Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 2674–2706
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids

M. Herrmann

Department of Mechanical and Aerospace Engineering, Arizona State University, Tempe, AZ 85287, USA

Received 13 February 2007; received in revised form 1 November 2007; accepted 6 November 2007
Available online 17 November 2007

Abstract

A balanced force refined level set grid method for two-phase flows on structured and unstructured flow solver grids is presented. To accurately track the phase interface location, an auxiliary, high-resolution equidistant Cartesian grid is introduced. In conjunction with a dual-layer narrow band approach, this refined level set grid method allows for parallel, efficient grid convergence and error estimation studies of the interface tracking method. The Navier–Stokes equations are solved on an unstructured flow solver grid with a novel balanced force algorithm for level set methods based on the recently proposed method by Francois et al. [M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comput. Phys. 213 (2006) 141–173] for volume of fluid methods on structured grids. To minimize spurious currents, a second order converging curvature evaluation technique for level set methods is presented. The results of several different test cases demonstrate the effectiveness of the proposed method, showing good mass conservation properties and second order converging spurious current magnitudes.

PACS: 47.55.Ca; 47.20.Dr; 68.03.g; 47.35.Pq; 47.20.Ma

Keywords: Level set method; Two-phase flow; Grid refinement; Curvature evaluation; Surface tension; Spurious currents; Rayleigh–Taylor; Surface wave

1. Introduction

In liquid/gas flows, surface tension forces often play an important role. For example, during the atomization of liquid jets by coaxial fast-moving gas streams, the details of the formation of small-scale drops from aerodynamically stretched out ligaments is governed by capillary forces [32]. From a numerical point of view, surface tension poses a unique challenge since it is a singular force, active only at the location of the phase interface. In addition, the situation is further complicated by the fact that material properties, like density and viscosity, exhibit a discontinuity at the same location.

E-mail address: marcus.herrmann@asu.edu

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.11.002

One of the prerequisites for correctly treating surface tension forces is therefore the ability to locate the position of the phase interface accurately. To this end, several phase interface tracking schemes exist for fixed grid flow solvers, among them the marker method [56], the volume-of-fluid (VoF) method [19] and the level set method [52]. Each of these tracking methods has it is advantages and disadvantages, such that no clear gold-standard has emerged that is applicable to the wide range of possible two-phase flow phenomena. In this work, we will track the phase interface by a level set method. Level set methods are efficient, handle topology changes automatically, can directly solve for interfaces moving normal to themselves due to, for example, phase change and are easy to implement in parallel. Their main drawback is that liquid volume conservation is not guaranteed. Thus, hybrid methods have been proposed that make use of better volume conservation properties of an auxiliary interface tracking method to correct the level set representation of the interface. Among these are the particle level set method [13] using marker particles, or the CLSVOF method [51] and MCLS method [57] using the volume of fluid method. However, errors in the level set representation of the interface position are detected and corrected locally, thus potentially resulting in significant fluctuations in the higher derivatives of the level set scalar, i.e. the curvature or curvature derivative [21]. While de-localization techniques of the error correction can partly avoid this problem [11], both correction and de-localiaztion add an additional level of complexity to the scheme that is not always desired. Since the observed volume conservation error in level set methods is proportional to the employed grid resolution, an alternative approach is to employ fine enough grids to control the error. AMR techniques can be used to adaptively refine the grid in the vicinity of the interface, see for example [4,27,49,63], however these methods are usually complex in parallel applications and difficult to domain-decompose efficiently, unless block or patch refinement strategies are used [34,49].

In this paper, we propose to follow an alternative approach, termed refined level set grid (RLSG) method. In many technical applications of two-phase flows, like for example the atomization of liquid jets and sheets, the same high grid resolution is required virtually everywhere along the phase interface. Thus, a more practical approach is to uniformly refine the grid surrounding the phase interface. To avoid complex data structures like oct-trees, we propose to solve the level set equations on a separate, high resolution, equidistant Cartesian grid. The flow solver grid on which the two-phase Navier–Stokes equations are solved is independent of the level set grid and can be either structured or unstructured. To a certain extent the RLSG method is similar to the recently proposed narrow-band locally refined level set (NBLR-LS) approach by Gomez et al. [18]. However, the latter uses two-different grid levels of the same base grid and thus assumes a tight geometric coupling between the two grids. It is furthermore limited to Cartesian grids, whereas the RLSG method can deal with arbitrary unstructured finite volume flow solver meshes.

Unstructured grid methods for computing surface tension driven flows have been proposed in the past, mostly based upon finite element approaches [30,31,59,63]. These methods solve both the level set equation and the Navier–Stokes or Stokes equations on the same grid. However, since the RLSG method allows for the independent refinement of the interface tracking grid, grid convergence studies with respect to the interface tracking error can easily be performed. Furthermore, the approach allows for a separation of the error associated with the level set interface tracking scheme from the error associated with the solution of the Navier–Stokes equations since both grids can be independently refined enabling the calculation of separate error estimates.

Different strategies exist to discretize the surface tension force once the location of the phase interface is known. The most commonly used method is due to Brackbill et al. [7] called Continuum Surface Force (CSF). Here, the ideally singular surface tension force is spread into a narrow band surrounding the phase interface by the use of regularized delta functions. These can take the form of a discrete derivative of a Heaviside scalar (the volume fraction in VoF methods, or a Heaviside transform of the level set scalar [49]), or smoothed delta functions, like the popular cosine approximation due to Peskin [38] in level set methods . Especially in level set methods, the use of smoothed delta functions can be problematic, since convergence under grid refinement is only guaranteed for certain, not commonly employed delta function approximations [12].

The CSF method is prone to generating unphysical flows, so-called spurious currents, near the location of the phase interface when surface tension forces are present. In the canonical test cases of an equilibrium column and an equilibrium sphere, these velocity errors can grow unbounded very fast, if they are not artificially damped by introducing viscosity. The amplitude of the spurious currents when damped by viscosity is of the order of $u \sim 0.01\sigma/\mu$ for classical VoF and level set methods and $u \sim 10^{-5}\sigma/\mu$ for marker methods [43], where

σ is the surface tension coefficient and μ is the viscosity. At small μ, depending on the quality of the employed numerical schemes, numerical viscosity can be dominant, resulting in an artificial decrease of the spurious currents as compared to σ/μ [62]. Still, grid converged numerical simulations are limited by a critical Laplace number, $La = \sigma\rho R/\mu^2$, where ρ is the density and R is a characteristic phase interface radius of curvature, since for large La, i.e., large σ, spurious currents start to dominate the physical flow [43].

The reason for the occurence of spurious currents is twofold. The first reason is a potential discrete imbalance on collocated grids between the surface tension force and the associated pressure jump across the phase interface [16]. To address this source of error, Young et al. [60] proposed a modification to the procedure of Kim et al. [24] to regain discrete consistency. However, they were using the CSF method with smoothed out delta functions in a level set context and, hence, the exact discrete balance was not achieved. Francois et al. [16] proposed a so-called "balanced force algorithm" for VoF schemes on structured Cartesian meshes that discretely balances the surface tension force and the associated pressure jump across the interface. In that paper, the discrete evaluation of the delta function as the derivative of the volume fraction scalar naturally results in the discrete balance when following a similar approach to the one proposed in Young et al. [60]. The approach by Francois et al. [16] eliminates spurious currents up to machine precision zero, if the interface curvature is prescribed exactly. Similar results can be obtained using the ghost fluid method (GFM) proposed in Fedkiw et al. [15] or the sharp interface method by Sussman et al. [53]. Here, jump conditions and the surface tension force are applied as singular source terms directly at the location of the phase interface, thereby directly avoiding a potential discrete imbalance of these terms.

The second source of error is due to errors associated with evaluating phase interface curvature. This source of error is typically independent of the way the surface tension force and pressure gradient/jump are treated and occurs thus in CSF, balanced force, sharp interface and GFM applications alike. Different strategies exist to increase the accuracy of curvature evaluation. For VoF methods, the height-function approach [48,50] allows second-order or higher converging curvature calculation. However, the required stencil sizes are large and thus problematic for interfaces close to each other, unless a recently proposed corrective procedure is employed [50]. For level set methods, curvature at the node location can be calculated with high-order accuracy, however, this approximates the phase interface curvature only to first order, due to the fact that nodal location and phase interface position typically do not coincide. To enhance accuracy, additional interpolation techniques are required [28].

In this paper, we will extend the balanced force algorithm of Francois et al. [16] and Young et al. [60] to unstructured flow solver grids using the RLSG level set method to track the phase interface. To achieve second-order converging curvature evaluation, an interface projected curvature evaluation method is proposed. The performance of the balanced force RLSG method is demonstrated analyzing inviscid and viscous equilibrium columns and spheres, zero-gravity oscillating columns and spheres, and damped surface waves on structured and unstructured flow solver grids. Finally, to demonstrate the capability of the new method in complex flows, a Rayleigh–Taylor instability is presented.

2. Governing equations

The equations governing the motion of an unsteady, incompressible, immiscible, two-fluid system are the Navier–Stokes equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla\cdot(\mu(\nabla\boldsymbol{u} + \nabla^{\mathrm{T}}\boldsymbol{u})) + \boldsymbol{g} + \frac{1}{\rho}\boldsymbol{T}_{\sigma}, \tag{1}$$

where $\boldsymbol{u}$ is the velocity, ρ the density, p the pressure, μ the dynamic viscosity, $\boldsymbol{g}$ the gravitational acceleration and $\boldsymbol{T}_{\sigma}$ the surface tension force which is non-zero only at the location of the phase interface $\boldsymbol{x}_f$,

$$\boldsymbol{T}_{\sigma}(\boldsymbol{x}) = \sigma\kappa\delta(\boldsymbol{x} - \boldsymbol{x}_f)\boldsymbol{n}, \tag{2}$$

with σ the assumed constant surface tension coefficient, κ the local mean surface curvature, $\boldsymbol{n}$ the local surface normal and δ the delta-function. Furthermore, the continuity equation results in a divergence-free constraint on the velocity field,

$$\nabla\cdot\boldsymbol{u} = 0. \tag{3}$$

The phase interface location $\boldsymbol{x}_f$ between the two fluids is described by a level set scalar G, with

$$G(\boldsymbol{x}_f, t) = 0 \tag{4}$$

at the interface, $G(\boldsymbol{x}, t) > 0$ in fluid 1 and $G(\boldsymbol{x}, t) < 0$ in fluid 2. Differentiating Eq. (4) with respect to time yields the level set equation,

$$\frac{\partial G}{\partial t} + \boldsymbol{u} \cdot \nabla G = 0. \tag{5}$$

For numerical accuracy it is advantageous, although not necessary, to define the level set scalar away from the interface to be a signed distance function,

$$|\nabla G| = 1. \tag{6}$$

Assuming ρ and μ constant within each fluid, density and viscosity at any point $\boldsymbol{x}$ can be calculated from

$$\rho(\boldsymbol{x}) = H(G)\rho_1 + (1 - H(G))\rho_2 \tag{7}$$

$$\mu(\boldsymbol{x}) = H(G)\mu_1 + (1 - H(G))\mu_2, \tag{8}$$

where indices 1 and 2 denote values in fluid 1, respectively 2 and H is the Heaviside function. Finally, the interface normal vector $\boldsymbol{n}$ and the interface curvature κ can be expressed in terms of the level set scalar as

$$\boldsymbol{n} = \frac{\nabla G}{|\nabla G|}, \quad \kappa = \nabla \cdot \boldsymbol{n}. \tag{9}$$

In summary, Eqs. (1), (3) and (5) have to be solved jointly to describe the incompressible, immiscible two-fluid system.

3. Numerical methods

In this section, we first describe the RLSG method used to solve the level set equation and discuss how the RLSG level set solution is coupled to structured and unstructured flow solver grids. Next, the level set-based balanced force algorithm for unstructured flow solver grids is presented and the performance of the resulting method is illustrated using the canonical test cases of equilibrium columns and spheres prescribing curvature exactly. Then, the method to calculate second-order converging interfacial curvatures is outlined. Finally, results are presented for curvature evaluation of columns and spheres on structured and unstructured flow solver grids.

3.1. Refined level set grid method

The key idea of the RLSG method is to solve all level set related equations on an additional, separate, equidistant, Cartesian grid, termed G-grid in the following. The grid for the flow solver, on the other hand, can be either structured or fully unstructured containing arbitrary elements. Since both grids, the G-grid and the flow solver grid are separate, the G-grid can be independently refined to ensure a grid converged interface representation respecting the externally defined liquid volume conserving flow field. However, the potential drawback of the RLSG method is that the thus required G-grid resolution could be prohibitively expensive, both in computational time and in memory. A number of different compression techniques have been proposed in the past to address this issue, see for example [8,22,33]. While these schemes are amenable to thread-level parallelism, domain decomposition parallelization is not straightforward. To achieve an efficient domain decomposition parallel implementation with straightforward dynamic load balancing and direct access to random nodes, the following two-level narrow band methodology is introduced.

On the first level, the flow solver grid is overlaid by an equidistant Cartesian super-grid encompassing that part of the computational domain where the interface might exist; see Fig. 1. Only those super-grid cells, or blocks, that contain part of the interface or are within a predefined distance from the interface are activated and stored in linked lists distributed among the processors partaking in the simulation. Note that this feature

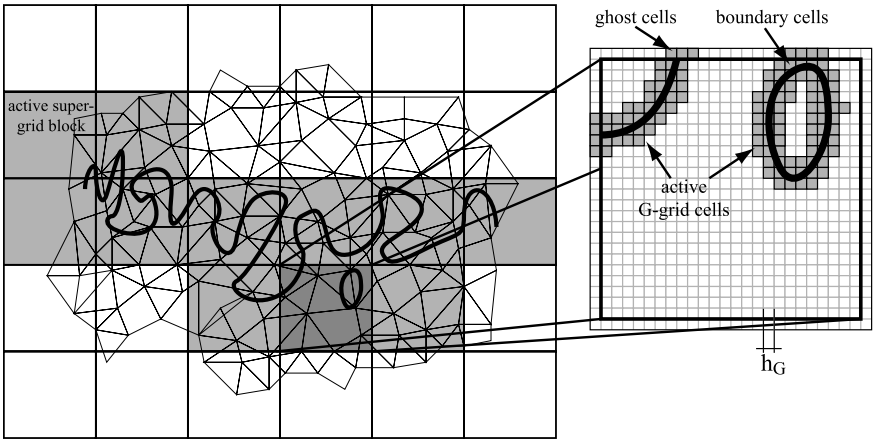


Fig. 1. RLSG grid definitions.

resembles the sparse block grid method proposed by Bridson [8]. To ensure fast and direct access to each block, each processor stores a copy of a super-grid integer i, j, k lookup table that contains either the super-grid block number if the block is local, or the negative value of the processor id where the block is stored.

On the second level, each active block contains an equidistant Cartesian grid of grid cell size h_G; see Fig. 1. Again, only those grid cells that contain part of the interface or are a predefined distance away from the interface are activated and stored. Note that the active cells thus form a band around the tracked interface.

The above approach effectively reduces the number of cells that are stored and on which the level set equations have to be solved from $O(N^3)$ to $O(N^2)$, where N is the number of cells in each spatial direction. This allows for high resolution of the interface, because at any given moment, only that small fraction of cells are active and stored that is in a small band around the interface. All active cells are stored in linked lists. However, each block stores for each grid cell a direct pointer into the linked list to allow for direct, fast access for arbitrary i, j, k coordinates.

To demonstrate the efficiency of the approach, consider a super-grid of size 128^3 and a local block size of 32^3. This yields a theoretical maximum resolution of slightly less than 70 billion cells. Assuming the number of active blocks to be $2 \cdot 128^2$, the lookup tables for the super-grid and local blocks take up only 40 MB of storage space using 128 processors (8 MB for the super-grid on each processor plus 32 MB for the processor's share of the local look-up tables). A global lookup table not using the dual level approach, on the other hand, would take 2 GB per processor. Assuming furthermore that 10×32^2 cells are active inside each active super-grid block, storage requirements for a double precision level set scalar value on each of the 128 processors is roughly 20 MB, well within the memory size of modern distributed memory massively parallel machines. The G-grid can thus efficiently provide flow solver sub-grid resolution of the phase interface.

```
calculate number of RLSG sub-cycling steps: n_G = Δt/Δt_G
for n = 1, n_G do
    do level set transport (Sec. 3.1.2)
    if (trigger re-initialization = TRUE) then (Sec. 3.1.3)
        do band generation (Sec. 3.1.1)
        do re-initialization (Sec. 3.1.3)
    end if
end do
do load balancing (Sec. 3.5)
```

Fig. 2. RLSG time step.

Fig. 2 outlines the sequence of operations of a RLSG time step. The time step size Δt is determined by the flow solver, thus in an initial step, a CFL-based criterion is used to determine the number of required time step sub-cycles n_G to advance the level set solution by Δt. Then, in each sub-cycling step, the level set transport equation, Eq. (5) is solved and a re-initialization trigger condition is evaluated. Should re-initialization be triggered, the band structure is regenerated before the level set scalar is re-initialized. After completion of all required sub-cycling steps, the G-grid is load balanced to ensure good parallel performance. The following subsections describe each individual step of the RLSG time step in detail.

3.1.1. Band generation

As outlined in the previous section, all level set equations are solved only on a narrow band around the interface. Since the interface can move through the computational domain, the band has to be regenerated frequently. The idea of using narrow bands or tubes to limit the computational cost was introduced in Adalsteinsson and Sethian [1] and Peng et al. [37]. While the former algorithm uses tubes consisting of two-dimensional square patches, the latter method builds the narrow band in terms of the G-scalar value, requiring G to be a signed distance function. Since using square patches is not efficient in three dimensions and G cannot be guaranteed to be close to a signed distance function at all times, we will employ a band growth algorithm that grows the band outwards from the interface location.

The algorithm to generate a band of width $2n_i$ is described in Figs. 3 and 4. Its main idea is to grow the narrow band one layer at a time, marking all cells that are already part of the band as the *body* ($\mathcal{B}$), the current outer layer of the body as the *skin* ($\mathcal{S}$) and the new band layer as *cloth* ($\mathcal{C}$), see Fig. 4.

In an initial step, all cells directly adjacent to the interface, i.e. cells with any $G_{i,j,k}G_{i\pm1,j,k} \leqslant 0$ or $G_{i,j,k}G_{i,j\pm1,k} \leqslant 0$ or $G_{i,j,k}G_{i,j,k\pm1} \leqslant 0$, are tagged as $\mathcal{S}$. To grow the band by one layer, a cloth layer is grown by marking all unmarked cells directly adjacent to $\mathcal{S}$-cells as $\mathcal{C}$. Here, directly adjacent again refers to any of the six neighbors of a cell in the three coordinate directions. If a new cloth cell did not previously exist, a new cell is generated and added. Special care must be taken in the parallel version of this algorithm, since the cloth layer can grow across domain boundaries. If the cloth layer grows into a local ghost cell, see Fig. 1, these cloth ghost cells are copied into their partner boundary cells in the adjacent super-grid block at the end of the cloth growth step. If not previously existing, that super-grid block is allocated and added to the list of

```
mark all cells directly adjacent to the G = 0 interface as S.
for n = 1, n_i do
   for all cells marked S do
      mark all unmarked neighbors of S as C
   end for
   sync ghostnodes with neighboring blocks
   mark all S cells as B
   mark all C cells as S
end for
```

Fig. 3. Band generation algorithm.

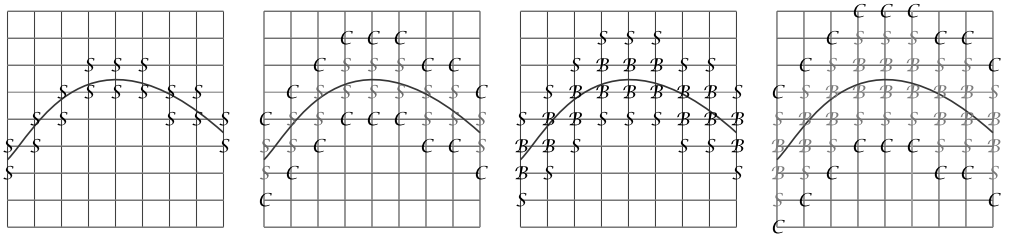


Fig. 4. Band growth example, from left to right: mark $G = 0$ adjacent cells as $\mathcal{S}$, mark all unmarked neighbors of $\mathcal{S}$ as $\mathcal{C}$, mark all $\mathcal{S}$ cells as $\mathcal{B}$ and all $\mathcal{C}$ cells as $\mathcal{S}$, mark all unmarked neighbors of $\mathcal{S}$ as $\mathcal{C}$.

active super-grid blocks. Finally, the skin layer is absorbed into the body by marking all $\mathcal{S}$-cells as $\mathcal{B}$ and the cloth layer becomes the new skin layer by marking all $\mathcal{C}$-cells as $\mathcal{S}$. Linked lists are used for the skin, cloth and body lists, in order to avoid costly memory copy operations since the number of cells inside the band are not known a-priori. In the manner described above, four different bands are generated, termed $\mathcal{T}$-, $\mathcal{N}$-, $\mathcal{W}$- and $\mathcal{X}$-band, used for transport, re-initialization, WENO-stencil and volume integration, respectively. The employed widths of the bands are similar to the ones proposed in Herrmann [21]: $n_{\mathcal{T}} = 8$, $n_{\mathcal{N}} = 3$, $n_{\mathcal{W}} = 3$ and $n_{\mathcal{X}} = \max\left(\text{int}\left(\sqrt{3}h/h_{\text{G}} - n_{\mathcal{T}} - n_{\mathcal{N}} - n_{\mathcal{W}} + 1\right), 0\right)$, where h is the characteristic flow solver grid size. The use of the individual bands will be described in the sections below.

3.1.2. Level set transport

The level set Eq. (5) is a Hamilton–Jacobi equation. We use the fifth-order WENO scheme for Hamilton–Jacobi equations of Jiang and Peng [23] in conjunction with a Roe flux with local Lax–Friedrichs entropy correction [36,47] to advance the level set scalar. Integration in time is performed by the third order TVD Runge–Kutta time discretization of Shu [46] with a CFL-number of unity.

The level set transport equation is solved only inside the $\mathcal{T}$-band, where, as suggested by Peng et al. [37], $\boldsymbol{u}$ in Eq. (5) is replaced by

$$\boldsymbol{u}_{\text{cut}} = c(G)\boldsymbol{u}, \tag{10}$$

with the cut-off function

$$c(G) = \begin{cases} 1 : \alpha \leqslant -3 \\ \frac{2}{27}\alpha^3 + \frac{1}{3}\alpha^2 : -3 < \alpha \leqslant 0 \\ 0 : \alpha > 0 \end{cases}, \tag{11}$$

and $\alpha = |G|/\Delta x - n_{\mathcal{T}}$. This ensures that no artificial oscillations are introduced at the $\mathcal{T}$-tube boundaries.

3.1.3. Re-initialization

For reasons of numerical accuracy, one would like to maintain G away from the interface $G = 0$ as smooth a field as possible. Chopp [9] proposed defining the level set scalar away from the interface to be a signed distance function, i.e., $|\nabla G| = 1$. Since solution of Eq. (5) will not maintain this property, a re-initialization procedure has to be applied to force $G \neq 0$ back to $|\nabla G| = 1$. Several different strategies exist to achieve this, here we use a PDE-based re-initialization [52],

$$\frac{\partial G}{\partial t^*} + S_0(|\nabla G| - 1) = 0, \tag{12}$$

with a modified sign function S_0 evaluated with G at $t^* = 0$ as [37]

$$S_0 = \frac{G}{\sqrt{G^2 + |\nabla G|^2 h_{\text{G}}^2}}. \tag{13}$$

We solve Eq. (12) inside the $\mathcal{T}$- and $\mathcal{N}$-bands, using a Godunov flux function together with the same fifth-order WENO scheme used for solving the level set transport equation in the $\mathcal{T}$-band and a simple first-order upwind scheme in the $\mathcal{N}$-band. The switch to the significantly more diffusive first-order scheme in the $\mathcal{N}$-band is to avoid instabilities sometimes observed at the outside edge of the $\mathcal{N}$-band when using the fifth-order WENO scheme there. Note that in order to evaluate gradients properly at the outside edge of the $\mathcal{N}$-band, cells must exist and be active in a band surrounding the $\mathcal{N}$-band. These layers of cells make up the $\mathcal{W}$-band and are used solely to be able to evaluate all gradient stencils inside the $\mathcal{N}$-band.

Unfortunately, it is well known that repeated application of the PDE-based re-initialization will inadvertently move the $G = 0$ isosurface and hence will not conserve fluid volume and thus fluid mass. While it is possible to construct more accurate re-initialization schemes, see for example [10], these tend to be computationally more expensive and hard to implement efficiently in parallel [20]. It is therefore desirable to limit the application of the PDE re-initialization procedure to situations where the divergence from G being a signed distance function would adversely impact numerical accuracy by using an appropriate trigger criterion.

Here we will use a slight modification to the criterion proposed by Gomez et al. [18]. The PDE-based re-initialization procedure is applied only if

$$\max(|\nabla G|) > \alpha_{\max} \quad \text{or} \quad \min(|\nabla G|) < \alpha_{\min}, \tag{14}$$

evaluated inside the $\mathcal{T}$-band. Also, Eq. (14) is used as a convergence criterion for the pseudo-time iteration of the re-initialization, while still limiting the maximum number of iteration steps to $n_{\max} = (n_{\mathcal{T}} + n_{\mathcal{N}})/C$, where $C = 0.5$ is the CFL-number used for the pseudo-time integration of Eq. (12). In the results presented in this paper we use $\alpha_{\max} = 2$ and $\alpha_{\min} = 10^{-4}$ unless otherwise stated. This results in typically 1–3 iteration steps until convergence is reached, should re-initialization be triggered.

3.2. RLSG-flow solver coupling

The Navier–Stokes equations and the level set equation are solved in two separate codes employing different domain decompositions. They thus require a parallel coupling strategy to exchange information. Here, we use the CHIMPS code coupling infrastructure [3] to facilitate the parallel data exchange between the flow solver and the RLSG-solver. Fig. 5 summarizes the coupled time step advancement. We stagger the solution of the level set equation and the Navier–Stokes equation in time, with the level set defined at the half time levels and the velocity vector at the full time level.

Per time step, two data exchange operations have to be performed. In the Navier–Stokes equation, the position of the phase interface influences two different terms. The first term is due to Eqs. (7) and (8), since $H(G)$ is a function of the position of the phase interface. For finite volume formulations, Eqs. (7) and (8) result in

$$\rho_{cv} = \psi_{cv}\rho_1 + (1 - \psi_{cv})\rho_2 \tag{15}$$

$$\mu_{cv} = \psi_{cv}\mu_1 + (1 - \psi_{cv})\mu_2, \tag{16}$$

with the volume fraction ψ_{cv} of control volume cv defined as

$$\psi_{cv} = 1/V_{cv} \int_{V_{cv}} H(G)dV, \tag{17}$$

and V_{cv} the volume of the control volume cv. In the RLSG method, the above integral is evaluated on the G-grid as

$$1/V_{cv} \int_{V_{cv}} H(G)dV = \frac{\sum_{i_G} V_{cv,i_G}\psi_{i_G}}{\sum_{i_G} V_{cv,i_G}}, \tag{18}$$

where V_{cv,i_G} is the joined intersection volume of the G-grid cell i_G and the flow solver control volume cv (see Fig. 6). The G-grid volume fraction ψ_{i_G} is calculated using an analytical formula developed by van der Pijl et al. [57],

initialize $\boldsymbol{u}^0_{cv}$ in flow solver and $G^{1/2}_{i_G}$ in RLSG solver
for $n = 0, n_{max}$ **do**
  do calculate $\psi^{n+1/2}_{i_G}$ and $\kappa^{n+1/2}_{i_G}$ in RLSG solver (Secs. 3.2 & 3.4)
  do integrate $\psi^{n+1/2}_{cv}$, $\kappa^{n+1/2}_{cv}$ from $\psi^{n+1/2}_{i_G}$, $\kappa^{n+1/2}_{i_G}$ using CHIMPS (Sec. 3.2)
  do calculate $\rho^{n+1/2}_{cv}$ and $\mu^{n+1/2}_{cv}$ from $\psi^{n+1/2}_{cv}$ in flow solver (Sec. 3.2)
  do solve $\boldsymbol{u}^n_{cv} \rightarrow \boldsymbol{u}^{n+1}_{cv}$ with balanced force method in flow solver (Sec. 3.3)
  do interpolate $\boldsymbol{u}^{n+1}_{i_G}$ from $\boldsymbol{u}^{n+1}_{cv}$ using CHIMPS (Sec. 3.2)
  do solve $G^{n+1/2}_{i_G} \rightarrow G^{n+3/2}_{i_G}$ in RLSG solver (Fig. 2)
end do

Fig. 5. Coupled time step advancement.

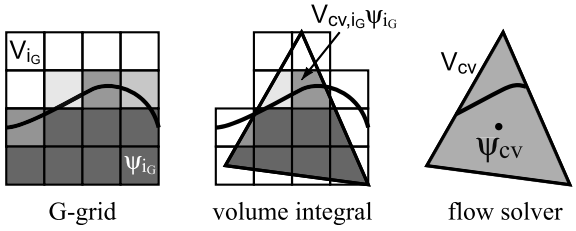


Fig. 6. Volume integration for unstructured flow solver grid cells.

$$\psi_{i_G} = \begin{cases} \psi^* : G_{i_G} \leqslant 0 \\ 1 - \psi^* : G_{i_G} > 0 \end{cases}, \tag{19}$$

with

$$\psi^* = \begin{cases} \frac{A^3 - B^3 - C^3 - D^3 + E^3}{6 D_\xi D_\zeta D_\eta} : D_\xi > \epsilon \wedge D_\eta > \epsilon \wedge D_\zeta > \epsilon \\ \frac{A^2 - C^2}{2 D_\xi D_\eta} : D_\xi > \epsilon \wedge D_\eta > \epsilon \wedge D_\zeta \leqslant \epsilon \\ \frac{A}{D_\xi} : D_\xi > \epsilon \wedge D_\eta \leqslant \epsilon \wedge D_\zeta \leqslant \epsilon \\ 0 : D_\xi \leqslant \epsilon \wedge D_\eta \leqslant \epsilon \wedge D_\zeta \leqslant \epsilon \wedge G_{i_G} \neq 0 \\ \frac{1}{2} : D_\xi \leqslant \epsilon \wedge D_\eta \leqslant \epsilon \wedge D_\zeta \leqslant \epsilon \wedge G_{i_G} = 0 \end{cases}, \tag{20}$$

where

$$\begin{aligned} A &= \max\left(\frac{1}{2}(D_\xi + D_\eta + D_\zeta) - |G_{i_G}|, 0\right) \\ B &= \max\left(\frac{1}{2}(D_\xi + D_\eta - D_\zeta) - |G_{i_G}|, 0\right) \\ C &= \max\left(\frac{1}{2}(D_\xi - D_\eta + D_\zeta) - |G_{i_G}|, 0\right) \\ D &= \max\left(\frac{1}{2}(-D_\xi + D_\eta + D_\zeta) - |G_{i_G}|, 0\right) \\ E &= \max\left(\frac{1}{2}(D_\xi - D_\eta - D_\zeta) - |G_{i_G}|, 0\right) \end{aligned} \tag{21}$$

with

$$\begin{aligned} &D_\xi = \max(D_x, D_y, D_z), \quad D_\zeta = \min(D_x, D_y, D_z) \\ &D_\eta = D_x + D_y + D_z - D_\xi - D_\zeta \end{aligned} \tag{22}$$

and

$$D_x = \left|\Delta x \frac{\partial G}{\partial x}\bigg|_{i_G}\right|, \quad D_y = \left|\Delta y \frac{\partial G}{\partial y}\bigg|_{i_G}\right|, \quad D_z = \left|\Delta z \frac{\partial G}{\partial z}\bigg|_{i_G}\right|. \tag{23}$$

The joined intersection volumes V_{cv,i_G} are calculated using CHIMPS [3], employing a Sutherland–Hodgman clipping procedure [54] to calculate the intersection volume between a Cartesian grid cell and convex tetra-, penta- and hexahedra.

The second term that is a function of the interface position is the surface tension force term, Eq. (2). This term could be calculated first on the G-grid, using a smoothed out version of the delta function δ_ϵ and then volume averaged to the flow solver grid,

$$\boldsymbol{T}_{\sigma_{cv}} = 1/V_{cv} \sum_{i_G} V_{cv,i_G} \boldsymbol{T}_{\sigma_{i_G}} = \frac{\sum_{i_G} V_{cv,i_G} \sigma \kappa_{i_G} \delta_\epsilon(G_{i_G}) (\nabla G)_{i_G}}{\sum_{i_G} V_{cv,i_G}}. \tag{24}$$

However, as will be seen later, this formulation is inconsistent with the balanced force algorithm. Instead, only the interface curvature is transferred from the G-grid to the flow solver grid,

$$\kappa_{cv} = \frac{\sum_{i_G} V_{cv,i_G} \delta_{i_G} \kappa_{i_G}}{\sum_{i_G} V_{cv,i_G} \delta_{i_G}}, \tag{25}$$

where $\delta_{i_G} = 0$ if $\psi_{i_G} = 0$ or $\psi_{i_G} = 1$ and $\delta_{i_G} = 1$ otherwise. The use of δ_{i_G} ensures that κ is treated as a surface quantity and not a volume quantity. The discrete form of evaluating κ_{i_G} on the G-grid will be discussed in a later section.

In order to couple the level set equation, Eq. (5), to the Navier–Stokes equations, $\boldsymbol{u}_{i_G}$ has to be calculated from $\boldsymbol{u}_{cv}$. Again the CHIMPS infrastructure is used and either tri-linear or C1, isotropic tri-cubic interpolation [26] is employed. It should be pointed out that strictly speaking neither one of these velocity interpolations can maintain a smooth curvature field under G-grid refinement. Let k_{int} be the discrete approximation on the G-grid to

$$k_{\text{int}} = \nabla \cdot (\nabla(\boldsymbol{u}_{i_G} \cdot \boldsymbol{n})). \tag{26}$$

To maintain smoothness of the curvature field, k_{int} would have to be continuous when switching between neighboring interpolation cells. Clearly, for tri-linear interpolation, this is not the case and even the isotropic tri-cubic interpolation [26] does not guarantee this property, since neither $\partial_{,xx}$ nor $\partial_{,yy}$ nor $\partial_{,zz}$ are kept continuous between neighboring interpolation cells. However, in the cases analyzed in this paper, tri-linear or tri-cubic interpolation was deemed sufficient.

3.3. Flow solver balanced force algorithm

The solution method of the Navier–Stokes equations is based on the fractional-step method for collocated variables on unstructured grids described in Mahesh et al. [29]. In the following, only the part of the algorithm that ensures discrete balance between surface tension forces and pressure gradient forces is outlined. It is based on the balanced force method for volume of fluid methods on collocated Cartesian grids [16].

For simplicity, we will omit the viscous term in the following discussion. The term is fully implemented and solved for in flux form, with the viscosity at the cell face calculated by the harmonic mean of the centroid viscosities of the two control volumes cv and nbr sharing the face f,

$$\mu_f = \frac{2\mu_{cv}\mu_{nbr}}{\mu_{cv} + \mu_{nbr}}. \tag{27}$$

The algorithm then reads

$$V_{cv} \frac{u^*_{i,cv} - u^n_{i,cv}}{\Delta t} + \sum_f u_f^{n+1/2} \frac{u^{n+1/2}_{i,cv} + u^{n+1/2}_{i,nbr}}{2} A_f = V_{cv} g + V_{cv} F^{n+1/2}_{i,cv} \tag{28}$$

$$\frac{u^{n+1}_{i,cv} - u^*_{i,cv}}{\Delta t} = -\frac{1}{\rho^{n+1/2}_{cv}} \frac{\partial p^{n+1/2}}{\partial x_i}, \tag{29}$$

where A_f is the face area, u_f the face normal velocity, $F_{i,cv}$ the density weighted surface tension force defined below, and superscripts denote time levels.

To define the force $F^{n+1/2}_{i,cv}$ at the control volume centroid, we first need to define the surface tension force at the cell face,

$$T^{n+1/2}_{\sigma_f} = \sigma \kappa_f^{n+1/2} (\nabla \psi)_f^{n+1/2}. \tag{30}$$

Here, the face curvature is calculated from the centroid curvature, Eq. (25),

$$\kappa_f^{n+1/2} = \frac{\alpha^{n+1/2}_{cv} \kappa^{n+1/2}_{cv} + \alpha^{n+1/2}_{nbr} \kappa^{n+1/2}_{nbr}}{\alpha^{n+1/2}_{cv} + \alpha^{n+1/2}_{nbr}} \tag{31}$$

with

$$\alpha_{cv}^{n+1/2} = \begin{cases} 1 : 0 < \psi_{cv}^{n+1/2} < 1 \\ 0 : \text{otherwise} \end{cases} \tag{32}$$

and

$$(\nabla\psi)_f^{n+1/2} = (\psi_{nbr}^{n+1/2} - \psi_{cv}^{n+1/2})/|\boldsymbol{s}_{cv,nbr}|. \tag{33}$$

Here, $\boldsymbol{s}_{cv,nbr}$ is the vector connecting the *cv* and *nbr* control volume centroids. Then, $F^{n+1/2}$ at the face becomes

$$F_f^{n+1/2} = T_{\sigma_f}^{n+1/2}/\rho_f^{n+1/2}, \tag{34}$$

with $\rho_f^{n+1/2} = (\rho_{cv}^{n+1/2} + \rho_{nbr}^{n+1/2})/2$. Finally, $F_f^{n+1/2}$ defined at the cell face needs to be transferred to the control volume centroid. It is crucial that for this, one uses exactly the same operation that is used for transferring $(\partial p/\partial n)_f$ to $(\partial p/\partial x_i)_{cv}$ in the pressure corrector step, Eq. (40). Here we use the face-area weighted least-squares method of Mahesh et al. [29] by minimizing

$$\epsilon_{cv} = \sum_f (F_{i,cv}^{n+1/2} n_{i,f} - F_f^{n+1/2})^2 A_f. \tag{35}$$

After solving Eq. (28) to obtain $u_{i,cv}^*$, the cell face normal velocities u_f^* are calculated,

$$u_f^* = \frac{1}{2}(u_{i,cv}^* + u_{i,nbr}^*)n_{i,f} - \frac{1}{2}\Delta t(F_{i,cv}^{n+1/2} + F_{i,nbr}^{n+1/2})n_{i,f} + \Delta t F_f^{n+1/2}. \tag{36}$$

This is essentially a modification of the procedure by Kim and Choi [24], first proposed by Young et al. [60]. To correct the face intermediate face velocities u_f^* to be divergence free, we then solve the following variable coefficient Poisson system,

$$\sum_f \frac{1}{\rho_f^{n+1/2}} \frac{\partial p^{n+1/2}}{\partial n} A_f = \frac{1}{\Delta t} \sum_f u_f^* A_f, \tag{37}$$

and then apply the correction

$$u_f^{n+1} = u_f^* - \Delta t P_f, \tag{38}$$

with

$$P_f = \frac{1}{\rho_f^{n+1/2}} (\nabla p^{n+1/2})_f = \frac{1}{\rho_f^{n+1/2}} \frac{p_{nbr}^{n+1/2} - p_{cv}^{n+1/2}}{|\boldsymbol{s}_{cv,nbr}|}. \tag{39}$$

Next, the centroid-based density weighted pressure gradient P_{cv} is calculated from the face-based density weighted gradient P_f using the same face-area weighted least-squares method employed in calculating F_f (see Eq. (35)),

$$\epsilon_{cv} = \sum_f (P_{i,cv} n_{i,f} - P_f)^2 A_f. \tag{40}$$

Finally, the control volume centroid velocity is corrected (cf. Eq. (29)),

$$u_{i,cv}^{n+1} = u_{i,cv}^* - \Delta t P_{i,cv}, \tag{41}$$

concluding the flow solver time step.

It should be pointed out that the control volume centroid velocities are not necessarily divergence free. This has implications for the mass-conservation property of any interface tracking scheme that uses these velocities. Indeed, as will be shown later, the non-divergence freeness of these velocities can lead to noticeable liquid volume conservation errors in certain test-cases. While staggered schemes do not have this shortcoming, collocated schemes are more suited for unstructured grids in complex geometries and are thus the method of choice here.

In addition to the usual CFL condition on the convective terms, the flow solver is subject to the capillary time step restriction [7], since surface tension forces are considered in an explicit way,

$$\Delta t \leqslant \Delta t_{\text{cap}} = \sqrt{\frac{(\rho_1 + \rho_2)h^3}{4\pi\sigma}}. \tag{42}$$

Here, h is the characteristic flow solver grid size.

3.3.1. Example: exact curvature equilibrium inviscid column and sphere

To illustrate the performance of the balanced force algorithm, we analyze the canonical test cases of the equilibrium inviscid column and sphere. In this case, the surface tension forces should exactly balance the pressure jump across the phase interface, resulting in the column and sphere remaining perfectly at rest.

It is instructive to analyze this test case for different grid layouts and interface tracking techniques. In cylindrical (spherical) coordinates, for the column (sphere) of radius R to remain at rest

$$\frac{\partial p}{\partial r} = -\sigma\kappa\delta(r - R) = -C\delta(r - R) \tag{43}$$

must hold, with both σ and κ known and assumed constant, $C = \sigma\kappa = \text{const}$. For VoF methods on staggered grids using the CSF method, the discrete form of Eq. (43) yields

$$\frac{\partial_h p}{\partial r} = C\frac{\partial_h \psi}{\partial r}. \tag{44}$$

It is easy to see that here, if the same discrete gradient operators are used for p and ψ, discrete balance is automatic and no special algorithmic steps must be taken, see for example [42,45]. For level set methods on staggered grids using a smoothed delta function, Eq. (43) in discrete form is

$$\frac{\partial_h p}{\partial r} = -C\delta_{\epsilon,h}(G - G_0). \tag{45}$$

Using the popular cosine approximation to $\delta_{\epsilon,h}$ [38] this does not ensure discrete balance. The solution to this inconsistency is, however, straightforward: simply choose $\delta_{\epsilon,h}$ to be the discrete gradient of an appropriate scalar, e.g. the exact Heaviside transform of the level set scalar as proposed by Sussman et al. [49] or even a mollified version of the Heaviside transform [52]. Then,

$$\frac{\partial_h p}{\partial r} = C\frac{\partial_h H_{\epsilon,h}(G - G_0)}{\partial r}, \tag{46}$$

and discrete balance is again automatic. As such, the schemes proposed by Sussmann et al. [52] for the stream function formulation and Sussman et al. [49] for non-collocated grids already constitute level set balanced force algorithms.

On collocated grids, discrete balance is not straightforward, since in the fractional step method, pressure is used to project and correct the face velocities, see Eq. (38). Hence the pressure gradient is not directly defined at the location of the control-volume velocities. Thus, Eq. (43) on collocated grids for VoF methods reads

$$\left(\frac{\partial_h p}{\partial r}\right)_{f\to cv} = C\frac{\partial_h \psi}{\partial r}, \tag{47}$$

where $f \to cv$ denotes an operation to calculate cv centroid values from face values, see Eq. (40) and the right hand side is typically directly evaluated at the centroid position. Again, discrete balance is not achieved, unless the balanced force methodology of Francois et al. [16] is followed, resulting in

$$\left(\frac{\partial_h p}{\partial r}\right)_{f\to cv} = \left(C\frac{\partial_h \psi}{\partial r}\right)_{f\to cv}, \tag{48}$$

where the inside of both brackets is to be evaluated at the face position. For level set methods on collocated grids the same observations as on staggered grids hold true, disqualifying the use of a smoothed delta function.

Instead, the gradient of an appropriate scalar field must be used, e.g. either the exact or mollified Heaviside transform of the level set scalar. This results in

$$\left(\frac{\partial_h p}{\partial r}\right)_{f\to cv} = \left(C\frac{\partial_h H_{\epsilon,h}(G-G_0)}{\partial r}\right)_{f\to cv} \tag{49}$$

and thus a balanced force algorithm on collocated grids. Throughout this paper, Eq. (48) instead of Eq. (49) is used, since the volume fraction ψ is readily available.

To demonstrate the performance of the balanced force algorithm, we employ the test case parameters suggested by Williams et al. [58] and used by Francois et al. [16]: a column (or sphere), of radius $R = 2$ is placed at the center of an $8 \times 8(\times 8)$ domain. The surface tension coefficient σ is set to 73, resulting in a theoretical pressure jump across the interface of $\Delta p^{\text{ex}} = 36.5$ for the column and $\Delta p^{\text{ex}} = 73$ for the sphere. The density inside the column/sphere is set to $\rho_1 = 1$ and the density outside the column/sphere ρ_2 is varied. Equidistant Cartesian and unstructured prism and tetrahedral flow solver grids are tested. The flow solver grid is characterized by the characteristic grid size h, whereas the equidistant Cartesian G-grid size is denoted by h_{G}.

The error in pressure is measured in two different ways [16],

$$E(\Delta p_{\max}) = \max(p_{cv}) - \min(p_{cv}) - \Delta p^{\text{ex}} \tag{50}$$

$$E(\Delta p_{\text{part}}) = \overline{p_{cv}}|_{r\leqslant R/2} - \overline{p_{cv}}|_{r\geqslant 3R/2} - \Delta p^{\text{ex}}, \tag{51}$$

where the bar indicates an arithmetic average over all control volumes fulfilling the given condition.

Table 1 summarizes the errors in velocity, pressure, and kinetic energy E_{kin} for the column after a single time step of size $\Delta t = 10^{-6}$ for varying density ratios if the exact curvature of the column is used for $\kappa_{i_{\text{G}}}$ in Eq. (25). As can be seen, both on the Cartesian and the unstructured prism flow solver grid, errors are machine precision zero, even for extremely large density ratios.

Table 2 summarizes the same quantities in the sphere test case. Again, both on the Cartesian and the tetrahedral flow solver grid, machine precision zero errors are achieved for varying density ratios, if the exact curvature is employed.

Thus, provided that the exact curvature is known, the balanced force algorithm results in machine zero spurious currents, even in the inviscid case. However, the exact curvature is rarely known, instead it has to be evaluated and is prone to errors. These curvature errors are then the sole source of error for spurious currents.

Table 1
Errors in velocity and pressure after single time step $\Delta t = 10^{-6}$ for varying density ratio in the inviscid, equilibrium column test case using exact curvature and $h = h_{\text{G}} = 0.2$

ρ_1/ρ_2	Cartesian				Prism			
	$L_\infty(u)$	E_{kin}	$E(\Delta p_{\max})$	$E(\Delta p_{\text{part}})$	$L_\infty(u)$	E_{kin}	$E(\Delta p_{\max})$	$E(\Delta p_{\text{part}})$
1	1.82e−16	1.06e−32	1.82e−16	4.58e−13	6.76e−20	6.86e−40	4.14e−16	3.41e−15
10^3	2.55e−16	7.84e−36	2.55e−16	1.97e−15	3.59e−17	6.28e−35	9.90e−14	9.21e−13
10^5	5.43e−16	8.43e−38	5.43e−16	2.29e−15	9.93e−19	2.26e−40	3.16e−16	1.58e−15
10^{10}	3.93e−18	3.39e−39	3.93e−18	7.40e−15	9.77e−19	1.74e−40	7.30e−16	1.19e−15

Cartesian flow solver grid (left) and prism flow solver grid (right).

Table 2
Errors in velocity and pressure after single time step $\Delta t = 10^{-6}$ for varying density ratio in the inviscid, equilibrium sphere test case using exact curvature and $h = h_{\text{G}} = 0.2$

ρ_1/ρ_2	Cartesian				Prism			
	$L_\infty(u)$	E_{kin}	$E(\Delta p_{\max})$	$E(\Delta p_{\text{part}})$	$L_\infty(u)$	E_{kin}	$E(\Delta p_{\max})$	$E(\Delta p_{\text{part}})$
1	5.49e−17	6.95e−34	6.57e−14	1.19e−12	3.31e−16	8.20e−33	1.31e−11	1.25e−14
10^3	1.15e−17	1.40e−36	1.15e−14	9.41e−14	1.07e−15	5.96e−35	4.81e−14	1.50e−15
10^5	1.21e−16	3.30e−39	5.30e−15	1.38e−14	2.58e−15	2.76e−36	1.42e−14	1.02e−15
10^{10}	5.47e−16	8.24e−36	3.54e−13	2.74e−13	9.95e−11	2.73e−35	2.40e−15	5.60e−16

Cartesian flow solver grid (left) and tetrahedral flow solver grid (right).

3.4. RLSG curvature evaluation

As noted in the previous section, only curvature errors result in spurious currents when employing the balanced force algorithm. Hence the task of minimizing spurious currents is equivalent to increasing the accuracy of curvature evaluation. In standard level set methods [44], curvature is evaluated at G-node locations by discretizing Eq. (9),

$$\kappa = \frac{G_{,xx}^2(G_{,y}^2 + G_{,z}^2) + G_{,yy}(G_{,x}^2 + G_{,z}^2) + G_{,zz}(G_{,x}^2 + G_{,y}^2)}{(G_{,x}^2 + G_{,y}^2 + G_{,z}^2)^{3/2}} - 2\frac{G_{,xy}G_{,x}G_{,y} + G_{,xz}G_{,x}G_{,z} + G_{,yz}G_{,y}G_{,z}}{(G_{,x}^2 + G_{,y}^2 + G_{,z}^2)^{3/2}}, \tag{52}$$

typically using a 27-point stencil. It is important to point out that this approach approximates the curvature of the G-isosurface that passes through the nodal point itself. It is therefore, at best, a first-order approximation to the curvature of the phase interface, which can be a distance h_G away from nodes directly adjacent to the interface (see Fig. 7). Figs. 8 and 9 demonstrate this first-order convergence rate under G-grid refinement for both the column and the sphere test case using either Cartesian flow solver grids (column and sphere), unstructured prism grids (column), or tetrahedral grids (sphere) with $h = 0.2$.

Since the root cause of the first-order convergence rate is the fact that curvature is not calculated at the interface itself, different approaches can be taken to overcome this problem. Introducing a polynomial representation of the interface in terms of interface-based coordinates is a viable approach in two dimensions, but becomes cumbersome in three dimensions. Here, we will follow an alternative approach using the fact that a quantity defined only on the interface itself, like curvature, can be distributed to the whole computational domain in a meaningful way by solving

$$\nabla\kappa \cdot \nabla G = 0. \tag{53}$$

This effectively sets κ constant in the front normal direction. Note that due to Eq. (25), Eq. (53) needs to be solved only for G-nodes adjacent to the interface. The problem is therefore similar to determining the initially accepted values in the Fast Marching Method [2]. For this purpose, Chopp [10] developed a Newton's method that determines the nearest point on the interface (called "base-point" in the following) for a given node in two dimensions. The method relies on approximating the level set scalar within each computational cell close to the interface by a bi-cubic spline. For this purpose, G need not be a distance function. We have extended Chopp's method to three dimensions using C1, isotropic tri-cubic interpolations [26]. We typically find the base-point within 2–4 Newton iterations. However, in some situations, the base-point is not in the interior of the region in which the tricubic approximation was taken but in an adjacent region, also bounded by the same node. In this case, the base-point is rejected, unless none of the alternative seven regions the node belongs to yields a valid base-point. Once the base-point coordinates have been determined, the base-point's curvature is calculated by tri-linear interpolation using the surrounding nodal curvature values. Using Eq. (53), the nodal curvature is then set equal to its base-point's curvature.

The resulting curvature errors under G-grid refinement using Chopp's method are shown in Figs. 8 and 9. They show second-order convergence and even on coarse grids, Chopp's method yields more than one order of magnitude better curvature estimates than the nodal-based evaluation. The drawback of Chopp's method is that for complex interface geometries in three dimensions, the Newton algorithm does not always converge.

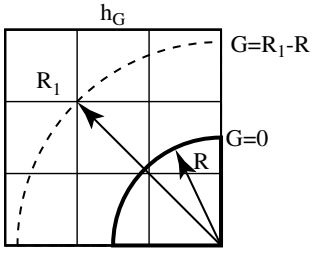


Fig. 7. Inherent phase interface curvature error when evaluating curvature at nodes. Curvature is determined to be $\kappa = 1/R_1 = 1/(R + O(h_G))$ instead of $\kappa = 1/R$.

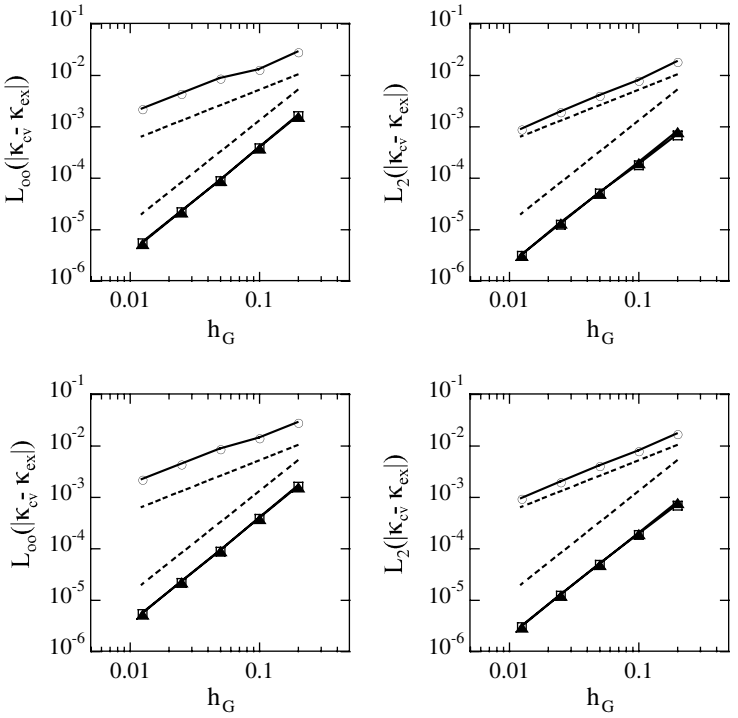


Fig. 8. Initial column curvature errors under G-grid refinement; flow solver grid $h = 0.2$; Cartesian flow solver grid (top), prism flow solver grid (bottom); nodal curvature (circles), direct front curvature (squares), Chopp front curvature (triangles), first-and second-order convergence (dashed lines).

The method thus lacks the stability required for complex interface geometries typically found in liquid/gas flows. Thus, the following method is proposed as an alternative.

Assuming that G is smooth in the vicinity of the phase interface, the base-point $\boldsymbol{x}_B$ for a given node $\boldsymbol{x}_G$ close to the interface can be explicitly calculated from

$$\boldsymbol{x}_B = \boldsymbol{x}_G - \mathrm{d}\boldsymbol{n} = \boldsymbol{x}_G - \frac{G}{|\nabla G|}\frac{\nabla G}{|\nabla G|}, \tag{54}$$

where all gradients are calculated using central differences. This approach is termed *direct front curvature* in the following. It gives good base-point estimates only for nodes close to the interface. However, due to the way Eq. (25) is evaluated, κ needs to be calculated only on nodes close to the interface, making the direct front curvature method viable. As before, once base-points have been determined, their curvature is again calculated using tri-linear interpolation from the surrounding nodal curvature values. Then, according to Eq. (53), the curvature values of nodes are set to their respective base-points' curvature values. Figs. 8 and 9 also include the curvature errors calculated by the direct method. As can be seen, they are virtually indistinguishable from the values obtained using Chopp's method yielding second-order convergence.

Comparing the obtained curvature errors to those calculated by Francois et al. [16], both Chopp's and the direct method give curvature errors an approximate factor of 5 lower than the 7×3 stencil height function method employed in that paper. While a height function approach could be employed in the RLSG method as well, since volume fractions ψ are readily available, the effective G-stencil needed would be $9 \times 5 \times 5$ (cf. Eqs. (17)–(19)), as compared to $4 \times 4 \times 4$ in the direct front curvature method. Smaller stencil sizes are especially important for complex interface geometries, since both the height function and all level set curvature methods are based on the assumption that all ψ and G values in the stencil relate to one continuous interface segment only. Auxiliary, non-contiguous, interface segments inside the stencil can introduce significant errors. These can only be avoided by temporarily removing the auxiliary interface segments during curvature

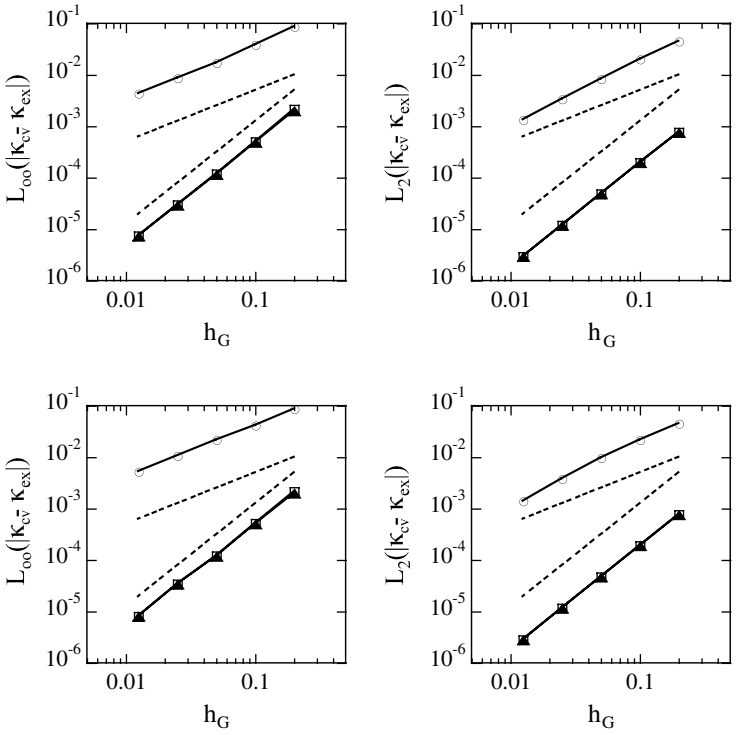


Fig. 9. Initial sphere curvature errors under G-grid refinement; flow solver grid $h = 0.2$; Cartesian flow solver grid (top), prism flow solver grid (bottom); nodal curvature (circles), direct front curvature (squares), Chopp front curvature (triangles), first-and second-order convergence (dashed lines).

calculation. Such methods have been recently proposed by Macklin and Lowengrub [28] for level sets and Sussman and Ohta [50] for height functions.

In the following, we will employ the direct front curvature method to calculate nodal curvature values on the G-grid. Fig. 10 shows the errors in velocity and pressure after a single time step of size $\Delta t = 10^{-6}$, using $\rho_1 = 1$ and $\rho_2 = 10^{-3}$ and refining the resolution h_G of the G-grid. As expected, due to the balanced force algorithm, errors in curvature evaluation result in errors in velocity and pressure, showing the same second-order convergence behavior (cf. Figs. 8 and 9).

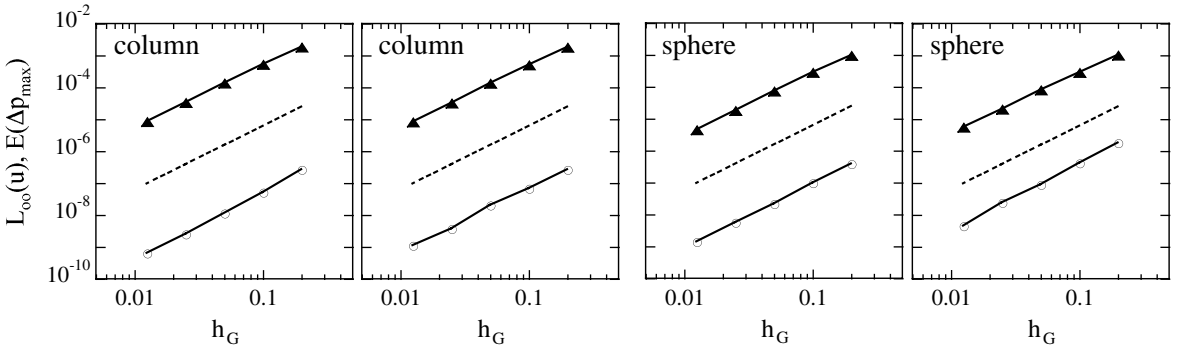


Fig. 10. Equilibrium inviscid column and sphere velocity (circle) and pressure (triangle) errors after 1 time step $\Delta t = 10^{-6}$ under G-grid refinement; flow solver grid $h = 0.2$, density ratio $\rho_1/\rho_2 = 10^3$; from left to right: column Cartesian flow solver grid, column prism flow solver grid, sphere Cartesian flow solver grid and sphere tetrahedral flow solver grid; dashed lines mark second-order convergence.

3.5. RLSG parallel implementation and load balancing

Parallel efficiency and scalability is key to allow extremely high resolution G-grids. In typical applications involving tracked interfaces, interfaces make up a small fraction of the flow solver volume only. Furthermore, the region occupied by the interface is not static. An efficient domain decomposition strategy for the flow solver is thus usually inefficient for the RLSG solver and vice versa. Dual constraint partitioning is possible, however, the dynamic movement of the interface through the flow solver domain would require frequent joined repartitioning of the G-grid and the otherwise static flow solver grid. Therefore, we propose to use independent partitioning for the flow solver grid and the G-grid. The former is domain decomposed at the beginning of a simulation and retains its partitioning throughout the simulation. The latter is dynamically decomposed and load balanced throughout the simulation, once a load-imbalance factor is greater than some threshold value.

Domain decomposition of the G-grid is straightforward using the super-grid. At the beginning of the simulation, all super-grid cells, active or not, are assigned to a processor and the negative value of the processor rank is stored in a super-grid i, j, k lookup table that is synchronized on all processors. Thus, if a band grows into a previously not activated super-grid block, a unique processor is already assigned to this block. After the band regeneration step, this strategy can lead to a load-imbalance in terms of number of local active cells inside the $\mathcal{T}$-and $\mathcal{N}$-band. To re-loadbalance, the processor with the largest number of active cells sends its super-grid block containing the most active cells to the processor with the smallest number of active cells. This process is repeated until either the threshold imbalance factor is reached, or the load balance does not decrease any more. Then, the super-grid i, j, k lookup table is synchronized to reflect any changed super-grid block processor ranks. Note that more advanced strategies also minimizing the domain edge count can be devised, however the strategy outlined above leads to good parallel performance and load-balance. Fig. 11 shows the parallel scalability of the RLSG solver using Zalesak's disk test case described in a later section with a Cartesian flow solver grid of $h = 1/128$ and a G-grid of $h_G = 1/1024$. A nearly constant, although slightly sub-optimal speed-up is observed over all counts of processors n_p.

4. Results

This section presents a range of test cases to illustrate the ability of the proposed method. First, two canonical interface tracking test cases with prescribed velocity fields are presented to test the interface tracking capabilities of the RLSG method. Then, the balanced force algorithm in conjunction with the refinement and grid convergence capabilities of the RLSG method is tested by calculating the long-time evolution of inviscid and viscous columns and spheres. Following, results for zero-gravity oscillating columns and spheres are presented. Next, the interplay of viscous and surface tension forces is verified using the analytical solution of a damped surface wave. Finally, to demonstrate the capability of the proposed method to eliminate interface tracking errors and to achieve grid converged solutions the results of a Rayleigh–Taylor study are presented.

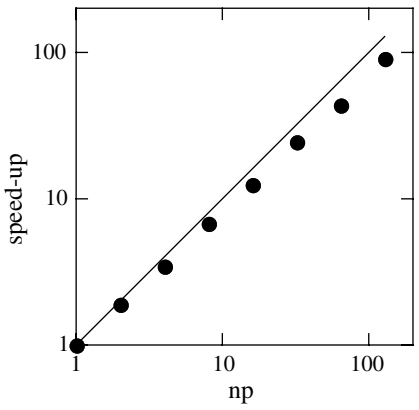


Fig. 11. RLSG distributed memory parallel speed-up for Zalesak's disk, $h = 1/128$ and $h_G = 1/1024$; line denotes optimal results, symbols marks RLSG solver results.

In the following subsections, h denotes the edge length of a flow solver grid element, be it a Cartesian hexahedral, prism, or tetrahedral element and h_G denotes the edge length of an equidistant Cartesian G-grid cell. All simulations are formally performed in three dimensions, however, if the test-case is not inherently three-dimensional, the third direction is resolved by a single element using periodic boundary conditions in that direction. The total liquid volume is calculated from the flow solver volume fractions:

$$V_1(t) = \sum_{cv} \psi_{cv}(t) V_{cv}, \tag{55}$$

resulting in the following definition of the volume error E_V and % volume loss,

$$E_V = \frac{|V_1(t=0) - V_1(t)|}{V_1(t=0)} \tag{56}$$

$$\%\text{volume loss} = \frac{V_1(t=0) - V_1(t)}{V_1(t=0)} \times 100\%. \tag{57}$$

The shape error E_{shape} is a measure for the amount of fluid that has propagated to the wrong side of the exact interface location. It is calculated by first dividing each G-grid cell into 1000 sub-cells in each coordinate direction, using tri-linear interpolation to calculate the level set scalar value G_s at each sub-cell centroid and then evaluating

$$E_{shape}(t) = \frac{\frac{1}{L}\sum_{i_G}\sum_{\text{sub-cells}} |H(G_s(t)) - H(G_s(t=0))| V_{\text{sub-cell}}}{2\sum_{i_G} V_{i_G}}, \tag{58}$$

with L the exact interface length or area at $t = 0$.

4.1. Zalesak's disk

The solid body rotation of a notched circle, also known as Zalesak's disk [61], is one of the standard test problems for evaluating the accuracy of level set methods in maintaining sharp corners. A disk of radius 15, notch width 5 and notch height 25 is placed in a 100×100 box at $(50, 75)$. The velocity field is given by

$$\boldsymbol{u}(\boldsymbol{x}, t) = (50 - y, x - 50)^T \tag{59}$$

and the flow solver constant time step size is set to $\Delta t = 2\pi/628$. Fig. 12 shows the shape of the interface at $t = 2\pi$ after one full rotation of the disk using an equidistant Cartesian flow solver grid with $h = 1$. Due to the linearity of the imposed velocity field, results for the flow solver prism grid are identical to those of the Cartesian flow solver grid and are thus not shown here. As is typical of level set methods, the sharp corners of the notched disk tend to be rounded, however, the overall shape of the notch is well preserved, even on the coarsest G-grid of $h_G = 1$. Refining the G-grid continuously improves the shape, until for the finest G-grid resolution visually no difference can be discerned to the exact solution. Table 3 summarizes the volume and shape errors for the different G-grid resolutions. It should be pointed out that the volume (or area) error in this particular test case is not a good criterion, since the simultaneous rounding of all four sharp corners leads to a

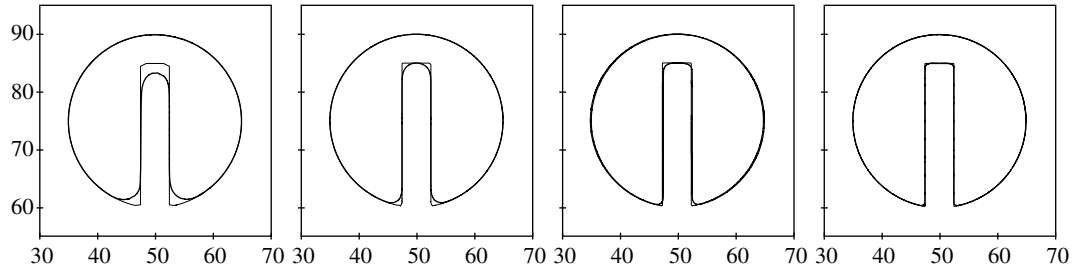


Fig. 12. Zalesak's disk after one full rotation with $h = 1$; exact solution (thin line) and $h_G = 1$, $h_G = 1/2$, $h_G = 1/4$ and $h_G = 1/8$ (from left to right).

Table 3
Volume and shape errors of Zalesak's disk after one full rotation with $h = 1$

h_G	% Volume loss	E_{shape}	$Order_{shape}$
1	−0.461	0.1611	–
1/2	0.101	0.0419	1.94
1/4	−0.013	0.0144	1.54
1/8	−0.007	0.0045	1.66

simultaneous volume loss and gain that can cancel. It is thus given for completeness only. The obtained shape error is comparable to that obtained using a HJ particle level set method [14] when using a factor two refined G-grid ($h_G = 1/2$) for the level set scalar. However, refining the G-grid, the shape error can be significantly reduced resulting in error levels that are significant smaller than the particle level set method.

4.2. Column in a deformation field

The column or circle in a deformation field problem introduced by Bell et al. [5] and applied as a level set test problem by Enright et al. [13] tests the ability of the level set method to resolve and maintain ever thinner filaments. A column of radius $R_0 = 0.15$ and center $(0.5, 0.75)^T$ is placed inside a unit sized box. The velocity field is given by the stream function

$$\Psi(\boldsymbol{x}, t) = \frac{1}{\pi} \sin^2(\pi x) \cos^2(\pi y) \cos(\pi t / T). \tag{60}$$

The resulting velocity field first stretches the column into ever thinner filaments that are wrapped around the center of the box, then slowly reverses and pulls the filaments back into the initial circular shape. All simulations are performed on a flow solver grid with $h = 1/128$ and $T = 8$ using a constant flow solver time step of $\Delta t = 1/256$. Fig. 13 shows the interface shape at the moment of maximum extension $t = 4$ for varying G-grid resolutions, while Fig. 14 shows the interface shape at $t = 8$ after reversal has been completed. For comparison purposes, the thin line in the figures denotes a high-resolution solution obtained using $h = 1/512$ and $h_G = 1/2048$. Two regions where errors predominantly occur are easy to identify: the trailing and the leading edge of the filament. The volume in those regions corresponds to the upper region and the lower region respectively of the initial column. The underlying reason for the poor volume conservation in the coarse G-grid case ($h_G = 1/128$) is twofold. For one, the incorrect merging of characteristics in the transport step, but especially during re-initialization moves the interface, resulting in annihilation of thin filament structures. But even if perfect transport and re-initialization could be performed, the $h_G = 1/128$ solution would loose significant amounts of volume because the trailing filament thickness falls below the grid resolution of $h_G = 1/128$ and is thus not resolvable by a fixed grid method.

Continuously refining the G-grid results in ever better shape and volume preservation, resulting in roughly second-order grid convergence under G-grid refinement, see Table 4. The obtained results compare well to the

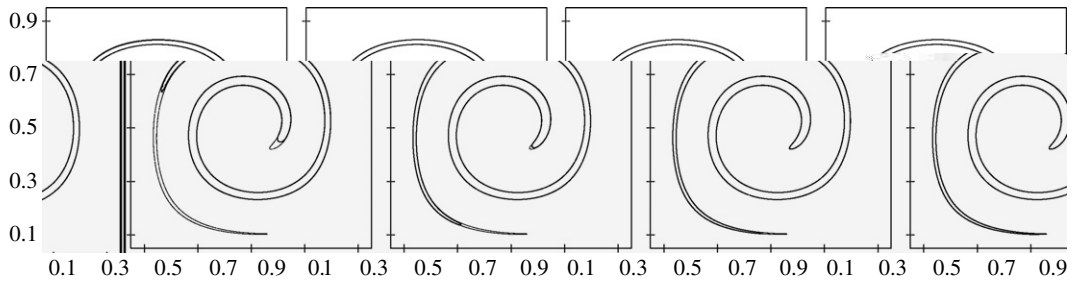


Fig. 13. Interface shape of column in a deformation field at $t = T/2$, $h = 1/128$; target solution (thin line) and $h_G = 1/128$, $h_G = 1/256$, $h_G = 1/512$ and $h_G = 1/1024$ (from left to right).

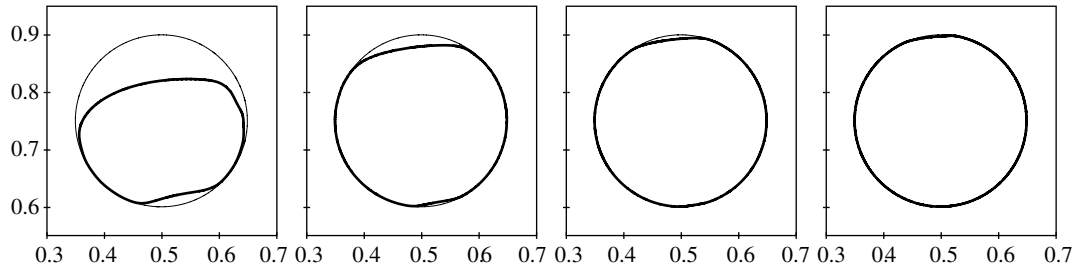


Fig. 14. Interface shape of column in a deformation field at $t = T$, $h = 1/128$; target solution (thin line) and $h_G = 1/128$, $h_G = 1/256$, $h_G = 1/512$ and $h_G = 1/1024$ (from left to right).

Table 4
Volume and shape errors of a column in a deformation field at $t = T$ with Cartesian flow solver grid and $h = 1/128$

h_G	% Volume loss	Order	E_{shape}	Order
1/128	30.86	–	0.02315	–
1/256	4.61	2.74	0.00363	2.67
1/512	1.00	2.20	0.00084	2.11
1/1024	0.28	1.87	0.00024	1.81

HJ particle level set method [14], resulting in slightly larger errors for a factor 2 finer G-grid and slightly smaller errors for a factor 4 refined G-grid.

Table 5 summarizes the results obtained using a prism flow solver grid. Since the prescribed velocity field is not linear, the employed tri-linear interpolation on a Cartesian and a prism element causes minute differences. While the interface shapes presented in Figs. 13 and 14 look visually identical, the error norms show a slight increase in error for the prism case.

4.3. Long-time evolution of the equilibrium inviscid column and sphere

As shown in Section 3.3, errors in curvature evaluation result in spurious currents that are small, but non-zero. Thus, the long-time behavior of the equilibrium column and sphere is of interest, since errors might accumulate and result in large erroneous velocities.

Fig. 15 shows the temporal evolution of the kinetic energy in the computational domain for both the inviscid column and sphere on Cartesian and unstructured flow solver grids. The flow solver characteristic grid size is $h = 0.4$ in all simulations, $\rho_1 = 1$, $\rho_2 = 10^{-3}$, $\sigma = 73$ and the fixed time step size is $\Delta t = 10^{-3}$. As observed by Francois et al. [16], the column seems to enter an oscillatory mode that appears quite stable on a Cartesian flow solver grid (top left), but shows a slight growth on the prism flow solver grid (top right). The inviscid sphere results, on the other hand, do not exhibit such a clear periodic behavior. In the Cartesian flow solver grid case, different periods seem to be superposed and the unstructured tetrahedral grid shows an increase in kinetic energy without reaching a periodic state. This is due to the fact that the unstructured grid lacks the symmetry of the Cartesian grids. This symmetry seems to initiate a periodic oscillation instead of a constant growth in spurious currents and is thus beneficial in this particular test case, but not indicative of the method's

Table 5
Volume and shape errors of a column in a deformation field at $t = T$ with prism flow solver grid and $h = 1/128$

h_G	% Volume loss	Order	E_{shape}	Order
1/128	30.98	–	0.02325	–
1/256	4.64	2.74	0.00370	2.65
1/512	1.03	2.17	0.00088	2.07
1/1024	0.28	1.88	0.00026	1.76

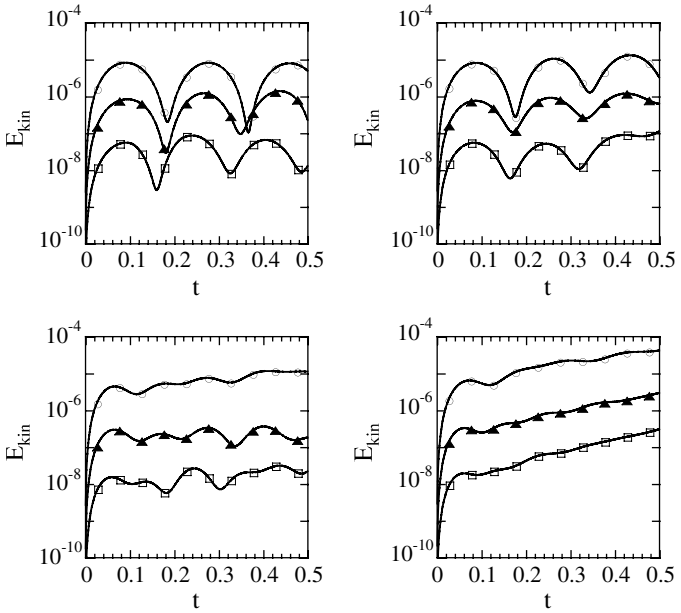


Fig. 15. Temporal evolution of kinetic energy for equilibrium inviscid column (top) and sphere (bottom) for 500 time steps under G-grid refinement: $h_G = 0.4$ (circle), $h_G = 0.2$ (triangle), $h_G = 0.1$ (box); flow solver grid $h = 0.4$, density ratio $\rho_1/\rho_2 = 10^3$; Cartesian flow solver grids (top left and bottom left), prism flow solver grid (top right) and tetrahedral flow solver grid (bottom right).

performance in a more general setting. In the general case, one can expect a growth of the kinetic energy along the lines of the unstructured grid results, necessitating at later times the use of viscous dissipation to control the spurious currents. However, the balanced force method exhibits very low levels of spurious currents and indeed, this level can be made even smaller if the G-grid is refined to increase the accuracy of the interface curvature evaluation. Fig. 16 shows the convergence rates for the maximum velocity error under G-grid refinement, using a flow solver resolution of $h = 0.4$. Close to second-order convergence can be observed both on structured and unstructured flow solver grids.

4.4. Long time evolution of the viscous equilibrium column and sphere

To evaluate the performance of the proposed RLSG balanced force method as compared to other numerical methods, we perform simulations of the long time evolution of viscous columns and spheres at equilibrium. A column of diameter $D = 0.4$ is placed in the center of a unit sized box resolved by either Cartesian

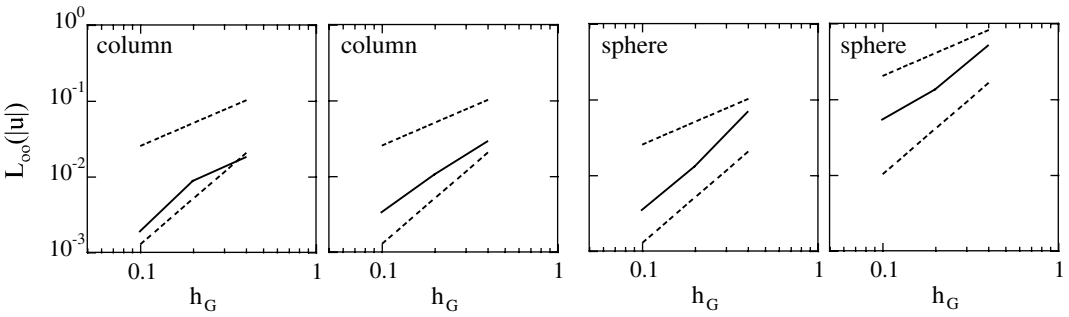


Fig. 16. Equilibrium inviscid column and sphere maximum velocity error during 500 time steps under G-grid refinement; flow solver grid $h = 0.4$, density ratio $\rho_1/\rho_2 = 10^3$; from left to right: column Cartesian flow solver grid, column prism flow solver grid, sphere Cartesian flow solver grid and sphere tetrahedral flow solver grid; dashed lines mark first and second-order convergence.

or prism cells with $h = 1/32$. The viscosity in both fluids is set to $\mu = 0.1$ and the surface tension coefficient is set to $\sigma = 1$. The Laplace number $La = 1/Oh^2 = \sigma\rho D/\mu^2$ is varied by changing the density in both fluids, keeping the density ratio fixed as $\rho_1/\rho_2 = 1$. The time step size is chosen to be $\Delta t = \Delta t_{\text{cap}}/2$, see Eq. (42). Table 6 compares the capillary number $Ca = |u_{\max}|\mu/\sigma$ for varying Laplace numbers at $t\sigma/(D\mu) = 250$ using a Cartesian flow solver grid to the benchmark results of Popinet and Zaleski [39] using a marker tracking method and Shin et al. [45] using a level contouring approach. The table also shows results obtained on the prism grid compared to the results of Marchandise et al. [30] employing a finite element method on a slightly finer unstructured grid. Results with the RLSG method show more than one order of magnitude smaller spurious currents for smaller values of the Laplace numbers on Cartesian grids and more than two-orders of magnitude better results on non-Cartesian grids. However, at very large Laplace numbers, the present results are not fully independent of the Laplace number due to small, decaying periodic oscillations in the spurious current magnitude that are present in the inviscid case as well, see Fig. 15.

Table 7 analyzes the grid convergence behavior. Again, the spurious currents of the RLSG method are significantly lower than those reported by Popinet and Zaleski [39] and convergence under grid refinement is roughly second order. Table 8 demonstrates that the reported capillary numbers are virtually independent of the chosen time step size, even for timestep sizes close to the stability limit.

Finally, Table 9 shows the spurious current capillary number for a fixed flow solver grid under G-grid refinement using a time step of $\Delta t = 0.9\Delta t_{\text{cap}}$ with Δt_{cap} based on the flow solver grid size $h = 1/16$. The results demonstrate that the capillary time step restriction is not based on the RLSG grid size h_{G}, but rather on the flow solver grid size h as expected, since at the finest G-grid ($h_{\text{G}} = 1/128$), the employed time step is more than 20 times larger than a stable capillary time step based on h_{G}. Overall convergence under G-grid refinement is approximately first order.

As a three dimensional test, the viscous sphere of Renardy and Renardy [42] is calculated. Here, a sphere of diameter $D = 0.25$ is placed at the center of a unit sized Cartesian box with symmetry boundary conditions at all sides. The surface tension coefficient is set to $\sigma = 0.357$, density and viscosity in both fluids are $\rho = 4$ and

Table 6
Dependence of the spurious current capillary number on the Laplace number for viscous equilibrium column and $D/h = D/h_{\text{G}} = 12.8$ as compared to Popinet and Zaleski [39] and Shin et al. [45] on Cartesian grid (top) and Marchandise et al. ($D/h = 16$) [30] on prism grid

	La					
	12	120	1200	12,000	120,000	1,200,000
Ca Cartesian	0.10e−6	0.11e−6	0.12e−6	1.44e−6	3.09e−6	0.71e−6
Ca [39]	6.76e−6	5.71e−6	5.99e−6	8.76e−6	–	–
Ca [45]	2.18e−6	2.18e−6	2.18e−6	2.22e−6	–	–
Ca prism	0.15e−6	0.15e−6	0.16e−6	2.17e−6	3.93e−6	0.96e−6
Ca [30]	85.1e−6	86.2e−6	85.9e−6	83.1e−6	–	–

Table 7
Grid convergence of the spurious current capillary number for viscous equilibrium column and $1/Oh^2 = 12,000$ compared to the results by Popinet and Zaleski [39]

	h, h_{G}			
	1/16	1/32	1/64	1/128
Ca	4.92e−6	1.44e−6	0.34e−6	0.05e−6
Ca [39]	37.60e−5	6.68e−6	1.07e−6	0.12e−6

Table 8
Time step convergence of the spurious current capillary number for viscous equilibrium column at $h = h_{\text{G}} = 1/32$ and $1/Oh^2 = 1200$

	$\Delta t/\Delta t_{\text{cap}}$			
	1	1/2	1/4	1/8
Ca	0.12e−6	0.12e−6	0.12e−6	0.13e−6

Table 9
Grid convergence of the spurious current capillary number for viscous equilibrium column, $1/Oh^2 = 12,000$ and fixed flow solver grid $h = 1/16$

	h_G			
	1/16	1/32	1/64	1/128
Ca	4.92e−6	4.46e−6	2.10e−6	8.92e−7
Order	–	0.14	1.09	1.24

$\mu = 1$, resulting in a Laplace number of $1/Oh^2 = 0.357$. The time step size is $\Delta t = 10^{-5}$ and the simulation is run for 200 time steps. Table 10 compares the maximum spurious current capillary number at $t = 0.002$ to those obtained by Renardy and Renardy [42] using the PROST volume of fluid algorithm on a staggered mesh. Although the RLSG method results in slightly smaller values, both methods yield comparable, high quality results.

4.5. Zero gravity column/drop oscillation

To further verify the implementation of the balanced force algorithm, this section presents results for zero gravity oscillating columns and spheres. The theoretical oscillation period for columns in the linear regime is given by [25]

$$\omega^2 = \frac{n(n^2-1)\sigma}{(\rho_1+\rho_2)R_0^3}, \tag{61}$$

while that for spheres can be calculated from [25]

$$\omega^2 = \frac{n(n^2-1)(n+2)\sigma}{[(n+1)\rho_1+n\rho_2]R_0^3}. \tag{62}$$

In all simulations a column respectively sphere of radius $R_0 = 2$ is placed in the center of a $[-10, 10]$ square box with slip boundary conditions on all sides and $\sigma = 1$, $\rho_1 = 1$, $\rho_2 = 0.01$, $\mu_1 = 0.01$ and $\mu_2 = 1 \cdot 10^{-4}$, resulting in a Laplace number of $La = 20000$. The column/sphere is initially perturbed by a mode $n = 2$ perturbation with an initial amplitude of $A_0 = 0.01R_0$. The time step size in all simulations is chosen as $\Delta t = 0.5\Delta t_{\text{cap}}$.

Table 11 shows the period of oscillation error $E_T = |T_{\text{calc}}\omega/2\pi - 1|$ for the oscillating column together with the results reported by Torres and Brackbill [55], whereas Table 13 lists the corresponding results for the oscillating sphere. Results better by a factor of roughly 2 are obtained even on prism grids as compared to the results reported in [55] and the oscillating sphere results show second order grid convergence. Table 12 analyzes the temporal convergence. Simulations are stable even at the capillary time step limit and show slightly better than first order convergence under time step refinement.

4.6. Damped surface waves

To verify the interplay of the surface tension term with the viscous terms we compare the results of the proposed method to the initial value theory of Prosperetti [40] for a small amplitude damped surface wave

Table 10
Grid convergence of the spurious current capillary number for viscous equilibrium sphere with parameters due to Renardy and Rendary [42]

	h, h_G			
	1/96	1/128	1/160	1/192
Ca	4.82e−5	3.44e−5	2.02e−5	1.43e−5
Ca [42]	6.28e−5	3.67e−5	2.67e−5	1.59e−5

Table 11
Zero gravity 2D column oscillation

	h, h_G		
	20/64	20/128	20/256
E_T Cartesian	4.04e−2	1.05e−2	0.37e−2
E_T prism	5.91e−2	1.65e−2	1.36e−2
E_T [55]	13.2e−2	6.1e−2	1.5e−2

Error in oscillation period as compared to linear theory [25].

Table 12
Zero gravity 2D column oscillation

	$\Delta t/\Delta t_{cap}$			
	1	1/2	1/4	1/8
E_T	4.26e-2	4.04e-2	3.90e-2	3.85e-2

Error in oscillation period on Cartesian grid $h = h_G = 20/64$ as compared to linear theory [25].

Table 13
Zero gravity 3D sphere oscillation

	h, h_G			
	20/64	20/96	20/128	20/160
E_T	8.50e−2	3.85e−2	2.08e−2	1.19e−2
Order	–	1.95	2.14	2.50

Error in oscillation period on Cartesian grid as compared to linear theory [25].

between two superposed immiscible fluids. The initial surface position inside a $[0, 2\pi] \times [0, 2\pi]$ box is given by a sinusoidal disturbance of wavelength $\lambda = 2\pi$ and amplitude $A_0 = 0.01\lambda$,

$$G(\boldsymbol{x}, t = 0) = y - y_0 + A_0 \cos(x - h_G/2), \tag{63}$$

with $y_0 = \pi$. Periodic boundary conditions are used in the x-direction and slip walls are imposed in the y-direction. The initial value solution for two fluids with equal kinematic viscosity ν and $\lambda = 2\pi$ can be written as [40]

$$A_{ex}(t) = \frac{4(1-4\beta)\nu^2}{8(1-4\beta)\nu^2 + \omega_0^2} A_0 \operatorname{erfc}\sqrt{\nu t} + \sum_{i=1}^{4} \frac{z_i}{Z_i} \left(\frac{\omega_0^2 A_0}{z_i^2 - \nu} \right) \exp[(z_i^2 - \nu)t] \operatorname{erfc}(z_i \sqrt{t}), \tag{64}$$

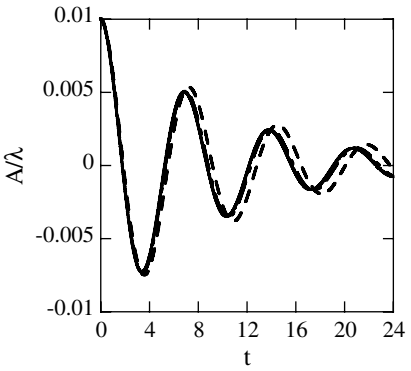
where z_i are the roots of

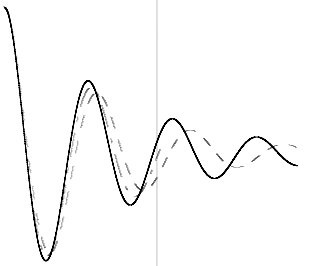
$$z^4 - 4\beta\sqrt{\nu} z^3 + 2(1-6\beta)\nu z^2 + 4(1-3\beta)\nu^{3/2} z + (1-4\beta)\nu^2 + \omega_0^2 = 0, \tag{65}$$

the dimensionless parameter β is given by $\beta = \rho_1\rho_2/(\rho_1 + \rho_2)^2$, the inviscid oscillation frequency is $\omega_0 = \sqrt{\frac{\sigma}{\rho_1+\rho_2}}$ and $Z_i = \prod_{\substack{j=1 \\ j \neq i}}^{4} (z_j - z_i)$.

We analyze two different cases for $\sigma = 2$, the first where both fluids have equal density, $\rho_1 = \rho_2 = 1$ and $\nu = 0.064720863$, and the second where the two fluids have a high density ratio, $\rho_1 = 1000$, $\rho_2 = 1$ and $\nu = 0.0064720863$. Results are obtained using a fixed time step size of $\Delta t = 0.003$ in the equal density case and $\Delta t = 0.06$ in the high density ratio case. In both cases we perform at least one re-initialization step per time step, i.e. we force the initial re-initialization trigger condition to be true, cp. Fig. 2.

Fig. 17 shows the temporal evolution of the non-dimensional disturbance amplitude A measured at $x = h_G/2$ under flow solver and G-grid refinement for both a Cartesian and a prism flow solver grid for the case of equal densities, whereas Fig. 18 shows the evolution of the corresponding non-dimensional error $E(t) = (A(t) - A_{ex}(t))/A_0$ and Table 14 summarizes its root mean square. The results show that for the





coarsest grids of $h = h_G = \lambda/16$, the oscillation frequency is not predicted well resulting in large errors in amplitude. However, refining both the flow solver and the *G*-grid results in excellent agreement with the theory. Comparing the amplitude errors to those reported in Gueyffier et al. [19] using a VoF method, the present results are superior on the Cartesian grid and similar on a prism grid of comparable h. Also, the reported divergence for wave amplitudes smaller than the grid size h was not observed here, indicating that the level set based interface tracking scheme handles sub-grid size disturbances to the interface geometry stably and more accurately. Comparing the results to those reported in Popinet and Zaleski [39] using a marker particle method, the present results on a Cartesian grid yield slightly worse results for coarse grids and almost identical results on the finest grid. The results reported by Gerlach et al. [17] using the PROST method show surprisingly high grid convergence rates and are thus at least one level of grid refinement better than the reported results.

Figs. 19 and 20 show the temporal evolution of the amplitude and the amplitude error for the high density ratio case, where as Table 15 summarizes the rms of the amplitude error. In the past, comparisons to the initial value theory have been limited to equal densities or small density ratios of the order of 10, the exception being Oevermann et al. [35] that studied a density ratio of 100. However, they report a large difference in frequency to the theoretical result that increases with increasing density ratio. Here, the results obtained with the RLSG method are in very good agreement with the theoretical predictions both on the Cartesian and the prism flow solver grid. Note that under grid refinement, the amplitude errors do not appear to converge to the exact theoretical solution, but rather appear to converge to a solution consistent with a slightly different frequency. This difference in frequency is however orders of magnitude smaller than that reported in Oevermann et al. [35]. Its interesting to point out that both marker particle [39] and VoF results [19] show similar slow convergence rates for a low density ratio of 10.

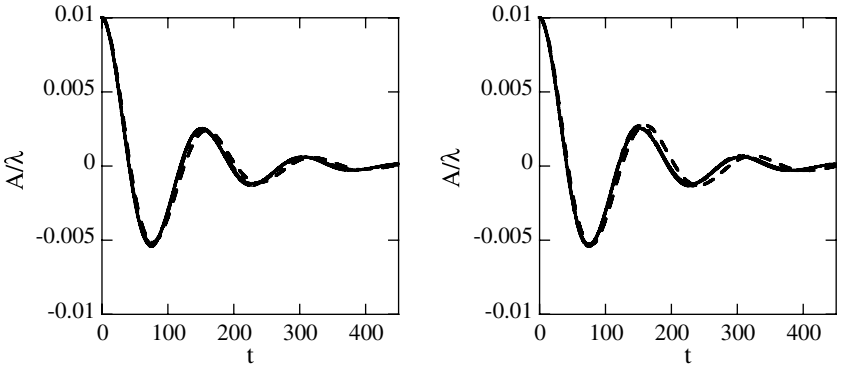


Fig. 19. Amplitude A of damped surface wave with $\rho_1/\rho_2 = 1000$; Cartesian grid (left) and prism grid (right) with $h = h_G = \lambda/16$ (dashed), $h = h_G = \lambda/32$ (dash-dot), $h = h_G = \lambda/64$ (dotted), $h = h_G = \lambda/128$ (solid).

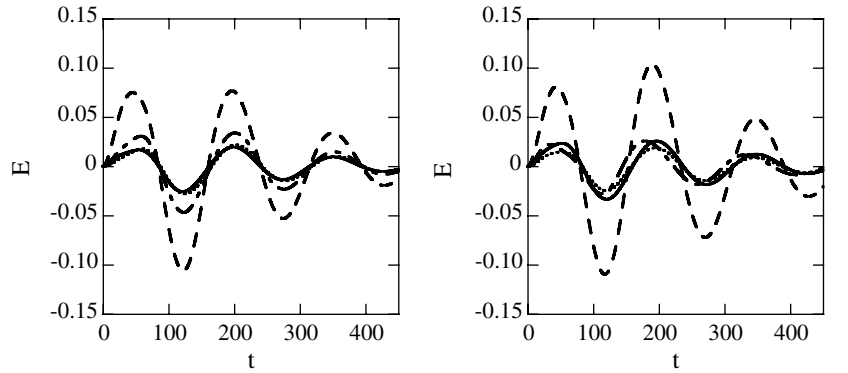


Fig. 20. Amplitude error E of damped surface wave with $\rho_1/\rho_2 = 1000$; Cartesian grid (left) and prism grid (right) with $h = h_G = \lambda/16$ (dashed), $h = h_G = \lambda/32$ (dash-dot), $h = h_G = \lambda/64$ (dotted), $h = h_G = \lambda/128$ (solid) and theory (thin line).

Table 15
Relative rms error E_{rms}/A_0 of damped surface wave amplitude for $\rho_1/\rho_2 = 1000$ and $t < 450$

λ/h, λ/h_G	Cartesian grid	Prism grid
16	0.0482	0.0564
32	0.0208	0.0141
64	0.0127	0.0113
128	0.0118	0.0157

4.7. Rayleigh–Taylor instability

To demonstrate the performance of the proposed method in a more complex flow, a Rayleigh–Taylor instability is computed. This is a common test problem performed by a variety of different methods [6,18,39,41]. A heavy fluid, $\rho_1 = 1.225$, $\mu_1 = 0.00313$, is placed above a light fluid, $\rho_2 = 0.1694$, $\mu_2 = 0.00313$, inside a domain of size 1×4. The interface between the two fluids is placed in the middle of the domain and is perturbed by a cosine wave of amplitude 0.05. The gravity constant is set to $g = 9.81$. We set the time step size constant to $\Delta t = 2.5 \cdot 10^{-4}$ and simulate up to $t = 0.9$. Fig. 21 shows the interface shape at different instances in time for a Cartesian flow solver grid of $h = 1/512$ and a G-grid of $h_G = 1/512$. As will be demonstrated below, this grid resolution ensures grid converged results and thus will be used as a reference solution in the following.

Figs. 22–24 show the interface shape for different resolutions of the Cartesian flow solver grid and G-grid, whereas Figs. 25–27 show the corresponding results for prism flow solver grids. Using the coarsest Cartesian flow solver grid of $h = 1/64$ presented in Fig. 22, one can already notice deviations from the reference solution at early times. While the stem and bubble shape is well captured, the fine scale geometry of the side arms is not well maintained. Up to $t = 0.8$, there appears almost no difference between the results using $h_G \geqslant 1/128$. This indicates that the deviations from the reference solution are due to errors in the flow representation and not due to errors in the interface tracking scheme. However, at $t = 0.9$, the very fine connecting bridge at the side arms can only be maintained by $h_G = 1/512$. Note that except for the difference in the details of the connecting bridge, the larger scale geometric features are consistent between different G-grid resolution with $h_G \geqslant 1/128$. Fig. 25 shows the corresponding results for a prism flow solver grid of $h = 1/64$. The obtained interface shapes are noticeable worse than in the Cartesian case showing deviations even in the lower stem and bubble shape. Again, good interface grid convergence is achieved for $h_G \geqslant 1/128$ up to $t = 0.8$, indicating that the observed deviations from the reference solution are due to insufficient flow solver resolution. Note that at $t = 0.9$, the $h_G = 1/512$ grid is not able to maintain the thin connecting bridge, since it is thinner than the reference solution.

At a Cartesian flow solver grid of $h = 1/128$ presented in Fig. 23, virtually no difference can be discerned between the reference solution and G-grid resolutions of $h_G \geqslant 1/128$ up to $t = 0.8$. At $t = 0.9$, however, the

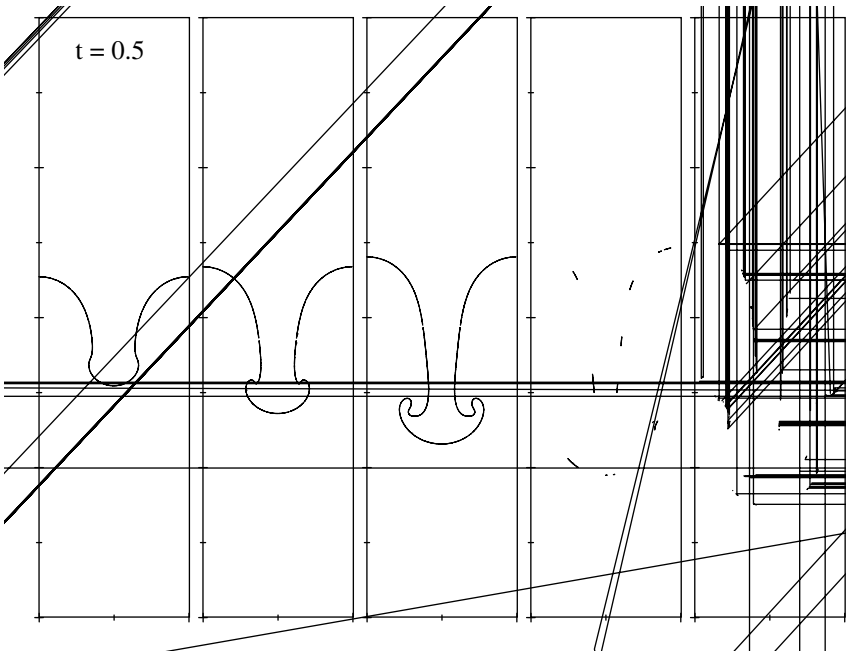


Fig. 21. Rayleigh–Taylor instability $h =$ $=$ interface shap

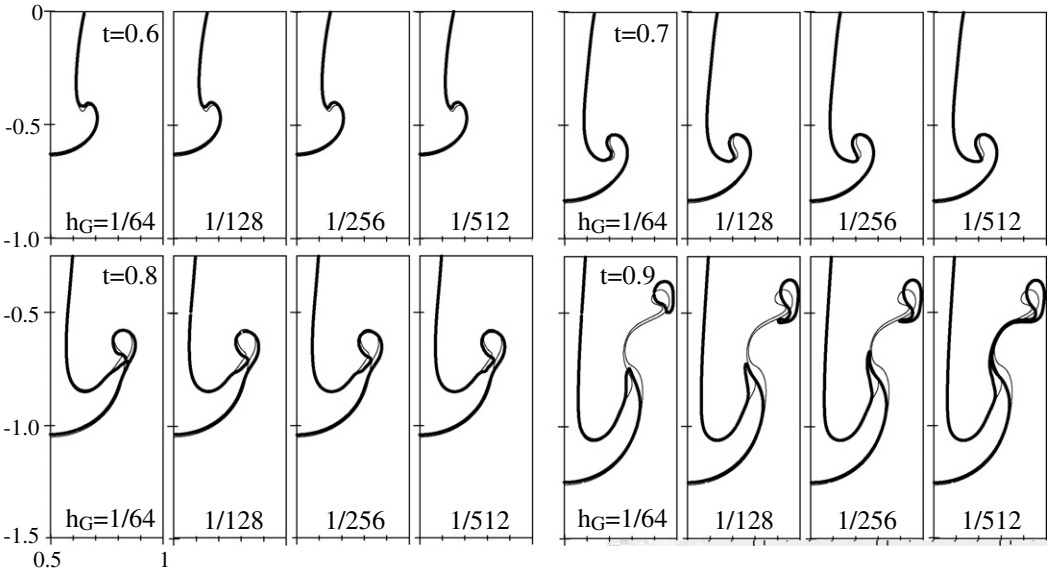
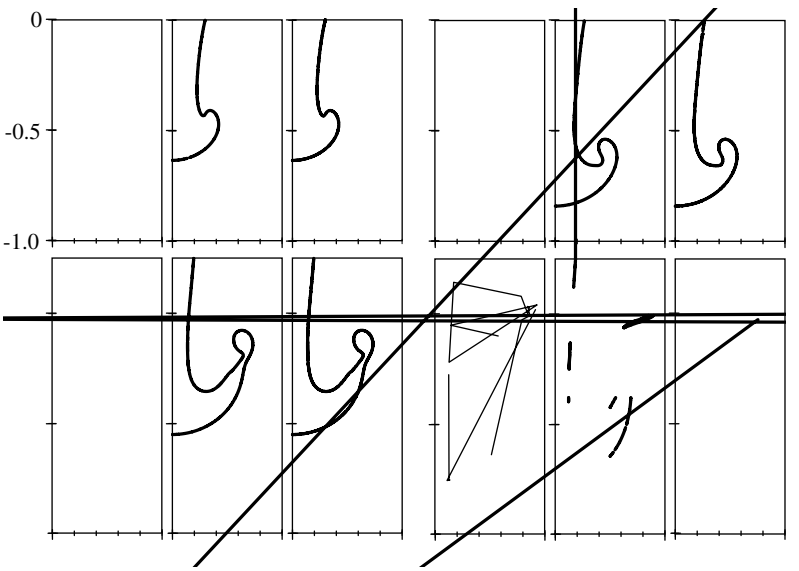


Fig. 22. Rayleigh–Taylor instability interface shapes under G-grid refinement $h_G = 1/64$, 1/128, 1/256 and 1/512 (left to right in each group) at $t = 0.6$, 0.7, 0.8 and 0.9 (top left to bottom right) and Cartesian flow solver grid $h = 1/64$. Thin line denotes reference solution.



thin connecting bridge is only supported by $h_G = 1/512$. Comparing the results at $t = 0.9$ of $h = 1/128$ to those of $h = 1/64$ (cf. Fig. 22), the finer grid flow solver results capture the shape of the interface significantly better. This indicates that the flowfield is well resolved by the $h = 1/128$ grid. This conclusion is even more evident in the prism flow solver case when comparing the $h = 1/128$ solution, Fig. 26, to the $h = 1/64$ solution, Fig. 25. At $h = 1/128$ both the stem shape and the bubble shape are well captured. Yet, although the interface shapes converge well for $h_G \geq 1/256$, there still exists a noticeable difference to the reference solution at the outside edges of the bubble, indicating that the employed prism grid might be insufficient to resolve the fine scale flow field there.

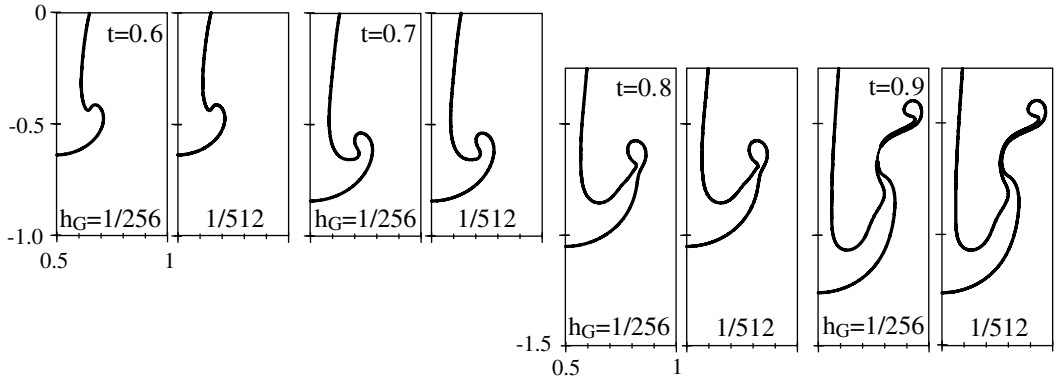
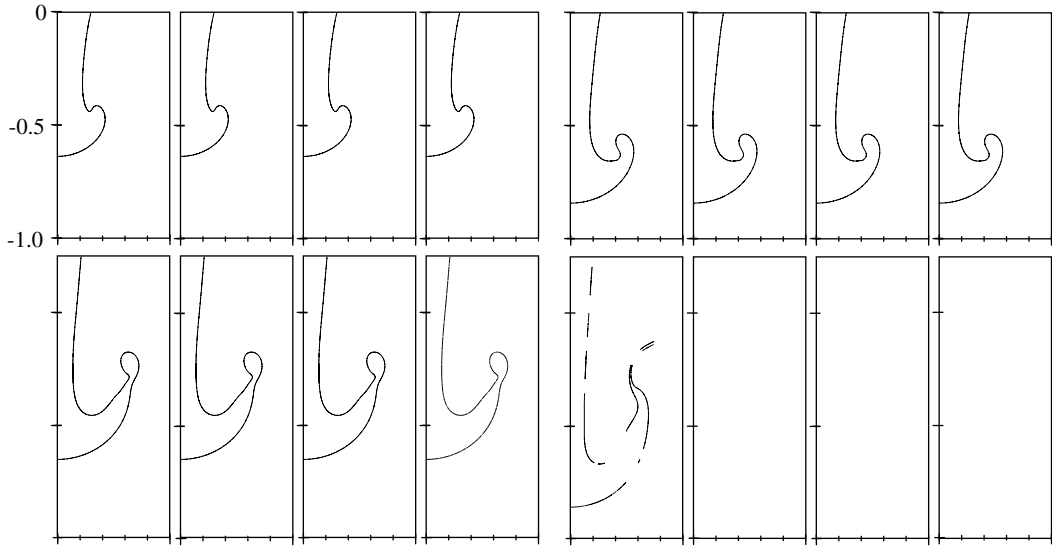


Fig. 24. Rayleigh–Taylor instability interface shapes under G-grid refinement $h_G = 1/256$ and $1/512$ (left to right in each group) at $t = 0.6$, 0.7, 0.8 and 0.9 (left to right) and Cartesian flow solver grid $h = 1/256$. Thin line denotes reference solution.



Refining the Cartesian flow solver grid further to $h = 1/256$, presented in Fig. 24, virtually no difference to the reference solution can be discerned, with the exception of $h_G = 1/256$ at $t = 0.9$, where again the complete fine connecting bridge is not supported by that G-grid resolution. Nonetheless, those parts of the bridge that can be maintained by the grid are in excellent agreement with the reference solution. A similar conclusion can be drawn for the prism flow solver grid with $h = 1/256$. Although not yet in full agreement with the reference solution, the interface shapes clearly converge to the reference solution under flow solver grid refinement.

To ascertain the volume/mass conservation properties of the method, Fig. 28 depicts the volume error E_V for the Cartesian flow solver solutions. Except for $h_G = 1/512$, all solutions show an increase in error at late times. This is due to the disappearance of the thin connecting bridge. Also, at constant flow solver grid size h and G-grid refinement, the volume errors converge to a non-zero value. Furthermore, the converged error decreases when increasing the flow solver grid resolution. This indicates the two sources of this error. First, $\boldsymbol{u}_{i_G}$ is based on interpolation of the flow solver cell center velocities $\boldsymbol{u}_{cv}^{n+1}$ which, in a collocated scheme, are not guaranteed to be divergence free. Second, the interpolation scheme itself can add additional divergence to the interpolated velocities $\boldsymbol{u}_{i_G}$. While the former inconsistency is hard to address in a collocated

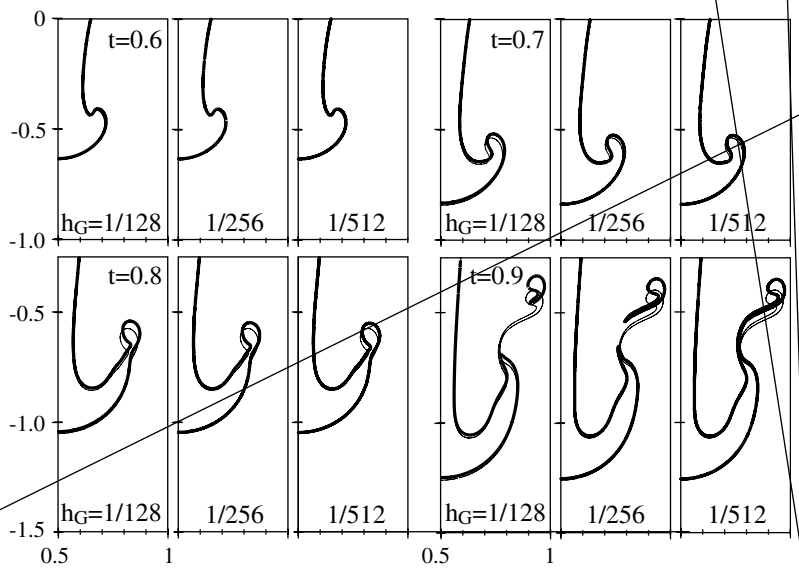
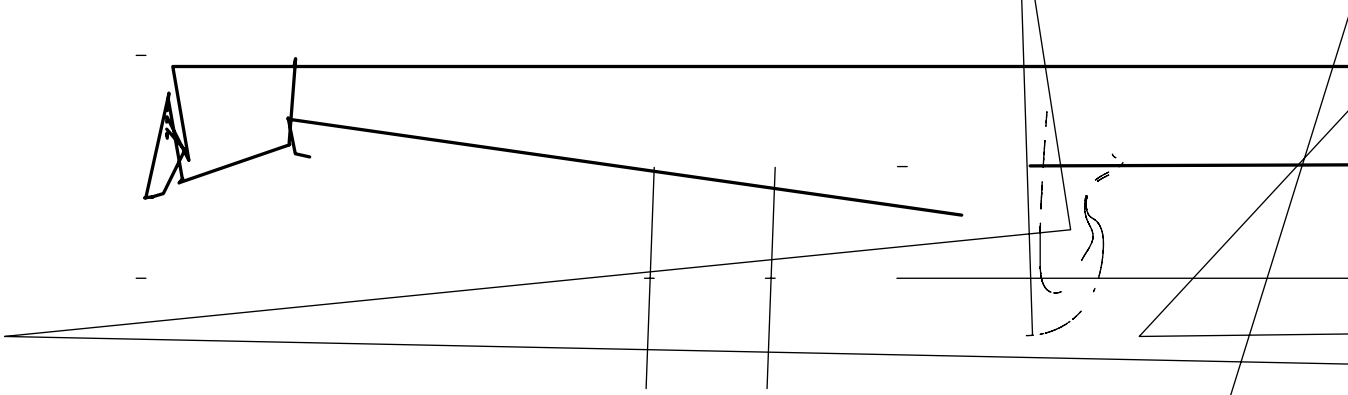
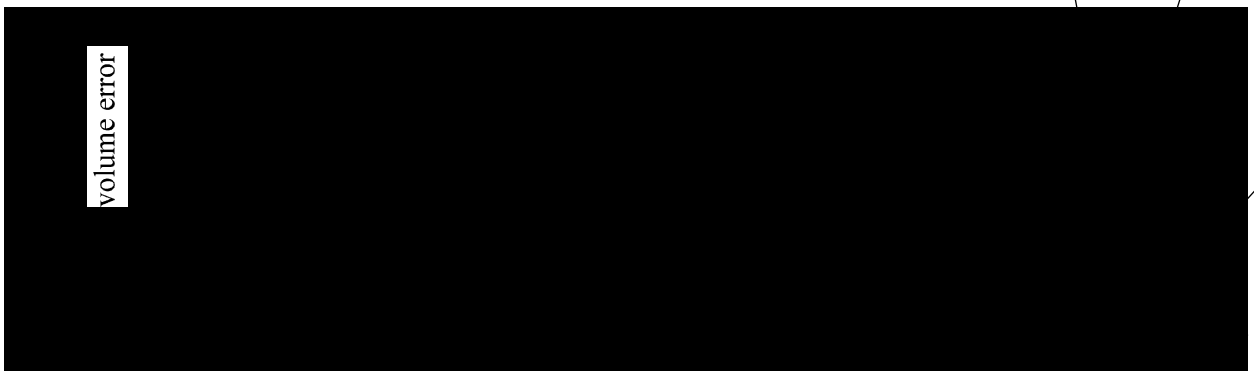


Fig. 26. Rayleigh–Taylor instability interface shapes under G-grid refinement $h_G = 1/128$, 1/256 and 1/512 (left to right in each group) at $t = 0.6$, 0.7, 0.8 and 0.9 (top left to bottom right) and prism flow solver grid $h = 1/128$. Thin line denotes reference solution.





unstructured flow solver, except by an additional projection-correction of the cell center velocities used for interpolation, the latter inconsistency can be addressed by improved, divergence preserving interpolation schemes. Nonetheless, the observed volume errors on fine G-grids are very small and well within acceptable limits.

5. Conclusion

A balanced force RLSG method has been presented for structured and unstructured flow solver grids. The interface location is tracked using a level set method solved on an auxiliary, high-resolution, equidistant Cartesian grid. A parallel dual-narrow band approach ensures that fine enough grids can be employed to control liquid volume conservation errors and to enable grid convergence studies with respect to interface tracking errors. The employed balanced force method ensures machine precision zero spurious currents on unstructured grids for arbitrary density ratios if the curvature can be evaluated exactly. Spurious current magnitude is directly related to errors in the evaluation of the interface curvature. To minimize spurious currents in actual applications, a robust second-order converging curvature evaluation scheme has been presented.

Acknowledgments

The author would like to thank Olivier Desjardins, Dokyun Kim, Guillaume Blanquart and Frank Ham

[19] D. Gueyffier, J. Li, A. Nadim, S. Scardovelli, S. Zaleski, Volume of Fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.
[20] M. Herrmann, A domain decomposition parallelization of the fast marching method, in: Annual Research Briefs-2003, Center for Turbulence Research, Stanford, CA, 2003, pp. 213–226.
[21] M. Herrmann, A Eulerian level set/vortex sheet method for two-phase interface dynamics, J. Comput. Phys. 203 (2) (2005) 539–571.
[22] B. Houston, M.B. Nielsen, C. Batty, O. Nilsson, K. Museth, Hierarchical RLE level set: a compact and versatile deformable surface representation, ACM Trans. Graph. 25 (1) (2006) 151–175.
[23] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 21 (6) (2000) 2126–2143.
[24] D. Kim, H. Choi, A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids, J. Comput. Phys. 162 (2) (2000) 411–428.
[25] H. Lamb, Hydrodynamics, Dover Publications, New York, 1945.
[26] F. Lekien, J. Marsden, Tricubic interpolation in three dimensions, Int. J. Numer. Methods Eng. 63 (2005) 455–471.
[27] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, Comput. Fluids 35 (10) (2006) 995–1010.
[28] P. Macklin, J. Lowengrub, An improved geometry-aware curvature discretization for level set methods: application to tumor growth, J. Comput. Phys. 215 (2) (2006) 392–401.
[29] K. Mahesh, G. Constantinescu, P. Moin, A numerical method for large-eddy simulation in complex geometries, J. Comput. Phys. 197 (1) (2004) 215–240.
[30] E. Marchandise, P. Geuzaine, N. Chevaugeon, J.-F. Remacle, A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics, J. Comput. Phys. (available online).
[31] E. Marchandise, J.-F. Remacle, N. Chevaugeona, A quadrature-free discontinuous Galerkin method for the level set equation, J. Comput. Phys. 212 (2006) 338–357.
[32] P. Marmottant, E. Villermaux, On spray formation, J. Fluid Mech. 498 (2004) 73–111.
[33] M.B. Nielsen, K. Museth, Dynamic tubular grid: an efficient data structure and algorithms for high resolution level sets, J. Sci. Comput. 26 (3) (2006) 261–299.
[34] R.R. Nourgaliev, T.G. Theofanous, High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set, J. Comput. Phys. 224 (2) (2007) 836–866.
[35] M. Oevermann, R. Klein, M. Berger, J. Goodman, A projection method for two-phase incompressible flow with surface tension and sharp interface resolution, Technical Report, Konrad-Zuse-Zentrum fuer Informationstechnik Berlin, Berlin, Germany, 2000.
[36] S. Osher, C.W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Num. Anal. 28 (4) (1991) 907–922.
[37] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[38] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[39] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.
[40] A. Prosperetti, Motion of two superposed viscous fluids, Phys. Fluids 24 (7) (1981) 1217–1223.
[41] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[42] Y. Renardy, M. Renardy, Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.
[43] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[44] J.A. Sethian, Level Set Methods and Fast Marching Methods, second ed., Cambridge University Press, Cambridge, UK, 1999.
[45] S. Shin, S.I. Abdel-Khalik, V. Daru, D. Juric, Accurate representation of surface tension using the level contour reconstruction method, J. Comput. Phys. 203 (2005) 493–516.
[46] C.W. Shu, Total-variation-diminishing time discretization, SIAM J. Sci. Stat. Comput. 9 (6) (1988) 1073–1084.
[47] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. 83 (1989) 32–78.
[48] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (1) (2003) 110–136.
[49] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comput. Phys. 148 (1999) 81–124.
[50] M. Sussman, M. Ohta, Improvements for calculating two-phase bubble and drop motion using an adaptive sharp interface method, Fluid Dyn. Mater. Proc. 3 (1) (2007) 21–36.
[51] M. Sussman, E.G. Puckett, A coupled level set and volume of fluid method for computing 3d and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2) (2000) 301–337.
[52] M. Sussman, P. Smereka, S. Osher, A level set method for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146.
[53] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2) (2007) 469–505.

[54] I.E. Sutherland, G.W. Hodgman, Reentrant polygon clipping, Commun. ACM 17 (1) (1974) 32–42.
[55] D.J. Torres, J.U. Brackbill, The point-set method: front-tracking without connectivity, J. Comput. Phys. 165 (2) (2000) 620–644.
[56] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[57] S.P. van der Pijl, A. Segal, C. Vuik, A mass-conserving level-set method for modelling of multi-phase flows, Int. J. Numer. Methods Fluids 47 (2005) 339–361.
[58] M.W. Williams, D.B. Kothe, E.G. Puckett, Convergence and accuracy of continuum surface tension models, in: W. Shyy, R. Narayanan (Eds.), Fluid Dynamics at Interfaces, Cambridge University Press, Cambridge, 1999, pp. 294–305.
[59] X. Yang, A.J. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, J. Comput. Phys. 217 (2) (2006) 364–394.
[60] Y.-N. Young, F.E. Ham, M. Herrmann, N. Mansour, Interaction between turbulent flow and free surfaces, in: P. Bradshaw (Ed.), Annual Research Briefs-2002, Center for Turbulence Research, Stanford, CA, 2002, pp. 301–312.
[61] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.
[62] S. Zaleski, Private communication, 2007.
[63] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing. II: application to two-and three-dimensional level-set simulations of multiphase flow, J. Comput. Phys. 208 (2) (2005) 626–650.