Simulating Interfacial Tension of a Falling Drop in a Moving Mesh Framework

Anja R. Paschedag [a,*], Blair Perot [b]

[a] *TU Berlin, Institute of Chemical Engineering, 10623 Berlin, Germany*

[b] *University of Massachusetts, MIE Department, ELAB 219, Amherst, MA 01003-2210*

Abstract

Processes in disperse multiphase systems are significantly determined by the interaction of momentum and mass transfer at the interface including particle deformation. A code for modeling such interactions for a single drop has been improved with special stress on implementation of interfacial tension. The influence of the surface tension on the momentum transfer is a function of the curvature. A new method to determine the curvature along the discretized interface is introduced and compared with two methods known from literature. The code operation could be stabilized and it could be shown, that an approach based on a spherical assumption for the interfacial shape gives significantly better results than an approach based on a polynomial assumption.

Key words: adaptive mesh, moving interface, finite volume, curvature, CFD

* Corresponding author. e-mail: anja.paschedag@tu-berlin.de

Notation

A_f	area of cell face, m^2
$\vec{e}$	unit vector in cell face perpendicular to edge, m
$\vec{F}_\sigma$	force caused by interfacial tension, N
$\vec{g}$	gravitational acceleration, m/s^2
$\mathfrak{H}$	curvature, $1/m$
$\mathbf{I}$	unit matrix, –
L_e	edge length, m
$\vec{m}$	vector tangential to interface and perpendicular to edge, m
$\vec{n}$	face normal vector, m
p	pressure, Pa
R	radius of the droplet, m
$\vec{r}$	vector from cell circumcenter to face circumcenter, m
$\vec{r}^{cg}$	vector from cell center of gravity to face center of gravity, m
t	time, s
$\vec{t}_e$	vector along edge, magnitude equal edge length, m
$\vec{v}$	velocity, m/s

Greek symbols

κ	correction factor in eq. (12), –
μ	dynamic viscosity, kg/ms
ν	kinematic viscosity, m^2/s
ρ	density, kg/m^3
σ	interfacial tension, N/m
τ	stress tensor, Pa

Indices

1	dispersed phase
2	continuous phase
c	cell
e	edge
f	cell face (part of interface)
i	interface
n	normal
t	tangential

1 Introduction

Two-phase systems with two fluid phases have a wide range of application in chemical engineering. Extraction processes, bubble columns and airlift loops are only some examples. For design and optimization of such apparatuses knowledge of the processes at the interface is indispensable. To study the interaction of momentum and mass transfer including particle deformation numerically is an important step to gain such knowledge.

The code uns3d is designed to simulate the flow of a two-phase system using an adaptive mesh technique. The model equations are solved continuously through both phases but with different physical parameters. Therefore, at the interface no boundary condition is required but the jump conditions have to be included in the discretized form of the governing equations at the right location.

The discretization is based on the unstructured mesh method with exact pressure projection. A detailed description can be found in [1], [2], [3].

2 Model

The governing equations for the flow field are the equations of continuity and momentum balance. As both phases are considered to be incompressible and of constant viscosity the Navier Stokes equations can be applied:

$$0 = \nabla \cdot \vec{v} \tag{1}$$

$$\rho \frac{\partial \vec{v}}{\partial t} = -\rho \vec{v} \cdot \nabla \vec{v} + \nabla \cdot (\mu \nabla \vec{v}) - \nabla p + \rho \vec{g} \tag{2}$$

Usual boundary conditions for walls, inlet, outlet, symmetry etc. can be applied at the outer boundaries. At the interface, a discontinuous change in the fluid properties and jump conditions determine the behavior of the flow.
For the systems considered, the two velocity fields are coupled by a non-slip condition:

$$\vec{v}_1 = \vec{v}_i = \vec{v}_2 \tag{3}$$

This equation also determines the velocity of the interface motion $\vec{v}_i$. For the moving mesh technique this velocity determines the velocity of the nodes at the interface which is the basis for the mesh adaptation.
The jump in the velocity gradients at the interface is related to differences in kinematic viscosity in both phases, the pressure jump over the interface and the force caused by interfacial tension. This can be formulated in equations for the shear stress [4]:

$$\begin{aligned} \tau_{nn,1} - p_1 &= \tau_{nn,2} - p_2 + 2\mathfrak{H}\sigma \\ \vec{\tau}_{n\mathbf{t},1} &= \vec{\tau}_{n\mathbf{t},2} + \vec{t} \cdot \nabla_S \sigma \end{aligned} \tag{4}$$

where the gradient along the interface is

$$\nabla_S = (\mathbf{I} - \vec{n}\vec{n}) \cdot \nabla \tag{5}$$

and the curvature $\mathfrak{H}$ is defined by

$$\mathfrak{H} = -\frac{1}{2}\nabla_S \cdot \vec{n} \tag{6}$$

and can be expressed as a function of two curvature radii R_{C1} and R_{C2}

$$\mathfrak{H} = -\frac{1}{R_C}\mathfrak{H} = -\frac{1}{2}\left(\frac{1}{R_{C1}} + \frac{1}{R_{C2}}\right) \tag{7}$$

The second approach which is often used assumes that the interface can locally be approximated by an elliptic shape.

3 Implementation

In the moving mesh approach the two fluids are distinguished only by their physical parameters. The Navier Stokes equations are solved for the whole domain and not separately for the phases. Therefore, the jump conditions can not be used as boundary conditions as in [5],[6] but have to be integrated into the discretized Navier Stokes equations at the interface. The consideration of the non-slip condition does not require additional action as the continuity of

the velocity field is a basic assumption for the discretization also within the phases.

The jump in the velocity gradients can be implemented straightforward, because the discretization procedure applied requires the expression of the fluxes at each cell face. If a cell face is considered which is a part of the interface (see Fig. 1) the computation of the fluxes must be based on the jump condition. Discretization of (4) gives:

normal

$$\frac{\mu_1}{r_1}(v_{nc1} - v_{nf}) - p_{c1} = \frac{\mu_2}{r_2}(v_{nf} - v_{nc2}) - p_{c2} + 2\mathfrak{H}\sigma \tag{8}$$

tangential

$$\frac{\mu_1}{r_1}(\vec{v}_{tc1} - \vec{v}_{tf}) = \frac{\mu_2}{r_2}(\vec{v}_{tf} - \vec{v}_{tc2}) + \vec{t} \cdot \nabla_S \sigma \tag{9}$$

where the velocity gradients are discretized between the cell centers and the center of the face in the interface while the pressure difference is determined between the two cells. Combining the continuous (4) and discretized (8) formulation different expression for the momentum flux across the interface are gained for both phases:

$$\begin{aligned}
\mu_1 \left(\left. \frac{\partial \vec{v}}{\partial n} \right|_{f1} \right) &= \frac{\vec{v}_{c2} - \vec{v}_{c1}}{\alpha_1} + \frac{p_{c2} - p_{c1} + 2\mathfrak{H}\sigma + \vec{t} \cdot \nabla_S \sigma}{\alpha_1} \quad \text{with} \quad \alpha_1 = 1 + \frac{\mu_2 r_1}{\mu_1 r_2} \\
\mu_2 \left(\left. \frac{\partial \vec{v}}{\partial n} \right|_{f2} \right) &= \frac{\vec{v}_{c2} - \vec{v}_{c1}}{\alpha_2} - \frac{p_{c2} - p_{c1} + 2\mathfrak{H}\sigma + \vec{t} \cdot \nabla_S \sigma}{\alpha_2} \quad \text{with} \quad \alpha_2 = 1 + \frac{\mu_1 r_2}{\mu_2 r_1}
\end{aligned} \tag{10}$$

The weighting factor for the velocity term is identical to the weighted harmonic mean approach for the viscosity published by Quan and Schmidt [7]. The influence of interfacial tension is not only implemented for the liquid cells

as in former versions but distributed. The normal part is implemented directly at the interface while the tangential part is split between the neighbor cells in both phases based on the viscosity ratio and the rato of cell sizes. This is especially significant for cases with a viscosity ratio close to one. For large differences in viscosities the former version is gained as limiting case.

It can be shown that in the discrete equations the normal velocity component in both cells is equal. This means, that the normal component of the interface tension stress and the pressure gradient cancel out and can be therefore removed from the equation. This is consistent with the code structure in the sense, that the transformation of discretized equations with primitive variables into equations for the stream function avoids a solution for the pressure. The remaining tangential component is relevant for the simulation of variable interfacial tension like in cases with changing temperature or concentrations at the interface.

In the total stress balance at the interface the normal interfacial tension stress is considered. The interfacial tension terms in eq. (4) can be expressed in the discretized form to be forces acting at the cell faces which are part of the interface. They can be either computed from the curvature at the midpoint of the face or from the forces acting at the edges. In this paper one procedure is investigated which computes the curvature based on a polynomial fit and two procedures which compute forces at the edges based on a spherical assumption for the local shape of the interface. Both approaches are of second order.The first approach has been implemented by Dai [2]. It fits the polynomial either to six or to twelve nodes. In the way it is implemented the forces are not guaranteed to be in local equilibrium.

The second approach computes the force at the faces based on the forces acting on the edges in a direction normal to the edge and normal to the curvature

radius direction. For a two-dimensional system this is shown in Fig. 2. They are proportional to the length of the edge and the interfacial tension at the edge. This determination is relevant in cases where the interfacial tension is not constant. The discretized force can be written as:

$$\vec{F}_{\sigma,f} = \sum_e \vec{F}_{\sigma,e} = \sum_e \frac{\kappa \vec{m}_e}{|\vec{m}_e|} L_e \sigma \tag{11}$$

where $\vec{m}_e$ is the binormal vector at edge e. The correction factor κ is defined in a way that $\sum_e (\kappa \vec{m}_e / |\vec{m}_e|) L_e = 2\mathfrak{H}$.

A major difficulty is the determination of the binormal vector. The two procedures assuming spherical shape are presented here vary in the way the binormal vector is computed. They differ mainly in the number of neighboring faces considered. The method used by Jasak and Tukovič [8] computes in a first step the normals at the interfacial mesh nodes from the normals of all interfacial faces surrounding this nodes. Then, the binormal on an edge is computed from average of the two nodal normals and the edge direction.

A new method shall be introduced here which considers the sphere defined by the four nodes which are corners of the two interfacial faces adjacent to the edge. The basic derivation is done for two dimensions as shown in Fig.3 where the circle is defined by the edge connecting the two faces (reduced to a point) and the two remaining nodes of the faces (reduced to lines). The direction of the binormal vector at the edge is normal to the edge and tangential to the circle. An non-scaled vector $\vec{m}_e$ satisfying this requirements can be computed as any linear combination of the two vectors $\vec{m}_1$ and $\vec{m}_2$ which are binormal to the edge and the radius at the edge midpoint themselves. One possibility for such a linear combination is:

$$\vec{m}_e = \frac{|\vec{r}_1|}{|\vec{r}_2|}\vec{m}_2 - \frac{|\vec{r}_2|}{|\vec{r}_1|}\vec{m}_1 = |\vec{r}_2|\frac{\vec{r}_1}{|\vec{r}_1|} + |\vec{r}_1|\frac{\vec{r}_2}{|\vec{r}_2|} \tag{12}$$

The vectors $\vec{r}$ point from the circumcenter of the face to the midpoint of the edge. To avoid problems if the circumcenter of a face is at the edge or outside the face the unit vectors $\vec{e} = \vec{r}/|\vec{r}|$ pointing in that direction can be computed in any way, e.g. based on the vector $\vec{r}^{cg}$ pointing from the center of gravity of the face to the edge center and the vector $\vec{t}_e$ along the edge:

$$\vec{e} = \frac{\vec{r}^{cg} - ((\vec{t}_e \cdot \vec{r}^{cg})\vec{t}_e)/(\vec{t}_e \cdot \vec{t}_e)}{|\vec{r}^{cg} - ((\vec{t}_e \cdot \vec{r}^{cg})\vec{t}_e)/(\vec{t}_e \cdot \vec{t}_e)|} \tag{13}$$

If this vector is used the computation is only correct if for the $|\vec{r}_i|$ in eq. (12) not the absolute value of $\vec{r}_i$ but the length with the sign relative to $\vec{e}_i$ is used:

$$\vec{m}_e = |\vec{r}_2|\vec{e}_1 + |\vec{r}_1|\vec{e}_2 \tag{14}$$

As all information in the code is stored with respect to the centers of gravity the circumcenters of the faces have to be computed to determine $\vec{r}_i$. (reference to paper/thesis from Blair?) The transformation of the three-dimensional system requires a correction factor, because the midpoint of the edge is not at the sphere. A correction is possible either by an iterative procedure or an algebraic expression.

This method uses information of less faces than the method of Jasak and Tukovič and therefore has less smoothing. This is favorable in the case of a physically not perfectly spherical interface but more vulnerable for numerical errors.

Finally, for both edge based methods the interfacial tension force is split into

a normal and a tangential part. At the edges, the tangential part of the force related to the face normal is usually much larger than the normal part but the tangential parts cancel nearly out if they are summed over the faces. To reduce the resulting numerical error the vectors at the edges are split and the normal and tangential parts are added separately for each face. For ease of further use the tangential part is kept as a force while for the normal part the stress is computed. The complexity of the computation of the interfacial tension term makes implicit handling difficult. Therefore, in the actual version of the code the term is implemented fully explicit. This requires time steps small enough to resolve the transport of capillary waves caused by the interfacial tension.

4 Results

The three procedures introduced will be compared for the case of a liquid drop in a gas. A basic case for evaluation is a resting drop without gravity field. In this situation the drop should maintain its shape and no significant velocities should be computed. Numerical errors can cause discrepancies e.g. because of imperfectness of the mesh but for the steady state these errors should be damped. The second test case is the falling drop.

The geometry used is a drop with a diameter of 200 μm within a box of 800 μm edge length. The outer boundary conditions state zero velocity at the beginning. During the simulations they are modified in a way that he domain is moved with the velocity of the centroid of the drop to ensure that the drop can not fall out of the domain without need of a very large domain.

To study the influence of the physical properties on the numerical stability density and viscosity for a water drop in air have been applied. The interfacial

tension had to be reduced by a factor of 100 to obtain numerically stable results. The values are listed in table 1.

ρ_{air}	μ_{air}	ρ_{H_2O}	μ_{H_2O}	σ
1.2 kg/m^3	$1.82 \cdot 10^{-5}$ Pa s	998 kg/m^3	10^{-3} Pa s	$0.0728 \cdot 10^{-2}$ N/m

Table 1

Physical properties

For comparison with other systems Table 2 gives the determining dimensionless numbers. As a reference velocity the final velocity of the simulation has been used. This is not the stationary fall velocity which reduces the applicability of the computed Reynolds and Weber number. For the resting drop velocity dependent numbers can not be used.

Re	We	Oh	μ_{H_2O}/μ_{air}	ρ_{H_2O}/ρ_{air}
1	$2 \cdot 10^{-3}$	0.083	55	832

Table 2

Dimensionless numbers

The smoothing algorithms in the code tend to unify the mesh resolution. Therefore, a mesh with a nearly uniform resolution has been generated which contains about 2000 cells in the drop and 2200 cells in the surrounding. This resolution provides stable results for a reasonable computing time. Accuracy studies for variable mesh resolution have not been performed.

Because of partly explicit discretization the time step is restricted by the Courant number and a similar criterion for capillary waves at the interface to values between 10^{-7} s to 10^{-5} s. Therefore, the physical times simulated are relatively short and the results should be understood as a feasibility test for the method rather than a physically relevant simulation.

The first test for judging the methods was done with the curvature computed in the first time step. In this step the shape of the drop did not change yet, so the theoretical curvature can be used for comparison. The error for the polynomial method is about 5% for the 12 node approximation and 2% for the 6 node approximation. In contrary, for the spherical method the error is in the range of the machine accuracy. This result is not surprising, because an approximation assuming the real shape should give much better results than a one for a different shape, even if it has the same order of accuracy.

For the resting drop in a system without gravity the development of the velocity magnitude at the interface and averaged over the whole drop was checked over 10,000 time steps which equals a physical time of about 0.03 s. After a short initial instability with velocity magnitudes of about 10^{-9} m/s the system stabilizes for the Tucovič method and the new one to velocity magnitudes of about 10^{-13} m/s which can be considered to be a well acceptable error. Simulations with the polynomial approach give velocities in the range of 10^{-4} m/s which is significant considering the fact that only a very short real time is simulated. At least, the shape of the drop is stable in all simulations which is a significant improvement gained by the way the surface tension terms are implemented.

Simulations of the falling drop have been carried out for 12,000 time steps equal to a physical time of about 0.008 s. Over this time the drop accelerates due to gravity, a constant sink velocity is not reached yet. To validate the simulations for such a short time, where no reliable measurements are available, a simulation with the commercial CFD-code Star-CD and with the solution of an ordinary differential equation for the sink velocity of a rigid sphere [9] have been used. Both assume an undeformed spherical particle which is completely acceptable for the configuration studied. The implementation in Star-CD has

been validated for falling and rising drops in different liquid/liquid systems [6]. The assumption of a rigid sphere in the analytical equation is for a small water drop in air nearly satisfied. Fig. 4 shows, that all three curves fall nearly together. At the end, where slight differences can be seen, the simulation with uns3d (newly introduced computation of curvature used) is between the other two other solutions and can therefore be considered to be satisfying.
The acceleration requires a permanent reduction of the time step to fulfill the Courant-Friedrichs-Lewy criterion. Therefore, even if the drop has not reached its terminal velocity during the simulations a continuation of the computation for a physically reasonable time is not possible at the moment.

4.1 Conclusion

The implementation of the interfacial tension terms into a finite volume code with moving adaptive mesh has been improved. The implementation is now done in a way that physical stability can be reflected by the simulations and that variable surface tensions, which cause a significant tangential force along the interface can be considered. This provides a basis for simulations sensitive to local changes in the interfacial tension, like in systems with Marangoni convection.
Simulations have been validated for a falling water drop in air and coincide with simulations using the commercial code Star-CD as well as an analytical limiting equation.
The curvature of the interface can be described by an polynomial or a spherical approximation. For the drop the spherical approach gives much better results and a higher stability of the solution because it is an exact fit. As the

spherical shape is the stable one for liquid particles under surface tension on which do not act other forces while the polynomial shape is quite arbitrary the spherical approach can be advised in general for such kind of simulations. The partly explicit discretization used in the code requires quite small time steps. Therefore, physically relevant times which allow a comparison of simulation results with experiments e.g. for the sink velocity can not be gained at the moment.

References

[1] B. Perot, R. Nallapati, A Moving Unstructured Staggered Mesh Method for the Simulation of Incompressible Free-Surface Flows, J. Comp. Physics 184(1) (2003) 192-214.

[2] M. Dai, H. Wang, J.B. Perot, D.P. Schmidt, A Numerical Method for Interface Tracking, Atomization and Sprays 12(5-6) (2002) 721-735.

[3] Ch. Wang, F. Giraldo, B. Perot, Analysis of an Exact Fractional Step Method, J. Comp. Physics 179 (2002) 1-17.

[4] Deen W.M., Analysis of Transport Phenomena, Oxford University Press, New York (1998).

[5] A. Paschedag, W.H. Piarah, M. Kraume, Sensitivity study of mass transfer at a single droplet, Int. J. Heat Mass Transfer 48 (2005) 3402-3410.

[6] M. Wegener, A.R. Paschedag, M. Kraume, The effect of Marangoni convection on unsteady drop mass transfer, in: Proceedings of the 11^{th} Workshop on Two-phase Flow Predictions, Merseburg, 5.-8.4.2005 (2005).

[7] Sh. Quan, D.P. Schmidt, A moving mesh interface tracking method for 3D incompressible two-phase flows, to be published.

[8] Jasak, Tukovic: Computation of the curvature.

[9] M. Kraume, Transportvorgnge in der Verfahrenstechnik: Grundlagen und apparative Umsetzung, Springer, Berlin (2004).

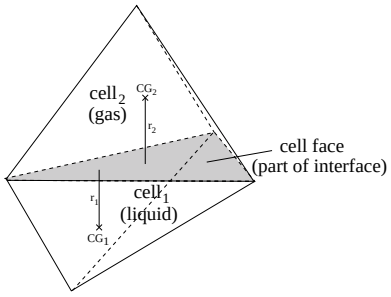

Fig. 1. Geometry of cells at the interface

Fig. 2. Forces at edges and averaging (2d)

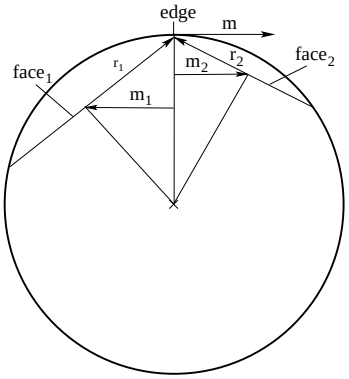

Fig. 3. Geometry for calculation of binormal vector

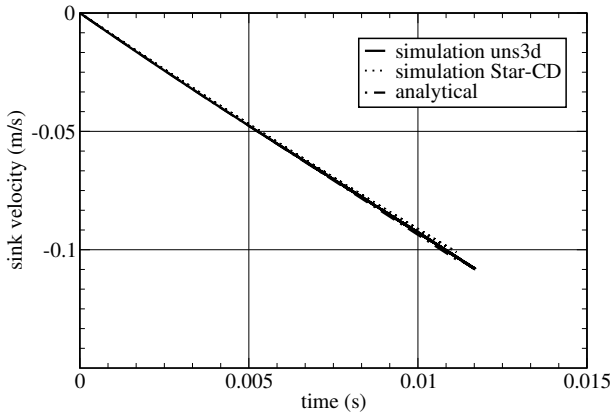

Fig. 4. Comparison of simulated particle velocity with other validated approaches