Technische Universität Wien

University of L'Aquila

Joint Master's Programme - MathMods

Mathematical Modelling in Engineering: Theory, Numerics, Applications

Diplom-Ingenieur	**Laurea Magistrale**
Technical Mathematics	**Ingegneria Matematica**
Technische Universität Wien (TUW)	University of L'Aquila (UAQ)

Master's Thesis

Multi-phase Interface Area Calculation Using Iso-Alpha Method

Supervisor

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Michael Harasek

Co-advisor

Projekt ass. Dr.techn. Bahram Haddadi Sisakht

Candidate

Jobin Raju

Student ID (UAQ): 261412
Student ID (TUW): 11848043

Academic Year 2019/2020

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Declaration of Authorship

I, Jobin RAJU, declare that this thesis titled, "Multi-phase Interface Area Calculation Using Iso-Alpha Method" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

TECHNISCHE UNIVERSITÄT WIEN

Abstract

Research Division: Thermal Process Engineering and
Simulation-Computational Fluid Dynamics
Institute of Chemical, Environmental and Bioscience Engineering

Master of Science

Multi-phase Interface Area Calculation Using Iso-Alpha Method

by Jobin RAJU

One observes absorption phenomena when fluids of different nature come in contact with each other. During absorption, mass is transferred across the fluid-fluid interface in both directions via diffusion of species. Computational Fluid Dynamics (CFD) techniques can be suitably exploited to evaluate the extent of mass transfer and its visualization. The key parameters that influence a specific specie absorption rate are the area of interface and a driving force. Existing area calculation algorithms are either based on approximation techniques using the magnitude of gradient alpha (volume fraction) or geometrical methods. In the former method, the magnitude of the gradient alpha (volume fraction) can be suitably manipulated to approximate the area of the interface. In the latter method, the mesh cells are split into two to form an interface separating the different phases and then calculating the area of the polygonal interface. Current research focuses on accurate evaluation of the interface area between the two phases using a geometrical method-'Iso-Alpha' and its application on mass transfer simulations. The case of a rising bubble with mass transfer is considered for the current study. The total interface area of the bubble is calculated in every time-step using both methods for comparison. The magnitude of the interface area is validated against the area obtained from the post-processing utility - paraView. Subsequently, the species mass transfer rate is also compared using both area calculation methods. Mass transfer during bubbly flows is commonly observed in chemical and biological processes such as oxidation, fermentation, etc. Hence, a quantitative estimation of transferred mass is an important aspect in industrial-scale packed column bubble-liquid reactors and related chemical engineering applications.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Acknowledgements

I would first like to thank my thesis co-supervisor Dr. Bahram Haddadi of the Institute of Chemical, Environmental and Bioscience Engineering. He introduced me to the world of numerical simulations using OpenFOAM and helped me explore the different dimensions of the programming way to CFD simulations. He allowed me to be part of his own research work and gave me the opportunity to present the research at the Austrian OpenFOAM conference. The door to Dr. Bahram was always open whenever I ran into trouble with the coding part of my thesis.

I would also like to extend my gratitude to my program coordinator Prof. Anton Arnold of the Institute of Analysis and Scientific Computing. I really appreciate the effort he put into coordinating with the different Departments at the University for my smooth studies.

I would also like to thank my family and friends for supporting me throughout my journey and motivating me to complete my thesis and coursework at Technische Universität Wien.

Finally, I want to thank the OpenFOAM community for supporting each other via forums and for keeping this amazing tool open source.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Figures

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Abbreviations

CFD	Computational Fluid Dynamics
EE	Eulerian-Eulerian
LE	Lagrangian-Eulerian
VoF	Volume of Fluid
MAC	Marker and Cell
SLIC	Simple Line Interface Construction
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
PISO	Pressure Implicit with Splitting of Operator
HRIC	High Resolution Interface Capturing
MULES	Multidimensional Universal Limiter for Explicit Solution
CICSAM	Compressive Interface Capturing Scheme for Arbitrary meshes
RDF	Reconstructed Distance Function
PLIC	Piecewise Linear Interface Construction

Key Symbols and Units

F_{body}	Body forces (N)
F_{surf}	Surface forces (N)
F_{coll}	Force due to collision between particles (N)
F_{lift}	Lift forces (N)
F_{vm}	Virtual mass forces (N)
m_p	Mass of particle (kg)
ρ_p	Density of the discrete particle (kg/m^3)
g	Acceleration due to gravity (m/s^2)
V_P	Volume of the discrete particle (m^3)
n_p	Number of particles per unit volume (m^{-3})
α_f	Volume fraction of the fluid phase
α_p	Volume fraction of the discrete phase
F_{int}	Interphase hydrodynamic forces (N)
F_D	Drag force (N)
Re	Reynolds Number
d_p	particle diameter (m)
α_1	phase 1 volume fraction
α_2	phase 2 volume fraction
τ	stress-strain tensor (N/m^2)
K	momentum exchange coefficient (kg/s)
τ_1	Relaxation time (s^{-1})
C_D	Drag coefficient
f	Drag function (N)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

ψ	RDF (m)
n	number of particles
$\hat{n}$	normal vector
C	Species concentration (mol/m^3)
u	Average flow velocity (m/s)
u_p	Particle velocity (m/s)
u_{12}	Interphase velocity (m/s)

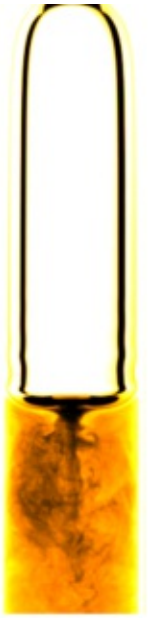
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 1

Introduction

1.1 Multi-phase Mass Transfer

Fluid flows observed in nature generally consist of more than one phase. The additional phases could be solid particles, gases, or even fluids with different material properties. The phases are likely to interact with each other during the flow. As a result of the interaction, there are a lot of possible outcomes depending on the nature of the interacting phases. A fluid flow with solid particles as the additional phase results in the transportation of particles along with the flow. For example, soil deposition along river banks. Solids can also absorb species from the fluid phase through the surface via adsorption. The interacting phenomenon becomes slightly complicated when gases are involved due to their ability to react chemically, get absorbed into the liquid phases, or adsorption onto solid phases. In the aforementioned situations, a common observation in the nature of the interaction is the transfer of mass from one phase to the other.



(A) Micro reactor[1]

(B) Bubble column reactor[2]

The current research focuses on the key aspects of such fluid-gaseous interactions where a mass transfer is involved. A typical mass transfer phenomenon is observed in aerated flows where the components of the gaseous

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

phase get partially absorbed and desorbed into the medium through the multi-phase interface. This methodology can be used to filter out certain elements from a gas mixture using a suitable liquid medium. The key parameters in such mass transfer processes are the area of interface and the driving force for interfacial species exchange between the phases. Since the mass transfer phenomena happen at the interface and it constantly varies its position relative to the domain, this makes it difficult to conduct experimental studies on the rate of mass transfer, flow patterns, and specie mass transport along with the flow. In the current research, a simplified case of a liquid-gas interaction with mass transfer in a rectangular column is conducted using Computational Fluid Dynamics (CFD) techniques using the numerical package OpenFOAM.

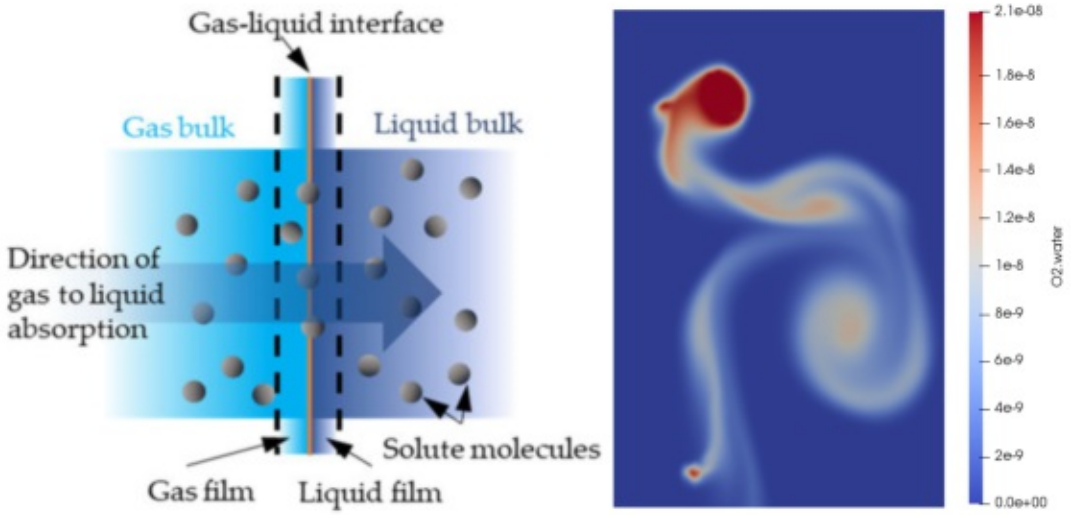


FIGURE 1.2: Mass transfer through an interface (left), simulation-rising bubble (right)[3]

1.2 Literature Review

Numerical simulation of multi-phase flows took a new turn with the introduction of the Volume of Fluid (VoF) method by Hirt and Nichols[4]. This opened up a new domain for scientific research: identification and reconstruction of the interface. Earlier methods made use of Marker and Cell[5] which demanded high memory requirements. Hirt and Nichols originally used the SOLA-VOF method. But later averaging methods replaced the MAC method. For a short period, the Simple Line Interface Construction (SLIC)[6] method served to approximate the interface between the two media. The SLIC method constructs simple vertical and horizontal partitions in the interface cell so that it conserves the averaged volume fraction while solving the VoF equations. The SLIC method is mass preserving in nature. This method

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

was later substituted by more accurate Piecewise Linear Interface Calculation (PLIC)[7] which involved the construction of a line of surface that preserves the normal direction of the curvature developed at the interface. The original PLIC method was later expanded from 2D to 3D polyhedral mesh cells[8]. Cummins[9] suggested improved interface reconstruction methods developed using the modified PLIC method. The 21st century saw lots of hybrid methods[10][11] such as isoAlpha method, isoRDF, etc based on geometrical interface reconstruction.

The advancements in the interface reconstruction methodologies assisted the research in the domain of interfacial mass transfer simulations[12]. Measurement of mass transferred posed a difficult problem to tackle. Correlation studies[13] were used initially for quantifying the mass transfer. This was followed by particle tracking methods making use of Lagrangian methods. One such method is the Smooth Particle Hydrodynamics (SPH) method developed by Tartakovsky et al[14]. Interface capturing methods were later introduced because of their success in calculating mass transfer on a fixed euler grid[15]. A different approach for quantifying mass transfer via species concentration was introduced by Haroun et al[16]. He proposed the Continuum Species Transfer model using VOF formulation[17, 18]. This approach successfully captured the species concentration transport across the interface.

Earlier attempts were made to model mass transfer models using the magnitude of gradient alpha to approximated the interface[19–21]. Due to poor local reconstruction of the interface, it had poor applicability. Later attempts were made to make use of geometrical methods such as PLIC[7, 22] for mass transfer. Gim et al. describe a modified PLIC like algorithm for CFD simulations involving mass transfer[23]. Further investigation into the available research on geometrical area based mass transfer numerical showed a need for further study and simulation-based experimentation in this domain.

1.3 Revisiting Mass Transfer Simulations

Research on mass transfer simulation is constantly evolving. Adapting new methodologies for more accurate results is finally the aim of every researcher. When it comes to mass transfer, the interface between the different phases is one of such hot points. Capturing the accurate physics at the interface is of paramount importance. As mentioned before, the calculation of the interface area between the phases requires special attention. Commonly used interface area calculation methods (gradient alpha methods) for mass transfer simulations make use of approximation techniques for calculation of the interface area. These measures are often inaccurate and lead to errors while estimating the amount of mass transferred between the phases at specific time frames. This calls for alternative techniques for the estimation of the interface area.

The choice of estimation techniques depends on the accuracy requirement or nature of specific mass transfer simulations. When accuracy of the highest order is required, researchers normally go for geometrical techniques for interface area approximation and this comes at a higher computational cost.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Instead of using a linear relationship with the available parameters, the geometrical techniques make use of interface reconstruction and then manually calculating the area of the interface using simple area calculation formulas.

Interface reconstruction is a complex procedure that is under active development even now. Current research makes use of the interface reconstruction algorithm: isoAlpha method originally proposed by Dr. Johan Roenby[10]. Though his original work point towards the development of solution procedure for advection equations and their numerical implementation, the same principle is attempted in the current research after suitable changes. Since the work is open-source, it is permitted to make changes and adapt to the specific needs of the current thesis.

1.4 Some Insights on the Present Research

The primary focus of the present research is the implementation of a geometrical interface calculation algorithm for simulations involving mass transfer. The results of the simulation are compared and analyzed for evaluating the suitability of geometrical methods for interface area calculation. The interface area between the phases is evaluated at every time step and compared with values obtained using traditional area calculation methods.

The results of the area calculation methods are compared against the values obtained using the post-processing utility paraFoam. The rate of mass transfer corresponding to both methods is also subjected to comparison. Taking the computational power requirement in mind, a simplified 2D version of a real-life rising bubble case is adopted and simulated in OpenFOAM.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 2

Multiphase Mass Transfer

We describe a flow as a multi-phase if two or more phases coexist in the same flow domain. There are three physical states of matter- solid, liquid, and gas- under normal circumstances in general. But from a modeling point of view, it is advantageous to classify the phases depending on the way they respond to inertia and interaction with the flow. Depending on the nature of the phases involved, we have different mathematical models for each scenario. Some examples of multi-phase systems are gas bubbles in a liquid, liquid droplets in gas, and solid particles in gas or liquid.

It is possible that during the interaction of multiple phases, mass is transferred from one phase to the other. In nature, it is common among gases to get absorbed into the solution they come in contact with. It is also common practice in chemical industries, where chemicals in the aqueous phase are treated with gaseous reactants in a fractional column. The gas is passed through the aqueous solution in a stirred environment wherein the mass transfer takes place between the phases. The below chart shows the available multiphase solution methodologies.

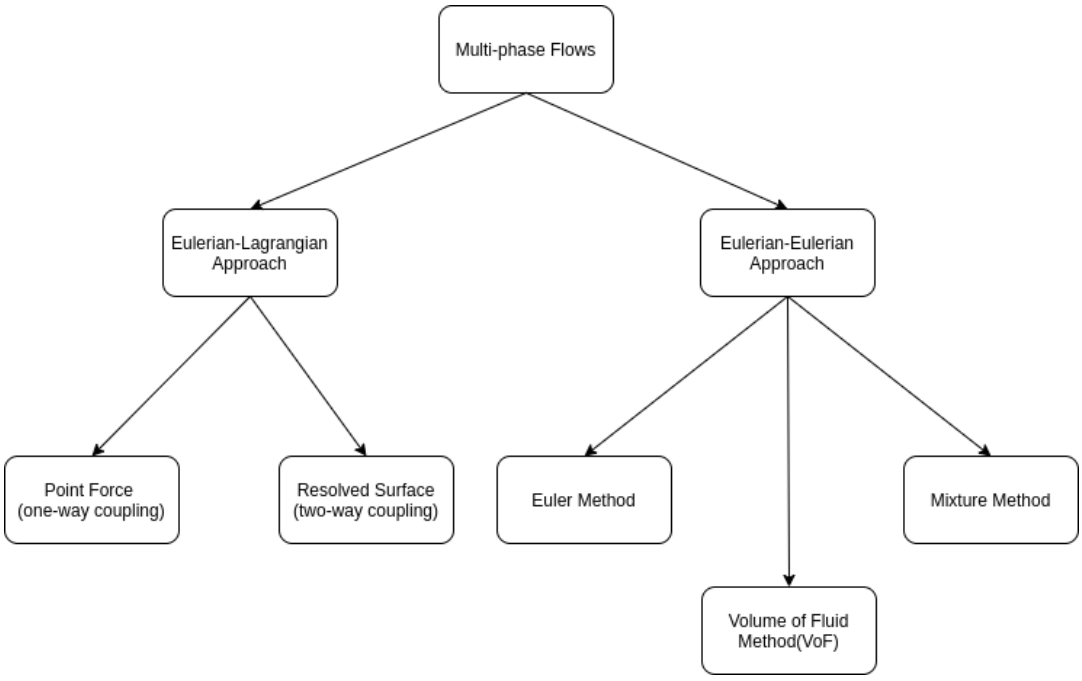


FIGURE 2.1: Multi-phase Flow Solution Methodologies

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.1 Lagrangian-Eulerian Approach

The Lagrangian-Eulerian model is usually employed for flows involving particulate matter in moving fluids. Equations of continuity and momentum are solved for the continuous Eulerian phase. The dispersed phase is solved using particle tracking. We use the flow field to track a large number of particles via one-way or two-way coupling of momentum, mass, and energy exchange between the phases. Properties such as the velocity and temperature of a Lagrangian particle are updated along the trajectory.

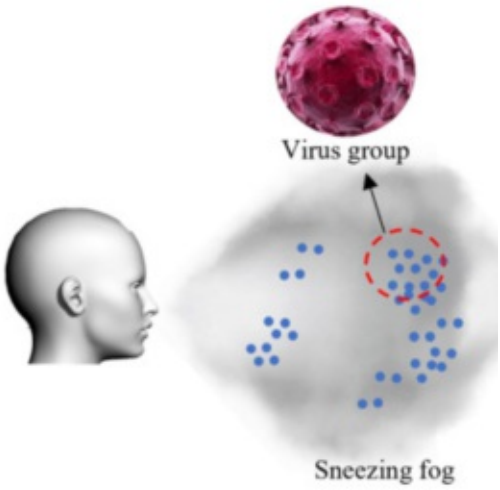


FIGURE 2.2: Euler-Lagrangian flow example in real life[24]

2.1.1 Lagrangian approaches: Point force and Resolved surface treatment

Depending on the type of coupling and relative size of the particle with respect to the grid, we categorize the lagrangian approaches into two: point force and resolved surface treatment[25].

For the point-force approach, the fluid or particle is described at a single point that moves at its own independent velocity. Hence this approach is also called Discrete Element Approach. The particle trajectory can be traced using an ODE, while the continuous phase is solved on a fixed eulerian grid.

In the resolved surface approach, the detailed flow around each particle is solved to a high resolution at the same time, treating the particles as discrete elements. The flow solution is numerically integrated over the surface.

ODE formulation for the particle movement due to the combined effect of external forces reads as shown below:

$$m_p \frac{dv}{dt} = F_{body} + F_{surf} + F_{coll}$$

Here the body force, F_{body} directly depends on the mass of the particle, F_{surf} represents the fluid dynamic forces and F_{coll} represents the forces due

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

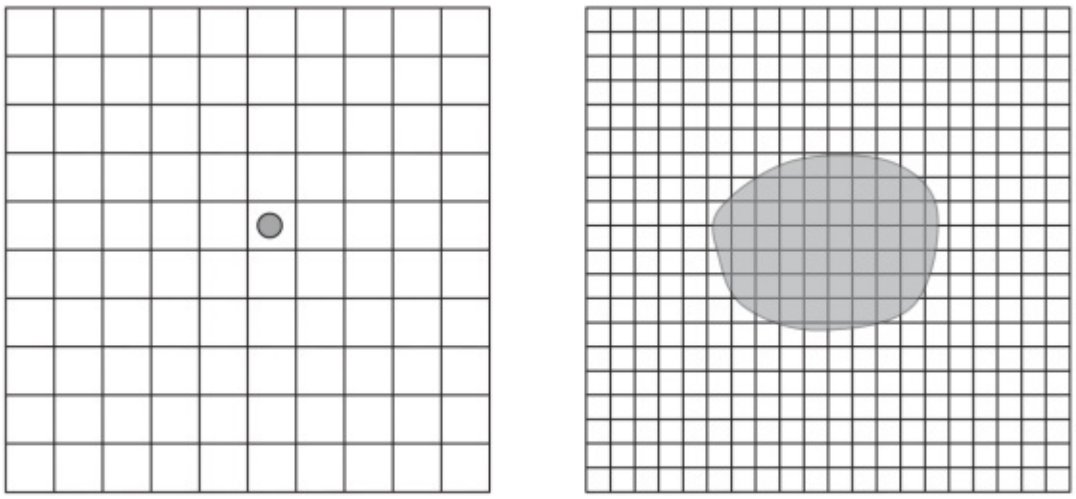


FIGURE 2.3: Comparison of particle size w.r.t grid resolution.
(left)point force (d<Δx) and
(right)resolved surface ($\Delta x << d$)

to particle-particle or particle-wall collisions. Depending on the treatment of F_{surf}, we can formulate the particle equation for the point force approach and resolved surface approach.

$$F_{body} = gm_p = g\rho_p V_p \tag{2.1}$$

where V_p is the particle volume and ρ_p is the volume-averaged density of the particle.

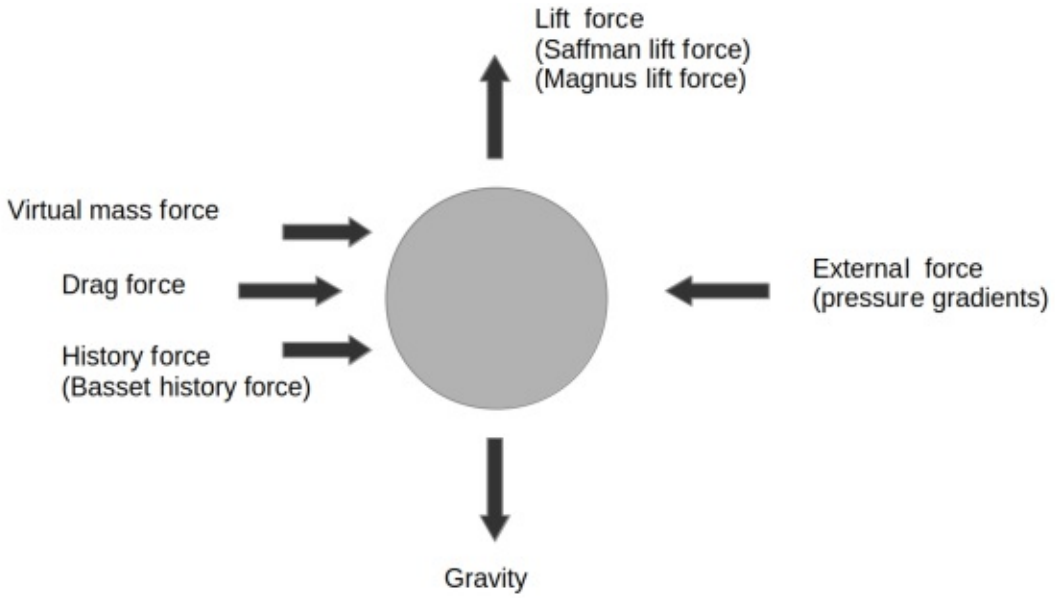


FIGURE 2.4: Forces acting on the discrete phase

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.1.2 Eulerian equations for a fixed grid

We have separate treatment of the momentum equations for the point-force and resolved-surface treatment of the discrete phase.
Continuous phase momentum equation (point-force treatment):

$$\rho\frac{\partial\alpha_f u_i}{\partial t}+\rho\frac{\partial\alpha_f u_i u_j}{\partial x_j}=\alpha_f\rho g-\alpha_f\frac{\partial p}{\partial x_i}+\alpha_f\mu\frac{\partial^2 u_i}{\partial x_j^2}-n_p F_{int,i} \tag{2.2}$$

where n_p is the number of particles per unit volume, α_f is the volume fraction of the fluid phase and $F_{int,i}$ is the interphase hydrodynamic force acting on the particles. The equation is applied to the entire domain.

Continuous phase momentum equation (resolved-surface treatment):

$$\rho\frac{\partial u_i}{\partial t}+\rho\frac{\partial u_i u_j}{\partial x_j}=\rho g-\frac{\partial p}{\partial x_i}+\mu\frac{\partial^2 u_i}{\partial x_j^2} \tag{2.3}$$

The above equation is solved only for those mesh cells excluding the particle cell volume.

Finally the continuity equation:

$$\frac{\partial\rho}{\partial t}+\nabla.(\rho u)=0 \tag{2.4}$$

2.1.3 Equations of Motion for Particles - Point Force approach

For point force approach F_{surf} is a function of surface averaged forces. In general, a surface-averaged force is a linear combination of specific forces, such as drag $\mathbf{F_D}$, lift **L**, added mass **A**, stress gradient **S**, history force **H**, etc.

$$F_{surf}=\sum F_{surf}=F_D+L+A+S+H+...$$

Balancing the forces in a Lagrangian frame of reference lets us predict the trajectory of the discrete phase particle. The Lagrangian-Eulerian model reads:

$$\frac{d\vec{u_p}}{dt}=F_D(\vec{u}-\vec{u_p})+\frac{\vec{g}(\rho_p-\rho)}{\rho_p}+\vec{F} \tag{2.5}$$

where $\vec{F}$ is an additional force term due to external factors such as centrifugal force. $\vec{F_D}(\vec{u}-\vec{u_p})$ is the drag force per unit particle mass, F_D is the inverse relaxation time defined as:

$$F_D=\frac{18\mu C_D Re}{\rho_p d_p^2 24} \tag{2.6}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Here, $\vec{u}$ is the fluid phase (phase 1) velocity, $\vec{u_p}$ is the particle velocity, C_D is the drag coefficient, μ is the molecular viscosity of the fluid, ρ is the fluid density, ρ_p is the density of the particle and d_p is the particle diameter. Re is the relative Reynolds number and it is defined as

$$Re = \frac{\rho d_p |\vec{u_p} - \vec{u}|}{\mu} \tag{2.7}$$

For very heavy particles ($\rho_p >> \rho_f$), the interphase force is often simplified to include only the particle drag $\mathbf{F_D}$. For very light particles with little or no collisions, the interphase forces can be expressed only using stress gradient $\mathbf{S}$. The stress gradient is defined below as:

$$S = \rho V_p D\vec{u}_p/Dt - g\rho V_p \tag{2.8}$$

where, V_p is volume of the particle. Hence the EL model reads:

$$\frac{d\vec{u_p}}{dt} = \rho V_p D\vec{u}_p/Dt - g\rho V_p + F \tag{2.9}$$

2.1.4 Equations of Motion for Particles - Resolved Surface approach

The resolved-surface approach allows for the details of the fluid pressure and shear stress to be integrated over the particle surface.

$$m_p \frac{dv}{dt} = F_{body} + F_{surf} + F_{coll} \tag{2.10}$$

where

$$F_{surf} = \int [-p + \mu_f(\partial u_i/\partial x_j + \partial u_j/\partial x_i)] n \cdot dA_p \tag{2.11}$$

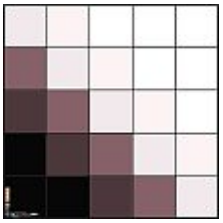
and F_{body} as defined in Equation 2.1. m_p is mass of the discrete particle, A_p is the surface area of the discrete particle and n is the unit normal from the surface. The effect of collision forces are insignificant compared to other forces and hence ignored during modeling purposes.

2.2 Eulerian-Eulerian Approach

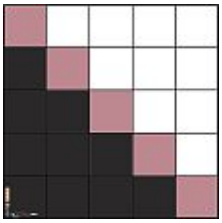
When the number of particles increases, the Lagrangian formulation becomes computationally intensive. The Eulerian formulation comes in handy in such situations since it is independent of the number of particles. Instead of solving separate equations for each particle, a set of momentum equations is solved for the discrete phase collectively after identifying the phases using an additional volume fraction parameter. In the EE approach, the different

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

phases are treated as an interpenetrating continuum. The equations of continuity and momentum are solved on a fixed grid. Depending on the nature of the interaction between the phases at the interface, the EE approach can be further subdivided into mixed and separated fluid approaches. The two main EE models are the Volume of Fluid model (mixed approach) and Eulerian model (separated fluid)[26].



(A) Dispersed - Continuous



(B) Continuous - Continuous

Dispersed-Continuous phase interactions can take any value of volume-fraction between 0 and 1. Whereas Continuous-Continuous phase interactions are restricted to a volume-fraction of either 0 or 1 (except in the interface region)

2.2.1 Euler-Euler (Two fluid) Model

The Eulerian model is the most sophisticated of multiphase models[26]. This model solves the continuity and momentum equations for all the phases. In addition to this, the coupling between the phases is achieved using different momentum transfer mechanisms. The model is also called the two-fluid model since it assumes both phases to be fluids regardless of whether it is a solid dispersed phase or gaseous phase. This model is appropriate for modeling fluidized beds, risers, pneumatic lines and particle-laden flows in which phases mix or separate[26][27]. In general, the EE model works well with the primary phase as a fluid and secondary phase which mixes well with the primary phase and the flow domain has volume fraction $>$ .10[28][29]. The EE model has limited capability to predict the flow pattern in such flows and hence VoF models are used whenever the interfaces between the different phases need to tracked especially in free-surface flows, slug flows, etc.

There exist different interphase exchange coefficient models depending on the nature of phases in the flow domain. Appropriate drag laws can be chosen for different processes. Several kinetic theory based formulas are also available for the granular stress in the viscous regime and frictional viscosity based formulation for the plastic regime stresses. The accuracy of the model depends on the precision of the closure models since the number of equations involved is high.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Conservation Equations

Continuity equation for phase 1:

$$\frac{\partial}{\partial t}(\alpha_1\rho_1) + \nabla.(\alpha_1\rho_1\vec{v_1}) = (\dot{m}_{12} - \dot{m}_{21}) + S_1 \tag{2.12}$$

Phases denoted by subscripts $(1,2)$, density by ρ_1, ρ_2, volume fraction by α_1, α_2, velocity v_1, v_2 and mass transfer between phases $\dot{m}_{12}$ is the mass transferred from phase 1 to phase 2. S is a user defined source term (zero in most cases). Also $\alpha_1 + \alpha_2 = 1$

Momentum Equation for phase 1:

$$\underbrace{\frac{\partial}{\partial t}(\alpha_1\rho_1\vec{v_1})}_{\text{local accln.}} + \underbrace{\nabla.(\alpha_1\rho_1\vec{v_1}\vec{v_1})}_{\text{convective accln.}} = \underbrace{-\alpha_1\nabla p}_{\text{pressure grad.}} + \underbrace{\nabla.\overline{\tau_1}}_{\text{viscous forces}} + \underbrace{\alpha_1\rho_1\vec{g}}_{\text{body force}} + \underbrace{K_{21}(\vec{v_2} - \vec{v_1})}_{\text{drag force}} + \underbrace{F}_{\text{ext. forces}} + \underbrace{(\dot{m}_{12}\vec{v_{12}} - \dot{m}_{21}\vec{v_{21}})}_{\text{mass-transfer momentum}} \tag{2.13}$$

The same set of equations can be similarly written for the second phase.

τ is the stress-strain tensor with the expression:

$$\overline{\tau_1} = \alpha_1\mu_1(\nabla\vec{v_1} + \nabla\vec{v_1}^T) + \alpha_1(\lambda_1 - \frac{2}{3}\mu_1)\nabla.\vec{v_1}\bar{I} \tag{2.14}$$

where μ and λ are the shear and bulk viscocity respectively. For incompressible flows, the second term in the RHS becomes zero.

$\vec{v_{12}}$ is the interphase velocity. If $\dot{m}_{12} > 0$, it implies mass from phase 1 is being transferred to phase 2 and hence $\vec{v_{12}} = \vec{v_1}$. If $\dot{m}_{12} < 0$, mass is transferred from phase 2 to phase 1 which implies $\vec{v_{12}} = \vec{v_2}$. We can similarly define $\vec{v_{21}}$ based on the sign of $\dot{m}$.

F refers to the external forces acting on the system such as lift force, external body force, virtual mass force, turbulent dispersion force(in case of turbulent flows), interaction force between phases, etc [25].

$$F = \vec{F_{ext}} + \vec{F_{lift}} + \vec{F_{vm}} + ... \tag{2.15}$$

The drag force in general has the form, $F_{drag} = \frac{1}{2}\rho A C_D(v-u)^2$, where C_D is the drag coefficient. Depending on the material properties of the phases, several closure models for drag force are available.
A comparison of the drag force expression with the drag force term in the momentum equation gives us:

$$K_{21} = \frac{1}{2}\rho_2 A C_D(v_2 - v_1) \tag{2.16}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

K_{21} is the momentum exchange coefficient. Many different K models are available for multiphase flows depending on the application.
K is defined as a function of $K(\rho, f, A, \tau)$ where the parameters are density, drag function, interface area, and particulate relaxation time respectively.

In a two-phase flow, the phase which is dominating the flow domain is the primary phase. The other phase is the secondary phase. In a fluid-fluid flow, the secondary domain is mostly droplets or bubbles. The momentum exchange coefficient for such flow domain has a general form:

$$K_{12} = \frac{\rho_1 f d_1 A_i}{6\tau_1} \tag{2.17}$$

Physically, the particulate relaxation time is the time taken by a particle to reach $(1 - 1/e)v_{flow}$ and is defined as

$$\tau_1 = \frac{\rho_1 d_1^2}{18\mu_2} \tag{2.18}$$

where d_1 is the bubble diameter, ρ_1 is bubble density and μ_2 is the viscosity of the bulk.

2.2.1.1 Popular Interface Area Models

The interface area is important for the exchange and evaluation of mass, momentum and energy transfer between the interacting phases. The ratio of surface area to volume is used to approximate the algebraic interfacial area (A_p). ie. A_p refers to the interface area per unit volume. For a spherical bubble or droplet:

$$A_p = \frac{\pi d^2}{\frac{1}{6}\pi d^3} = \frac{6}{d} \tag{2.19}$$

where d is the diameter of the bubble/droplet.

Particle Model
The particle model approximates the interface area concentration for a dispersed phase as:

$$A_i = \alpha A_p = \frac{6\alpha}{d} \tag{2.20}$$

where A_p is defined in equation 2.19 and α is the volume fraction of the discrete phase.

Symmetric Model
The symmetric model makes use of the volume fraction of the primary phase

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to ensure that the interface area is calculated only on the interface of the phases. ie. The interface area concentration becomes zero as α tends to 1.

$$A_i = \frac{6\alpha(1-\alpha)}{d} \tag{2.21}$$

where α is the volume fraction of the dispersed phase.

2.2.1.2 Popular drag models

The drag function f depends on the drag coefficient which in turn depends on the Reynolds number (Re). Some of the drag models for the EE model is described below.

Schiller and Neumann Model
For the Schiller and Neumann model[26], we define the drag function f as:

$$f = \frac{C_D Re}{24} \tag{2.22}$$

where, $C_D = \begin{cases} 24(1+0.15Re^{0.687})/Re & Re \leq 1000 \\ 0.44 & Re \geq 1000 \end{cases}$

Here Re is the relative Reynolds number which can be calculated using:

$$Re = \frac{\rho_2 |\vec{v}_1 - \vec{v}_2| d_1}{\mu_2} \tag{2.23}$$

This model is the default model in popular commercial solvers.

Morsi and Alexander Model
For the Morsi and Alexander model[30], the drag function, f is defined as:

$$f = \frac{C_D Re}{24} \tag{2.24}$$

where, $C_D = a_1 + \left(\frac{a_2}{Re}\right) + \left(\frac{a_3}{Re^2}\right)$
and the constants a_1, a_2, a_3 are defined in table 2.1

This model is found to be less stable compared to other models even though it is the most complete model since it adjusts the function values over a range of Reynolds numbers.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

a_1, a_2, a_3	Range of Re
0, 24, 0	0<Re<0.1
3.69, 22.73, 0.0903	0.1<Re<1
1.222, 29.1667, -3.8889	1<Re<10
0.3644, 98.33, -2788	100<Re<1000
0.357, 148.62, -47500	1000<Re<5000
0.46, -490.546, 578700	5000<Re<10000
0.5191, -1662.5, 5416700	Re$\geq$ 10000

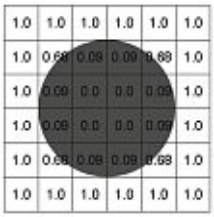
TABLE 2.1: Values of constants in Morsi and Alexander model

2.2.2 Volume of Fluid (VoF) Model

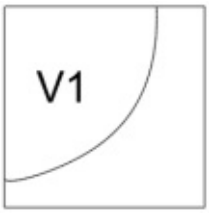
The VoF model is a simplified version of the full Eulerian model for continuous-continuous phase interactions. In this model, a surface tracking technique is applied on a fixed Eulerian grid. Here, we are interested in the position of the interface. In the VoF model, each cell holds only one value (averaged) for the simulation variables such as velocity, temperature, pressure, etc. A single set of continuity and momentum equations are solved for the different phases throughout the domain. Hence this model is less memory intensive compared to other models[31] when benchmarked on the same grid. At the same time, in practical applications, the VoF model requires a much finer computational grid compared to the EE model.

The interface between the different phases is captured by introducing a parameter α. It is defined as the fraction of fluid in a mesh cell.

$$\alpha = \frac{V_1}{V} \tag{2.25}$$



(A) Volume fraction of a Circular patch in a grid



(B) Immersed volume V1

Here, values in the grid cells correspond to the volume fraction. V is the total volume of the cell and V_1 is the immersed volume of each individual interface cell. The value of α is 1 if the cell has only phase 1 and 0 otherwise. Those cells with α value between 0 and 1 correspond to interface cells. The normal direction of the interface can be calculated by picking those cells whose value of alpha varies rapidly. Hence, evaluating the gradient of α gives the direction of propagation of the interface. Knowing the position of the interface from the initial condition, we need to find the time evolution of the interface as the flow progresses. Since the interface is being transported

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

along with the flow, we use a transport equation to advect the interface. The transport of a scalar quantity F in 2D is described as:

$$\frac{\partial F}{\partial t} + u\frac{\partial F}{\partial x} + v\frac{\partial F}{\partial y} = 0 \tag{2.26}$$

Since VoF approach proposes a single set of parameters in each grid cell, we use averaged variables. We define the global parameters for the Eulerian grid as follows:

$$\bar{v} = \frac{1}{V}\int_V v dV = \alpha\bar{v}_1 + (1-\alpha)\bar{v}_2 = \bar{v}_1 + \bar{v}_2 \tag{2.27}$$

$$\bar{p} = \frac{1}{V}\int_p p dV = \alpha\bar{p}_1 + (1-\alpha)\bar{p}_2 = \bar{p}_1 + \bar{p}_2 \tag{2.28}$$

$$\rho = \alpha\rho_1 + (1-\alpha)\rho_2 \tag{2.29}$$

$$\mu = \alpha\mu_1 + (1-\alpha)\mu_2 \tag{2.30}$$

For a scalar quantity a_i and a vectorial quantity b_i, both defined in phase i, the following relations hold when averaging over the volume of the cell[32]:

$$\overline{\nabla a_i} = \nabla\bar{a}_i + \frac{1}{V}\int_{A_{ij}} a_i n_{ij} dA, \tag{2.31}$$

$$\overline{\nabla . b_i} = \nabla . \bar{b}_i + \frac{1}{V}\int_{A_{ij}} b_i n_{ij} dA, \tag{2.32}$$

$$\overline{\frac{\partial a_i}{\partial t}} = \frac{\partial \overline{a_i}}{\partial t} - \frac{1}{V}\int_{A_{ij}} a_i n_{ij} . w dA \tag{2.33}$$

where n_{ij} is the normal from phase i to the other. These equations enable us to map model equations onto the mesh.

Equation for mass conservation

Due to averaging of parameters, we can conveniently assume that the mass of liquid in a mesh cell is directly proportional to the fluid volume. ie. $M(t) = \rho_l V_l$. This allows us to write:

$$\overline{\frac{\partial \rho_1}{\partial t} + \nabla.(\rho_1 v_1)} = \bar{0} = 0 \tag{2.34}$$

$$\overline{\frac{\partial \rho_1}{\partial t} + \nabla.(\rho_1 v_1)} = \overline{\frac{\partial \rho_1}{\partial t}} + \overline{\nabla.(\rho_1 v_1)} \tag{2.35}$$

$$= \frac{\partial \bar{\rho_1}}{\partial t} - \frac{1}{V}\int_{A_{12}} \rho_1 n_{12} . w dV + \nabla.(\overline{\rho_1 v_1}) + \frac{1}{V}\int_{A_{12}} \rho_1 n_{12} v_1 dV \tag{2.36}$$

$$= \rho_1\frac{\partial \alpha}{\partial t} + \nabla.(\rho_1\bar{v_1}) + \frac{1}{V}\int_{A_{12}} \rho_1 n_{12} . (\bar{v}_1 - w) dV \tag{2.37}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

where, $\bar{\rho_1} = \frac{1}{V}\int_{A_{12}} \rho_1 dV = \frac{V_1}{V}\frac{1}{V_1}\int_{A_{12}} \rho_1 dV = \alpha\rho_1$ and w is the average velocity at the interface.

Similarly for phase 2, we have:

$$\overline{\frac{\partial \rho_1}{\partial t} + \nabla.(\rho_1 v_1)} = 0 = \rho_2 \frac{\partial(1-\alpha)}{\partial t} + \nabla.\rho_2\bar{v_2} - \frac{1}{V}\int_{A_{12}} \rho_2 n_{12}.(\bar{v_2} - w)dV \tag{2.38}$$

The integral term accounts for the mass transfer between the two phases. We denote $\dot{m}_{21}$ as the mass transferred from phase 1 to phase 2.ie,

$$\dot{m}_{21} = \frac{1}{V}\int_{A_{12}} \rho_1 n_{12}.(\bar{v_1} - w)dV \tag{2.39}$$

$$\dot{m}_{12} = \frac{1}{V}\int_{A_{12}} \rho_2 n_{12}.(\bar{v_2} - w)dV \tag{2.40}$$

For a simple multiphase flow, the mass transfer is zero and we set $\dot{m}$ as 0. Depending on the physics of the simulation, mass transfer can be either one way or both ways between the phases. We can generalise the mass conservation for both phases as:

$$\frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1\bar{v_1}) = \frac{\dot{m}}{\rho_1} \tag{2.41}$$

$$\frac{\partial \alpha_2}{\partial t} + \nabla.(\alpha_2\bar{v_2}) = -\frac{\dot{m}}{\rho_2} \tag{2.42}$$

We also know that $\alpha_1 + \alpha_2 = 1$. This lets us to write global mass conservation equation:

$$\nabla.\bar{v} = \dot{m}\Big(\frac{1}{\rho_1} - \frac{1}{\rho_2}\Big) \tag{2.43}$$

Volume fraction equation from mass conservation
The equation obtained from the previous section contains terms such as v_1 and v_2 which are not available at hand. Hence we need to rewrite the equation using the cell averaging formula for phase velocities as shown below.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1 \overline{v_1}) = \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1(\alpha_1 + \alpha_2)\overline{v_1}) \tag{2.44}$$

$$= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1\alpha_1\overline{v_1}) + \nabla.(\alpha_1\alpha_2\overline{v_1}) \tag{2.45}$$

$$= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1\alpha_1\overline{v_1}) + \nabla.(\alpha_1\alpha_2\overline{v_1}) - (\nabla.(\alpha_1\alpha_2\overline{v_2}) - \nabla.(\alpha_1\alpha_2\overline{v_2})) \tag{2.46}$$

$$= \frac{\partial \alpha_1}{\partial t} + \nabla.(\alpha_1(\alpha_1\overline{v_1} + \alpha_2\overline{v_2})) + \nabla.(\alpha_1\alpha_2(\overline{v_1} - \overline{v_2})) \tag{2.47}$$

$$= \frac{\partial \alpha}{\partial t} + \nabla.(\alpha_1\overline{v}) + \nabla.(\alpha_1\alpha_2\bar{v}_r) \tag{2.48}$$

We now have the time evolution equation of α using known parameters. ie.

$$\frac{\partial \alpha}{\partial t} + \nabla.(\alpha\bar{v}) + \nabla.(\alpha(1-\alpha)\bar{v}_r) = \frac{\dot{m}}{\rho_l} \tag{2.49}$$

where, $\overline{v_r} = \overline{v_1} - \overline{v_2}$ is the compressive velocity. It is also called added artificial velocity since it compress the interface and insures the stiffness of the interface. But evaluation of $\bar{v}_r$ requires $\overline{v_1}$ and $\overline{v_2}$. Hence we need to approximate $\bar{v}_r$ using the below relation:

$$\bar{v}_{r,f} = n_f min\Big[C_\alpha \frac{|\phi|}{|S_f|}, max_F(\frac{|\phi|}{|S_F|})\Big] \tag{2.50}$$

where n_f is the face unit vector, ϕ the flux through the face, and S_F it's surface. C_α lets the user take control of the spread of the interface over the cells.

2.3 Choice of Multi-phase Models

Lagrangian-Eulerian models are suitable for simulating particle-laden flows or sprays. In general, it is used for modeling flows where the secondary phase has particulate nature and mass transfer is not involved between the phases.

Euler models can be conveniently used for numerical simulations involving mass transfer. Among Euler models, the VoF model is the most preferred model due to its intrinsic ability to separate the different phases using a scalar function. The model is easy to implement and has less intensive calculations per mesh cell compared to Euler-Euler (Two-fluid method). The scalar function has strict upper and lower bounds and hence acts as an interface between the phases. This is important for mass transfer calculations since the interface area is directly proportional to the rate of mass transfer. This scalar function can be manipulated appropriately for the construction of a strict interface between the phases.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Previous studies on species transfer across the interface such as the CST method were carried out based on the VoF approach. The current study attempts to explore the possibility of geometrical interface area calculation methods in mass transfer simulations together with existing mass transfer models. In the later chapters, we see how the interface is redefined using geometrical interface reconstruction algorithms and how it compares to the existing interface reconstruction algorithms.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 3

Interface Area Calculation

In the previous chapter, we discussed the development of the volume fraction time evolution equation 2.49. In this equation, we saw that the RHS source term corresponds to the mass transfer from one phase to the other. Once the evaluation of the mass flux(J) is done, we need to find the interface area between the phases. In the VoF model, we have a continuous-continuous domain. Hence it is possible to define a strict interface that passes through only one cell and divides the phases into two. There are several methods available to approximate the interface area using available information such as volume fraction. The accuracy of the mass transfer model greatly depends on the evaluation of the interface area. In the coming sections, we discuss the various methods to evaluate the interface area.

3.1 Gradient Alpha Method

The magnitude of the gradient of volume fraction-α gives the approximate area of the interface between the phases[33].

Interface area:

$$A_{interface} = |\nabla \alpha| \tag{3.1}$$

The above formula gives us the interface area per unit volume. The idea behind this approximation comes from measure theory and properties of function spaces.

Let f be a Lipschitz continuous function, $f : R^n \longrightarrow R^m$ where $m \geq n$. According to the area formula, the space measure of $f(A)$ can be obtained by integrating the jacobian of f over A, where A is a subset of R^n.

As an example, let $n \geq 1$ and $m = n + 1$. ie $f : R^n \longrightarrow R^{n+1}$. We can write $f = (f^1, .., f^n)$.

$$Df = \begin{pmatrix} f^1_{x_1} & \cdots & f^1_{x_n} \\ \vdots & \ddots & \vdots \\ f^{n+1}_{x_1} & \cdots & f^{n+1}_{x_n} \end{pmatrix}_{(n+1)\times n} \tag{3.2}$$

Df is the jacobian of f.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

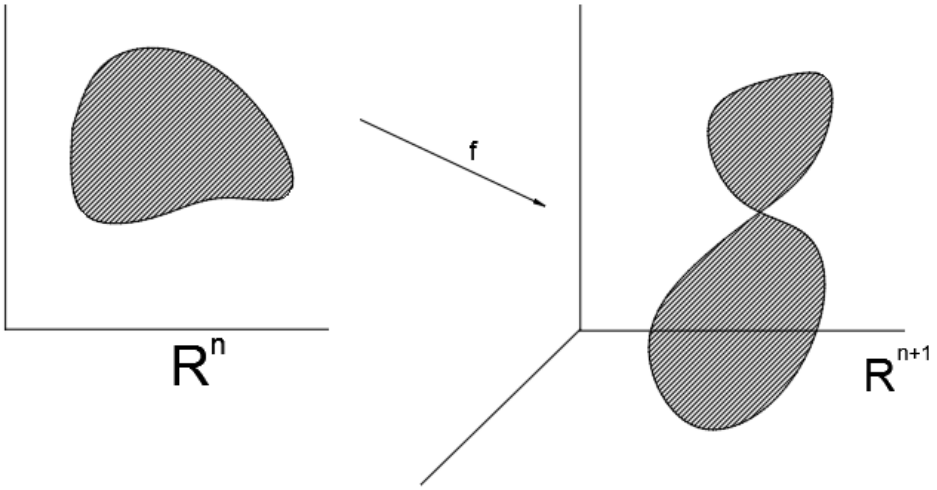


FIGURE 3.1: Mapping from R^n to R^{n+1} space

Let $A \subseteq R^n$. Hence we can write:

$$S = f(A) \subseteq R^{n+1}$$

The surface area of the n dimensional domain, $H^n(S)$ is given by the area formula:

$$H^n(S) = \int_A \left(\sum_{k=1}^{n+1} \left[\frac{\partial(f^1, ..., f^{n+1})}{\partial(x_1, ..., x_n)} \right]^2 \right)^{\frac{1}{2}} dx \tag{3.3}$$

In 3D, the above equation becomes:

$$A = \int_V \left[(\frac{\partial f}{\partial x_1})^2 + (\frac{\partial f}{\partial x_2})^2 + (\frac{\partial f}{\partial x_3})^2 \right]^{\frac{1}{2}} dx \tag{3.4}$$

where A is the interface area. In a mesh cell with volume V, the above equation can be further simplified as:

$$A = \left[(\frac{\partial f}{\partial x_1})^2 + (\frac{\partial f}{\partial x_2})^2 + (\frac{\partial f}{\partial x_3})^2 \right]^{\frac{1}{2}} V \tag{3.5}$$

$$= |\nabla f| V \tag{3.6}$$

In our present 2D case, f is same as α. Dividing equation 3.5 with V gives equation 3.1 .ie, the interface area per unit volume in a cell.

3.2 Geometrical Area Calculation Methods

Under the continuous-continuous flow domain assumption of the VoF model, we now try to reconstruct geometrically a plane on the interface cell and then calculate the area of the reconstructed surface using simple area calculation formulae.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.2.1 Iso-Alpha Method

In this method, we define a parameter f based on the interpolated volume fraction at the cell vertices. Here we introduce the concept of isovalues and isosurfaces. The optimum value of the parameter f used to construct the interface is called the isovalue. The interface now constructed is called the isosurface. We say the value of f is optimum when the geometrical volume fraction matches with the volume fraction obtained from solving the α-equation for interface advection[10].

Steps for Interface Reconstruction

Step 1: Identification of Interface Cells. Since we have a continuous-continuous flow domain, we have volume fraction-α 0 and 1 everywhere except on the interface cells. Hence, a cell is an interface cell if it satisfies:

$$\epsilon < \alpha_i < 1 - \epsilon \tag{3.7}$$

where ϵ is a user-defined tolerance.

Step 2: Defining the parameter f: We now interpolate the α values to the interface cell vertices using the information (α values) from neighboring cells. The inverse of the distances between the grid points and cell centers is used as interpolation weights. We denote the interpolated volume fractions as $f_1, f_2, f_3, ..f_n$, where n is the number of vertices in the interface cell. (See figure 3.2).

$$f_v = \frac{\sum_{k \in C_v} w_k \alpha_k}{\sum_{k \in C_v} w_k} \tag{3.8}$$

where C_v are the cells near the interface cell whose α values are used to obtain f_v

Step 3: Construction of $f - isoface$: An edge of the interface cell is cut based on the following rule:
Let (X_k, X_l) be the edge of an interface cell. The egde has f values f_k, f_l. An arbitrary $f_0 - isoface$ should satisfy $f_k < f_0 < f_l$. The edge is then cut by linear interpolation of the edges using the equation:

$$X_{cut} = X_k + \frac{f - f_k}{f_l - f_k}(X_l - X_k) \tag{3.9}$$

Following this rule, all the edges of the interface cell are cut. Now, these points can be connected to form the $f - isoface$ of the interface cell.

Step 4: Geometrical Volume Fraction Calculation: The isoface cuts the interface cell into 2 polyhedral cell domains A and B. Geometrical volume

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

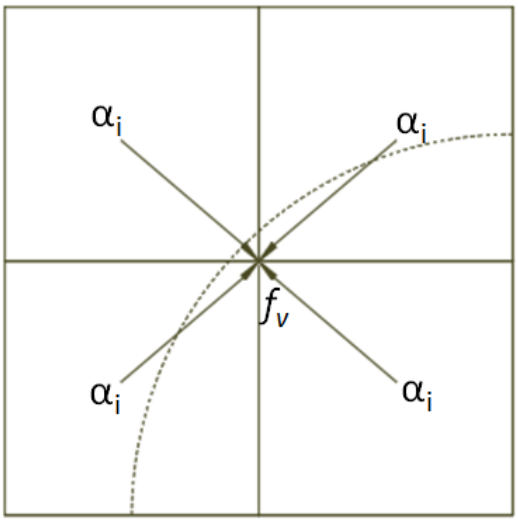


FIGURE 3.2: Interpolation of volume fraction to the interface cell vertices

fraction of the interface cell is calculated as:

$$\alpha(f) = \frac{vol(A_i(f))}{V_i} \tag{3.10}$$

where $vol(A_i(f))$ is the volume of the immersed cell corresponding to the $f - isovalue$ and V_i is the total volume of the interface cell.

The iso-surface now forms two polyhedral volumes within the cell. To find the volume of the immersed volume, we split the cell using pyramid decomposition. We choose an arbitrary point inside the immersed volume and connect the vertices to that point. It decomposes the polyhedron into smaller pyramids whose volume is calculated using the formula:

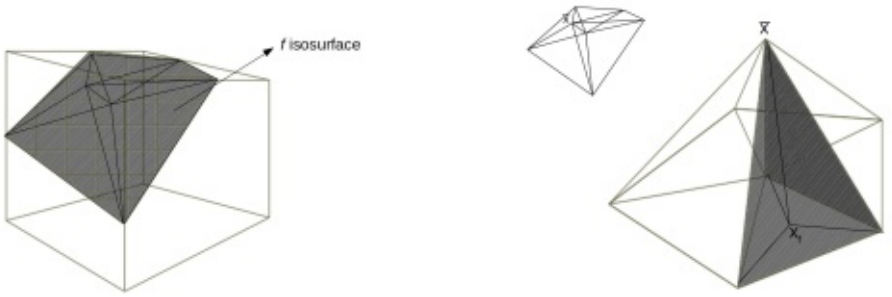
$$V = \sum_f \frac{1}{3}|n_f.(x_f - \bar{x})| \quad where, \quad \bar{x} = \frac{1}{N_f}\sum_f x_f \tag{3.11}$$

f refers to the faces of the polyhedron. n_f, x_f are the normal vectors and face centres respectively defined below:

$$n_f = \sum_{k=1}^{N_v} n_{f,k}, \quad where \quad n_{f,k} = \frac{1}{2}(x_{k+1} - x_k)(\bar{x} - x_k) \quad and \quad \bar{x} = \frac{1}{N_v}\sum_{k=1}^{N_v} x_k \tag{3.12}$$

$$\hat{n}_f = \frac{n_f}{|n_f|} \quad and \quad x_f = \sum_{k=1}^{N_v} \frac{n_{f,k}}{n_f}\frac{x_k + x_{k+1} + \bar{x}}{3} \tag{3.13}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



(A) f iso-surface interface

(B) Polyhedron formed by connecting isosurface vertices to centre of immersed volume($\bar{x}$)

FIGURE 3.3: Immersed Volume Calculation Steps

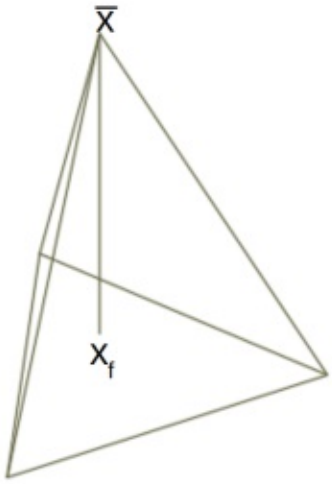


FIGURE 3.4: Pyramid formed by connecting vertices of triangulated iso-surface with $\bar{x}$

Finding the Optimum iso-value

We saw that $\tilde{\alpha}$ is a function of f^*. The value of f can vary anywhere between $min(f_1, f_2, ...f_n)$ and $max(f_1, f_2, ..f_n)$. It means that $\tilde{\alpha}$ can vary anywhere between 0 and 1. When $\tilde{\alpha}$ matches with the α from the volume fraction equation, we say that the current f^* value is the optimum iso-value.ie,

$$\tilde{\alpha}(f^*) = \alpha_i \tag{3.14}$$

We find the optimum isovalue using a root-finding procedure described below:

We assume that the value of $\tilde{\alpha^*}$ varies as a cubic polynomial in f. The aim is to form a cubic polynomial equation using available values of f and α and solving it using a Vandermonte matrix and LU factorization.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1. Geometrically calculate f_1, f_n for all vertices.
2. Evaluate $\tilde{\alpha}(f)$ for all $f_1, f_2, ..f_N$ and find an interval $[f_k, f_l]$ which is likely to have the f^*. ie $f^* \in [f_k, f_l]$ corresponding to an edge.
3. Evaluate two additional $\tilde{\alpha}(f)$ in $[f_k, f_l]$ so that we have fours supporting equations to find the coefficients of the cubic polynomial.
4. The resulting 4x4 matrix system can be solved using LU decomposition such that $|\frac{V_A(f^*)}{V_i} - \alpha_i| < \epsilon$, where ϵ is the tolerance input.

3.2.2 Iso-RDF Method

RDF stands for Reconstructed Distance Function. In this method, we reconstruct the interface using a reconstructed distance function. This distance function takes into account the orientation of the reconstructed interface which was not accounted for in the iso-Alpha method. The method follows a similar algorithm until finding the interface area vector x_s and normal vector $\hat{n_s}$. We use these values to calculate the distance function. The distance function then allows for more accurate estimates for the area and normal vectors[11].

Steps for Interface Reconstruction

Step 1: Similar to the iso-Alpha method, we find the interface cells subject to a user-defined tolerance.

Step 2: Calculate RDF, Ψ on all the interface cells and the neighboring cells which share at least one point with the interface cells.

Calculation of RDF

We first define the distance function, $\tilde{\Psi}$ as the perpendicular distance from a neighboring cell to the interface cell.

$$\tilde{\Psi} = \hat{n}_{S,j}.(x_i - x_{S,j}) \tag{3.15}$$

We use these distances from all neighboring cells to calculate the RDF of each interface cell using:

$$\Psi_i = \frac{\sum_j w_{ij}\tilde{\Psi}_{ij}}{\sum_j w_{ij}} \tag{3.16}$$

where w is the weighing function and is calculated as,

$$w_{ij} = \frac{|\hat{n}_{S,j}.(x_i - x_{S,j})|^A}{|x_i - x_{S,j}|^A}, \quad with \quad A = 2 \tag{3.17}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

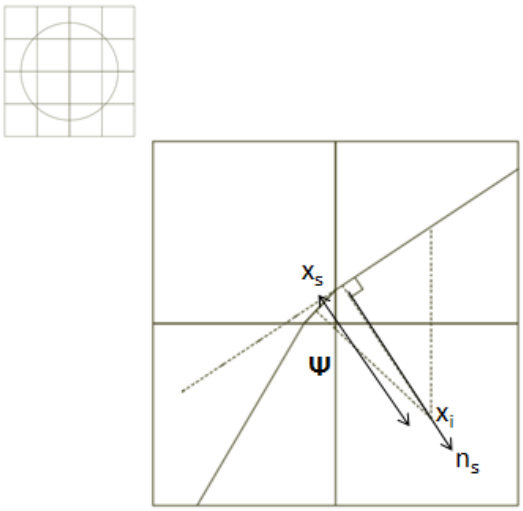


FIGURE 3.5: Distance function on a cell near the interface

Step 3: Interpolate the RDF values to the cell vertices. Here we use the least squares method to calculate the weights for interpolation.

$$f_v = \frac{\sum_{k \in C_v} w_k \Psi_k}{\sum_{k \in C_v} w_k} \tag{3.18}$$

This step is similar to the second step in the iso-Alpha method.

Step 4: Face centre, $x_{S,i}^{new}$ and face unit normal, $\hat{n}_{S,i}^{new}$ are calculated using the new Ψ_v values at the vertex. We follow step 3 and step 4 as we saw in the iso-Alpha method replacing f_v with ψ_v.

We continue recalculating the new face centers and unit normals until the difference in the new and old normals converge to a predefined tolerance.

ie, the residual = $\frac{1}{N} \sum_i^N |1 - \hat{n}_{S,i}.\hat{n}_{S,i}^{new}|$

3.2.3 PLIC-RDF Method

The PLIC-RDF method makes use of the best out of both PLIC-Piecewise Linear Interface Construction and RDF methods. The method reconstructs the interface via the equation of a plane as a starting point and later corrects the interface normal using a distance function as explained in the RDF method[11].

Steps for Interface Reconstruction

Step 1: Identify the interface cells based on a user defined tolerance ϵ
ie, $\epsilon < \alpha_i < 1 - \epsilon$

Step 2: Estimation of $\hat{n}_s$. We use the normal based on gradient alpha for

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

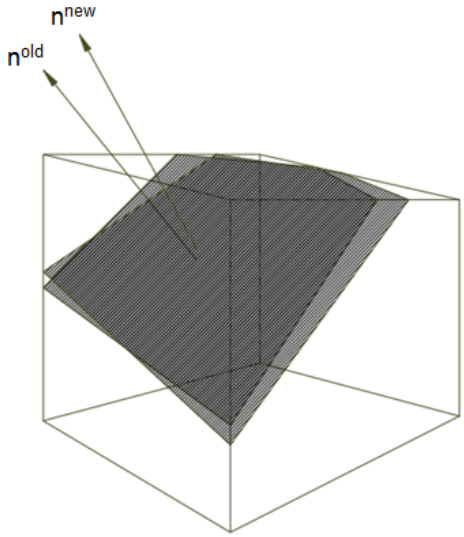


FIGURE 3.6: New and old normal vectors on the isosurface

the first time-step. For later time-steps we use the normal vector from previous timesteps ($\hat{n}_S^{old}$) and interpolate it using weighing functions under the condition that β_i is less than 10 degrees. If $\beta > 10^o$, we use $\nabla\alpha/|\nabla\alpha|$ as initial guess for the normal. We define β as:

$$\beta_i = \frac{\sum_j arccos(\hat{n}_{S,i}.\hat{n}_{S,j})|n_{S,j}|}{\sum_j |n_{S,j}|} \tag{3.19}$$

$$\hat{n}_{S,i} = \frac{\sum_j w_{ij}\hat{n}_{S,j}^{old}}{\sum_j w_{ij}} \tag{3.20}$$

where w is the weighing factor, defined as:

$$w_{ij} = |n_{S,j}^{old} x[(x_i - u_i\Delta t) - x_{S,j}] \tag{3.21}$$

with u_i as the flow field velocity in cell i.

The PLIC surface has the equation of the form:

$$x_v.\hat{n}_{S,i} - d_v = 0 \tag{3.22}$$

We use the estimated normal $\hat{n}_{S,i}$ and vertex x_v to calculate the projected vertex position d_v. Using d_v as the vertex values, we calculate the center $x_{S,i}$ of the d^* isosurface having $A(d^*)$ as volume fraction. We follow similar procedure as in steps 3 and 4 of iso-Alpha method to find the optimum d^* value by solving $A(d^*) = \alpha_i$.

Step 3: Calculation of RDF. Based on the values of x_S and $\hat{n}_S$ we calculate ψ as described in the iso-RDF method.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Step 4: Calculation of improved interface normal using $\nabla\Psi$.ie

$$\hat{n}_S^{new} = \nabla\Psi/|\nabla\Psi| \tag{3.23}$$

Step 5: Residual calculation for convergence. The residual *res* is evaluated as the difference between $\hat{n}_s$ and $\hat{n}_S^{new}$.

ie, the residual = $\frac{1}{N}\sum_i^N |1 - \hat{n}_{S,i}.\hat{n}_{S,i}^{new}|$

Once the residual is below the tolerance set by the user we stop the iteration. Otherwise, we update the value of $\hat{n}_S$. ie, $\hat{n}_S = \hat{n}_S^{new}$

3.3 Choice of Interface Area Models

We know that an ideal interface between the phases should only be one cell thick. So that the species transfer from one phase to the other happens only through that one cell divided by the interface. If the interface is spread over more than one cell, it leads to excessive and non-physical mass transfer from one phase to the other. The gradient alpha method has this drawback since it is an area approximation method.

Geometrical area calculation methods are more accurate regarding interface reconstruction. As a result, it is possible to obtain an interface through a minimum number of cells compared to gradient alpha method. In the present study, we have chosen the iso-Alpha method for interface area calculation. The iso-Alpha method is the fastest method among the three geometrical methods described in the chapter. Moreover, its implementation in the mass transfer model is not excessively complicated. The motive of the study is the effectiveness of the geometrical interface reconstruction methods and their feasibility in mass transfer simulations. Hence we adopted the iso-Alpha method for the current research.

A comparison of the gradAlpha and isoAlpha methods is made and its effect on the rate of mass transfer between the phases is assessed in the results chapter.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 4

Numerical Model Implementation

This chapter discusses the implementation of the multiphase mass transfer model coupled with the interface area calculation library. OpenFOAM solver is used for this purpose and is available as an open-source software package. At a fundamental level, OpenFOAM is a collection of C++ libraries specifically for the purpose of solving Computational Fluid Dynamics problems. Since it is open-source, the users are free to use existing solvers and modify them to suit specific engineering problems.

We make use of the VoF multiphase model for the current research on mass transfer. The *interFoam* solver is the base solver for incompressible multiphase problems using VoF. In this thesis, we use multiPhaseFoam solver developed by Bahram Haddadi (TU Wien) as a starting point for the problem.

4.1 OpenFOAM Solver: multiPhaseFoam

multiPhaseFoam can solve multiphase problems involving two compressible immiscible fluids using the VoF model and is based on the interFoam solver. It also has the capabilities to solve non-isothermal flows with mass transfer between the phases. Turbulence models can be chosen between laminar, RAS or LES.

4.1.1 Mass Transfer Model

Mass transfer between the phases is incorporated as a source term in the volume fraction equation. The evaluation of the term involves the calculation of the mass flux at the interface and area of the interface (discussed in chapter 3). In order to track the transferred mass, we solve an additional species transport equation[34].

The rate of mass transfer can be formulated as:

rate of mass transfer, $\dot{m}$ = k x (interfacial area)x(concentration difference)

ie:

$$\frac{dm_i}{dt} = k_{i1} A (C_i^* - C_i) \tag{4.1}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

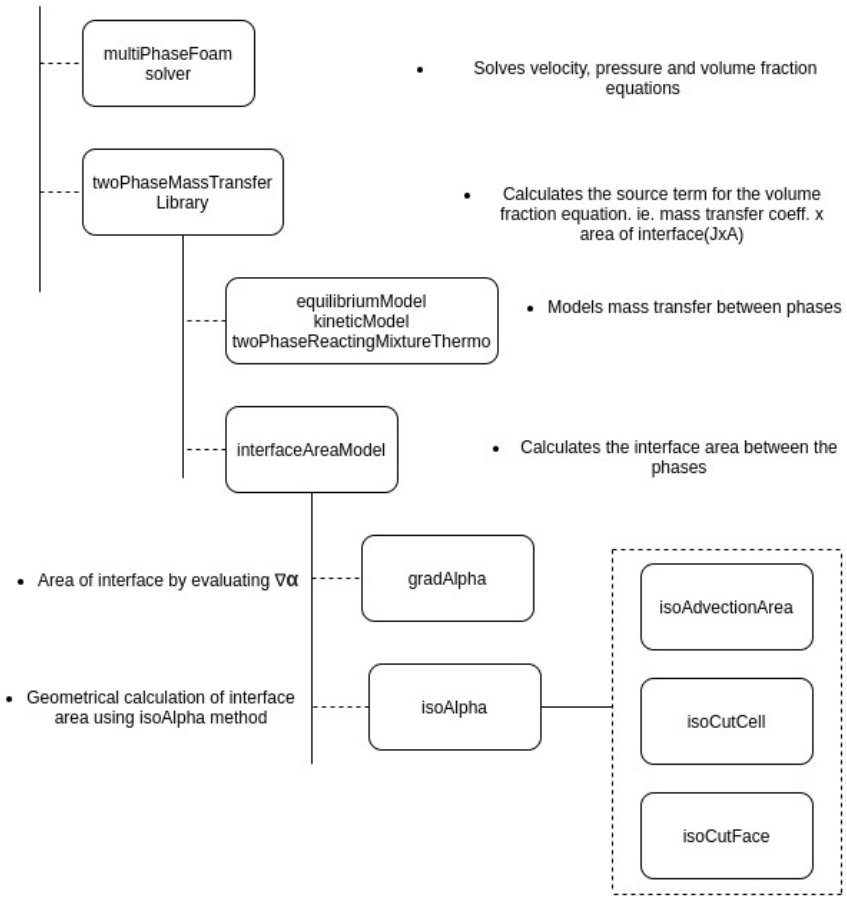


FIGURE 4.1: multiPhaseFoam solver structure

where k_{i1} is the mass transfer coefficient of specie i in phase 1. C_i^* is the saturation concentration (concentration at the interface when in equilibrium) and C_i is the specie concentration in the bulk (phase 1 or phase 2) and A is the interfacial area.

OpenFOAM uses Y_i parameter for the specie concentration. It is related to the actual specie concentration C_i as:

$$Y_i = \frac{C_i}{\rho} \tag{4.2}$$

where ρ is the density of the bulk fluid.

And the specie transport equation would read as:

$$\frac{\partial \alpha_1 \rho_1 Y_i}{\partial t} + \nabla \cdot (\alpha_1 \rho_1 U_1 Y_i) - \nabla D \cdot \nabla(Y_i) = \frac{dm_i}{dt} \tag{4.3}$$

Here α is the volume fraction of the bulk phase and D is the diffusion coefficient

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In multiPhaseFoam, the following models let us calculate the mass transfer terms:

- interfaceAreaModel
- equilibriumModel
- kineticModel

The implementation details of the **interfaceAreaModel** is discussed in the previous chapter.

equilibriumModel and **kineticModel** work together to return the equilibrium value of concentrations at the interface or in the fluid bulk in equilibrium. The species concentration at either side of the interfaces is related using Henry's Law. The law states:

$$C_{i1} = HC_{i2} \tag{4.4}$$

where C_{i1} is the concentration of specie-i in phase 1 and H is the Henry's constant which is conveniently taken as 1 for numerical modeling purposes.

4.1.2 PIMPLE Algorithm for VoF Model

PIMPLE is a hybrid predictor-corrector algorithm using SIMPLE (Semi-Implicit Method for Pressure Linked Equations) and PISO (Pressure Implicit with Splitting of Operators) combined. We have chosen PIMPLE algorithm due to its higher stability and capability to handle flows with Courant numbers higher than 1 thereby letting us simulate flows with larger time steps[35][36].

The different NSE solution algorithms such as SIMPLE, PISO, PIMPLE, etc., differ only in the number of inner and outer iterations. The inner iterations are performed on the Poisson pressure equation. The pressure value now obtained is used to correct the intermediate velocities and pressure. This contributes to the PISO loops in the PIMPLE algorithm. The outer iterations are used to update the velocity matrix in the momentum equations. According to the convention used in OpenFOAM, *nCorrectors* refers to the number of inner iterations and *nOuterCorrectors* refers to the outer iterations. The PIMPLE algorithm works as a PISO algorithm if the *nOuterCorrectors* is set as one.

4.1.3 Discretization of NS Equations

The integral form of the momentum equation reads:

$$\int_V \frac{\partial \rho u}{\partial t} dV + \int_V \nabla \cdot \rho u u dV = -\int_V \nabla p dV + \int_V \nabla \cdot (\nabla u) dV + \int_V S_s dV \tag{4.5}$$

where S_s is the source term arising from forces such as surface tension forces etc.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

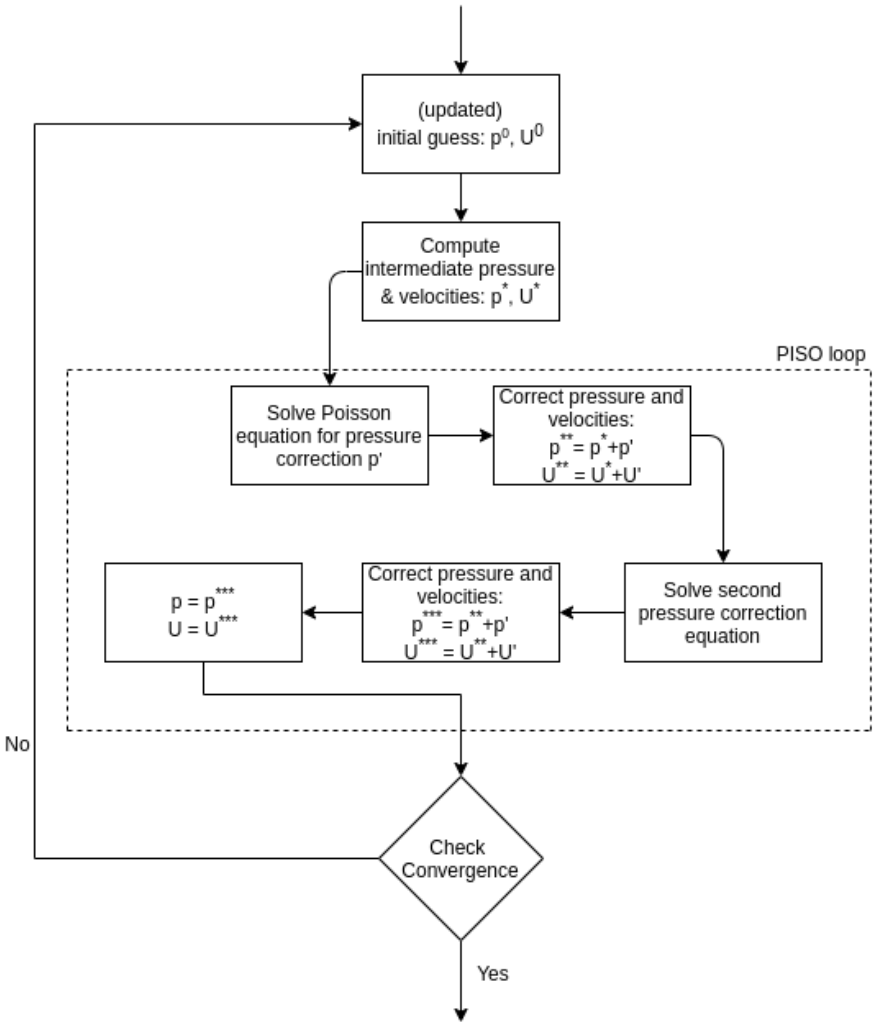


FIGURE 4.2: PIMPLE Algorithm flow chart

Finite Volume Method Discretization of the individual terms can be summarized as follows[37][38][39][40]:

Discrete Continuity equation:
We use Gauss's identities for the discretization of the spatial terms. The below identity is used for the discretization of the continuity equation:

$$\int_V \nabla \cdot a dV = \oint_{\partial V} dV \cdot a \tag{4.6}$$

Replacing the surface integral with summation over the cell faces gives us:

$$\int_V \nabla \cdot u dV = \sum_f S \cdot u \tag{4.7}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

where S is the magnitude of the cell face area and u is the interpolated value of velocity onto the cell faces. In order to account the direction of the face area vector, we split the summation into 'owned' and 'neighbouring' faces:

$$\sum_f S \cdot u = \sum_{owner} S_f \cdot u - \sum_{neighbour} S_f \cdot u \tag{4.8}$$

where S_f is the face area vector.

Temporal term:
Let ϕ be a time dependent parameter. The first order accurate linear variation in time reads[35]:

$$\phi(t + \Delta t) = \phi(t) + \Delta t\left(\frac{\partial \phi}{\partial t}\right) \tag{4.9}$$

Using the above variation of the function in time, the temporal integral corresponding to the parameters $u(t), \rho(t)$:

$$\int_V \frac{\partial(\rho u)}{\partial t} dV \approx \frac{\rho^{n+1}u^{n+1} - \rho^n u^n}{\Delta t} V \tag{4.10}$$

Convection term:
Here we again use Gauss's divergence theorem for converting the volume integral to surface integral.

$$\int_V \nabla \cdot (\rho u u) dV = \underbrace{\int_S ds \cdot (\rho u u)}_{\text{gauss div thm.}} \approx \sum_f S \cdot (\rho u)_f u_f = \sum_f F u_f \tag{4.11}$$

where S is the surface area on the mesh cell faces and F is the flux through the face F defined as $F = S \cdot (\rho u)_f$. This operation requires interpolated values of velocities on the cell faces. Linear interpolation and other named interpolation techniques are available in OpenFOAM.

Diffusion term:
The diffusion term is discretized in a similar way as the convection term using the assumption of linear variation of u and using Gauss's divergence theorem:

$$\int_V \nabla \cdot (\mu \nabla u) dV = \oint_f dS \cdot (\mu \nabla u) \approx \sum_f \mu (S \cdot \nabla_f u) \tag{4.12}$$

In the above equation ∇_f is the face gradient and is calculated similar to equation 4.8.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Pressure term:

$$\int_V \nabla p dV = \oint_f p \cdot n dS = \sum_f S \cdot P_f \tag{4.13}$$

where p_f is the interpolated pressure values on the cell faces in a collocated grid.

Source terms:

The source terms are first linearized and then integrated over the control volume.
$S(\psi) = S_I\psi + S_E$, where S_I and S_E may depend on the dependent variable, say ψ.

$$\int_V S(\psi) dV = S_I V\psi + S_E V \tag{4.14}$$

Without going into the specifics of available discretisation schemes in OpenFOAM, we derive the semi discretized form of the Navier Stokes equations as shown below:

$$(\frac{\partial \rho u}{\partial t})_p + \sum F u_f - \sum_f \mu(S \cdot \nabla_f u) = -(\nabla p)_p + S \tag{4.15}$$

The above equation can be written algebraically as:

$$a_P u_P + \sum_N a_N u_N = -\nabla p + S \tag{4.16}$$

where subscript p refers to values in the current control volume and subscript N refers to the values from neighboring cells. This is done to assist the solution algorithms such as SIMPLE, PISO, PIMPLE, etc. a_P is a diagonal matrix. The invert of a diagonal matrix can be easily found by taking the reciprocals of the diagonal elements. a_N contains all the off-diagonal elements.

The above equation can be further simplified as:

$$a_P u_P = H(u) - \nabla p \tag{4.17}$$

$$u_P = \frac{H(u)}{a_P} - \frac{\nabla p}{a_P} \tag{4.18}$$

where $H(u) = -\sum a_N u_N + S$. The above equation is equivalent to the matrix equation:

$$[A][u] = [B] \tag{4.19}$$

4.1.4 Pressure-Velocity Coupling

In the discretized Navier stokes equations, we observe a linear dependence of velocity on pressure and vice versa. This inter-equation coupling is called

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

pressure-velocity coupling[41].

Momentum Predictor
In each time step in the PIMPLE algorithm, we find intermediate velocities u^* until the residual error drops below the tolerance set by the user. ie,

$$a_P u_P^* = H(u^*) - \nabla p \tag{4.20}$$

Since the Navier Stokes equation is implicitly solved in the PIMPLE algorithm, we use the values from the previous time-step to formulate the $H(u^*)$ matrix and the ∇p term. Hence the solution process is iterative in nature due to this inaccuracy in approximating the $H(u^*)$ term.

The intermediate velocity u^* can be found as:

$$u_p^* = a^{-1}H(u^*) - a^{-1}\nabla p \tag{4.21}$$

Pressure/Velocity Corrector
The predicted velocity does not satisfy the continuity condition since we assumed the pressure in the momentum predictor step. Hence we force the continuity condition to find a pressure value which helps to find a velocity that satisfies both continuity and momentum equations by iteration.

Imposing continuity condition on the momentum predictor step gives us:

$$\nabla \cdot (a^{-1}\nabla p) = \nabla \cdot (a^{-1}H(u^*) - \dot{m}(\frac{1}{\rho_1} - \frac{1}{\rho_2})) \tag{4.22}$$

The additional term on the right-hand side arises due to mass transfer between the phases. The poisson equation above is solved for p and used to correct the velocity as shown below:

$$u^{**} = a^{-1}H(u^*) - a^{-1}\nabla p^* \tag{4.23}$$

This constitutes the PISO loop of the PIMPLE algorithm. The pressure-velocity corrector step is repeated as many times as *nCorrectors* set in the OpenFOAM solver. And the velocity matrix (a) is updated as many times as specified in *nOuterCorrectors*. This constitutes the PIMPLE algorithm.

4.1.5 Solving Volume Fraction Equation: MULES

In the VoF multi-phase flow, we have an interface between the phases transported along the flow advected by the flow velocity. The parameter that characterizes this interface is the volume fraction-α. Hence the solution for the α equation requires a strict boundedness between 0 and 1 since $\alpha_1 + \alpha_2 = 1$. MULES (Multidimensional Universal Limiter for Explicit Solution) algorithm in OpenFOAM solves the α equation explicitly[38][39].

There are many solvers for passive advection of the interface such as the Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) used in ANSYS, High-Resolution Interface Capturing (HRIC) used

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

in StarCCM+ and isoAdvector (OpenFOAM) which is still under development. Performance comparison of the above methods suggest that MULES has superior interface reconstruction capabilities compared to HRIC and CICSAM[10].

MULES solver solves equations of the form:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot F = \alpha S_p + S_u \tag{4.24}$$

where, α is the volume fraction and F - the flux of α. S_p and S_u are the implicit and explicit source terms respectively.

In OpenFOAM, MULES solver can be evoked by calling the function:

$$MULES(\alpha, \phi, F, S_p, S_u, 1, 0) \tag{4.25}$$

where ϕ is a surface flux used to determine the upwind direction of the α flux-F. 1 and 0 in the argument list refer to the upper and lower bounds of the α solution.

If the source terms are absent one can also call the function as: $MULES(\alpha, \phi, F, 1, 0)$.

MULES Algorithm

MULES is the OpenFOAM implementation of the Flux Corrected Transport (FCT) theory[42]. Consider the hyperbolic equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \vec{F} = 0 \tag{4.26}$$

Our problem is to maintain the boundedness of the solution of the above equation. An explicit discretization of the above equation reads:

$$\frac{\phi + i^{n+1} - \phi_i^n}{\Delta t} V + \sum_f (\vec{F}^n \cdot \vec{S})_f = 0 \tag{4.27}$$

where F is the quantity to be transported and $\vec{F}$ its flux. i denotes the cell under consideration. For a one dimensional grid, the above equation can be re-written as:

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta t}{V}(F_{i+1/2}^n - F_{i-1/2}^n) \tag{4.28}$$

where $\vec{F}^n = (\vec{F}^n \cdot \vec{S})_f$ denotes the total flux.

The OpenFOAM way of solving the above equation is described below[38]:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1. Compute the transportive flux (F) using low and high order schemes. We denote them as F^L and F^H respectively.
2. We define an anti-diffusive flux $A = F^H - F^L$
3. Calculate Corrected flux, $F^C = F^L + \lambda A$ with $0 \leq \lambda \leq 1$
4. Equation 4.28 is now reformulated as:

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta t}{V}(F^C_{i+1/2} - F^C_{i-1/2}) \tag{4.29}$$

We observe that the evaluation of λ brings boundedness to the equation to be solved. OpenFOAM uses iterative methods for finding λ.

4.1.6 Numerical Schemes

OpenFOAM offers a variety of discretization schemes for the different terms in Navier Stokes and other associated equations. The chosen discretization schemes are specified in the *fvSchemes* dictionary in OF. Given below is a short description of the available discretization schemes and interpolation schemes[43].

Time Schemes

The $(\partial/\partial t)$ terms are available in the *ddtSchemes* sub-dictionary in OF. The following are the popular time schemes available in OpenFOAM:

Scheme	Description
Euler	First Order, Bounded, Implicit
CrankNicolson	Second Order, Bounded, Implicit
backward	Second Order, Implicit
steadyState	Does not solve for time derivatives

TABLE 4.1: Numerical schemes for temporal terms

Divergence Schemes

OpenFOAM by default uses Gauss divergence schemes. It also has a variety of interpolation schemes to choose from. The general format of the numerical scheme would look like:

$$Gauss < interpolationScheme > \tag{4.30}$$

The popular interpolation schemes in OF are:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Scheme	Description
linear	Second order, unbounded
upwind	First order, bounded
QUICK	First/second order, bounded
linearUpwind	First/second order, bounded

TABLE 4.2: Interpolation schemes for divergence terms

Laplacian Schemes

A common laplacian term encountered is the viscous term-$\nabla \cdot (\nu \nabla u)$ in NS equations. The evaluation of the laplacian term requires a discretization scheme, interpolation scheme and normal surface gradient scheme ie. ∇u. Selection is made via the following argument:

$$Gauss < interpolationScheme >< snGradScheme > \tag{4.31}$$

Interpolation schemes can be chosen from the previous table. Popular snGradSchemes are: corrected, uncorrected, limited.

Gradient Schemes

Evaluation of the gradient terms require selection of a discretization scheme. The following schemes are available:

Scheme	Description
Gauss <interpolationScheme>	Second order, Gaussian integration
leastSquares	Second order, least squares
cellLimited <gradScheme>	Cell limited version of one of the above schemes
faceLimited <gradScheme>	Face limited version of one of the above schemes

TABLE 4.3: Numerical schemes for gradient terms

4.2 Simulation Setup

In the framework of OpenFOAM, the simulation parameters such as the geometry, mesh, initial conditions, choice of discretization schemes, simulation time and time-step size, etc are set in the case directory. The important simulation settings are detailed in this section.

Geometry and Mesh

The geometry of the simulation is a simplified case of a spherical bubble in

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

a rectangular column. In 2D, the problem is a circular bubble(phase 1) in a rectangular domain(phase 2-water).

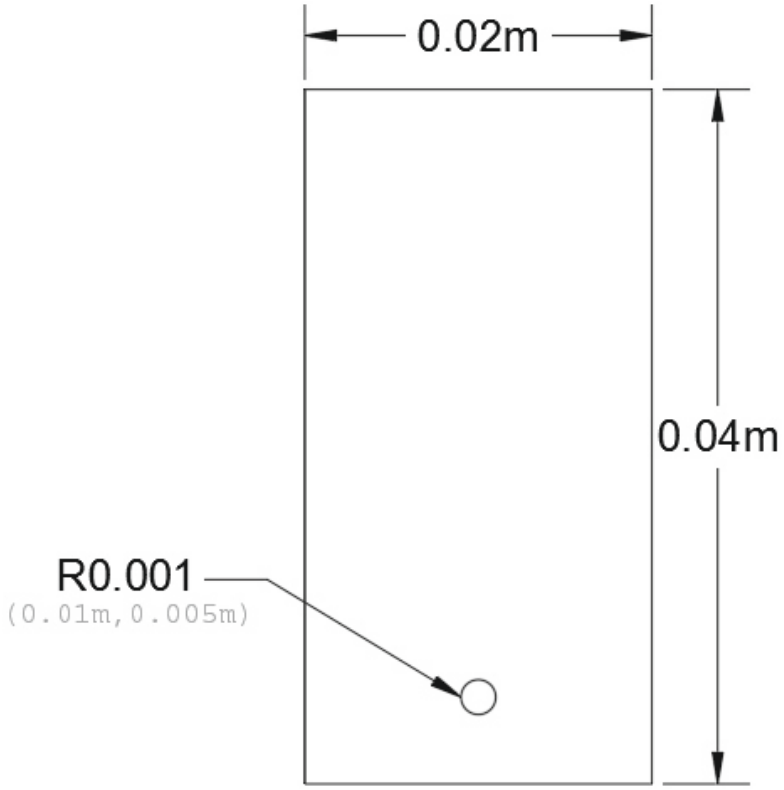


FIGURE 4.3: 2D Geometry for Simulation

blockMesh utility in OpenFOAM is used for the construction of a regular hexahedral mesh. The mesh is 3-dimensional with one cell thickness in the z-direction(OpenFOAM processes 2D geometry as 3D geometry with one cell thickness).

The rectangular column has the dimensions: 0.02m x 0.04m x 0.001m. The domain has 80000 mesh cells. The center of the bubble of radius 0.001m is located at (0.01m, 0.005m) with the left bottom corner as the reference origin. The meshed geometry is shown in figure4.3.

Time stepping

As a general rule for transient simulations to be stable, the Courant number(Co) is kept under 1. The explicit MULES algorithm works well with $Co < 0.25$ Hence the time step size is chosen accordingly. In the present simulation the chosen time step δt is $1x10^{-4}$. The simulation is run for 0.5 sec.

Initial and Boundary Conditions

The setFields utility in OpenFOAM maps non-uniform initial conditions onto the mesh. In the present simulation, we have 2 phases-water and air- with

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

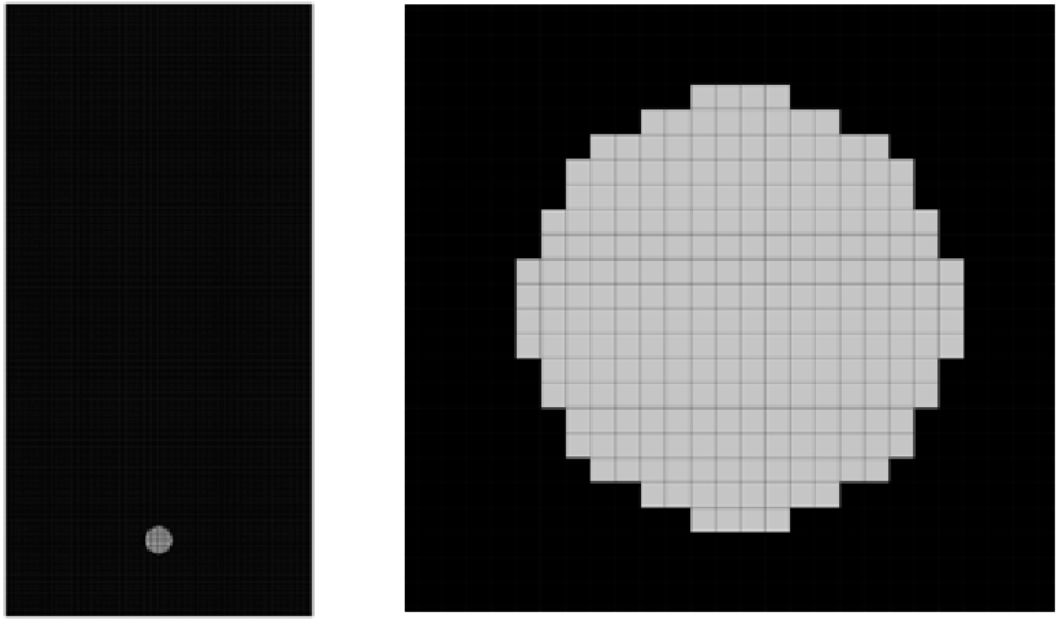


FIGURE 4.4: Meshed domain for Simulation at t=0
full domain(left) and closer view of the bubble(right)

corresponding volume fractions(α) 1 and 0. The mapped values are used as the initial condition for the α equation.

Boundary conditions are required for solving the Navier-Stokes equation. The α, pressure and velocity boundary conditions are as follows:

Velocity, U at the walls: (0 0 0)
Pressure, p at the walls: zeroGradient ie. $\nabla p = 0$
Volume fraction, α at the walls: zeroGradient ie. $\nabla \alpha = 0$

Discretisation Schemes
Information regarding discretization schemes is set in the fvSchemes dictionary in OpenFOAM. The file is located in the case/system directory. Chosen discretisation schemes for the simulation are as follows:

- Time Scheme, $(\partial/\partial t)$: Euler-first order implicit
- Gradient Scheme, (∇): Gauss Linear
- Divergent Schemes, $(\nabla \cdot)$: Gauss vanLeer, Gauss Linear, Gauss Upwind
- Laplacian Schemes, $\nabla \cdot (\nu \nabla U)$: Gauss Linear Uncorrected
- Interpolation Scheme: Linear

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 5

Results

A 'rising bubble with mass transfer' is taken as the test case. The simulation is run using OpenFOAM 5. Further analysis of the impact of isoAdvectorArea interface area calculation library is discussed in the subsequent sections. The simulation results are also compared against the regular rising bubble simulation which uses gradient-alpha interface area calculation algorithm for mass transfer. The simulation results are post-processed using paraView (paraFoam) visualization application. This chapter discusses the details of the transient simulation results.

5.1 Volume Fraction Results

This result helps us visualize the interface between the two phases as the bubble moves upwards due to buoyancy.

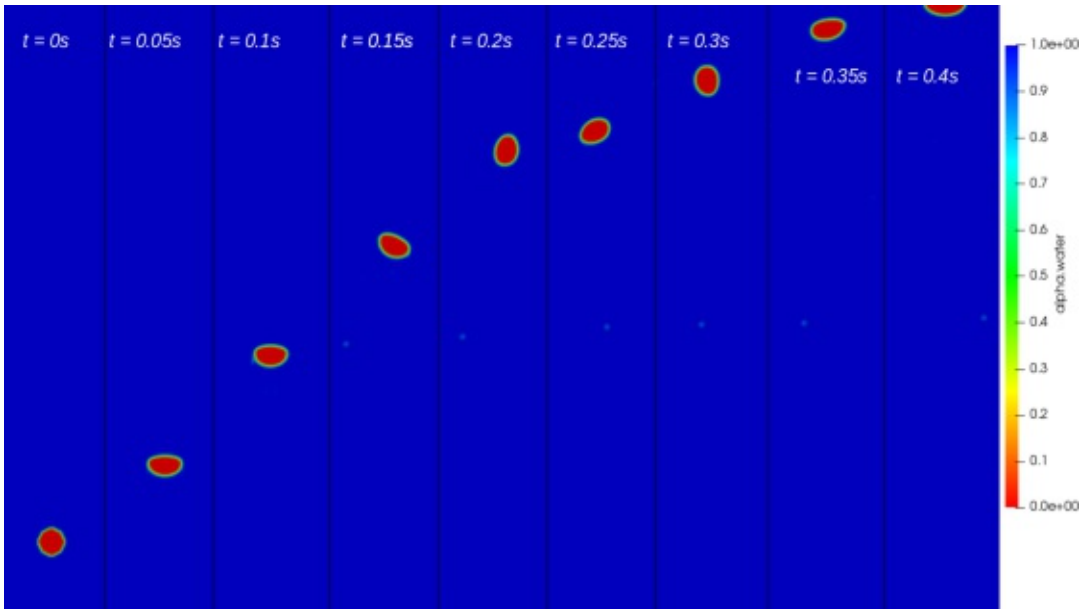


FIGURE 5.1: Volume fraction results: isoAlpha method

We observe a zig-zag motion of the bubble as it move upwards which is identical as in real life. See figure 5.1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.2 Interface Area Comparison

The interface area is extracted from the simulation and plotted in figure5.2.

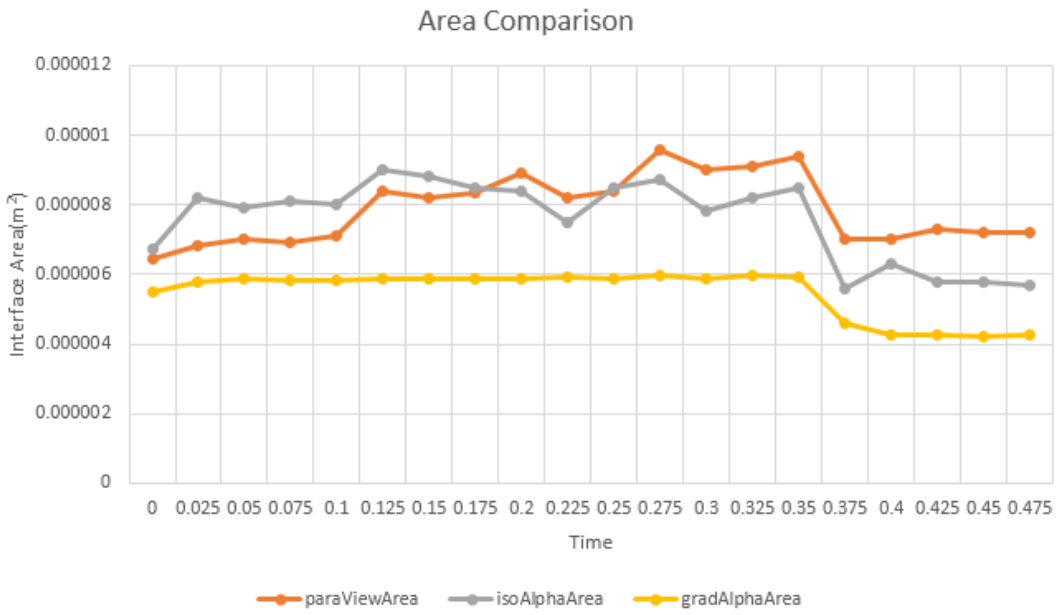


FIGURE 5.2: Interface area comparison

Interpretation of the above chart: We know that at time, t=0 , the bubble has the lowest possible surface area due to its circular shape. Once it starts rising upwards, the shape of the bubble distorts to an oval-like shape and as a result, the surface area increases. We observe an oscillation in the values of the interface area as the bubble rises. This is due to surface tension and simultaneous species transfer from the bubble to the fluid medium. We notice that the interface area increases till the time reaches around 0.375s. At that point, the bubble reaches the top and no longer retains the spherical (circular) shape and simulation start to subside. The flat part of the graph corresponds to this observation.

We also notice that the default gradAlpha method under-approximates the area and fails to capture the rapid changes in the interface area of the rising bubble as a result of mass transfer. The isoAlpha algorithm closely follows the interface area obtained from paraView post-processing utility. This is used as a form of validation in our current research. The interface area (in m^2) calculations from the gradAlpha and isoAlpha methods are tabulated below.

5.3 Analysis of the Reconstructed Interface

We know that a strict interface is only one cell thick. In figure 5.4 we see that the gradAlpha method(left) reconstructs an interface that is four to five cell thick. The interface reconstructed by isoAlpha method is sharper and only

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Area comparison chart

Time	paraViewArea	isoAlphaArea	gradAlphaArea
0	6.44E-06	6.748E-06	5.5E-06
0.025	6.84E-06	8.2E-06	5.8E-06
0.05	7.008E-06	7.9E-06	5.88E-06
0.075	6.9E-06	8.1E-06	5.81E-06
0.1	7.1E-06	8E-06	5.82E-06
0.125	8.4E-06	9E-06	5.86E-06
0.15	8.2E-06	8.8E-06	5.89E-06
0.175	8.32E-06	8.5E-06	5.89E-06
0.2	8.9E-06	8.4E-06	5.86E-06
0.225	8.2E-06	7.5E-06	5.93E-06
0.25	8.4E-06	8.5E-06	5.87E-06
0.275	9.6E-06	8.7E-06	5.95E-06
0.3	9E-06	7.8E-06	5.89E-06
0.325	9.1E-06	8.2E-06	5.96E-06
0.35	9.4E-06	8.5E-06	5.91E-06
0.375	7.01E-06	5.6E-06	4.59E-06
0.4	7E-06	6.3E-06	4.26E-06
0.425	7.3E-06	5.8E-06	4.24E-06
0.45	7.2E-06	5.8E-06	4.23E-06
0.475	7.2E-06	5.7E-06	4.24E-06

FIGURE 5.3: Interface area from isoAlpha and gradAlpha compared with area obtained from paraFoam

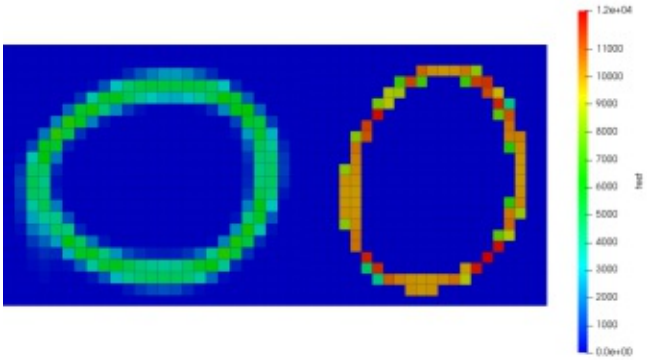


FIGURE 5.4: A closer look at the interface: gradAlpha (left) Vs isoAlpha (right) at an arbitraty timestep

two mesh cells thick in most parts of the interface. The reconstructed interface as time-lapse is shown below to compare the efficiency of the gradAlpha and isoAlpha methods.

As a result of the smeared interface, pseudo mass transfer through these non-interfacial cells around the actual interface cells is expected. Thus we expect a higher mass transfer rate in the gradAlpha method compared to the isoAlpha method despite the observation that isoAlpha method records a higher interface area according to the interface area table above.

In the area results, we see fluctuations in the isoAlpha area. This is attributed to the pseudo-interface area cells during the simulation as well as the interface orientation error intrinsic to the isoAlpha algorithm.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

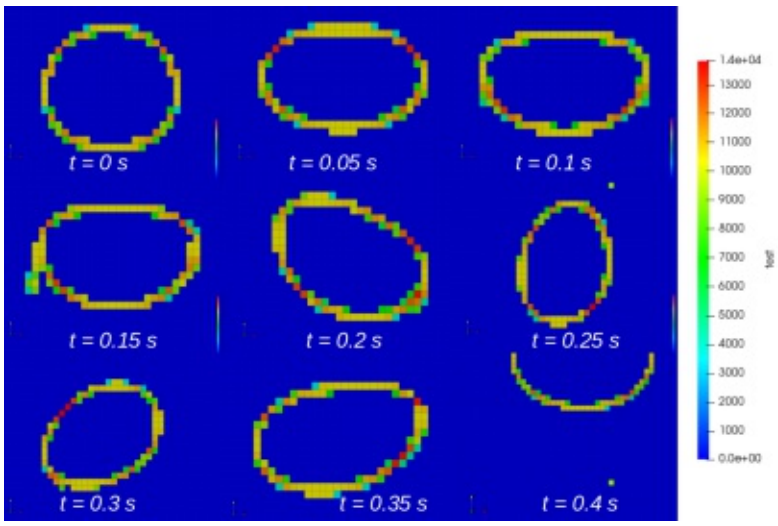


FIGURE 5.5: A closer look at the isoAlpha interface through regular time-steps

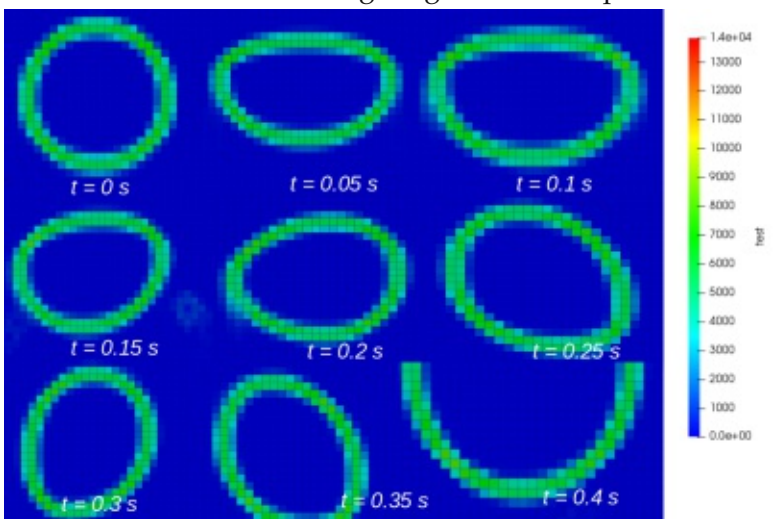


FIGURE 5.6: A closer look at the gradAlpha interface through regular time-steps

5.4 Specie Mass Transfer Results

The rate of mass transfer per unit volume is calculated during the simulation. We saw earlier that the mass transfer rate is:

$$\dot{m} = kA\Delta C \tag{5.1}$$

The mass transfer rate ($\dot{m}$) plotted against time is shown in Figure 5.7. We see that the average mass transfer rate of isoAlpha method is less than gradAlpha. It is due to the fact that gradAlpha method has far more interface cells distributed around the interface which contributes towards pseudo mass transfer from those cells. This is observed in the simulation results of the O_2 mass transfer and simultaneous comparison of mass transfer values

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

(mdot).

A comparison of the O_2 specie transfer results from isoAlpha and gradAlpha algorithms is shown below.

gradAlpha Vs isoAlpha Side-by-side Comparison

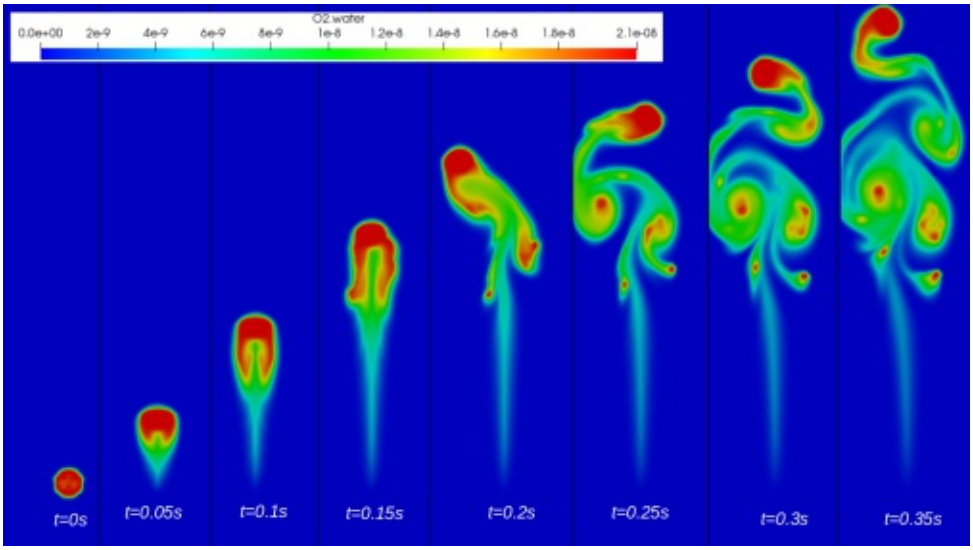


FIGURE 5.7: O2 Specie mass transfer simulation results: gradAlpha method

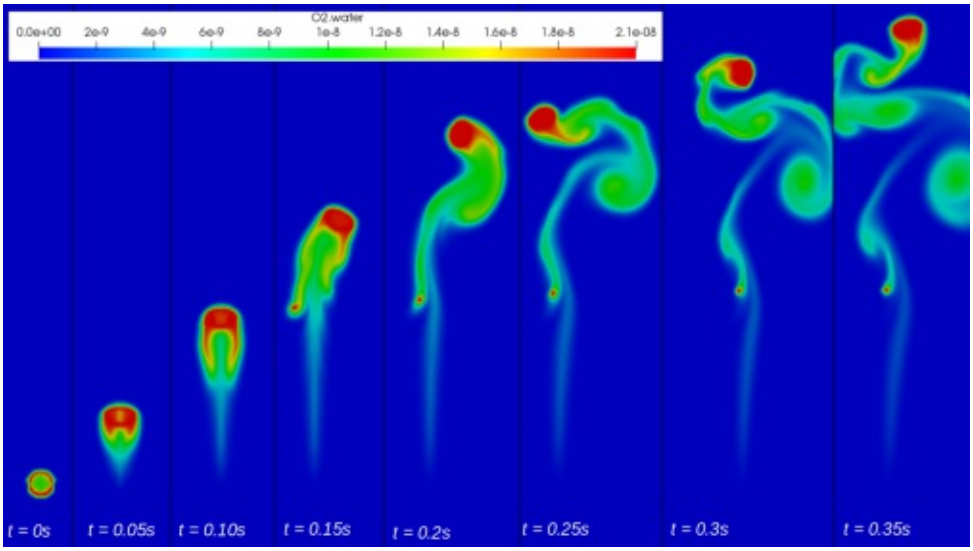


FIGURE 5.8: O2 Specie mass transfer simulation results: isoAlpha method

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

	isoAlpha	gradAlpha
Time	O2 mass transfer rate(kg/m3/s)	O2 mass transfer rate(kg/m3/s)
0.05	0.24	0.21
0.1	0.26	0.252
0.15	0.29	0.28
0.2	0.3	0.32
0.25	0.29	0.321
0.3	0.28	0.34
0.35	0.24	0.3
0.4	0.2	0.3
0.45	0.08	0.18
0.5	0.06	0.16

FIGURE 5.9: Mass transfer using isoAlpha and gradAlpha

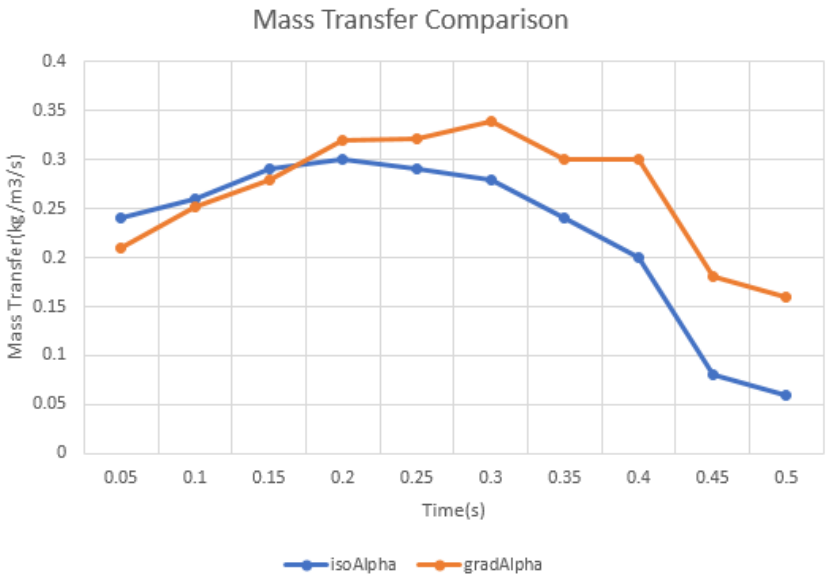


FIGURE 5.10: Averaged rate of mass-transfer plotted against timesteps

From the graphs, we notice that the rate of mass transfer in isoAlpha method is lesser than gradAlpha method. Hence we infer that the specie transfer happens only through the exact interface cells and the amount of pseudo mass transfer is reduced in the isoAlpha method. This can be better understood by comparing the mdot.O2 results from paraFoam.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

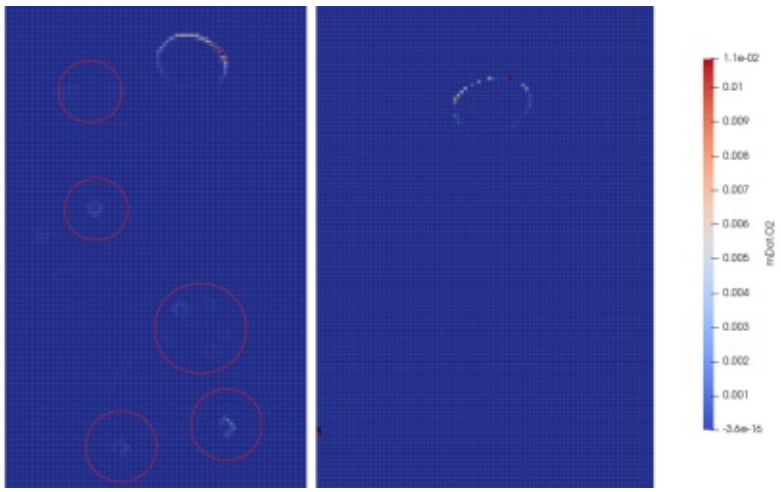


FIGURE 5.11: mdot.O2 mass transfer rate result at t = 0.25 s
gradAlpha result(left) and
isoAlpha result(right)

In the above picture, we notice that mass transfer happens through cells other than the rising bubble in the gradAlpha result(marked in red circles). For calculating the rate of mass transfer, we sum the mass transfer rate (mdot) from every cell in the domain. Hence in the gradAlpha result, we have pseudo mass transfer since the already accounted transferred mass is added again to the main transferred mass. In the case of isoAlpha, we do not have this pseudo mass transfer since we calculate mdot only from the interface cells as observed in the isoAlpha result in figure 5.11. The higher mass transfer rate in the isoAlpha result is accounted for the above-mentioned reasons.

Computational Time Comparison

The simulations are performed using an intel i7 4710HQ processor on a single thread. The time taken for isoAlpha and gradAlpha simulations are as follows:

Area model	**Simulation Time**	**Mesh cells**
isoAlpha	2 hrs 42 mins	80000
gradAlpha	1 hr 51 mins	80000

TABLE 5.1: Simulation time comparison

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Chapter 6

Conclusion

Geometrical interface area calculation library- isoAdvectorArea is implemented in the multiPhaseFoam solver. The original solver relies entirely on the gradient alpha method for interface area calculation. Results of the isoAdvectorArea library when compared with the values from the post processing utility-paraFoam shows that the method follows similar curves in estimating the interface area.

Even though the results look seemingly fine considering more accurate mean values, the oscillations in the area values is still an area which requires further analysis. In a preliminary analysis, this observation can be attributed to the mass transfer around the bubble. This causes the area calculation library to reconstruct a pseudo interface near the actual interface. So this forces the solver to consider those areas while calculating the mass transfer and interface advection (solution of $\alpha - equation$). Also, the current algorithm has its own innate inaccuracies. The interface reconstructed by the algorithm focuses less on the orientation of the interface. This could also be one of the possible reasons for the fluctuations in the interface area.

The research as a whole proves the possibility to implement geometrical methods in simulations involving mass transfer with higher accuracy. It should also be noted that the price of higher accuracy comes at higher computational cost which is quite the case in the current research.

Interface reconstruction is still an area of active research in science. As a next step, other geometrical interface reconstruction methods need to be attempted in order to rectify the drawbacks of the current algorithm. Other interface reconstruction algorithms include iso-RDF, PLIC, PLIC-RDF methods etc.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Bibliography

[1] TUHH. *DFG Schwerpunktprogramm 1740: Reaktive Blasenströmungen*. 2019. URL: http://www.dfg-spp1740.de/.

[2] Pedro Maximiano Raimundo. "Analysis and modelization of local hydrodynamics in bubble columns". In: *Chemical and Process Engineering* (2016).

[3] Dovydas Barauskas et al. "Selective Ultrasonic Gravimetric Sensors Based on Capacitive Micromachined Ultrasound Transducer Structure—A Review". In: *MDPI* (2020).

[4] C. W. Hirt and B. D. Nichols. "Volume of Fluid(VOF) Method for the Dynamics of Free Boundaries". In: *Journal of Computational Physics* (1979).

[5] F. H. Harlow and J. E. Welch. "The MAC Method: A Computing Technique for Solving Viscous, Incompressible, Transient Fluid Flow Problems Involving Free Surfaces". In: *The Physics of Fluids* (1966).

[6] Paul Woodward W. F. Noh. "SLIC(Simple Line Interface Calculation)". In: *Lecture Notes in Physics* (1976).

[7] D. L. Youngs. "Time-dependent Multi-material Flow with Large Fluid Distortion". In: *Numerical Methods for Fluid Dynamics* (1982).

[8] Ashley J. James Xiaofeng Yang. "Analytic relations for reconstructing piecewise linear interfaces in triangular and tetrahedral grids". In: *Journal of Computational Physics* (2006).

[9] William J. Rider and Douglas B. Kothe. "A computational method for sharp interface advection". In: *Journal of Computational Physics* (1997).

[10] Henrik Bredmose Johan Roenby and Hrvoje Jasak. "Reconstructing Volume Tracking". In: *Royal Society Open Science* (2016).

[11] Johan Roenby Henning Scheufler. "Accurate and efficient surface reconstruction from volume fraction data on general meshes". In: *Journal of Computational Physics* (2019).

[12] R.L Sherwood T.K Pigford and C.R Wilke. *Mass Transfer*. New York: McGraw-Hill Book Company, 1975.

[13] J. Dudley. "Mass transfer in bubble columns: A comparison of correlations". In: *Water Research* 29 (1995).

[14] Alexandre M Tartakovsky and Paul Meakin. "Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics". In: *Advances in Water Resources* (2006).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[15] Paul Meakin and Alexandre M Tartakovsky. "Modeling and simulation of pore- scale multiphase fluid flow and reactive transport in fractured and porous media". In: *Chemical Engineering Science* (2009).

[16] L. Raynal Y. Haroun D. Legendre. "Volume of fluid method for interfacial reactive mass transfer: Application to stable liquid film". In: *Chemical Engineering Science* (2010).

[17] Holger Marschall et al. "Numerical simulation of species transfer across fluid interfaces in free-surface flows using openfoam". In: *Chemical Engineering Science* (2012).

[18] Daniel Deising et al. "A unified single-field model framework for volume-of-fluid simulations of interfacial species transfer applied to bubbly flows". In: *Chemical Engineering Science* (2016).

[19] J.R. Thome M. Magnini B. Pulvirenti. "Numerical investigation of hydrodynamics and heat transfer of elongated bubbles during flow boiling in a microchannel". In: *International Journal of Heat and Mass Transfer* (2013).

[20] J.R. Thome M. Magnini B. Pulvirenti. "Numerical investigation of the influence of leading and sequential bubbles on slug flow boiling within a microchannel". In: *International Journal of Heat and Mass Transfer* (2013).

[21] R. Banerjee. "A Numerical Study of Combined Heat and Mass Transfer in an Inclined Channel Using the VOF Multiphase Model". In: *Numerical Heat Transfer, Part A: Applications: An International Journal of Computation and Methodology* (2007).

[22] D.B. Kothe W.J. Rider. "Reconstructing volume tracking". In: *Journal of Computational Physics* (1998).

[23] Victoria Timchenko Gim Yau Soh Guan Heng Yeoh. "An algorithm to calculate interfacial area for multiphase mass transfer through the volume-of-fluid method". In: *International Journal of Heat and Mass Transfer* (2016).

[24] Shanbi Peng, Qikun Chen, and Enbin Liu. "The role of computational fluid dynamics tools on investigation of pathogen transmission: Prevention and control". In: *Science of The Total Environment* 746 (2020), p. 142090. ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2020.142090. URL: http://www.sciencedirect.com/science/article/pii/S0048969720356199.

[25] Clayton T. Crowe. *Multiphase Flow Handbook*. Boca Raton, FL: Taylor & Francis, 2006.

[26] ANSYS. *ANSYS Fluent Theory Guide*. 275 Technology Drive Canonsburg, PA 15317, 2013.

[27] Rodolfo Piccioli Alex Read. "Introduction to Multiphase Flows". In: *STAR South East Asian Conference* (2015).

[28] Guerrero Et al. "Comparison between Eulerian and VOF models for two-phase flow assessment in vertical pipes". In: *Tecnología y Futuro* (2017).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[29] Mitra Javan Afshin Eghbalzadeh. "Comparison of Mixture and VOF Models for Numerical Simulation of Air–entrainment in Skimming Flow over Stepped Spillways". In: *International Conference on Modern Hydraulic Engineering* (2012).

[30] Lars-Andre Tokheim Sumudu S. Karunarathne. "Comparison of the Influence of Drag Models in CFD Simulation of Particle Mixing and Segregation in a Rotating Cylinder". In: *Proceedings of the 58th SIMS* (2017).

[31] Marguerite Graveleau. *PORE-SCALE SIMULATION OF MASS TRANSFER ACROSS IMMISCIBLE INTERFACES*. Stanford University: Master Thesis, 2015.

[32] Stephen Whitaker. "The method of volume averaging". In: *volume 13, Springer Science and Business Media* (1998).

[33] Ronald F. Gariepy Lawrence C. Evans. *Measure Theory and Finite Properties of Functions*. Taylor & Francis, 1992.

[34] D. Cappelli. "A detailed description of reactngTwoPhaseEulerFoam focussing on the links between mass and heat transfer at the phase interface". In: *Proceedings of CFD with OpenSource Software* (2018).

[35] Hrvoje Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Phd Thesis, University of Cambridge, 2003.

[36] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2012. ISBN: 9783642976513. URL: https://books.google.at/books?id=ye3tCAAAQBAJ.

[37] Ajit Kumar. *FVM implementation of NS Equation discretization in OpenFoam*. Shiv Nadar University, 2018.

[38] Santiago Márquez Damián. *An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces*. Phd Thesis, UNIVERSIDAD NACIONAL DEL LITORAL, 2013.

[39] S Márquez Damian. "Description and utilization of interFoam multiphase solver". In: *International Center for Computational Methods in Engineering* (2012).

[40] Henrik Rusche. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. Phd Thesis, Imperial College London, 2003.

[41] CFD-Online. *Velocity-pressure coupling*. 2005. URL: https://www.cfd-online.com/Wiki/Velocity-pressure_coupling.

[42] David LBook Jay P Boris. "Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works". In: *Journal of Computational Physics* (1973).

[43] CFD-Direct. *OpenFOAM V6 User Guide: 4.4 Numerical Schemes*. 2018. URL: https://cfd.direct/openfoam/user-guide/v6-fvschemes/.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN