Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 220 (2006) 216–248
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Parallelization of an Euler–Lagrange model using mixed domain decomposition and a mirror domain technique: Application to dispersed gas–liquid two-phase flow

D. Darmana, N.G. Deen *, J.A.M. Kuipers

Fundamentals of Chemical Reaction Engineering, Faculty of Science and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Received 21 October 2005; received in revised form 4 May 2006; accepted 8 May 2006
Available online 7 July 2006

Abstract

We report a parallel algorithm applicable to a Euler–Lagrange model embedding four-way coupling. The model describing the dispersed phase dynamics accounts for bubble–bubble collisions and is parallelized using a *mirror domain* technique while the pressure Poisson equation for the continuous phase is solved using a domain decomposition technique implemented in the PETSc library [S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page: http://www.mcs.anl.gov/petsc, 2001]. The parallel algorithm is verified and it is found that it gives the same results for both phases as compared to the serial algorithm. Furthermore the algorithm shows good scalability up to 32 processors. Using the proposed method, a homogeneous bubbly flow in a laboratory scale bubble column can be simulated at very high gas hold-up (37%) while consuming a reasonable amount of calculation wall time.

Keywords: Bubble column; Discrete bubble model; Four-way coupling; Direct bubble–bubble interaction; Collision model; Coalescence model; Computational fluid dynamics; Parallel computation

1. Introduction

Bubble columns are encountered in a wide range of applications such as the Fischer–Tropsch process for hydrocarbon synthesis, hydrogenation of unsaturated oil, oxidation of hydrocarbons, fermentation and (biological) wastewater treatment. Due to the simplicity in operation, low operating cost and the good mass, as well as heat transfer characteristics, bubble columns are often preferred over other types of reactors. The determination of global parameters such as the integral gas hold-up is of primary importance for scale-up and design purposes. Unfortunately the global parameters are strongly influenced by the local flow phenomena.

* Corresponding author. Tel.: +31 53 489 4138; fax: +31 53 489 2882.
E-mail address: N.G.Deen@utwente.nl (N.G. Deen).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.05.011

Nomenclature

Notation

$\mathscr{B}$	set of bubbles, dimensionless
C	model coefficient, dimensionless
d	diameter, m
E	eccentricity, dimensionless; efficiency, dimensionless
$E\ddot{o}$	Eötvös number, $E\ddot{o} = (\rho_l - \rho_b)\mathbf{g}d_b^2/\sigma$, dimensionless
$\mathbf{F}$	force vector, N
$\mathbf{g}$	gravity acceleration, m s^{-2}
h	film thickness, m
$\mathbf{I}$	unit tensor, dimensionless
m	mass, kg
Mo	Morton number, $Mo = g\mu_l^4(\rho_l - \rho_b)/(\rho_l^2\sigma^3)$, dimensionless
$\mathscr{N}$	set of possible collision partners, dimensionless
$\mathscr{O}$	set of obstacles, dimensionless
p	pressure, N m^{-2}
$\mathscr{P}$	set of processors, dimensionless
R	radius, m
$\mathbf{r}$	bubble position vector, m
Re	Reynolds number, $Re = \rho_l\lvert\mathbf{v} - \mathbf{u}\rvert d_b/\mu_l$, dimensionless
S	speed-up, dimensionless
$\mathbf{S}$	characteristic filtered strain rate, s^{-1}
t	time, s
T	calculation time, s
$\mathbf{u}$	liquid velocity vector, m s^{-1}
$\overline{\mathbf{u}}$	liquid mean velocity, m s^{-1}
$\mathbf{u}'$	liquid velocity fluctuation vector, m s^{-1}
$\mathbf{v}$	bubble velocity vector, m s^{-1}
v	bubble velocity component, m s^{-1}
V	volume, m^3
$\mathscr{W}$	set of neighbor list window, dimensionless
y	distance to the wall, m

Greek letters

ε	volume fraction, dimensionless
δt	time step, s
Δ	subgrid length scale, m
γ	shear rate, s^{-1}
Φ	volume averaged momentum transfer due to interphase forces, N m^{-3}
μ	viscosity, kg m^{-1} s^{-1}
ψ	Lagrangian quantity
Ψ	Eulerian quantity
ρ	density, kg m^{-3}
σ	interfacial tension, N m^{-1}
τ	stress tensor, N m^{-2}; film drainage time, s
ω	mapping function, dimensionless

Indices

b	bubble
c	coalescence

D	drag
e	event
eff	effective
G	gravity
l	liquid
L	lift
n	normal direction
P	pressure
$\mathscr{P}$	processor
rel	relative
s	serial
S	subgrid
T	turbulent
VM	virtual mass
W	wall

The behavior of bubble columns is quite complex and detailed understanding of its dynamics is lacking. As the local properties in the two-phase flow have proven to be difficult to measure in industrial equipment, there is a growing interest during the last decades to develop models which can accurately predict the detailed characteristics of bubble columns.

Due to advances in computer hardware and numerical solution methods computational fluid dynamics (CFD) has emerged as a powerful tool for both scientists and engineers. Two models are widely used for describing hydrodynamics of bubble columns, i.e. the Euler–Euler (E–E) model and Euler–Lagrange (E–L) model. The E–E model employs the volume or ensemble averaged mass and momentum conservation equations to describe the time dependent motion of both phases [2–4]. The number of bubbles present in a computational cell is represented by a volume fraction and the information on the bubble size distribution is often obtained by incorporating population balance equations, which take into account break-up and coalescence of bubbles as well as growth or shrinkage of bubbles as a result of mass transfer.

The E–L model on the other hand adopts a continuum description for the liquid phase and additionally tracks each individual bubble using Newtonian equations of motion taking into account the four-way interaction, i.e. the mutual bubble–liquid and the mutual bubble–bubble or bubble–wall interaction. This allows for a direct consideration of additional effects related to bubble–bubble and bubble–liquid interaction. Mass transfer with and without chemical reaction, bubble coalescence and re-dispersion can be incorporated directly [3,5–7]. Unlike the E–E model, the E–L model does not require additional models to predict the bubble size distribution since this information is already part of the solution.

One main limitation of the E–L model is the number of bubbles that can be treated since for each individual bubble one equation of motion needs to be solved. Without taking into account direct bubble–bubble interaction Kitagawa et al. [8] and Sommerfeld et al. [9] have succeeded to simulate bubble columns with about 10^5 bubbles simultaneously present in the column. By neglecting the bubble–bubble interaction, however, bubbles could overlap with each other. As a consequence, in dense swarms, considerable overlap between the bubbles can prevail. If the superficial gas velocity is sufficiently high, the local gas fraction can approach unity leading to numerical problems. Sommerfeld [10] claims that if the volume fraction of the gas exceeds 1×10^{-3}, the bubble–bubble interaction becomes so important in describing the fluid dynamics in the bubble column that four-way coupling is needed. Furthermore, the bubble interaction is also required as a first step if bubble coalescence needs to be considered.

By introducing direct bubble–bubble interaction into the model, the computational effort is dramatically increased since the algorithm to solve the collision sequence is both CPU and memory intensive. Furthermore the time step required in the bubble tracking algorithm is no longer determined by the numerical stability criteria but by the time scale of the collision events, which normally are an order of magnitude smaller than the bubble tracking time scale. With a collision model Darmana et al. [7] reported that typically only 10^4 bubbles

can be simulated using a state of the art PC. In combination with the necessity to perform simulations in three dimensions with sufficiently fine spatial and temporal resolution, the method proves to be computationally very demanding and time consuming. On top of that, the model should be able to perform simulations over a sufficiently long time to obtain reasonable statistics, as needed for proper analysis of the column dynamics. This leads to the conclusion that this method is less attractive in handling large scale bubble columns.

Recently parallelization strategies have received considerable attention in the multiphase CFD community [11,12]. By solving a problem in parallel, not only the time required to solve the problem can be reduced significantly, also the problem size that can be handled is increased, since the memory requirements can be distributed. From a numerical point of view, the Eulerian part of the model is easier to solve in parallel since the discretized form of the governing equations will lead to a large linear system of the form $Ax = b$ where the coefficient matrix A is usually sparse. This matrix problem can be solved in parallel efficiently using a domain decomposition technique. Parallelization of the discrete part on the other hand is not straightforward due to the serial nature of the event driven algorithm implemented for tracking bubble–bubble and bubble–wall collisions [13].

In this paper a detailed parallelization strategy is presented for solving a 3D transient Euler–Lagrange model taking into account the four-way interaction. The continuous phase is described using the volume-averaged Navier–Stokes equations, whereas the dispersed phase is described by the Newtonian equations of motion for each individual bubble. The exchange of momentum between the gas and the liquid as well as the momentum exchange between the bubbles (four-way coupling) will be accounted for. The equations describing the bubble motion will be presented in detail, incorporating all relevant forces acting on the bubble. In order to prevent, physically impossible, bubble–bubble overlap, the highly optimized direct bubble–bubble collision algorithm of Hoomans et al. [13] is implemented. The Eulerian part of the model is parallelized using the well-known domain decomposition technique, whereas a so-called *mirror domain technique* is employed to solve the Lagrangian part in a parallel fashion. To allow the parallel code to run in both distributed as well as shared memory architecture, the massage passing interface (MPI) [14,15] paradigm is used.

2. Model formulation

The transient, three-dimensional Euler–Lagrange model described in this paper consists of two coupled parts: a part describing the bubble motion and a part describing the liquid phase motion. The model requires constitutive equations for the forces acting on a bubble. The interaction between the gas and the liquid phase is incorporated via the liquid volume fraction and a source term in the liquid phase momentum equation. The interaction between the bubbles is modeled via a collision model, including bubble bouncing and coalescence.

2.1. Bubble dynamics

The motion of each individual bubble is computed from the bubble force balance. The liquid phase contributions are taken into account by the interphase momentum transfer experienced by each individual bubble. For an incompressible bubble, the equations can be written as:

$$m_b \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \sum \mathbf{F} \tag{1}$$

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v} \tag{2}$$

The net force acting on each individual bubble is calculated by considering all the relevant forces. It is assumed that the net force is composed of separate, uncoupled contributions due to, respectively, gravity, pressure, drag, lift, virtual mass and wall forces:

$$\sum \mathbf{F} = \mathbf{F}_\mathrm{G} + \mathbf{F}_\mathrm{P} + \mathbf{F}_\mathrm{D} + \mathbf{F}_\mathrm{L} + \mathbf{F}_\mathrm{VM} + \mathbf{F}_\mathrm{W} \tag{3}$$

A brief introduction of these forces will be given here. A more detailed discussion of these forces can be found in the review papers of Magnaudet and Eames [16] and Jakobsen et al. [17,18].

The gravity force acting on a bubble in a liquid is given by:

$$\mathbf{F}_{\mathrm{G}} = \rho_{\mathrm{b}} V_{\mathrm{b}} \mathbf{g} \tag{4}$$

The far field pressure force incorporates contributions of the Archimedes buoyancy force, inertial forces and viscous strain and is given by:

$$\mathbf{F}_{\mathrm{P}} = -V_{\mathrm{b}} \nabla p \tag{5}$$

The drag exerted on a bubble rising through a liquid is expressed as:

$$\mathbf{F}_{\mathrm{D}} = -\frac{1}{2} C_{\mathrm{D}} \rho_{\mathrm{l}} \pi R_b^2 |\mathbf{v} - \mathbf{u}| (\mathbf{v} - \mathbf{u}) \tag{6}$$

where the drag coefficient is given by the following relation:

$$C_{\mathrm{D}} = \max\left[\min\left[\frac{16}{Re}(1 + 0.15Re^{0.687}), \frac{48}{Re}\right], \frac{8}{3}\frac{E\ddot{o}}{E\ddot{o} + 4}\right] \tag{7}$$

This drag relation is based on an extensive number of bubble rise velocity measurements and was derived for $10^{-2} < E\ddot{o} < 10^3$; $10^{-14} < Mo < 10^7$ and $10^{-3} < Re < 10^5$, where the Eötvös number, $E\ddot{o} = (\rho_{\mathrm{l}} - \rho_{\mathrm{b}})\mathbf{g} d_{\mathrm{b}}^2/\sigma$ represents the dimensionless size, the Morton number, $Mo = g\mu_{\mathrm{l}}^4(\rho_{\mathrm{l}} - \rho_{\mathrm{b}})/(\rho_{\mathrm{l}}^2\sigma^3)$ is a dimensionless parameter describing the system properties, and the Reynolds number, $Re = \rho_{\mathrm{l}}|\mathbf{v} - \mathbf{u}|d_{\mathrm{b}}/\mu_{\mathrm{l}}$ represents the dimensionless velocity. In this work we use an air–water system, i.e. $Mo = 2.52 \times 10^{-11}$ and bubble sizes ranging from 4 to 10 mm, i.e. $2.1 < E\ddot{o} < 13.4$. The steady relative velocity, $v_{\mathrm{rel}} = |\mathbf{v} - \mathbf{u}|$ can straightforwardly be obtained, when only the gravity, pressure force and drag force are considered, leading to:

$$C_{\mathrm{D}} \frac{1}{2} \rho_{\mathrm{l}} v_{\mathrm{rel}}^2 \frac{\pi d_{\mathrm{b}}^2}{4} = (\rho_{\mathrm{l}} - \rho_{\mathrm{b}}) g_z \frac{\pi d_{\mathrm{b}}^3}{6} \tag{8}$$

For air bubbles of this size rising in water, the drag coefficient given in Eq. (7) reduces to:

$$C_{\mathrm{D}} = \frac{8}{3}\frac{E\ddot{o}}{E\ddot{o} + 4} \tag{9}$$

When Eq. (9) is substituted in Eq. (8), the following relation for the rise velocity is obtained:

$$v_{\mathrm{rel}} = \sqrt{\frac{2\sigma}{d_{\mathrm{b}}\rho_{\mathrm{l}}} + \frac{(\rho_{\mathrm{l}} - \rho_{\mathrm{b}}) g_z d_{\mathrm{b}}}{2\rho_{\mathrm{l}}}} \tag{10}$$

For an air–water system and bubble sizes as indicated earlier, rise velocities of $v_{\mathrm{rel}} \approx 0.25$ m/s are obtained, corresponding to Reynolds numbers in the range of 1000–2500. Note that the influence of shape deformations for large bubbles is implicitly accounted for through the drag relation.

A bubble rising in a non-uniform, liquid flow field experiences a lift force due to vorticity or shear in this flow field. The shear induced lift force acting on a bubble is usually written as [19]:

$$\mathbf{F}_{\mathrm{L}} = -C_{\mathrm{L}} \rho_{\mathrm{l}} V_{\mathrm{b}} (\mathbf{v} - \mathbf{u}) \times \nabla \times \mathbf{u} \tag{11}$$

In this work we use the lift coefficient C_{L} that was derived for $1.39 \leqslant E\ddot{o} \leqslant 5.74$; $5.5 \leqslant \log_{10} Mo \leqslant -2.8$; and shear rates $\gamma < 8.3\ \mathrm{s}^{-1}$ by Tomiyama et al. [20]:

$$C_{\mathrm{L}} = \begin{cases} \min[0.288\tanh(0.121Re), f(E\ddot{o}_d)], & E\ddot{o}_d < 4 \\ f(E\ddot{o}_d), & 4 < E\ddot{o}_d \leqslant 10 \\ -0.29, & E\ddot{o}_d > 10 \end{cases} \tag{12}$$

with

$$f(E\ddot{o}_d) = 0.00105E\ddot{o}_d^3 - 0.0159E\ddot{o}_d^2 - 0.0204E\ddot{o}_d + 0.474 \tag{13}$$

and

$$E\ddot{o}_d = \frac{E\ddot{o}}{E^{2/3}}, \quad E = \frac{1}{1 + 0.163E\ddot{o}^{0.757}} \tag{14}$$

where $E\ddot{o}_d$ is a modified Eötvös number using the horizontal diameter of the bubble that is obtained from the bubble aspect ratio E, which was determined experimentally by Wellek et al. [21] for $E\ddot{o} < 40$ and $Mo \leqslant 10^{-6}$. It is stressed here that small bubbles will tend to move towards the side of low liquid velocities, i.e. wall peaking, whereas large bubbles deform and due to wake effects tend to laterally move in the opposite direction, i.e. core peaking. It is noted here that although Eq. (12) is not strictly valid for the Morton numbers studied in this work, it is considered the best available closure for relatively large bubbles.

Accelerating bubbles experience a resistance, which is described as the virtual mass force [19]:

$$\mathbf{F}_{\mathrm{VM}} = -C_{\mathrm{VM}}\rho_{\mathrm{l}}V_{\mathrm{b}}\left(\frac{\mathrm{D}_{\mathrm{b}}\mathbf{v}}{\mathrm{D}_{\mathrm{b}}t} - \frac{\mathrm{D}_{\mathrm{l}}\mathbf{u}}{\mathrm{D}_{\mathrm{l}}t}\right) \tag{15}$$

where the D/D_t operators denote the substantiative derivatives pertaining to the respective phases. In this work it is assumed that the virtual mass force does not depend on the local void fraction and a virtual mass coefficient of $C_{\mathrm{VM}} = 0.5$ is used.

Bubbles in the vicinity of a solid wall experience a force referred to as the wall force [22]:

$$\mathbf{F}_{\mathrm{W}} = -C_{\mathrm{W}}\frac{d}{2}\left[\frac{1}{y^2} - \frac{1}{(L-y)^2}\right]\rho_{\mathrm{l}}|(\mathbf{v}-\mathbf{u})\cdot\boldsymbol{n}_z|^2\boldsymbol{n}_w \tag{16}$$

where $\mathbf{n}_z$ and $\mathbf{n}_w$, respectively, are the normal unit vectors in the vertical and wall normal direction, L is the dimension of the system in the wall normal direction, and y is the distance to the wall in that direction. Finally, the wall force coefficient C_{W} is given by:

$$C_{\mathrm{W}} = \begin{cases} \exp(-0.933E\ddot{o} + 0.179), & 1 \leqslant E\ddot{o} \leqslant 5 \\ 0.007E\ddot{o} + 0.04, & 5 < E\ddot{o} \leqslant 33 \end{cases} \tag{17}$$

2.2. Liquid phase hydrodynamics

The liquid phase hydrodynamics is described by the volume-averaged Navier–Stokes equations, which consist of continuity and momentum equations. The presence of the bubbles is reflected by the liquid phase volume fraction ε_{l} and the interphase momentum transfer rate Φ:

$$\frac{\partial}{\partial t}(\varepsilon_{\mathrm{l}}\rho_{\mathrm{l}}) + \nabla\cdot\varepsilon_{\mathrm{l}}\rho_{\mathrm{l}}\mathbf{u} = 0 \tag{18}$$

$$\frac{\partial}{\partial t}(\varepsilon_{\mathrm{l}}\rho_{\mathrm{l}}\mathbf{u}) + \nabla\cdot\varepsilon_{\mathrm{l}}\rho_{\mathrm{l}}\mathbf{u}\mathbf{u} = -\varepsilon_{\mathrm{l}}\nabla p - \nabla\cdot\varepsilon_{\mathrm{l}}\tau_{\mathrm{l}} + \varepsilon_{\mathrm{l}}\rho_{\mathrm{l}}\mathbf{g} + \Phi \tag{19}$$

The liquid phase is assumed to be Newtonian, thus the stress tensor τ_{l} can be represented as:

$$\tau_{\mathrm{l}} = -\mu_{\mathrm{eff,l}}\left[((\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\mathrm{T}}) - \frac{2}{3}\mathbf{I}(\nabla\cdot\mathbf{u})\right] \tag{20}$$

where $\mu_{\mathrm{eff,l}}$ is the effective viscosity. In the present model the effective viscosity is composed of two contributions, the molecular viscosity and the turbulent viscosity:

$$\mu_{\mathrm{eff,l}} = \mu_{\mathrm{L,l}} + \mu_{\mathrm{T,l}} \tag{21}$$

where the turbulent viscosity $\mu_{\mathrm{T,l}}$ is calculated using the subgrid scale (SGS) model of Smagorinsky [23]:

$$\mu_{\mathrm{T,l}} = \rho_{\mathrm{l}}(C_{\mathrm{S}}\Delta)^2|\mathbf{S}| \tag{22}$$

where C_{S} represents a model constant with a typical value of 0.1, $\mathbf{S}$ the characteristic filtered strain rate and $\Delta = (V_{\mathrm{cell}})^{1/3}$ the SGS length scale.

2.3. Collision model

In this paper a hard sphere collision model resembling the model developed by Hoomans et al. [13] is used to process the sequence of collisions between bubbles or between bubble and both internal and external obstacles

(i.e. walls) in the computational domain. This model can be illustrated as follows: consider a set of bubbles consisting of N_b bubbles in total with index $\mathscr{B} = \{0, 1, \ldots, N_\mathrm{b} - 1\}$ and a set of obstacles $\mathscr{O}$. For every bubble $\ell \in \mathscr{B}$ we can define a set of possible collision partners $\mathscr{N}(\ell)$ as:

$$\mathscr{N}(\ell) := (\mathscr{B} \cup \mathscr{O}) \setminus \{\ell\} \tag{23}$$

Using the relation reported by Allen and Tildesley [24], we can determine the time required for a bubble ℓ to collide with a collision partner $m \in \mathscr{N}(\ell)$ from their initial positions and velocities (i.e. the collision time):

$$\delta t_{\ell,m} = \frac{-\mathbf{r}_{\ell m} \cdot \mathbf{v}_{\ell m} - \sqrt{(\mathbf{r}_{\ell m} \cdot \mathbf{v}_{\ell m})^2 - \mathbf{v}_{\ell m}^2(-\mathbf{r}_{\ell m}^2 - (R_\ell + R_m)^2)}}{\mathbf{v}_{\ell m}^2} \tag{24}$$

where $\mathbf{r}_{\ell m} = \mathbf{r}_\ell - \mathbf{r}_m$ and $\mathbf{v}_{\ell m} = \mathbf{v}_\ell - \mathbf{v}_m$. Note that if $\mathbf{r}_{\ell m} \cdot \mathbf{v}_{\ell m} > 0$ the bubbles are moving away from each other and will not collide. Furthermore it is assumed that the relative velocity is constant during $\delta t_{\ell,m}$.

For each bubble ℓ the *individual* minimum collision time with other partners

$$\delta t_{\ell,n}^{\mathscr{N}} = \min(\delta t_{\ell,m}) \quad \forall m \in \mathscr{N}(\ell) \tag{25}$$

is determined, where n is the corresponding partner. Subsequently the *global* minimum collision time $\delta t_{a,b}^{\mathrm{e}}$ is determined from all individual minimum collision times:

$$\delta t_{a,b}^{\mathrm{e}} = \min(\delta t_{\ell,n}^{\mathscr{N}}) \quad \forall \ell \in \mathscr{B} \tag{26}$$

where $\delta t_{a,b}^{\mathrm{e}}$ represents the time to the next collision event e. First, all bubble positions will be updated to the instant of the collision, using a simple first-order integration:

$$\mathbf{r}_\ell(t + \delta t_{a,b}^{\mathrm{e}}) = \mathbf{r}_\ell(t) + \mathbf{v}_\ell \delta t_{a,b}^{\mathrm{e}} \tag{27}$$

Following the movement of all bubbles, collision partners a and b are touching and the corresponding collision event is treated subsequently. Two types of collision events can take place: the collision partners can bounce or they can coalesce. The former process will be explained in this section while the latter will be explained in Section 2.4. When the bubbles bounce, the momentum is exchanged based on conservation of momentum. In this case, the velocities of both bubbles are divided into a normal and a tangential component with respect to the line connecting the centers of mass of both bubbles (see Fig. 1). The tangential component does not change due to a collision while the normal component is changed according to the following relation (elastic bouncing):

$$\mathbf{v}_a^* = 2\frac{m_a\mathbf{v}_a + m_b\mathbf{v}_b}{m_a + m_b} - \mathbf{v}_a \tag{28}$$

Updating the velocities of the pair a, b following a collision concludes one cycle of a collision event.

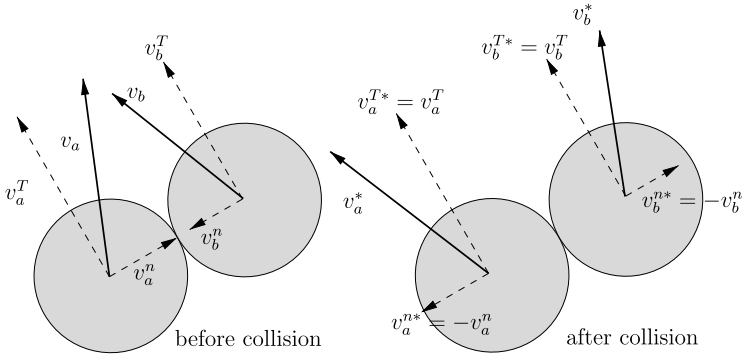


Fig. 1. Configuration of bubble bounce following a collision event.

2.4. Coalescence model

Incorporation of the bubble coalescence mechanism into Euler–Lagrange modeling has been undertaken by Sommerfeld et al. [9] and Van den Hengel et al. [25]. Sommerfeld et al. [9] predicted the collision using a stochastic inter-bubble collision model. Coalescence is incorporated directly by comparing the contact time with the film drainage time. In the approach adopted by Van den Hengel et al. [25] bubble collisions are directly calculated using the method given in the previous section while the coalescence process is predicted using a stochastic approach based on the model of Chesters [26] and Lee et al. [27]. In the present study Sommerfeld's approach to determine coalescence in combination with direct calculation of collision is used. The implementation is straightforward: for a given bubble collision pair a and b predicted by the collision mechanism explained in the previous section the film-drainage time for coalescence to occur is calculated based on the model of Prince and Blanch [28] as follows:

$$\tau_{ab} = \sqrt{\frac{R_{ab}^3 \rho_l}{16\sigma}} \ln\left(\frac{h_0}{h_f}\right) \tag{29}$$

where the initial film thickness for an air–water system was taken as $h_0 = 10^{-4}$ m [29] and the final film thickness just before film breakage was taken as $h_f = 10^{-8}$ m [28,29]. The equivalent bubble radius for a system of two different sized bubbles is obtained from [30]:

$$R_{ab} = 2.0\left(\frac{1}{R_a} + \frac{1}{R_b}\right)^{-1} \tag{30}$$

The contact time between two bubbles is calculated by assuming that it is proportional to a deformation distance divided by the normal component of the collision velocity [9] (see illustration in Fig. 2):

$$t_{ab}^c = \frac{C_c R_{ab}}{|v_a^n - v_b^n|} \tag{31}$$

where C_c represents the deformation distance normalized by the effective bubble radius and should be considered as a (calibration) factor. When the contact time is less than the film breakage time ($t_{ab}^c < \tau_{ab}$) coalescence will not occur and the bubbles will bounce. In all other cases ($t_{ab}^c \geqslant \tau_{ab}$) coalescence will commence and the properties of the new bubble are obtained from the encounter rules summarized in Table 1.

It is noted that despite a vast amount of the literature on this topic, there is still no consensus on the exact formulation of such model. In this work we have made a modest attempt to put together a coalescence model that includes the most important aspects of the coalescence process (i.e. film drainage) employing as much as possible information available from the model (i.e. bubble sizes and approach velocity). The simulations presented in this work should therefore be considered as a demonstration of the possibilities a discrete bubble model offers to study coalescing flows and to demonstrate the impact the inclusion of a coalescence model has on the resulting hydrodynamics. Further research on the closure model for coalescence is still required.

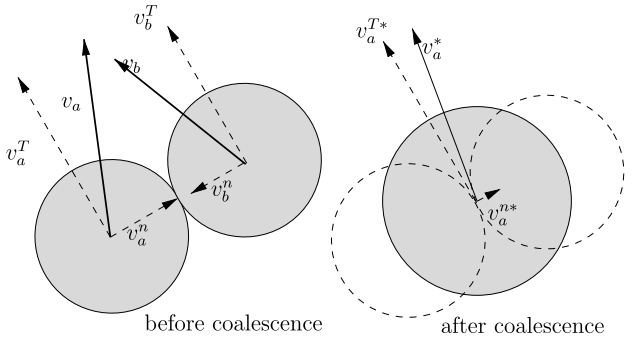


Fig. 2. Configuration of bubble collision and deformation uses in coalescence model (adapted from [9]).

Table 1
Change of bubble properties in a coalescence event

Parameter	Before coalescence	After coalescence
Index	a, b $(a < b)$	a
Mass	m_a, m_b	$m_a^* = m_a + m_b$
Volume	V_a, V_b	$V_a^* = V_a + V_b$
Position	$\mathbf{r}_a$, $\mathbf{r}_b$	$\mathbf{r}_a^* = \frac{\mathbf{r}_a m_a + \mathbf{r}_b m_b}{m_a^*}$
Velocity	$\mathbf{v}_a$, $\mathbf{v}_b$	$\mathbf{v}_a^* = \frac{\mathbf{v}_a m_a + \mathbf{v}_b m_b}{m_a^*}$

3. Numerical solution method

In this section the numerical technique employed to solve the model presented in Section 2 is described.

3.1. Time step

To resolve the time-dependent motion of the gas and the liquid phases, the discrete bubble model employs three different time scales (see Fig. 3). The biggest time step δt_{flow} is used to solve the Navier–Stokes equations to obtain the liquid phase flow field taking into account the interphase coupling. The flow time step is divided into a fixed number of smaller bubble time steps, δt_{bub}. During this time step the forces experienced by each individual bubble are determined. Based on the net force, Eq. (2) is used to determine the bubble acceleration, which is required to obtain the bubble velocity at the end of the bubble time step. Within each bubble time step, the velocity of the bubbles is assumed to change only due to binary collisions between bubbles. Subsequently, an even smaller collision time step $\delta t_{a,b}$ is used to resolve the direct bubble–bubble and bubble–wall interaction. The size of this time step is not fixed and is determined by the sequence of collision events explained in Section 3.3.

3.2. Interphase coupling

The coupling between the gas and the liquid phases appears through the liquid volume fraction ε_{l} and the interphase momentum transfer Φ. Since the liquid phase and the bubbles are defined in different reference frames (i.e., respectively, Eulerian and Lagrangian), a mapping technique which couples the two reference frames is required. This mapping technique translates the Lagrangian bubble quantities to the Eulerian grid, which are required as closure for the liquid phase equations and vice versa (Euler to Lagrange and Lagrange to Euler).

Kitagawa et al. [8] give the following criteria for the mapping function:

(1) It should be a smooth function, i.e. the first derivatives should be continuous.
(2) It should have an absolute maximum around the position where the variable is transferred.

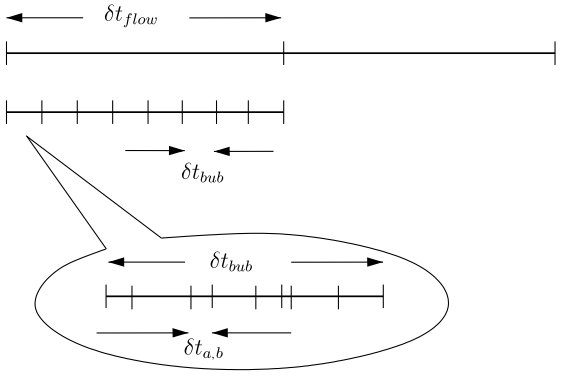


Fig. 3. Schematic representation of the three time steps employed in the Euler–Lagrange discrete bubble model.

(3) For practical reasons it should have a finite domain. At the boundaries of the domain, the function should be zero.
(4) The integral of the function over the entire domain should equal to unity to ensure the conservation of variable being transferred.

Kitagawa et al. [8] propose to use a Lagrangian template function which converts the dispersed phase volume fraction to a spatially differentiable distribution. Using Gaussian and goniometric functions they found that false numerical velocity fluctuations can be removed and the velocity fluctuation of the continuous phase due to the migration of dispersed elements (bubbles) through the Eulerian frame can be accurately captured. Using the same line of thought, Deen et al. [31] proposed to use a fourth-order polynomial function to obtain liquid quantities at the bubble position since the integration of this function is cheaper compared to a Gaussian function or the function proposed by Peskin [32]. They found that by employing this technique, a grid independent solution can be obtained.

Following the successful application of the Lagrangian template technique in the Euler–Lagrange framework, we adopt this technique for our model in the present study as well. The template function used is a clipped fourth-order polynomial function following the work of Deen et al. [31]:

$$\omega(\ell) = \omega(x - r_\ell) = \begin{cases} \frac{15}{16}\left[\frac{(x-r_\ell)^4}{n^5} - 2\frac{(x-r_\ell)^2}{n^3} + \frac{1}{n}\right], & -n \leqslant (x - r_\ell) \leqslant n \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

where $2n$ is the width of the mapping window.

Fig. 4 schematically shows how the Euler–Lagrange two-way coupling is carried out. The template function is constructed at the center mass of a bubble ℓ. This template is moving along with the bubble. In any computational cell j the integral of this function, $\int_{\Omega_j} \omega(\ell)\,\mathrm{d}\Omega$ represents the influence of bubble ℓ on cell j or the influence of the Eulerian value in cell j on bubble ℓ. Note that in 3D space the integral is evaluated as follows:

$$\int_{\Omega_j} \omega(\ell)\,\mathrm{d}\Omega = \int_{\Omega_{j,z}} \int_{\Omega_{j,y}} \int_{\Omega_{j,x}} \omega(x - r_{\ell,x})\omega(y - r_{\ell,y})\omega(z - r_{\ell,z})\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \tag{33}$$

Given bubble ℓ and the width of the mapping window $2n$, the liquid volume fraction in computational cell j is calculated using the following formula:

$$\varepsilon_\mathrm{l}(j) = 1 - \frac{\sum_{\forall \ell \in \mathscr{B}} V_\mathrm{b}(\ell) \int_{\Omega_j} \omega(\ell)\,\mathrm{d}\Omega}{V_\mathrm{cell}} \tag{34}$$

The momentum transfer rate from the bubbles to the liquid in a computational cell j, $\Phi(j)$, can be calculated as:

$$\Phi(j) = \frac{\sum_{\forall \ell \in \mathscr{B}} \phi(\ell) \int_{\Omega_j} \omega(\ell)\,\mathrm{d}\Omega}{V_\mathrm{cell}} \tag{35}$$

where ϕ is the reaction of the momentum transfer exerted on the bubbles, $\phi = -\sum \mathbf{F}$.

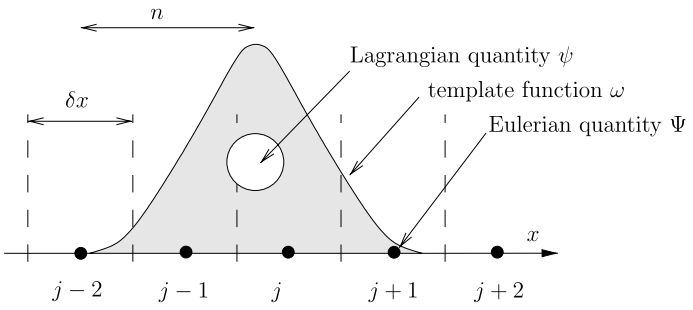


Fig. 4. Lagrangian and Eulerian two-way coupling using a template window function.

The calculation of the force exerted on the bubbles requires Eulerian quantities such as the liquid pressure and velocity to be defined at the position of the bubbles. However, since these quantities are stored in the Eulerian computational cell, again a mapping function should be defined. Using the Lagrangian template function, the Eulerian quantity Ψ at a bubble ℓ position, $\psi(\ell)$ is calculated as:

$$\psi(\ell) = \sum_{\forall j \in \mathscr{C}} \Psi(j) \int_{\Omega_j} \omega(\ell)\, \mathrm{d}\Omega \tag{36}$$

3.3. Bubble dynamics

Solving for the bubble dynamics requires the calculation of the force closures given in Section 2.1. The liquid quantities at the position of bubbles required to calculate this force are determined using the Euler to Lagrange mapping function defined in Eq. (36). Using an explicit first-order scheme, the bubble velocity at the new time level is calculated as follows:

$$\mathbf{v}^{n+1} = \frac{\sum \mathbf{F}^n}{m_b} \delta t_{\mathrm{bub}} + \mathbf{v}^n \tag{37}$$

Next, the bubble volume as well as the interphase force for each bubble is mapped to the Eulerian grid using the relation given in Eqs. (34) and (35), respectively, to obtain the liquid phase volume fraction ε_{l} and the volumetric interphase momentum transfer rate Φ.

Subsequently bubbles are moved by taking into account the interactions between bubbles or between bubbles and confining walls. The method as explained in Section 2.3 is sufficient for this task, however, it is (unnecessarily) expensive for three reasons:

- The set of possible collision partners $\mathscr{N}(\ell)$ consists of all bubbles and obstacles in the entire domain, however, partners which are located far away from the bubble ℓ are unlikely to collide in an immediate event.
- According to Eq. (26) $\delta t_{\ell,m}$ is determined twice since $\delta t_{\ell,m}$ is equal to $\delta t_{m,\ell}$.
- Given N_{e} collision events during bubble time step δt_{bub}, using the method described in Section 2.3 one should perform $N_{\mathrm{e}} \times N_{\mathrm{b}}$ bubble movements as given by Eq. (27) and evaluate N_{e} times the new global minimum collision time $\delta t^{\mathrm{e}}_{a,b}$. This procedure also implies that bubbles that are not involved in collisions will (unnecessarily) be moved N_{e} times on a straight line.

To increase the algorithm efficiency in finding $\delta t^{\mathrm{e}}_{a,b}$ for every collision event, we used the concept of a neighbor list window. Using this concept a dynamic set of $\mathscr{W}(\ell) \subset \mathscr{N}(\ell)$ is introduced, which consists of the neighboring possible collision partners of bubble ℓ that are located within a finite region close to the bubble:

$$\mathscr{W}(\ell) := \mathscr{W}(\ell) \subset \mathscr{N}(\ell) | \forall m \in \mathscr{N}(\ell), \|\mathbf{r}_{\ell m}\| < R_w \tag{38}$$

To ensure that $\delta t_{\ell,m}$ is not calculated twice, we use a simple restriction rule that bubbles with lower index will only "see" neighboring bubbles or obstacles with higher index but not the other way around. Using this rule, $\mathscr{W}(\ell)$ is divided into two unique subsets, namely $\mathscr{W}(\ell)^+$ and $\mathscr{W}(\ell)^-$ where neighbors with index less and greater than index of bubble ℓ are, respectively, stored (see Fig. 5 for illustration of the neighbor list window concept).

Using the concept of neighbor list window, we redefine the global minimum collision time given by Eq. (25) as follows:

$$\delta t^{\mathscr{N}}_{\ell,n} = \min(\delta t_{\ell,m}) \quad \forall m \in \mathscr{W}(\ell)^+ \tag{39}$$

From this point the definition given in Eq. (39) is used instead of Eq. (26) to calculate the individual minimal collision time $\delta t^{\mathscr{N}}_{l,n}$ unless mentioned otherwise. Next, the procedure to move bubbles can be optimized by only moving the bubbles which are actually involved in a collision event and recalculate the individual minimum collision times for the partners that just collided as well as all members of their neighbor list. Using this technique, bubbles will have different timeframes, as the ones which take part in a collision will be moved and their

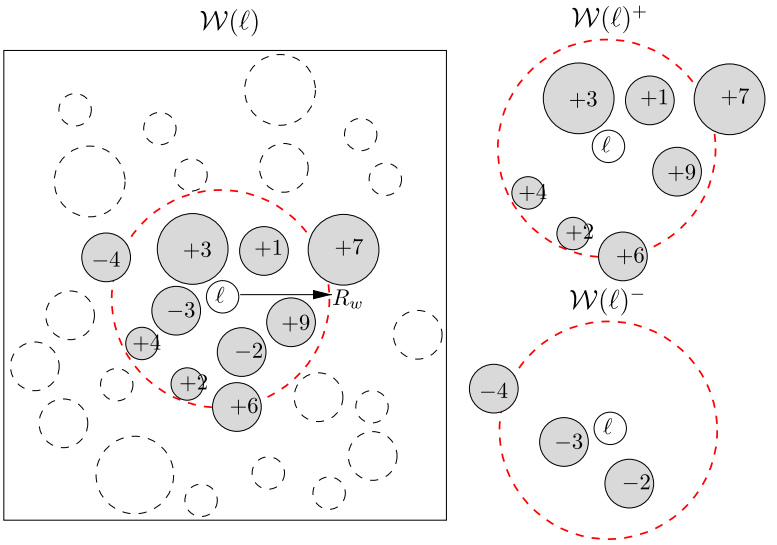


Fig. 5. The neighbor list of bubble ℓ, $\mathscr{W}(\ell)$ (shown as grey bubble) consists of all bubbles (or obstacles) within radius R_w from bubble ℓ. The number represents the index of the bubble relative to bubble ℓ. In the right picture, the neighbor list is divided based on the index higher or lower than the index ℓ.

time will be advanced to the present time whereas the other bubbles remain at their original times and positions.

To keep track of the timeframe for each individual bubble, we introduce a variable $t_\ell;\ 0 \leqslant t_\ell < \delta t_{\text{bub}}\ \forall \ell \in \mathscr{B}$ while $t^{\text{e}};\ 0 \leqslant t^{\text{e}} < \delta t_{\text{bub}}$ is used to account for the accumulation of the global minimum collision time for the whole sequence of collision events e occurred in the time interval δt_{bub}. Furthermore the individual minimum collision time given in Eq. (25) is modified accordingly by taking into account the latest time and positions of the bubbles to calculate the new individual minimum collision time following collision event e:

$$\delta t_{\ell,n}^{\mathscr{N}*} = t^{\text{e}} + \min(\delta t_{\ell,m}^{*}) \quad \forall m \in \mathscr{W}(\ell)^+ \tag{40}$$

where $\delta t_{\ell,m}^{*}$ is the collision time between bubbles ℓ and m evaluated for the bubble position at time t^{e}.

After all the collisions have been resolved, the time and positions for all bubbles are updated to the time level δt_{bub}, which concludes the calculation procedure for the bubbles dynamics. For the sake of clarity the reader can refer to Algorithm 1 for the complete procedure of the optimized bubble dynamics.

3.4. Liquid phase numerical scheme

The numerical solution of the liquid phase conservation equations is based on the SIMPLE algorithm [33] and applied to solve the volume averaged Navier–Stokes equation. The computational cells are labelled by indices (i,j,k) which are located at the cell center and a staggered grid is employed to prevent numerical instability. Using this arrangement the scalar variables are defined at the cell centers whereas the velocities are defined at the cell faces.

Applying first-order time differencing and fully implicit treatment of the convective fluxes, the discretized form of the continuity equation for the continuous phase (Eq. (18)) becomes:

$$\begin{aligned}(\varepsilon_{\text{l}}\rho_{\text{l}})_{i,j,k}^{n+1} - (\varepsilon_{\text{l}}\rho_{\text{l}})_{i,j,k}^{n} + \frac{\delta t}{\delta x}\left\{\langle\varepsilon_{\text{l}}\rho_{\text{l}}u_x\rangle_{i+\frac{1}{2},j,k}^{n+1} - \langle\varepsilon_{\text{l}}\rho_{\text{l}}u_x\rangle_{i-\frac{1}{2},j,k}^{n+1}\right\} + \frac{\delta t}{\delta y}\left\{\langle\varepsilon_{\text{l}}\rho_{\text{l}}u_y\rangle_{i,j+\frac{1}{2},k}^{n+1} - \langle\varepsilon_{\text{l}}\rho_{\text{l}}u_y\rangle_{i,j-\frac{1}{2},k}^{n+1}\right\}\\ + \frac{\delta t}{\delta z}\left\{\langle\varepsilon_{\text{l}}\rho_{\text{l}}u_z\rangle_{i,j,k+\frac{1}{2}}^{n+1} - \langle\varepsilon_{\text{l}}\rho_{\text{l}}u_z\rangle_{i,j,k-\frac{1}{2}}^{n+1}\right\} = 0\end{aligned} \tag{41}$$

where the superscripts n and $n+1$ indicate the old and the new time level, respectively. For the discretization of mass and momentum convection terms, the second-order accurate Barton scheme [34] is applied.

Algorithm 1. Optimized bubble dynamics algorithm.

Initialize by setting $t^{e} = 0,\ t_\ell = 0\ \forall \ell \in \mathscr{B}$
Calculate interphase force, velocity and up-list neighbor: $\forall \ell \in \mathscr{B} \Rightarrow \sum \mathbf{F}(\ell), \mathbf{v}(\ell), \mathscr{W}(\ell)^{+}$
Map bubble volume and interphase momentum to Eulerian cell:
$\forall \ell \in \mathscr{B} \Rightarrow V_{b}(\ell) \rightarrow \varepsilon_{l}; \mathbf{F}(\ell) \rightarrow \Phi$
Calculate *individual* collision time: $\forall \ell \in \mathscr{B} \Rightarrow \delta t^{\mathscr{N}}_{\ell,n}$
Determine *global* minimum collision time: $\delta t^{e}_{a,b}$
while $(t^{e} + \delta t^{e}_{a,b}) < \delta t_{bub}$ **do**
  Advance time: $t^{e} = t^{e} + \delta t^{e}_{a,b}$
  Update position and time for the collision pair:
  $\forall \ell \in \{a, b\} \Rightarrow \mathbf{r}_\ell(t^{e}) = \mathbf{r}_\ell(t_\ell) + \mathbf{v}_\ell \cdot (t^{e} - t_\ell); t_\ell = t^{e}$
  Process collision between pair a and b
  Calculate new *individual* collision time:
  $\forall \ell \in \{a, b, \mathscr{W}(a)^{-}, \mathscr{W}(b)^{-}\} \Rightarrow \delta t^{\mathscr{N}*}_{\ell,n}$
  Determine new *global* minimum collision time: $\delta t^{e}_{a,b}$
endwhile
Move all bubbles: $\forall \ell \in \mathscr{B}$ calculate $\mathbf{r}_\ell(\delta t_{bub}) = \mathbf{r}_\ell(t_\ell) + \mathbf{v}_\ell \cdot (\delta t_{bub} - t_\ell)$

In the discretization of the momentum equation (Eq. (19)) the terms associated with the continuous phase pressure gradients are treated fully implicitly while the interphase momentum transfer and other terms are treated explicitly. The discretization of the continuous phase momentum equation (Eq. (19)) in each direction is, respectively, given by:

$$(\varepsilon_l \rho_l u_x)^{n+1}_{i+\frac{1}{2},j,k} = A^{n}_{i+\frac{1}{2},j,k} - (\varepsilon_l)^{n+1}_{i+\frac{1}{2},j,k} \frac{\delta t}{\delta x}\left\{(p)^{n+1}_{i+1,j,k} - (p)^{n+1}_{i,j,k}\right\} \tag{42}$$

$$(\varepsilon_l \rho_l u_y)^{n+1}_{i,j+\frac{1}{2},k} = B^{n}_{i,j+\frac{1}{2},k} - (\varepsilon_l)^{n+1}_{i,j+\frac{1}{2},k} \frac{\delta t}{\delta y}\left\{(p)^{n+1}_{i,j+1,k} - (p)^{n+1}_{i,j,k}\right\} \tag{43}$$

$$(\varepsilon_l \rho_l u_z)^{n+1}_{i,j,k+\frac{1}{2}} = C^{n}_{i,j,k+\frac{1}{2}} - (\varepsilon_l)^{n+1}_{i,j,k+\frac{1}{2}} \frac{\delta t}{\delta z}\left\{(p)^{n+1}_{i,j,k+1} - (p)^{n+1}_{i,j,k}\right\} \tag{44}$$

where momentum convection, viscous interaction, gravity and interphase momentum transfer have been collected in the explicit terms A^n, B^n and C^n.

The numerical solution of the discretized model equations evolves through a sequence of computational cycles, or time steps, with a duration δt. For each computational cycle the advanced $(n+1)$-level values at time $t + \delta t$ of all key variables have to be calculated through the entire computational domain. This calculation requires the old n-level values at time t, which are known from either the previous computational cycle or the specified initial conditions. Then each computational cycle consists of two distinct phases:

- Calculation of the explicit terms A^n, B^n and C^n in the momentum equation for all interior cells.
- Implicit computation of the pressure for the entire computational mesh with an iterative procedure. This implicit procedure consists of several steps.

The first step involves the calculation of the mass residuals for the liquid phase $D_{i,j,k}$ from the continuity equations (Eq. (18)), for each interior cell:

$$\begin{aligned} D^{*}_{i,j,k} = {} & (\varepsilon_l \rho_l)^{*}_{i,j,k} - (\varepsilon_l \rho_l)^{n}_{i,j,k} + \frac{\delta t}{\delta x}\left\{\langle \varepsilon_l \rho_l u_x\rangle^{*}_{i+\frac{1}{2},j,k} - \langle \varepsilon_l \rho_l u_x\rangle^{*}_{i-\frac{1}{2},j,k}\right\} + \frac{\delta t}{\delta y}\left\{\langle \varepsilon_l \rho_l u_y\rangle^{*}_{i,j+\frac{1}{2},k} - \langle \varepsilon_l \rho_l u_y\rangle^{*}_{i,j-\frac{1}{2},k}\right\} \\ & + \frac{\delta t}{\delta z}\left\{\langle \varepsilon_l \rho_l u_z\rangle^{*}_{i,j,k+\frac{1}{2}} - \langle \varepsilon_l \rho_l u_z\rangle^{*}_{i,j,k-\frac{1}{2}}\right\} \end{aligned} \tag{45}$$

where the superscript $(*)$ refers to the most recently obtained values. If the convergence criterion:

$$D^{*}_{i,j,k} < \mathrm{eps}(\varepsilon_l \rho_l)^{*}_{i,j,k} \tag{46}$$

is not satisfied simultaneously for all internal computational cells, then a whole field pressure correction is calculated using the following relation:

$$J^n_{i-1,j,k}\delta p^{\text{new}}_{i-1,j,k} + J^n_{i+1,j,k}\delta p^{\text{new}}_{i+1,j,k} + J^n_{i,j-1,k}\delta p^{\text{new}}_{i,j-1,k} + J^n_{i,j+1,k}\delta p^{\text{new}}_{i,j+1,k} + J^n_{i,j,k-1}\delta p^{\text{new}}_{i,j,k-1} + J^n_{i,j,k+1}\delta p^{\text{new}}_{i,j,k+1} + J^n_{i,j,k}\delta p^{\text{new}}_{i,j,k} = -D^*_{i,j,k} \tag{47}$$

where J^n represents the Jacobi matrix which contains the derivative of defect D with respect to the liquid pressure at time level n and has been obtained analytically from the continuity equation for the liquid phase in combination with the momentum equations. Applying Eq. (47) for all internal computational cells results in a set of linear equations that can be assembled in the matrix form as:

$$\mathbf{J}^n \cdot \delta\mathbf{p}^{\text{new}} = -\mathbf{D}^* \tag{48}$$

To save computational effort the elements of the Jacobi matrix are evaluated at the old time level. By solving Eq. (48), a pressure correction term is obtained and new pressure is subsequently calculated followed by calculation of the new velocity field.

Provided that the corresponding estimates of the mass residual (Eq. (45)) do not meet the convergence criteria simultaneously for all interior computational cells, the pressure correction equation (Eq. (47)) is again calculated using the updated velocity field to compute the mass residual **D** for all cells. This iterative process is repeated until the convergence criteria is satisfied or the specified maximum allowable number of iterations reached.

4. Model validation

Model validation for the Euler–Lagrange code used in the present study has been conducted by Delnoij et al. [35]. The authors have shown that the model can correctly predict the terminal rise velocity of a single bubble for a given drag coefficient. Validation for bubble plumes was made by Darmana et al. [7] by comparing calculation results from the model with experimental measurement data obtained with particle image velocimetry (PIV) by Deen et al. [31]. The PIV measurements were performed in a 3D bubble column filled with distilled water. The column has a square cross-section ($W \times D$) of $0.15 \times 0.15\ \text{m}^2$ and a height (L) of 0.45 m. Air with a superficial gas velocity of 4.9 mm/s was introduced into the system through a perforated plate. The plate contained 49 holes with a diameter of 1 mm, which were positioned in the middle of the plate at a square pitch of 6.25 mm.

Fig. 6 shows the comparison taken from Darmana et al. [7]. It can be seen here that both instantaneous and time averaged liquid velocities predicted by the model are in good agreement with the experimental data. Furthermore the correctly predicted velocity fluctuations indicate that the experimentally observed meandering of the bubble plume is well predicted by the present model.

5. Parallelization strategy

The dynamic nature of bubbles makes their spatial distribution non-uniform. This implies that the number of collisions is considerably higher in more dense regions when compared with dilute regions [36] which implies that partitioning of the model based on bubble position (i.e. the domain decomposition) cannot give a high parallel efficiency.

In the present model, parallelization for the disperse phase is carried out using a so-called *mirror domain technique*. Contrary to the domain decomposition technique, where each processor holds a unique computational subdomain and synchronizes the data only at the subdomain boundary, each processor in the mirror domain technique holds an identical complete computational dataset (i.e. data is mirrored through all processors). Unique data subsets are determined for each processor and calculations are conducted by each processor only for this subset of data. Since the calculations on each processor are done only for unique portion of data, the data is no longer identical throughout the processors, hence synchronization by interchanging data between processors (creating a new complete mirror dataset) is required after each calculation step. The main advantage of the mirror domain technique as compared to traditional domain decomposition techniques is

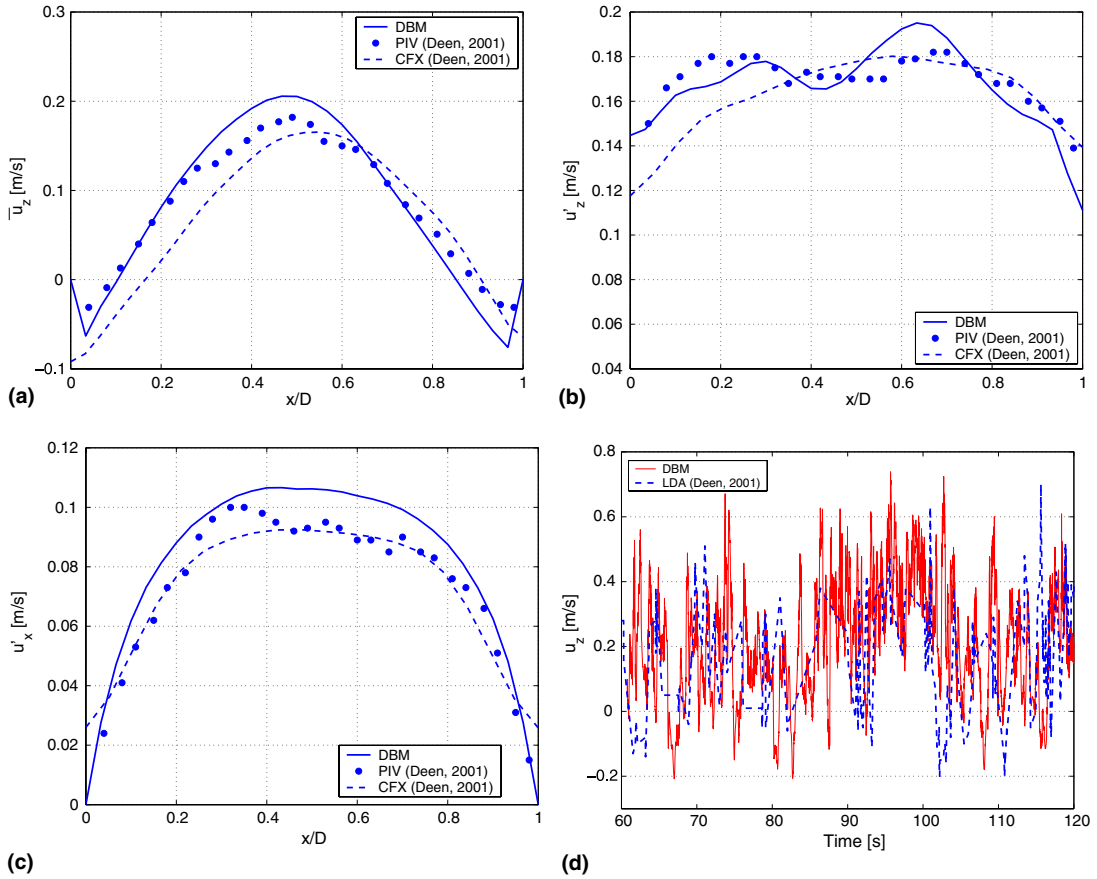


Fig. 6. Figure taken from Darmana et al. [7]. (a–c) Comparison of simulated and experimental liquid profiles, respectively: vertical average velocity ($\overline{u}_z$), vertical velocity fluctuations (u'_z) and horizontal velocity fluctuations (u'_x) at height of $z/H = 0.56$ and a depth $y/W = 0.5$. (d) Time history of the vertical liquid velocity (u_z) at the same position.

that it enables a proper load balancing of the calculations of the Lagrangian variables, since the computational tasks are explicitly evenly divided over the available processors.

This section describes the parallelization strategy of the numerical method explained in Section 3. The model is partitioned and distributed over a set of processors $\mathscr{P} = \{0, 1, \ldots, N_P - 1\}$ using the mirror domain technique for the dispersed phase while a domain decomposition technique is adopted in solving the continuous phase. The interprocessor communications are carried out using the message passing interface (MPI) paradigm.

5.1. Discrete phase

Using the mirror domain technique for the disperse phase, we exploit the fact that for a given identical initial data set of bubble positions and velocities as well as the continuous phase flow field across the processors, the calculation of the bubble dynamics can be performed independently for each individual bubble in a effective parallel fashion. Data uniformity throughout processors is maintained by either calculating part of the data locally and interchange between processors or execute an identical procedure on identical data locally on each processor.

The serial algorithm shown in Algorithm 1 is modified using the mirror domain concept. The resulting parallel algorithm executed on each processor $\mathscr{P}$ is given in Algorithm 2. First the initialization procedure is executed. Subsequently a local bubble list $\mathscr{B}^{\mathscr{P}} \subseteq \mathscr{B}$ is determined. Two types of bubble partitioning are used: based on memory location and based on bubble index using a round-robin rule. To calculate the interphase

force and velocity, the first partitioning technique is used in view of the fact that the message passing interface can transfer continuous data in memory much faster than repeatedly transferring single data. The bubble partitioning using round-robin is used in determining the collision time, since bubbles only *see* other bubbles with a higher index, thus partitioning based on the memory location would only result in poor parallel performance as the processors with a higher index will calculate bubbles with a higher index with less associated possible bubble neighbors, while processors with a lower index will calculate bubbles with a lower index with more bubble associated neighbors.

For all local bubbles, the interphase forces, velocities and up-list neighbors are calculated next, as is illustrated in Fig. 7. The bubble velocities previously calculated locally are combined to obtained a complete set of bubble velocities in all processors. Subsequently the local interphase forces and the bubble volume is mapped to the local Eulerian grid to obtained local liquid volume fraction and interphase momentum transfer. Using the sum operator available in MPI, these local quantities are combined and redistributed to all processors to obtain the global liquid volume fraction and interphase momentum transfer rate.

Subsequently the collision sequence is determined, as illustrated in Fig. 8. First the individual collision time is calculated for all local bubbles partitioned in a round-robin manner. For all local individual collision times previously calculated, the local minimum collision time is determined. By making use of the minimum operator available in MPI, the local minimum collision times of all processors are compared to determine the global minimum collision time and distribute the result to all processors. While the accumulated event time t^{e} is smaller than the bubble time step, all processors execute an identical procedure which consecutively consists of updating the accumulated event time, updating the positions and individual times for the collision partners, and process the collision between partners a and b.

Algorithm 2. The parallel algorithm executed on each processor $\mathscr{P}$ for the optimized bubble dynamics using the mirror domain concept.

Initialize by setting: $t^{\mathrm{e}} = 0,\ t_\ell = 0\ \forall \ell \in \mathscr{B}$
Determine *local* bubble set: $\mathscr{B}^{\mathscr{P}} \subseteq \mathscr{B}$
Calculate *local* interphase force, velocity and up-list neighbor: $\forall \ell \in \mathscr{B}^{\mathscr{P}} \Rightarrow \sum \mathbf{F}(\ell), \mathbf{v}(\ell), \mathscr{W}(\ell)^+$
$\forall \ell \in \mathscr{B}^P$ gather and scatter $\mathbf{v}(\ell)$
Map *local* bubble volume and *local* interphase momentum to the Eulerian cell:
$\forall \ell \in \mathscr{B}^P \Rightarrow V_{\mathrm{b}}(\ell) \rightarrow \varepsilon_{\mathrm{l}}; \mathbf{F}(\ell) \rightarrow \Phi$
Gather (with sum operator) and scatter: ε_{l} and Φ
Calculate *local* individual collision time: $\forall \ell \in \mathscr{B}^{\mathscr{P}} \Rightarrow \delta t^{\mathscr{N}}_{\ell,n}$
Determine *local* minimum collision time: $\delta t^{\mathrm{e}}_{a,b}$
Gather (with minimal operator) and scatter to obtain *global* minimum collision time: $\delta t^{\mathrm{e}}_{a,b}$
while $(t^{\mathrm{e}} + \delta t^{\mathrm{e}}_{a,b}) < \delta t_{\mathrm{bub}}$ **do**
  Advance time: $t^{\mathrm{e}} = t^{\mathrm{e}} + \delta t^{\mathrm{e}}_{a,b}$
  Update position and time for the collision pair:
  $\forall \ell \in \{a, b\} \Rightarrow \mathbf{r}_\ell(t^{\mathrm{e}}) = \mathbf{r}_\ell(t_\ell) + \mathbf{v}_\ell \cdot (t^{\mathrm{e}} - t_\ell); t_\ell = t^{\mathrm{e}}$
  Process collision between pair a and b
  Calculate new *local* individual collision time:
  $\forall \ell \in \{\{a, b, \mathscr{W}(a)^-, \mathscr{W}(b)^-\} \bigcap \mathscr{B}^{\mathscr{P}}\} \Rightarrow \delta t^{\mathscr{N}*}_{\ell,n}$
  Determine new *local* minimum collision time: $\delta t^{\mathrm{e}}_{a,b}$
  Gather (with minimal operator) and scatter to obtain new *global* minimum collision time: $\delta t^{\mathrm{e}}_{a,b}$
end while
Locally move all bubbles: $\forall \ell \in \mathscr{B}^{\mathscr{P}} \Rightarrow \mathbf{r}_\ell(\delta t_{\mathrm{bub}}) = \mathbf{r}_\ell(t_\ell) + \mathbf{v}_\ell \cdot (\delta t_{\mathrm{bub}} - t_\ell)$
$\forall \ell \in \mathscr{B}^P$ gather and scatter $\mathbf{r}(\ell)$

Subsequently, the new local individual collision times are determined for the collision pair and all their down-list neighbors. The calculations however, are carried out only on the processors associated with the considered bubbles. Based on all local individual collision times a new local minimum collision time is determined. Again, using the MPI minimum operator the global minimum collision time is determined and distributed to

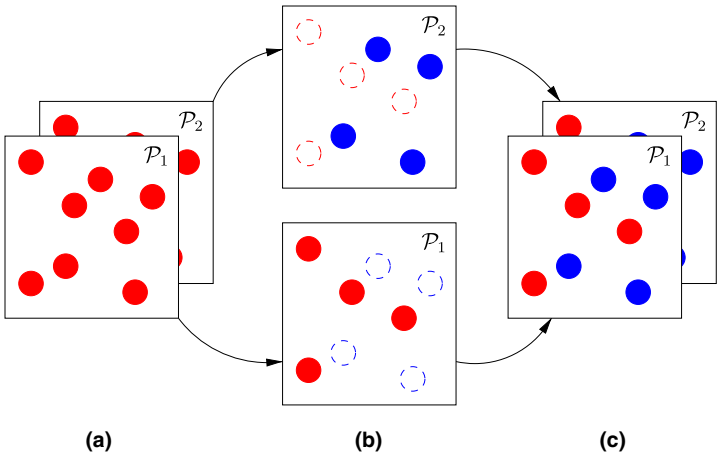


Fig. 7. Illustration of parallel strategy for bubble force calculation, Euler to Lagrange mapping and Lagrange to Euler mapping: (a) mirrored bubble data at all processors $\mathscr{P}_n$; (b) new data for unique subsets of bubbles are calculated in parallel on each processor $\mathscr{P}_n$; (c) updated local data is gathered and mirrored through all processors.

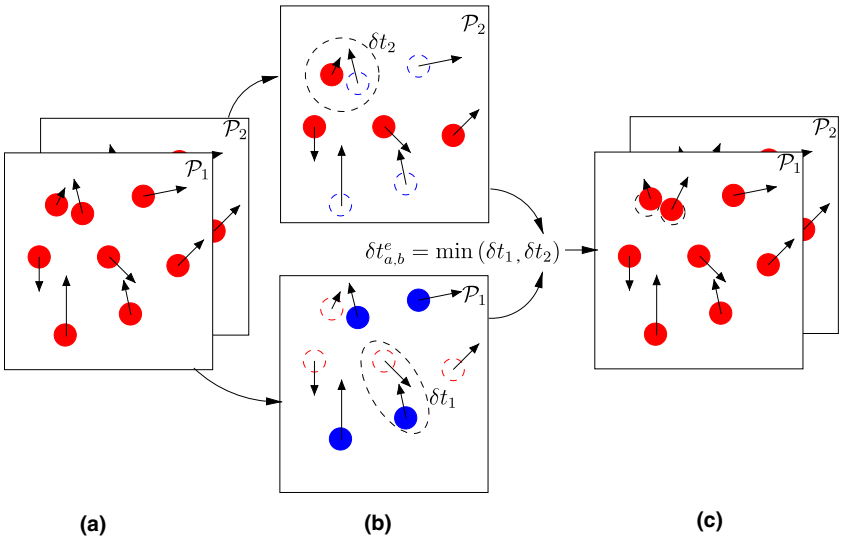


Fig. 8. Illustration of parallel strategy for bubble encounter detection: (a) mirrored bubble velocity and position data at all processors $\mathscr{P}_n$; (b) each processor $\mathscr{P}_n$ calculates the local minimum encounter time for a unique subset of bubbles; (c) processor local minimum encounter times are compared to find the global minimum encounter time $\delta t^e_{a,b}$ followed by updating the bubble positions at all processors by $\delta t^e_{a,b}$.

all processors. These steps are repeated until the next accumulated event time is exceeding the bubble time step. Finally all processors locally move the bubbles to time level δt_{bub} and gather the bubble positions from other processors to obtain a complete set of bubbles positions.

Using the present approach, a total of $4 \times N_{\text{v}}$ variables defined on the Eulerian grid (Φ in three directions $+ \varepsilon_{\text{l}}$) and $6 \times N_{\text{b}}$ variables defined on the Lagrangian bubble positions (velocities and positions in three directions) have to be interchanged between processors outside the while loop for every δt_{bub}, while for every collision event 3 variables consisting of the local minimum collision time and collision partners have to be interchanged. All communications except the ones inside the while loop are carried out in a non-blocking fashion (i.e. overlapping with the calculation) to reduce the effective communication time. The amount of data that has to be communicated for every collision event is very small hence cannot benefit from overlapping communication with calculation. It is noted that there is no memory reduction for the main bubble variables such as

positions and velocities, as these variable have to be known on each processor, however, significant memory reduction arises from the up-list neighbor requirements. For a maximum number of $N^+_{\mathscr{W}}$ up-list neighbors per bubble, a total of $N_b \times N^+_{\mathscr{W}}$ memory blocks should be allocated in a serial calculation. Whereas for a number of $N^{\mathscr{P}}_b$ local bubbles, only $N^{\mathscr{P}}_b \times N^+_{\mathscr{W}}$ memory blocks are required on each processors for the parallel calculation. As the number of main bubble variables is much less than $N^+_{\mathscr{W}}$, one might expect to have a memory reduction factor of $\approx N_P$ by running a simulation in parallel with N_P number of processors.

5.2. Continuous phase

The continuous phase calculation is parallelized by making use of the PETSc library version 2.3.0 [1,37,38]. PETSc is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. It employs the MPI standard for all message-passing communication. PETSc has been used for a wide variety of applications, including computational fluid dynamics, structural dynamics, materials modeling and econometrics. Many of the solvers are appropriate for problems discretized using either structured grids or unstructured grids. In the present study we consider PETSc as a black box hence it will only briefly be discussed. Interested readers are referred to the PETSc user manual or publications on application simulations developed by PETSc users [39–42].

The discrete bubble model uses the linear solver component of PETSc to solve the pressure correction equation given in Eq. (48) and uses the efficient parallel data formats provided by PETSc to store the Jacobian matrix and the defect vector. As the parallel calculation for the discrete phase requires that complete data sets of the liquid velocity and pressure are known on all of the processors, solving the pressure correction matrix using PETSc is straightforward. For instance, one can divide the hepta-diagonal matrix in several block rows according to the number of processors involved in the calculation and fill the corresponding blocks locally as shown in Fig. 9. By calling the PETSc linear solver command in each processor the matrix is solved iteratively in parallel by taking into account coupling with other matrix elements which reside on the other processors. After convergence is reached, the solution block of the linear equations (i.e. the pressure correction terms) are available locally. Subsequently each processor can interchange their local pressure correction terms to obtain a complete set of pressure corrections followed by the calculation of new pressure and velocities for the whole domain.

PETSc provides interfaces to various Krylov methods, such as conjugate gradient (CG), generalized minimal residual (GMRES), biconjugate gradient (BCG), etc. It also provides access to various preconditioners such as Jacobi, block Jacobi, additive Schwartz, etc. Several tests have been conducted using various combination of the Krylov methods and the associated preconditioners. It was found that the combination of the

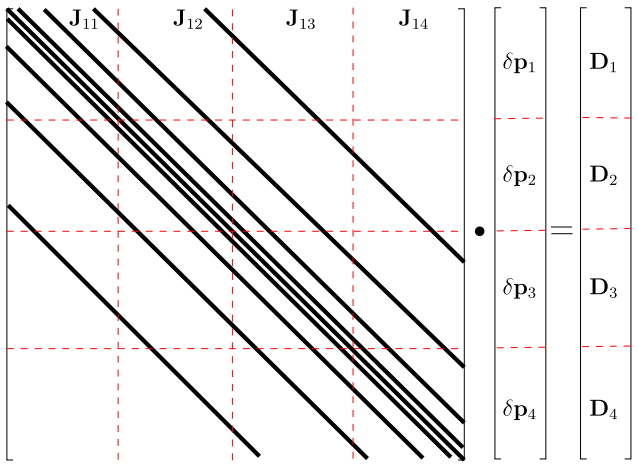


Fig. 9. Illustration of the decomposition of an hepta-diagonal matrix resembling the three-dimensional pressure correction equations into 4 block rows.

conjugate gradient Krylov method with the block Jacobi preconditioner gives the best performance for the present model. The number of subdomain blocks for the block Jacobi is set to one (default value) so that each processor gets a complete subdomain of the problem and does a single local incomplete factorization on the Jacobian corresponding to this subdomain [42]. To increase convergence rate, elements in the off diagonal block matrix which are responsible for the coupling terms between processors should be minimized. Therefore in the present study the matrix is arranged in such way that each block rows correspond with the physical domain partition which gives fewest elements in off-diagonal blocks. By rearranging the matrix structures in this way, the solution is obtained about 20% faster compared to regular matrix partition as shown in Fig. 9.

6. Parallel algorithm verification and benchmark

In this section the parallel algorithm is subject to verification and benchmarking. A homogeneous bubbly flow in a square lab-scale bubble column with medium to high gas hold-up is selected as a test case since it represents a previously "beyond reach" case to simulate with an Euler–Lagrange model. The simulation conditions are summarized in Table 2. Air is injected from 625 nozzles located at the bottom of the column into an initially quiescent liquid. The boundary conditions are imposed to the column using the flag matrix concept of Kuipers et al. [43] as illustrated in Fig. 10. The definition of each boundary condition is given in Table 3. The

Table 2
Simulation conditions for the simulation of homogeneous bubbly flow in a lab-scale bubble column

Physical domain	$0.2\ \text{m} \times 0.2\ \text{m} \times 0.6\ \text{m}$
Computational cell	$60 \times 60 \times 180$
δt_{flow}	10^{-3} s
δt_{bub}	10^{-4} s
Liquid density	10^3 kg/m^3
Liquid viscosity	10^{-3} Pa s
Gas density	1 kg/m^3
Surface tension	0.073 N/m
Gravitational acceleration	9.81 m/s^2
Initial bubble diameter	4×10^{-3} m
Size of window mapping	$3d_b$ m
Number of gas nozzle	625 (uniformly arranged with 8 mm^2 pitch distance)

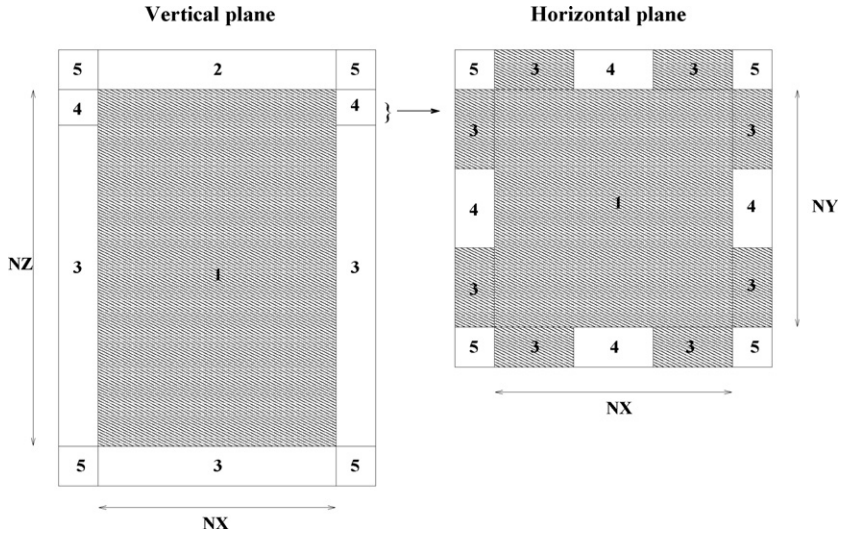


Fig. 10. Typical boundary conditions used in simulations with the discrete bubble model. The vertical plane is at $j = NY/2$ while the horizontal plane is at $k = NZ$.

Table 3
Cell flags and corresponding cell types used in defining boundary conditions

Flag	Boundary conditions
1	Interior cell, no boundary conditions specified
2	Impermeable wall, free slip boundary
3	Impermeable wall, no slip boundary
4	Prescribed pressure cell, free slip boundary
5	Corner cell, no boundary conditions specified

configuration of boundary conditions used in the simulations has been carefully investigated. The prescribed pressure cells close to the column surface wall are required as inlet as well as outlet channel to compensate for the change of liquid volume due to bubbles entering and leaving the column. The width of this pressure cell slit is one-third of the total width of the column and located in the middle. It was found that this configuration avoids instabilities developing at the top surface of the column [7].

The simulation was conducted on an SGI Altix 3700 system, consisting of total 416 CPUs (Intel Itanium 2, 1.3 GHz, 3 Mbyte cache each). Every node in the Altix is a CC-NUMA machine (i.e. cache-coherent non-uniform memory access). In the CC-NUMA model, the system runs one operating system and shows only a single memory image to the user eventhough the memory is physically distributed over the processors. Since processors can access their 'own' memory (i.e. memory on the same physical board as the processor) much faster than that of other processors, memory access is non-uniform (NUMA). In the present simulation, the domain partitioning in the continuous phase calculation was applied in the z direction only as can be seen in Fig. 11. This partition configuration has been selected since it gives the smallest inter-domain connection, which minimizes data communication between processors.

6.1. Parallel verification

For verification and benchmarking purposes a superficial gas velocity of 3 cm/s is used. The verification is performed by comparing the simulation results obtained by the serial and parallel algorithms. The liquid phase velocity in the vertical direction is compared, moreover we also compare the mapped bubble vertical velocity. For this comparison series of simulations were conducted from identical initial conditions. The comparison was made after the flow has evolved during 500 flow time steps. Fig. 12 (top) shows the vertical liquid velocity along the vertical center-line of the column while Fig. 12 (bottom) shows the mapped bubble vertical velocity on the same line. As we can see, after 500 flow time steps, there are no significant differences between the serial and parallel solutions.

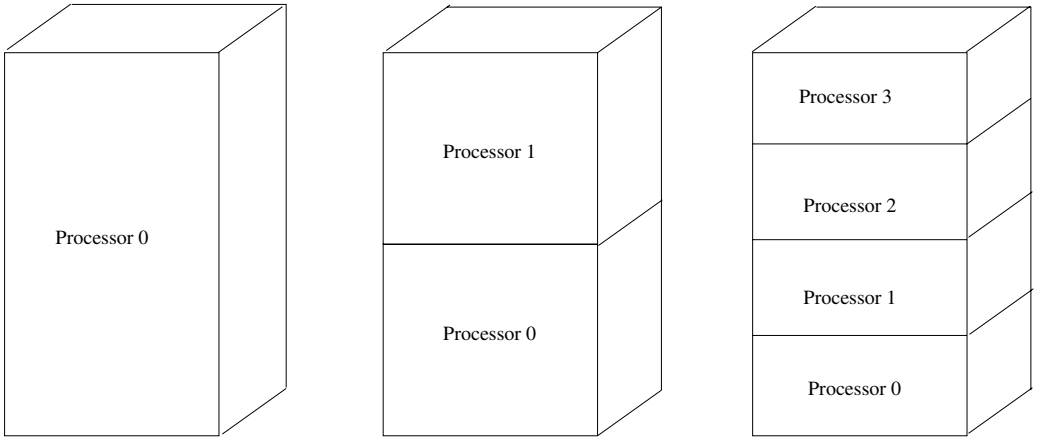


Fig. 11. Domain partitioning for parallel flow solver calculation using 1, 2 and 4 processors.

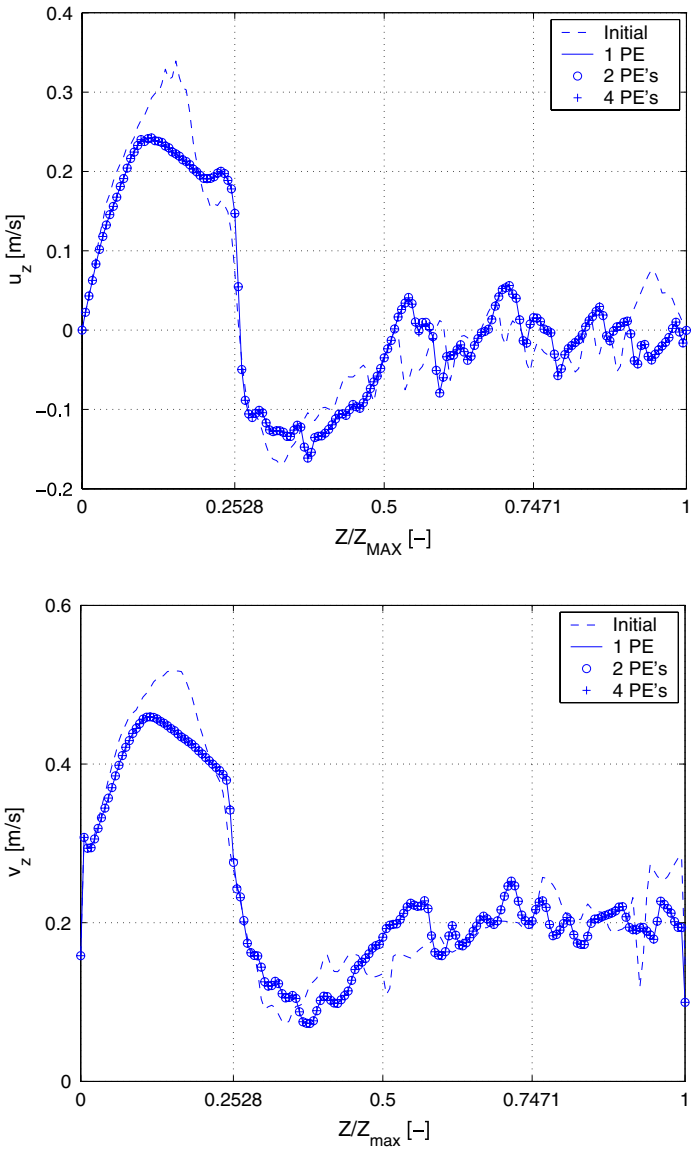


Fig. 12. Comparison of u_z (top) and v_z (bottom) at the vertical center-line of the column, the vertical lines in the figure represent the locations of the partition domains. The simulation is started from identical initial conditions for the serial and parallel cases. The comparison was made after the flow has evolved during 500 flow time steps from the initial condition.

The parallelization of the parallel bubble solver algorithm is verified by comparing the time series of the l_2 norm of the consecutive event contact points. The event contact point between collision partners a and b is determined using the following relation:

$$\mathbf{x}_c = \mathbf{x}_a + \{R_a + 0.5(|\mathbf{x}_{ab}| - (R_a + R_b))\} \cdot \frac{\mathbf{x}_{ab}}{|\mathbf{x}_{ab}|} \tag{49}$$

with $\mathbf{x}_{ab} = \mathbf{x}_b - \mathbf{x}_a$ the translation vector between collision partners a and b. Fig. 13 shows the comparison of the event contact point between collision partners recorded in the simulation using the serial and parallel algorithms. As we can see, the differences between sequential and parallel solution are not observable, which implies that similar events are obtained using both algorithms.

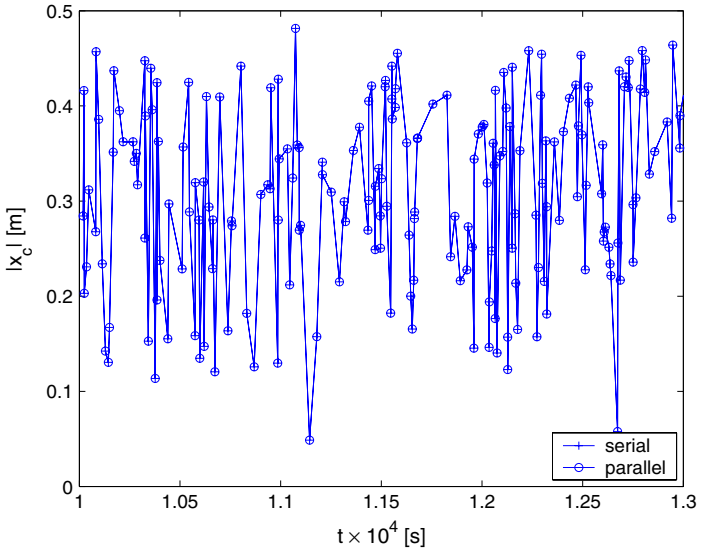


Fig. 13. Time series of the l_2 norm of a sequence of 200 event contact points as obtained from the serial and parallel algorithms.

6.2. Parallel performance

According to Ferziger and Peric [44], the analysis of the performance of parallel programs can be characterized by the *speed-up* factor and the *efficiency* defined, respectively, by:

$$S_n = \frac{T_s}{T_n} \tag{50}$$

and

$$E_n = \frac{T_s}{nT_n} \tag{51}$$

where T_s is the execution time for the best serial algorithm on a single processor and T_n is the execution time for the parallelized algorithm using n processors.

In this study the measurement of calculation times was conducted after the flow was fully developed, which can be assessed by monitoring the number of bubbles present in the column. When the column reaches a steady state condition (in the terms of bubble hold-up) the number of bubbles entering and leaving the column is more or less similar. For the present study, the steady state is reached 10 s after the bubbles have entered the column.

Fig. 14 (top) shows the speed-up obtained with 1, 2, 4, 8, 16 and 32 processors while Fig. 14 (bottom) shows the corresponding efficiency. As can be seen from these figures, the proposed parallel algorithm demonstrates good scalability. Using 32 processors a speed-up of more than 20 can be reached while the corresponding efficiency is still relatively high. The performance of the total model in terms of speed-up and efficiency is a weighted average of the underlying dispersed and the continuum parts. For this reason, the curves for the total model always lie in between the curves for the two separate parts.

7. Application to bubbly flow

In this section, the proposed method is applied to simulate the buoyancy driven flow in a square bubble column for an air–water system. The square bubble column introduced in Section 6.1 is used as a base configuration. Two cases are simulated; a case with and without coalescence model, to investigate the influence of coalescence on the bubble size distribution and the hydrodynamic characteristics.

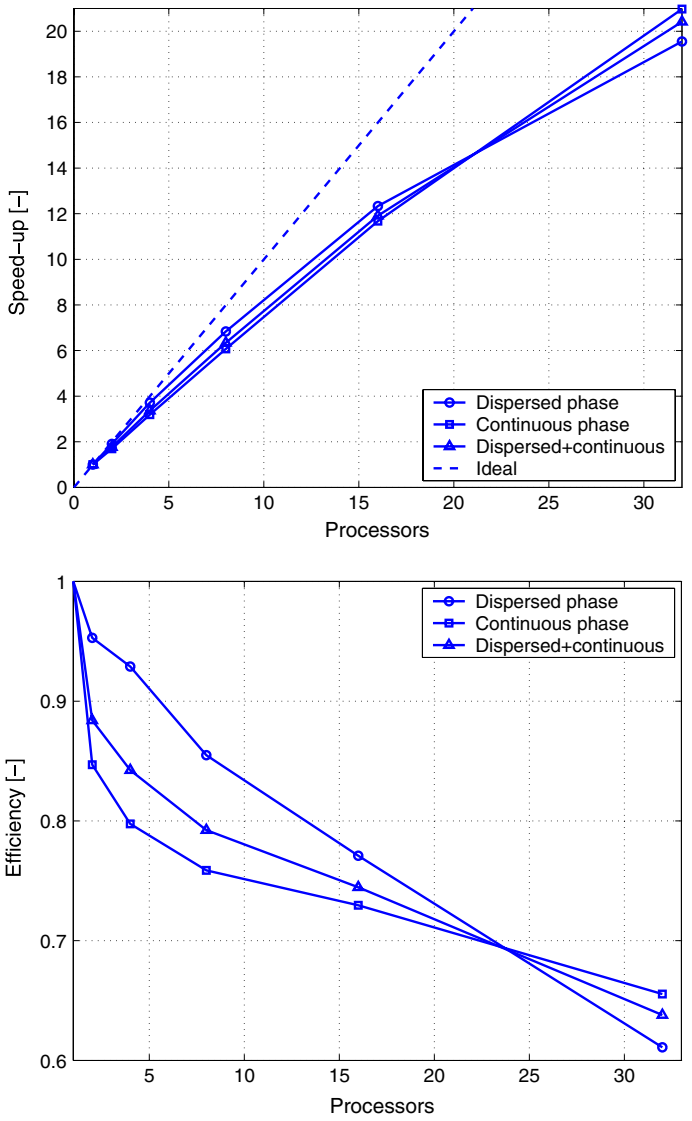


Fig. 14. Speed-up (top) and efficiency (bottom) of the dispersed and continuous phase solver for different number of processors to evaluate the performance of the parallel algorithm on a $60 \times 60 \times 180$ numerical grid containing $\approx 10^5$ bubbles. The measurement was conducted after the flow in the column is fully developed.

Air is injected through 625 individual nozzles into an initially quiescent liquid as shown in Fig. 15. Shortly after the bubbles are released, they start to rise in the column and drag the liquid upwards. For the case that the nozzles are only present in the center area of the column, Darmana et al. [7] observed a mushroom-shaped bubble plume during the initial period of bubble injection. In the present simulation this shape is not observed due to the uniform aeration, which induces a uniform liquid flow. Instead of generating a mushroom-shaped bubble plume, the bubbles rise in a uniform fashion.

After about 3 s the first bubble escapes from the column. Liquid vortices are generated close the surface of the column with upward direction in the center region and downward direction close the corners of the column. These type of liquid vortices are normally responsible for creating large scale fluctuations as they will travel downwards in a region close to the wall and influence the bubbles close to the inlet region. However in the present configuration, the down-flow which is developed near the wall region is counteracted by the bubbles moving upward in that region resulting in suppression of the liquid down-flow and a local bubble velocity reduction.

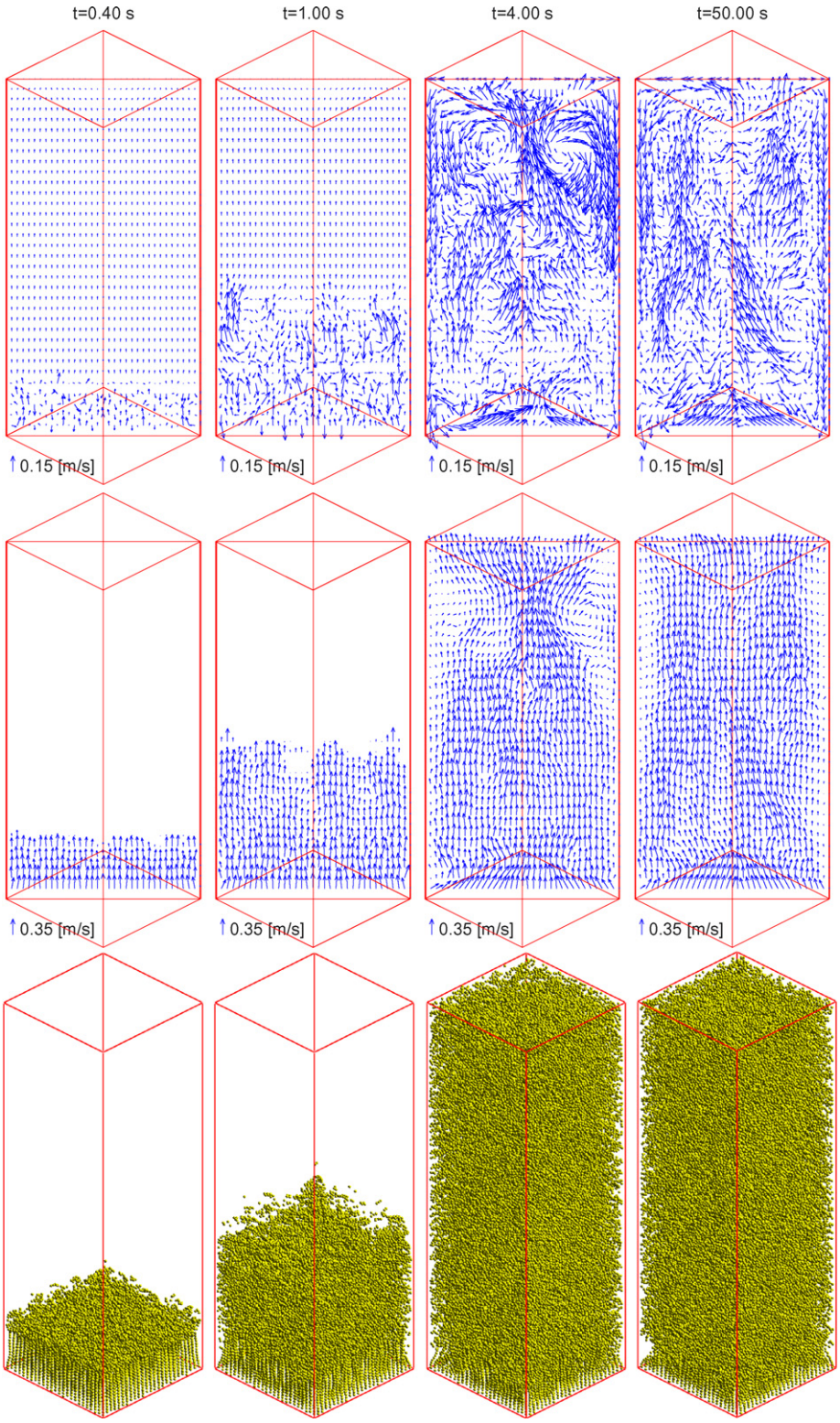


Fig. 15. Series of corresponding liquid velocity fields, bubble velocity fields and bubble positions obtained from the simulation of an non-coalescing air–water bubble column at different times after the air flow was switched on. Gas superficial velocity = 3 cm/s. The gas volume fraction is approximately 15%.

A fully developed flow condition is reached after about 10 s of operation. A typical snapshot of the flow structures after the flow has become fully developed is shown in Fig. 16(a). The figure clearly depicts that the bubble trajectories are rectilinear in the small region close to the inlet region of the column. In this region, bubble velocities are mainly directed vertically resulting in a smooth path of bubbles. However as the bubbles move further from the inlet, interaction with the liquid as well as interaction with the other bubbles becomes more pronounced which turns the bubble trajectory into a non-smooth path. In this situation the horizontal components of the bubble velocities become more significant resulting in non-smooth bubble trajectories. The pronounced liquid agitation which prevails during the first few seconds of the simulation has disappeared and is replaced by various small vortices, which are distributed randomly in the liquid phase. These vortices are not strong enough to influence the bubbles trajectory in general as can be seen in the figure showing the bubble

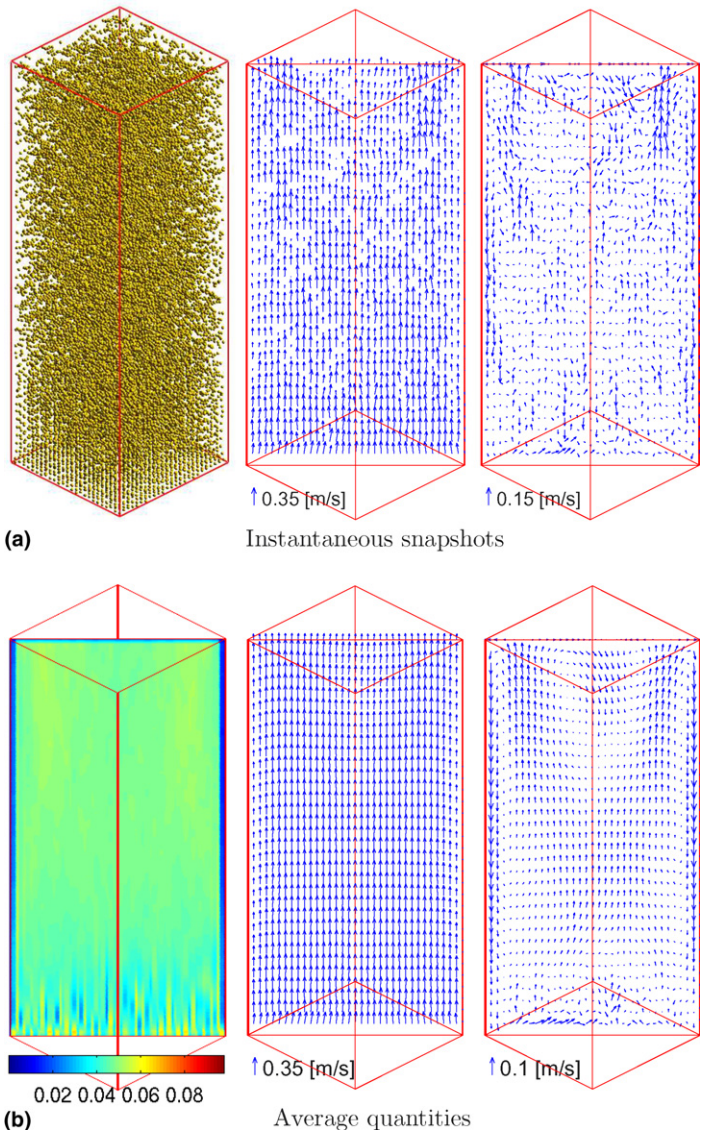


Fig. 16. Snapshots of instantaneous bubble positions, bubble velocities and corresponding liquid velocities at $t = 65$ s (a) and the corresponding averaged quantities of the gas phase volume fraction, bubble velocity and liquid velocity (b). Averaging is taking place for the last 45 s. Simulation results with a gas superficial velocity of 1 cm/s and the coalescence model turned off.

path-way. A weak down-flow is observed in the vicinity of each corner of the column which slightly reduces the bubble velocities in these areas.

The time-averages of the gas phase volume fraction, bubbles velocity and the liquid velocity of the column are shown in Fig. 16(b) The time-average of quantity ϕ is calculated as:

$$\overline{\phi} = \frac{1}{N_t} \sum_{i=1}^{N_t} \phi_i \tag{52}$$

where N_t is the number of time steps used in the averaging.

The time-averaged quantities clearly show that the column exhibits uniform behavior. The gas volume fraction is uniform almost everywhere except close to the corner regions where there are slightly less bubbles present. Variation of the gas volume fraction can also be observed close to the inlet region. In this region stripes corresponding to higher and lower volume fractions are discernible as a direct result of the rectilinear bubble motion in this region.

On average the bubble velocity is relatively uniform except close to the corner area where the bubble velocity is lower than in the rest of the column. A very low average liquid velocity is observed in the entire column. A weak large scale circulation pattern is observed in the upper half region of the column where up-flow is located in the center region while down-flow is located in the corners. The down-flow however terminates to exist at about one third of the column height.

Next, the influence of coalescence on the flow structures is investigated. A fully developed flow condition in the column which was obtained from the model without coalescence was used as the initial condition. After the coalescence model was turned on, the simulation was run for another 10 s to allow the column to reach its new fully developed condition. A typical snapshots of a fully developed flow structure when the bubble coalescence model is taken into account is given in Fig. 17(a). It can be seen that the column exhibits different flow structures compared to the case without coalescence. First of all a non-uniform bubble size distribution is obtained as a direct result of the coalescence process. The variation of the bubble size automatically induces a variation of the bubble velocities. As the speed of a bubble increases, it will catch the slower bubbles on its path, leading to a continuous growth of the bubbles as they ascend.

The lift coefficient, which initially tends to disperse the bubbles towards the wall, changes sign as the bubbles are getting bigger and as a consequence cause to move the bubbles toward the center of the column instead. This behavior produces a narrowing of the bubble swarm near the top region of the column as can be observed in Fig. 17(a). The bubble size distribution also influences the liquid velocity as the fast moving bubbles will induce a higher liquid velocity as well. The snapshots clearly show that the liquid velocity gradually turns into a irregular pattern as the mean bubble size increases. In the corner regions where there are less bubbles, strong downward liquid flow is developed. This flow appears to push the bubble dispersion toward the center column even further.

The corresponding time-averaged flow field of the coalescence case is given in Fig. 17(b). The average gas volume fraction clearly shows the narrowing behavior of the bubble dispersion. The average bubble velocity is higher compared to the case without coalescence and increases further with increasing height. Furthermore, the regions in the corners with low bubble velocity are wider compared to the non-coalescencing case. The average liquid velocity clearly shows large scale circulation patterns with upward flow in the center of the column and downward flow in the corners.

A more quantitative comparison is obtained by comparing the time-average liquid and bubbles velocities along the horizontal axis at a height of $h = 0.45$ m. Fig. 18(a) and (b) shows the lateral profiles of the liquid and bubble velocity, respectively. As can be seen from this figure, without coalescence model, the average liquid velocity is very small. In the central region of the column the averaged liquid velocity is -1 cm/s while a maximal velocity of 4 cm/s is observed in the region close to the wall. When the coalescence model is turned on, the averaged liquid velocity is dramatically changed as the averaged liquid velocity now shows velocities of about 12.5 cm/s in the central part of the column. A similar picture emerges for the average bubble velocity as the velocity at the center of the column is increased from 20 cm/s for the case without coalescence to 35 cm/s for the case with coalescence.

The influence of coalescence on the average liquid velocity fluctuations is shown in Fig. 19. The average liquid velocity fluctuations are calculated as:

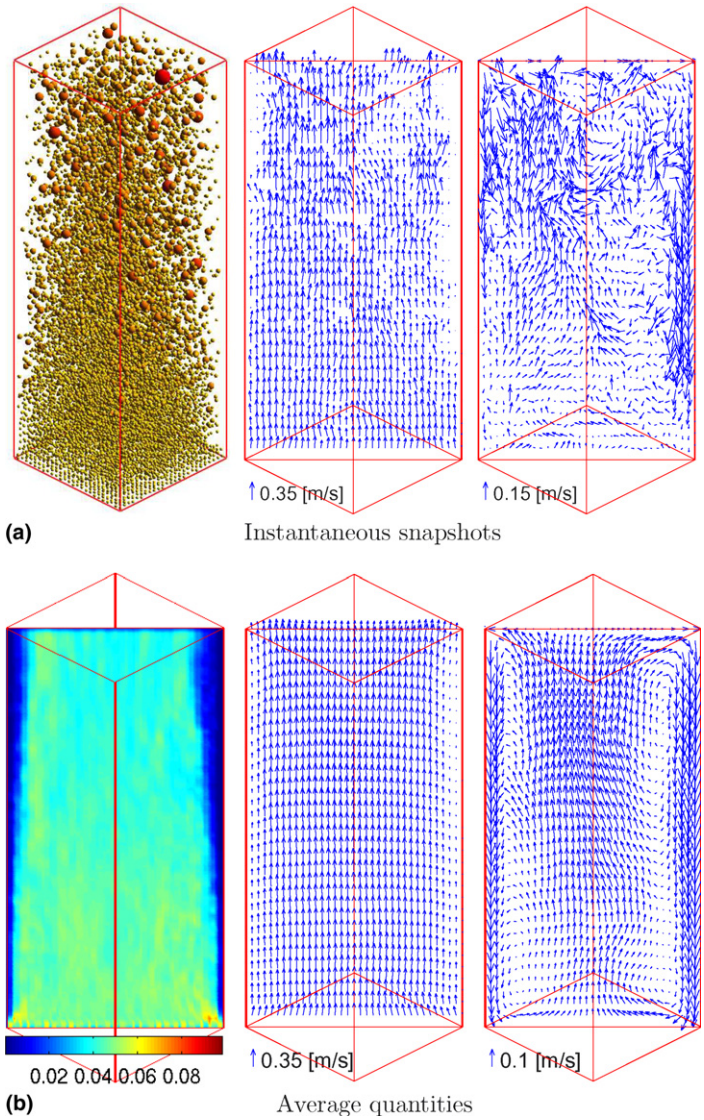
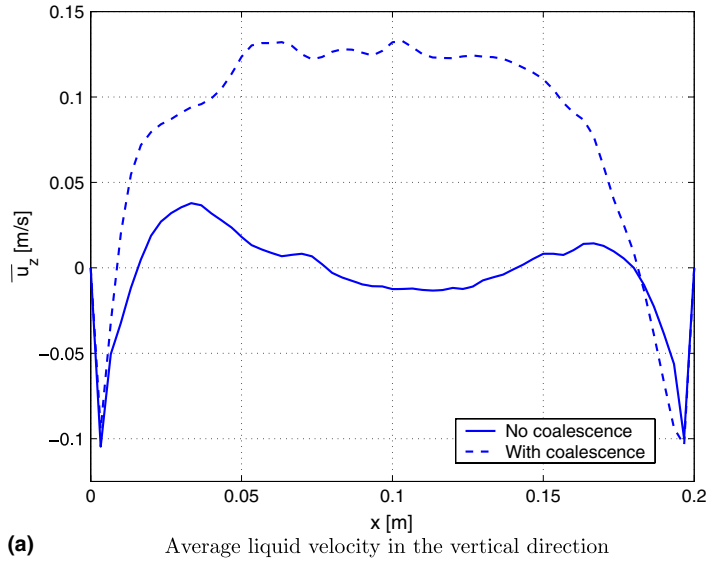


Fig. 17. Snapshots of instantaneous bubble positions, bubbles velocities and corresponding liquid velocities (a) and the averaged quantities of the gas phase volume fraction, bubble velocity and liquid velocity (b). Simulation results with a gas superficial velocity of 1 cm/s and the coalescence model turned on.

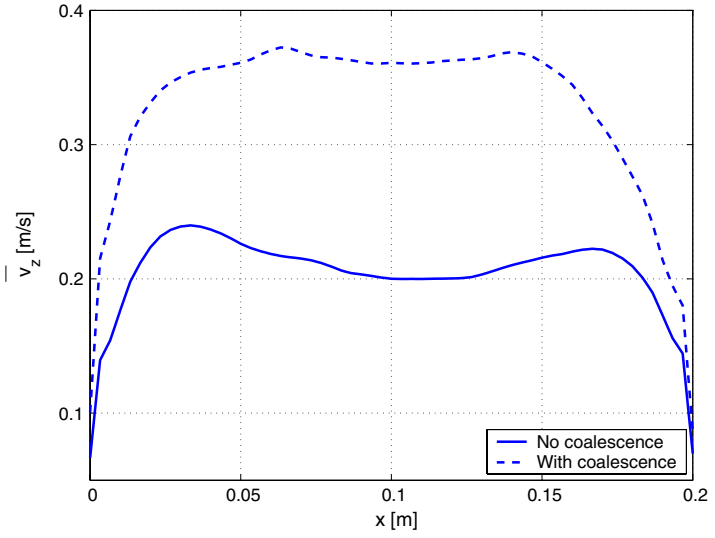
$$\mathbf{u}' = \frac{1}{N_t}\sqrt{\sum_{i=1}^{N_t}(\mathbf{u}_i - \overline{\mathbf{u}})^2} \tag{53}$$

As can be seen from Fig. 19, the coalescence model also amplifies the velocity fluctuations. The fluctuations in the vertical direction which was formerly uniform has doubled in the center region of the column and increased by a factor of three near the wall when the coalescence is taken into account. A similar result is obtained for the fluctuations in the horizontal direction where the fluctuations are increased four times over almost the entire width of the column. The amplification of the velocity fluctuations arises since the coalescence leads to the formation of regions with lower gas hold-up where liquid vortices start to develop. These vortices in turn will influence the bubble motion and the induced velocity fluctuations.

Fig. 20 (top) shows the bubble size distribution in four different vertical regions of the column. The regions are four non-overlapping, equally-sized compartments. All bubbles inside one particular compartment are



(a) Average liquid velocity in the vertical direction

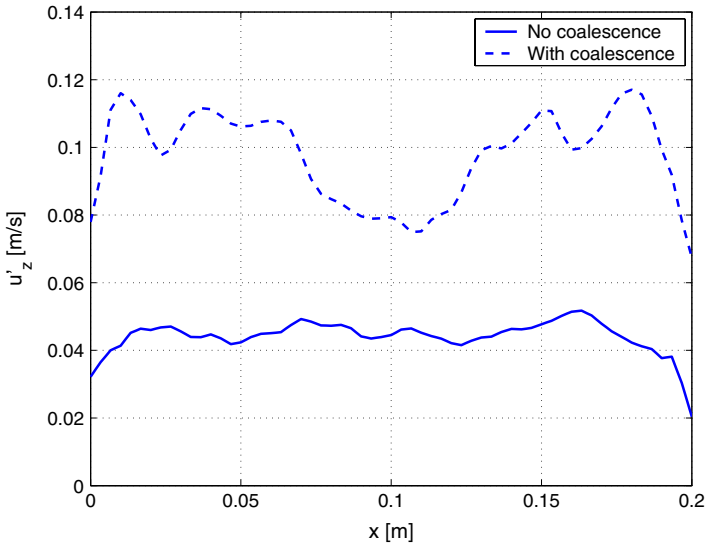


(b) Average bubble velocity in the vertical direction

Fig. 18. Comparison of the average liquid velocities (a) and average bubble velocities (b) between the cases with and without coalescence at a height of $z = 0.45$ m and a depth of $y = 0.1$ m. The gas enters the column uniformly at a superficial velocity of 1 cm/s.

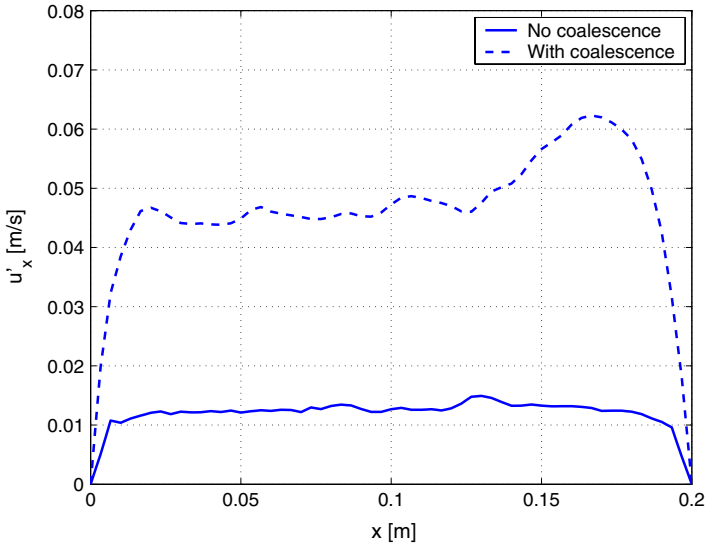
grouped based on their volume resulting from binary coalescence events. Ten different bubble classes are used, resembling bubble sizes resulting from 1 to 9 coalescence events. Furthermore, the tenth class contains bubbles that experienced more than 9 coalescence events. The number of bubbles in each class is normalized by the total number of bubbles in the compartment.

In the lower part of the column, the bubble size appears to be homogeneous with 90% of the bubbles having the initial size of 4 mm, while about 10% of the bubble population has experienced one coalescence event. A sudden change can be observed at $h = 0.225$ m as more than 50% of the bubbles in this region have already coalesced (i.e. about 30% coalesced once while the rest of the bubbles coalesced more than once). At $h = 0.375$ m only 30% of the total bubble population did not experience coalescence. This number is more or less equal to the number of bubbles that already coalesced once while 15% of the bubbles in this region coalesced twice. In this region we can also observe that about 5% of the bubbles coalesced more than 8 times. In the two upper regions of the column the bubble distribution is more or less similar, however, we can clearly



Average liquid velocity fluctuations in the vertical z-direction

(a)



Average liquid velocity fluctuations in the horizontal x-direction

(b)

Fig. 19. Comparison of the average liquid velocity fluctuations in the vertical (a) and horizontal direction (b) between cases with and without coalescence at a height of $z = 0.45$ m and a depth of $y = 0.1$ m. The gas enters the column uniformly at a superficial velocity of 1 cm/s.

see that the number of bubbles that coalesced more than 9 times has doubled in the upper region compared to the region below into more than 10%.

For each region the average of the bubble size distribution is represented by the Sauter mean diameter which reflects the mean bubble size for all bubbles in the region averaged on basis of specific area and is given as:

$$d_{32} = \sum_{i=1}^{N_b} \frac{d_i^3}{d_i^2} \tag{54}$$

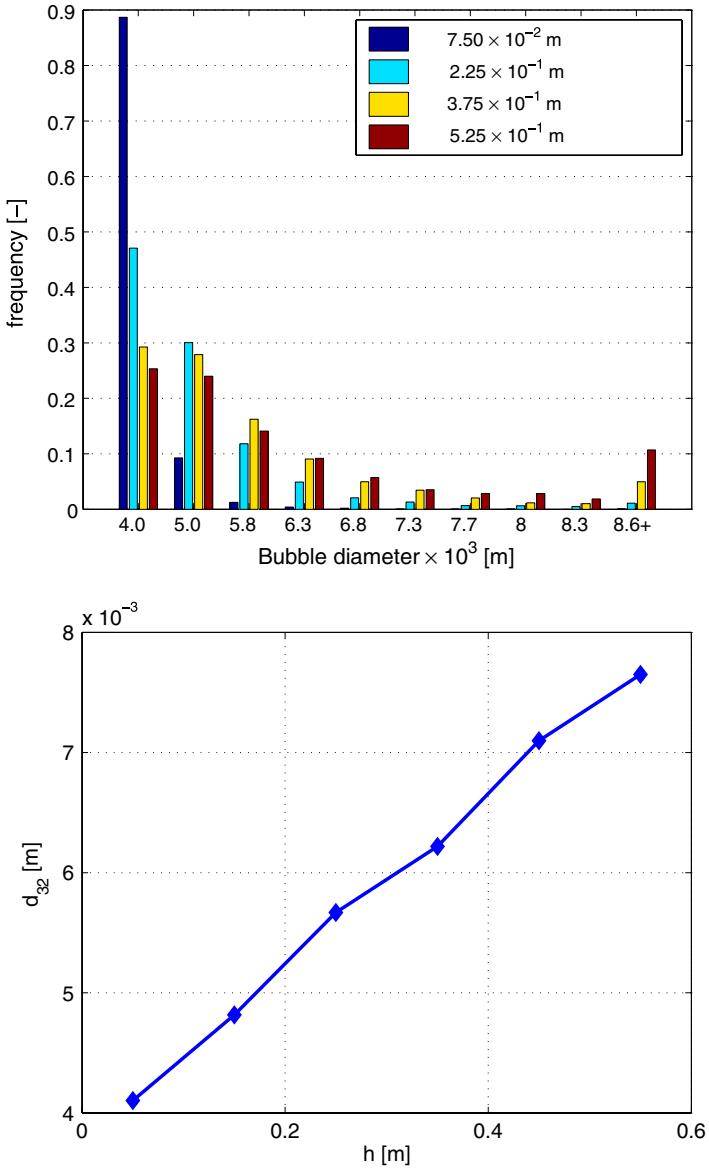


Fig. 20. Top: bubble size distribution in four regions along the vertical axis of the column ($x = 0.1$ m; $y = 0.1$ m). Bottom: Sauter mean diameter (d_{32}) along the vertical axis of the column. The gas enters the column uniformly at a superficial velocity of 1 cm/s.

Fig. 20 (bottom) shows the Sauter mean diameter at six different locations measured from the bottom of the column shown in Fig. 17(a). As can be seen for this particular case, due to coalescence the volume mean diameter is increased almost linearly as a function of distance from the inlet. It also appears from this simulation that due to coalescence the bubbles have doubled in size during their residence in the column.

With the present parallel algorithm, the limitation of the discrete bubble model on the gas hold-up that can be treated has virtually been eliminated. In this study we use the model to predict the integral gas hold-up as a function of the superficial gas velocity. Using a gas inlet consisting of multiple nozzles that are uniformly arranged at the base of the column, Harteveld et al. [45] showed that a much higher gas hold-up can be obtained compared to other types of inlets such as sintered or porous plates.

The column geometry and the nozzle arrangement explained in Section 6 is used as a base model. Without the coalescence model taken into account, cases were run with superficial velocities ranging between 1 cm/s

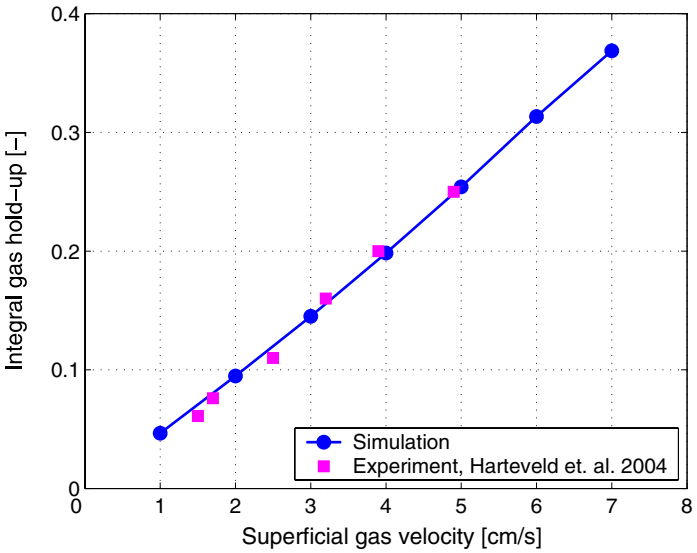


Fig. 21. Dependency of the integral gas hold-up in the bubble column predicted by the present model in comparison with the experimental measurement with similar superficial gas velocity by Harteveld et al. [45].

and 7 cm/s and the average integral gas hold-up under fully developed conditions is monitored and compared with experimental measurement data of Harteveld et al. [45]. It is noted that the column used in the work of Harteveld is cylindrical, whereas it is square in our work. This has an effect on the liquid down-flow, which mostly takes place in the circumferential wall area and in the corners, respectively. It is believed however that the flow in the core in the column is hardly affected by the geometry, which allows us to make a direct comparison. Fig. 21 shows the comparison between the simulation results and the experimental measurement data. As can be seen a gas hold-up up to 37% can be obtained with the present model. The number of bubbles simultaneously present in the column ranges from about 3.3×10^4 at a superficial velocity of 1 cm/s to 2.7×10^5 at a superficial velocity of 7 cm/s. The increase in gas hold-up with the superficial gas velocity is almost linear as can be inferred from Fig. 21. The simulation results shows perfect agreement with the experimental measurements.

8. Conclusions

In this study, a parallelization strategy for a two-phase Euler–Lagrange model for bubbly flow has been successfully developed accounting for four-way coupling. A new mapping technique based on the work of Kitagawa et al. [8] has implemented which relates data in the Eulerian and Lagrangian frame.

The implementation of the parallel algorithm was verified by comparing the computational results obtained from the serial and parallel algorithms. It was demonstrated that both algorithms give the same results. Speed-up and efficiency measurements were performed to investigate the performance of the parallel algorithm. A maximum speed-up up to 20 can be reached using 32 processors.

Subsequently the proposed model was used to investigate the influence of coalescence on the hydrodynamics of a bubble column. We found that the coalescence phenomenon changes the flow structures considerably, furthermore the average velocities and velocity fluctuations of both phases are changed considerably. In the present study break-up is not yet taken into account thus incorporation of coalescence in our calculations admittedly tends to give overprediction in bubble size distribution.

Finally the model is used to predict the integral gas hold-up in a homogeneous bubble column as a function of the gas superficial velocity. A maximum gas hold-up of 37% can be achieved with the present model. A perfect agreement of our computations with the experimental data by Harteveld et al. [45] was found for.

Acknowledgments

The authors thank DSM and the Institute of Mechanics, Processes and Control – Twente (IMPACT) for their financial support to the project. Contribution from the National Computing Facilities Foundation (NCF) for the use of supercomputer facilities, with financial support from the Netherlands Organization for Scientific Research (NWO) is acknowledged.

References

[1] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page: <http://www.mcs.anl.gov/petsc>, 2001.
[2] N.G. Deen, T. Solberg, B.H. Hjertager, Large eddy simulation of the gas–liquid flow in a square cross-sectioned bubble column, Chemical Engineering Science 56 (2001) 6341–6349.
[3] A. Sokolichin, G. Eigenberger, Gas–liquid flow in bubble columns and loop reactors: Part i. Detailed modelling and numerical simulation, Chemical Engineering Science 49 (1994) 5735–5746.
[4] Y. Pan, M.P. Dudukovic, Numerical investigation of gas-driven flow in 2-d bubble columns, AIChE Journal 46 (2000) 434–449.
[5] A. Tomiyama, H.H.I. Zun, Y. Makino, T. Sakaguchi, A three-dimensional particle tracking method for bubbly flow simulation, Nuclear Engineering and Design 175 (1997) 77–86.
[6] E. Delnoij, F.A. Lammers, J.A.M. Kuipers, W.P.M. van Swaaij, Dynamic simulation of dispersed gas–liquid two-phase flow using a discrete bubble model, Chemical Engineering Science 52 (1997) 1429–1458.
[7] D. Darmana, N.G. Deen, J.A.M. Kuipers, Detailed modeling of hydrodynamics, mass transfer and chemical reactions in a bubble column using a discrete bubble model, Chemical Engineering Science 60 (2005) 3383–3404.
[8] A. Kitagawa, Y. Murai, F. Yamamoto, Two-way coupling of Eulerian–Lagrangian model for dispersed multiphase flows using filtering functions, International Journal of Multiphase Flow 27 (2001) 2129–2153.
[9] M. Sommerfeld, E. Bourloutski, D. Bröder, Euler/Lagrange calculations of bubbly flows with consideration of bubble coalescence, The Canadian Journal of Chemical Engineering 81 (2003) 508–518.
[10] M. Sommerfeld, Overview and fundamentals, in: V.K.I. for Fluid Mechanics, Theoretical and Experimental Modelling of Particulate Flow, Lecture Series No. 2000-6, 2000, pp. 1–62.
[11] H. Enwald, E. Peirano, A.E. Almstedt, B. Leckner, Simulation of the fluid dynamics of a bubbling fluidized bed. Experimental validation of the two fluid model and evaluation of a parallel multiblock solver, Chemical Engineering Science 54 (1999) 311–328.
[12] H. Lindborg, V. Eide, S. Unger, S.T. Henriksen, H.A. Jakobsen, Parallelization and performance optimization of a dynamic pde fixed bed reactor model for practical application, Computers and Chemical Engineering 28 (2004) 1585–1597.
[13] B.P.B. Hoomans, J.A.M. Kuipers, W.J. Briels, W.P.M. van Swaaij, Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidised bed: a hard-sphere approach, Chemical Engineering Science 51 (1) (1996) 99–118.
[14] G. Burns, R. Daoud, J. Vaigl, LAM: an open cluster environment for MPI, in: Proceedings of Supercomputing Symposium, 1994, pp. 379–386. URL: <http://www.lam-mpi.org/download/files/lam-papers.tar.gz>.
[15] J.M. Squyres, A. Lumsdaine, A component architecture for LAM/MPIProceedings, 10th European PVM/MPI Users' Group Meeting, no. 2840 in Lecture Notes in Computer Science, Springer-Verlag, Venice, Italy, 2003, pp. 379–387.
[16] J. Magnaudet, I. Eames, The motion of high-Reynolds-number bubbles in inhomogeneous flows, Annual Reviews of Fluid Mechanics 32 (2000) 659–708.
[17] H. Jakobsen, B. Sannæs, S. Grevskott, H. Svendsen, Modeling of vertical bubble-driven flows, Industrial and Engineering Chemistry Research 36 (1997) 4052–4074.
[18] H. Jakobsen, H. Lindborg, C. Dorao, Modeling of bubble column reactors: progress and limitations, Industrial and Engineering Chemistry Research 44 (2005) 5107–5151.
[19] T.R. Auton, Lift force on a spherical body in a rotational flow, Journal of Fluid Mechanics 183 (1987) 199–218.
[20] A. Tomiyama, H. Tamai, I. Zun, S. Hosokawa, Transverse migration of single bubbles in simple shear flows, Chemical Engineering Science 57 (2002) 1849–1858.
[21] R.M. Wellek, A.K. Agrawal, A.H.P. Skelland, Shape of liquid drops moving in liquid media, AIChE Journal 12 (1966) 854–862.
[22] A. Tomiyama, T. Matsuoka, T. Fukuda, T. Sakaguchi, A simple numerical method for solving an incompressible two-fluid model in a general curvilinear coordinate system, in: A. Serizawa, T. Fukano, J. Bataille (Eds.), Advances in Multiphase Flow, Society of Petroleum Engineers Inc., Elsevier, Amsterdam, 1995, pp. 241–252.
[23] J. Smagorinsky, General circulation experiment with the primitive equations, Monthly Weather Review 91 (1963) 99–165.
[24] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Oxford Science Publications, Oxford, 1987.
[25] E.I.V. van den Hengel, N.G. Deen, J.A.M. Kuipers, Application of coalescence and breakup models in a discrete bubble model for bubble columns, Ind. Eng. Chem. Res. 44 (2005) 5233–5245.
[26] A.K. Chesters, The modelling of coalescence processes in fluid–liquid dispersion: a review of current understanding, Trans. IChemE 69 (1991) 259–270.
[27] C.H. Lee, L.E. Erickson, L.A. Glasgow, Bubble breakup and coalescence in turbulent gas–liquid dispersion, Chem. Eng. Commun 59 (1987) 65–84.
[28] M.J. Prince, H.W. Blanch, Bubble coalescence and break-up in air-sparged bubble columns, AIChE Journal 36 (1990) 1485–1499.

[29] R. Kirkpatrick, M. Lockett, Influence of approach velocity on bubble coalescence, Chemical Engineering Science 29 (1974) 2362–2373.
[30] A.K. Chesters, G. Hofman, Bubble coalescence in pure liquids, Applied Scientific Research 38 (1982) 353–361.
[31] N.G. Deen, M. van Sint Annaland, J.A.M. Kuipers, Multi-scale modeling of dispersed gas–liquid two-phase flow, Chemical Engineering Science 59 (2004) 1853–1861.
[32] C.S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (1977) 220–252.
[33] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transfer 15 (1972) 1787–1806.
[34] J. Centrella, J.R. Wilson, Planar numerical cosmology. II. The difference equations and numerical tests, Astronomy & Astrophysics Journal Supplement Series 54 (1984) 229–249.
[35] E. Delnoij, J.A.M. Kuipers, W.P.M. van Swaaij, A three-dimensional CFD model for gas–liquid bubble columns, Chemical Engineering Science 54 (1999) 2217–2226.
[36] B.P.B. Hoomans, Granular dynamic of gas–solid two-phase flows, Ph.D. Thesis, University of Twente, 1999.
[37] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual, Tech. Rep. ANL-95/11 – Revision 2.1.5, Argonne National Laboratory, 2004.
[38] S. Balay, V. Eijkhout, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, Basel, 1997, pp. 163–202.
[39] G.B. Deng, J. Piquet, X. Vasseur, M. Visonneau, A new fully coupled method for computing turbulent flows, Computers and Fluids 30 (2001) 445–472.
[40] B.G.M.V. Wachem, J.C. Schouten, Experimental validation of 3-d Lagrangian vof model: bubble shape and rise velocity, AIChE Journal 48 (12) (2002) 2744–2753.
[41] W.D. Gropp, D.K. Kaushik, D.E. Keyes, B.F. Smith, High performance parallel implicit CFD, Journal of Parallel Computing 27 (2001) 337–362.
[42] P. Wang, S. Balay, K. Sepehmoori, J. Abate, B. Smith, G.A. Pope, A fully implicit parallel EOS compositional simulator for large scale reservoir simulation, in: SPE 15th Reservoir Simulation Symposium, Society of Petroleum Engineers Inc., 1999, pp. 63–71.
[43] J.A.M. Kuipers, K.J. van Duin, F.P.H. van Beckum, W.P.M. van Swaaij, Computer simulation of the hydrodynamics of a two dimensional gas-fluidized bed, Computer Chemical Engineering 17 (1993) 839–858.
[44] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer, New York, 1999.
[45] W.K. Harteveld, J.E. Julia, R.F. Mudde, H.E.A. van den Akker, Large scale vortical structures in bubble columns for gas fraction in the range of 5–25%, in: Proceedings of CHISA 2004 Conference, Prague, Czech Republic, 2004, pp. 2963–2982.