fig. 14 and Fig. 18 we can deduce a computational time per time-step of about 1 second. This leads to a total of about 1.4 million CPU hours corresponding to a total expense of about 6 days which can be easily realised on present-day computing facilities.

In Section 6 a simulation of a turbulent downflow loaded with 10000 deformable bubbles at a total gas volume fraction of 20% was carried out. First and second order statistics of the velocity field as well as the average volume fraction field were investigated. The bubbles are pushed away from the wall by the lift force, as visible from the $\bar{f}$ profile. At fixed volumetric flow rate, an increase of 47% of the wall shear stress was observed compared to the case without bubble. The equilibrium velocity profile shows a wide flat region in the core of the channel equal to 0.69. These findings are in good agreement with earlier results reported in [3] at smaller computational domain and lower volume fractions. However, at high volume fractions, the rms of all the velocity fluctuation components was found to decrease significantly except in a narrow region close to the centre line.

Ultimately, a very high (currently record) bubble number has been achieved

for DNS of turbulent channel flows. This shows the feasibility of such simulations with the proposed combination of numerical methods and parallelization techniques. It paves the way to future investigations of dispersed bubbly flow at high gas concentrations. This paper extends recent studies on bubbly flows [8, 9] where the effect of surface tension, variable density and viscosity and deformability are either neglected or approximated by resorting to additional models. Hence, the methodology presented here, which simulate all these effects starting from fundamental equations, can be used as reference for validations. Furthermore, the generated database can be analysed as a source of information for deriving averaged equation models [10] on a wide range of gas volume fractions.

Acknowledgements

This work is part of the research programme ‘Boiling flow regime maps for safe designing’ with project number 12386, which was (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

References

- [1] S. B. Pope, Turbulent flows, IOP Publishing, 2001.
- [2] G. Tryggvason, S. Dabiri, B. Aboulhasanzadeh, J. Lu, Multiscale considerations in direct numerical simulations of multiphase flows a, *Physics of Fluids* 25 (3) (2013) 031302.
- [3] J. Lu, G. Tryggvason, Numerical study of turbulent bubbly downflows in a vertical channel, *Physics of Fluids* 18 (10) (2006) 103302.
- [4] J. Lu, G. Tryggvason, Effect of bubble size in turbulent bubbly downflow in a vertical channel, *Chemical Engineering Science* 62 (11) (2007) 3008–3018.

- [5] J. Lu, G. Tryggvason, Effect of bubble deformability in turbulent bubbly upflow in a vertical channel, *Physics of Fluids* 20 (4) (2008) 040701.
- 995 [6] I. A. Bolotnov, K. E. Jansen, D. A. Drew, A. A. Oberai, R. T. Lahey, M. Z. Podowski, Detached direct numerical simulations of turbulent two-phase bubbly channel flow, *International Journal of Multiphase Flow* 37 (6) (2011) 647–659.
- [7] G. Bois, Direct numerical simulation of a turbulent bubbly flow in a vertical channel: Towards an improved second-order reynolds stress model, *Nuclear Engineering and Design*.
1000
- [8] C. Santarelli, J. Fröhlich, Direct numerical simulations of spherical bubbles in vertical turbulent channel flow. influence of bubble size and bidispersity, *International Journal of Multiphase Flow* 81 (2016) 27–45.
- 1005 [9] V. Spandan, V. Meschini, R. Ostilla-Mónico, D. Lohse, G. Querzoli, M. D. de Tullio, R. Verzicco, A parallel interaction potential approach coupled with the immersed boundary method for fully resolved simulations of deformable interfaces and membranes, *Journal of computational physics* 348 (2017) 567–590.
- 1010 [10] A. Prosperetti, G. Tryggvason, Computational methods for multiphase flow, Cambridge University Press, 2007.
- [11] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics* 39 (1) (1981) 201–225. doi:10.1016/0021-9991(81)90145-5.
- 1015 [12] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *Journal of Computational Physics* 100 (1) (1992) 25–37. doi:10.1016/0021-9991(92)90307-K.
- [13] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *Journal of Computational Physics* 114 (1) (1994) 146–159.
1020

- [14] S. Shin, I. Yoon, D. Juric, The local front reconstruction method for direct simulation of two-and three-dimensional multiphase flows, *Journal of Computational Physics* 230 (17) (2011) 6605–6646.
- 1025 [15] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, *Journal of Computational Physics* 162 (2) (2000) 301–337. doi:10.1006/jcph.2000.6537.
- [16] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, *Journal of Computational Physics* 141 (2) (1998) 112–152. doi:10.1006/jcph.1998.5906.
- 1030 [17] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split eulerian–lagrangian advection, *International Journal for Numerical Methods in Fluids* 41 (3) (2003) 251–274. doi:10.1002/flid.431.
- [18] P. Liovic, M. Rudman, J.-L. Liow, D. Lakehal, D. Kothe, A 3d unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction, *Computers & fluids* 35 (10) (2006) 1011–1032. doi:10.1016/j.compfluid.2005.09.003.
- 1035 [19] P. Cifani, W. R. Michalek, G. J. M. Priems, J. G. M. Kuerten, C. W. M. van der Geld, B. J. Geurts, A comparison between the surface compression method and an interface reconstruction method for the vof approach, *Computers & Fluids* 136 (2016) 421–435. doi:10.1016/j.compfluid.2016.06.026.
- 1040 [20] W. F. Noh, P. Woodward, Slic (simple line interface calculation), in: *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics* June 28–July 2, 1976 Twente University, Enschede, Springer, 1976, pp. 330–340.
- 1045 [21] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, W. J. Rider, A high-order projection method for tracking fluid interfaces in variable

- density incompressible flows, *Journal of Computational Physics* 130 (2)
 1050 (1997) 269–282. doi:10.1006/jcph.1996.5590.
- [22] Y. Renardy, M. Renardy, Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method, *Journal of Computational Physics* 183 (2) (2002) 400–421. doi:10.1006/jcph.2002.7190.
- [23] B. P. Leonard, A. P. Lock, M. K. MacVean, Conservative explicit
 1055 unrestricted-time-step multidimensional constancy-preserving advection schemes, *Monthly Weather Review* 124 (11) (1996) 2588–2606.
- [24] S. J. Cummins, M. M. Francois, D. B. Kothe, Estimating curvature from volume fractions, *Computers & Structures* 83 (6) (2005) 425–434. doi:10.1016/j.compstruc.2004.08.017.
- 1060 [25] M. W. Baltussen, J. A. M. Kuipers, N. G. Deen, A critical comparison of surface tension models for the volume of fluid method, *Chemical Engineering Science* 109 (2014) 65–74.
- [26] S. Popinet, Numerical models of surface tension, *Annual Review of Fluid Mechanics* 50 (2018) 1–28.
- 1065 [27] E. Coya-jee, B. J. Boersma, Numerical simulation of drop impact on a liquid–liquid interface with a multiple marker front-capturing method, *Journal of Computational Physics* 228 (12) (2009) 4444–4467.
- [28] P. Cifani, TBFsolver. doi:<https://doi.org/10.5281/zenodo.1222098>.
- [29] J. Kim, P. Moin, R. Moser, Turbulence statistics in fully developed channel
 1070 flow at low Reynolds number, *Journal of Fluid Mechanics* 177 (1987) 133–166.
- [30] R. D. Moser, J. Kim, N. N. Mansour, Direct numerical simulation of turbulent channel flow up to $Re \tau = 590$, *Physics of Fluids* 11 (4) (1999) 943–945.

- 1075 [31] A. W. Vreman, J. G. M. Kuerten, Comparison of direct numerical simulation databases of turbulent channel flow at $Re_\tau = 180$, *Physics of Fluids* 26 (1) (2014) 015102.
- [32] A. W. Vreman, J. G. M. Kuerten, Statistics of spatial derivatives of velocity and pressure in turbulent channel flow, *Physics of Fluids* 26 (8) (2014) 085103.
- 1080 [33] A. W. Vreman, The projection method for the incompressible navier–stokes equations: The pressure near a no-slip wall, *Journal of Computational Physics* 263 (2014) 353–374.
- [34] S. Dong, J. Shen, A time-stepping scheme involving constant coefficient matrices for phase-field simulations of two-phase incompressible flows with large density ratios, *Journal of Computational Physics* 231 (17) (2012) 5788–5804.
- 1085 [35] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, *Journal of Computational Physics* 228 (16) (2009) 5838–5866. doi:doi:10.1016/j.jcp.2009.04.042.
- 1090 [36] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, *International Journal for Numerical Methods in Fluids* 60 (11) (2009) 1259–1288. doi:10.1002/flid.1934.
- [37] E. P. van der Poel, R. Ostilla-Mónico, J. Donners, R. Verzicco, A pencil distributed finite difference code for strongly turbulent wall-bounded flows, *Computers & Fluids* 116 (2015) 10–16.
- 1095 [38] M. Kwakkel, W.-P. Breugem, B. J. Boersma, Extension of a clsvof method for droplet-laden flows with a coalescence/breakup model, *Journal of Computational Physics* 253 (2013) 166–188.

- 1100 [39] L. Dagum, R. Menon, Openmp: an industry standard api for shared-memory programming, *IEEE computational science and engineering* 5 (1) (1998) 46–55.
- [40] W. Gropp, E. Lusk, N. Doss, A. Skjellum, A high-performance, portable implementation of the MPI message passing interface standard, *Parallel computing* 22 (6) (1996) 789–828. doi:10.1016/0167-8191(96)00024-5.
- 1105 [41] M. J. Lee, B. Do Oh, Y. B. Kim, Canonical fractional-step methods and consistent boundary conditions for the incompressible navier–stokes equations, *Journal of Computational Physics* 168 (1) (2001) 73–100.
- [42] P. Wesseling, *Principles of computational fluid dynamics*, Vol. 29, Springer Science & Business Media, 2009.
- 1110 [43] M. M. Rai, P. Moin, Direct simulations of turbulent flow using finite-difference schemes, *Journal of Computational Physics* 96 (1) (1991) 15–53.
- [44] R. Verzicco, P. Orlandi, A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates, *Journal of Computational Physics* 123 (2) (1996) 402–414.
- 1115 [45] T. F. Chan, W. L. Wan, Robust multigrid methods for nonsmooth coefficient elliptic linear systems, *Journal of Computational and Applied Mathematics* 123 (1) (2000) 323–352.
- [46] B. P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, *Computer methods in applied mechanics and engineering* 19 (1) (1979) 59–98.
- 1120 [47] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, *Mathematische Annalen* 100 (1) (1928) 32–74. doi:10.1007/BF01448839.
- [48] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, *Journal of Computational Physics* 100 (2) (1992) 335–354. doi:10.1016/0021-9991(92)90240-Y.
- 1125

- [49] F. Denner, B. G. M. van Wachem, Numerical time-step restrictions as a result of capillary waves, *Journal of Computational Physics* 285 (2015) 24–40.
- [50] P. Wesseling, Introduction to multigrid methods.
- [51] M. S. Dodd, A. Ferrante, A fast pressure-correction method for incompressible two-fluid flows, *Journal of Computational Physics* 273 (2014) 416–434.
- [52] C. Canuto, M. Y. Hussaini, A. M. Quarteroni, A. Thomas Jr, et al., Spectral methods in fluid dynamics, Springer Science & Business Media, 2012.
- [53] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, M. W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, *Journal of Computational Physics* 213 (1) (2006) 141–173. doi:10.1016/j.jcp.2005.08.004.
- [54] M. D. Torrey, L. D. Cloutman, R. C. Mjolsness, C. Hirt, Nasa-vof2d: a computer program for incompressible flows with free surfaces, Tech. rep., Los Alamos National Lab., NM (USA) (1985).
- [55] D. J. E. Harvie, M. R. Davidson, M. Rudman, An analysis of parasitic current generation in volume of fluid simulations, *Applied mathematical modelling* 30 (10) (2006) 1056–1066. doi:10.1016/j.apm.2005.08.015.
- [56] K.-L. Pan, C. K. Law, B. Zhou, Experimental and mechanistic description of merging and bouncing in head-on binary droplet collision, *Journal of Applied Physics* 103 (6) (2008) 064901.
- [57] S. Afkhami, M. Bussmann, Height functions for applying contact angles to 3d vof simulations, *International Journal for Numerical Methods in Fluids* 61 (8) (2009) 827–847.
- [58] J. Lu, S. Biswas, G. Tryggvason, A dns study of laminar bubbly flows in a vertical channel, *International Journal of Multiphase Flow* 32 (6) (2006) 643–660.

- [59] M. Frigo, S. G. Johnson, The fastest fourier transform in the west, Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE (1997).
- [60] M. Frigo, S. G. Johnson, The design and implementation of fftw3, Proceedings of the IEEE 93 (2) (2005) 216–231.
- 1160 [61] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users’ Guide, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
- [62] S. K. Wang, S. J. Lee, O. C. Jones, R. T. Lahey, 3-d turbulence structure
1165 and phase distribution measurements in bubbly two-phase flows, International Journal of Multiphase Flow 13 (3) (1987) 327–343.
- [63] O. N. Kashinsky, V. V. Randin, Downward bubbly gas–liquid flow in a vertical pipe, International Journal of Multiphase Flow 25 (1) (1999) 109–138.
- 1170 [64] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, Journal of Computational Physics 164 (1) (2000) 228–237. doi:10.1006/jcph.2000.6567.