BRNO UNIVERSITY OF TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING
HEAT TRANSFER AND FLUID FLOW LABORATORY

EFFECT OF FLOW PARAMETERS OF WATER AND AIR ATOMIZED SPRAYS ON COOLING INTENSITY OF HOT SURFACES

PhD THESIS

AUTHOR	Ing. JAN BOHÁČEK
SUPERVISOR	doc. Ing. JAROSLAV HORSKÝ, CSc.

BRNO 2011

ABSTRACT

The present thesis is focused on an overall description of water jets and air atomized jets for cooling purposes using CFD methods namely ANSYS FLUENT. It comprises two main parts – the micro and the macro model. The micro model concerns with a numerical description of single droplet dynamics whereas the macro model deals with a numerical modeling of water jets as complicated droplet structures emanating from solid stream nozzle and flat fan nozzle. By and large, it is based on multiphase models and User Defined Functions (UDFs), which represents the background of the present thesis. In most of cases, the presented numerical models were compared either with experimental data or another numerical model.

In the first part, the theory of each of three multiphase models is discussed. The first one, the Volume Of Fluid model (VOF), was used for simulation of single droplet dynamics designated as a micro model whilst last two multiphase models, the Euler-Euler model and the Euler-Lagrange model, were applied in the case of modeling of the entire water jet structure, which is contrarily designated as a macro model.

The micro model concerns with a numerical study of free-falling water droplet. For small droplet diameters (~100μm) the standard surface tension model (Continuum Surface Force model, CSF) was proved to cause significant unphysical parasitic currents. Therefore, the thesis is also devoted to surface tension as a source term of body forces imposed in momentum equation, normal, curvature calculation and related issues.

The macro model covers a numerical study of dynamics of the entire water jet structure i.e. the space between the nozzle exit and the wall where the jet impinges. It accounts for the complete geometry, for instance, support rolls, a slab and a mold bottom of a continuous caster.

Firstly, the physics of a solid jet impact onto a hot plate was simulated using both, the VOF and the Euler-Lagrange model. As regards the case with the VOF model, a model for film boiling was designed and tested.

Finally, both, the Euler-Euler model and the Euler-Lagrange model, were used for simulation of a flat jet horizontally spraying onto a hot slab inside a confined domain bounded by support rolls and a mold bottom. Concerning the simulation with the Euler-Euler model, a secondary breakup model was introduced based on the wave stability atomization theory. Concerning the Euler-Lagrange simulation, the dispersed phase (Lagrange particles) formed rather a continuous phase in some places, and therefore the coupling between Lagrange particles and the VOF model via UDFs was proposed.

KEYWORDS

Droplet, vapor layer, film boiling, jet atomization, solid jet, flat jet, cooling nozzles, user defined function, UDF, Volume of Fluid model, Lagrange particles, surface tension model, interface normal, curvature, Continuum Surface Force, HTC, Fluent, CFD.

BIBLIOGRAPHIC CITATION

BOHÁČEK, J. *Effect of Flow Parameters of Water and Air Atomizer Sprays on Cooling Intensity of Hot Surfaces*. Brno: Brno University of Technology, Faculty of Mechanical engineering, 2010. 102 pp., Supervisor of doctoral thesis: doc. Ing. Jaroslav Horský, CSc.

DECLARATION

I declare that the thesis hereby submitted for the Philosophiae Doctor degree at Brno University of Technology is my own work under supervision of doc. Ing. Jaroslav Horský, CSc., has not been previously submitted at any other University for any degree and all sources were quoted completely and correctly.

Place : ......Brno.............
Date : ....

....
Author’s signature

ACKNOWLEDGEMENTS

Herein, I would like to express my sincere gratitude to my supervisor Dr. Jaroslav Horský for his valuable advices and helpful attitude. I am also indebted to Prof. Miroslav Raudenský for his keen interest and deadline reminders. I would like to extend my thanks to Dr. Pavel Rudolf for his ideas and suggestions in the field of CFD. Further, I thank Dr. Michal Pohanka for his mathematical remarks, hardware, and software support. I am also very grateful to my parents, grandparents, my sister, and my girlfriend for their encouragement and essential support.

DEDICATION

This piece of work is dedicated to my grandfather František Boháček, who passed away in 1996. He was a very skillful man, extremely talented in manual work. He worked with his father at their own locksmithery, to where other companies and schools used to send their apprentices for trainings. After World War II, when the Socialist system began to rule, his business was naturally stopped; he was placed as an in-line worker and step by step mentally consumed. Finally, he found the piece within his family and writing poems. Despite many unpleasant life twists he was always a joyful and a happy person so that the entire hometown was proud of him.

CONTENT

1 Introduction

By the word primary steel processing, we mostly understand continuous casting, centrifugal casting but also hot and cold rolling. Heat transfer is the most characteristic and prevailing process for all of them, which may be attended by other side processes such as solidification, mechanical deformation, and oxidation and so on. To ensure perfect quality of final products the cooling control is naturally desirable. It requires the cooling intensity to be optimized. In majority of cases the needed cooling is provided by arrays of nozzles in that the operating fluid is the most frequently water but also mixture of both, air and water, and last but not least different emulsions supporting better lubrication of working surfaces. In particular, it is the cooling intensity that is responsible for the final quality of product, and thus the cooling system should be reasonably designed in terms of not only suitable cooling but also energy savings. Essentially, four different ways can be used to design a cooling system. Designers often rely on their own experiences and proceed intuitively. Another way is the using of some of plenty correlations that were put together based on experimental data and allow to calculate heat transfer coefficient as a function of flow parameters. However, those correlations are always restricted to a certain range of selected parameters, hardly ever account for more than one nozzle or even curved surfaces. The most sophisticated and precise method is seemingly a laboratory experiment and the consequent inverse task, which is capable of the thermal boundary condition reconstruction. Performing of experiments is however economically and energy demanding; therefore, it gives a chance to Computational Fluid Dynamics that can simulate fields of velocities, pressures, temperatures numerically provided boundary conditions are correctly defined. It is CFD that is the background of this thesis and hopefully brings new pieces of knowledge into cooling problems solved numerically using CFD software.

1.1 General description of issues to be solved

Imagine a hot steel plate of the thickness in order of millimeters or centimeters with one side exposed to a spraying jet that involves a very intensive cooling. Thermal boundary condition in the jet footprint and surroundings is naturally unknown, time dependent and changes along with surface temperature, flow parameters and physical properties of the operating fluid. On the other hand, other thermal boundary conditions around are either usually known or can be easily defined considering some simplifying assumptions e.g. an adiabatic wall etc.

The computational domain must be extended to fluid region i.e. the region where the jet is spraying, so that the heat transfer can be solved using CFD methods. The extension must be sufficient enough in order to define accurate boundary conditions.

In solid region it is only the equation of unsteady heat conduction to be solved whereas in fluid region it is Navier-Stokes equation modified for two phase flow that has to be primarily solved along with continuity equation. Further, there are energy equation and equations for turbulent properties in fluid region. Fluid region, which is described by several partial differential equations, is obviously more complicated in terms of numerical schemes than solid region described only by a single diffusion equation. According to underlying physics and flow nature the most suitable multiphase model can be selected that is however still rather general and has to be concretized via extra subroutines. Fundamentals of multiphase models are briefly discussed in below mentioned chapters. The computational model can further become more complicated when phase changes occur that may have an impact on solution stability since they play as source terms generally in most of foregoing equations.

1.2 Survey on current state of knowledge in field of numerical simulations related to heat transfer and multiphase flows

First of all, the problem has to be defined. The main interest and objective of this study is to come up with some practical numerical approaches that can be used for simulation of spray cooling. However, before proceeding to generalization of the problem into e.g. Lagrange or Euler particles, it should be noted that the underlying physics from the micro scale point of view should be well understood. For these ‘micro-scale’ problems several numerical approaches can be used to capture nature of sharp interface between the liquid and the ambient gas. Later, three multiphase models are discussed (Euler-Euler, Euler-Lagrange, Volume of Fluid model). There are basically several main differences among interface tracking methods i.e. methods that simulate two or more immiscible liquids. Numerical schemes are either applied on a staggered grid (Euler grid) in that nodes do not change positions (Volume of Fluid etc.) or a Lagrange grid (level set method) that is free to deform in the whole domain along with free-surface changes. Lagrangian methods have advantage over Eulerian methods so that it explicitly calculates the position of interface, whereas in the case of Eulerian methods the interface is reconstructed from volume fractions. On the contrary Lagrangian methods have significant problems with mass conservation compared with Eulerian methods.

Here, a short survey on mainly single droplet problems solved numerically is given. Following paper summaries of other authors give an overview firstly on physical issues related to free surface flows solved mostly using VOF method and secondly on numerical modeling of entire sprays structure.
As shown later in next chapters, the Volume of Fluid model is most likely the most frequently used interface tracking scheme; however, it generally suffers from the smearing of interface. In Volume of Fluid model the volume fractions of the secondary phase are advected using a scalar transport equation. The non-linear term can be discretized by several schemes. The paper by Waclawczyk [1] concerns with the effect of Courant number (CFL) value on smearing of interface between two immiscible liquids. CICSAM and HRIC discretization schemes were considered. The first one depends on Courant number implicitly and was found to give more precise results than HRIC for CFL < 0.5, whereas HRIC showed to preserved better interface for CFL > 0.5. For VOF calculations in general the author claims that CFL should be always chosen smaller than CFL < 0.5. Hokr [2] studied an effect of geometry on numerical diffusion. Besides other things, he found out that the value of Courant number ranging between 0 and 1 still strongly influences the numerical diffusion especially for perturbed meshes. For VOF calculations the recommended value of Courant number should lie below 0.5. Butler [3] made a big progress when he simulated a 2D droplet with/without gravity using lattice Boltzmann method (LBM). Unlike other conventional multi-phase models, LBM does not calculate the motion of ambient gas. It also employs staggered grid, but it rather mark cells as empty those with the ambient gas. Different wall boundary conditions exist for fluid and interface cells. In this paper, effects of surface tension were not taken into account and also the role of viscous forces was neglected. Doctoral thesis by Ubbink [4] concerns a methodology capable of predicting the topology between to immiscible fluids on an arbitrary Eulerian mesh. The two fluids are modeled as a single continuum with a fluid property jump at the interface. A volume fraction is used to identify fluids or the interface. A new high resolution differencing scheme was developed to keep the transitional area between the two fluids restricted to one cell width and also to satisfy the conservation of the flow properties at all times. Štrubelj [5] successfully attempted to develop an interface

sharpening method for a standard two-fluid mathematical model. His sharpening method is based on conservative level set method. He tested his model on Rayleigh-Taylor instability case in that the more dense fluid was immersing into a lighter phase due to gravity. For future work he proposed model in that only large interfaces would be sharpened. Several researches attempted to couple the mass conservative Volume of Fluid method with Level Set Method in order to handle more precisely with surface tension dominant flows and provide more accurate information about the interface position. A good paper on this topic was presented by Shepel [6], who implemented successfully the Level Set Method in the commercial system CFX 4 and also FIDAP, which removes the gaseous phase from consideration. Models were tested on the broken-dam problem and the collapsing cylinder of water. Having of sharp interface and knowledge of interface normal, curvatures are also very important in order to apply surface forces such as surface tension. Afkhami [7] concerned with a 3D 2 mm single droplet impinging upon the inclined flat surface (45°) with velocity of 1 m/s. For this purposes the Volume of Fluid method was used to track interface. Height Function approach was used for the calculation of second order accurate curvatures and surface normal. The precise definition of interface allowed for more accurate simulation of surface tension effects. The model also differed between the advancing and the receding angle.

Several papers came out with simulations of single droplet impact onto hot surfaces. It is not always only the Volume of Fluid model that is used to track the interface between phases. The paper by Pasandideh-Fard [8] presents a study on impact of water droplets onto a hot stainless steel surface within temperature range (50-120°C). Even for temperatures around 120°C the boiling did not occur, so it was not considered within simulations. The interface was tracked using mass conservative VOF method. The main objective of study was to simulate heat transfer coefficient distribution along radial coordinate vs. time. The velocity varied from 0.5 to 4 m/s and the range of droplet diameters was (0.1-2.0 mm). It was shown that the impact velocity only slightly enhances the cooling intensity within tested range of velocities. Francois [9] in her Doctoral thesis deals with a design of a micro-scale cooling design using droplets as an operating fluid. It is a detailed numerical study on droplet impact utilizing the immersed boundary method. Droplet spreading parameters and heat flux from the wall are primarily studied. Effect of grid size and most of physical properties of droplet on droplet spreading is discussed. The film boiling and the transition from the nucleate boiling to the film boiling after n-heptane droplet were successfully studied in paper by Harvie [10]. A new model named Bounce was developed that is composed of the Volume of Fluid Model and a one-dimensional algorithm used to calculate flow within the vapor layer. Results were compared with photographs from experiments. The model was also tested for the nucleate boiling regime, it however failed to predict both, the realistic droplet spreading and heat fluxes. It was caused by the fact that there is a partial contact between the droplet and the surface during the nucleate boiling. However, the Bounce model considered the vapor layer to be everywhere. Another interesting work on the film boiling can be found in Ge's paper [11] in that the 3.8 mm droplet impinges with 1 m/s onto hot surface (200°C). The level-set function was used to identify interface within staggered 3D grid. The surface tension was taken into account via the Continuum Surface model. The droplet dynamics was simulated in 3D whereas the flow within vapor layer was simulated in two dimensions. Results showed and confirmed peaks of HTC near the moving contact line formerly presented by several authors. One of fundamental papers on film boiling is the paper by Welch and Wilson [12] in that the VOF model with surface tension model was used for simulation of the rising bubble. The interface was reconstructed using linear segments, temperature gradients were calculated within each phase and heat fluxes for corresponding thermal conductivities were determined consequently. The mass sources for vapor fluid were determined based on temperature gradients across interface.

The aforementioned papers dealt with the heat transfer from a surface to a water droplet bulk. The following summaries are given on papers concerning with evaporation from the free surface. Frackowiak [13] in 2nd International Symposium on Non Equilibrium Processes presented a model for evaporation from droplet surfaces for high dense droplet loaded flows. He considered a coupling between the macro model of external aerothermic field and the micro model simulating flow inside the nozzle. Each droplet in the macro model is considered as a sphere with a constant radius and position dependent vapor flow rate with radial and tangential velocities. Tangential velocities are determined from the simulation of flow inside the droplet whereas the radial velocities are determined based on vapor theory. The shear stresses on interface and heat flux distribution is deduced from the calculation of the external field. Yuan [14] studied natural convection and forced convection film boiling around a stagnant and a moving sphere, respectively. The Volume of Fluid Method based on piecewise-linear interface reconstruction was used to track the interface. However, it was modified in the way that a double staggered grid was used instead of a single staggered grid. He claimed that a collocated grid arrangement is not suitable for interfacial flows, which is the case of film boiling on droplets. Velocities are stored in face centers whereas pressure is stored in the cell center. The second grid is so shifted that it has its corners in cell centers of the first grid. Pressures are than stored in corners of the first grid. The basic idea is that if velocities of the first grid are parallel to faces i.e. there is no mass flux through this cell; there is still contribution of mass flux into the second grid. The evaporation model for droplets at low Weber numbers depositing on hot surface was numerically studied in excellent paper by Strotos [15]. Wall temperatures were considered up to 100°C, and thus the evaporation took place only at droplet free surface. The Volume of Fluid model was coupled with Fick’s law that uses the local vapor concentration as a driving force for evaporation. Also model by Spalding was tested; however, it requires certain knowledge of flow conditions around droplet and reference length, thus Fick’s model should be used preferably. Cao [16] studied an effect of vapor layer on drag coefficient in both, laminar or turbulent regime, around the droplet falling. The laminar regime is found around the leading edge while the turbulent regime is induced in the wake. Several correlations for estimation of drag coefficient were established and the drag coefficient was shown to decrease when the vapor layer is formed. A rather theoretical contribution on the evaporation of droplets can be found in Gubarev’s paper [17]. The problem was solved using modified equation of heat conduction for two phase mixture in that the droplet stood as a heat sink. The range (20–100 μm) of droplet diameters was tested with the initial temperature of 10°C. The surface temperature of the hot plate was 1000°C. Different droplet-to-surface distances were calculated (0.1–0.4 mm) and also other information such as the time necessary for the complete evaporation of droplets, the time of heating droplets, and vapor layer thickness were evaluated.

When the droplet impinges onto the flat surface, the dynamics of the subsequent spreading is driven by the advancing and the receding angle of the moving contact line. The droplet impact (diameter of 2.5 mm), the spreading, the recoiling, and also the rebounding process were studied in paper by Gunjal [18]. Also in this work the VOF technique was employed considering surface tension effects via Continuum Surface Force model. A range of Reynolds numbers (550-2500) was studied. Oscillation process of droplet calculated using numerical procedures however did not agree with experimental data. Different contact angles were tested to study the wetting of Teflon and glass surfaces. Ganesan [19] modeled a droplet spreading on smooth flat surface considering a dynamic contact angle. The Arbitrary

Lagrangian Eulerian approach, which advects the interface explicitly, was used to solve the problem. Either the advancing and the receding contact angle or the equilibrium contact angle can be input. Instead of no-slip or free-slip boundary condition, the slip with defined friction was used. Other very detailed information on moving contact lines can be found in papers by Hocking [20], Cox [21]. Valuable information about dynamic contact angles of plenty of liquids can be also found in the thesis by Renabothu [22]. Kandlikar [23] studied the dependency of the dynamic contact angle on the increasing temperature of the surface. For example, in the case of water droplet impinging onto stainless steel the behavior of the dynamic contact angle is as follows. Below temperatures 140–150 °C, the advancing contact angle is approximately 125° whereas the receding contact angle is around 50°. When this temperature range is reached, a jump in the receding angle is observed to the same value as the advancing contact angle and the transition to the boiling regime is observed. Lunkad [24] in his paper presents VOF simulations of a droplet impact on an inclined or a horizontal surface. Droplet diameters were ranging from 1.8 to 3.3 mm and impact velocities were in range of 1.00 – 3.25 m/s. The standard VOF model in Fluent along with Continuous Surface Force was used. Each simulation was performed with both, the static and the dynamic contact angle. Dynamic contact angle was adjusted based on experimental data and a User Defined Function was employed to define this time dependent contact angle. The spreading and sliding regimes of droplets on inclined surfaces were predicted well using the static angle. The regimes of splash and rebound were however not.

All aforementioned simulations were done either for low or medium Reynolds numbers. Several authors published results on high speed droplet impacts. Interesting results on high speed water droplet impact can be found in paper by Haller [25] in that the droplet impact is simulated within Front Tracking Method using two separate grids. The first one is a staggered Eulerian grid and the second Lagrangian grid is tracked with wave fronts. Results show a prompt jet spreading immediately after the first droplet contact. Results further confirm Heymann’s previous proposals on the maximal impact pressure near the contact line. It was also shown that the droplets cavitates in some portions of its bulk. A high speed droplet impact (186 m/s) on hot surface (600 K) was simulated using the Lagrange algorithm with moving unstructured triangular mesh coupled with a vapor layer model. The model is capable of the radial jet spreading prediction and also calculates the wave propagation inside the droplet bulk. The model is suitable for simulating high-speed impacts on either cold or hot surfaces. A high speed water droplet impact (305 m/s, droplet diameter of 2 mm) was solved by the finite element solver Dyna3D in paper by Adler [26]. The objectives were to study mechanistic principles of rain erosion in aeronautics. Therefore, only droplet dynamics was simulated and also deformations of substrate were considered. On the other hand, the flow of the ambient gas was not solved and also any heat transfer was not taken into account. The computational grid was dynamically changing its position and the boundary condition in each surface node was updated. This approach with deforming meshes is not suitable for highly distorted meshes, moreover, when there is no remeshing employed.

The whole aforementioned overview summarized numerical methods for modeling of free surface flows. Different physical problems were studied such as the advancing and the receding angle at the contact line, evaporation, film boiling, distribution of heat transfer coefficient, wave propagation, impact pressure etc. In the following lines there are several papers summaries given about numerical modeling of the whole jet, the spray pattern.
Cho’s paper [27] covers the cold flow modeling in the Runout Table in Hot Strip Mill, verified by experimental data. The water pattern on the moving steel strip was solved using the VOF model and pressure distribution on the strip wall was evaluated. Simulation of the

cold flow gave satisfactory results when compared with experimental data. However, heat transfer was not simulated. Also in other paper by Kulju [28] the VOF model was used to simulate the entire jet; however, it was the submerged jet that was studied and thus it was far from modeling of complicated water structure. The submerged jet of diameter of 20 mm was studied in film boiling regime. The surface temperatures were ranging from 400K to 1300K. The VOF model was enhanced by source terms for the formation of vapor. His simulations gave good results. Right underneath the jet, no vapor film occurred due to jet dynamics. Several diameters further the constant vapor film was observed until it started breaking up into vapor bubbles.

In Narumenchi's paper [29], the nucleate boiling within submerged jets impinging onto a hot surface (120°C), with R134a and boiling water as an operating fluid was solved using CFD code Fluent using the Euler-Euler model modified in order to account for the bubble generation, bubble departure frequency, the heat and the momentum transfer, and also the dissipation of turbulent properties. It should be noted that free stream temperature of the bulk corresponded to saturation temperature, and thus the model did not simulate bubble collapse.

In most of papers authors however inclined to simplify complicated jet structures and to use Lagrange particles instead. The modeling of jets started with optimization of diesel sprays a long time ago. Mostly the well-known Lagrange based code KIVA was used for modeling of evaporating dispersed jets and it developed into the powerful tool, which was later modified and used as an optional model in new incoming FVM based codes like FLUENT, STAR-CD, CFX etc. For example the paper by Senda [30] deals with a process of diesel spray impinging on a flat wall with a high temperature. KIVA original code was used and modified to account for a dispersion process on the wall, a breakup of impinging droplet, and also vaporization of droplets with the temperature above the saturation temperature. Note that diesel jets are diametrically different compared to cooling jets and thus their behavior must naturally differ consequently. The main differences are the feeding pressure, the orifice diameters, and physical properties of operating fluid.

Similar scenario can be found in paper by Grover [31] who also stuck to KIVA code. In this case the evaporative solid cone spray impinging normally onto a flat plate was solved and results were verified experimentally. The model was modified in order to improve prediction of the spreading rate of liquid and vapor phase and also to account for viscous dissipation at high Weber numbers. The wall boundary condition was based on single droplet computational studies and considered one wall film parcel and four splashed droplets. This model was shown to give better results than the model by Senda.

Yao [32] issued a valuable report on a transport phenomenon of small droplets <<50 μm using both, experimental and numerical technique. Although the main objective of the work is to contribute into the field of fire suppression, it brings important information on how a bulk air flow affects momentum of droplet mainly around obstacles in flow. Droplet segregation, aggregation is numerically studied using Euler-Lagrange approach.

In Tonini's paper [33] the Euler-Lagrange model was developed to simulate dense sprays in diesel jet applications. When the dispersed phase loading is low enough usually less than 10% of cell volume, than the standard Euler-Lagrange model can be employed. The basic idea in paper is to calculate a volume fraction of the dispersed phase in the current cell. If the total volume of dispersed phase is higher than the current cell volume, the volume fraction of the current phase is set to 1 and the rest of dispersed phase volume is assigned to liquid fractions in surrounding cells based on the parcel-to-cell relative distance. Afterwards, in between Euler time steps cell-virtual variables are updated using mass, momentum, species, and energy source terms. Different mesh densities even with dynamic refinement were employed in order to test mesh sensitivity and convergence behavior.

Bhattacharya [34] developed the analytical model for the evaporation process during Ultra-Fast Cooling in Runout Table of Hot Strip Mill based on simulations using Discrete Phase Model (DPM). The aim was to reveal whether spray evaporating cooling has a sufficient potential to achieve the remarkably high strip cooling rate (300 °C/s), in lieu of conventional laminar jet impingement cooling. DPM model was successfully modified, so that there was only vapor presented in first fluid layer on the strip surface and its temperature was kept at strip temperature throughout the evaporating process. Only boiling law was activated as soon as the Lagrange droplet hit the strip. The high cooling rate of 300°/s could become real with droplet diameters less than 70 μm. It should be noted that the cooling rate was not obtained from simulation comprising spraying jets in the whole domain. The cooling rate was rather derived from DPM calculation in that only one single droplet was floating on the hot surface until it was completely vaporized. Then a correlation was developed to estimate the cooling intensity.

2 Multiphase models in CFD

2.1 Numerical modeling of interfacial flows, Volume of Fluid method

So far several interface-tracking procedures were published in papers and most of them were developed to deal with a specific problem. Some methods were due to their robustness and versatility implemented into commercial software. Among these methods one could name e.g. the front-tracking method [35], the boundary integral method [36], the phase-field method [37], the Second Gradient method [38], the Level Set Method [39], [40], the Volume of Fluid (VOF) method [41], [42].
According to the published papers, the most often used interface tracking procedure is evidently the Volume of Fluid method (VOF). Therefore, it is briefly discussed here.

The volume of fluid method (VOF) is based on a function F whose value is unity at any point occupied by fluid e.g. water and zero otherwise e.g. air [41], [42]. The VOF model does not allow for void regions where no fluid of any type is present. The average value of F in a computational cell represents the fractional volume of the cell occupied by the fluid, while a zero value indicates that the cell contains no fluid. Cells with F values between zero and one must then contain free boundary.

$$F = \frac{V_{sp}}{V_{cell}} \qquad \text{Eq.1}$$

In addition to defining which cells contain a boundary, the VOF method defines where fluid is located in a boundary cell. The normal direction to the boundary lies in the direction in which the value of F changes most rapidly. Although F is a step function, its derivatives are computed in a special way. Finally, knowing both the normal direction and the value of F in a boundary cell, a line cutting the cell can be constructed that approximates the interface there. This boundary location can then be used in setting boundary conditions. In addition, surface curvatures can be computed from the F distribution when surface tension force must be considered.
The time dependence of F is governed by a continuity equation

$$\frac{\partial}{\partial t}(F\rho_l) + \frac{\partial}{\partial x_i}(F\rho_l u_i) = \dot{m}_{gl} - \dot{m}_{\lg} + S_F\,, \qquad \text{Eq.2}$$

where $\dot{m}_{gl}$ is the mass transfer from secondary (liquid) phase *l* to primary (gas) phase *g*, $\dot{m}_{\lg}$ is the mass transfer from primary phase *g* to secondary phase *l*. S_F denotes a source term, which normally equals zero.
The volume fraction equation is not solved for the primary phase (gas, air). The primary phase volume fraction is difference between unity and the fractional volume *F* occupied by the secondary phase (liquid, water).
A single momentum equation is solved throughout the domain, and the resulting velocity field is shared among the phases. The momentum equation is depicted below.

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i}{\partial x_j} u_j = A_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \qquad \text{Eq.3}$$

It must be noted that the momentum equation is dependent on the volume fractions of present phases through the properties ρ and μ. These are then derived from the knowledge of the fractional volume F and they have the form as following:

$$\rho = \rho_l F + (1-F)\rho_g \qquad \text{Eq.4}$$

$$\mu = \mu_l F + (1-F)\mu_g \qquad \text{Eq.5}$$

This mass weighted definition of material properties results from mass conservation and surely should be more precise than e.g. the volume weighted definition. On the other hand, the expression for mixture viscosity is an approximation. There can be found different approaches in literature that are discussed later [6].

2.1.1 On stability condition – time step size

Firstly, the time step must fulfill the CFL condition due to the convective terms of the Navier–Stokes equations. This convective stability condition is given by the following formulation:

$$CFL > \frac{c \cdot \Delta t_c}{\Delta x} \qquad \text{Eq.6}$$

Generally, CFL should be less than unity. However, for VOF calculations it was shown that CFL should be rather below the value of 0.5.

Secondly, the explicit discretization of the surface tension term induces another restrictive condition. This surface tension induced stability condition ensures that the capillary waves on free surface will be not amplified. Most of numerical models employing the CSF approach use the formulation by Brackbill [43] which is given as the following:

$$\Delta t_\rho \sim \sqrt{c_1 \frac{\rho}{\sigma} \Delta x^3} \qquad \text{Eq.7}$$

The Brackbill's stability condition is linked with the density; however, it does not take into account the viscous term. When the inertia phenomenon is neglected the capillary time step is defined as follows:

$$\Delta t_\eta \sim \mathrm{c}_2 \frac{\eta}{\sigma} \Delta x \qquad \text{Eq.8}$$

The combination of Eq.7 and Eq.8 comprising constants c1 and c2 formulates the capillary time step Δt_σ as follows:

$$\Delta t_\sigma = 0.5\left(\Delta t_\eta + \sqrt{\Delta t_\eta^2 + 4\Delta t_\rho^2}\right) \qquad \text{Eq.9}$$

The computational time step Δt is then the minimum from both, the convective time step Δt_c and the capillary time step Δt_σ.

$$\Delta t \le \min\left(\Delta t_c, \Delta t_\sigma\right) \qquad \text{Eq.10}$$

In [44], only low and medium Reynolds numbers were considered, thus, the only constant c2 was crucial. The damping of capillary waves was observed for the constant c2 equaled to 4. Although authors noted that it depends on the algorithms used.

2.1.2 Reconstruction of interface from fractional volume F

Two immiscible fluids are separated by the interface between them. Since the interface is not explicitly calculated, it has to be rather reconstructed from volume fractions available within each cell.

The geometry reconstruction is one possible way how to identify the interface. First reconstruction scheme so-called Donor-Acceptor scheme uses standard interpolation schemes such as the upwind scheme to calculate face fluxes through cell faces, but a special procedure is used to prevent numerical diffusion of the interface. Donor cell offers a certain portion o liquid to the Acceptor cell, thus that the amount of liquid is limited by the filled volume in the Donor cell and by the free volume in the Acceptor cell. Another more sophisticated scheme is the piecewise-linear in that the interface is represented as a linear segment in 2D and a planar segment in 3D. This scheme preserves the thickness of the interface within one cell thickness; however, in some cases it suffers from a poor convergence.

Apart from geometry reconstruction schemes there are different procedures for the capturing of the interface position. Unlike geometric interface reconstruction methods, high-resolution schemes do not introduce geometric representation of the interface. The interface position is captured using high-resolution advection schemes CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) [4] and HRIC (High Resolution Interface Capturing Scheme) [46]. Both are based on the normalized variable diagram NVD [47]. High resolution schemes were built to assure lack of the numerical diffusion and compressive character i.e. sharpening of the step interface profile.

NVD is based on the convective boundedness criterion (CBC) that states that the variable distribution between the centers of the neighborhood control volumes remain smooth.

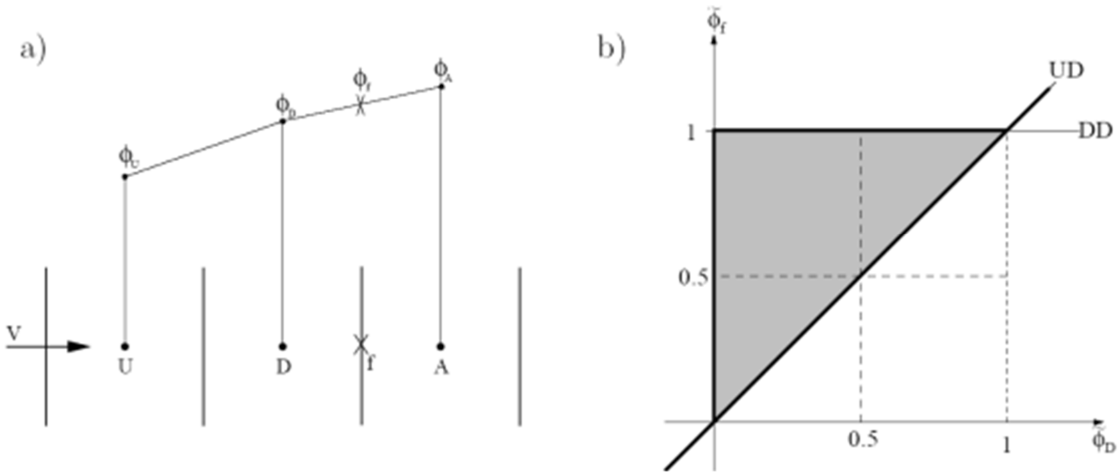


Fig. 1 *a) Boundedness criterion, U upwind, D donor, A acceptor cells, b) NVD, shaded region indicates where CBC is satisfied*

$$\tilde{\Phi}_f = \frac{\Phi_f - \Phi_U}{\Phi_A - \Phi_U} \qquad \text{Eq.11}$$

$$\tilde{\Phi}_D = \frac{\Phi_D - \Phi_U}{\Phi_A - \Phi_U} \qquad \text{Eq.12}$$

CICSAM is the combination of the HYPER-C scheme [45] and the ULTIMATE-QUCKEST scheme [47]. Moreover, there is added another assumption about the dependence of the region where the CBC is satisfied on the CFL condition. CFL condition is included implicitly. The HYPER-C scheme employs downwind differencing scheme, while the ULTIMATE-QUICKEST scheme employs the third order accurate QUICK. Whether the first one or the second is used, it depends on the angle between unit vector normal to the interface and the unit vector parallel to the line between centers of the donor D and acceptor A cells. When interface position is normal to the direction of the flow so called blending factor γf is set to unity and HYPER-C scheme is used. In the case of tangential orientation of the interface γf equals 0 and ULTIMATE-QUICKEST scheme is employed.

$$\tilde{\Phi}_f = \gamma_f \tilde{\Phi}_{CBC} + (1-\gamma_f)\tilde{\Phi}_{fUQ}, \quad 0 \le \gamma_f \le 1 \qquad \text{Eq.13}$$

Similarly to CICSAM, HRIC scheme is also based on the NVD, however, CFL condition is defined explicitly. Again the blending factor γf is introduced, to switch smoothly between the DDS and UDS schemes.

$$\tilde{\Phi}_f^* = \gamma_f \tilde{\Phi}_{DDS} + (1-\gamma_f)\tilde{\Phi}_{UDS}, \quad 0 \le \gamma_f \le 1 \qquad \text{Eq.14}$$

Blending of the UDS and the DDS schemes is dynamic and $\tilde{\Phi}_f^*$ is corrected with respect to the local Courant number. The goal of this correction is to force continuous switching between schemes also in time domain.

One can notice that main difference between the CICSAM and the HRIC are the order of accuracy of the component schemes. However, each of them is just first order accurate in complete formulation. Since the HRIC dependence on the CFL is explicit, it should be rationally less sensitive on the value of the cell Courant number.

Waclawczyk [1] showed that the cell Courant number should be chosen smaller than 0.5 no matter which scheme is used.

2.1.3 Level Set Method as alternative for interface sharpening within VOF model

In previous chapter different procedures for the interface reconstruction were discussed. In some cases the sharpness of interface cannot be maintained using aforementioned schemes. Therefore, other suitable procedures must be employed. The one of prospective ways is the coupling with Level Set Method (LSM) that is actually a stand-alone interface tracking method that explicitly calculates the interface position. Here, the LSM fundamentals are introduced.

In the Level Set Method, the interface between the two phases is represented by a continuous scalar function $\Phi(\mathbf{x},t)$, which is set to zero on interface, positive on one side, and negative on the other. The function $\Phi(\mathbf{x},t)$ is called level set function and is typically defined as the signed distance to the interface. When the interface is advected by the flow, the evolution of the level set function is given by

$$\frac{\partial \Phi}{\partial t} + \frac{\partial \Phi}{\partial x_i} u_i = 0 \qquad \text{Eq.15}$$

It should be noted that the Level Set Method is traditionally solved in non-conservative form. Since it is not keeping with conservative FVM method used in this work, it will be further not discussed. The conservative form of the fluid flow and level set interface-tracking equations have been used and validated e.g. by Sussman and Puckett [48] in the coupled level set-VOF method.

When solving the advection equation for $\Phi(\mathbf{x},t)$, the level set function $\Phi(\mathbf{x},t)$ stops to be the signed distance from the interface, even if it is properly initialized in time $t = 0$ sec. Consequently, both level set function needs to be reinitialized regularly, preferably at every time step. An efficient method to do this was proposed by Sussman and Fatemi [49], and is based on solving for the steady-state solution of the following equation

$$\frac{\partial \Phi}{\partial \tau} + sign(\Phi_0)\left(\left|\frac{\partial \Phi}{\partial x_i}\right| - 1\right) = \lambda \delta_\varepsilon(\Phi)\left|\frac{\partial \Phi}{\partial x_i}\right| , \qquad \text{Eq.16}$$

where τ is a timelike variable, thus, it is different from the physical time t. Φ_0 is the initial distribution of the level set function before reinitialization, and λ is a correction coefficient calculated in such a way as to ensure mass conservation.

In the interfacial flows, where two fluids are present, the density and viscosity are typically interpolated across the interface as follows:

$$\rho = \rho_l + (\rho_g - \rho_l) H_\varepsilon(\Phi) \qquad \text{Eq.17}$$

$$\mu = \mu_l + (\mu_g - \mu_l) H_\varepsilon(\Phi), \qquad \text{Eq.18}$$

where $H_\varepsilon(\Phi)$ is a smoothed Heaviside function and is expressed by the following formulas [50]:

$$H_\varepsilon(\Phi) = \begin{cases} 0 \quad if \quad \Phi < -\varepsilon \\ (\Phi + \varepsilon)/(2\varepsilon) + \sin(\pi\Phi/\varepsilon)/(2\pi) \quad if \quad |\Phi| \le \varepsilon \\ 1 \quad if \quad \Phi > \varepsilon \end{cases}, \qquad \text{Eq.19}$$

where ε is a small parameter of the order of the size of a mesh cell close to the interface.

2.1.4 Modeling of surface tension

Surface tension σ is an attractive property of the surface of a liquid. It causes the surface portion of liquid to be attracted to another surface, such as that of another portion of liquid. It has the dimension of force per unit length [N/m], or of energy per unit area [51].

Surface tension effects do not mostly play an important role in interfacial flows, however, it should be always determined based on the value of two dimensionless quantities: the Reynolds number *Re* and the capillary number *Ca*, or the Reynolds number *Re* and the Weber number *We*.

For Re<<1, the quantity of interest is the capillary number:

$$Ca = \frac{\mu U}{\sigma} \qquad \text{Eq.20}$$

where U is the free-stream velocity. Surface tension effect can be neglected if Ca >>1. For Re>>1, the quantity of interest is the Weber number:

$$We = \frac{\rho D U^2}{\sigma} \qquad \text{Eq.21}$$

Surface tension effect can be neglected if We >>1 [52].

The first usable surface tension model developed for Volume of Fluid method, Level Set Method and Front Tracking method (FT). It is the continuum surface force method (CSF) designed by Brackbill et al [43]. The force at the surface is expressed as a volume force using the divergence theorem. It is then applied as a source term to the momentum equation. It has the following form:

$$F_{s-t} = \sigma \frac{\rho \kappa \frac{\partial F}{\partial x_i}}{0.5\left(\rho_g + \rho_l\right)} \qquad \text{Eq.22}$$

where κ is the surface curvature, defined in terms of the divergence of the unit normal.

$$\kappa = \frac{\partial n_i}{\partial x_i} \qquad \text{Eq.23}$$

The CSF method tends to generate unphysical flow (‘‘spurious currents’’) near the interface when surface tension forces are dominant. These spurious currents are best illustrated in the limiting case of an inviscid static drop in equilibrium without gravity where Young-Laplace equation applies.

$$\Delta p = \sigma \frac{\partial n_i}{\partial x_i} \qquad \text{Eq.24}$$

The major reason for the spurious currents is a numerical imbalance of the surface tension force and the associated pressure gradient. In the context of sharp interface representation techniques such as VOF and FT, several studies have proposed different ways to reduce these spurious currents by either improving curvature estimation, improving the flow algorithm, or by combining better algorithms with interface curvatures estimation.
The order of magnitude of the spurious velocities u_{sp} can be estimated with respect to the surface tension coefficient and the dynamic viscosity μ of the droplet [53][53]. It can be calculated using the following formula:

$$u_{sp} = C_p \frac{\sigma}{\mu}, \qquad \text{Eq.25}$$

where C_p is a constant, which depends on the quality of the numerical modeling of surface tension forces. The ideal value of C_p is zero; however, typical values lie between 10^{-3} and 10^{-10}.

Recently, ghost fluid methods (GFM) have been proposed to impose sharper boundary conditions on embedded boundaries. Since GFM require knowledge of the distance from the interface, and since this information is naturally carried in LS methods, GFM have been applied successfully to model interfacial flow with surface tension [54] in conjunction with a LS technique. However, results in [54] also show a persistence of spurious currents as well as a loss of mass conservation.

The CSF method describes the discontinuous interfacial pressure jump as smooth, while the GFM method describes the discontinuous interfacial pressure jump as sharp. Thus, the GFM represents the sharp ST model (sharp surface force – SSF). As mentioned above, the GFM method requires the knowledge of the distance function Φ. The pressure jump $\sigma\kappa$ is applied only when Φ changes sign.

To simply illustrate the GFM method, 1D example is shown in Fig. 2. Usually, 1D Laplace's equation is solved and is given by the following formula:

$$\frac{\delta^2 p}{\delta x^2} = 0 \qquad \text{Eq.26}$$

$$\frac{p_{i+1} - p_i}{\Delta x^2} - \frac{p_i - p_{i-1}}{\Delta x^2} = 0 \qquad \text{Eq.27}$$

However, when the jump boundary condition across interface $[p] = A$ is present, then the Laplace's equation is modified to:

$$\frac{p_{i+1} - [p] - p_i}{\Delta x^2} - \frac{p_i - p_{i-1}}{\Delta x^2} = 0 \qquad \text{Eq.28}$$

$$\frac{p_{i+1} - p_i}{\Delta x^2} - \frac{p_i - p_{i-1}}{\Delta x^2} = \frac{A}{\Delta x^2} \qquad \text{Eq.29}$$

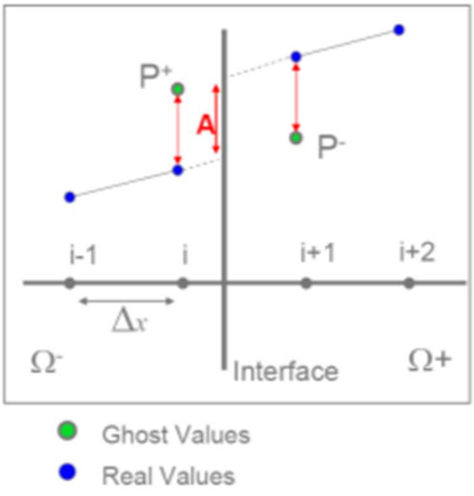


Fig. 2 *1D example of the GFM method*

2.1.5 Different ways for curvature estimation within VOF method

Basically, curvature estimation is determined by the method of an interface reconstruction. If a piecewise linear interface calculation (PLIC) [55] is used, then the normal to interface is a priori known because of linear interface approximation.
Unlike a geometric reconstruction schemes, if a different interface reconstruction scheme is used, the interface normal and the curvature are defined in the different way.
The simplest estimation of both, the interface normal and the curvature, is computed from local gradients of volume fractions.

$$n_i = \frac{\partial F}{\partial x_i} \qquad \text{Eq.30}$$

This approach is more likely inaccurate since the volume fraction function is a discontinuous function. However, it is still used in some commercial CFD software such as ANSYS Fluent, etc.

Height Functions (HF) for normal estimation

The height function HF technique [56], [57] gives more accurate results as regards the interface normal. The orientation of the interface is determined from the normal vector *n* which is evaluated as gradient of volume fraction. In 2D, a 7x3 stencil is constructed around a cell *(i,j)*. If $|n_y| \geq |n_x|$ height functions are constructed by integrating volume fractions in the vertical direction as

$$h_{i,j} = \sum_{j-3}^{j+3} F_{i,j} \Delta y_j \qquad \text{Eq.31}$$

where Δy_j denotes the mesh size in y-direction. The height functions are then used to compute the curvature at the center of the cell *(i,j)*:

$$\kappa = \frac{h_{xx}}{\left(1 + h_x^2\right)^{3/2}} \qquad \text{Eq.32}$$

where h_{xx} and h_x are discretized using second-order central differences:

$$h_{xx} = \frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x_i^2} \qquad \text{Eq.33}$$

$$h_x = \frac{h_{i+1,j} - h_{i-1,j}}{\Delta x_i} \qquad \text{Eq.34}$$

The normal n to the interface at the center of the cell (i,j) is calculated as:

$$n_i = \begin{bmatrix} \frac{\partial h}{\partial x} \\ -1 \end{bmatrix} \qquad \text{Eq.35}$$

Curvature derived from VOF function convolved with smooth kernel

Other way how to define the interface normal and the curvature is based on the low-order kernels widely used in the smoothed particle hydrodynamics (SPH) method [58], [59]. It gives good smoothing properties for smoothing lengths of twice the particle spacing. The cubic B-spline which has continuous first and second derivatives is used for smoothing of volume fractions and is given by

$$K(r,\gamma)=\frac{1}{\gamma^2}\begin{cases}\dfrac{40}{7\pi}\left(1-6\left(\dfrac{r}{\gamma}\right)^2+6\left(\dfrac{r}{\gamma}\right)^3\right) & if\ \dfrac{r}{\gamma}<0.5\\ \dfrac{80}{7\pi}\left(1-\dfrac{r}{\gamma}\right)^3 & F\ \dfrac{r}{\gamma}<1.0\\ 0 & otherwise\end{cases} \qquad \text{Eq.36}$$

Given the kernel K and the smoothing length γ, a smoothed volume fraction F field $\tilde{F}$ is defined by

$$\tilde{F}_{i,j}=\sum_{k,l}F_{k,l}K\left(\left|r_{i,j}-r_{k,l}\right|,\gamma\right)dxdy=K\cdot F \qquad \text{Eq.37}$$

The interface normal is then defined by

$$n_i=\left(\frac{\partial K}{\partial x}\cdot C,\frac{\partial K}{\partial y}\cdot C\right) \qquad \text{Eq.38}$$

The curvature is written in term of n

$$\kappa=\frac{1}{\left|n_i\right|}\left(\frac{n_x}{\left|n_i\right|}\frac{\partial\left|n_i\right|}{\partial x}+\frac{n_y}{\left|n_i\right|}\frac{\partial\left|n_i\right|}{\partial y}-\left(\frac{\partial\left|n_i\right|}{\partial x}+\frac{\partial\left|n_i\right|}{\partial y}\right)\right) \qquad \text{Eq.39}$$

DAN, DAC model

Both, Direction Averaged Normal model (DAN) and Direction Averaged Curvature model, are the other possible technique how to determine the normal and curvature of the interface. The rough approximation of the normal is first obtained using discretization of the *F* field. Then it needs to be improved that is done by so called distance function *f*. Since DAN approach is not conservative (the volume fraction field within the reconstructed interface does not have to be identical to the original field), thus, further correction is needed with respect to the mass conservation.

The calculation of the curvature is done using DAC model. It does not require any iteration process, however, it requires wider stencil around the cell of interest. Both models were claimed to be second-order accurate [59], [60], [61].

Reconstructing distance function for curvature estimation

Further, reconstructing a distance function method (RDF) [62] is another rather new approach how to calculate both, the interface normal and the curvature. It is based on the construction of the distance function Φ for cells in a local region near the interface. For the purposes of constructing the distance function, the cell g is deemed to be a interface cell if

$$\left|\nabla F\right|_g > \frac{B}{\Delta x} \qquad \text{Eq.40}$$

where ∇F is gradient of volume fraction defined using second-order centered differences. B is a scalar parameter set to 0.01. For any cell *(i,j)* with centroid coordinates (x_{ij}, y_{ij}) in local region near an interface cell g, simple geometry is used to find the normal distance from piece-wise linear interface in cell g to cell *(i,j)*.

$$D_{gij} = SD\sqrt{\left(x_{ij} - x_A\right)^2 + \left(y_{ij} - y_A\right)^2}\,, \qquad \text{Eq.41}$$

where

$$y_A = n_{gy}\left(n_{gx}X_1 + n_{gy}Y_1\right) + \left(n_{gx}\right)^2 y_{ij} - n_{gx}n_{gy}x_{ij} \qquad \text{Eq.42}$$

$$x_A = n_{gx}\left(n_{gx}X_1 + n_{gy}Y_1\right) + \left(1 - n_{gx}^2\right)\left(x_{ij} - y_{ij}\frac{n_{gx}}{n_{gy}}\right) \qquad \text{Eq.43}$$

and SD is the sign of the distance function and is given by:

$$SD = sign\left(n_g\left(x_{ij} - x_{\text{int}}\right)\right) \qquad \text{Eq.44}$$

where x_{int} is the centroid of the line segment in cell g and (X_1,Y_1) is any point on this line segment. For every cell (i,j) in the vicinity of the interface, a distance calculation to all nearby interface cells is performed, giving a distance Φ_{ij} constructed of weighted contributions from all nearby interface cells:

$$\Phi_{ij} = \frac{\sum_{g}^{N_g} W_{gij} D_{gij}}{\sum_{g}^{N_g} W_{gij}}\,, \qquad \text{Eq.45}$$

where W_{gij} is the weighted function and is defined as follows:

$$W_{gij} = F_g\left(1 - F_g\right)\left(\left|\cos\theta_{gij}\right|\right)^A \qquad \text{Eq.46}$$

where θ_{gij} is the angle between X_{gij} and the interface normal n_g. The illustrative scheme of the relevant geometry is depicted in Fig. 3. Both, the five-point and nine-point stencils are used to compute the normal, however, the nine-point stencil was found to be more accurate with respect to error norm results. The A constant equals to 25. Lower values of A would degrade the accuracy. Higher values of A show better convergence of $|\nabla\Phi|$, however, additional oscillations appear.

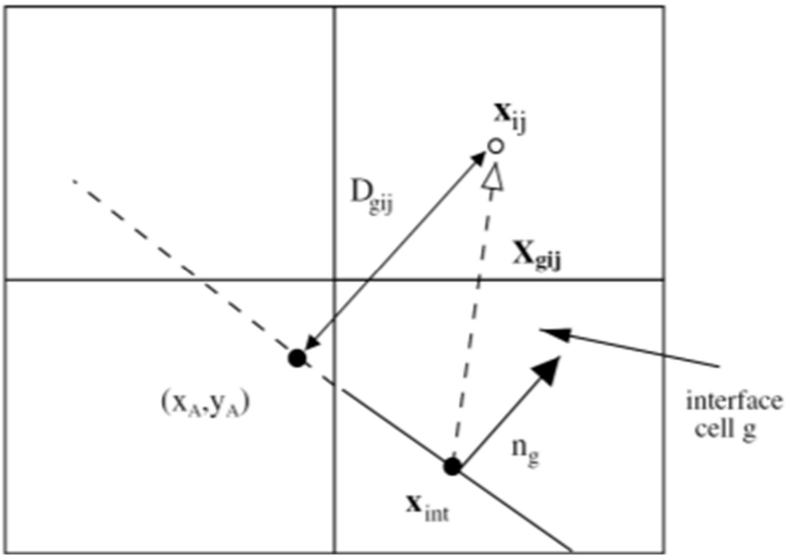


Fig. 3 *Scheme of the geometric quantities needed to reconstruct the distance function near an interface cell g*

Summary

In most VOF methods, the interfacial normal vector is estimated as ∇F which often results in a poor estimate of normal, simply due to the fact that the volume fraction function F is a discontinuous (Heaviside) function.

To combat the volume fraction based curvature problems, three approaches are plausible: (1) map the volume fractions into a function that is more smoothly varying so that standard finite-difference stencils can more precisely capture the variation of function F. It is provided by a convolution integral with a smoothing kernel K. (2) specialize the second-order derivative estimates via a more geometrically-based operator that is focused in the interface cells. (3) combine both, category (1) and category (2) to avoid high frequency aliasing errors.

Generally, HF method is best when product of both, the curvature κ and the mesh size Δx, are sufficiently low, otherwise either the convolution technique or RDF method should be used. According to this fact, it is possible to design some reliable method which switches between HF and RDF methods. This hybrid combination might be convenient especially in case of more complex interface topologies.

For problems within unstructured meshes, only the CV method and RDF are possible. HF methods, DAN, DAC methods can be only used within structured meshes.

2.1.6 Contact angle, contact line motion

The contact angle θ, is the angle between the solid surface and the tangent line drawn against the droplet surface [22]. Depending on the value of the contact angle, wetting is characterized as complete wetting (when $\theta = 0°$), partial wetting (when $0° < \theta < 90°$), and dewetting (when $\theta > 90°$).

The point where the three phases (solid, liquid, and vapor in Fig. 4) meet is known as the three phase boundary. Young's equation represents the relationship between contact angle and interfacial tensions:

$$\cos\theta = \frac{\sigma_{sg} - \sigma_{st}}{\sigma_{\mathrm{lg}}} \;, \qquad \text{Eq.47}$$

where σ_{sv}, σ_{sl} and σ_{lv} are the interfacial tensions of solid-gas, solid-liquid, liquid-gas, respectively.

Wetting is classified into two different categories, equilibrium or static (equilibrium contact angle), when the three phase boundary is stationary, and dynamic (dynamic contact angle), when the three phase boundary is moving.

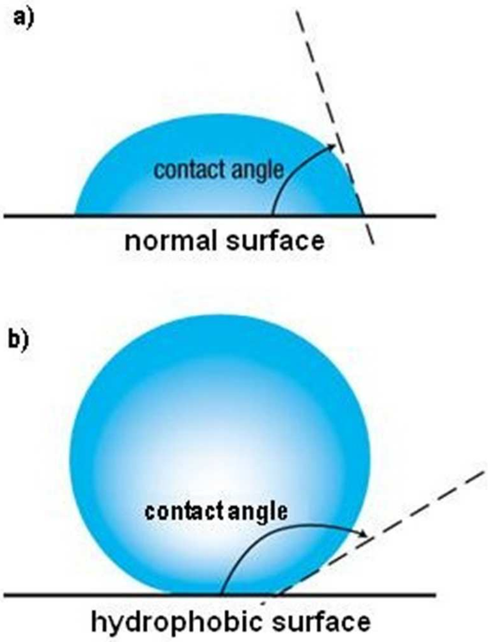


Fig. 4 *three phase contact line; a) wetting surface, b) hydrophobic surface*

The one of methods to measure the equilibrium contact angle is the sessile drop method. The one of widely used techniques to experimentally measure the dynamic contact angle is the direct optical visualization employing high speed camera to catch fast changes in contact angle.

The dynamic wetting phenomenon is important in numerous industrial and natural processes. When a fluid is displacing another immiscible fluid on a solid surface, the point at which the three phases meet is known as three phase boundary. If the three phase contact line is in continuous motion, it is called dynamic wetting. The angle between the moving contact line and solid surface is defined as the dynamic contact angle; depending upon the direction of the contact line movement, it is classified either as the advancing contact angle or as the receding contact angle. When the three phase contact line is moving in the forward direction (solid goes into the liquid), the dynamic contact angle is termed as the advancing contact angle, and when the contact line motion is in the reverse direction (solid comes out of the liquid), it is called the receding contact angle.

Hydrodynamic model

It is a macroscopic model [21] which considers that the three phase contact line motion is due to viscous dissipation, assuming the bulk viscous friction is the main resistance force for three phase contact line motion. This model does not take solid surface properties into account. For solid-liquid-gas systems, the hydrodynamic model is given by the following equation:

$$(\theta_d)^3 = (\theta_s)^3 \pm 9\frac{\eta U}{\sigma_{\lg}}\ln\left(\frac{L}{L_s}\right) \qquad \text{Eq.48}$$

where θd, θs are dynamic and equilibrium contact angles, respectively. η is the liquid viscosity. U is the three phase contact line speed. σ_{lv} is the liquid-gas interfacial tension. L is the capillary length, L_s is the slip length. In Eq.48, the positive sign is for the advancing movement and the negative sign corresponds to the receding movement of the contact line. Natural logarithm is considered to be a fitting parameter. The capillary length L can be calculated from

$$L = \sqrt{\frac{2\sigma_{\lg}}{\rho g}} \qquad \text{Eq.49}$$

As regards the size of the slip region L_s, the value of L_s is generally assumed to be 1-2 molecular layers over the surface.

Molecular-Kinetic Model

The molecular-kinetic model predicts the contact angle dependence on contact line velocity by incorporating the microscopic properties of the solid surface. The dynamic contact angle dependent on the velocity of the three phase contact line is expressed as [63]:

$$\cos\theta_d = \cos\theta_s \mp \frac{2kT}{\sigma_{lv}\lambda^2}\arcsin h\left(\frac{U}{2K_w\lambda}\right) \qquad \text{Eq.50}$$

where k is the Boltzmann constant, T is the absolute temperature, λ is the distance between adsorption sites, U is the velocity of the three phase contact line, Kw is the quasi-equilibrium rate constant. The negative sign is for the advancing movement and the positive sign is for the receding movement of the contact line. λ and Kw should be in order of 1 nm and 106 s^{-1}.

Combination of Molecular and Hydrodynamic Model

During the motion of the contact line, non-hydrodynamic frictional force is dominant near the three phase boundary whereas the bulk viscous force is dominant a bit more away from the three phase boundary. Both models are combined and the dynamic contact angle is given by the following formula:

$$(\theta_d)^3 = \left[\arccos\left(\cos\theta_s \mp \frac{2kT}{\sigma_{lv}\lambda^2}\arcsin h\left(\frac{U}{2K_w\lambda}\right)\right)\right]^3 \pm 9\frac{\eta U}{\sigma_{\lg}}\ln\left(\frac{L}{L_s}\right) \qquad \text{Eq.51}$$

Contact angle of water-air-steel system

The equilibrium contact angle is found between the two limiting values given by the static advancing and static receding contact angle values. The contact angle depends on the surface and the history of the droplet.

In most of available literature, the equilibrium contact angle was found to be 87°. In [22], the dynamics on the droplet impingement was tested in range of surface temperatures from 100°C to 220°C and the advancing and receding contact angles were measured. Different surface roughnesses were made to search for contact angle – roughness dependence.

The dynamic advancing contact angle begins at a value of around 130°. Further, there is an increase to 160° at surface temperature of approximately 200°C due to transition to the film boiling.

The dynamic receding contact angle begins at value of 60°. In transition region, its value jumps to the same value as that of the advancing contact angle.

The values of both, the advancing and the receding contact angle, of 60° and 130°, respectively, can be considered as reference values.

Summary

As regards the hydrodynamic model, the literature presents discrepant conclusions. In some cases, the hydrodynamic model fits well the experimental data. In other cases, it fits experimental data only quantitatively. Some authors claimed the hydrodynamic model to be valid only for low capillary numbers (Ca<1). Further, the dynamic contact angles of water on a PET surface reveal that the hydrodynamic model gives reasonable results for the low velocity region data; however, the estimated value for the slip length is too small compared to the molecular dimensions.

The molecular kinetic model was developed within the measurements of dynamic contact angles of glass-water-benzen system and the results were in good agreement with experimental results. On the contrary, employing of the molecular kinetic model within PET-water-air system reveals that there is a need for two different sets of parameters (λ and Kw) for high and low velocities. This surprising multi-mechanism behavior was observed also in case of PET-glycerol-air system.

Both, the hydrodynamic and molecular kinetic model, fail in prediction of the dynamic contact angle, but the combined molecular-hydrodynamic model does not.

Generally, none of them can be thoughtlessly used for the prediction of the dynamic contact angle of the arbitrary three phase system. It is always necessary to confront the model results with the experimental data.

2.2 *Euler-Euler model*

Here, the theory of the Euler-Euler model is outlined. Likewise VOF model discussed in 2.1, also Euler-Euler model incorporates the idea of volume fraction F, which occupies each particular computational cell. When only primary phase is present, it equals zero. Contrarily, when only secondary phase is present, F=1. For a cell containing a mixture of both phases the volume fraction F ranges from 0 to unity. Let the phase q be the only secondary phase. Then the continuity equation for this phase is

$$\frac{1}{\rho_{rq}}\left(\frac{\partial}{\partial t}\left(F_q\rho_q\right)+\frac{\partial}{\partial x_i}\left(F_q\rho_q c_{qi}\right)=\sum_{p=1}^{n}\left(\dot{m}_{pq}-\dot{m}_{qp}\right)+S_q\right), \qquad \text{Eq.52}$$

where the $\dot{m}$ stands for mass transfer from the primary phase to the secondary phase and vice versa. The quantity S characterizes a source term. The continuity equation is always calculated only for the secondary phase. The primary phase is calculated according to the following fact:

$$\sum_{q=1}^{n} F_q = 1 \qquad \text{Eq.53}$$

Further, the momentum equations are introduced for both phases. The momentum equation for the phase q is given by

$$\begin{aligned}\frac{\partial}{\partial t}\left(\alpha_q \rho_q c_{qi}\right) + \frac{\partial}{\partial x_i}\left(\alpha_q \rho_q c_{qi} c_{qj}\right) = -\alpha_q \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_i} + \alpha_q \rho_q g_i + \\ \sum_{p=1}^{n}\left(K_{pq}\left(c_{pi} - c_{qi}\right) + \dot{m}_{pq} c_{pqi} - \dot{m}_{qp} c_{qpi}\right) + \\ \left(F_{qi} + F_{lift,qi} + F_{vm,qi}\right),\end{aligned} \qquad \text{Eq.54}$$

Where *g* is the acceleration due to gravity, *F* is an external body force, F_{lift} is a lift force, F_{vm} is a virtual mass force and R_{pq} is an interaction force between phases. Effects of lift forces are very often neglected as discussed later in chapter 4.1.2. The virtual mass force is significant only when the secondary phase density is much lower than the primary phase density and moreover the secondary phase accelerates relative to the primary phase. In the most of cases it can be neglected. The interaction force between phases apparently is the most important quantity, depends on so-called interphase momentum exchange coefficient K_{pq} and the relative velocity between both phases. The interaction force R_{pq} is given by

$$R_{pqi} = \sum_{p=1}^{n} K_{pq}\left(c_{pi} - c_{qi}\right) \qquad \text{Eq.55}$$

The interphase momentum exchange coefficient K_{pq} is calculated as

$$K_{pq} = \frac{F_p F_q \rho_p f}{\tau_p}, \qquad \text{Eq.56}$$

where *f* is a drag function and τ_p is a relaxation time of droplet. The drag function *f* is a function of a drag coefficient C_D and Reynolds number *Re*.

$$f = \frac{C_D \,\mathrm{Re}}{24} \qquad \text{Eq.57}$$

For the most of cases it is recommended to use the symmetric model for the calculation of drag coefficient. The drag coefficient C_D [64] is defined as

$$C_D = \begin{cases} 24(1 + 0.15\,\mathrm{Re}^{0.687})/\mathrm{Re} & \mathrm{Re} \le 1000 \\ 0.44 & \mathrm{Re} > 1000 \end{cases} \qquad \text{Eq.58}$$

Next, the diameter of the secondary phase is supposed to be equal the diameter of the primary phase.

Apart from continuity and momentum equations, there is also energy equation to be solved, which can be written for each phase as

$$\frac{\partial}{\partial t}\left(F_q \rho_q h_q\right) + \frac{\partial}{\partial x_j}\left(F_q \rho_q c_{qi} h_q\right) = -F_q \frac{\partial p_q}{\partial t} + \tau_{qij} : \frac{\partial c_{qi}}{\partial x_i} - \frac{\partial q_{qi}}{\partial x_i} + S_q + \\ + \sum_{p=1}^{n}\left(Q_{pq} + \dot{m}_{pq} h_{pq} - \dot{m}_{qp} h_{qp}\right) , \qquad \text{Eq.59}$$

where the most notable quantity is the intensity of heat exchange between phases Q_{pq}, which is given by following formula

$$Q_{pq} = h_{pq}\left(T_p - T_q\right), \qquad \text{Eq.60}$$

where h_{pq} is the heat transfer coefficient between phases and is related to experimentally determined Nusselt number Nu by Ranz and Marshall [65], [66].

$$Nu_p = 2.0 + 0.6\,\mathrm{Re}_p^{1/2}\,\mathrm{Pr}^{1/3} \qquad \text{Eq.61}$$

As regards the modeling of turbulence, two options are available. The first one solves turbulence properties for mixture i.e. always only one field for each turbulent quantity exists. The second one solves transport equations for turbulence on a per phase basis i.e. each phase occupies different fields of turbulent properties.

It should be noted that a simplified version of Euler-Euler can be used instead. It is so-called Mixture model, which solves just one set of momentum equations and rather defines the slip velocity between phases. In some cases, the slip velocity can be negligible therefore it does not have to be taken into account. The slip velocity was found to be related to the relaxation time of droplet τ_p, drag function f and the droplet acceleration [64].

2.3 Euler-Lagrange model

The first 2D Lagrangian method was developed in the Lagrangian-incompressible code (LINC) in 1967. While other Lagrangian codes could solve 3D flows, the formulation of the LINC was applied on the staggered mesh and the coupling between pressure and velocity fields could be provided. Thus, the new MAC method was introduced and was used to study behavior of elastic-plastic materials and surface tension effects. The LINC method was also the first application where the Finite Volume Method was implemented [68]. The Lagrangian method (KIVA code) has mainly broken into the automotive industry namely the design of diesel jets, combustion chambers and manifolds.

In Euler-Lagrange model the fluid phase is treated as a continuum and is solved by the time-averaged Navier-Stokes equations, while the dispersed phase is treated as a large number of particles, droplets that are tracked through the whole computational domain and can exchange momentum, mass and energy with the continuous phase.

The discrete phase model assumes that the dispersed phase is sparse enough so it does not have any significant effect on the continuous phase. In other words, it says that the volume fraction F of the dispersed phase is rather low, commonly less than 10%.

Besides, say, standard Euler-Lagrange approach, a discrete phase model for dense dispersed phase was designed to overcome the limitation on the volume fraction *F*. The volume fraction of liquid dispersed phase can be thus almost up to unity; however, volume fractions equal 1 are not handled. Momentum and continuity equations were extended in terms of adding momentum exchange terms with the continuous phase, the volume fraction of the continuous phase is included in both equations and the velocity field is adopted from the Lagrangian tracking solution.

The basic form of continuity and momentum equation is given by the following set of equations:

$$\left(\frac{\partial}{\partial t}(\rho)+\frac{\partial}{\partial x_i}(\rho c_i)=S\right), \qquad \text{Eq.62}$$

where *S* is the mass source added to the continuous phase from e.g. vaporized liquid droplets or any other source.

$$\frac{\partial}{\partial t}(\rho c_i)+\frac{\partial}{\partial x_i}(\rho c_i c_j)=-\frac{\partial p}{\partial x_i}+\frac{\partial \tau_{ij}}{\partial x_i}+\rho g_i+F, \qquad \text{Eq.63}$$

where *F* is the momentum source added to the momentum equation due to e.g. drag force acting onto droplets.

The trajectory of droplets is calculated by integration of the force balance on each particle mass in a Lagrangian frame of reference. In Cartesian coordinates this force balance is written as

$$\frac{du_{di}}{dt}=F_D(u_i-u_{Di})+\frac{g_i}{\rho_p}(\rho_p-\rho), \qquad \text{Eq.64}$$

where F_D is defined by the particle relaxation time τ_p and the drag function *f*. The drag force F_D can be written as

$$F_D = f/\tau_p \qquad \text{Eq.65}$$

The drag function *f* has been already defined in chapter 2.2. The drag coefficient C_D depends mainly on the particle shape, Reynolds number and turbulent properties of the flow. For spherical droplets the most suitable drag law should be the drag law for spherical particles (Eq.58). In the case of dynamically distorting droplets, when the drag coefficient varies with the time, dynamic drag coefficient is usually more accurate.

The particle relaxation time is defined by the following formula

$$\tau_p=\frac{\rho_d d_d^2}{18\mu} \qquad \text{Eq.66}$$

Except of the drag force F_D, also other forces acting onto the droplet can be included such as the gravity force, forces in rotating frame of reference, Brownian force etc.

The velocity of particle is defined as a derivation of the position with respect to the time.

$$\frac{dx_i}{dt} = u_{di} \qquad \text{Eq.67}$$

Thereof, Eq.64 and Eq.67 make a set of ordinary differential equations, which are coupled together and can be solved by either analytical integration or numerical discretization schemes.

When the flow is turbulent, it is possible to account for effects of turbulence on droplet motion. There are two approaches that can be used.

The first is the **Stochastic Tracking** approach, where the fluctuating component of velocity u' is added to the trajectory equations. The new particle path is computed from sufficient number of particles. This stochastic approach is also used for the determination of the continuous phase velocity that can result in a poor convergence during calculation. The value of fluctuating velocity component is kept constant for the characteristic lifetime of eddies. This approach is said not to be appropriate for diffusion-dominated flows.

Droplet dispersion by turbulence can also be modeled with **Particle Cloud model**, which calculates turbulent dispersion around a mean trajectory using statistical methods. The concentration of droplets around the mean trajectory is given by the Gaussian probability density function dependent on the turbulent intensity. The mean trajectory is derived as an average trajectory from all particle trajectories in the cloud.

As regards heat and mass transfer within droplet, several regimes can occur. When the droplet temperature T_d is below the vaporization temperature T_{vap}, a simple ordinary differential equation can be written to describe heat transfer from droplet to ambient based on a correlation for the heat transfer coefficient. This equation has the following form

$$m_d c_{dp} \frac{dT_D}{dt} = hA_d \left(T_\infty - T_d\right) , \qquad \text{Eq.68}$$

where A_d is the surface area of the droplet, h is the heat transfer coefficient using the correlation described in the chapter 2.2. When the droplet temperature T_d is higher than the vaporization temperature T_{vap} and is lower than the boiling temperature T_{boil}, than the droplet evaporates and the vaporization rate is driven by the molar flux of vapor, which is defined as

$$N_i = k\left(C_{iA} - C_{i\infty}\right), \qquad \text{Eq.69}$$

Where C_{iA}, $C_{i\infty}$ are vapor concentration at the droplet surface and the vapor concentration in the continuous phase respectively. The variable k is the mass transfer coefficient, which can be calculated from the correlation for Sherwood number Sh defined in [65], [66]. Vapor concentration on the droplet surface is calculated from the ideal gas law considering the saturated vapor pressure p_{sat}. The vapor concentration in the continuous phase is determined in a similar way, but the calculation contains also the mole fraction of vapor species.

When the droplet temperature reaches the boiling point, the droplet temperature is kept at fixed boiling temperature until the temperature of the bulk continuous phase drops below the boiling point. A boiling rate was defined by Kuo [67], who claimed the derivative of droplet diameter with respect to time depends on Reynolds number, latent heat, the bulk and the droplet temperature etc. and is given by following equation:

$$\frac{d(d_d)}{dt} = \frac{4k_\infty}{\rho_d c_{gp,\infty} d_d}\left(1 + 0.23\sqrt{\mathrm{Re}_d}\right)\ln\left(1 + \frac{c_{gp,\infty}\left(T_\infty - T_d\right)}{h_{fg}}\right) \qquad \text{Eq.70}$$

Here, it should be noted that both, the vaporization and the boiling law, can be only applied when the transport equation for species is being solved. Transport equation for species i calculates mass fraction M_i and it is actually a convection-diffusion equation of the scalar M_i.

In an open domain the droplet fate is driven by those aforementioned laws, whereas droplet collisions with the boundary are controlled by prescribed boundary conditions. Commonly used boundary condition for droplets is e.g. droplet **reflection** boundary condition. Droplet can rebound from the boundary either with or without a loss of momentum that is given by the coefficient of restitution. In fact, two coefficients of restitutions exist. The first one is dedicated to the normal direction i.e. it says the amount of droplet momentum which retained by the droplet in normal direction. Similarly, the second one represents the tangential coefficient of restitution. When the droplet particle retains all of its momentum, the coefficient of restitution equals unity. When the coefficient of restitution is zero, none of the momentum is retained.

The simplest boundary condition is obviously the condition of the **escaped** droplet. This BC is usually used together with any outlet BC for the continuous phase. In some cases, it is required that the droplet sticks to the boundary and the whole volatile fraction is changed into the vapor. The BC for this droplet fate is called the BC of the **trapped** particle.

The last but one BC discussed here is the **wall-jet** type boundary condition. Several of wall-jet models were designed to describe the underlying physics of the jet impingement. One of them was the wall-jet model proposed by Naber and Reitz [69], which was firstly incorporated into KIVA code. They formulated the relation between the Weber number of the impinging droplet and the Weber number of the rebounding droplet. This model calculates the direction of the rebounded droplet and its velocity; however, it does not account for droplet dispersion process. Several modifications of wall-jet model were published later such as that one by Senda [70], who considered a different dispersion phenomena for temperatures below the boiling temperature and those ones above the boiling temperature. For surface temperatures below the Leidenfrost point another wall-jet model was made up by Grover [71], which considers three splashing parcels and one wall film parcel that represent the shattering of a splashing droplet on the wall. Since the viscous dissipation can be dominant for high Weber numbers, it was thus included in the energy conservation as a source term.

The last wall BC named **wall-film** type boundary condition, which is implemented in Fluent, is the most complex since it is made up of droplets that can either stick to the wall, spread, splash or even rebound from the wall. This, say, droplet regime is judged by the impact energy E defined by

$$E = \left(\mathrm{Re}\left(\frac{1}{\min\left(h_0 / d, 1\right) + \delta_l / d}\right)\right)^{0.5} \qquad \text{Eq.71}$$

and the wall temperature T_w. The variable h_0 stands for the film height, δ_l is the boundary layer thickness and d is the droplet diameter. The impact energy less than 16 corresponds to the sticking regime of the droplet and the droplet velocity is set equal to the wall velocity. The droplet regime with the second lowest impact energy is the spreading regime when the droplet velocity is set using the wall-jet model. As the critical impact energy, when the splash regime occurs, is the value of 57.7 and the droplet can be shattered into the predefined number of splashed parcels. It should be noted that the number of parcels does not equal the number of splashed droplets. In other words, in each splashed parcel more than one splashed droplet can be. For each splashed parcel a different diameter is calculated according to a cumulative probability distribution function [72]. This distribution functions itself naturally do not give physical results with respect to Weber number We and thus the expression for d_{max} was defined to determine the maximal droplet diameters for each distribution function [73][73]. The total number of splashed droplets is obtained from the amount of mass splashed from the surface, which is defined as a quadratic function of the splashing energy and it follows experimental findings of Mundo [72]. Similarly, the velocity of droplet is also determined using a probability function and experimentally verified data. Finally, an energy balance is performed for new formed droplets so that the total energy of them does not exceed that of the old droplets.

3 Single water droplet

In this chapter, the motion of water droplet is studied in Fluent. The droplet free-fall is simulated using the VOF method. For low Weber and Capillary numbers surface tension effects must be included. Modeling of surface tension is theoretically and numerically analyzed. The droplet spreading on the surface and also heat transfer is not discussed here.

3.1 Theory of droplet free fall

Momentum is generally transferred between both, air and water phases, through mass transfer interphase drag, lift, gravity and buoyancy. The lift force is important when the density of droplet is much lower than density of ambient, e.g. bubbly flows. Thus, the lift force can be neglected. Further, mass transfer between phases is not considered and so it can be also neglected. Moreover, buoyancy force given by the following formula

$$F_b = \rho_g g \frac{\pi D^3}{6} \qquad \text{Eq.72}$$

can be also neglected due to very low density of air. Then the force balance on the droplet can be written as:

$$m\frac{dv}{dt} = mg - C_D A \rho_g \frac{v^2}{2}, \qquad \text{Eq.73}$$

where C_D is the drag coefficient. In general, the drag coefficient C_D is a function of particle shape and its orientation with respect to the flow, Reynolds number, turbulence level and Mach number.

In many cases, drag coefficient of sphere can be considered to be constant of 0.45 and the drag coefficient is following so called Newton's law. It is correct in range of Reynolds numbers (8e+02–3e+05); however, it is completely different, especially in low Reynolds numbers. Effect of Reynolds number on the drag coefficient of a smooth sphere moving inside an air has been experimentally studied. With increasing number, the flow begins to separate and form vortices behind the sphere. The pressure in the wake is further reduced, thus, the drag is increasing. At the critical Reynolds number (Re_c~3e+05) the boundary layer becomes turbulent and the separation point is moved rearward, sharply reducing the form drag and decreasing the drag coefficient. Besides experiments, many empirical formulas have been developed to describe drag coefficient C_D as a function of Reynolds number. Both, experimental data and several empirical formulas are shown in Fig. 5. The Morsi-Alexander model is one of the most precise. It is able to follow the standard drag curve very well; however, it is evidently complicated compared to other drag models (APPENDIX I).

One should note that these models are derived from experiments with solid sphere. In reality, the droplet could deform, some wrinkles might appear on surface or the droplet could even breakup into smaller droplets. Thus, the Drag coefficient would vary along with droplet deformations and would be completely different from the drag of solid sphere. Droplet distortion occurs because of an uneven difference between external aerodynamic and internal hydrodynamic and internal hydrostatic pressures at the droplet surface. This difference in pressure has to be balanced by the surface curvature and surface tension. The surface tension force tries to maintain the droplet shape. Whether the droplet breakup will happen or not depends on so called critical Weber number We_C defined as the following:

$$We_C = \frac{\rho_g U^2 d}{\sigma} \quad \text{Eq.74}$$

Experimental observations of several authors for low viscosity liquids (water) provide a value of We_C of about

$$5 < We_C < 20 \quad \text{Eq.75}$$

with the most commonly used value being

$$We_C \approx 12 \quad \text{Eq.76}$$

Within the margin $5 < We_C < 20$ there is a dependence on the Reynolds number not considered into the above mentioned approaches.

The terminal velocity of large droplets is about

$$U_t \approx \left(\sqrt{2} to 1.7\right)\left(\frac{\sigma g \Delta\rho_{gl}}{\rho_g^2}\right)^{1/4} \quad \text{Eq.77}$$

$$We_C \approx 4.8 to 7.1 \quad \text{Eq.78}$$

Thereof, the terminal velocity of water droplet of diameter of 2 mm should range from 6.5 to 7.8 m/s. Consequently, corresponding Weber and Reynolds numbers are 1.46, 2.11, 902 and 1083, respectively. Since $We < We_C$, no breakup will occur and thus, drag coefficient can be considered the same as the drag coefficient C_D of solid sphere.

For completeness, the drag of distorted droplet is greater for two reasons. Firstly, the frontal area of droplet is bigger. Secondly, the bigger curvature at the equator leads to earlier flow separation.

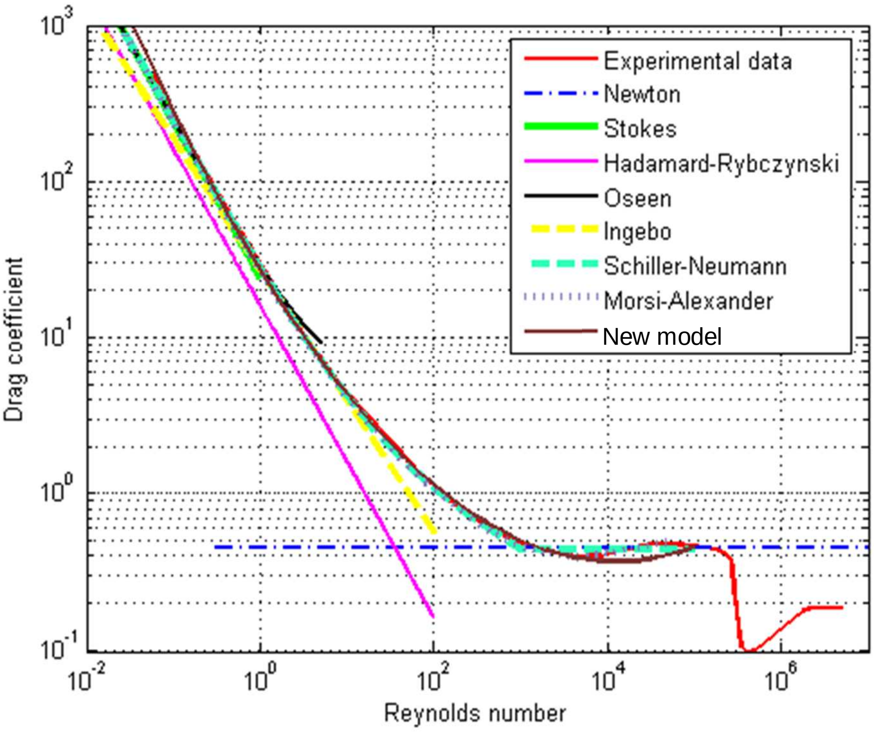


Fig. 5 *Drag coefficient* C_D *dependent on dimensionless Reynolds number; different models*

3.2 Experimental droplet data from literature survey

Many researchers have been concerning with droplet investigations. It has begun with the exploration of rain erosion of aircraft components and blade erosion of gas or steam turbines. Some experiments were carried out with rigid spheres and some with liquid droplets. The best experimental data was presented by Maybank and Briosi (1956), Scott, Wood and Thurston (1964), Clift, Grace and Weber (1978), Lane and Green (1956). The experimental data of all of them are in excellent agreement up to droplet radius of 6 mm and is shown as a single curve in Fig. 6. The most reliable data was obtained for large droplets free-falling in ambient air and normal gravity. The experimental data obtained in a convergent wind tunnel was found to be less reliable mainly because of the artificial production of turbulence.

To make the picture complete, the drag coefficient dependence on Reynolds number for both, the water droplet and the solid droplet, is shown in Fig. 7. It is clearly seen that drag coefficients are nearly identical up to Reynolds number of 1000. Further, the increase in drag coefficient of the water droplet is caused by the droplet distortion. Other very useful information about experimental data, data capturing methodology and data correlation can be found in [75].

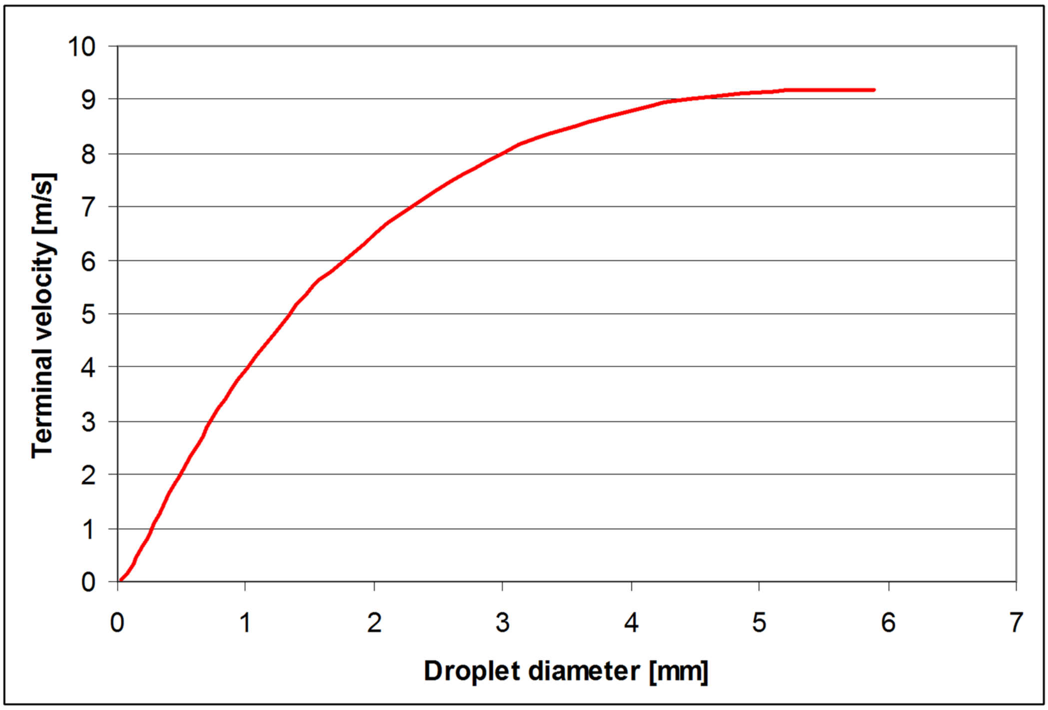


Fig. 6 *Terminal velocity dependent on droplet diameter*

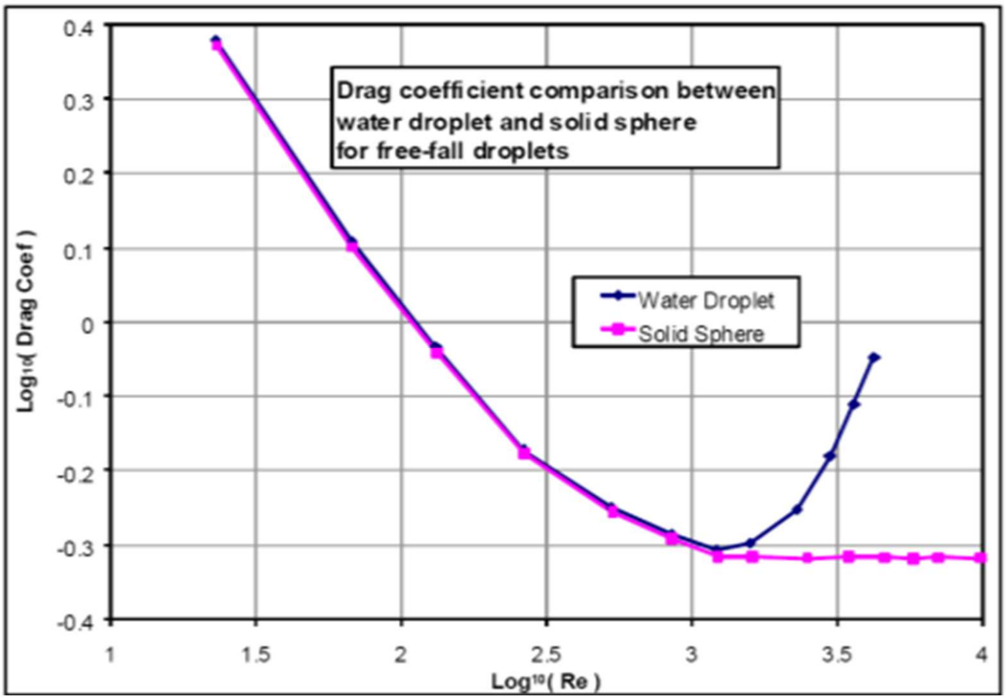


Fig. 7 *Drag coefficient on free-falling droplets against Reynolds number*

3.3 New drag law design

Based on experimental data, new drag law has been developed valid up to Reynolds number of 1e+03. The method of least squares was employed to fit experimental data. The new drag law is shown in Fig. 5 and is given by the following formula:

$$C_D = \mathrm{Re}^{=0.9116} \cdot \exp(0.04833 \cdot \log(\mathrm{Re})^2 + 3.2983) \qquad \text{Eq.79}$$

The new drag law was compared with the experimental data and both, Schiller-Neumann and Morsi-Alexander model. The absolute errors are plotted for each of them in Fig. 8 and are defined as follows:

$$\Delta x = x_0 - x\,, \qquad \text{Eq.80}$$

Where x_0 is the measured value and x is the actual value. The new drag law obviously produces the lowest absolute errors compared with other drag laws.

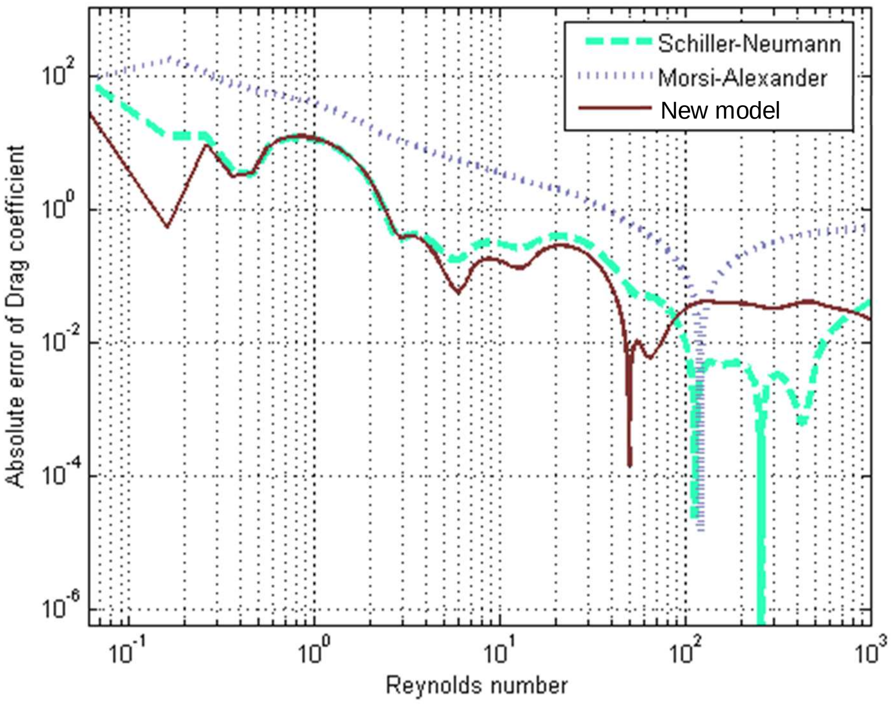


Fig. 8 Absolute error of Drag coefficient for three different drag laws

3.4 Droplet acceleration, terminal velocity

The purpose of determining the droplet acceleration was to validate computational models. The equation Eq.73 was numerically solved considering both, constant and variable drag coefficient. The solution was found for two droplet sizes (droplet diameter of 2.0 and 0.2 mm).

3.4.1 Constant drag coefficient

The water droplet of the defined diameter is falling considering the force of gravity and the drag force. The drag coefficient was considered to be constant of 0.5. The purpose of the consideration of constant drag coefficient was just to test three different numerical methods on simple ordinary differential equation and determine the droplet acceleration and the droplet terminal velocity. The explicit Euler, the midpoint and Runge–Kutta method were arranged with respect to their complexity and accuracy in MATLAB.

For the sake of brevity, only one numerical configuration is presented below, see Tab. 1. Results of all three numerical methods are compared in Fig. 9. Results are evidently coincident. However, the explicit Euler method was unstable for higher time steps, thus, the most complex Runge–Kutta method was finally used for all other numerical configurations. Time dependent velocities are clearly shown in Fig. 9. The calculated terminal velocity was approximately 6.5 m/s. However, it must be noted that constant drag coefficient was considered.

Tab. 1 *Numerical configuration*

Initial velocity v_0 [m/s]	0
Drag coefficient [-]	0.5
Density of air [kgm^{-3}]	1.25
Acceleration of gravity [ms^{-2}]	9.81
Droplet diameter [mm]	2
Time step [s]	0.0001
Terminal velocity [m/s]	**6.5**

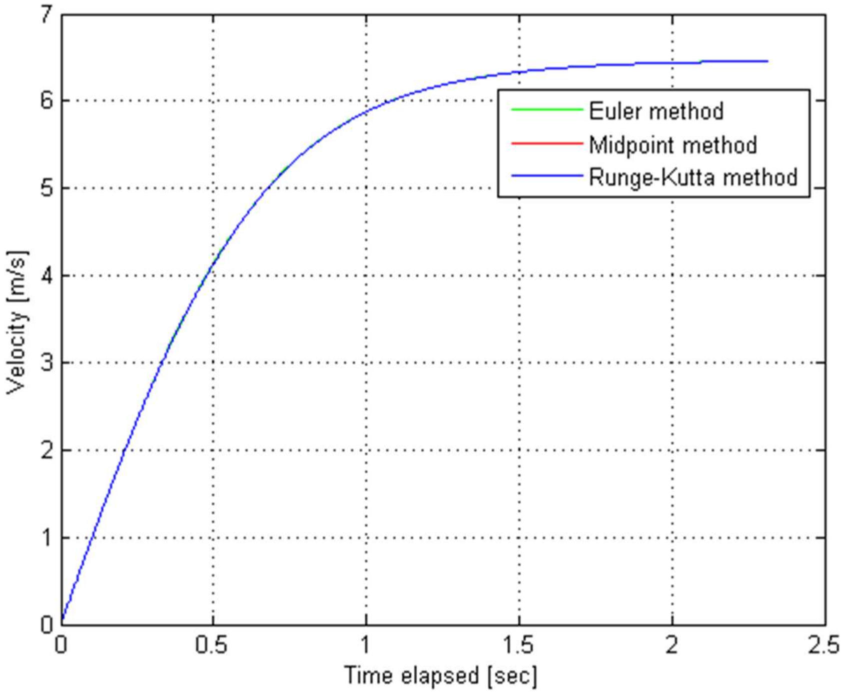


Fig. 9 *Velocity development of water droplet falling onto plane surface, three different numerical methods*

3.4.2 Variable drag coefficient

The Runge-Kutta method was used to solve Eq.73 with the new drag law valid up to Reynolds number of 1e+03 (APPENDIX II). Reynolds number of the droplet of diameter of 2 mm was not presumed to be higher than 1e+03, thus, aforementioned drag law could be used. The numerical configuration was copied from the previous chapter. As shown in Fig. 10, the drag coefficient develops until the terminal velocity of 6.37 m/s is reached. In first time steps, the drag coefficient was held constant of 492 since there were no experimental data available. If have a look on axis of the time elapsed, the elapsed time until employing variable drag coefficient is very low anyway. Thus, for low Reynolds flow, the lack of experimental drag coefficients should produce insignificant error. The time elapsed until the terminal velocity was reached was approximately 2 sec, which is roughly the same as in the case with the constant drag coefficient of 0.5. The terminal velocity was found to be slightly smaller than in simplified case. It was around 6.37 m/s.

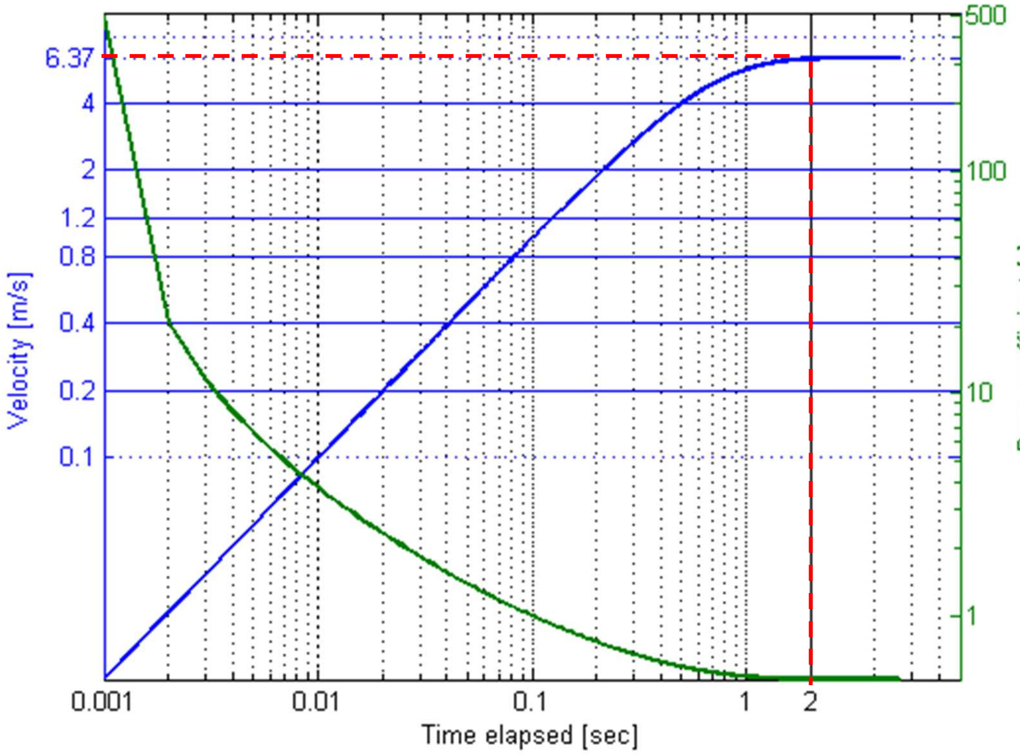


Fig. 10 *Velocity and Drag coefficient development for water droplet of diameter of 2 mm under gravity, starting with velocity of 0 m/s*

In case of droplet diameter of 0.2 mm, the calculated terminal velocity was 1.8 m/s and the time elapsed until the terminal velocity was found to be approximately 0.2 s. The drag coefficient C_D did not drop below the value of 0.7, whereas in case of droplet diameter of 2.0 mm, the drag coefficient was 0.5. Results are summarized in Tab. 2.

Tab. 2 *Numerical results of free-falling droplet*

Droplet diameter [mm]	**0.2**	**2.0**
Terminal velocity [m/s]	1.8	6.37
Time until terminal velocity reached [s]	0.4	2.0
Drag coefficient within terminal velocity [-]	0.7	0.5

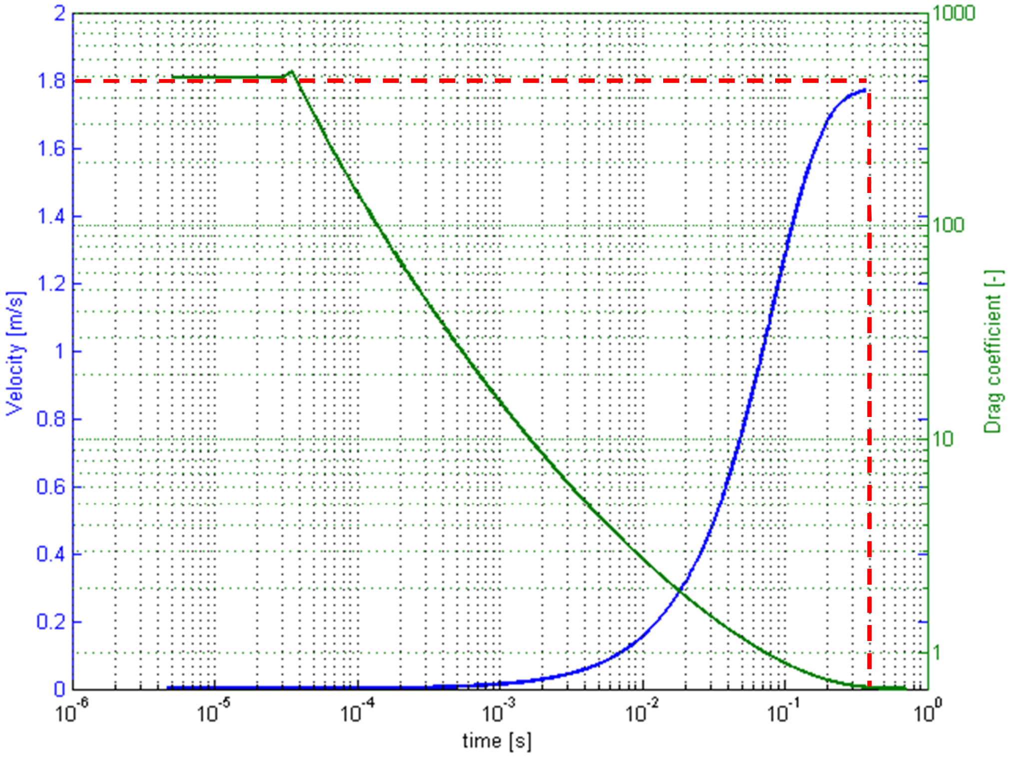


Fig. 11 *Velocity and Drag coefficient development for water droplet of diameter of 0.2 mm under gravity, starting with velocity of 0 m/s*

3.5 Calculation of terminal velocity of free-falling droplet using FLUENT

Three different cases were set up and solved in commercial CFD package FLUENT employing user defined functions (UDFs) and script files. The purpose of all of them was to study the flow field inside and outside the droplet. Moreover, the first test case was aimed at the determination of the terminal velocity. In the third test case, the time dependence of droplet velocity within gravity was studied.

The VOF model [41], [42] together with surface tension model [43] was used to simulate the flow of two immiscible phases (water and air).

3.5.1 CFD simulation of terminal velocity of free-falling droplet

Two different droplet diameters (0.2 and 2.0 mm) were used in computations. The calculations were carried out as 2D axisymmetric and the principle of **the droplet frame of reference** was used to avoid the need of the dynamic mesh refinement i.e. the position of the droplet remained the same within time scale. The droplet frame of reference is given by the superposition of the global flow **u** and the constant translation velocity **v** of the droplet.

$$\vec{v} = \vec{u} - \overrightarrow{U_d}\ , \qquad \text{Eq.81}$$

where U_d is the local droplet velocity.

The computational domain was rectangular, originally with mapped mesh. To obtain more precise results, the grid was refined along the interface and in regions of large velocity gradients. It is shown in Fig. 12. Due to the rather low Reynolds number and very small droplet velocities, the flow was modeled as laminar.

Pressure outlet, velocity inlet, slip wall, axis were imposed as boundary condition. Pressure outlet BC corresponded to the ambient (the relative pressure of 0 Pa). Velocity inlet BC was controlled and adapted by the UDF. The free slip was considered on the wall.

The user defined function (UDF) and script file were employed to adjust the velocity i.e. the velocity inlet boundary condition (see APPENDIX II). At first, the droplet velocity was calculated. Then, if the velocity is in the same direction as the force of gravity, its value is added to the velocity inlet boundary condition. Otherwise, the value of droplet velocity was subtracted from the velocity inlet boundary condition. Model settings and results are shown in Tab. 3. Solution settings, material properties are available in APPENDIX III.

To sum up, the solution of 2 mm droplet converged to the constant value of velocity inlet BC. As mentioned in Tab. 2, the value of the velocity inlet BC was found to be 6.56 m/s, which is very close to the experimental data and the numerical solution of Newton's law.

However, in case of 0.2 mm droplet, the model did not converge to one single value of terminal velocity. The velocity inlet BC was oscillating between 1.80 m/s and 2.01 m/s. According to the experimental data and MATLAB simulations, the terminal velocity should be approximately 1.8 m/s. Oscillating behavior in this case is connected with the CSF surface tension model used. As described in section 2.1.4, the prediction of both, the interface normal and the curvature, is not enough accurate and it causes spurious currents, consequently. Since rather low droplet velocity, those spurious currents were clearly seen, significantly affected the flow near the interface and led to the oscillating prediction of the terminal velocity.

As regards the post-processing of 2 mm droplet, contours of velocity, path lines and the droplet interface are shown in Fig. 13. Due to the shear stresses in the vicinity of the interface, the water circulated with the maximal axial velocity magnitude of 0.2 m/s and the wake of the length nearly of 5 mm is formed right behind the droplet.

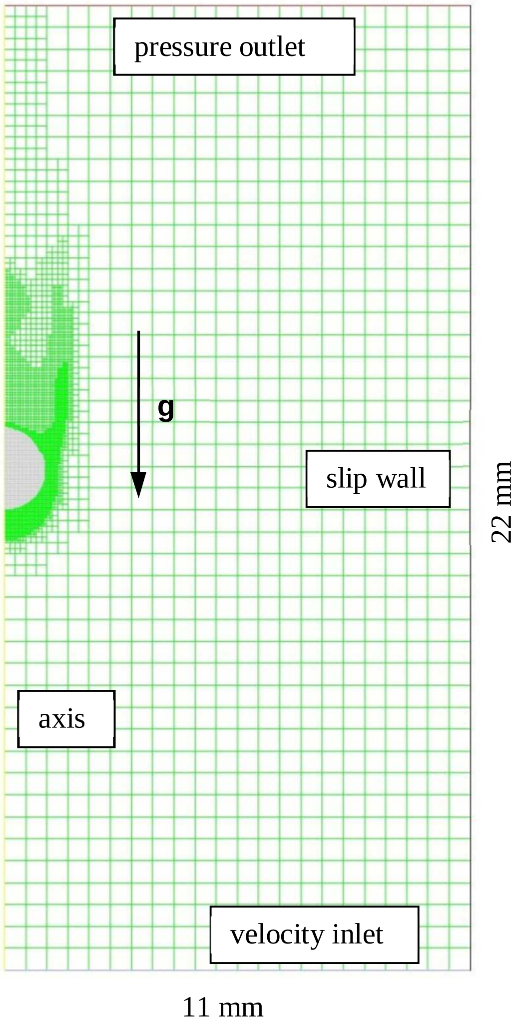


Fig. 12 *Mesh grid with four refinement levels; droplet diameter of 2.0 mm*

Tab. 3 *Model settings and results*

droplet diameter [mm]	**0.2**	**2.0**
time step [s]	4.3e-08	3.3e-07
number of iterations per time step [-]	15	15
size of original elements [m]	0.0005	0.0005
level of refinement [-]	8	4
size of the finest elements [m]	approx. 1.9e-06	approx. 3.9e-06
droplet velocity [m/s]	**oscillating (1.8 – 2.0)**	**6.56**

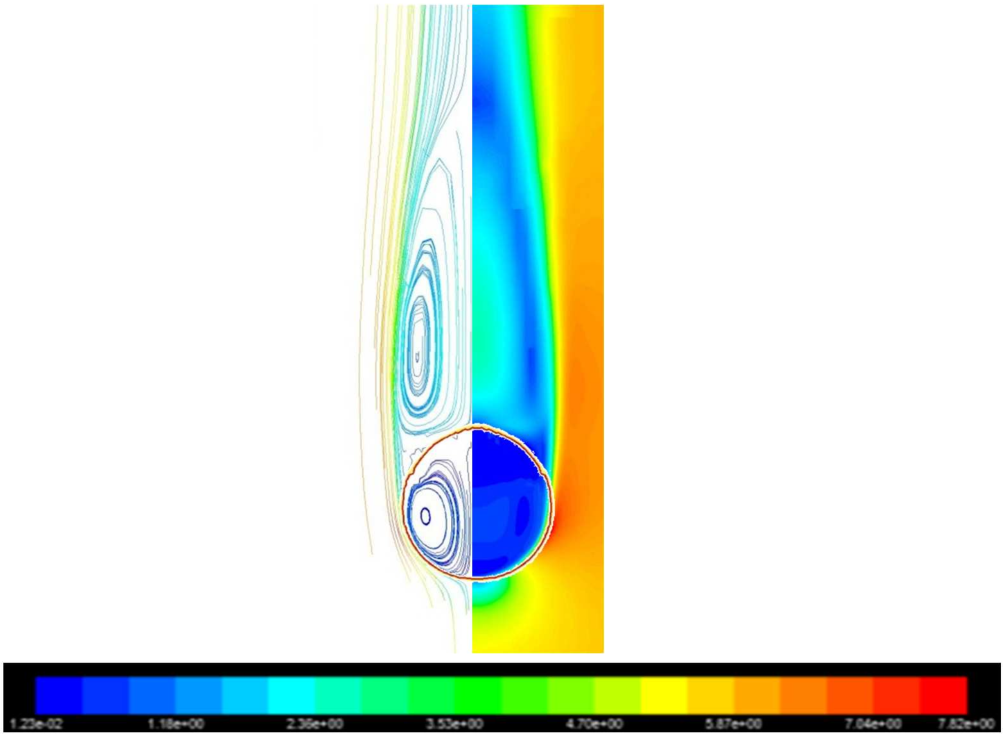


Fig. 13 *Free-falling droplet (diameter of 2 mm), contours of velocity [m/s], path lines, droplet interface*

3.5.2 CFD simulation of velocity field around free-falling droplet and its interior

This model was developed to avoid a multiphase model and thus, reduce the computational costs. The computational costs were, however, mainly reduced due to steady solver used. The droplet interface was considered as a wall and the exchange of momentum was provided via UDF described in APPENDIX IV. Although the real droplet surface could be distorted and wrinkled, based on experiments, it was supposed to be spherical all the time. Therefore, the interface was spherical and static, simultaneously. Since the interface was represented by the wall (see Fig. 14), there were two coincident walls in fact – the first one bounding the air fluid zone with the no-slip BC imposed, the second one bounding the water fluid zone with the free-slip BC imposed. Settings of material properties, solver settings and boundary conditions were the same as in section 3.5.1. However, the velocity inlet BC was constant and corresponded to the experimentally validated terminal droplet velocity (2 mm droplet ≈ 6.5 m/s).

To summarize, the droplet surface was idealized and imposed as the static spherical wall type BC. The solving of the transport equations was conducted in the steady mode that saved a lot of computational time. The calculation was more than 20 times faster than the aforementioned VOF calculation (section 3.5.1). However, since the droplet surface was represented by the spherical static wall, it was necessary to deal with the shear condition. Since the results from the previous VOF calculation revealed the wake behind the droplet, the no-slip shear condition was finally imposed. If the free-slip shear condition was imposed, no boundary layer separation and the wake, consequently, would appear. In fact, the shear condition is much more complex and lies between the free-slip and the no-slip BC.

Only the droplet of the diameter of 2 mm was simulated and compared with the aforementioned VOF calculation. Contours of velocity, path lines and the droplet interface are shown in Fig. 15. The water circulated with the maximal axial velocity magnitude of 0.7 m/s that is more than 3 times higher than in the aforementioned VOF calculation. This discrepancy can be explained by the method of momentum exchange through the interface applied described in APPENDIX IV. The length of the wake in the axial direction was 3 mm that is 60% of the wake length from the previous VOF calculation (section 3.5.1).

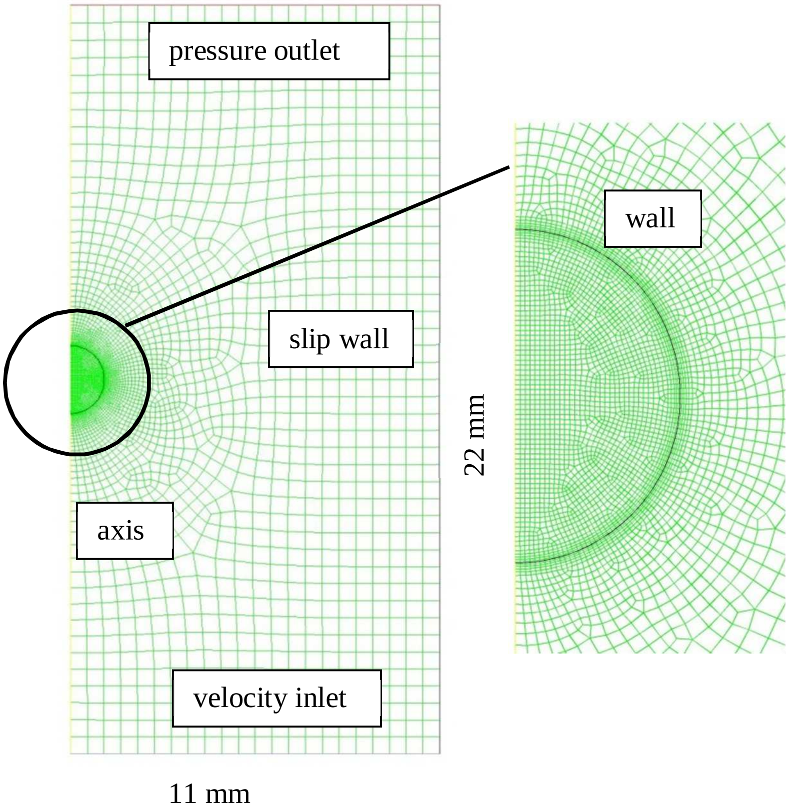


Fig. 14 *Mesh grid of the computational domain containing droplet of diameter of 2 mm represented by the spherical wall*

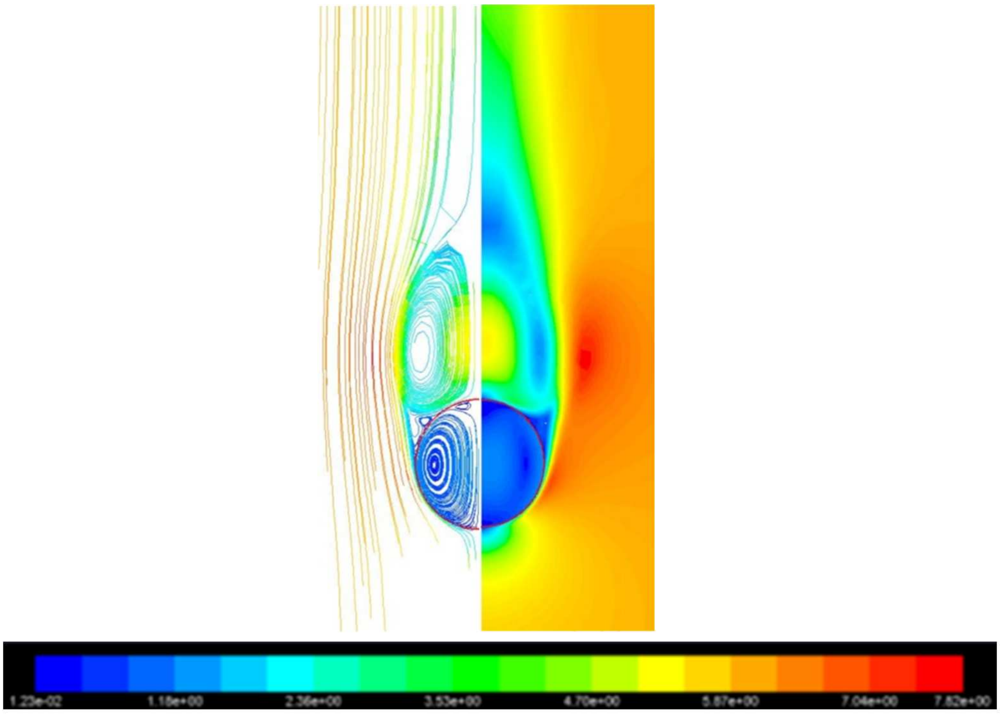


Fig. 15 *Both, flow field around the droplet represented by the spherical wall and the flow field inside the droplet interior*

3.5.3 CFD simulation of droplet accelerating in gravity field

The purpose of this study was to investigate the acceleration of a free-falling droplet with initial velocity of zero and compare it with the solution of Newton's law employing the dynamic drag coefficient.

The droplet was confined to the rectangular 2D zone. The boundary conditions were three slip walls and the axis; thereof, there was neither an inlet nor an outlet imposed. As in the previous test cases, the model was axisymmetric with the same model settings (APPENDIX III) excluding the VOF settings described below. Three droplet diameters were considered (0.2 mm, 0.8 mm, and 2.0 mm).

As regards initial conditions, the droplet was positioned in the center of domain and it was patched with all components of velocity equaled to zero. Since the droplet was moving through the domain, the domain had to be large enough to avoid the droplet hitting the wall. Another reason for a large domain was that the flow field near the droplet interface could be affected otherwise.

The dynamic mesh adaption with several refinement levels had to be used due to much bigger dimensions of the domain than the droplet size, to save computational resources. For this purposes, the script file and the UDF were developed to control the grid coarseness near the droplet interface (APPENDIX V).

The basic grid was mapped and identical for all droplet sizes. The basic quadrilateral cell had the length of 0.5 mm. According to the droplet diameter, the basic grid was several times refined in the vicinity of the droplet interface.

In case of the droplet diameter of 2 mm, four different refinement levels were tested within one model settings to study the dependence of results on the grid (see Tab. 4).

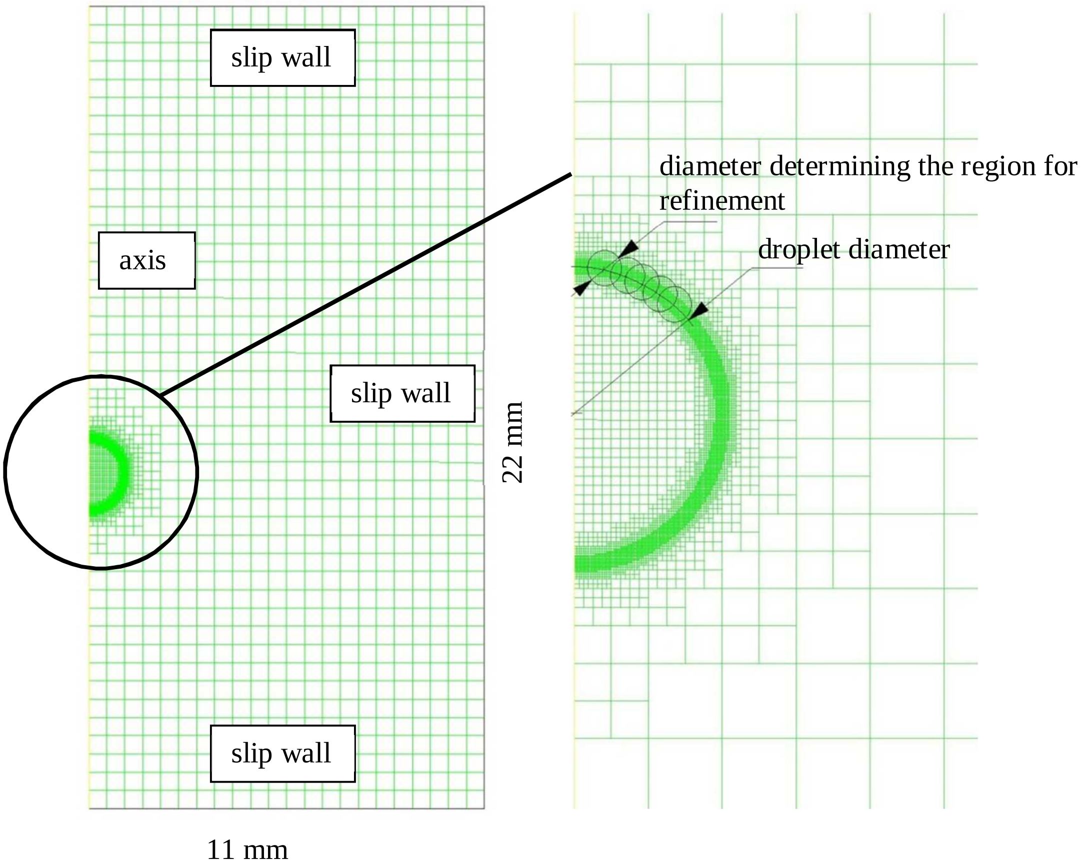


Fig. 16 *Mesh grid and its refinement in the vicinity of the droplet interface; 6 refinement levels, droplet diameter of 2 mm*

The droplet velocity dependencies versus time scale for each case along with the numerical solution of Newton's law are depicted in Fig. 17. The computational results are in a good agreement with the Newton's law solution. It should be noted that no affinity between the results and the mesh quality was found. The most significant discrepancy was found for the finest grid.

Tab. 4 *Droplet diameter of 2 mm; each element size indicates unique grid with appropriate level of refinement*

No. of case	Levels of refinement	Element size [mm]	Time step [s]
1	4	0.03125	1e-05
2	5	0.01563	7e-06
3	6	0.00781	4e-06
4	7	0.00391	2e-06

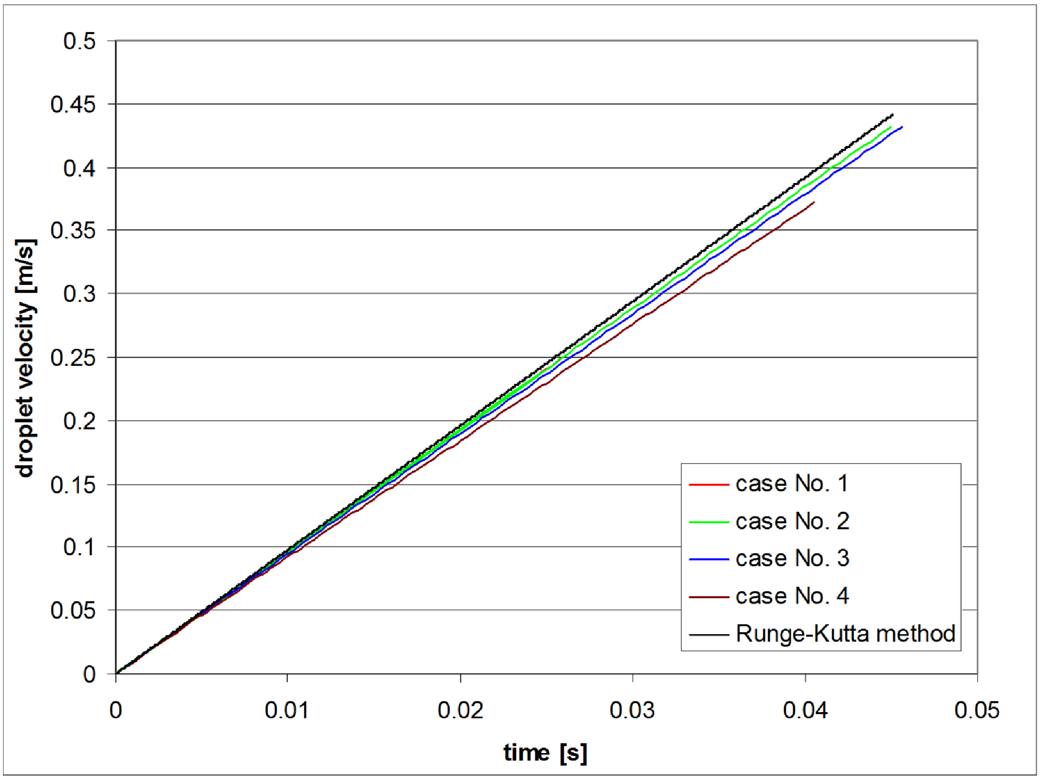


Fig. 17 *Droplet velocity vs. time scale for droplet diameter of 2 mm; four different grids, results compared with numerical solution of Newton's law*

The same model settings and the mesh handling were used in cases with other droplet diameters (0.8 mm, 0.2 mm). In case of the droplet diameter of 0.8 mm, the surface tension model implemented in VOF model exhibited a malicious effect on results. Since the curvature of the interface was 2.5times higher than in aforementioned case with 2 mm droplet, the error in the estimation of the normal to the interface was higher, which lead to more significant spurious currents. This unwished fact was confirmed by the series of three cases (see Tab. 5) with droplet velocity dependent surface tension imposed via the UDF available in APPENDIX VI. The purpose of that UDF is to adjust the surface tension in order to rather artificially minimize spurious currents. The behavior of the droplet velocity is shown in Fig. 18. There is obvious difference between the case No.1 and the case No.3. The higher value of the surface tension is, the lower acceleration of the droplet is observed. As regards the mesh sensitivity issue, there was just negligible difference between the case No.1 and the case No. 5.

Tab. 5 *Droplet diameter of 0.8 mm; surface tension and time step settings*

No. of case	Levels of refinement	Surface tension [N/m]	Time step [s]
1	5	0.011 x droplet velocity	5e-06
2	6	0.011 x droplet velocity	4e-06
3	5	0.001 x droplet velocity	5e-06

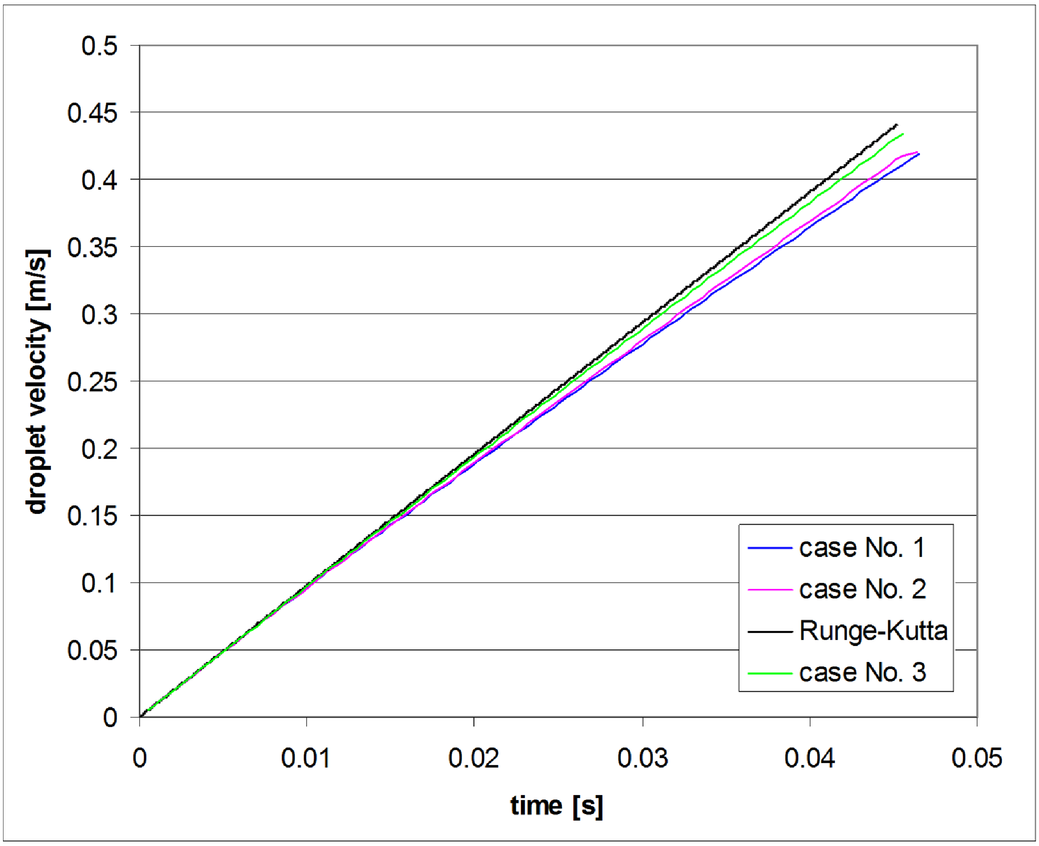


Fig. 18 *Droplet velocity vs. time scale for droplet diameter of 0.8 mm; three different cases, varying surface tension, results compared with numerical solution of Newton's law*

As regards the smallest droplet diameter of 0.2 mm simulated, effects of spurious currents on the droplet velocity were of the greatest importance. In Tab. 6, the list of performed cases is shown with surface tension settings. If the surface tension is switched off, the computational results reliably follow the solution of Newton's law until the droplet surface starts to distort due to lack of surface tension. The higher the value of the surface tension is, the lower the acceleration of the droplet is observed i.e. the spurious currents cause the drag coefficient to rise up (see Fig. 19). In case No. 8, the steepest grow of the surface tension led to the most significant spurious currents that consequently yielded to far more different results from the real ones. The terminal velocity was found to be below the value of 0.2 m/s while the true value was approximately 1.8 m/s.

Tab. 6 *Droplet diameter of 0.2 mm; surface tension and time step settings*

No. of case	Levels of refinement	Surface tension [N/m]	Time step [s]
1	7	0	2.5e-06
2	7	0.18 x droplet velocity	2.5e-06
3	7	0.09 x droplet velocity	2.5e-06
4	7	0.045 x droplet velocity	2.5e-06
5	7	0.0225 x droplet	2.5e-06
6	7	0.011 x droplet velocity	2.5e-06
7	7	0.0011 x droplet	2.5e-06
8	7	0.0001 x droplet	2.5e-06
9	8	0.011 x droplet velocity	1e-06

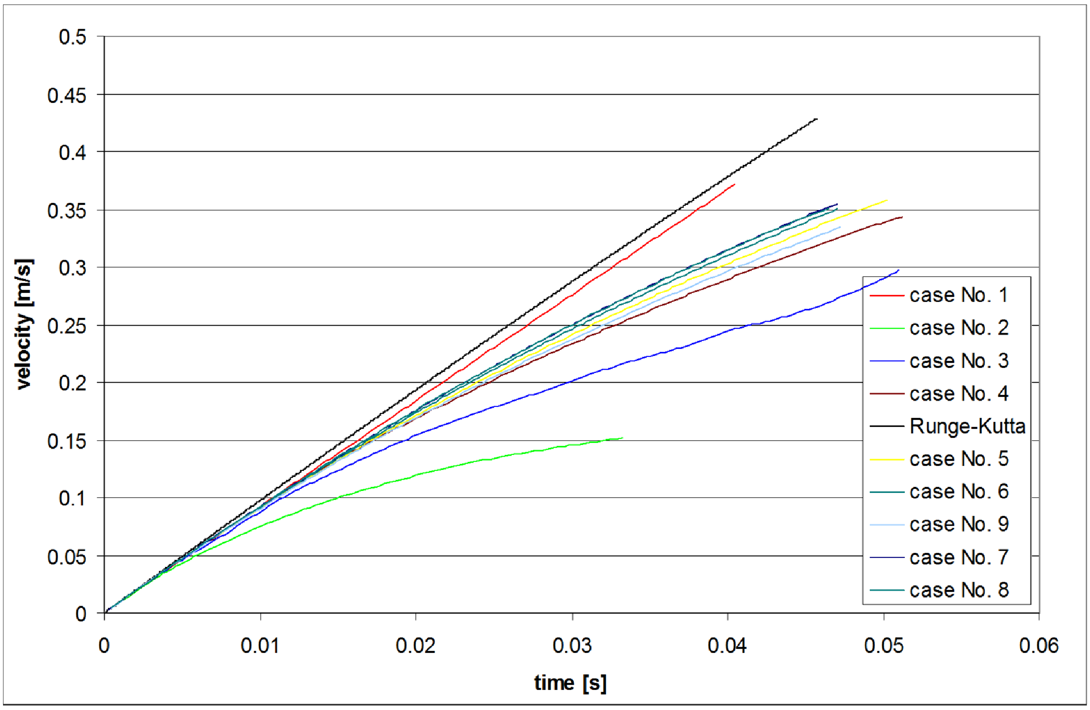


Fig. 19 *Droplet velocity vs. time scale for droplet diameter of 0.2 mm; nine different cases, varying surface tension, results compared with numerical solution of Newton's law*

3.6 Droplet without gravity, spurious currents

In this chapter, 2D droplet of diameter of 2 mm is patched in the center of a rectangular domain 4x4 mm. If gravity is turned off and surface tension is the only force acting on the droplet surface, the velocity field should remain constant, equaled to zero in reality. However, according to the accuracy of the numerical model, spurious currents appear.

Here, the level of spurious currents is studied for CSF model for different mesh sizes. Further, there is question how much accurate normals to interface are. Four different methods are used for calculation of interface normal and curvature consequently. Finally new surface tension model is proposed partially based on Height Functions.

3.6.1 Spurious currents related to CSF model

As mentioned above, cases with four different mesh sizes were tested to reveal spurious currents within CSF model implemented in Fluent. The 2 mm droplet was patched in the center of the rectangular domain 4x4 mm. The basic element size was 2e-04 m. The element size was further refined to test mesh sensitivity. During computation the droplet was not moving since the force of gravity of turned off. The force resulting from surface tension was the only force acting on the droplet.

As regards the best flow time for evaluation, the non-dimensional time was defined as follows:

$$\tau = \frac{t\sigma}{\mu D} \qquad \text{Eq.82}$$

and it was 100.

In [44], the importance of time step was stressed. For high or medium Reynolds number the Brackbill stability condition should be used (Eq.7), however, for low Reynolds numbers and high Capillary numbers different stability condition containing viscosity must be used instead (Eq.8). L_1 and L_∞ errors were used for evaluation and they are defined as follows:

$$L_\infty = \max|v_i| \qquad \text{Eq.83}$$

$$L_1^2 = 2\sum Vv^2\left(F\rho_l + (1-F)\rho_g\right)/m \qquad \text{Eq.84}$$

The L_∞ norm returns the maximal velocity magnitude presented wherever inside the computational domain, whereas the L_1 norm returns the velocity magnitude corresponding to the kinetic energy of the whole system. To get some statistic information, data file was captured around time $\tau = 100$ and the standard deviation was evaluated. Results are shown in Tab. 7. Missing data was not calculated. The L_∞ norms are obviously increasing with mesh refinement. The L_1 norms are contrarily oscillating around the value of 0.05 m/s. In other words, the CSF model does not converge with mesh refinement to more accurate results; it rather amplifies spurious currents no matter which time step condition is used.

Tab. 7 *L_∞ and L_1 errors for velocity*

mesh size [mm]	time step [s]	mean L_∞	std L_∞	meanL_1	stdL_1
0.0002	4.20E-06	0.1823	0.0574	0.0696	0.0303
	4.95E-08	-	-	-	-
0.0001	1.48E-06	0.1218	0.0458	0.0369	0.0221
	2.48E-08	-	-	-	-
0.00005	5.25E-07	0.1597	0.0278	0.0577	0.0178
	1.24E-08	0.1607	0.0372	0.0464	0.029
0.000025	1.85E-07	0.2151	0.0766	0.036	0.0137
	6.20E-09	0.2309	0.0931	0.036	0.0203

3.6.2 Calculation of normals to interface, curvatures

Four different approaches were employed to calculate interface normals. First two approaches simply calculated normals from gradient macro C_VOF_G(c,t) and reconstructed gradient macro C_VOF_RG(c,t). Their definition can be found in [76]. The third approach was the

ALE-like scheme proposed in [43]. Normals were calculated in every computational node from four neighboring cells in 2D. The curvature was calculated at cell centers from the divergence of cell-centered normal and from the derivative of the magnitude of the normal vector. In the fourth model, Height Functions [56] were constructed within each cell containing interface with help of either horizontal (3x7 cells) or vertical (7x3 cells) stencil.

The accuracy of those four procedures was assessed with help of L_1 and L_∞ errors for the angle between a calculated and a true normal. Results are shown in Tab. 8.

Tab. 8 *L_∞ and L_1 errors for angles between calculated normals and true normals*

	L_∞ [°]	L1 [°]
C_VOF_G(c,t)	4.99	1.739
C_VOF_RG(c,t)	2.6	0.954
ALE-like scheme	2.34	0.975
Height Functions	0.763	0.262

From Tab. 8, it is obvious that non-limited gradient of volume fractions gave the most inaccurate normals. Reconstructed gradient and ALE-like scheme gave similar results. Height Functions gave the best results. L_∞ error was 3 times smaller than the L_∞ error for ALE-like scheme and L_1 error was even almost 4 times smaller.

On the basis of these results, only ALE-like scheme and HF approach were used for curvature calculation. The procedures were discussed in detail in [43] and [57], respectively. The accuracy of ALE-like scheme was very poor. The droplet diameter was 2 mm that stood for curvature of 1000. In some interface cells the curvature error was even worse than 100%. As regards curvatures obtained within HF approach, it gave far more precise results. If the most normal direction was either vertical or horizontal, the error was approximately 5%. However, when the normal direction was getting more diagonal the errors were increasing almost up to 40%.

To get more precise results especially regarding the interface curvatures, a new method was imposed. An illustrative scheme is shown in Fig. 20.

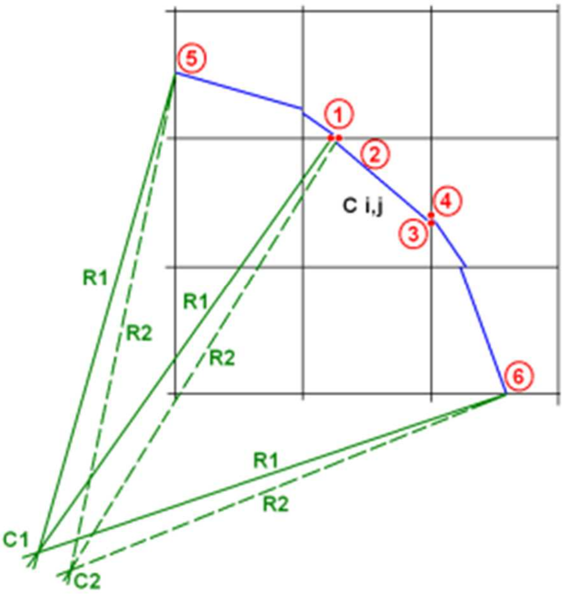


Fig. 20 *Scheme showing how the curvature is estimated*

Each normal to the interface was firstly estimated according to HF technique. At next step, around each cell containing the interface, a 3x3 stencil was constructed (see Fig. 20) and the curvature was calculated from radii where each radius corresponds to the circle circumscribed around three points of a piecewise linear interface in the vicinity of the cell of interest $C_{i,j}$. Four circles were constructed in such a way that points 5 and 6 were the same for each circle. Only the point *P* between them was varying from 1 to 4. Afterwards, both, line *a* from point *5* to point *P* and line *b* from point *6* to point *P*, were formed and line slopes were calculated as:

$$k_a = \frac{y_P - y_1}{x_P - x_1}, \qquad \text{Eq.85}$$

$$k_b = \frac{y_P - y_6}{x_P - x_6}. \qquad \text{Eq.86}$$

Than position of each circle center $C_i\ (x_i, y_i)$ is given by following equations:

$$x_i = \frac{k_a k_b (y_5 - y_6) + k_b (x_5 + x_P) - k_a (x_P + x_6)}{2(k_b - k_a)}, \qquad \text{Eq.87}$$

$$y_i = \frac{k_a (y_5 + y_P) - k_b (x_6 + x_P) + (x_5 - x_6)}{2(k_a - k_b)}. \qquad \text{Eq.88}$$

In this way, four different, but very near circle centers are calculated and the only one circle center *C(x,y)* is evaluated as an arithmetic average given by following formula:

$$x = \frac{1}{4} \sum x_i\ . \qquad \text{Eq.89}$$

Likewise, the *y* coordinate of circle center is calculated. The curvature is calculated as an inverse value of circle C radius, as follows

$$\kappa = \left((x_5 - x)^2 + (y_5 - y)^2 \right)^{-0.5} . \qquad \text{Eq.90}$$

Knowing the circle center C and the center of cell $c_{i,j}$, the normal previously obtained by HF approach is recomputed.

3.6.3 Surface tension as volume source term, pressure correction equation

In Fluent, the surface tension is model as Continuous Surface Force (CSF) and is given by the Eq.22 imposed as a source term to momentum equation. These source terms are not non-zero only in interface cells but also in nearby cells and this leads to smeared pressure jump across the interface. Unlike CSF model, the surface tension model presented here put non-zero source terms only into interface cells and it is defined by the following formula:

$$F_{s-t} = \frac{\sigma\kappa \cdot l}{V} \qquad \text{Eq.91}$$

In each interface cell, this source term stands for exact pressure jump across the interface with the length *l*. In Fluent, pressure is discretized in face centers, while surface tension in cell centers that consequently causes pressure imbalance. For an exact balance between surface tension force and pressure gradient surface tension has to be imposed in both, predictor and corrector equation, and surface tension has to be discretized in face centers instead of cell centers. As discussed in section 2.1.4, the Ghost Fluid Method can be used to determine sharp boundary condition so that the particular face center contains non-zero surface tension only if the distance function changes its sign.

4 Cooling process using CFD, atomized sprays

In this section, the flow inside two manufactured cooling nozzles is firstly described with the help of CFD methods. The first nozzle is the full-cone nozzle and the second nozzle is the flat jet nozzle. Furthermore, the water jet of the last one was studied. First of all, the Euler-Euler approach was used to model the flat water jet within a very simple rectangular domain. Since the Euler-Euler model available in Fluent does not offer the possibility of droplet breakup modeling by default, a simple breakup model was developed, imposed in the way of UDF and tested on a solid jet in 2D. However, the breakup model developed came out to be rather tricky and tedious than to give reliable results. Therefore, the breakup model was abandoned. The aforementioned case with the simple rectangular domain was further replaced by a much more complex geometry found in a first cooling section above foot rolls in a real continuous slab caster (see APPENDIX VII). Two approaches for two-phase flows were employed. First, the Euler-Euler model was believed to be the most appropriate; however, the solution was very computationally expensive and slow. For that reason, the Euler-Lagrange model was used instead. Similarly to the Euler-Euler model, also the Euler-Lagrange model was firstly tested within the simple rectangular domain. Afterwards, the full 3D geometry with the mold bottom, the foot roll, the slab surface was taken into account. To simplify the task, the "cold" flow was solved at the beginning i.e. no heat transfer was considered. Unlike the Euler-Euler model, the water jet breakup was successfully simulated and the calculated droplet size distribution fitted well the experimental data. The Euler-Lagrange model was coupled with the multiphase VOF model because there was a continuous water bulk in the foot roll gap and it could not be modeled otherwise. Thereafter, the vaporization and the boiling of droplets were simulated.

4.1 Flow inside cooling nozzles

Knowledge of internal flow inside nozzle is important for several reasons. Firstly it provides a nozzle designer with information about velocity, pressure, and temperature field that is very helpful for nozzle optimization. Further, results can be used for the reduction of pressure losses, the turbulence pick-up. Last but not least, results can be used to generate a profile file that is later imposed as the inlet boundary condition.

In next sections, the internal flow inside two different nozzles was solved in FLUENT. The first nozzle [77] is a full-cone nozzle and the simulations were done within the author's diploma thesis. The second nozzle is a flat jet nozzle that is a part of a secondary cooling system in a real continuous caster.

4.1.1 Full-cone nozzle

The nozzle (Lechler 460.844) belongs to the category of full-cone nozzles, whose water jet pattern is a full cone and the droplet and the velocity distribution depends on the nozzle interior geometry. The interior of this particular full-cone nozzle is relatively complicated and too confined to be studied experimentally. The internal nozzle geometry is shown in Fig. 21. A special insert is pressed to the main chamber and makes the water swirl that consequently causes the formation of the full-cone jet pattern. The geometrical model was built in 3D CAD modeler Solid Works. Afterwards, the mesh processing was performed in the preprocessor Gambit. The solution itself was done in the commercial CFD package Fluent. This problem was solved using the single phase model, since there was only water presented and besides, the multiphase modeling would be difficult and rather tedious. The total number of tetrahedral elements was 732 687 within the mesh grid. According to the Equisize Skew quality check function implemented in Gambit, the quality of the worst element was 0.81, which is still sufficient to get converged solution. As regards equations that were solved, momentum equations for all three dimensions, the continuity equation and two transport equations for k-epsilon turbulence model were solved iteratively.

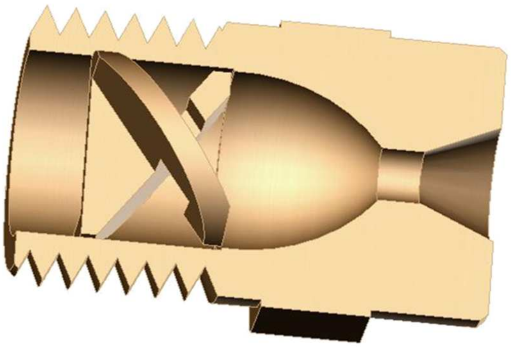


Fig. 21 *The interior of Lechler 460.844*

The description of BC is given in Tab. 9. The flow was considered to be 3D, turbulent and unsteady. Other important settings are shown in Tab. 10. In Fig. 22 and Fig. 23, the contours of velocity are shown in two perpendicular sections. This study revealed a wake formation right behind the angled tips of the special insert and the water circulation inside the chamber.

It must be noted that the result credibility near the nozzle exit is doubtful, since in fact, there is probably water-air mixing in the conical region i.e. a two-phase flow, which was not considered within the computational model. Therefore, the only results in the proximity of the special insert were believed to be valuable.

Tab. 9 *Description of boundary conditions*

Name	Type of BC	Position
wall	Wall	Entire internal nozzle housing
inlet	Velocity inlet	Inlet diameter of 14 mm
outlet	Pressure outlet	Nozzle exit into atmosphere (overpressure of 0 Pa)
water	Fluid	Entire domain contains water

Tab. 10 *Other settings*

Water as a continuum	ρ=998.2 kg/m^3, η=0.001 kg/(m.s)
Turbulence model	k-epsilon, realizable, non-equilibrium wall functions
Unsteady flow	2nd order implicit, Δt=1e-06 s
Velocity inlet	$c = \frac{4Q}{\pi d^2} = 1.95 m/s$
Pressure outlet	Atmospheric pressure, turbulent intensity of 2%, hydraulic diameter of 0.014 m
Discretization scheme of momentum equation	2nd order upwind

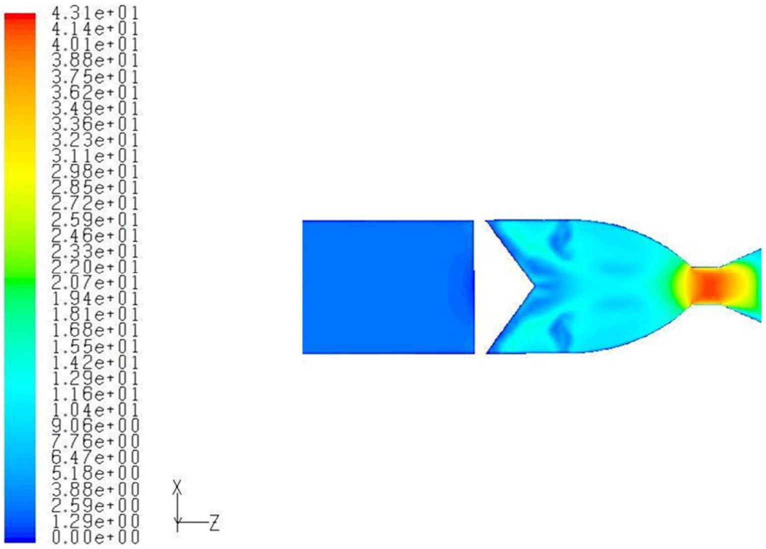


Fig. 22 *Contours of velocity in longitudinal section going through nozzle axis*

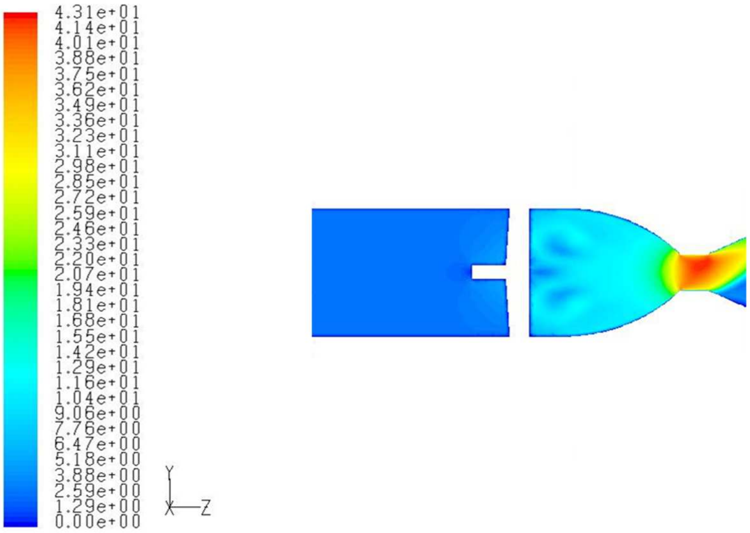


Fig. 23 *Contours of velocity in longitudinal section perpendicular to the one shown in Fig. 22*

4.1.2 Flat jet nozzle

The second nozzle here presented (Lechler 600.429.16.33) belongs to the type of a flat jet nozzle that produces a jet of flat jet pattern. The nozzle geometry is given introduced in APPENDIX VIII. The cooling liquid is again the water. The flow rate through the nozzle was 16.52 l/min. The nozzle inlet was connected with the pipe of the inside diameter of 7.8 mm, thereof the mean inlet velocity can be considered to be 5.76 m/s. The water properties depend mainly on its temperature. The inlet water temperature ranges between (40-50°C). For the average temperature of 45°C, basic properties are listed in Tab. 11.

Tab. 11 *Water properties at atmospheric pressure and temperature of 45°C*

Density [kg/m^3]	990.22
Dynamic viscosity [kg/m-s]	0.000596
Saturation vapour pressure [Pa]	9582
Specific heat capacity [J/kg-K]	4180
Specific latent heat of vaporization [kJ/kg-K]	2260
Thermal conductivity [W/m-K]	0.61

The nozzle inside is quite simple. The water is entering through the annular inlet, passing through the sharp edged constriction to the chamber and the elliptic orifice, respectively. Further, the water is emitted to the ambient air as discussed later.

The minimal cross-sectional area is in the place of the constriction (12.6 mm^2) and in the nozzle orifice (9 mm^2). The local values of velocity of 22m/s should be exceeded in the constriction. Due to high water velocities and the sharp edged constriction, the flow must be turbulent with significant mixing layers, adverse pressure gradients and boundary layer separation. Firstly, five different meshes were employed to detect the possible flow sensitivity. The geometry of the whole domain and also other settings were preserved. The computational domain is presented in Fig. 24.

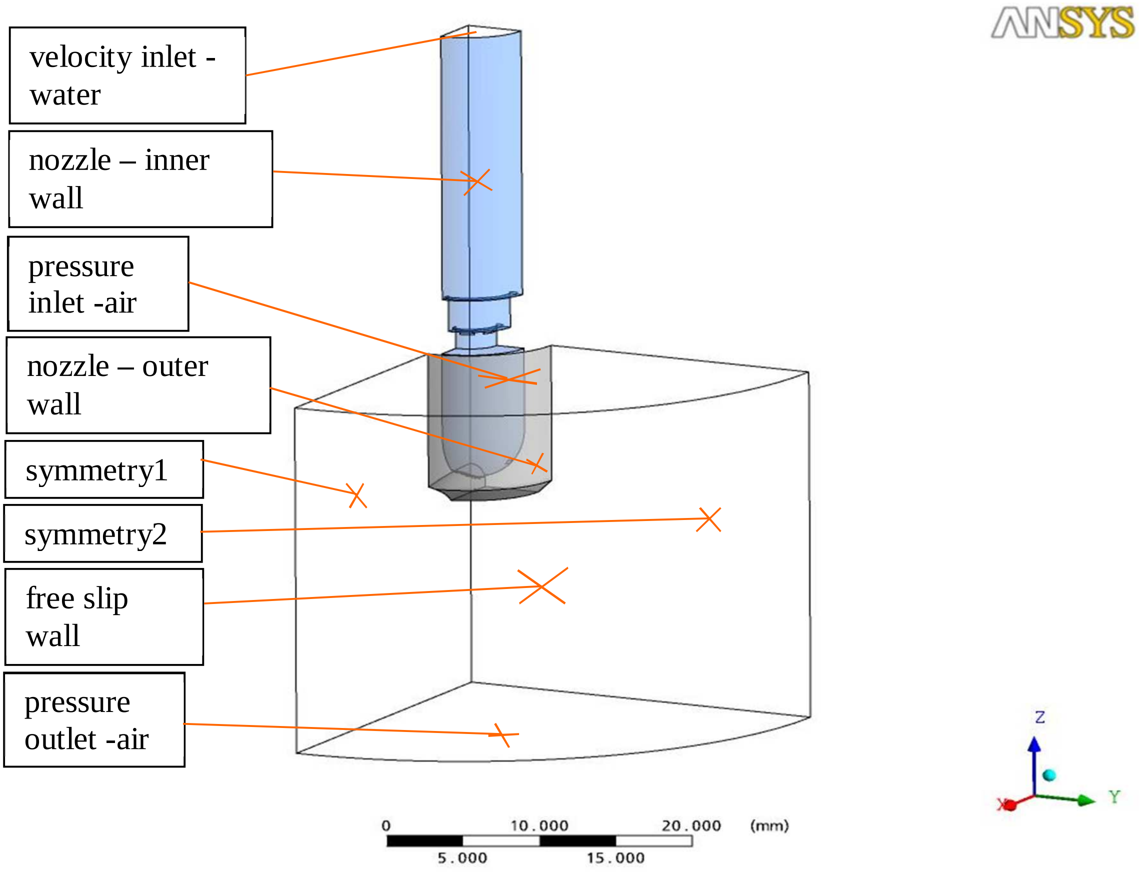


Fig. 24 *Computational domain of Lechler, no. 600.429.16.33*

Due to planes of symmetry, only the quarter of the full geometry could be modeled. The water was a secondary phase and the air was a primary phase. Since the atomization takes place and very fine droplets are formed, the one of Euler homogenous models had to be used. Due to a relative simplicity, the Mixture model was used instead of the full Euler model. Further, the relative velocity between phases was not considered because of the assumption of its negligible effect on the flow inside the nozzle. The turbulence was modeled using the k-ε realizable model with non-equilibrium wall functions.

As inlet boundary conditions were used the velocity inlet (5.76 m/s) with the volume fraction of water of unity and the pressure inlet (0 Pa) with the volume fraction of air of unity.

The only outlet boundary condition was the pressure outlet (0 Pa) with the backflow volume fraction of water of zero.

All nozzle walls were no slip walls. The free slip wall was used to define the open-air region. The turbulence properties at inlet and outlet boundaries were estimated instinctively (The turbulent intensity in range of (2-4%), the hydraulic diameter of 0.025 m).

The starting mesh contained hexagonal elements inside the nozzle and tetragonal elements outside the nozzle. Several meshes refinements were carried out using y+ or phase gradient adaption subsequently. The example of refined mesh in ZX plane is shown in Fig. 25. The refinement process is presented in Fig. 26.

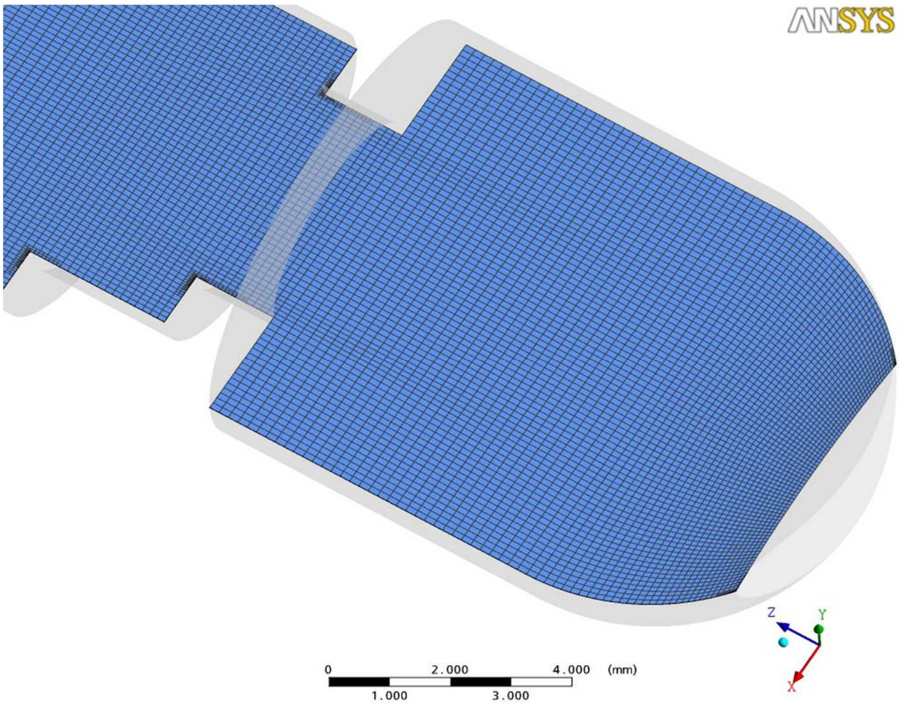


Fig. 25 *Hexagonal mesh inside nozzle*

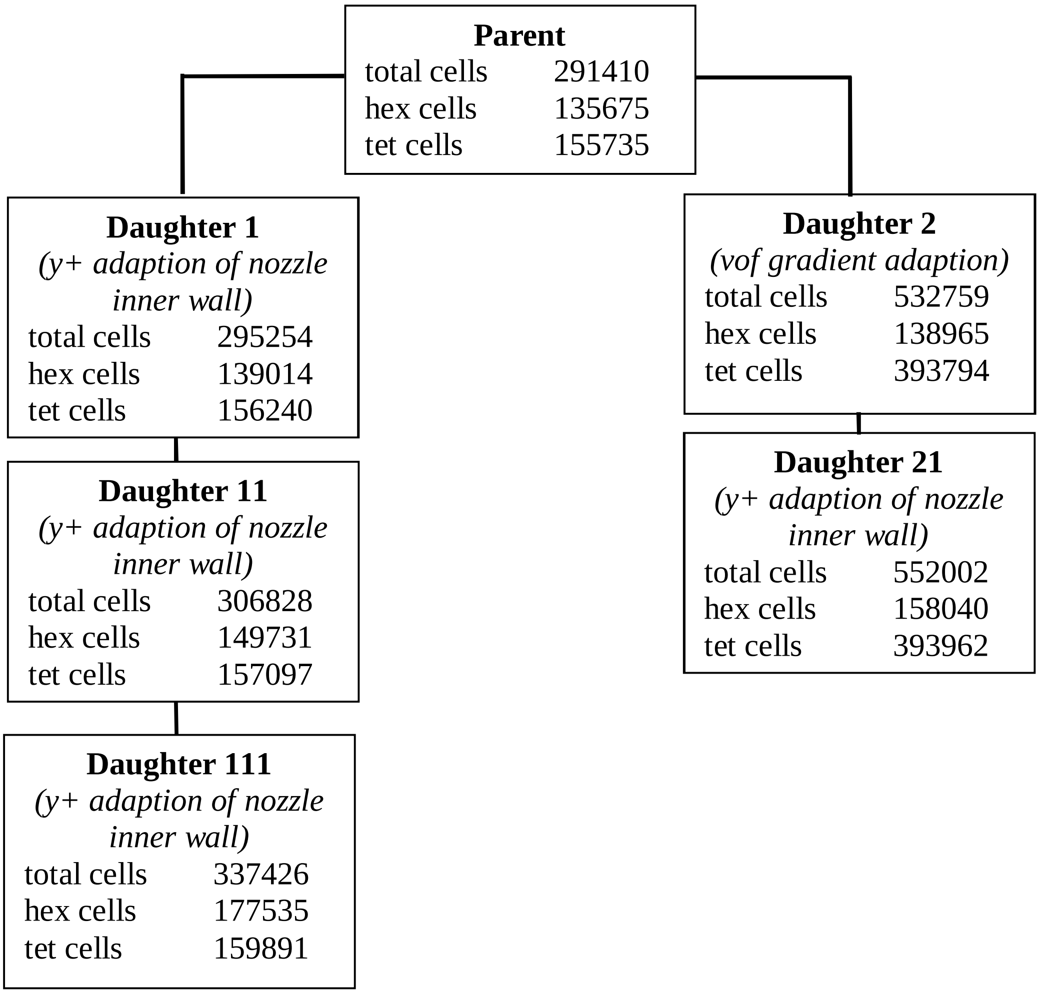


Fig. 26 *Order of refinement procedure*

It must be emphasized that phases shared the same velocity field and thus, flow fields should be relevant only inside the nozzle. In Fig. 27 - Fig. 30, velocity fields, total pressures, fields of turbulent kinetic energy are shown just for one case in different nozzle cross-sections and the brief discussion is provided.

Both, contours and vectors of velocity, are shown in ZX plane in Fig. 27 and in the ZY plane in Fig. 28. If one follows the stream near the wall, he will encounter a first recirculation of velocity at the top of the insert. The static pressure is increasing and the turbulent boundary layer is separated. Afterwards, the water crosses over the first sharp edge of the insert. Due to the pressure gradient, the velocity recirculation and the boundary layer separation take place again. The prompt change of cross-sectional area is 0.59. Near the second sharp edge of the insert the water is rapidly accelerated. There is the tiny wake behind that edge. That wake is also the place of the lowest static pressure inside the nozzle. However, the saturation vapor pressure of 9582 Pa is not exceeded and thus, no cavitation can appear inside the nozzle. The prompt change of cross-sectional area is 0.44. The water passes the tiny wake and sticks back to the wall. Further, the water, which goes pass the third sharp edge, enters the chamber and the crucial mixing layer is formed due to a certain velocity difference. It means there is also a velocity recirculation in the chamber. Due to pressure losses in recirculation area the mixing layer is supported. Velocities in the upper part of the recirculation region are almost zeros.

Since the nozzle has no axis of symmetry, the velocity field is also not symmetric as clearly shown in Fig. 27 - Fig. 28.

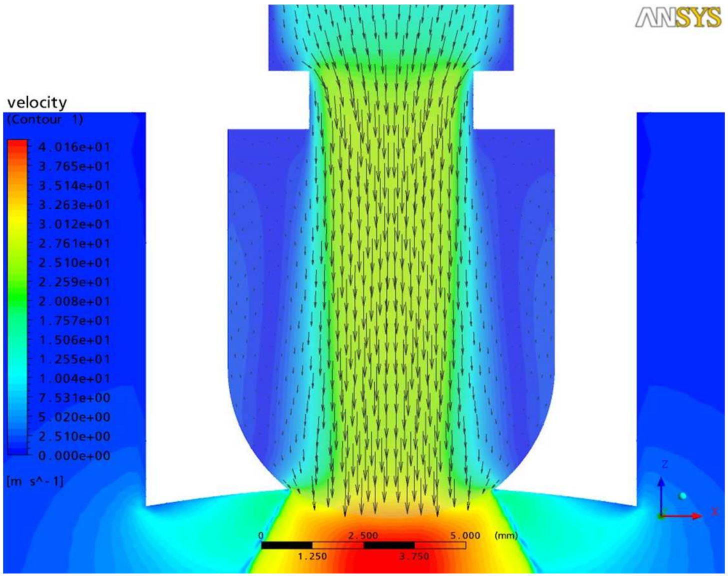


Fig. 27 *Contours and vectors of velocity in ZX plane*

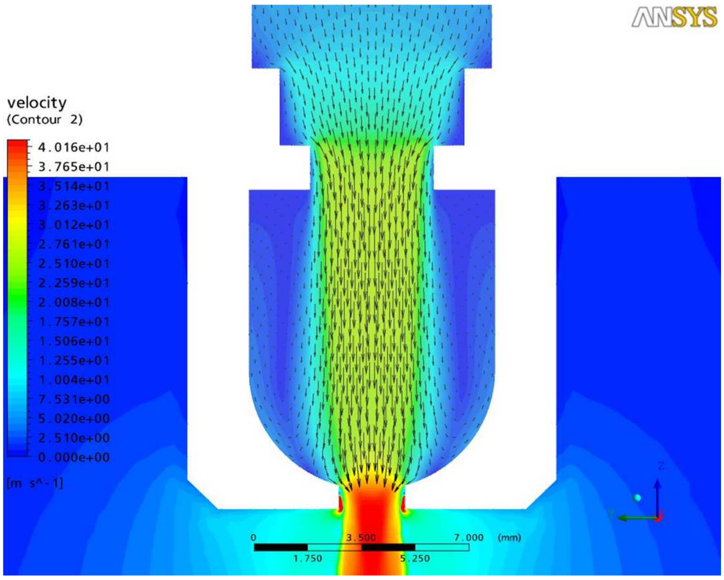


Fig. 28 *Contours and vectors of velocity in ZY plane*

Concerning the mesh sensitivity, there was not noticeable discrepancy among the velocity fields of all cases considered. However, if it is observed in more details, small differences can be found there. For instance, there is the difference of 5% of velocity x-component in the cross-sectional area 3mm far from the exit. It should be noted that velocity x-components are in order of 0.1 m/s contrary to velocity z-components, which are 100 times higher. Moreover, the spray angle is affected mainly by velocity y-components and the difference of velocity y-components was less than 5%. The effect of VOF phase gradient adaption refinement on the velocity field was found to be negligible.

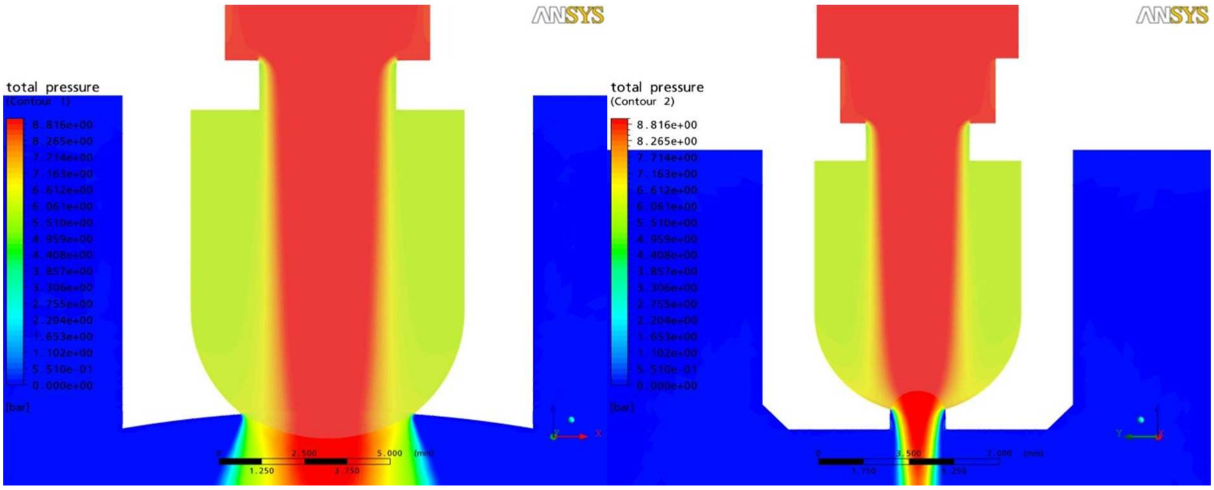


Fig. 29 *Contours total pressure in ZX plane and ZY plane*

Since the near wall refinement caused wall functions to be employed from the lower y+, the pressure losses increased, especially in the place of the special insert. From all cases, the most of pressure losses is obviously found in the main chamber.

In Fig. 30, contours of turbulent kinetic energy (tke), which represent velocity fluctuations (see Eq.92), are shown.

$$tke = \frac{1}{2}\overline{u_i'^2} \qquad \text{Eq.92}$$

As clearly seen, the tke remains quite low in the stream core contrary to regions of a velocity recirculation. As the mixing layer width is increasing, the tke is increasing. The highest tke is located on the edge of the nozzle orifice, where the water is further accelerated to the outside on one hand and on other hand, the water is dragged back to the chamber.

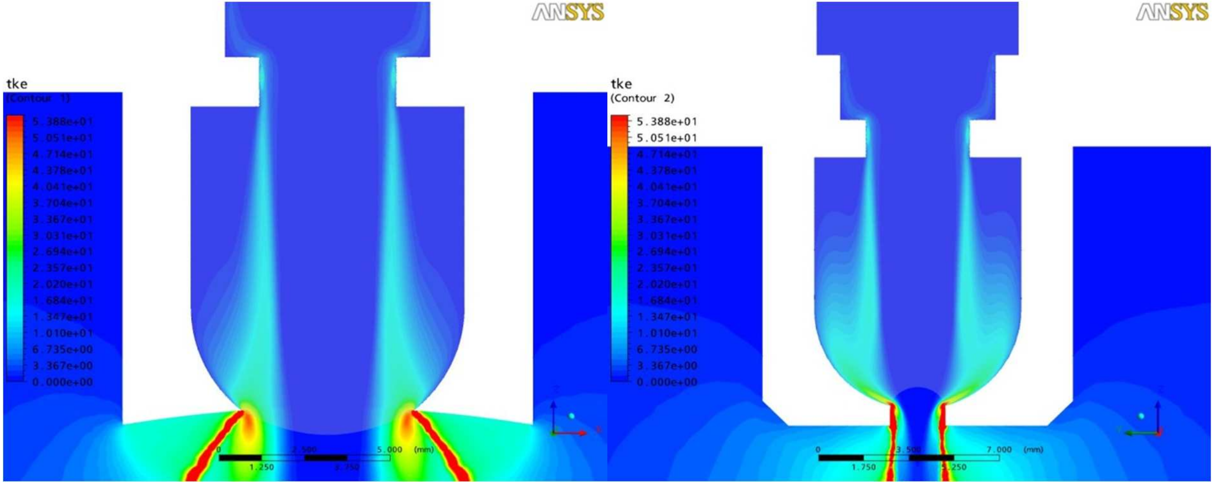


Fig. 30 *Contours turbulent kinetic energy in ZX plane and ZY plane*

If compare results for each case, the tke is suppressed with the mesh refinement. Especially, it is noticeable in the mixing layer.

In Fig. 31, the contours of velocity z-component are depicted in the cross-section located 3 mm far from the nozzle exit. It is remarkable that the velocity field of the stream core is still annular, whereas the lateral backflow is non-symmetric. Since the velocity z-component is quite high, there is no big difference between each case. As regards the velocity x-component and z-component, the velocity distributions were almost the same.

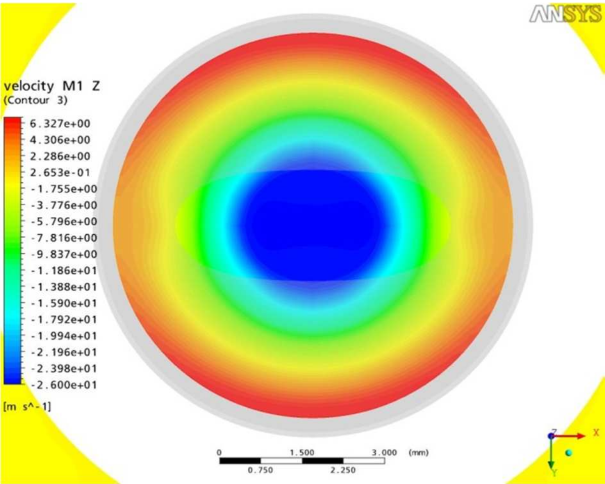


Fig. 31 *Contours velocity z-component 3mm far upstream from nozzle exit*

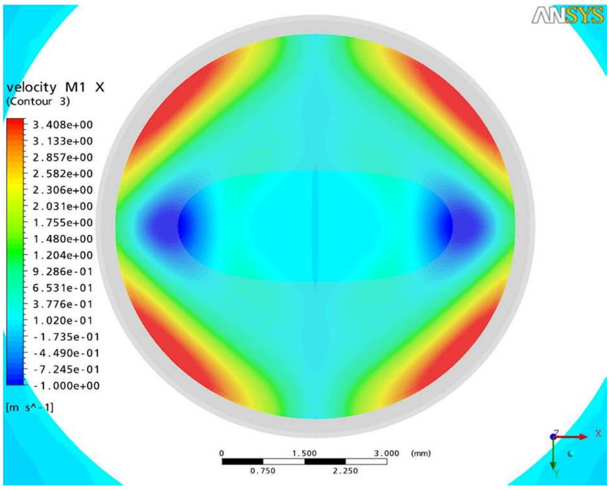


Fig. 32 *Contours velocity x-component 3mm far upstream from nozzle exit*

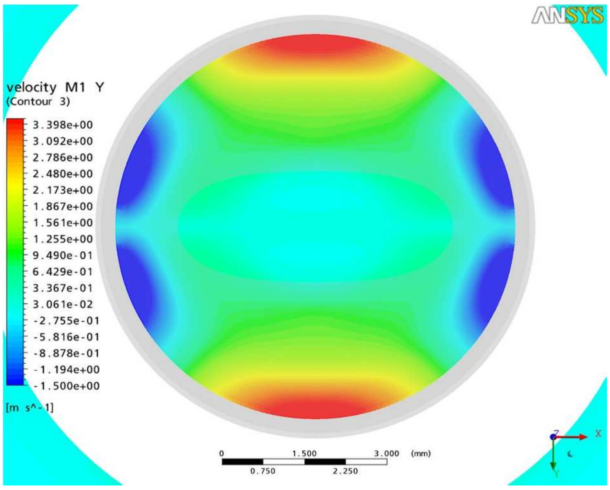


Fig. 33 *Contours velocity y-component 3mm far upstream from nozzle exit*

In summary, all of the simulations have shown a significant effect of the special part on the flow inside the nozzle. The special part also causes higher pressure drop of the nozzle. However, it can probably provide more stable jet and more uniform atomization, consequently. Simulations proved the flow to be quite dependent on mesh refinements and thus, boundary layers should be resolved carefully. Wall functions were employed to solve boundary layers, because it is numerically robust and not computationally demanding. On the other hand, errors in boundary layers near sharp edges and near walls with low Reynolds numbers could be significant.

Concerning the model of turbulence, all of simulations were performed with the k-ε mixture multiphase turbulence model. It means that always only one equation for both, the air and the water, was solved. Since there was no air inside the nozzle, it was found as the most appropriate.

It should be noted that the flow was solved as absolutely symmetric because of 2 planes of symmetry. It means no flux through it. Using multiphase modeling with the employing symmetry planes can be a bit tricky, because the normal velocities in near wall cells gravity centers could be different for each phase and thus, there can be artificial accumulation of one of the phases.

Further, the flow inside the Lechler nozzle, no. 600.429.16.33 was solved using the **Euler-Euler model**, because it was difficult to achieve convergence with using the **Mixture model** and slip velocity switched on. However, it was expected that Euler-Euler model will increase the computation time, since it solves transport equation for each phase separately.

The flow inside the nozzle was simulated using Euler-Euler multiphase model turbulence model with default settings was used for turbulence modeling. In the first case, the mixture k-ε model based on mass weighted transport equations for both k and ε was taken into account. In the second case, the per phase k-ε turbulence model solving turbulence fields separately for both the primary and the secondary phase was employed. In Fig. 34, contours of volume fraction, x-component, y-component, z-component velocity for mixture and per phase k-ε turbulence models are shown in the cross section through the nozzle bottom, respectively. The centered contours are always for mixture k-ε model and contours below are for per phase k-ε model. The profile of water volume fraction is narrower for per phase model. Contours of x-component velocity signify faster spreading of water for per phase model. Contours of y-component velocity causes higher focusing to jet symmetry plane for per phase model and thus, also signify faster spreading denoting larger spray angle.

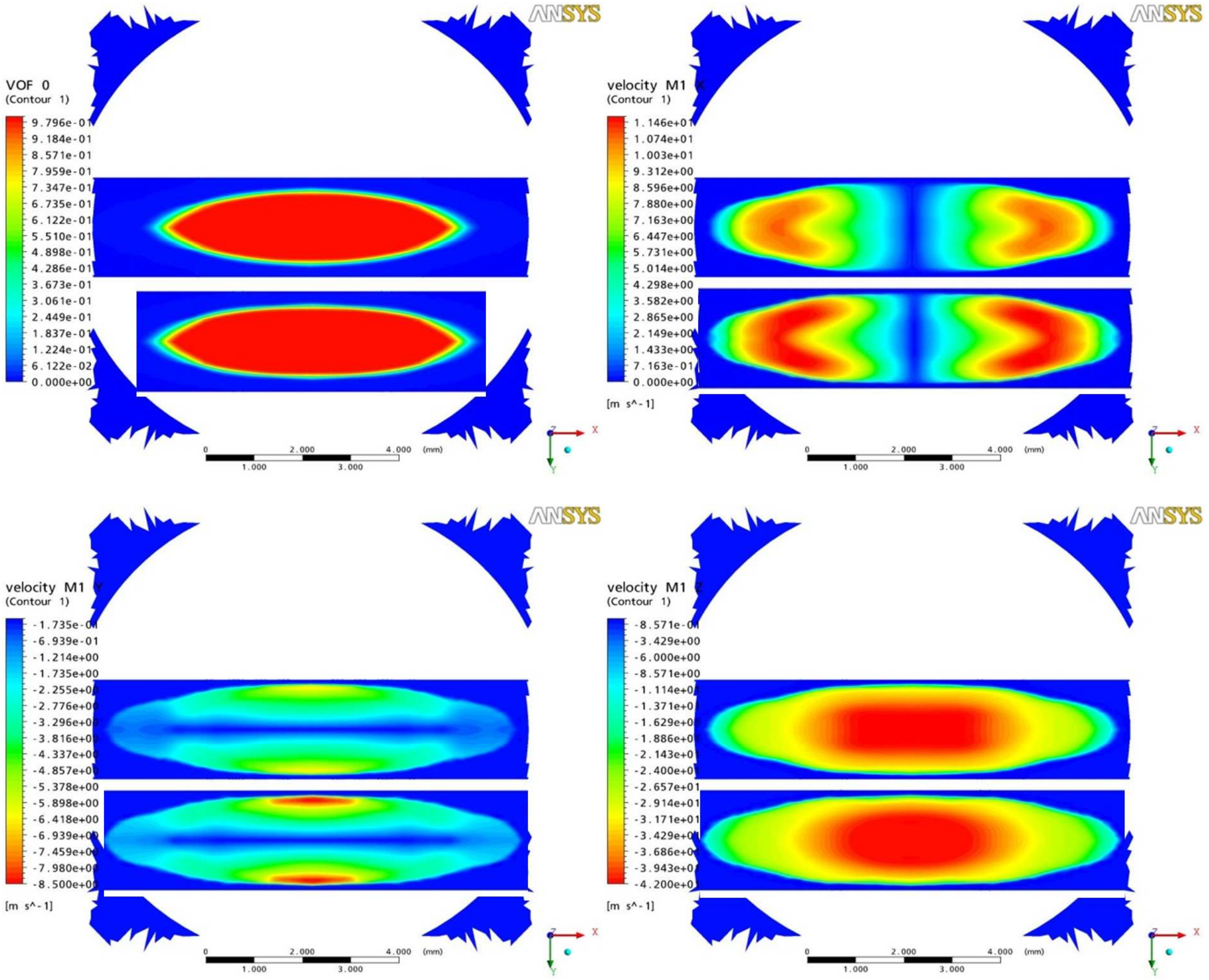


Fig. 34 *Contours of volume fraction, x-component, y-component, z-component velocity for mixture and per phase k-ε turbulence models, respectively*

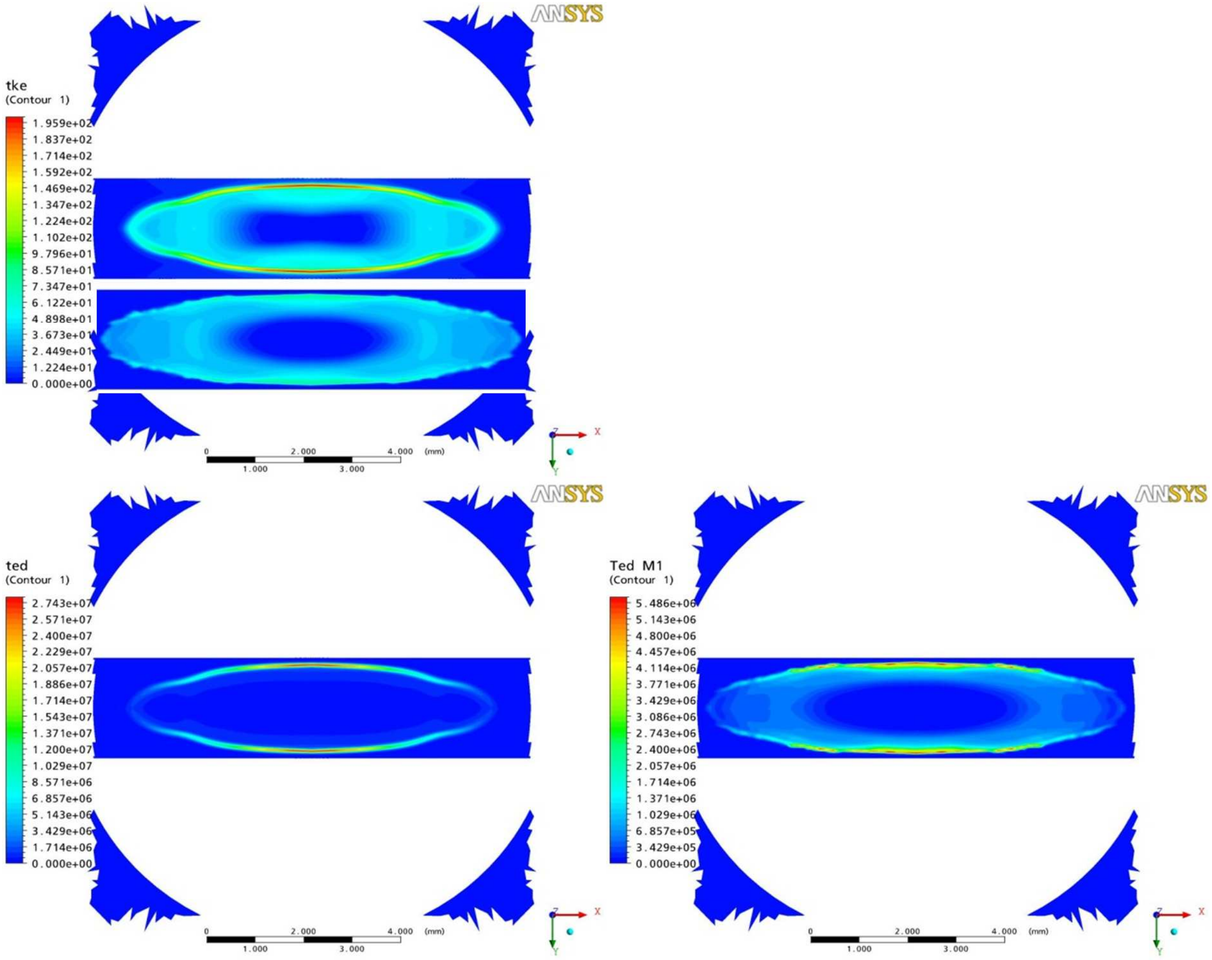
As expected, turbulent kinetic energy dominates in the shear layer between water and air (see Fig. 35). Maximum values of turbulent kinetic energy are reached near the flattened surface of the water jet. In the case of mixture k-ε model, the maximum turbulent kinetic energy of 200 m^2/s^2 is two times higher than maximum turbulent kinetic energy in the case of per phase k-ε model. Naturally, if the water is injected through the orifice to the ambient then, due to completely different fluid densities, different turbulent fields must exist for each fluid. High density ratios can lead to over prediction of turbulent properties such as turbulent intensity downstream and thereof, for instance unrealistic velocities could appear. Thus, results from the calculation with per phase k-ε turbulence model were used for following calculations. Velocity fields, k, ε and water volume fraction fields were stored in the square section inscribed into the bottom of the nozzle and imposed as a velocity inlet boundary condition in next calculations.

$$F_{lift,i} = -C_l \rho_{air} \alpha_{droplet} \varepsilon_{ijk} \left(v_{air,j} - v_{water,j}\right)\left(\varepsilon_{kmn} \frac{\partial v_{air,m}}{\partial x_n}\right)_k \qquad \text{Eq.93}$$

A lift force acts on the droplet only in direction perpendicular to its motion. The default value of lift coefficient in Fluent is 0.5; however, it is valid only for a clean bubble in an inviscid flow. Experiments of Beyerlein have shown lift coefficient to be strongly dependent on volume fraction and is almost zero for moderate volume fractions. The modified lift coefficient is given by the following formula:

$$C_l = 6.51 \times 10^{-4} \alpha_d^{-1.2}, \qquad \text{Eq.94}$$

where the α_d is a droplet volume fraction.



mixture k-ε turbulence model per phase k-ε turbulence model

Fig. 35 *Contours of turbulent kinetic energy, turbulent dissipation rate for mixture and per phase k-ε turbulence models, respectively*

The water droplet is supposed to move along the z-axis. Considering only the z-component of air velocity to be nonzero and rates of its change along x and y-axis to be the same, the total lift force F_{lift} is given by the following formula:

$$F_{lift} = C_l \rho_{air} \alpha_{droplet} \sqrt{2} \left| (v_{air} - v_{water}) \frac{\partial v_{air}}{\partial x_1} \right|, \quad \frac{\partial v_{air}}{\partial x_1} = \frac{\partial v_{air}}{\partial x_2} \qquad \text{Eq.95}$$

The lift force is directly proportional to the density of primary phase (air). Since the air density is about 1 kg/m3, the lift force is the only function of the relative velocity between the primary and the secondary phase and the rate of air velocity change. Let us consider the relative velocity to be in order of unity, which is incidentally noticeable from simulations. Gradient of air velocity in x direction where moderate volume fraction of water is present is in order of thousand. In this case, the lift force is in order of hundred i.e. one order smaller than gravity and one or two orders smaller than the drag force. Since the lift force would be much smaller than the drag force and the gravity force and moreover, the value of lift coefficient is still not clear in water droplet dispersions, therefore, the lift force was neglected.

4.2 Flow outside cooling nozzles, spray modeling in continuous casting

4.2.1 Modeling of water jet breakup within Euler-Euler model

The Euler-Euler model, which was previously discussed in chapter 2.2, belongs to widely used homogeneous multiphase models implemented doubtlessly in all of CFD packages working within the finite volume method (FVM).

Regarding the exchange of momentum, the droplet diameter is used in calculation of the drag force, which is then implemented into the momentum equations as a source term. In Fluent, there is no simple breakup model available within the Euler-Euler model. Despite the lack of such a simple breakup model, the Population Balance Model (PBM) [78], which is based on the Monte Carlo model, exists and is represented by the Population Balance Equation shown below.

$$\begin{aligned}\frac{\partial}{\partial t}\left[n(V,t)\right]+\frac{\partial}{\partial x_i}\left[c_i\cdot n(V,t)\right]+\frac{\partial}{\partial x_i}\left[G_V\cdot n(V,t)\right]=&\frac{1}{2}\int_0^V a(V-V',V')n(V-V',t)n(V',t)dV'-\\&\int_0^\infty a(V,V')n(V,t)n(V',t)dV'+\\&\int_{\Omega_V} vg(V')\beta(V|V')n(V',t)dV'-g(V)n(V,t)\end{aligned} \quad \text{Eq.96}$$

The aforementioned mentioned equation is the transport equation for the number density function $n(x_i,\phi,t)$. The third term on the left hand side of Eq.96 stands for the growth term. On the right hand side, the terms stand for the birth due to the aggregation, the death due to aggregation, the birth due to the breakage, and the death due to the breakage, respectively. The using of PBM requires an extensive experience, an extra license that was not available. To make matters worse, the convergence behavior of the Euler-Euler model itself is poor. The convergence is further deteriorated taking into account the PBM since some other extra equations have to be solved. Therefore, the PBM was not applied for the modeling of water jet and simpler breakup models were rather searched for.

There were many breakup models found in literature [73] [79]. Some of them considered a critical dimensionless numbers such as Weber number, Rabin number as the impulse for droplet breakup. On the contrary, some of them do not specify these critical numbers, but solve some additional equations instead (TAB model, WAVE model etc.)

TAB model (Taylor analogy breakup model) is based upon analogy to mechanical oscillator [73], [74]. The equation of damped, forced oscillator is given by the following formula:

$$m\,{}^{d^2y}\!/_{dy^2}+b\,{}^{dy}\!/_{dy}+ky=F, \quad \text{Eq.97}$$

where the stiffness k is represented by the surface tension σ in the case of droplet. The damping coefficient accounts for effects of viscosity and forces of inertia stand for forced oscillating. The breakup of droplet is supposed when the droplet distortion y exceeds the value of unity. In other words, the breakup occurs when the droplet distortion is equal to the half of radius. The size of new droplets is derived from the energy conservation provided non-disturbed and non-oscillating. Further, the size distribution is supposed to follow Rosin-Rammler distribution with a spread factor of 3.5. The smoothness depends on number of parcels specified. The TAB model is found to work well within low Weber numbers.

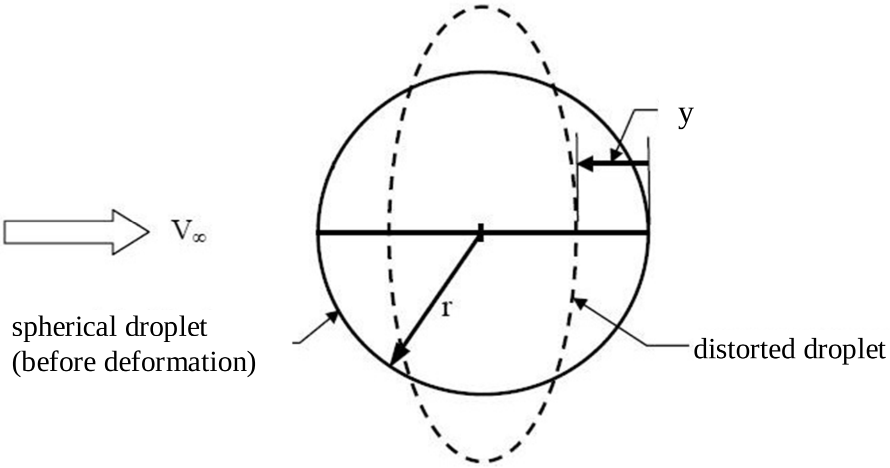


Fig. 36 *Idea of droplet distortion within the TAB model*

Wave model (also Blob jet model) is based on the wave stability atomization theory [79]. It assumes blobs of certain diameter to leave the exit. The size of first blob is equal to the exit diameter. New droplets are formed from the parent droplet and their size is proportional to the wave length of the fastest growing or most unstable surface wave. The mass of new droplets is then subtracted from the parent droplet. The change of diameter of parent droplet is supposed to follow this equation:

$$\frac{da}{dt} = -\frac{a-r}{\tau}, \quad (r \le a), \qquad \text{Eq.98}$$

where τ is the breakup time defined as:

$$\tau = 3.726 B_1 a / \Lambda\Omega, \qquad \text{Eq.99}$$

where Λ is the most probable wavelength and Ω is the maximum grow rate. The radius of new droplets is assumed to follow the following equations.

$$r = B_0\Lambda \quad (B_0\Lambda \le a) \qquad \text{Eq.100}$$
$$r = \min\begin{cases} \left(3\pi a^2 U / 2\Omega\right)^{0.33} \\ \left(3a^2\Lambda / 4\right)^{0.33} \end{cases} \quad (B_0\Lambda > a)$$

The most probable wavelength Λ is as follows:

$$\Lambda = a9.02\frac{\left(1+0.45Z^{0.5}\right)\left(1+0.4T^{0.7}\right)}{\left(1+0.87We_2^{1.67}\right)^{0.6}}, \qquad \text{Eq.101}$$

The maximum grow rate *Ω* is given by:

$$\Omega = \left(\frac{\rho_1 a^3}{\sigma}\right)^{-0.5}\frac{0.34+0.38We_2^{1.5}}{\left(1+Z\right)\left(1+1.4T^{0.6}\right)}, \qquad \text{Eq.102}$$

where *We* is Weber number, *Re* is Reynolds number a *Z* is Ohnesorge number and *T* is Taylor number:

$$Z = \frac{We_1^{0.5}}{\mathrm{Re}_1}, \quad T = ZWe_2^{0.5}, \quad We_1 = \frac{\rho_1 U^2 a}{\sigma}, \quad We_2 = \frac{\rho_2 U^2 a}{\sigma}, \quad \mathrm{Re}_1 = \frac{Ua}{\nu_1} \qquad \text{Eq.103}$$

These models can be commonly found in discrete phase model and herein, it was decided to apply the blob jet model to the Euler-Euler model.
The diameter evolution was defined using a scalar equation written as follows.

$$\frac{\partial a}{\partial t} + \frac{\partial}{\partial x_i} a c_i = S_a \,. \qquad \text{Eq.104}$$

The equation was solved without any diffusion term and the source term accounts for the change of diameter due to breakup. The source term was defined via UDF with using aforementioned formulas (APPENDIX IX). It was tuned on the simple solid jet within quite coarse 2d mesh (see Fig. 37). The dead zone is a consequence of the fact that UDS was solved only where water was presented. The Fig. 37 is the final field of diameter used in momentum equations. It must be noted the model is however only half-baked, since it does not account for new formed droplets. Nevertheless, it might be possible to obtain, say, averaged diameters in different cross sections.
Since the droplet diameter is used only for calculation of drag force acting on the droplet surface and there was not found any effect of droplet diameter on the spray jet pattern in previous calculations, this breakup study was not used in the next cases.

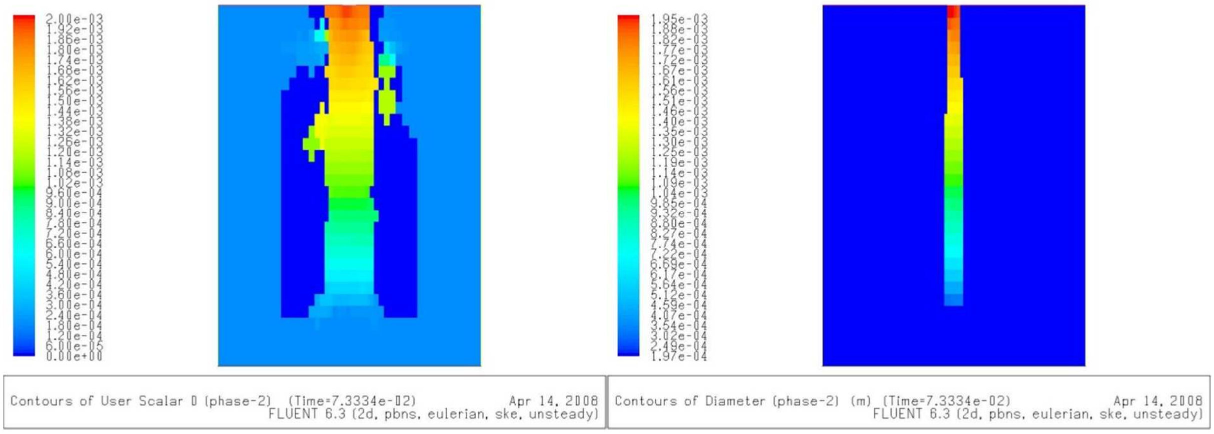


Fig. 37 *UDS field of diameter and final field of diameter, respectively*

4.2.2 Flat jet in continuous casting using Euler-Euler model

This section is again focused on already mentioned flat jet nozzle in chapter 4.1.2, which is utilized within the secondary cooling section above the foot roll in a real continuous slab caster. In APPENDIX VII, the drawing of the slab caster detail is shown i.e. the area right below the mold bottom and the slab caster description is given in brief. On a company's request the slab caster dimensions could not be unfortunately disclosed. Firstly, the water jet was simulated in a wall-bounded rectangular domain to avoid the extremely narrow geometry and the angular gap between the foot roll and the slab, where the water might be collected.

In this case, the water was spraying on the vertical moving zone with velocity corresponding to the slab motion i.e. downwards speed of 5.0 m/min. The top and the bottom walls were imposed with no slip condition, whereas planes of symmetry were used as BC on side walls. The velocity profile with corresponding turbulence and multiphase properties from the previous calculation (see Fig. 34, Fig. 35) was assigned to the square velocity inlet. Using such a complex velocity profile could not be straightforward, since there can be found local face fluxes outgoing from the domain. In other words, it meant that imposed turbulent properties were ignored. Further, especially in this case, there were very high values of turbulent quantities on water-air interface and regarding the solution of field of turbulent quantities, this could make difficulties as well. The pressure outlet of 0 Pa was imposed in the same plane as the velocity inlet.

Two different meshes were used in simulations. The first one was the hybrid mesh containing hexagonal elements in the region of high speed water jet. In other words, the hexagonal mesh was built inside the region where the water jet was expected. The rest of domain was filled up with a tetragonal mesh. The second one was consisting solely of tetragonal elements. The tetragonal mesh quality was naturally much worse than the hexagonal one, especially in the vicinity of nozzle exit.

As definition of the velocity inlet implies, Euler-Euler model was used in calculation along with the k-ε per phase turbulence model. Both mass and momentum equations were solved on the per phase basis. Interaction between the primary and the secondary phase was described only using the drag force. The computing of the drag force was based on symmetric model, which supposes the same diameter for both, the primary and the secondary phase. The symmetric model is useful in cases when the secondary phase can become primary phase in some region of a computational domain. Moreover, unlike other models for drag force, the definition of the drag coefficient is evidently simpler that provides the presumption of a more stable calculation.

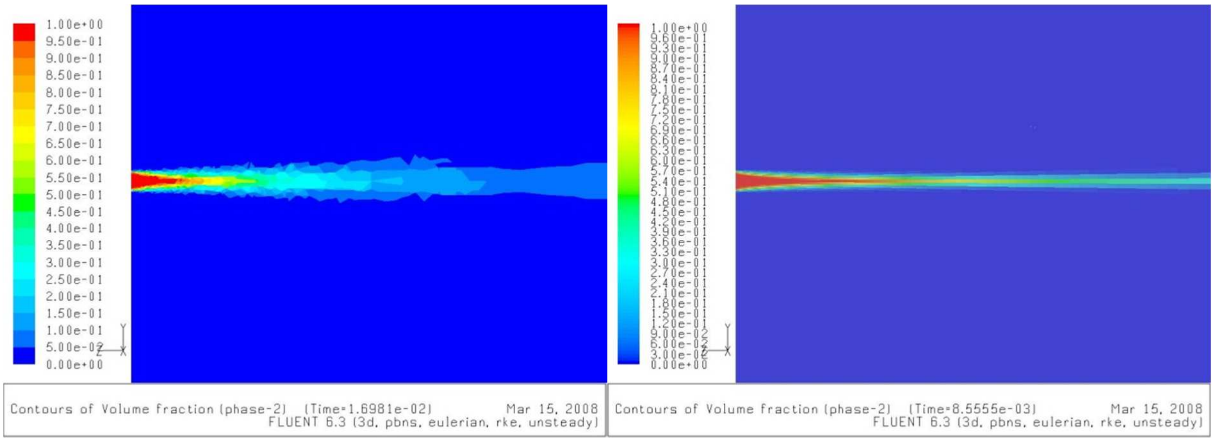


Fig. 38 *Volume fraction of water for tet mesh and hybrid mesh, respectively*

To precisely describe the inlet profile the cell size of 0.1mm was used. Expecting velocities of about 40m/s required time step less than t = 5e-06 s to keep the Courant number less than 2.0. Such a low time step results in very long computational time needed for a slab to pass through the domain at least once. All these calculations were intentionally terminated when the flow time 0.015 s was exceeded.

The convergence behavior in the case of the tetragonal mesh was very poor. It was approximately one order worse than for the hexagonal mesh in terms of normalized residuals. Although the water jet pattern looked like more realistic than water jet obtained within the hexagonal hybrid mesh, results within the hexagonal mesh were considered to be more accurate. Contours of velocity magnitude are shown in a longitudinal section along the jet axis for both, the tetragonal and the hybrid mesh, in Fig. 38. It is hard to judge which calculation is more accurate. Generally, the hexagonal mesh provides a better stability during a calculation and thus, should ensure more precise results. However, the water seemed to artificially align to the hexagonal cells oriented in the flow direction. Therefore, the breakup of the water jet could be unnaturally suppressed.

In Fig. 39, volume fraction of the water with water velocity vectors, iso-values of a water volume fraction with contours of a total pressure, contours of a turbulent kinetic energy of water in a top view, contours of a turbulent kinetic energy of the water in a side view, and iso-values of the water volume fraction colored by a velocity magnitude of water are depicted, respectively.

As described in Tab. 12, four different cases were carried out and quantities such as water velocity, air velocity, and volume fraction of water were compared in a line perpendicular to the flow direction and located in the center plane of flat jet. The line to nozzle distance was 5, 10, 20, 50, and 100mm, respectively.

The velocity plots are in good agreement within all cases. On the contrary, water volume fraction of water is very low for the tetragonal mesh cases compared to the hybrid mesh due to the spreading of the water in the perpendicular direction to the center plane of water jet symmetry. It should be noted that this spreading was probably due to not converged solution, thus, it has no physical meaning (Fig. 40).

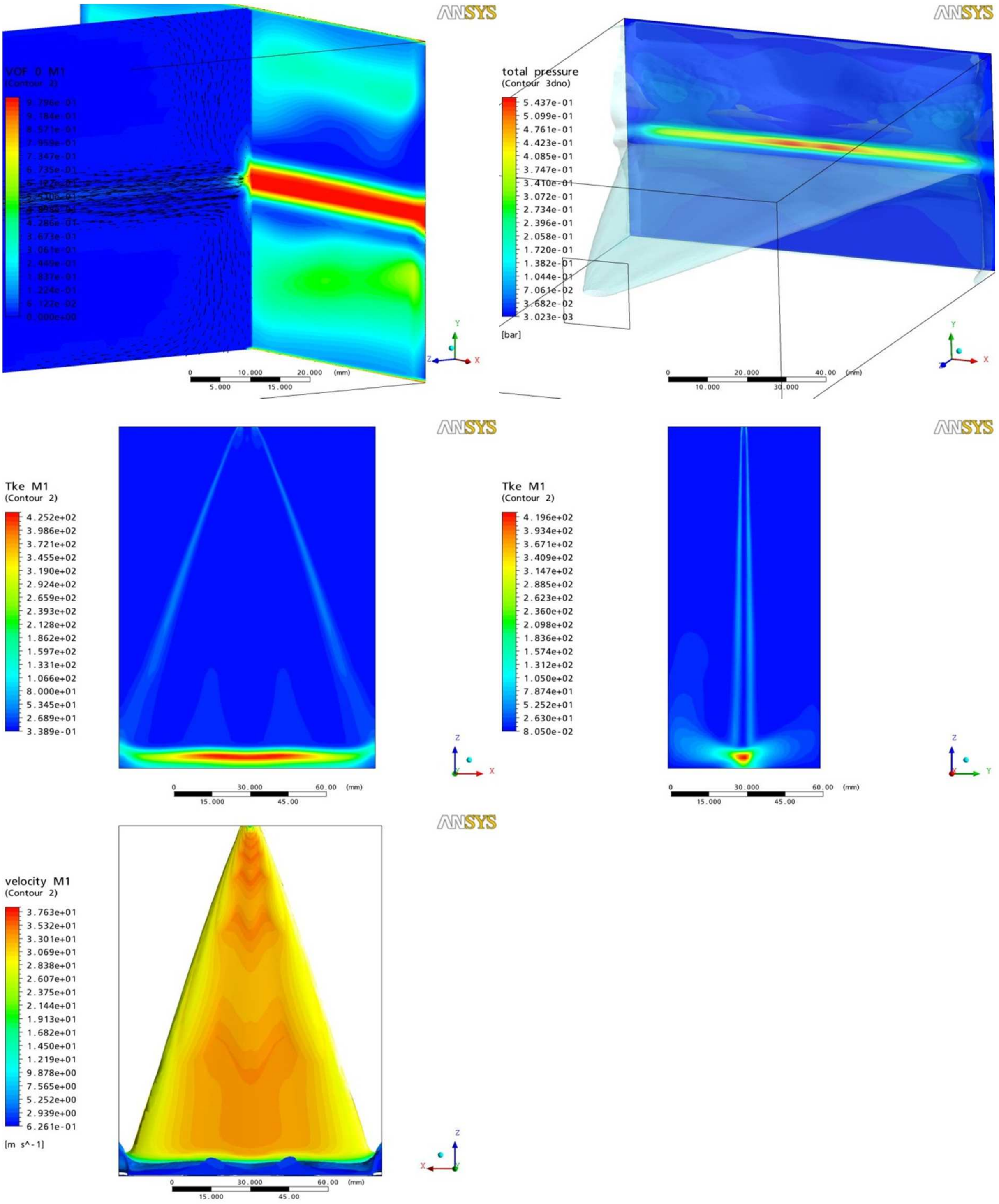


Fig. 39 *Volume fraction of water with water velocity vectors, iso-value of water volume fraction with contours of total pressure, contours of turbulent kinetic energy of water in top view, contours of turbulent kinetic energy of water in side view, and iso-value of water volume fraction colored by velocity magnitude of water, respectively*

Tab. 12 *Case notation of water jet modeling in wall-bounded region*

1	mixture k-ε profile, hybrid mesh, droplet diameter of 200μm
2	per phase k-ε profile , tetragonal mesh, droplet diameter of 200μm
3	per phase k-ε profile, tetragonal mesh, droplet diameter of 600μm
4	per phase k-ε profile, hybrid mesh, droplet diameter of 200μm

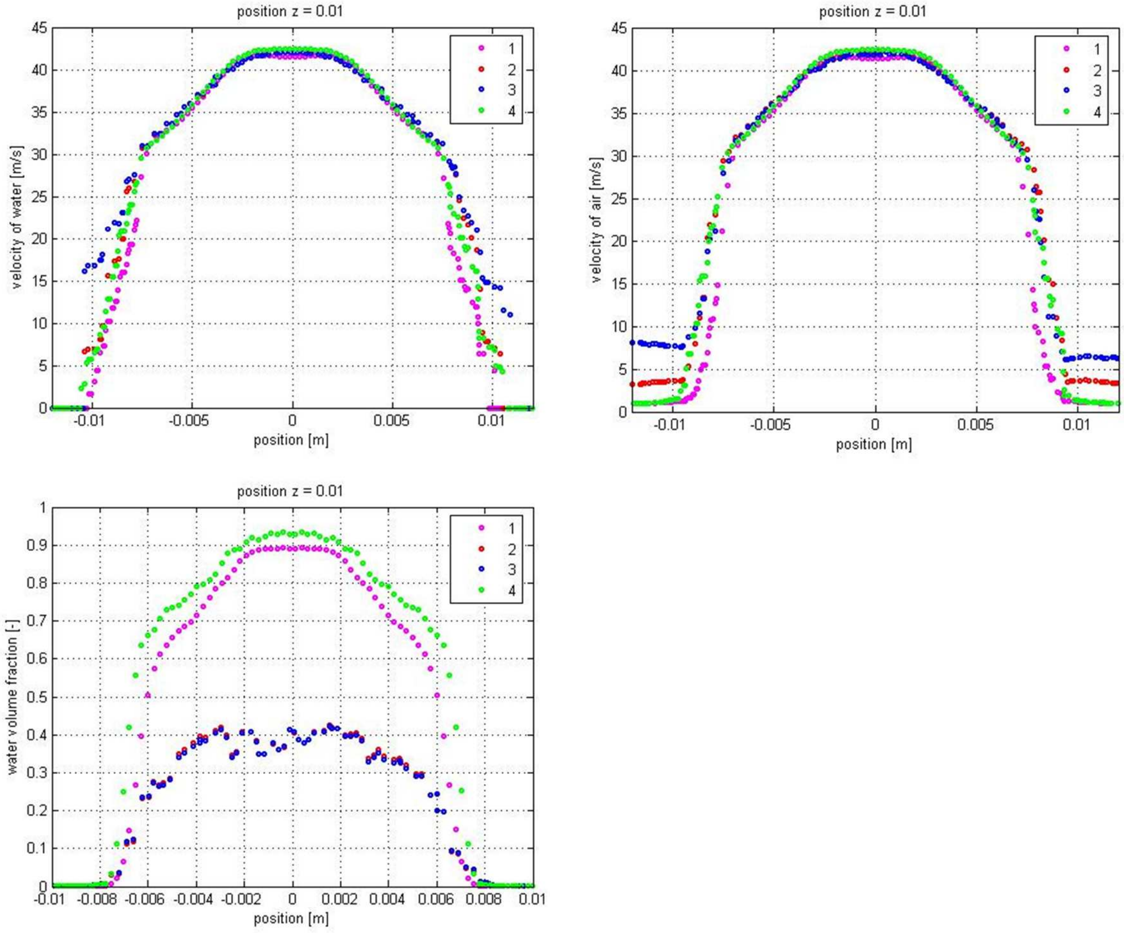


Fig. 40 *Dependence of droplet velocity, air velocity, and volume fraction of water on position, respectively, z = const = 10mm*

To sum up, water jet modeling in the simplified wall-bounded rectangular domain required very long computational times. In other words, the time step was very high compared to the total time. Further, results were different for both, the tetragonal mesh and the hybrid mesh.

In the next step, the simplified geometry was replaced by the full geometry i.e. the fluid region bounded by the mold bottom, the foot roll surface, the slab surface was taken into account.

The full geometry is shown in Fig. 41. Only the fluid zone was solved i.e. the solid zone (the foot roll, the slab) were not taken into account. Both, the wall impinging region and the gap between the foot roll and the slab, were expected to make difficulties in convergence behavior.

Two different meshes were tested. The first one was the hybrid mesh with tetragonal and hexagonal elements. The second one consisted purely of hexagonal elements; however, some hexagonal elements were swept from pave surface mesh. Furthermore, the second one was made of rather coarse cell elements and adapted in regions of large gradients. In the first case the convergence was very slow. To make matters worse, the solution was limited in terms of turbulent viscosity. The second case was without problem with limiting of turbulent viscosity; however, the convergence was still very slow. Further, the non-conformal interface between adapted cell elements caused unreal distribution of solved variables in some regions. This was caused because of still too coarse mesh even in the region of adapted cell elements.

Velocity inlet profile, which was imposed as velocity inlet boundary condition, had different values than those imposed. Moreover, all the imposed values seemed to be changing during calculations. This trouble could be as a result of a non-uniform data and wrong interpolation, consequently. The imposed velocity inlet profile should be fixed after several first iterations. It should be noted that if the flow is going out of the domain than all other imposed values on particular cell face are neglected by solver.

Very low time steps were used during calculations. Further, while the domain was filled with the water, convergence troubles appeared and it was not even possible to get to time when the mass in is equal to the mass out. The calculation had to be always aborted.

Unlike the water jet modeling in rectangular domain, the strange lack of water was found in the center line of the jet. This is shown in Fig. 41. Similar behavior can be also found in paper of S.E.Gant [80] (see Fig. 42). He has modeled the flow pattern of full cone nozzle. Although he imposed a smooth parabolic inlet profile with a constant radial component of velocity, he has obtained a hollow cone jet instead of the full cone jet. To say that this is wrong and unrealistic could be doubtful if compared with experimental results of St-Georges & Buchlin [81]. The velocity profile develops into the profile with two peaks outside the center line as shown in Fig. 43.

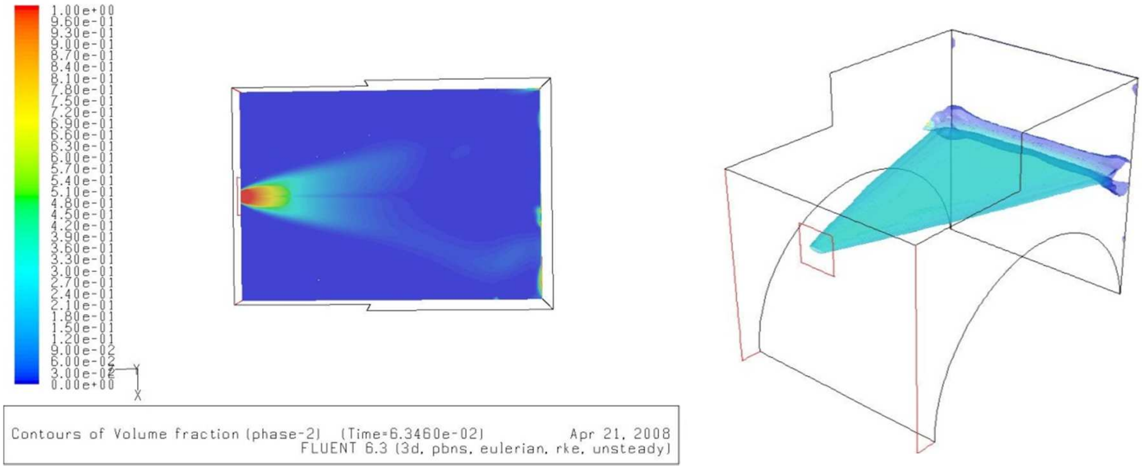


Fig. 41 *Volume fraction of water with strange dip in the centre (on the left), full geometry with the flat jet silhouette (on the right)*

Due to closed geometry and the fact that the water is cumulated in the gap between the foot roll and the slab, convergence was made more difficult. Moreover, it was not even possible to reach the time, when the mass in is equal to mass out.

Euler-Euler multiphase model was found to be very computationally expensive, even with convergence troubles; thus, it was decided to use Euler-Lagrange approach instead.

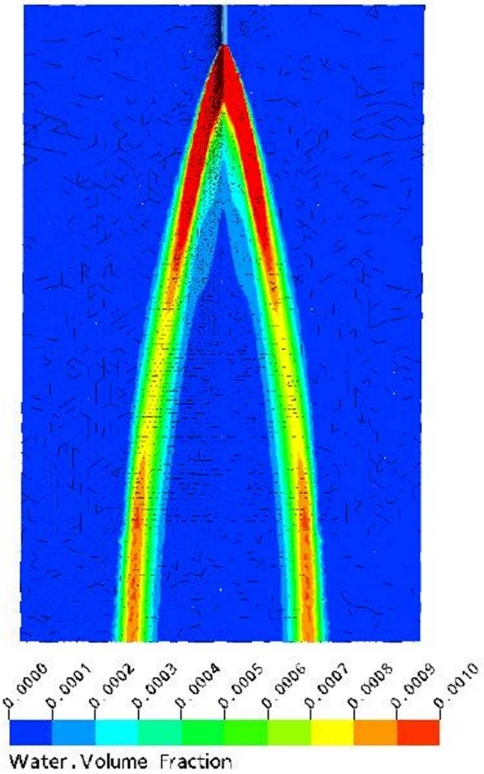


Fig. 42 *Contours of volume fraction for full cone nozzle by S.E.Gant*

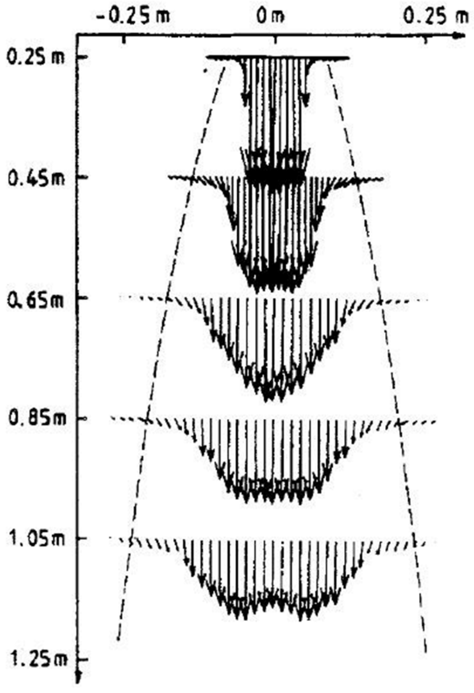


Fig. 43 *Experimental results of St-Georges & Buchlin, full cone nozzle*

4.2.3 Flat jet in continuous casting, Euler-Lagrange model

The theory of Euler-Lagrange model was discussed in section 2.3. Also in here, the water jet was firstly modeled in the rectangular wall-bounded domain for the reason of model tuning. Afterwards, the full geometry was taken into account.

As regards the first case, the flat jet was perpendicularly passing through the plane and droplet data was captured, simultaneously. The nozzle-wall distance was of 140 mm. The computational domain seeded with particles is shown in Fig. 44. Evaluation of reliability of computations was done using comparison between data from simulation and experimental data. It must be emphasized that experimental data were obtained for different flat jet nozzle; however, both nozzles have almost the same parameters. Both, distribution of droplet diameter and velocity distribution, were used for the comparison. Settings for the flat fan atomization model are shown in Tab. 13. Some important settings are shown in Tab. 14.

Tab. 13 *Flat fan atomization model setup*

x-position of center origin [m]	0
y-position of center origin	0
z-position of center origin	0.0063024
x-position of virtual origin	0
y-position of virtual origin	0
z-position of virtual origin	0.0015
x-component of normal vector	0
y-component of normal vector	1 (horizontally spraying flat jet)
z-component of normal vector	0
flow rate [kg/s]	0.279
half of spray angle [°]	27.5
flat fan width [m]	0.0015
Flat fan sheet constant [-]	12
Dispersion angle [°]	6

Both, the air flow and the flow of droplets, were coupled in the following manner. The drag force acts on droplet and droplet motion acts on the air flow. The effect of air turbulence on droplet motion was not taken into account.

As regards collisions of droplets, collisions were not taken into account since the solution could be then very mesh sensitive.

Tab. 14 *Solution setup*

Number of continuous phase iterations per DPM iteration	5
Number of continuous iteration per flow time step	10
Flow time step	0.0001 s
Number of particle streams	10

Primary breakup was predicted using flat fan jet atomization model itself and the secondary breakup was calculated using TAB (Taylor analogy breakup) model, since it was recommended in literature. The other possible breakup model (WAVE model) was also tested. However, TAB model has given better results and thus, was used in next calculations.

In Fig. 45, there is shown velocity distribution for both, experiment and simulation. Different settings of breakup parameters did not notably affect the velocity distribution.

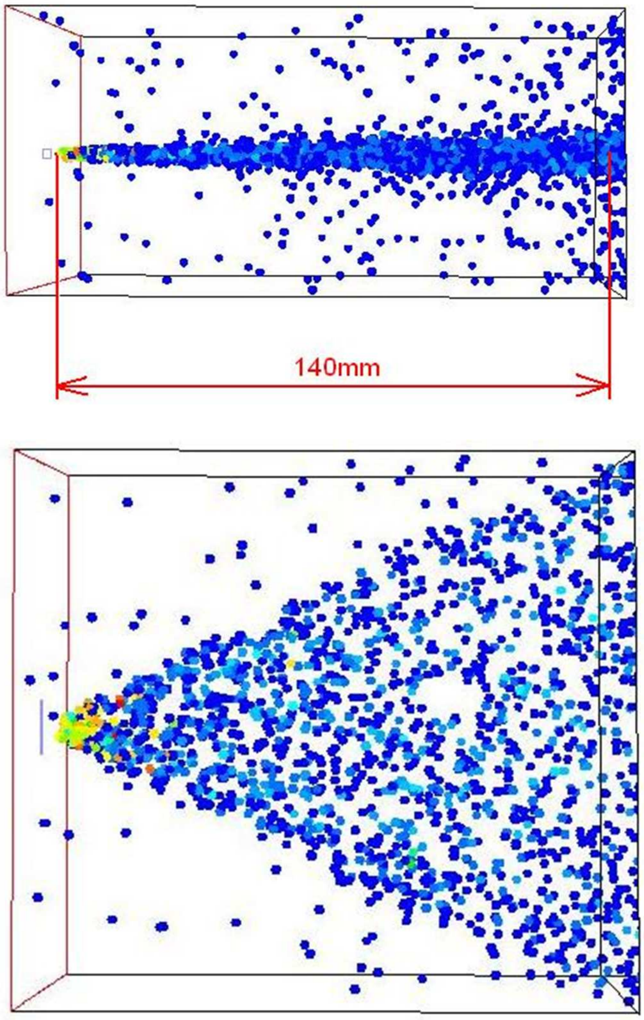


Fig. 44 *Geometry configuration*

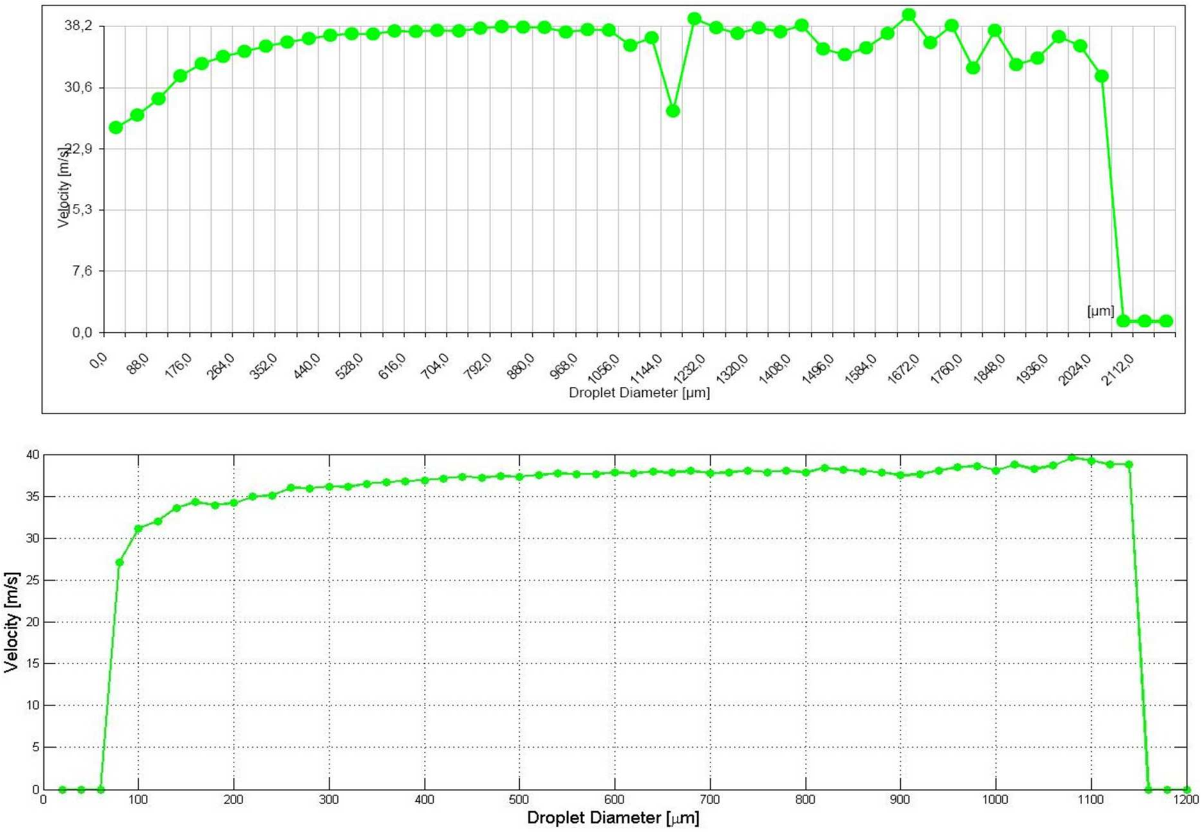


Fig. 45 *Velocity distribution from experiment (top), velocity distribution from simulations (bottom)*

The distribution of velocity was found to agree well with experimental data almost for all variations in breakup models constants.

The WAVE model (Blob jet model) was tested with different settings of both constants, B0 and B1. The constant B0 accounts for the radius of new formed droplet and the constant B1 is used in breakup time definition. Default values are as follows: B0=0.61 and B1=30; however, it can be found B1 of 1.73 in literature. The fundamentals of WAVE model is described in previous chapter. Different combination of both, B0 and B1, were tested as shown in Tab. 15. Corresponding plots are accordingly shown in Fig. 46.

Tab. 15 *Constant setup*

B0	**B1**
0.1	1
0.3	1
0.3	30
0.5	30
0.61	1
0.61	20

None of tested combinations of both, B0 and B1, fit well the experimental data (see Fig. 46). However, it might be possible to get better fit with different values of B0 and B1. The best fit with experimental data was obtained with TAB model and number of parcels of 2000 as shown in Fig. 47. In the same figure, dotted lines represent results from cases with different settings of droplet parcels and droplet streams. The number of parcels was changed in order to fit the experimental data. From the figure it is obvious that the increasing of the number of parcels leads to more realistic results. The number of parcels defines the number of different droplet diameters which can be occupied after breakup of parent droplet. More parcels signify smoother droplet distribution, however, it is more computationally demanding.

It must be noted, that number of particle streams was constant in all cases (Tab. 14). A higher number of particle streams would cause smoother droplet distribution after primary breakup and a smoother droplet distribution further downstream, consequently.

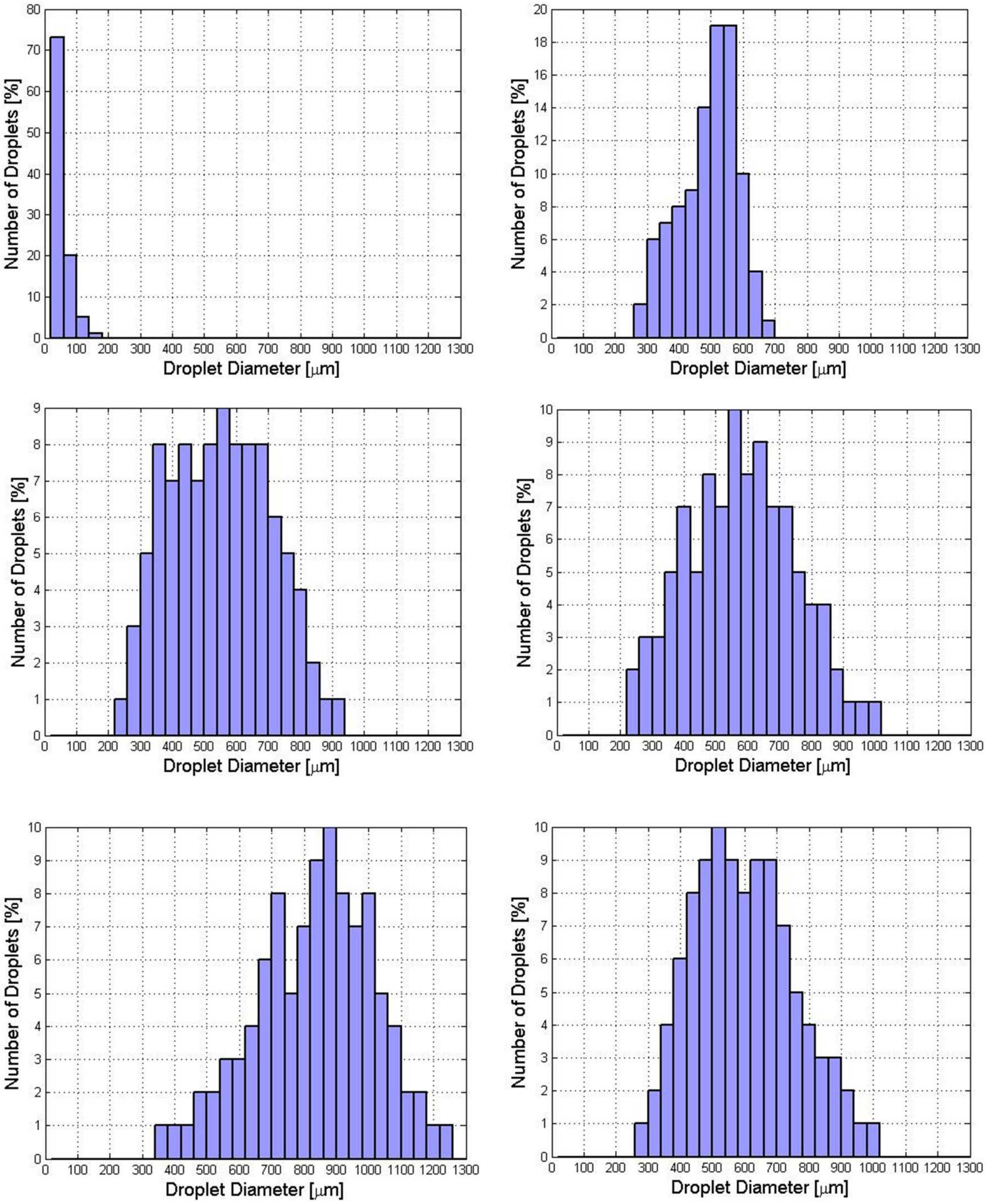


Fig. 46 *Droplet number distribution for different setting of both, B0 and B1*

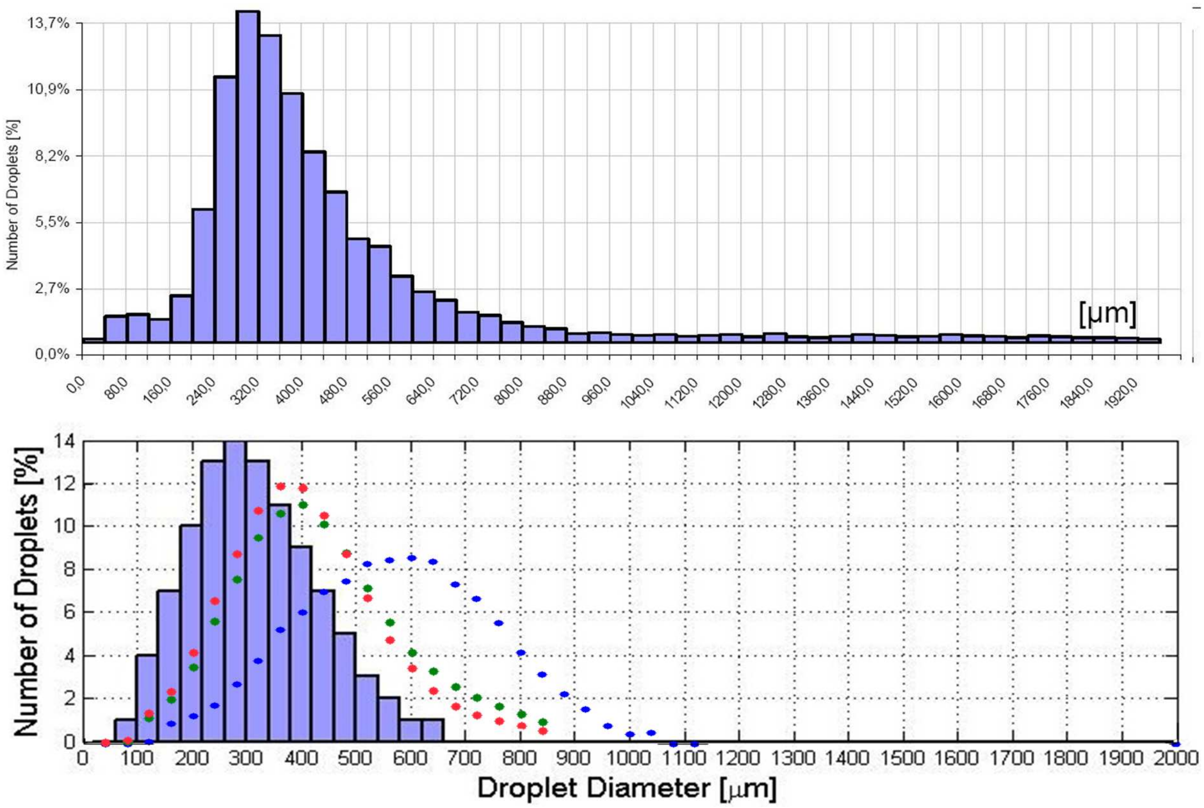


Fig. 47 *Droplet distribution from experiment (top), velocity distribution from simulations (bottom), red dotted line denotes the case with 10 streams and 1500 parcels, green dotted line denotes the case with 10 streams and 1000 parcels and blue dotted lined denotes the case with 10 streams and 50 parcels*

After the model tuning, the water jet was simulated within the full geometry. The same geometry was used as in the case of Euler-Euler modeling; however, the sharp edge in vicinity of the foot roll and slab contact was cut off to improve mesh quality. Thereof, new surface with small area, which is not present in real, was created.

Tab. 16 *Three different simulations were performed*

1) symmetry boundary condition was used
2) periodic boundary condition was used instead of symmetry BC
3) symmetry BC, coupling between DPM and VOF

Regarding the first case, the nozzle angle offset was set to zero. In other words, the water jet was spraying horizontally and thus, there was no overlapping, water jets were colliding. Boundary condition for discrete phase model on symmetry was set to reflection with constant coefficient of restitution of 1. Boundary condition on slab surface (the wall of direct impingement) was set to wall jet BC i.e. each particle was considered as a solid jet impinging onto the rigid surface within certain angle (see chapter 2.3). This boundary condition is generally recommended for heat transfer calculations with high temperatures.

The size of elements was 2 mm. Convergence behavior during these calculations was very good as in previous calculations within rectangular domain. In Fig. 48, contours of air velocity magnitude are shown. A characteristic deflection of the airflow is noticeable due to the pressure difference.

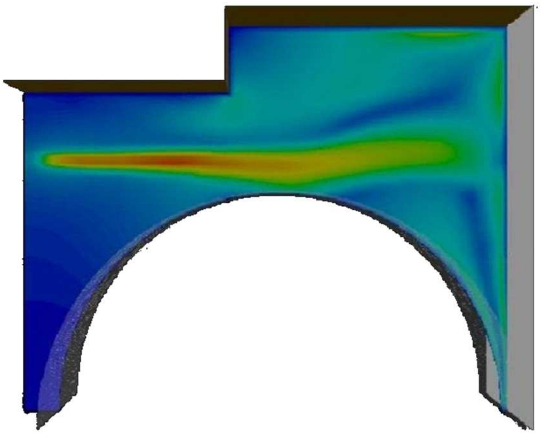


Fig. 48 *Side view on contours of air velocity magnitude in center plane*

As regards the second case, the effect of the angle offset on the flow character was observed. It was set to 5°. A water jet collision with the foot roll was expected; however, it did not happen. Symmetry boundary condition was substituted with periodic boundary condition to simulate overlapping of water jets.

In both cases, the first one and the second one, droplets were allowed to escape from domain as soon as they hit either the bottom of mold or the small artificial surface at the bottom. This procedure ensured that droplets were not cumulated in the domain and computational costs were not increased; however, the flow description of flow was completely wrong in the gap between the foot roll and the slab.

As regards the third case, both, the proper flow description in the foot roll-slab gap and a mass conservation, were desired. It was supposed that droplet energy is wasted after impact onto the mold bottom, symmetry planes and also in the gap between the foot roll and the slab. Hence, droplets rather form into a continuous region of fluid (water) after collision with either one of boundaries or certain portion of water. Therefore, the coupling between discrete phase model and volume of fluid method was considered as the best way. This coupling was done via UDF (APPENDIX X). Initially, there was a certain amount of water patched in the gap to improve convergence and shorten the computational time.

The computational procedure was as follows. Firstly at the beginning of time step, the continuous phase represented by mixture of water and air was solved using VOF model. Then, positions and velocities of droplets represented by Lagrange particles were updated considering coupling between the continuous and the discrete phase. In other words, each droplet loses a part of its energy due to the drag force and consequently, this portion of energy is imposed as a momentum source in transport equations. Further, UDF for discrete phase sources was used to detect both, droplets which hit the bottom of mold, the planes of symmetry, the foot roll and droplets which enter the computational cell with volume fraction of water $F >= 0.5$. The mass and the momentum of each droplet were assigned to mass source and momentum sources, respectively and the droplet as a Lagrange particle was removed from the calculation. After that, the whole process came to the next time step and computational procedures repeated again.

It should be noted that none of droplet mass was imposed in source term during just one single time step i.e. it was rather spread within a longer time period to provide smooth source distribution and avoid overshooting values. Such a mass source term was given by the following formula:

$$S = \frac{Q}{s_p \cdot V_{cell} \cdot t} , \qquad \text{Eq.105}$$

where Q is the flow rate [kgs^{-1}], s_p is the stream strength i.e. number of droplets in stream [$droplets.s^{-1}$], V_{cell} is the volume of a cell and t is the suitable time period. As mentioned above, the basic element size was 2 mm. Knowing the flat jet velocity of around 40 ms^{-1}, it is easy to determine the suitable time step (approximately 1e-05 s) based on Courant number. The time period t for smoothing of sources was determined intuitively and tested numerically. Three different values of time period t were considered (5e-05s, 1e-04 and 2e-04 s) and results were compared in terms of water mass generated by each case (see Fig. 49 and description below).

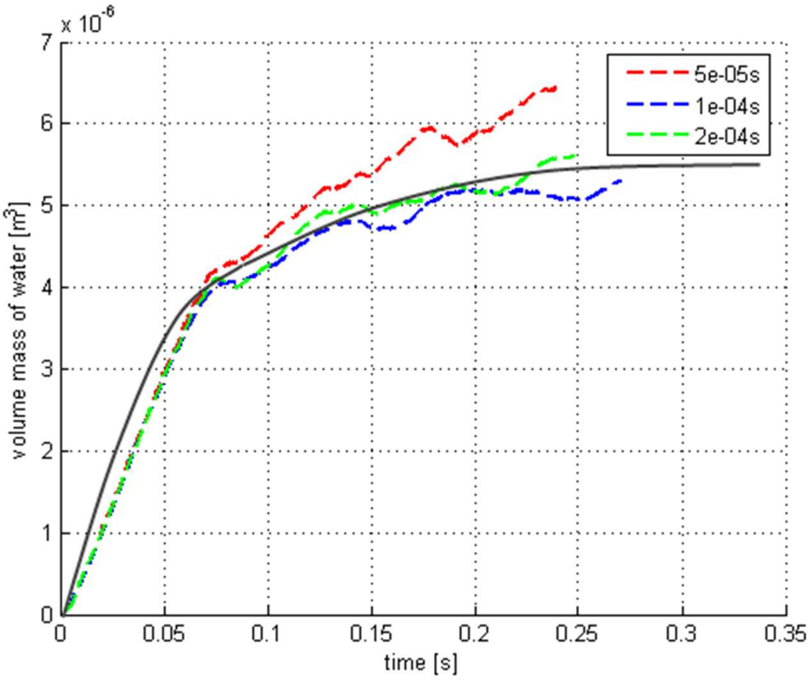


Fig. 49 *Total volume fraction of water within whole computational domain dependent on time (dotted lines represents simulations for different time periods t, whereas grey line represents expected behavior)*

In Fig. 49, the volume mass of water dependent on time stands for the actual difference between total volume fraction of water and the initial volume fraction of water within entire domain. It is clear, that the difference is firstly quickly increasing, but as soon water starts to escape from domain through the opening, the difference decreases until it remains constant i.e. inflow is the same as the outflow. All of tested time periods gave satisfactory results and thus the time period t was said to be a parameter that does not have an effect on mass generated by the source term (Eq.105).

It should be noted that this comparison does not say anything about the mass conservation between Lagrange and VOF model i.e. whether the water inflow is same as the water outflow. For this purpose, an additional UDF was designed in order to calculate both, cumulative mass delivered by Lagrange particles lowered for water outflow, and actual volume mass of water currently present in domain lowered for the initial volume fraction of water. Regarding the first variable, it should show linearly growing trend until water starts to flow out of the domain. Then, it shows rather exponential trend and finally it should ideally stay constant. Regarding the second variable, it is the same variable as shown in Fig. 49. A relation exists between these two variables. They are not coincident. The second one starts to grow a little later, but then it should collinearly follow the first variable until water starts to flow out of the domain. Knowing this condition of collinearity, the source term given by Eq.105 can be multiplied by a constant *C* and the model for the coupling can be tuned to fulfill mass conservation consequently.

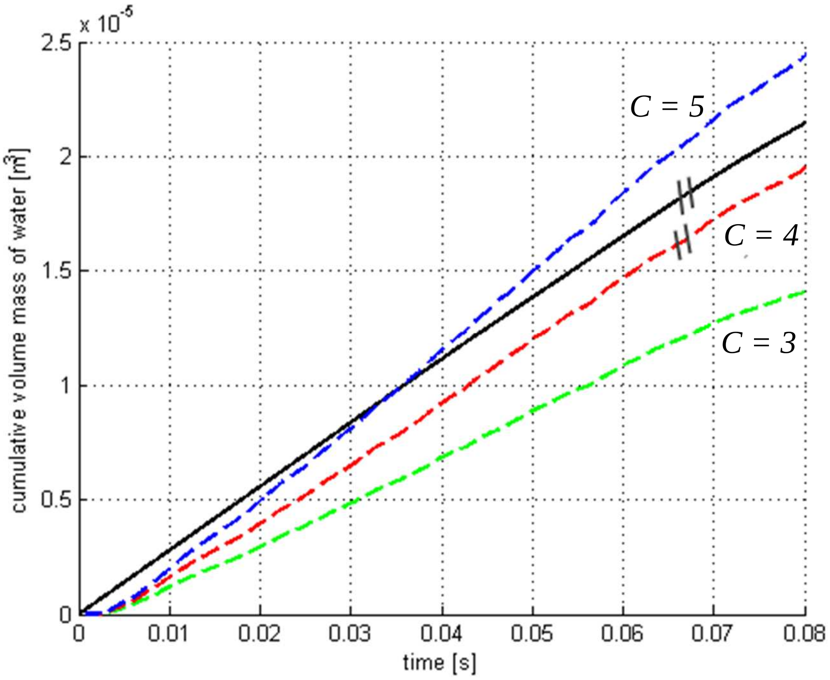


Fig. 50 *Volume mass of water dependent on time (black line represents cumulative volume mass caused by Lagrange particles being aborted, dotted lines represents total volume fraction of water lowered for initial water content for different constants C)*

It is evident that the constant C = 4 shows the best mass conservation (see the red dotted line).

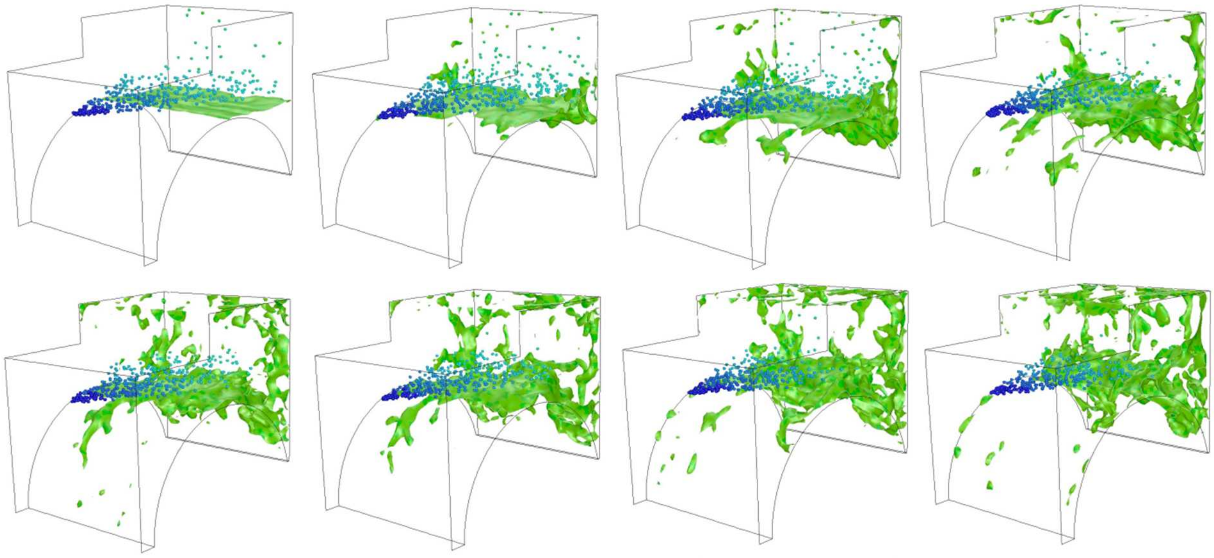


Fig. 51 *Isometric view of the whole computational domain with flat jet represented by water droplets and continuous water in gap between the slab and the foot roll (droplets are represented in blue, water volume fraction of 0.5 is in green)*

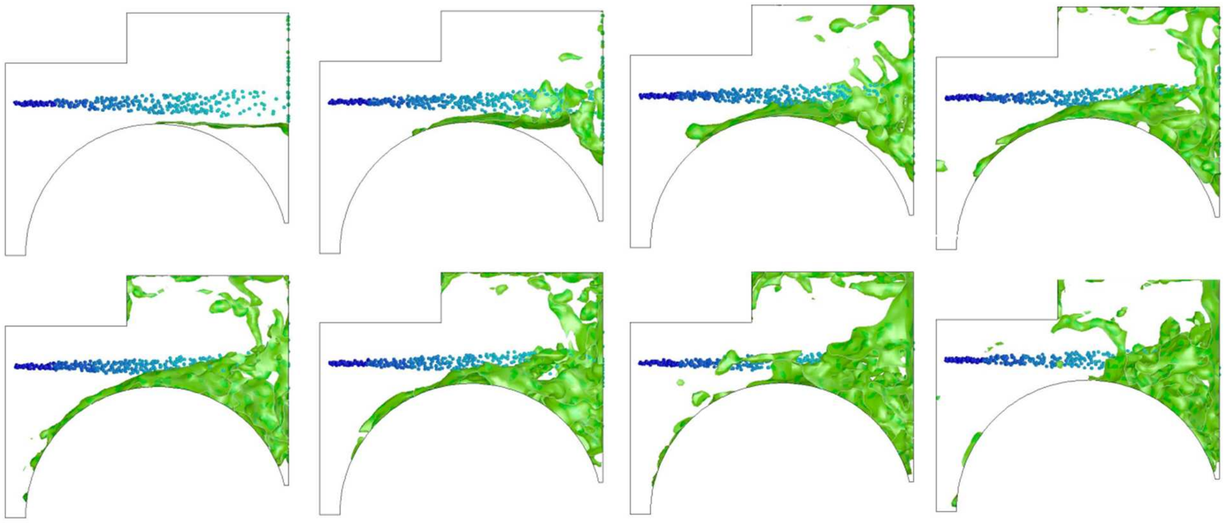


Fig. 52 *Side view*

In Fig. 50 and Fig. 51, the flat jet is horizontally spraying onto the slab. It is shown in blue colormap. Further, in regions where water forms into continuous phase, it is represented by water volume fraction of 0.5 (green). Both figures stand for a time marching from 0 s to 0.3 s. Several issues had to be sorted out before running a successful simulation, which should be discussed here. The first one is related to the application of drag force, which can be described by the following equation

$$F_D = \frac{\mu}{\rho_p d_p^2} \frac{18 \cdot C_D \,\mathrm{Re}}{24}, \qquad \text{Eq.106}$$

where μ denotes dynamic viscosity of ambient and index points at droplet properties. Drag coefficient C_D was formerly defined in chapter 2.2. The issue lie in fact that discretization scheme for volume fraction of water is not able to track absolutely sharp interface i.e. in some regions it is rather smeared within air phase that consequently causes unrealistically high drag forces. An additional UDF was designed in order to treat ambient with water volume fraction < 0.5 as it was pure air. This UDF for drag force is described in APPENDIX X. Another issued can be found when considering a break up model. In aforementioned simulations, The Taylor Analogy Breakup model was considered. However, in here the breakup model was not taken into account since the breakup is mainly based on Weber number, which would be artificially altered in cells containing water volume fractions >0 and it would consequently lead into inadequately fast droplet breakup. Unfortunately, it is not possible to change any breakup model parameters via UDFs, and thus the breakup model was switched off during these simulations. To sum up the model for coupling between the DPM and the VOF, it was found to be numerically stable especially due to smoothed source terms. It showed quite fair results which were not however validated by experiment. The time step was one order lower (1e-05 s) than the time step in the case of just the DPM. UDFs used for the coupling are available in APPENDIX X.

In heat transfer calculations the coupling was however not included because it would increase complexity especially model for vapor layer would have to play together with the coupling model. Thermal radiation was also not considered due to its complexity and high computational demands. The full geometry including the solid zone (the slab) was employed.

These calculations might theoretically give good results in the nozzle footprint and its close surroundings; however, as shown later there were not in agreement with the reality at all.

Droplet evaporation and droplet boiling were considered as the most dominant heat transfer mechanisms. The transport of non-reacting species was employed to describe the formation of the vapor and vapor mixing with ambient air. Material properties are listed in APPENDIX XI. Some properties were found to be highly temperature dependent, especially saturation vapor pressure. Therefore, piecewise-linear interpolation was used describe these properties. Saturated vapor pressure is obviously the most temperature dependent quantity as shown in APPENDIX XI.

Heat transfer calculations were tested within the following settings. The temperature of the slab was lowered to 200°C in order to avoid high sources in transport equations in ‘kick-off’ simulations. The operating fluid temperature was 50°C. As regards settings of DPM that were fixed in all simulations, the number of continuum phase iterations per DPM iteration was 5. Next, the under-relaxation factor for DPM was 0.2. The secondary breakup model was not considered, since it led to droplet temperatures below 0°C when using TAB model. Author wanted to note that the increasing diffusion coefficient should naturally enhance evaporation process; however, this was not observed in simulations. In Tab. 17, there are results from 9 simulations. Results are compared in terms of average heat flux from slab surface. Variable parameters are always shown for each simulation. Basically, the calculated heat fluxes are give or take 10 times smaller than those being realistic. Anyway, one can study parameters that have effect on the calculated heat flux. It is obviously size of cells adjacent to the slab surface that has the most significant effect. It is caused by the sensitive difference between the slab temperature and the temperature in the cell centroid. Unfortunately, the mesh refinement near the slab is restricted by the DPM limitation on volume fraction of dispersed phase that should be lower than 10%. Turning on turbulent model did not show any noticeable difference in results compared with laminar simulations. The lower the number of particle streams was, the higher the heat fluxes were calculated. The higher number of streams also led to convergence problems resulting from volume fraction limitation for DPM. As regards the effect of saturated vapor pressure on the heat flux, its jump slightly increases the heat flux. In Fig. 53, the horizontally spraying flat jet is shown along with contours of heat flux on the slab surface. The cooling intensity is evidently higher in nozzle footprint than in surroundings.

The vast discrepancy between calculated heat fluxes and realistic values can be explained using the paper by Bhattacharya [34]. He claimed that the DPM was developed for simulation of heat transfer from hot gaseous phase to liquid droplet phase and thus, not for simulation of evaporative heat transfer from heated solid surface to liquid phase. He employed a simplified modification of DPM in order to simulate evaporation of a single droplet. The temperature in cells adjacent to the slab was held constant at slab temperature throughout the whole evaporation process. Further, the mass fraction of vapor was set to unity in this cell layer adjacent to the slab.

Tab. 17 *Model settings with calculated average heat flux from slab surface*

wall BC	**flow**	**number of streams**	**height of first cell layer**	**saturated vapor pressure**	**heat flux**
		[-]	[mm]	[Pa]	[W/m2]
wall film	laminar	90	2	piecewise linear	8000
		30			10000
wall jet					7000
			0.2	2658	62000
				26580	68000
				265800	74000
			0.1	2658	100000
				265800	115000
	turbulent		0.2		100000

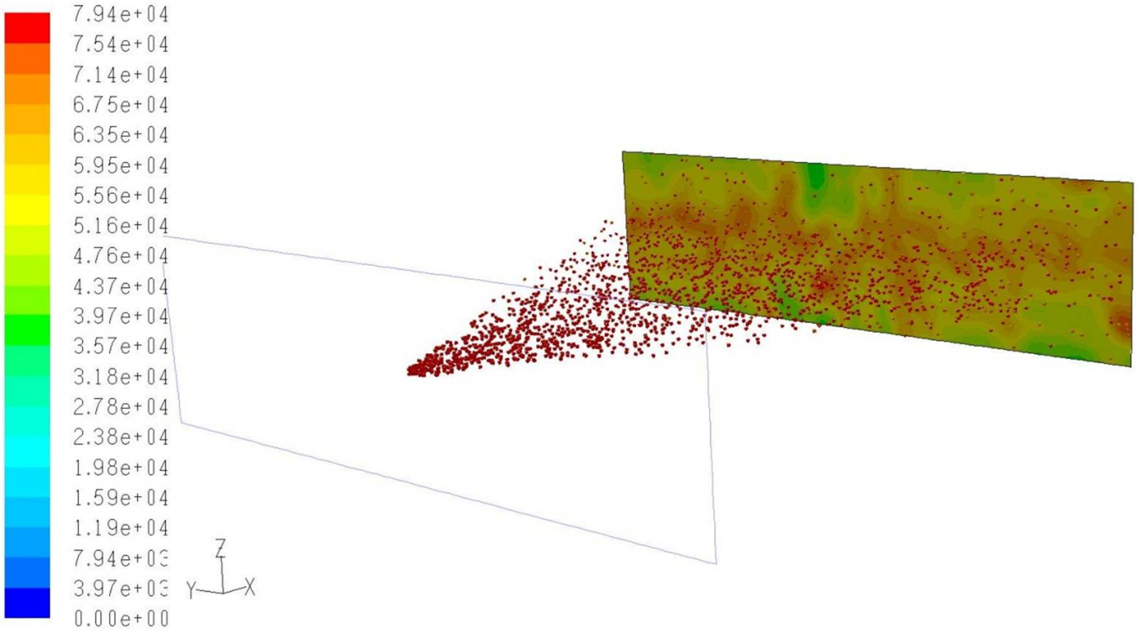


Fig. 53 *Droplets of flat jet and contours of heat flux [W/m^2]*

4.3 Flow outside cooling nozzles, solid jet impinging onto hot plate

It is well-known that solid water jet generally offers great cooling capabilities within very small impacted area compared with other nozzle types such as a full-cone nozzle, a hollow-cone nozzle etc. Conversely, if the area to be cooled down is much larger than the solid jet footprint, it is convenient to use full-cone nozzle instead.

Here, in this chapter, only one particular nozzle was chosen to be modeled. It was the solid jet nozzle with the exit diameter of 2 mm (see Fig. 54). The flow rate of 1.2 l/min was simulated. The solid jet was perpendicularly impinging onto hot circular plate of diameter of 60 mm. The nozzle-to-wall distance was 10.6 mm. Either a constant temperature or a constant heat flux was applied as a thermal boundary condition on the impact wall. Firstly, the water temperature was below the boiling point thus the vapor layer was not formed. The solid jet spreading, the transition from laminar to turbulent flow and heat transfer coefficients were studied. Several models for predicting turbulence were tested (k-ε, k-ω, RSM, LES); however, only k- ε model gave fair results with acceptable computational costs.

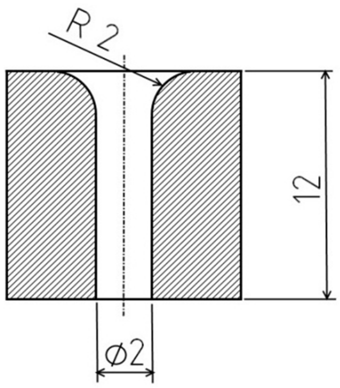


Fig. 54 *Solid jet*

4.3.1 Tuning of constants in k-ε model

Forced convection around the flat plate was chosen in order to tune up constants in k-ε model, since it was studied by many researches and reliable correlations are available for wide range of Reynolds and Prandtl numbers either for constant heat flux or constant temperature. In this case, correlations for constant plate temperature were employed [82].

Numerical model was similar to experimental set-up described in [82]. Model was 2D axisymmetric and water inlet was far enough from the entering edge of horizontal plate to ensure correct boundary layer development. At the entering edge, the laminar boundary layer starts to develop. For critical Reynolds number Re_c = 1e+05, the transition from laminar to turbulent flow occurs and cooling intensity is enhanced. For laminar flow, the correlation for local Nusselt number has the following form:

$$Nu_x = \frac{h_x x}{k} = 0.332\mathrm{Re}_x^{0.5}\,\mathrm{Pr}^{0.33} \quad 0.6 \le \mathrm{Pr} \le 50 \qquad \text{Eq.107}$$

For turbulent flow, the local Nusselt number is:

$$Nu_x = \frac{h_x x}{k} = 0.0296\mathrm{Re}_x^{4/5}\,\mathrm{Pr}^{1/3} \quad 0.6 \le \mathrm{Pr} \le 50 \qquad \text{Eq.108}$$

Physical properties of the flowing medium correspond to so-called film temperature, which is defined as:

$$T_f = \frac{T_s + T_\infty}{2} \qquad \text{Eq.109}$$

As regards the numerical model, several water velocities were considered (6, 12, 24 m/s). The transition from laminar to turbulent regime was derived from the critical Reynolds number and the laminar zone was imposed accordingly.

Enhanced wall treatment with realizable k-ε model was used for turbulence description [83]. It is a two-layer model combined with so-called enhanced wall functions. In general, turbulence enhances heat transfer. In numerical modeling, this enhancement is represented by turbulent conductivity k_t, which is defined as:

$$k_t = \frac{c_p \mu_t}{\mathrm{Pr}_t}, \quad \text{Eq.110}$$

where Pr_t is turbulent Prandtl number and it must be found by experiment. Here, this constant was found to be 1.3. In Fig. 55, there is shown both, correlation results for turbulent region and computational results for two turbulent Prandtl numbers. Correlation fails to predict transition HTC; however, simulation is obviously capable of that. The black curve ($\mathrm{Pr}_t = 1.3$) follows very well the correlation. This is for the water velocity of 6 m/s. Further, other velocities (12, 24 m/s) were taken into account to check the versatility of turbulent Prandtl numbers. In Fig. 56, each curve stands for the relative error of HTC between computational and correlation results. It is evident that a higher velocity implies a higher error. Despite the 6% error in case of 24 m/s, it is still worthful result within the bounds of CFD capabilities.

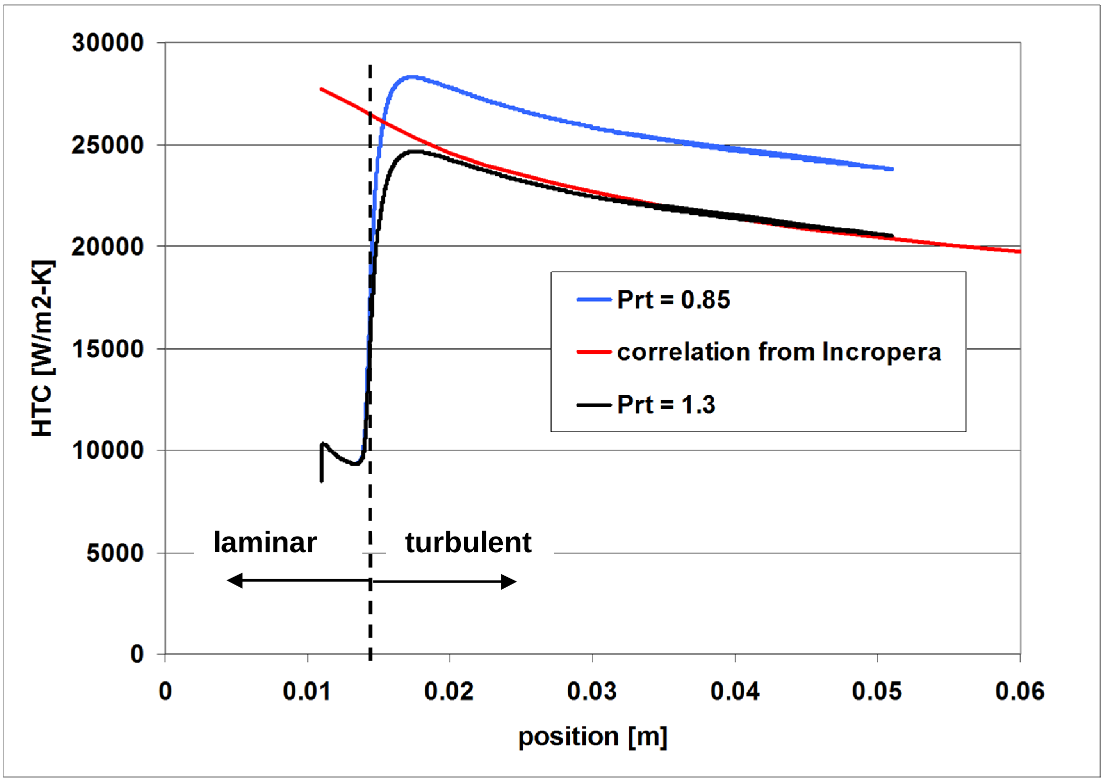


Fig. 55 *Heat transfer coefficient in turbulent region, T_s = 60°C, T_∞ = 40°C, velocity 6 m/s*

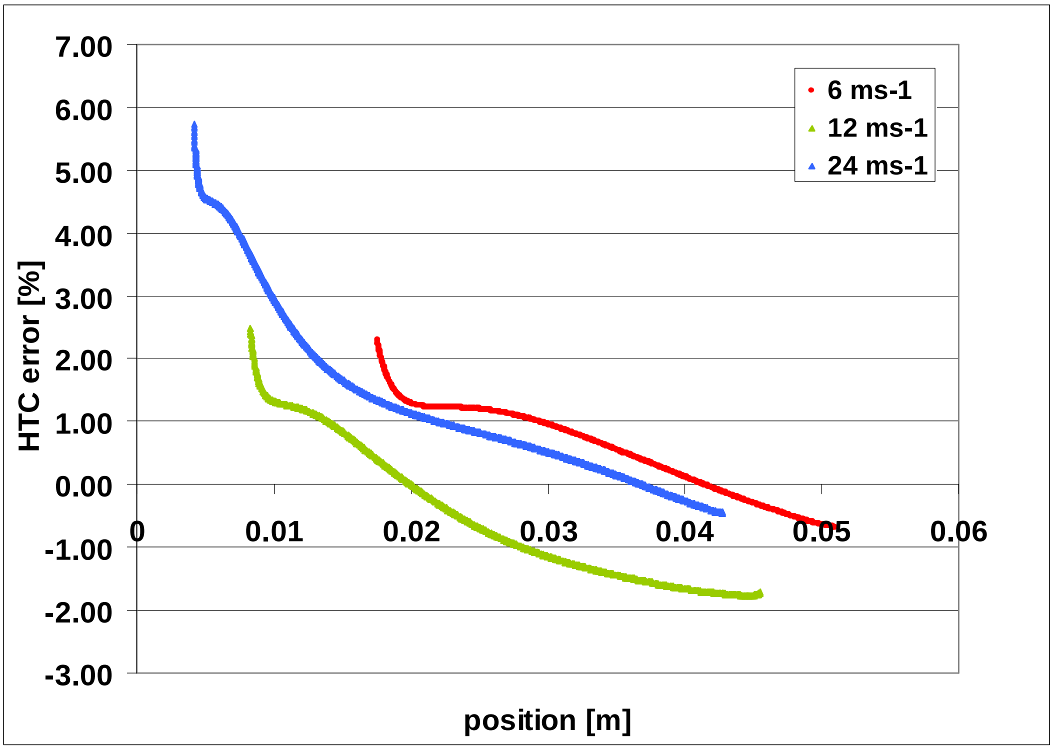


Fig. 56 *HTC error along the plate length for different water velocities*

To sum up, realizable k-ε model with enhanced wall treatment successfully simulated turbulent quantities along the hot horizontal plate. The value of y* was kept as close as possible to unity [83]. The default value of turbulent Prandtl number of 0.85 was found to overestimate HTCs in velocity range of (6 – 24 m/s). The turbulent Prandtl number of 1.3 was found to be the most accurate and was used in next calculations.

4.3.2 *Heat transfer without boiling*

Now, we have moved on to the numerical modeling of heat transfer between water solid jet and the horizontal hot plate. Since the nature of solid jet spreading and heat transfer is not known a priori and one must look on CFD results very critically, it was gone through literature to look up suitable experimental data. In [84], authors carried out several heat transfer experiments with confined, unconfined and submerged water jets within different thermal conditions on heated horizontal plate. They always used a constant heat flux instead of having constant surface temperature. One very important thing was observed in case of the single, circular, unconfined water jet. It was the transition from laminar flow to turbulence. The transition started around x/d=5.5 and end around x/d=9. It should be noted that transition location is not a constant value and it depends on jet velocity, nozzle-to-wall distance H, nozzle type and some other factors.

To make it worse, information about the transition location can be found only experimentally, since CFD does not allow simulating a transition from laminar to turbulent flow. In literature [86], several correlations were found usually describing an average Nusselt number Nu_{avg} vs. dimensionless position from jet axis x/D. However, none of them account for transition to turbulence. In several papers [84], [85], transition location was experimentally detected. In Fig. 57, several points are shown corresponding to particular jet velocity and dimensionless position where the transition to turbulent flow begins. Experimental data gathered by Wu, S.

(red points in Fig. 57) were ascertained as most reliable and were simply fitted with line defined as

$$x/d = -\frac{2}{3}u + 11.8\,, \qquad \text{Eq.111}$$

where u is the jet velocity. This equation was later used in calculations to specify laminar zone. Here, it is strongly recommended to use this equation only for undisturbed rather laminar jets and velocity ranges (4-13 m/s).

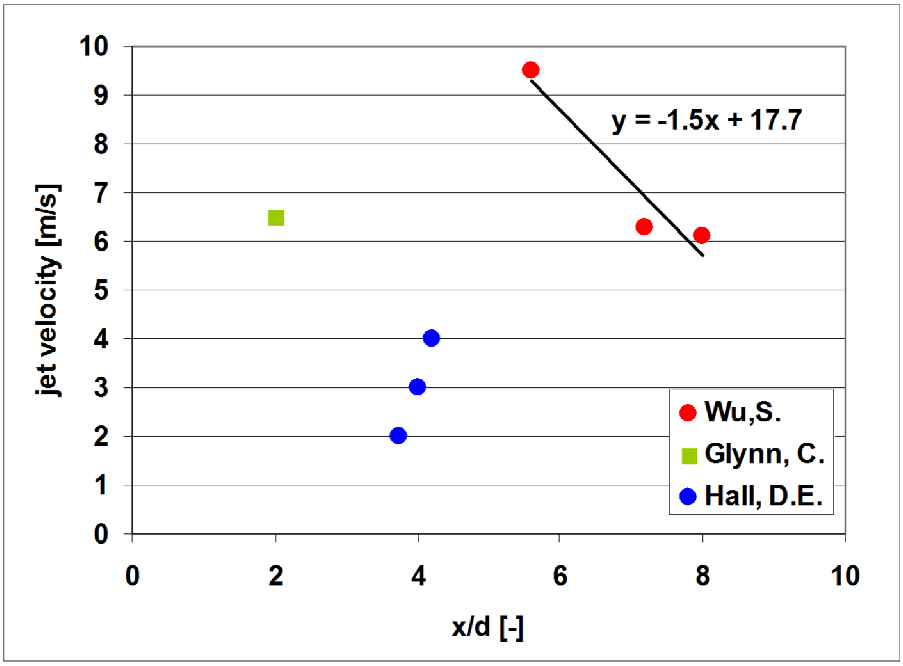


Fig. 57 *Each point stands for transition location from laminar to turbulent flow*

As mentioned several lines above, computational model was built according to the experimental setup described in [84]. The nozzle orifice diameter was 2.0 mm, the nozzle-to-wall distance was 10.6 mm. Inlet temperature of water was 50°C. Flow rate through nozzle was 1.2 l/min. The impacted area was of a round shape (diameter of 60 mm) with a constant heat flux 398 kW/m^2. Thereof, dimensionless position from the jet axis was in range of (0 – 15). Model setting carefully followed experimental set-up.

In Fig. 58, both, contours of velocity magnitude and phase interface, are shown. In the same figure, there are illustratively presented experimental (red points) and calculated HTCs (yellow and orange curve). The yellow one stands for the case simulating transition to turbulence and the red one for the case with turbulent flow in the entire computational domain.

In Fig. 59, the aforementioned behavior of HTC is depicted with appropriate scales. Relative errors for each computational model are also presented. The maximum error for turbulent model is about 40%, while the maximum error for the transient model is just 20%.

To summarize this chapter 4.3.2, it was found that water jet spreading undergoes two regimes. The first one is located near the impingement region, where the flow is relaminated. The second one appears in a certain distance from the jet axis. In this position of transition, the laminar flow switches to turbulent flow. This position was described by Eq.111 applicable just for prescribed conditions.

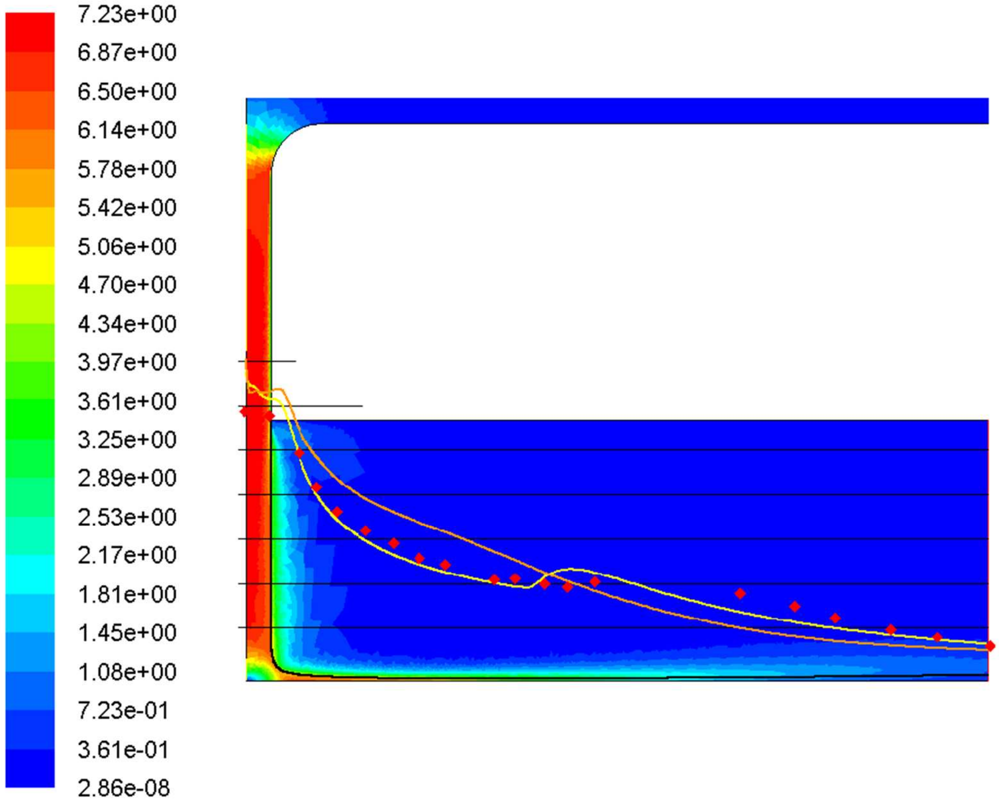


Fig. 58 *Spreading of unconfined water jet on hot wall with illustrative trend of HTC; contours of velocity magnitude [m/s]; red points (Wu's experiment), orange curve (calculation with k-epsilon model), yellow curve (accounts for transition to turbulence)*

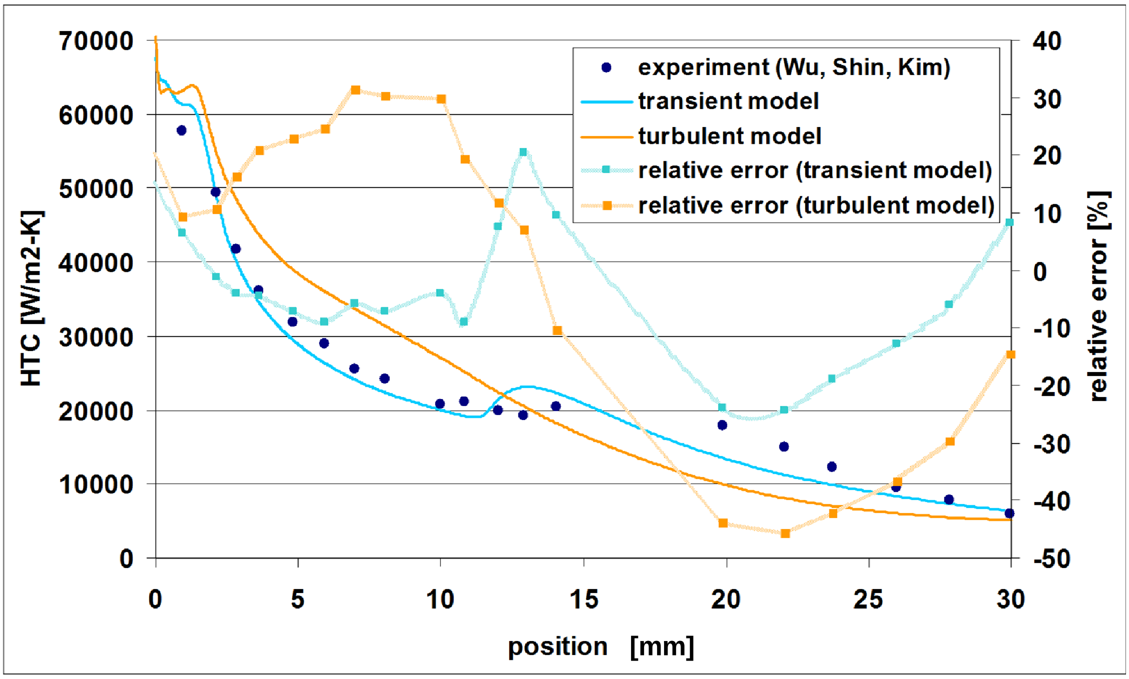


Fig. 59 *HTCs dependent on radial position; experimental data (blue mark), transition model (blue curve) and its relative error, turbulent model (orange curve) and its relative error*

4.3.3 Heat transfer with film boiling

Here, it was desired to shed a light on a modeling of film boiling. In chapter 4.3.3 heat transfer between unconfined solid jet and hot surface was discussed; however, boiling was not taken into account. Here, two different film boiling models are discussed. Regarding the first model, the geometry and model settings were adopted from the aforementioned model without boiling. Only the thermal boundary condition on the impact wall was altered. This thermal boundary condition of a constant heat flux was increased from 39.8 W/cm^2 to 141.4 W/cm^2 (this choice based on [84]) to provide film boiling existence.

As regards the second film boiling model, the different water jet [28] was considered instead. The unconfined water jet of diameter of 20 mm was placed into axisymmetric rectangular domain, which was 800 mm in width and 60 mm in height.

Vapor layer modeling via non-physical approach

The vapor layer was expected to be continuous with time independent thickness. In [84], it was clearly shown that film boiling occurs when the liquid reaches temperature of 150°C. Afterwards, the water layer thickness is expected to grow. In other words, it means that condensation does not take place and only vaporization can be thus considered.

Note that the aforementioned multiphase flow i.e. water plus air was solved with VOF model. Here, the third phase was added and it was the vapor.

The UDF was designed to calculate vapor mass and energy sources for continuity and energy equation. The algorithm was designed as follows. Firstly, the closest cell to the jet axis near impact wall with temperature of 150°C was detected. Since the vapor layer formation was expected to start inside this cell, the vapor mass source term was imposed and it was given by the following formula:

$$S_{vapour} = Q_l \frac{\rho_{vapour}}{\rho_l}, \qquad \text{Eq.112}$$

where vapor density to water density ratio provides that the vapor volume fraction will be equaled to unity. Corresponding to this vapor mass source term, source term for water continuity had to be imposed. It was given as follows:

$$S_l = -S_{vapour} \qquad \text{Eq.113}$$

Energy source term due to latent heat was constructed accordingly.

$$S_h = -S_{vapour} \cdot L \qquad \text{Eq.114}$$

Secondly, vapor flow rate through the outlet boundary condition was checked every each iteration. As soon as it was the same as the sum of vapor mass sources, new source terms were imposed in downstream cell and the algorithm looped until interface had the saturation temperature. The procedure is schematically shown in Fig. 60.

The main idea of this approach was to simply simulate a time-averaged vapor layer and solve it within steady mode. Unsteady solution would be greatly impractical because of small mesh size and consequent time step restrictions resulting from Courant number. It was desired to simulate the flow in whole domain as in 4.3.2.

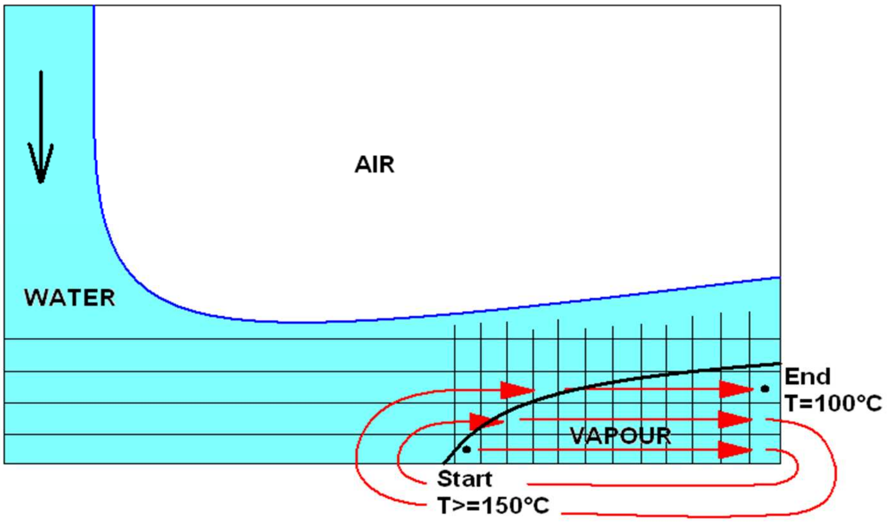


Fig. 60 *Illustrative outline of looping algorithm for deriving source terms*

In Fig. 61, contours of vapor volume fractions are shown in both, full and detailed, views. The blurred interface between water and air can be obviously observed. Moreover, one can see significant wrinkles that definitely imply an unsteady flow and a tearing of vapor layer. The blurry interface is a consequence of implicit formulation of volume fractions.

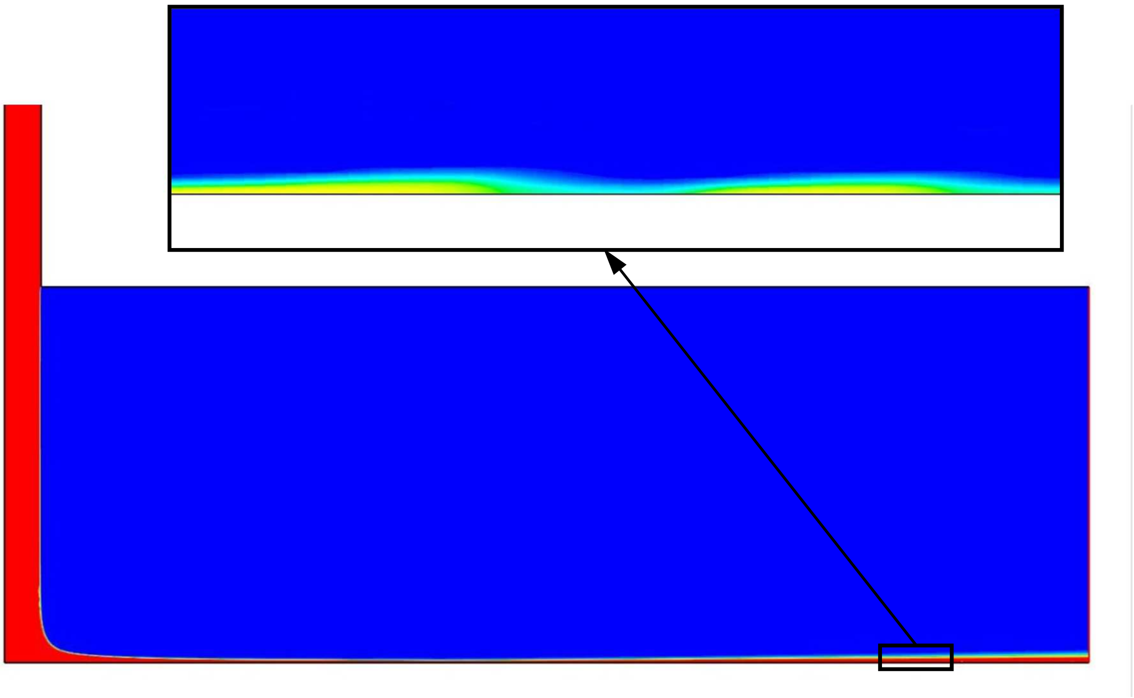


Fig. 61 *Detail of vapor layer on hot wall*

To sum up, steady calculation of vapor layer was very fast and might give approximate estimate of vapor layer thickness. Unfortunately, the primary assumption of steady film was disproved. Therefore this approach is not suitable in this case. In the next chapter, a different approach is presented.

Vapor layer model based on heat flux through phase interface

Unlike the approach mentioned in the previous chapter, here, the vapor layer can be continuous, but it can also oscillate and even break into small bubbles. The model geometry with appropriate boundary conditions was adopted from [28]. The confined water jet was placed in an axisymmetric rectangular domain, which was 800 mm in width and 60 mm in height. Source terms for vapor generation were constructed based on studies of Welch and Yuan [12], [14]. In each interface cell the heat flux was calculated (see Fig. 62) according to the following equation:

$$Q = k\frac{dT}{dn_i} = k\left(\frac{dT}{dx_i}\cdot n_i\right)/\left|n_i\right| \quad [W/m^2] \qquad \text{Eq.115}$$

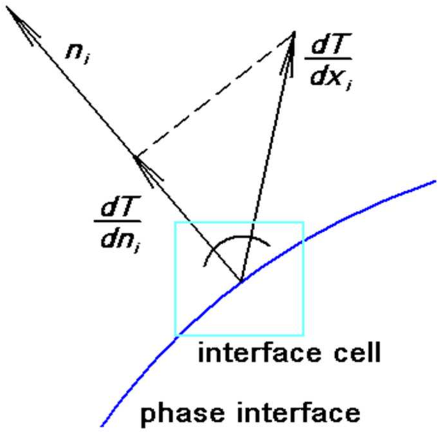


Fig. 62 *Scheme for idea about heat flux calculation*

The source term for vapor phase was consequently of the following form:

$$S_{vapor} = \frac{A \cdot Q}{L \cdot V} \qquad \text{Eq.116}$$

Corresponding to this vapor mass source term, source term for water continuity had to be imposed. It was given as follows:

$$S_l = -S_{vapour} \qquad \text{Eq.117}$$

Energy source term due to latent heat was constructed accordingly.

$$S_h = -S_{vapour} \cdot L \qquad \text{Eq.118}$$

These all source terms were explicitly applied via UDFs to transport equations and were solved iteratively. The UDFs are available in APPENDIX XII. Results for case with inlet velocity of 1 m/s, inlet temperature of 50°C and impact wall temperature of 130°C are shown in Fig. 63. In the first figure, contours of velocity together with the detail of vapor layer are presented. The vapor layer is discontinuous, changes rapidly, forms stripes that immediately condense and do not penetrate the bulk volume. In the second figure, heat transfer coefficient dependent on radial coordinate is depicted. Before transition to boiling regime the HTC is evidently smooth function. However, the boiling occurs at several diameters from the jet axis, the HTC values oscillate and are enhanced.

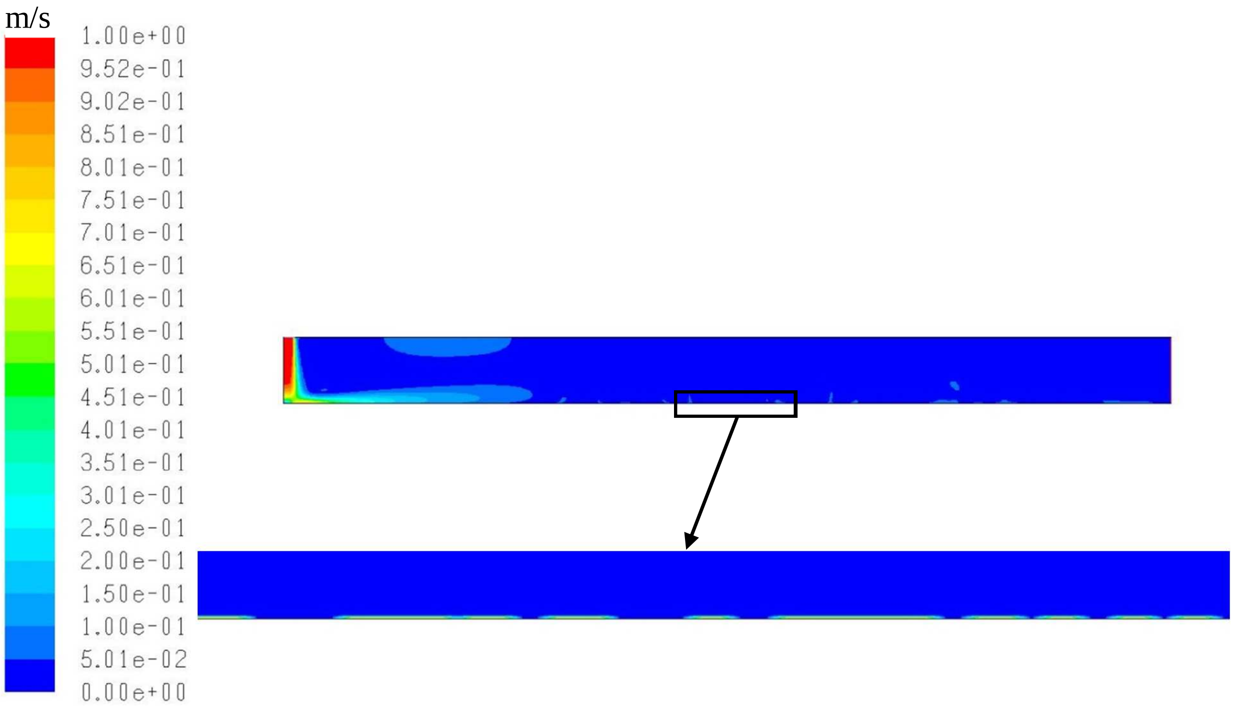


Fig. 63 *Velocity contours (top) and detail of vapor layer (bottom)*

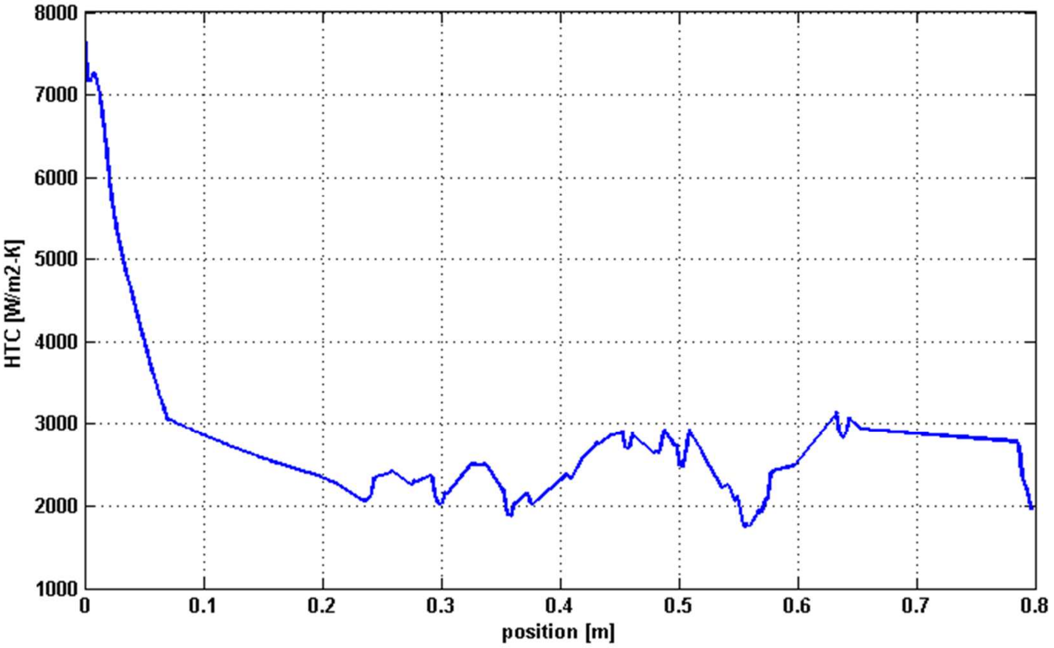


Fig. 64 *Dependency of Heat Transfer Coefficient on radial position*

4.3.4 Euler-Lagrange model used instead of VOF model

In chapter 2.3, a few words were remarked about the history of Lagrange model. Further, the model theory was discussed in brief especially wall boundary conditions.

In this chapter, the water jet was attempted to be modeled via Euler-Lagrange model, though the problem did not meet the limitations of the Lagrange model (see 2.3) especially the restriction on the volume fraction of the dispersed phase. The model geometry was the same as in the previous subheads. However, the mesh grid was much coarser in the case of Lagrange model. The size of the basic element was 0.2 mm and the finest element height near the impact wall was 0.05 mm, which was one order higher than in the case of VOF model (0.0036 mm).

In this model, the air was handled like the continuous phase, whereas the water jet was introduced as a dispersed phase i.e. the mixture of droplets of different diameters.

Concerning boundary conditions for the continuous phase (air), the pressure outlet BC was identical with that from the model in the previous chapter. However, the pressure inlet BC was removed and replaced by the wall BC. Other BCs such as the axis, the impact wall and the upper wall were retained.

It should be noted that the Euler and Lagrange models are entirely different compared with each other. Therefore, besides other things, it results in different boundary conditions. For Lagrange model, the inlet and the wall boundary condition are obviously the most important and the most crucial in most of applications.

Here, the inlet BC condition was represented by so-called the plain-orifice atomizer model, which based on Reynolds number, Weber number, internal nozzle geometry, saturation vapor pressure and liquid physical properties determines the working state of the nozzle and thus calculates the nozzle exit velocity, the droplet diameter distribution, the spray angle and the Sauter mean diameter. In other words, it defines the primary water jet breakup. The setup for this atomization model is given in Tab. 18.

Tab. 18 *Plain orifice atomizer model setup*

x-position [m]	0
y-position [m]	0
Temperature [°C]	50
Flow rate [kg/s]	0.02
Start time [s]	0
Stop time [s]	10
Vapor pressure [Pa]	12330
Injector inner diameter [m]	0.002
Orifice length [m]	0.012
Corner radius of curvature [m]	0.002
Constant A	4

The vapor pressure for the given temperature was calculated using the following expression

$$p = \exp\left(20.386 - \frac{5132}{T}\right), \tag{Eq.119}$$

where temperature is in Kelvin. In Tab. 18, the position of injection, the liquid temperature, flow rate is defined. Further, the start time and the stop time of injection have to be specified. The nozzle geometry is described using the nozzle diameter, orifice length and the corner radius inside the nozzle. Since the plain atomizer model was designed particularly for diesel jets i.e. different nozzle states can occur, the parameter A has to be specified. In fact, A is the spray angle and is said to range from 4 to 6.

The constant temperature of 100°C was imposed as the boundary condition on the impact wall. This BC itself does not say anything about the droplet treatment when they impinge onto the wall. For that reason an extra wall BC has to be selected to describe the droplet fade. Both, the wall-film and the wall-jet boundary conditions, were tested here in order to reveal the more appropriate one for other given conditions and settings.

The two-way coupling between the continuous phase (air) and the dispersed phase was considered. According to the results in chapter 4.3.2 , the flow regime is laminar near the jet axis, than undergoes a transition to the turbulent flow and from a certain distance is fully turbulent. Regardless this complex flow regime, the flow was simulated as a laminar. It caused a better stability during the solution.

Firstly, the cold flow is described in detail and results are compared with those mentioned in chapter 4.3.2. Flow variables such as velocity, static pressure are of interest. As regards the evaluation of water jet velocities, it is done by an UDF described in APPENDIX XIII. Since the jet is simulated using Lagrange model, the velocity within each cell is calculated as a mass-weighted velocity of particles located in a particular cell. In Fig. 65, axial velocities in three different positions are shown. The top graph compares axial velocity profiles of both, Euler-Euler model and Euler-Lagrange model, in the distance of 2 mm from the impact wall. The middle and the bottom graphs are for distance of 4 mm and 8 mm, respectively. It is evident that the Euler-Euler model gives a velocity depended on the radial position also inside the water jet. On the other hand, the Euler-Lagrange model gives an approximately constant value of the axial velocity (6.4 m/s). Furthermore, this value of 6.4 m/s does not change within the entire nozzle-to-wall distance. This is caused by relatively large Lagrange particles (~1.7mm) and high kinetic energy that is only slightly reduced for the drag force. However, the discrepancies as regards axial velocities inside the water jet are negligible (7%) compared to the differences inside the plain air flow (100%). In the case of the Euler-Lagrange model, the momentum transport in the radial direction is apparently significantly suppressed.

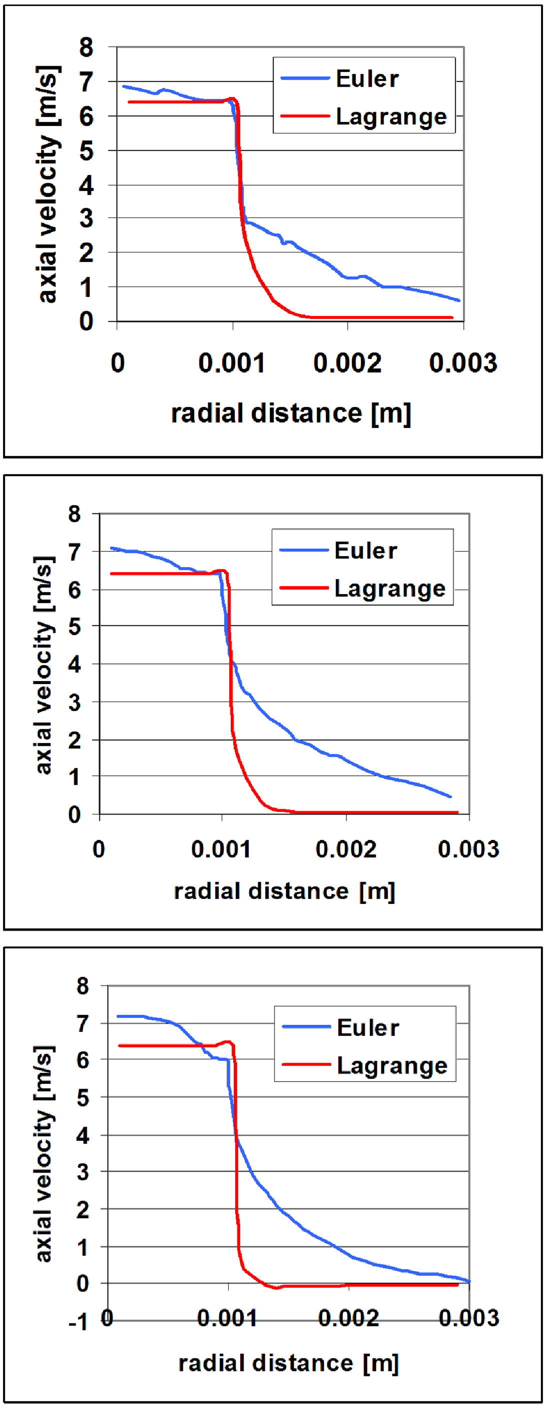


Fig. 65 *Profiles of axial velocities in 2, 4 and 8 mm from the impact wall*

In Fig. 66, radial velocity profiles of both, Euler and Lagrange model are shown for distance of 2, 6, and 18 from the jet axis. Only in the top graph, results are evidently at least similar and the trend of the Lagrange curve is the same as that of the Euler. On the contrary, Lagrange profiles in the middle and the bottom graphs differ significantly from Euler profiles. By and large, radial velocities calculated by Lagrange model are dropping faster than those calculated by Euler model. In summary, development of error in velocity field is obvious and probable causes of these errors can be stated. Considering the solid jet injection for the Lagrange particles as a beginning of the error growth, the first cause is immediately obvious.

It is the inlet BC for Lagrange particles. The aforementioned inlet BC for Lagrange particles called the plain-orifice atomizer model predicts the droplet diameter and its velocity regardless the position from the jet axis. In other words, a smoothly varying velocity profile exists in reality, while the Lagrange inlet velocity field does not account for the radial position that can consequently lead to errors. Secondly, other discrepancies in velocity field are found in the free stream water jet. In reality, the water jet is solid, with a free interface. The axial velocity profile is not constant. Moreover, it is developing with the axial coordinates. On the contrary, the axial velocity profile is constant in the case of the Lagrange model. The nature of the jet is incomparably different. It consists of droplet streams (10), where each stream is represented by a line of droplets, the drag force is the only force acting on the droplet and there is no momentum interaction between droplets. Since there is not a momentum exchange between droplets, the development of velocity in radial direction cannot occur. Further, in both cases, the same jump in axial velocities is observed on the interface; however, profiles of radial velocities in air bulk differ between each other (see Fig. 65). This might be caused by a rather coarse mesh within the air bulk. In the Lagrange model, air axial velocities are dropping much faster with radial coordinates. At last, the most noticeable errors are evidently caused after the jet impact onto the wall, where a wall BC for Lagrange particles is applied. As mentioned above, two wall BCs were tested (wall-jet and wall-film BC). The aforementioned results were acquired with the wall-jet BC. When have a look in Fig. 66, it is evident that the wall jet represented by the Lagrange streams of droplets loses its momentum very quickly compared to the results within the Euler model. Concerning the wall-film BC, it showed up that this BC is not suitable for modeling this type of the water jet, since droplets of higher impact energy were splashing on the impact wall, which was not realistic.

Apart from the comparison of velocity fields, impact pressure distribution within the water jet footprint is discussed here. The true value of a maximal impact pressure follows from the Bernoulli equation.

$$p_{MAX} = \rho \frac{v^2}{2} \qquad \text{Eq.120}$$

For given conditions i.e. the flow rate of 0.02 kg/s and the nozzle exit diameter of 2 mm, the max impact pressure can be 20.2 kPa. The blue line in Fig. 67 represents the impact pressure distribution for the Euler model in jet footprint within radial distance from 0 to 0.03 m. At first sight, it is evident that the max impact pressure exceeds p_{MAX} calculated using Eq.120. The overshoot is about 5 kPa that might be caused by a numerical compressibility. The impact pressure distribution has the peak in the jet center and then it drops with the increasing radial coordinate roughly following the Gaussian curve. Quite different results were obtained in the case of the Lagrange model. The red line in Fig. 67 stands for impact pressures from the Lagrange model simply calculated from impact energy of droplets. Because of more or less constant droplet impact velocities and no momentum exchange between them the impact pressure is constant in the jet footprint.

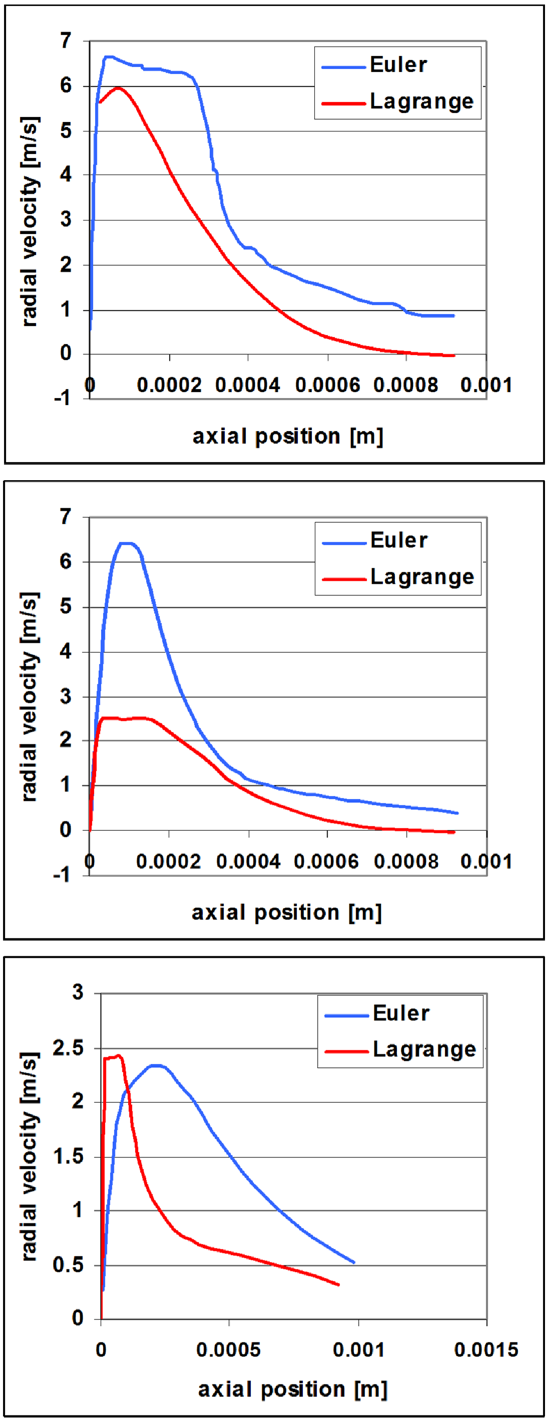


Fig. 66 *Profiles of radial velocities in 2, 6 and 18 mm from the jet axis*

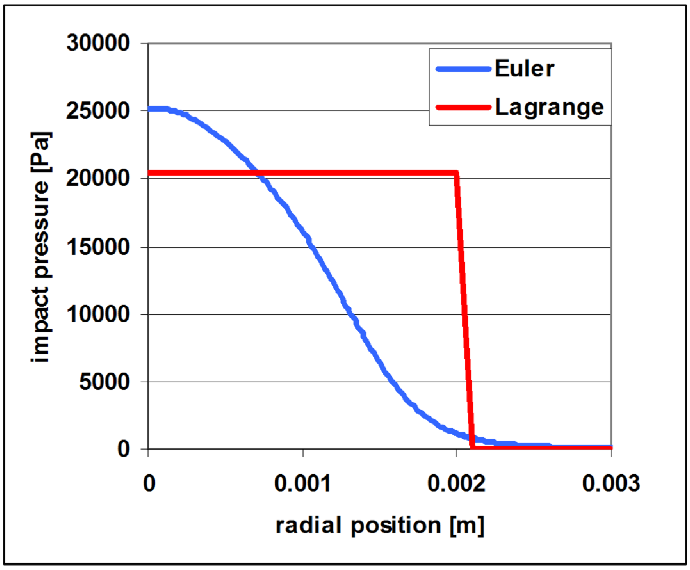


Fig. 67 *Impact pressure distribution in jet footprint for both, Euler and Lagrange models*

Despite rather different velocity and pressure fields between the Lagrange and the Euler model, heat transfer cases were tested with the thermal BC of a constant temperature of 100°C.

Both, at first, each droplet was handled as a solid particle and thus, only the inert heating of droplets was considered. For the modeling of the inert heating of droplets the inert heating law was used, which is a standard heat transfer law for DPM particles (described in 2.3). The calculated HTCs are represented by the red line in Fig. 68. The order of results is definitely eye-catching, because in the true order it is not 100 but 1000. The max HTC in the jet axis should be around 60000 W/m2-K. For the reason of totally different results than expected, it was assumed that if the HTC applied on the droplet surface was increased, it would lead to significantly enhanced cooling intensity. In APPENDIX XIV, there is described an UDF that customize the inert heating law such that it multiplies the HTC on the droplet surface by a suitable constant. More constants were tested and it was found that larger constants slow down convergence. Finally, the convergence was found for the constant equaled 5. The blue line in Fig. 68 stands for results with the user-defined inert law. Though HTCs further from the jet axis were more than two times increased, the HTC peak within the jet axis unexpectedly remained the same. The last attempt how to enhance the cooling intensity was done using turning on the species transport and different DPM laws namely the vaporization law and the boiling law. To make the picture complete, the continuous phase is represented by the mixture of particular species that are oxygen, nitrogen and the vapor. In this case of the three species, an additional transport equation is solved for each species except the most abundant one (nitrogen). When the droplet temperature gets to the vaporization temperature (10°C), the inert heating law is switched to the vaporization law previously described in chapter 2.3. Similarly, when the droplet temperature gets to the boiling temperature (100°C), the vaporization law is switched to the boiling law. The green line in Fig. 68 however depicts disappointing results, which tightly follow the results of the case with the inert heating law only applied.

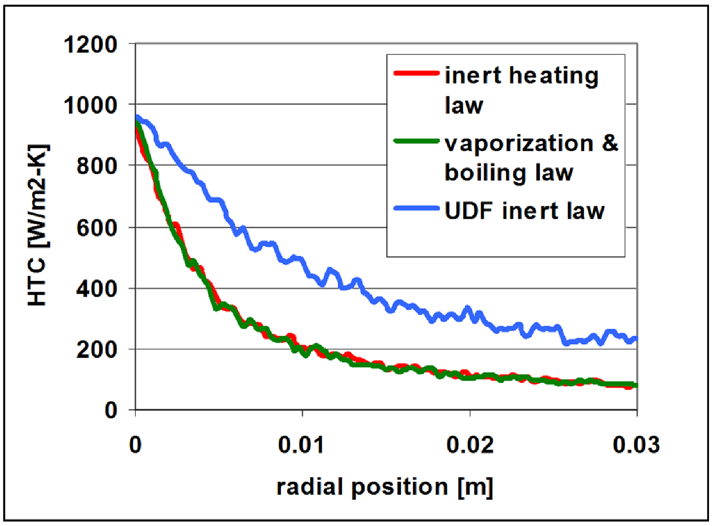


Fig. 68 *Heat transfer coefficient dependent on radial position for three different DPM laws*

To summarize, the application of the Euler-Lagrange model here on the modeling of the solid jet turned out to be just a bit suitable for a modeling of a cold flow i.e. velocity and pressure fields. The only advantage above the Euler-Euler model is that its computational costs are much lower and transient problems can be thus solved. Regarding the heat transfer calculation, the Euler-Lagrange model completely failed to predict even the order of the right HTCs.

To conclude this chapter, the Euler-Lagrange model is claimed to be not appropriate model for a modeling of solid jet flows especially those coupled with heat transfer problems.

5 Conclusions and Discussion

The title of thesis is Effect of flow parameters of water and air atomized sprays on cooling intensity of hot surfaces that obviously says nothing about procedures and approaches used to gain valuable results. Thus, the fact that solely CFD methods namely the commercial code Fluent were used is the first very important remark in conclusions. Furthermore, it is very important to note that unlike non-commercial in-house codes the CFD package Fluent as a general-purpose modeling tool does have a certain lack in flexibility and numerous lines must be inevitably respected. Anyway, even without additional user changes to Fluent's algorithms it is still capable CFD tool. As accidentally indicated it is possible to design your own user defined subroutines (UDFs) when a standard model suffers from incompleteness. UDFs in this thesis stand for a basic building stone.
The thesis is divided into two seemingly independent parts. The first part concerns numerical study of a single droplet denoted as a micro-scale point of view. The second part is dedicated to simulations of the entire water jet emanating from a solid jet nozzle and a flat fan nozzle using the Euler-Euler and the discrete phase model (DPM).

In the first part, the detailed description of the free surface tracking model (Volume of Fluid, VOF) precedes results from simulations. In brief, the only difference compared to single phase flow modeling is that the volume fraction is used to track and identify the interface position. It was found out that in addition to the standard convective stability condition (CFL) a special stability condition exists and must be obeyed. This stability condition prevents waves on free interface from being amplified. In single phase flow simulations the convective stability condition introduced by Courant number (CFL) should be less than 2. However, here in VOF simulations it was below 0.5.

The main objective of the first part was originally a comprehensive explanation of heat transfer coefficient (HTC) on a hot surface during a single water droplet impact and a consequent droplet spreading. Further, it was also intended to study the velocity field both outside and inside the moving droplet. First simulation settings were as the following. The droplet of the specified diameter was patched inside a two dimensional domain with the initial velocity of zero. In other words the droplet was static at the beginning of simulation. The gravity force, the capillary forces and the drag force were the only forces acting on the droplet. Naturally, the droplet velocity was expected to follow the Newton's law. Numerical solution of Newton's law (ODE solved in MATLAB) was used for verification of results from simulations. Note that the algorithm for Newton's law was also designed to account for the drag force considering both, constant and dynamic drag coefficients. Results from simulations unexpectedly varied significantly within the range of tested droplet diameters 2.0 - 0.2 mm. The case with 2 mm droplet diameter agreed perfectly with Newton's law solved numerically. However, the case with the droplet diameter of 0.2 mm was far from to be desired because of so-called spurious or parasitic currents that appeared during simulation and prevented the droplet from the physical acceleration. For the smallest diameters the droplet even didn't start falling towards the bottom surface, it rather kept on fluctuating around the same position. These parasitic currents were evidently caused by the surface tension model (CSF - Continuum Surface Force model). Because of the presence of spurious currents especially in droplet diameters common during spray cooling the HTC distribution could not be for that reason studied. Instead of running meaningless simulations seeking for HTC distribution, several different models were set up in order to give detailed information on parasitic currents, droplet terminal velocities obtained from simulations, and proposals how to get rid of parasitic currents.

The first modeling approach was aimed at terminal velocity of free falling droplet. Simulations were carried out in the droplet frame of reference in order to maintain the droplet in the same position i.e. the droplet was without move and there was rather a non-zero velocity field around it. This approach allowed for static mesh refinement along the interface and in places of large velocity gradients. With help of this approach, pressure and velocity field could be studied with success, but unfortunately only in the case of 2.0 mm droplet. This approach was also found to be suitable for calculation of terminal velocity (2.0 mm droplet ~ terminal velocity of 6.56 m/s). As regards the failure of the model in the case of 0.2 mm droplet, it was caused by some lacks of CSF model. Despite this weakness of the model, the terminal velocity (0.2 mm droplet) agreed quite well with real droplet behavior (terminal velocity of 1.8 m/s). The model was able to capture the recirculation area inside the droplet and the wake formed behind it.

The purpose of the second modeling approach was to simplify the previous model in terms of forgetting multiphase models and using single phase model instead. The circular droplet was introduced as an axisymmetric circle surrounded by walls. The momentum transfer across the wall was provided through User Defined Function (UDF). This study gave satisfactory results as regards velocity and pressure field in full range of tested diameters. Note that it was not capable of computing terminal velocity. The terminal velocity was actually used as inlet velocity boundary condition. The advantage of this model was that simulations could be run within the steady solver and thus it greatly reduced computational time consequently. Just one remark on momentum transfer through the interface should be made: The UDF accessed velocities in fluid cells (air fluid) adjacent to the interface (wall) and then imposed a 'relevant' portion of stored velocities in each of counter-cells in droplet bulk. The 'relevant' portion of velocity is always less than unity and in this case, it was determined from the previous simulations.

The third model directly simulated a droplet free-fall under gravity with the help of the dynamic grid adaption procedure implemented via journal file. This model was used to study the effect of the CSF model on the droplet acceleration during the free fall. The droplet with the diameter of 2.0 mm was tested only on mesh sensitivity and it was found out that the finer the mesh was, the lower the resulting acceleration was. The actual droplet speed was nearly linearly dependent on time; however, in simulations the finer mesh led to the drop in droplet speed. That's why this model was also the confirmation of several paper works that the CSF model diverges with the mesh refinement. While the 2 mm droplet was tested on mesh sensitivity, the 0.2 mm droplet was tested within the range of different surface tensions. The physical value of surface tension (0.072 N/m) even completely prevented the droplet from falling. The droplet rather kept on fluctuating around the initial position. The lower the value of surface tension was, the more realistic the droplet acceleration was. This malfunction of the standard surface tension model (the CSF model) initiated the next step that was aimed at how normals, curvatures, and the entire CSF model are implemented into Fluent code. Before this was done, several simulations were performed without gravity in order to objectively study parasitic currents in terms of maximal and mean velocity values. Only the 2.0 mm droplet was considered and as expected, errors in velocities grew up with the mesh refinement. Maximal values of parasitic currents for the finest mesh (0.025 mm) were around 0.25 m/s, whereas mean values were around 0.036 m/s for the same grid. It should be noted that the convective stability condition (CFL) was always considerably less than 2 that is the restriction for convective flows. Here, however, even more restrictive conditions were used in VOF calculations. The first capillary stability condition previously proposed by Brackbill is a function of fluid density and surface tension. The second one was derived for low Re flows, ignores fluid density, but takes into account viscosity. The second condition is also generally more restrictive. No matter which stability condition was used, results from simulations were always nearly the same.

As mentioned in upper lines, there was an author's intention to have a closer look at how normals, curvatures are calculated within Fluent code and whether any more accurate procedures could be implemented instead. The 2D circle (diameter of 2 mm filled with water) patched in rectangle (4x4 mm filled with air) was used as the topology for testing of different procedures of normal and curvature calculation. Note that exact normals were known a priori and thus, might be compared with those obtained numerically. As most of commercial CFD packages also Fluent calculates normal from gradient of volume fraction, which is the easiest way, but not as accurate as will be shown later. First, normals were calculated using standard macros available in Fluent namely C_VOF_G(c,t) and C_VOF_RG(c,t). Since the surface tension model comes from work by Brackbill, also the original procedure based on ALE-like scheme was implemented via UDFs and used for calculation of normals. The worst results were obtained with the gradient macro C_VOF_G(c,t). The reconstructed macro and the procedure based on ALE-like scheme gave quantitatively very similar results. The mean deviation from the actual normal was about 0.95° and the maximum deviation was approximately 2.5°. Contrary to these, say, standard procedures, the Height Functions were employed through UDFs and the maximum deviation dropped by 300% and the mean deviation dropped even by 400%. Unlike the foregoing procedures, the Height Functions can only be used for mapped meshes since it requires a construction of 7x3 (or 7x3) stencil in 2 dimensions around each cell containing a two phase interface. Further, based on calculated normals curvatures were determined for each interface cell. The ALE-like scheme gave evenly distributed errors no matter the orientation of normal. In some cells the error was even more than 100%. On the contrary Height Functions exhibited far more precise results especially when the most normal direction was oriented vertically or horizontally (5% error). However, when the normal direction was rather transverse to the grid orientation, then errors were unexpectedly even 40%. For this reason, the new method was proposed that is in principle based on construction of circles from three points. These points are the end points of linear segments that introduce the interface. The procedure calculates curvature from the distance between the circle center and the appropriate position of the linear segment. Besides normals and curvatures, also a new volume source term for surface tension was proposed. It takes into account the length of interface within the cell and should be applied not in interface cells but rather in cells within equidistant distance of dx (size of cell) from the interface inside the denser phase. Note that neither the new method for curvature calculation nor the new volume source term for surface tension were tested practically; however, it is expected to be more accurate than the standard methods and converge with mesh refinement.

To sum up, the original objective of calculation of HTC distribution within the spreading droplet was abandoned because serious problem with surface tension model appeared. This surface tension issue was studied instead and both, new methods and results from simulations, hopefully brings an important insight into the matter.

So far only the part denoted as the micro-scale model has been discussed; however, not a single word was said about the macro scale model that should shed light on possible ways of simulating an entire water jet as a complicated droplet structure coupled with heat transfer. Considering a water jet impinging onto a hot surface, several heat transfer mechanisms can be distinguished. When the liquid temperature is below the temperature of the boiling point, then only convective-conductive heat transfer can take place. When the liquid temperature is however enhanced above the boiling point, then a certain portion of water mostly adjacent to hot surface can be evaporated and consequently transformed into either separate vapor bubbles or continuous vapor layer depending mainly on the temperature of the hot surface. Note that the word 'jet' in general does not say anything about the flow pattern. It even does not say whether the continuous flow emanating from a nozzle exit breaks into droplets or rather stays continuous. The thesis concerns with both types of jets. The first type of jet is produced by a solid jet nozzle, which can be simply e.g. just a pipe with a sharp edged exit. The second type is a flat-fan nozzle that produces a chisel-like flow pattern.

Simulations of the two phase flow outside of nozzles generally require complex inlet boundary conditions that must be either obtained experimentally or with the help of CFD. Inlet boundary conditions can be quite easily obtained from single phase simulations. Author affords to supplement the thesis with some results on the flow modeling inside the full-cone nozzle (Lechler 460.844) that were formerly cited in his master thesis. Next, the flow inside the flat-fan nozzle Lechler (600.429.16.33) was simulated. Results such as velocities, volume fractions, and turbulent properties were later imposed in simulations of the entire jet as the inlet boundary conditions. Concerning the modeling of a solid jet, no separate single phase calculations of internal nozzle flow were performed because it was done simply together with the jet calculation within one single model. The main reason that it could be done this way is that the model was small enough in terms of number of finite volumes.

In the rest of oncoming lines the following topics will be discussed. Firstly, it will be the solid water jet impinging onto hot surface simulated using the VOF model. Secondly, it will be the single flat-fan nozzle as a representative of the whole nozzle array in the real continuous slab caster, more specifically above the first row of foot rolls, right underneath the mold bottom.

Numerical study of heat transfer between the solid jet and the hot surface required additional simulations that were used to tune up constants of k-epsilon turbulence model, namely the turbulent Prandtl number. Since the water jet spreading on surface is seemingly similar to the flow around the flat plate and reliable correlations for HTC exists for such a flow around a horizontally placed plate, simple 2D model was built up and suitable value of the turbulent Prandtl number was identified (1.3). It should be noted that the 2D model had to account for the transition from laminar to turbulent regime. The position of transition was however not calculated by the model, but it was rather defined by the correlation beforehand. Further, note that the value of 1.3 was optimal for the velocity range from 6 to 24 m/s that corresponded to radial velocities of the spreading jet. For smaller velocities the turbulent Prandtl number would be smaller and vice versa. First, a lower constant heat flux (398 kW/m^2) from heated surface was chosen in order to prevent the boiling regime. The nozzle of 2 mm diameter supplied the water flow rate of 1.2 l/min. The HTC distribution was studied up to radial distance of 30 mm. Two models were tested. The first one did not account for the transition from laminar to turbulent regime, whereas the second one did so. Results were compared with experimental data and were in the good agreement. In the case of the first model, the maximum relative error was 45%. In the second case, it was only 20%. Note that this model was capable of simulating convective heat transfer but not evaporation. The evaporation, actually the film boiling, was taken into account in the next step.

Two different models for film boiling were put together with the help of UDFs. The first model considered that the first portion of water will evaporate within the cell of temperature higher than 150°C. The most important assumption of the model was a steady behavior of the vapor layer, because it was designed for the steady state solver. Not many jobs were calculated, but all of them ran very fast. However, the flow was unfortunately unsteady in terms of fluctuating vapor layer. Hence, for this particular geometry and flow settings it could not be used with success. For different flow conditions it is likely to work and due to its robustness and fast computations, it could be very useful in some applications.

Unlike the first rather unphysical and non-conservative model, the second model had a physical background. The evaporation was provided through mass and momentum source terms based on heat fluxes going always through one particular cell containing interface. The evaporation could also start from the hot wall where no vapor cells were adjacent to this wall.

According to the direction of heat flux in each particular cell either the evaporation or the condensation could be present. This model came out from paper work of other researchers (Welch, Yuan). Results showed quite realistic behavior i.e. when the boiling was initiated, the jump in HTC and transition from continuous to oscillating profile was observed. However, the model was not compared with experiments and might require some minor modifications to be successfully used in any possible application.

In the thesis there is also a chapter that handles the problem of impinging solid jet in a completely different way. Instead of using the VOF model, the Discrete Phase Model was tested. The main idea consisted in significantly rougher boundary layer mesh and much lower computational costs consequently. It should be noted that using of finer mesh would not be anyhow beneficial anyway, since the rule for maximum volume fraction of dispersed phase was violated. Firstly only the cold flow namely velocity and pressure field were studied. Unlike the previous model, this model gave very uniform nearly rectangular profiles of axial velocities for the different distances from the nozzle exit. Moreover, although the interaction with surrounding air was considered, the momentum exchange was much lower than in the previous model. Nevertheless, core jet velocities were in a good agreement. On the contrary, it was not the case of radial velocities that varied much from those obtained in the previous model. Radial velocities in this model were descending much faster with increasing radial distance. As regards the pressure distribution within the nozzle footprint, the DPM model gave again almost rectangular distribution, while the previous VOF model gave more realistic Gauss-like distribution. Finally, several heat transfer calculations were performed. Firstly, only the convective heat transfer using the standard inert heating law was solved. The character of HTC distribution was relatively fine, but absolute values were far from to be desired (one order lower). This vast difference was attempted to be resolved by user defined inert heating law, which considered much higher HTC on the droplet surface. This approach however raised HTC only slightly and rather in higher radial distances than in jet axis. Employing vaporization and boiling laws gave similar results to those using only the inert heating law. This was accredited to very poor convergence especially for continuity equation that was caused by partial violation of the rule for volume fraction of droplets. At the time of being this model is not useful for simulations of impinging solid jets.

Besides the solid jet, the jet emanating from the particular flat-fan nozzle (600.429.16.33) was investigated. Unlike the nozzle producing solid jet, this particular flat fan nozzle was chosen, since it is mounted within the real continuous slab caster and stands for the basic unit in the secondary cooling system comprising hundreds of these nozzles, manifolds and high-performance pumps. To be more specific, in this case flat fan nozzle is mounted in the first row of nozzle just above the first row of foot rolls underneath the mold bottom. In foregoing paragraphs several words were mentioned on the modeling of internal flow. Here, the results namely profiles of several variables were used as inlet boundary conditions. It is appropriate to give notice on the complexity of the computational domain. In vertical direction the fluid region is very narrow e.g. the flat jet passes the foot roll only in distance of several millimeters. Further, the flat jet hits the very hot slab surface (>1000°C), its flow veers left, right, down, and up rapidly to be veered soon again by symmetry conditions, mold at the top, and the foot roll in the bottom. Moreover, the angle of zero can be found between the slab and the foot roll. This closely tighten up space must definitely cause tremendously unstable and turbulent flow that makes calculations more difficult.

It was decided to solve this using the Euler-Euler model. Since this model is not capable of predicting the secondary breakup, the breakup model based on WAVE model was developed and employed using UDFs. It was tested on 2D jet. Although it successfully predicted the evolution of parent droplet diameters, it did not account for diameters of child droplets. This could be overcome using some averaging method and thus obtain a mean diameter within each cell that would represent both, parent and child diameters. This was however considered only theoretically; therefore, only the constant droplet diameter was considered within the Euler-Euler model. Later on, using of the Euler-Euler model was not anyhow advantageous because it required very small time steps (5e-06 s). To make matters worse, the model suffered from the lack of water along the center line of flat jet which was pretty weird on the one hand but on the other hand it was confirmed by literature (St-Georges and Buchlin).

Because of these bottlenecks, the Discrete Phase Model was used instead. Inlet boundary conditions were determined from internal flow simulations. The primary breakup was simulated using the flat fan atomization model. As regards the secondary breakup, both, the Taylor Analogy (TAB) and the Blob jet breakup models, were tested. Experimental data (velocity and droplet distributions) were used to tune up the model constants. The TAB model gave markedly more accurate results than the Blob jet model. For the number of parcels of 2000 and streams count of 10 results agreed very well with experimental data. Since the discrete phase cannot be formed into a continuous fluid region and this is actually what seemingly happens in the gap between the slab and the foot roll, the model for coupling between the discrete phase and the continuous phase was developed and tuned in terms of mass conservation. To make a picture complete, the discrete phase was simulated by the DPM model, whereas the continuous phase was simulated by the VOF model. Because of a certain smearing of free interface the drag force acting on each droplet had to be modified i.e. up to a specific water volume fraction limit the droplet behaved like it was passing only through a cell containing 100% of air. Similar but much more complicated changes would have to be done to secondary breakup model, but in this case the secondary breakup was simply rather not taken into account. The model for coupling was quite robust, but most importantly, results were likely to be realistic.

In next step, the heat transfer was included but the temperature of mold was lowered from more than 1000°C to 200°C to avoid large source terms. For these ‘kick-off’ simulations the model for coupling was not considered. Several model settings were tested and especially strong dependence on boundary layer was identified. The thinner the first cell layer was, the worse convergence behavior was detected. Of course, this was caused by violating the rule for the maximum allowed volume fraction of dispersed phase within each particular cell. On the other hand, the thinner the layer was, the more accurate results were gained. Results were ranging from 10kW/m2 for rough mesh to 100kW/m2 for fine mesh. The reality can be still even one order higher. This incapability of the present model can be explained simply explained by the very low heat transfer between the hot slab and the droplet. The first responsible originator could be the missing radiation exchange between the slab and the semitransparent droplet. The second originator definitely is the low temperature in cell adjacent to hot slab that is consequently used for calculation for the calculation of evaporative heat transfer. It might be useful to hold temperature of near slab cells at temperature of the slab and perhaps it might be also useful and more realistic to keep the vapor fraction equal to unity. These proposals were however not put in practice.

To summarize the thesis it is convenient to define three different 'folders' in which each particular piece of this thesis can be filed separately. Let's start with the folder that does not sound just good. In here, we could leave numerical models that led into blind passage such as the simulation of 0.2 mm droplet free-falling and finally impinging on hot surface using the standard surface tension model or the Euler-Euler model in modeling of flat jet as a basic unit in continuous slab caster etc. The second folder sounds somewhat better, because in there we can place models and ideas that were partially able to solve the problem, but would require other subroutines or modifications that would lead to desired results. Frankly, it could not be brought to an end, since it would be too much extensive work that would definitely ask for more time to spend on it. In this second folder we could put e.g. the secondary breakup model for the Euler-Euler model or the new approach for implementation of Capillary forces. At last, the third folder can be filled in with models, simulations that worked reliably e.g. subroutines for calculation of interface normals and curvatures within the VOF model, numerical study of HTC distribution for solid jet spreading on hot surface with the temperature below the boiling point, simulation of flat jet using the DPM model, the model for coupling between the DPM and the VOF model.

The author is pretty sure that there might be some mistyped characters, discrepancies, imperfections, but in the same measure he is convinced that the thesis can yield notable profit to other researches.

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

symbol	unit	description
A	[m/s2]	acceleration
A	[Pa]	pressure jump across interface
A	[-]	exponent for calculation of weighted function
a	[m3/s]	aggregation kernel
a	[m]	diameter of parent droplet
A_D	[m2]	surface area of droplet
B	[-]	scalar parameter (0.01)
b	[kg/s]	damping coefficient
B_0	[-]	breakup model constant
B_1	[-]	breakup model constant
c	[m/s]	velocity magnitude
C_∞	[kgmol/m3]	vapor concentration in bulk gas
C	[m]	position of circle center
C	[-]	constant for coupling between VOF and DPM
c_1	[-]	constant for Brackbill's time step
c_2	[-]	constant for capillary time step
C_A	[kgmol/m3]	vapor concentration at droplet surface
C_D	[-]	drag coefficient
C_l	[-]	lift coefficient
C_p	[-]	constant that depends on quality of surface tension model
c_p	[J/kg-K]	heat capacity
D	[m]	characteristic length (droplet diameter)
D	[m]	distance function
d	[m]	droplet diameter
d	[m]	jet diameter
E	[-]	impact energy
F	[-]	fractional volume of liquid
F	[N/m3]	body force
F_b	[N]	buoyancy force
F_D	[s-1]	drag force
$\tilde{F}_{i,j}$	[-]	smoothed volume fraction F
F_{lift}	[N/m3]	lift force
F_{s-t}	[N/m3]	force resulting from surface tension
F_{vm}	[N/m3]	virtual mass force
g	[m/s2]	gravity acceleration
G_v	[m3/s]	grow rate of droplet
H	[-]	smoothed Heaviside function
h	[J/kg]	enthalpy
h	[W/m2-K]	heat transfer coefficient

h_0	[m]	height of film layer
h_{fg}	[J/kg]	latent heat of vaporization
$h_{i,j}$	[m]	height function in cell containing interface
K	[-]	kernel for smoothing of vole fraction
k	[J/K]	Boltzmann constant
K	[kg/m3-s2]	interface momentum exchange coefficient
k	[m/s]	mass transfer coefficient
k	[W/m-K]	thermal conductivity of bulk gas
k	[-]	slope of line
k	[N/m]	spring constant
k_t	[W/m-K]	turbulent thermal conductivity
K_w	[s-1]	quasi-equilibrium rate constant
L	[m]	capillary length
l	[m]	length of linear segment representing interface
L	[J/kg]	latent heat of vaporization
L_1	[m/s]	mass-weighted velocity of spurious currents
L_∞	[m/s]	maximal velocity of spurious currents
Ls	[m]	slip length
m	[kg]	mass of droplet
$\dot{m}_{gl}$	[kg/m3-s]	mass transfer from secondary phase to primary phase
$\dot{m}_{lg}$	[kg/m3-s]	mass transfer from primary phase to secondary phase
n	[droplets/m3]	number density function
n_i	[m]	unit normal to interface
N_i	[kgmol/m2-s]	molar flux of vapor
p	[Pa]	static pressure
Q	[m2.K/s2]	heat exchange between phases
Q	[l/min]	flow rate
Q	[W/m2]	heat flux through interface
Q_l	[kg/s]	mass flow rate of liquid (water)
r	[m]	cell-to-interface distance
R	[N/m3]	interaction force between phases
S	[kg/m2-s]	vapor generation rate
SD	[-]	sign of distance function
S_F	[kg/m3-s]	mass source
s_p	[droplets/s]	strength of droplet stream in DPM
t	[s]	time
T	[K]	absolute temperature
t	[s]	suitable time period
T_∞	[K]	free stream temperature
T_f	[K]	film temperature
T_s	[K]	surface temperature
U	[m/s]	free-stream velocity

U	[m/s]	velocity of three phase moving contact line
u	[m/s]	absolute velocity
U_d	[m/s]	local droplet velocity
u_i	[m/s]	vector of velocity
U_t	[m/s]	terminal velocity
v	[m/s]	constant translation velocity of droplet
V	[m3]	volume of droplet
V'	[m3/s]	droplet volume breaking per unit time
V_{cell}	[m3]	volume of computational cell
V_{sp}	[m3]	volume of secondary phase within computational cell
W	[-]	weighted function
x_i	[m]	position vector
Δp	[Pa]	pressure jump across interface
Δt	[s]	time step
Δx	[m]	size of particular cell in specified direction
Δy	[m]	size of particular cell in specified direction

Greek symbols

symbol	**unit**	**description**
α	[-]	fractional volume
γ	[m]	smoothing length
γ_f	[-]	blending factor
δ_l	[m]	boundary layer thickness
$\Delta\rho_{gl}$	[kg/m3]	difference between density of gas and liquid
ε	[m]	small number corresponding to size of interface cell
ε_{ijk}	[-]	Levi-Civita symbol
θ	[°]	contact angle
κ	[1/m]	interface curvature
λ	[m]	distance between adsorption sites
Λ	[m]	the most probable wavelength
μ	[kg/m-s]	dynamic viscosity
ν	[m2/s]	kinematic viscosity
ρ	[kg/m3]	fluid density
σ	[N/m]	surface tension
τ	[-]	dimensionless time
τ	[s]	breakup time
τ_{ij}	[N/m2]	stress tensor
τ_p	[s]	particle relaxation time
Φ	[]	arbitrary variable
$\tilde{\Phi}$	[-]	normalized variable
Φ	[m]	level set function (signed distance to interface)
Ω	[1/s]	the maximum grow rate

Indices

index	description
∞	free stream
A	value in acceptor cell
A	droplet surface
cell	particular cell
d	dynamic
f	face value
g	gas phase
i	i th cell
i+1	i+1 th cell
l	liquid phase
lg	liquid-gas
p	primary phase
p	particle, droplet
q	secondary phase
s	static
sg	solid-gas
sl	solid-liquid
sp	secondary phase
sp	refers to spurious current
U	value in upwind cell
η	viscosity taken into account
ρ	according to Brackbill
σ	combination of density and viscosity stability conditions

Dimensionless numbers

symbol	description
Re	Reynolds number
We	Weber number
T	Taylor number
Nu	Nusselt number
Ca	Capillary number
Pr	Prandtl number
Z	Ohnesorge number

LIST OF ABBREVIATIONS

1D	one dimensional
2D	two dimensional
3D	three dimensional
ALE	arbitrary Lagrangian-Eulerian computing method
BC	boundary condition
CAD	computer-aided design
CBC	convective boundedness criterion
CFD	computational fluid dynamics
CFL	Courant-Fridrichs-Lewy condition
CICSAM	compressive interface capturing scheme for arbitrary meshes
CSF	continuum surface force
CV	convolution of volume fraction
DAC	direction averaged curvature
DAN	direction averaged normal
DDS	donor differencing scheme
DPM	discrete phase model
FT	front tracking method
FVM	finite volume method
GFM	ghost fluid method
HF	height function
HRIC	high resolution interface capturing scheme
HTC	heat transfer coefficient
LBM	lattice Boltzmann method
LES	large eddy simulation
LINC	Lagrange-incompressible code
LSM	level set method
MAC	marker and cell method
NVD	normalized variable diagram
PBM	population balance model
PET	polyethylene
PLIC	piecewise linear interface construction
RDF	reconstructing distance function
RSM	Reynolds stress model
SPH	smoothed particle hydrodynamics
SSF	sharp surface force
TAB	Taylor analogy breakup
tke	turbulent kinetic energy
UDF	user defined function
UDS	user defined scalar
VOF	volume of fluid model

REFERENCES

[1] WACLAWCZYK, T., KORONOWICZ, T., *Modeling of the wave breaking with CICSAM and HRIC High-Resolution Schemes*, Proceedings of European Conference on Computational Fluid Dynamics, pp. 1-19, 2006

[2] HOKR, M., MARYŠKA, J., Influence of mesh geometry to numerical diffusion in upwind scheme for porous media solute transport, Proceedings of Algoritmy, pp. 123-131, 2005

[3] XING, X. Q., BUTLER, D.L., YANG, C., *Lattice Boltzmann-based single-phase method for free surface tracking of droplet motions*, International Journal for numerical methods in fluids, vol. 53, pp. 333-351, 2007

[4] UBBINK, O., Numerical prediction of two fluid system with sharp interfaces, Doctoral Thesis, University of London, 1997

[5] ŠTRUBELJ, L., TISELJ, I., Free surface modeling with two-fluid model and reduced numerical diffusion of the interface, Two-phase convective flow modeling, France, 2008

[6] SHEPEL, S.V., SMITH, B.L., Paolucci, S., *Implementation of a Level Set Interface Tracking Method in the FIDAP and CFX-4 Codes*, Journal of Fluids Engineering, vol. 127, pp. 674-687, 2005

[7] AFKAMI, S., BUSSMAN, M., *A Three-Dimensional Model of Droplet Impact Onto an Incline, Euromech Colloquium 479*, Numerical Simulation of Multiphase Flow with Deformable Interfaces, pp. 1-3, 2006

[8] PASANDIDEH-FARD, M., AZIZ, S.D., CHANDRA, S., MOSTAGHIMI, J., *Cooling effectiveness of a water drop impinging on a hot surface*, International Journal of Heat and Fluid Flow, vol. 22, pp. 201-210, 2001

[9] FRANCOIS, M., *Computations of drop dynamics with heat transfer*, Doctoral thesis, University of Florida, 2002

[10] HARVIE, D.J.E., FLETCHER, D.F., A hydrodynamic and thermodynamic simulation of droplet impacts on hot surfaces, Part II: validation and applications, International Journal of Heat and Mass Transfer 44, pp. 2643-2659, 2001

[11] GE, Y., FAN, L.-S., 3-D modeling of the dynamics and heat transfer characteristics of subcooled droplet impact on a surface with film boiling, International Journal of Heat and Mass Transfer, vol. 49, pp. 4231-4249, 2006

[12] WELCH, S.W.J., WILSON, J., *A Volume of Fluid Based Method for Fluid Flows with Phase Change*, Journal of Computational Physics, vol. 160, pp. 662-682, 2000

[13] FRACKOWIAK, B., STRZELECKI, A., LAVERGNE, G., *Tridimesional direct numerical simulation of droplet evaporation*, 2nd International Symposium on Non Equilibrium Processes, Combustion and Atmospheric Phenomenon SOTCHI, Russia, 2005

[14] YUAN, M.H., YANG, Y.H., LI, T.S., HU, Z.H., *Numerical simulation of film boiling on a sphere with a volume of fluid tracking method*, International Journal of Heat and Mass Transfer, vol. 51, pp. 1646-1657, 2008

[15] STROTOS, G., GAVAISES, M., THEODORAKAKOS, A., BERGELES, G., *Numerical investigation on the evaporation on droplets depositing on heated surfaces at low Weber numbers*, International Journal of Heat and Mass Transfer, vol. 51, pp. 1516-1529, 2007

[16] CAO, X., TOBITA, Y., *Drag Coefficient for a hot Particle/Droplet with Vapor Film*, Journal of Nuclear Science and Technology, vol. 38, pp. 721-728, 2001

[17] GUBAREV, V.Y., SHATSKIKH, Y.V., *Heat Transfer between a Gas-Droplet Medium and a High-Temperature Surface*, Heat and Mass Transfer and Physical Gasdynamics, vol. 43, pp.775-780, 2005

[18] GUNJAL, P.R., RANADE, V.V., CHADHARI, R.V., *Dynamics of Droplet Impact on Solid Surface: Experiments and VOF Simulations*, AIChe Journal, vol. 51, pp.59-79, 2005

[19] GANESAN, S., TOBISKA, L., Modeling and simulation of moving contact line problems with wetting effects, Comput. Visual Sci, pp.1-8, 2007

[20] HOCKING, L.M., *Rival contact-angle models and the spreading of drops*, J. Fluid Mech., vol 239, pp. 671-681, 1992

[21] COX, R.G., The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow, J. Fluid Mech., vol. 168, pp. 169-194, 1986

[22] RENABOTHU, S.R., *Dynamic wetting: hydrodynamic or molecular-kinetic?*, the Master thesis, Texas Technical University, 2004

[23] KANDLIKAR, S.G., STEINKE, M.E., Contact angles of droplets during spread and recoil after impinging on a heated surface, Trans IChemE, vol. 79,pp. 491-499, 2001

[24] LUNKAD, S.F., BUWA, V.V., NIGAM, K.D.P., *Numerical simulations of drop impact and spreading on horizontal and inclined surfaces*, Chemical Engineering Science, vol. 62, pp. 7214-7224, 2007

[25] HALLER, K.K., VENTIKOS, Y., POULIKAKOS, D., MONKEWITZ, P., *Computational study of high-speed liquid droplet impact*, Journal of Applied Physics, vol. 92, pp. 1-8, 2002

[26] ADLER, W.F., *Waterdrop impact modeling*, Wear, vol. 186, pp. 341-351, 1995

[27] CHO, M.J., THOMAS, B.G., LEE, P.L., *Three-Dimensional Numerical Study of Impinging Water Jets In Runout Table Cooling Processes*, Metallurgical and Materials Transactions B, vol. 39, pp. 593-603, 2008

[28] KULJU, T., MAURINEN, E., KEISKI, R., *Filmboiling studies with CFD*, Proceedings of the EnePro conference, pp. 117-119, 2009

[29] NARUMENCHI, S., TROSHKO, A., BHARATHAN, D., HASSANI, V., *Numerical simulations of nucleate boiling in impinging jets: Applications in power electronics cooling*, International Journal of Heat and Mass Transfer, vol. 51, pp. 1-12, 2008

[30] SENDA, J., KOBAYSHI, M., IWASHITA, S., FUJIMOTO, H., *Modeling od Diesel Spray Impinging on Flat Wall*, Proceedings of International Symposium COMODIA, pp. 411-417, 1994

[31] GROVER, R.O., ASSANIS, D.N., *A spray wall impingement model based upon conservation principles*, report for W.E. Lay Automotive Laboratory, 2001

[32] YAO, S.C., HUNG, L.S., *Investigation of Droplet Penetrations through Complex Openings of Compartments*, Report for National Institute of Standards and Technology, pp. 1-79, 1998

[33] TONINI, S., GAVAISES, M., THEODORAKAKOS, A., *Modeling of high-pressure dense diesel sprays with adaptive local grid refinement*, International Journal of Heat and Fluid Flow, vol. 29, pp. 427-448, 2008

[34] BHATTACHARYA, P., SAMANTA, A.N., CHAKRABORTY, S., *Spray evaporative cooling to achieve ultra fast cooling in runout table*, International Journal of Thermal Sciences, vol. 48, pp. 1741-1747, 2009

[35] TRYGGVASON, G., BUNNER, B., ESMAELI, A., JURIC, D., AL-RAWAHI, N., TAUBER, W., HAN, J., NAS, S., JAN, Y.-J., *A Front Tracking Method for the Computations of Multiphase Flow*, J. Comput. Phys. 169, pp. 708-759, 2001

[36] HOU, T.Y., LOWENGRUB, J.S., SHELLEY, M.J., *Boundary Integral Methods for Multicomponent Fluids and Multiphase Materials*, J. Comput. Phys. 169, pp. 302-323, 2001

[37] JACQMIN, D., Calculation of Two-Phase Navier-Stokes Flows Using Phase-Field Modeling, J.Comput. Phys., 155, pp. 96-127, 1999

[38] JAMET, D., LEBAIGUE, O., COUTRIS, N., DELHAYE, J.M., The Second Gradient Method for the Direct Numerical Simulation of Liquid-Vapor Flows With Phase Change, J.Comput. Phys. 169, pp. 624-651, 2001

[39] SETHIAN, J.A., *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry*, Fluid Mechanics, Computer Vision, and Materials science, Cambridge University Press, Cambridge, England, 1999

[40] OSHER, S., FEDKIW, R.P., *Level Set Methods: an Overview and some Recent Results*, J.Comput. Phys. 169, pp. 463-502, 2001

[41] NICHOLS, B. D. , HIRT, C. W. AND HOTCHKISS, R. S. , *A fractional volume of fluid method for free boundary dynamics*, Seventh International Conference on Numerical Methods in Fluid Dynamics, Volume 141/1981, pp. 304-309

[42] G. STROTOS, M. GAVAISES, A. THEODORAKAKOS, G. BERGELES, *Numerical investigation of the cooling effectiveness of a droplet impinging on a heated surface*, Heat Mass Transfer, 2008, doi:10.1016/j.ijheatmasstransfer.2008.02.036

[43] BRACKBILL, J.U., KOTHE, D.B., ZEMACH, C., *A continuum method for modeling surface tension*, J. Comput. Phys. 100 (1992) 335–354

[44] GALUSINSKI, C., VIGNEAUX, P., *On stability condition for bifluid flows with surface tension: Application to microfluidic*, Journal of Computational Physics volume 227, pp. 6140-6164, 2008

[45] UBBINK, O., ISSA, R.I., Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes, J. Comput. Phys., 153, 26-50, (1999)

[46] MUZAFERIJA, S., PERIC, M., SAMES, P., SCHELIN, T., *A two-fluid Navier-Stokes solver to simulate water entry*, Twenty-Second Symposium on Naval Hydrodynamics, (1998)

[47] LEONARD, B.P., The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, Comput. Methods Appl. Mech. and Engng., 88, 17-74, (1991)

[48] SUSSMAN, M., AND PUCKETT, E. G., A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows, J.Comput. Phys. 162, pp. 301-337, 2000

[49] SUSSMAN, M., AND FATEMI, E., An Efficient Interface-Preserving Level Set Redistancing Algorithm and its Application to Interfacial Incompressible Fluid Flow, SIAM J. Sci. Comput. (USA) 20, pp. 1165-1191, 1997

[50] SUSSMAN, M., SMEREKA, P., OSHER, S., *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*, J.Comput. Phys. 114, pp. 146-159, 1994

[51] http://en.wikipedia.org/wiki/Surface_tension

[52] http://www.fluentusers.com/fluent/doc/ori/v120/fluent/fluent12.0/help/html/th/node305.htm

[53] VINCENT, S., CALTAGIRONE, J., Test-case number 10: Parasitic currents induced by surface tension (PC), 2003

[54] KANG, M., FEDKIW, R.P., LIU, X.-D., A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360

[55] GUEYFFIER, D., NADIM, A., LI, J., SCARDOVELLI, R., AND ZALESKI, S., *Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows*, J. Comput. Phys., 152(1998) 423-456.

[56] AFKAMI, S., BUSSMANN, M., *Height Function-Based contact angles for VOF simulations of contact line phenomena*, International Journal of Numerical Analysis and Modeling, Volume 1, pp. 1-18, 2004

[57] AFKHAMI, S., BUSSMANN, M., *Height functions for applying contact angle to 2D VOF simulations*, International Journal for Numerical Methods in Fluids, Volume 57, pp. 453-472, 2008

[58] RUDMAN, M., A Volume-tracking Method for Incompressible multifluid flows with large density variations, Int. Journal for Numerical methods in Fluids , Volume 28, pp 357-378, 1998

[59] LORSTAD, D., FUCHS, L., *High-order surface tension Vof-model for 3D bubble flows with high density ratio,* Journal of Computational Physics, Volume 200, pp. 153 – 176, 2004

[60] LORSTAD D., *Numerical modeling of deforming bubble transport related to cavitating hydraulic turbines*, PhD Thesis, Department of Heat and Power Engineering, Lund University, 2003. Free download at: http://www.fm.vok.lth.se/.

[61] LORSTAD, D., FRANCOIS, M., SHYY, W., FUCHS, L., *Assessment of Volume of Fluid and immersed boundary methods for droplet calculations*, Int. J. Numer. Methods Fluids to be published, 2004

[62] CUMMINS, S. J., FRANCOIS, M. M., KOTHE, D. B., *Estimating curvature from volume fractions*, Computers and Structures, Volume 83, pp. 425-434, 2005

[63] BLAKE, T. D., BERG, J.C., DEKKER M., *Dynamic Contact Angle and Wetting Kinetics, In Wettability*, New York, pp 251, 1993

[64] SCHILLER, L., NAUMANN, Z., Z.Ver.Deutsch. Ing., pp.77-318, 1935

[65] RANZ, W.,E., MARSHALL, JR., *Evaporation from drops, part I.*, Chem. Eng. Prog. Vol. 48, pp. 141-146, 1952

[66] RANZ, W.,E., MARSHALL, JR., *Evaporation from drops, part II.*, Chem. Eng. Prog. Vol. 48, pp. 173-180, 1952

[67] KUO, K. K. Y., *Principles of Combustions*, John Wiley and Sons, New York, 1986

[68] http://www.hr.lanl.gov

[69] NABER, J.D., REITZ, R.D., *Modeling engine spray/wall impingement*, SAE Paper, No.881316, pp. 1-23, 1988

[70] SENDA, J., KOBAYASHI, M., IWASHITA, S., FUJIMOTO, H., *Modeling on diesel Spray Impinging on Flat Wall*, International Symposium COMODIA 94, 1994

[71] GROVER, R. O., ASSANIS, D.N., *A Spray Wall Impingement Model Based Upon Conservation Principles*, W.E. Lay Automotive Laboratory, University of Michigan

[72] MUNDO, C., SOMMERFELD, M., TROPEA, C., *Droplet-Wall Collisions: Experimental Studies of the Deformation and Breakup Process*, International Journal of Multiphase Flow, vol. 21, pp. 151-173, 1995

[73] O'ROURKE, P.J., AMSDEN, A.A., A Spray/Wall Interaction Submodel for the KIVA-3 Wall Film Model, SAE Paper, 2000

[74] TAYLOR, G. I., *The Shape and Acceleration of a Drop in a High Speed Air Stream*, Technical report, In the Scientific Papers of G. I. Taylor, ed., G. K. Batchelor, 1963

[75] LUXFORD, G., Experimental and modelling investigation of the deformation, drag and break-up of drizzle droplets subjected to strong aerodynamic forces in relation to SLD aircraft icing, PhD thesis, Cranfield university, 2005

[76] http://www.fluentusers.com/fluent/doc/ori/v121/fluent/fluent12.1/help/html/udf/node97.htm

[77] BOHACEK, J., *Ejector type cooling nozzle*, Diploma thesis, Brno University of Technology, 2006

[78] P. K. DAS, *Monte carlo simulation of drop breakage on the basis of drop volume*, Computers chem. Engng Vol. 20, No. 3, pp. 307-313, 1996

[79] REITZ, R. D., *Mechanisms of Atomization Processes in High-Pressure Vaporizing Sprays*, Atomization and Spray Technology, Volume 3, pp.309-337, 1987

[80] GANT, S.E., CFD Modelling of Water Spray Barriers, Health & Safety Laboratory, project, 2006

[81] ST-GEORGES, M. AND J.M. BUCHLIN, *Detailed single spray experimental measurements and one-dimensional modelling*, Int. J. Multiphase Flow, vol. 20(6): pp. 979-992, 1994

[82] INCROPERA, F. P., DE WITT, D. P., *Fundamentals of Mass and Heat Transfer*, John Wiley & Son, Third edition, pp. 393-395, 1990

[83] http://www.fluentusers.com/fluent/doc

[84] WU, S., SHIN, C. H., KIM, K. M., CHO, H. H., Single-phase convection and boiling heat transfer: Confined single and array-circular impinging jets

[85] HALL, D.E., INCROPERA, F. P., VISKANTA, R., Jet Impingement Boiling from a circular free-Surface Jet During Quenching: Part 1 – Single-Phase Jet, Journal of Heat Transfer, vol. 123, pp. 901-910, 2001

[86] ZUCKERMAN, N., LIOR, N., Jet Impingement Heat Transfer: Physics, Correlations, and Numerical Modeling, Advances in Heat Transfer, vol. 39, pp. 565-631

AUTHOR'S PUBLICATIONS

[1] BOHÁČEK, J., *Ejector type cooling nozzle*, Experimental Fluid mechanics 2006, pp.30-38, ISBN 80-7372-141-4, (2006), Liberec

[2] HORÁK, A.; BOHÁČEK, J., *High- speed droplet impact during hydraulic descaling process*, IM2007, pp.79-80, ISBN 978-80-87012-06-2, (2007), Academy of Sciences of the Czech Republic

[3] HNÍZDIL, M.; BOHÁČEK, J., *Obrazová analýza vodního proudu*, Engineering Mechanics 2007, pp.79-80, ISBN 978-80-87012-06-2, (2007), Academy of Science of the Czech Republic

[4] KOTRBÁČEK, P.; HORSKÝ, J.; BOHÁČEK, J., *Modification of billet continuous casting machine from square to round geometry*, 6th European Conference on Continuous Casting, pp.1-10, ISBN 88-85298-65-6, (2008), associazione Italiana di Metallurgia

[5] VAN STEDEN, H.; BOHÁČEK, J.; RAUDENSKÝ, M.; KOTRBÁČEK, P., *Intensification of roll cooling in hot rolling with a roll wiper*, Research Disclosure Journal, Vol.2009, (2009), No.546003, pp.1-8, ISSN 0374-4353, Kenneth Mason Publications Ltd

[6] RAUDENSKÝ, M.; BOHÁČEK, J., *Leidenfrost Phenomena at Hot Sprayed Surface*, 7th ECI International Conference on Boiling Heat Transfer, pp.1-9, (2009)

[7] BOHÁČEK, J., *Modelling of a Free-Falling Water Droplet, Handicap of CSF Approach and Its Improvement Design*, Vodní paprsek, pp.19-28, ISBN 978-80-86407-81-4, (2009), Ústav geoniky AVČR,v.v.i.

[8] BOHÁČEK, J., HORÁK, A., Numerical study of droplet dynamics impinging onto steel plate covered with scale layer, Front. Mech. Eng. China 2010, 5(4): 389–398

[9] HORSKÝ, J.; KOTRBÁČEK, P.; HRABOVSKÝ, J.; HNÍZDIL, M.; BOHÁČEK, J., *Spray Nozzle Testing for HTC Calculation*, ArcelorMittal Chicago, USA

[10] BOHÁČEK, J., *Surface Tension Model for High Viscosity Ratios implemented in VOF model*, Proceedings of 23rd European Conference on Liquid Atomization and Spray Systems, pp.51-52, ISBN 978-80-7399-997-1, (2010)

APPENDICES

APPENDIX I

Free-falling droplet – Runge–Kutta method, variable drag coefficient

Newton’s law was solved in Matlab considering variable drag coefficient. The complete M-file script is presented below with comments. Firstly, constants and variables are defined. Secondly, Runge-Kutta method is employed taking into account the new drag law. Finally, the post-processing is carried out.

```
% numerical solution of Newton’s law considering
% drag coefficient is considered to be velocity dependent
% Runge-Kutta method
clc; clear all; close;
v0 = 0; % initial velocity
mu = 0.001; % dynamic viscosity of water
mu_air = 0.000018; % dynamic viscosity of air
ro = 1000; % density of water
ro_air = 1.25;
g = 9.81; % acceleration of gravity
D = 0.002; % droplet diameter
A = pi*D^2/4; % front surface driving drag force
V = pi*D^3/6; % droplet volume
m = ro*V; % droplet weight
dt = 0.001; % time step
sterm = 30; % terminating trajectory, droplet touches the surface
s = 0; % initialization of trajectory
t = 0; % initialization of time
k1 = 0; k2 = 0; % coefficients following from midpoint method; initialization
fid = fopen('data_2.0mm.txt', 'wt');
p = load('coefficients.txt');
tterm = 0.1;
while (s <= sterm)
Re = ro_air*v0*D/mu_air;
   if (Re < 0.05875) % smallest Reynolds number determined from exp
      Cd = 492;
   else
      Cd = Re^p(1,2)*exp(p(1,1)*log(Re)^2+p(1,3)); % new drag law
   end
k1 = (dt/m)*(-(1/2)*ro_air*Cd*A*(v0)^2 + m*g);
k2 = (dt/m)*(-(1/2)*ro_air*Cd*A*(v0+k1/2)^2 + m*g);
k3 = (dt/m)*(-(1/2)*ro_air*Cd*A*(v0+k2/2)^2 + m*g);
k4 = (dt/m)*(-(1/2)*ro_air*Cd*A*(v0+k3)^2 + m*g);
v = v0 + k1/6+k2/3+k3/3+k4/6;
s = s + (v0+v)*dt/2;
v0 = v;
t = t+dt;
fprintf(fid, '%2.8f %2.8f %2.8f %4.8f\n ', t, v, s, Cd); % print elapsed time, velocity, trajectory, drag coefficient to file
end
%%
% plot both, velocity and drag coefficient
fclose(fid);
data = load('data_2.0mm.txt');
[haxes,hline1,hline2] = plotyy(data(:,1),data(:,2),data(:,1),data(:,4),'loglog');
xlabel('Time elapsed [sec]');
set(haxes(1),'YTick',[0 0.1 0.2 0.4 0.8 1.2 2 4 6.37]);
grid on;
set(haxes(2),'YTick',[0.51 1 10 100 500]);
grid on;
set(haxes(1),'XTick',[0.001 0.01 0.1 1 2]);
set(haxes(2),'XTick',[]);
axes(haxes(1));
ylabel('Velocity [m/s]');
```

```
ylim([0 10]);
axes(haxes(2));
ylabel('Drag coefficient [-]');
ylim([0 500]);
set(hline1,'LineWidth',2);
set(hline2,'LineWidth',2);
```

APPENDIX II

UDF and script file for setting variable velocity inlet in model for determining terminal velocity of free-falling droplet

The below mentioned script file is written in SCHEME programming language. The principal purpose is to execute the UDF function and preset the velocity inlet BC to a new calculated velocity, consequently.
The UDF calculates the droplet velocity using the following formula:

$$u = \frac{\sum_{i=1}^{n} m_i u_i}{m}, \qquad \text{Eq.121}$$

where *n* corresponds to the cells that contain the volume fraction $F>0.9$. Further, the difference between the velocity droplet BC and the droplet velocity is either added or subtracted from the velocity inlet BC.

script file:

```
(define (first_step)
    (ti-menu-load-string "solve set time-step 3.3e-07 ")   ;; define time step
   (rp-var-define 'vel 0.0 'real #f)                        ;; assign the value of 0 to vel variable of real
type
)
(define (set_velocity)
  (let
   (
    (break #f)
   )

   (do ((i 0 (+ i 1)))   ((or(> i 5000000)break ))
     (set! break (not (and
           (and
             (ti-menu-load-string "solve dual-time-iterate 1 15 ")    ;; number of iteration per time step
             (ti-menu-load-string "define/user-defined/execute-on-demand \"calc_vel::libudf_vel2\" ")
;;execute the UDF
             (ti-menu-load-string "define/b-c/velocity-inlet air_inlet mixture no no yes yes no (rpgetvar 'vel)
")  ;; adjust the velocity inlet BC
             )
     )))
   )
   (if break (begin (newline)(newline)(display "job interrupted!")(newline)))
  )
)
```

UDF:

```
/*************************************************************************
   UDF that calculates droplet velocity
*************************************************************************
#define USE_FLUENT_IO_API 0
#include "udf.h"

DEFINE_ON_DEMAND(calc_vel)

{
  real vel = 0.0, mom = 0.0, vol = 0.0, dr_vel = 0.0;
  cell_t c;
```

```
  int zone_ID = 2;
  Thread *water_thread;
  Domain *water_domain;
  water_domain = Get_Domain(3);
  water_thread = Lookup_Thread(water_domain,zone_ID);
  vel = RP_Get_Real("vel");

  begin_c_loop(c, water_thread)
  {
    if (C_VOF(c,water_thread) > 0.95)
    {
     mom +=
C_VOF(c,water_thread)*C_VOLUME(c,water_thread)*C_U(c,water_thread);
     vol += C_VOF(c,water_thread)*C_VOLUME(c,water_thread);
    }
  }
  end_c_loop(c, water_thread)
  vel += mom/vol;
  dr_vel = mom/vol;
  Message("droplet momentum is %f \n", mom);
  Message("droplet volume is %f \n", vol);
  Message("droplet velocity is %f \n", dr_vel);
  RP_Set_Real("vel", vel);
  Message("inlet air velocity is %f \n", vel);
}
```

APPENDIX III

Model settings for model calculating terminal velocity of free-falling droplet

Material properties, solver settings, boundary conditions are presented here.

```
FLUENT
Version: axi, dp, pbns, vof, lam, unsteady (axi, double precision, pressure-based, VOF, laminar, unsteady)

Models
------

   Model                    Settings
   -----------------------------------------------------
   Space                    Axisymmetric
   Time                     Unsteady, 1st-Order Implicit
   Viscous                   Laminar


Boundary Conditions
-------------------

   Zones

      name              id   type
      -------------------------------------
      mixture            2    fluid
      air_outlet         4    pressure-outlet
      surrounding_wall   10   wall
      air_inlet          5    velocity-inlet
      axis               3    axis
      default-interior   7    interior

   Boundary Conditions


      air_outlet

         Condition                          Value
         ---------------------------------------------
         Gauge Pressure (pascal)                0

      surrounding_wall

         Condition                                   Value
         ----------------------------------------------------------------------------------------

         X-component of shear stress (pascal)             0
         Y-component of shear stress (pascal)             0

      air_inlet

         Condition                       Value
         ---------------------------------------------
         Velocity Specification Method        2
         Reference Frame                  0
         Velocity Magnitude (m/s)            determined by UDF


Solver Controls
---------------

   Equations
```

```
    Equation        Solved
    ------------------------
    Flow            yes
    Volume Fraction   yes

  Numerics

    Numeric                        Enabled
    ---------------------------------------
    Absolute Velocity Formulation   yes


  Relaxation

    Variable     Relaxation Factor
    -------------------------------
    Pressure     1
    Density      1
    Body Forces   1
    Momentum     1

  Linear Solver

              Solver     Termination   Residual Reduction
    Variable     Type       Criterion    Tolerance
    --------------------------------------------------------
    Pressure     V-Cycle    0.1
    X-Momentum   Flexible   0.1          0.7
    Y-Momentum   Flexible   0.1          0.7

  Pressure-Velocity Coupling

    Parameter                    Value
    -----------------------------------
    Type                        PISO
    Skewness-Neighbour Coupling   yes
    Skewness Correction           1
    Neighbour Correction          1

  Discretization Scheme

    Variable         Scheme
    -------------------------------------
    Pressure         PRESTO!
    Momentum         Second Order Upwind
    Volume Fraction   CICSAM

  Solution Limits

    Quantity                    Limit
    ---------------------------------
    Minimum Absolute Pressure   1
    Maximum Absolute Pressure   5e+10
    Minimum Temperature         1
    Maximum Temperature         5000

Material Properties
-------------------
```

```
Material: water (fluid)

   Property                 Units     Method    Value(s)
   ---------------------------------------------------------------
   Density                  kg/m3     constant  998.2
   Viscosity                kg/m-s    constant  0.001003

Material: air (fluid)

   Property                 Units     Method    Value(s)
   -----------------------------------------------------------------
   Density                  kg/m3     constant  1.225
   Viscosity                kg/m-s    constant  1.7894e-05
```

APPENDIX IV

UDF for momentum exchange between water and ambient air

The simple procedure was employed to exchange the momentum between the two phases, the water and the air. In the air zone in defined normal distance from the interface, the velocity field is copied and transferred to the droplet interface/wall belonging to the water region. It is rather an artificial approach; however, it can be apparently tuned to get the very similar results to those obtained in section 3.5.1.

```
/**********************************************************************
 UDF that provides exchange of momentum through water-air interphase
***********************************************************************/
#define USE_FLUENT_IO_API 0
#include "udf.h"

DEFINE_ADJUST(set_vel,d)
{
    int water_ID = 7;                                            /* put the
ID of water-air interphase belonging to water */
    int air_ID = 3;                                              /* put the
ID of air */
     real distance = 0.000006, radius = 0.0001, dev;             /* distance
is distance from interface to point, where velocity vector is copied from
*/
     real x[2], xc[2], vel_axial, vel_radial, X_pos[2];          /* radius
of droplet, xc position of droplet */
     cell_t c;
     face_t f;
     Thread *water_th;
     Thread *air_th;

     water_th = Lookup_Thread(d,water_ID);
     air_th = Lookup_Thread(d,air_ID);

begin_f_loop(f,water_th)                                         /* loops
over faces in a intphs_air_th thread  */
  {
   F_CENTROID(x,f,water_th);
   X_pos[0] = 0.011 + (radius + distance)*(x[0] - 0.011)/radius;
   X_pos[1] = (radius + distance)*x[1]/radius;
   dev = 0.00001;
   vel_axial  = 0.0;
   vel_radial = 0.0;

    begin_c_loop(c, air_th)                                      /* loops
over cells in a air cell thread  */
    {
     C_CENTROID(xc,c,air_th);
       if ((pow((pow(xc[0]-X_pos[0],2)+pow(xc[1]-X_pos[1],2)),0.5)) < dev)
       {
      dev = (pow((pow(xc[0]-X_pos[0],2)+pow(xc[1]-X_pos[1],2)),0.5));
      vel_axial  = C_U(c,air_th);
      vel_radial = C_V(c,air_th);
       }
    }
    end_c_loop(c, air_th)

   F_U(f,water_th) = vel_axial;
   F_V(f,water_th) = vel_radial;
  }
end_f_loop(f,water_th)
}
```

APPENDIX V

UDF and script for auto grid adaption

Both, the script file named auto_adaption and the UDF named mark_for_refinement, are coupled together. The purpose of the script file is to start iterating, to redefine text user interface (TUI) variables described in Tab. 19, to execute the UDF, to perform a grid refinement and a coarsening within each refinement level, respectively. All those commands are performed within a loop that is interrupted when a prescribed time is exceeded.

The UDF accesses aforementioned variables defined in the TUI. For each refinement level, the single set of those variables exists. The UDF detects the droplet interface or rather the cells containing the interface (0.1<F<0.9). The strip of circles of radius of *rad_lim* is then constructed along the interface. In other words, the interface cells represent the centers of those circles. The user defined memory (UDM) is allocated. If the grid cell is situated in the strip of circles, it marked for refinement. Else it is either marked for coarsening or refinement depending on whether the cell is an air cell or a water cell and whether the value of the *drop_ref* is 1 or 0.

The combination of the script file and the UDF proved to be very flexible and suitable in cases, where automatic dynamic adaption fails.

Tab. 19 *description of text user interface variables used*

TUI variable	description
rad_lim	radius of the circle different for each level of refinement; the grid is refined within this circle
vol_lim	limit of cell volume stands for a threshold assisting by the decision if the grid should be coarsened or not
drop_ref	Integer number (0 or 1); if 1 the grid will be refined inside the droplet interior, if 0 the grid will be preserved.

script file:

```
(define (auto_adaption)
  (let
   (
    (break #f)
   )
   (do ((i 0 (+ i 1)))   ((or(> i 1000)break ))
     (set! break (not (and
           (and
             (ti-menu-load-string "solve set time-step 4e-06 ")
             (rp-var-define 'rad_lim 0.001 'real #f)
             (rp-var-define 'vol_lim 1e-07 'real #f)
             (rp-var-define 'drop_ref 1 'integer #f)
             (ti-menu-load-string "solve dual-time-iterate 20 20 ")

             (ti-menu-load-string "adapt free-registers yes ")
             (ti-menu-load-string "adapt set init-node-flags ")
             (ti-menu-load-string "adapt set min-cell-volume 1e-7 ")
             (ti-menu-load-string "define/user-defined/execute-on-demand
\"mark_for_refinement::libudf\" ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 0.5 1 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 1.5 2 ")
             (ti-menu-load-string "adapt exchange-marks 0 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
```

```
             (ti-menu-load-string "adapt free-registers yes ")
             (ti-menu-load-string "adapt set init-node-flags ")
             (ti-menu-load-string "adapt set min-cell-volume 4e-8 ")
             (rpsetvar 'rad_lim 0.0005)
             (rpsetvar 'vol_lim 4e-8)
             (rpsetvar 'drop_ref 1)
             (ti-menu-load-string "define/user-defined/execute-on-demand
\"mark_for_refinement::libudf\" ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 0.5 1 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 1.5 2 ")
             (ti-menu-load-string "adapt exchange-marks 0 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")

             (ti-menu-load-string "adapt free-registers yes ")
             (ti-menu-load-string "adapt set init-node-flags ")
             (ti-menu-load-string "adapt set min-cell-volume 1e-8 ")
             (rpsetvar 'rad_lim 0.0003)
             (rpsetvar 'vol_lim 1e-8)
             (rpsetvar 'drop_ref 1)
             (ti-menu-load-string "define/user-defined/execute-on-demand
\"mark_for_refinement::libudf\" ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 0.5 1 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 1.5 2 ")
             (ti-menu-load-string "adapt exchange-marks 0 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")

             (ti-menu-load-string "adapt free-registers yes ")
             (ti-menu-load-string "adapt set init-node-flags ")
             (ti-menu-load-string "adapt set min-cell-volume 2e-9 ")
             (rpsetvar 'rad_lim 0.00015)
             (rpsetvar 'vol_lim 2e-9)
             (rpsetvar 'drop_ref 0)
             (ti-menu-load-string "define/user-defined/execute-on-demand
\"mark_for_refinement::libudf\" ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 0.5 1 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 1.5 2 ")
             (ti-menu-load-string "adapt exchange-marks 0 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")

             (ti-menu-load-string "adapt free-registers yes ")
             (ti-menu-load-string "adapt set init-node-flags ")
             (ti-menu-load-string "adapt set min-cell-volume 5e-10 ")
             (rpsetvar 'rad_lim 0.0001)
             (rpsetvar 'vol_lim 5e-10)
             (rpsetvar 'drop_ref 0)
             (ti-menu-load-string "define/user-defined/execute-on-demand
\"mark_for_refinement::libudf\" ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 0.5 1 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 1.5 2 ")
```

```
             (ti-menu-load-string "adapt exchange-marks 0 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")

             (ti-menu-load-string "adapt free-registers yes ")
             (ti-menu-load-string "adapt set init-node-flags ")
             (ti-menu-load-string "adapt set min-cell-volume 1.5e-10 ")
             (rpsetvar 'rad_lim 0.00005)
             (rpsetvar 'vol_lim 1.5e-10)
             (rpsetvar 'drop_ref 0)
             (ti-menu-load-string "define/user-defined/execute-on-demand
\"mark_for_refinement::libudf\" ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 0.5 1 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
             (ti-menu-load-string "adapt mark-inout-iso-range yes mixture
udm-0 1.5 2 ")
             (ti-menu-load-string "adapt exchange-marks 0 ")
             (ti-menu-load-string "adapt adapt-to-register 0 0 0 yes ")
           )
     )))
   )
   (if break (begin (newline)(newline)(display "job
interrupted!")(newline)))
  )
)
```

UDF:

```
/* marking cells for refinement or coarsening */
#define USE_FLUENT_IO_API 0
#include "udf.h"

DEFINE_ON_DEMAND(mark_for_refinement)
{

  real radius_limit,volume_limit,l;  /* distance from actual cell centroid
and circle centre */
  int drop_refin,zone_ID = 2;        /* put the zone_ID requested */
  cell_t c;
  cell_t ct;
  Thread *water_thread;
  Domain *water_domain;
  real xc[ND_ND];                    /* two different cell centroids
defined */
  real x[ND_ND];

  radius_limit = RP_Get_Real("rad_lim");
  volume_limit = RP_Get_Real("vol_lim");              /* get the values from
TUI*/
  drop_refin = RP_Get_Integer("drop_ref");            /* if 1 then drop will
be refined, if 0 then drop mesh will be preserved*/
  water_domain = Get_Domain(3);
  water_thread = Lookup_Thread(water_domain,zone_ID);
  Message("radius_limit is %e\n",radius_limit);
  Message("volume_limit is %e\n",volume_limit);
  Message("drop_refinement is %d\n",drop_refin);

         begin_c_loop(c, water_thread)
         {
```

```
           C_UDMI(c,water_thread,0) = 0;
         }
         end_c_loop(c, water_thread)

           begin_c_loop(c, water_thread)
         {
             if ((C_VOF(c,water_thread) > 0.1) && (C_VOF(c,water_thread) <
0.9))
                {
                   C_CENTROID(xc,c,water_thread);
                   begin_c_loop(ct, water_thread)
                   {
                    C_CENTROID(x,ct,water_thread);
                    l = sqrt(pow((xc[0]-x[0]),2) + pow((xc[1]-x[1]),2));
                    if ((l < radius_limit) || ((C_VOF(ct,water_thread) >
0.5) && (drop_refin == 1)))
                       {
                        C_UDMI(ct,water_thread,0) = 1;    /* will be
refined */

                       }
                              else if (((l >= radius_limit) &&
(C_VOLUME_2D(ct,water_thread) < volume_limit) && (C_VOF(ct,water_thread) <=
0.5) && (C_UDMI(ct,water_thread,0) != 1)) || ((l >= radius_limit) &&
(C_VOLUME_2D(ct,water_thread) < volume_limit) && (C_VOF(ct,water_thread) >
0.5) && (drop_refin == 0) && (C_UDMI(ct,water_thread,0) != 1)))
                       {
                        C_UDMI(ct,water_thread,0) = 2;    /* will be
coarsed */
                                 }
                   }
                   end_c_loop(ct, water_thread)
                }
           }
         end_c_loop(c, water_thread)
                                     }
```

APPENDIX VI

UDF that adjusts surface tension dependent on droplet velocity

The UDF adjusts the surface tension. It is zero at the beginning of calculation. Then it begins linearly rising up to the value of 0.072. The surface tension is linearly dependent on the droplet velocity and is given by the following formula:

$$\sigma = const \cdot U \,,$$

where const is the appropriate constant and U is the droplet velocity.

```
#include "udf.h"

DEFINE_EXECUTE_AT_END(calc_vel)

{
  real vel, mom = 0.0, vol = 0.0, sigma;
  cell_t c;
  int zone_ID = 2;
  Thread *water_thread;
  Domain *water_domain;
  water_domain = Get_Domain(3);
  water_thread = Lookup_Thread(water_domain,zone_ID);
  sigma = RP_Get_Real("sigma");

  begin_c_loop(c, water_thread)
  {
    if (C_VOF(c,water_thread) > 0.95)
    {
     mom                                                                  +=
C_VOF(c,water_thread)*C_VOLUME(c,water_thread)*C_U(c,water_thread);
     vol += C_VOF(c,water_thread)*C_VOLUME(c,water_thread);
    }
  }
  end_c_loop(c, water_thread)

  vel = mom/vol;
  RP_Set_Real("vel", vel);     /* vel must be defined as variable in TUI */
  Message("droplet velocity is %f \n", vel);
  Message("current surface tension is is %f \n", sigma);
}


DEFINE_PROPERTY(sur_ten,c,t)

{
real sigma, vel;
vel = RP_Get_Real("vel");

sigma = 0.001*vel;
RP_Set_Real("sigma", sigma);

return sigma;
}
```

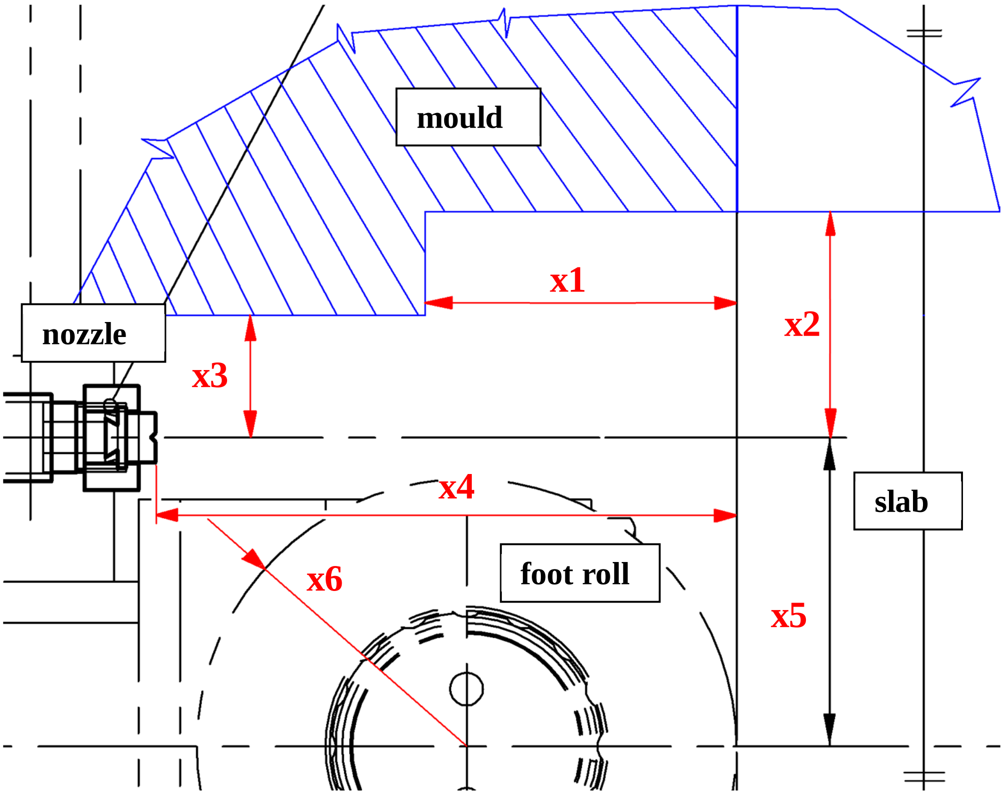
APPENDIX VII

Detail of both, mold bottom and foot roll gap

Firstly, a brief description of the continuous caster is given in here. Secondly, the crucial area is depicted in figure. For the whole continuous caster, the secondary cooling configuration is divided into separated loops. The flow rate per each loop can vary. For the casting speed of 5m/min, the flow rate per foot roll loop should be equally distributed into 2 headers of the same geometry, with the same number of nozzles to provide a uniform spraying for both sides of a slab. However, the flow rate per nozzle is not constant because of the non-symmetric header geometry, different local pressure losses coefficients per each nozzle and even nozzle clogging and its wear. For all that, the flow rate per nozzle was considered to be constant.

The drawing of the side view of the zone of interest is shown below. There is the mold bottom on the top, the flat jet with the certain angle offset on the left, the slab on the right and the foot roll at the bottom. The front view is not present; however, there is the horizontal row of the flat jet nozzles with a uniform spacing.

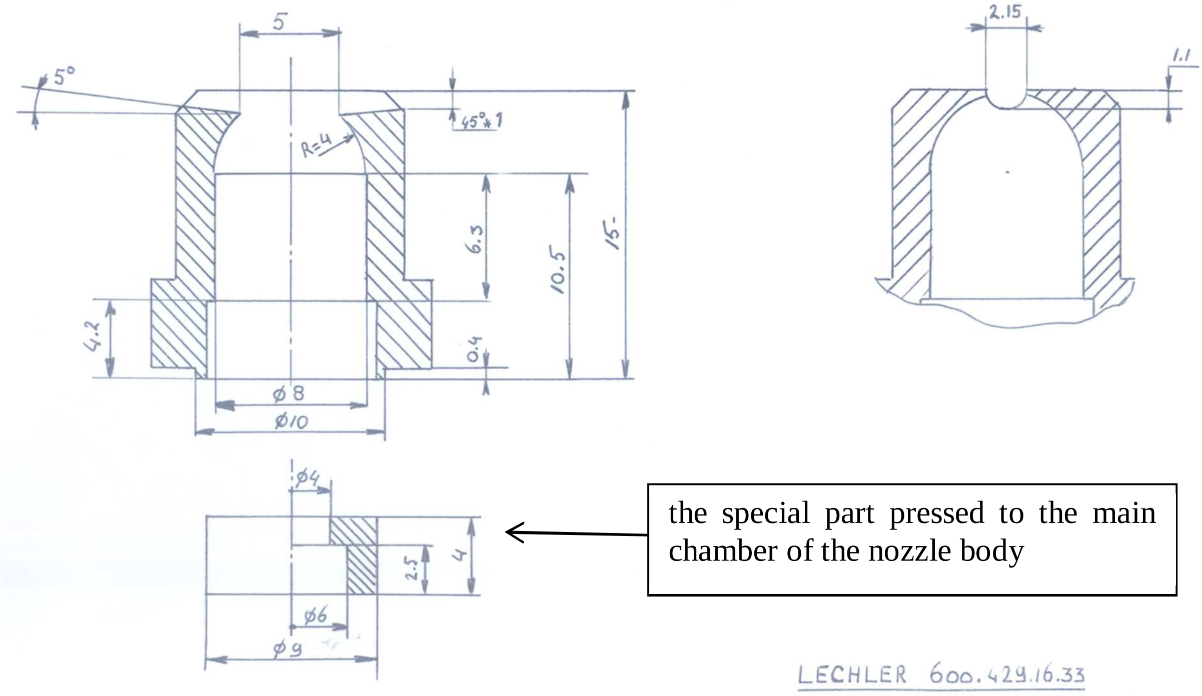
Unfortunately, it must be noted that on a company's request, the dimensions could not be disclosed.



APPENDIX VIII

Geometry of flat jet

The nozzle compounds of two parts, which are the main body and the special insert pressed to the main chamber of the nozzle body.



the special part pressed to the main chamber of the nozzle body

APPENDIX IX

Blob jet model implemented via UDF into Euler-Euler model

Two UDFs were employed to implement the Blob jet model into the homogeneous Euler-Euler model. The first UDF calculated the source term attended in scalar equation that propagates the droplet diameter. Firstly, relative velocity between both of phases, dimensionless numbers, and flux were computed within each cell. Then, the source term was assigned with respect to the droplet diameter in a previous time step. As regards the second UDF, the droplet diameter was updated considering both, the smallest allowable droplet diameter and the volume fraction, as a threshold limit.

```
/* source term for blob jet model, WAVE model */

#include "udf.h"
#define SIGMA 0.0723              /*surface tension*/
#define VISCOSITY 0.000001002     /*kinematic viscosity of water*/
#define RO_AIR 1.225
#define RO_DROP 998.0
#define B1 0.1                    /* for breakup time calculation, higher = 
higher breakup time */
#define B0 0.01                 /* for diameter calculation, higher = higher 
diameter */

DEFINE_SOURCE( diameter_source, c, t, dS, eqn)
{
#if !RP_HOST

face_t f;
Thread *tf;
real Q = 0.0;
Domain *water_domain;
Domain *air_domain;
real  timestep,loss,U_drop[3],U_air[3],U_rel,a,  Omega,  Tau,  Lambda,  Z,  T, 
We_1, We_2, Re_1, vol, source;
int zone_id = 2;
water_domain = Get_Domain(3);
Thread *water_thread = Lookup_Thread(water_domain,zone_id);
air_domain = Get_Domain(2);
Thread *air_thread = Lookup_Thread(air_domain,zone_id);
real x[ND_ND];
int n;

timestep = CURRENT_TIMESTEP;
C_CENTROID(x,c,t);
U_drop[0] = C_U(c,water_thread);           /* component of velocity of 
droplet and air in each cell */
U_drop[1] = C_V(c,water_thread);
U_drop[2] = C_W(c,water_thread);
U_air[0] = C_U(c,air_thread);
U_air[1] = C_V(c,air_thread);
U_air[2] = C_W(c,air_thread);

/*if  different  velocities,  calculates  U_rel,  else  hardcode  U_rel  to 
0.000001 */

if (NV_MAG(U_drop) != NV_MAG(U_air))
      {
       U_rel = NV_MAG(U_drop)-NV_MAG(U_air);
      }
      else
      {
       U_rel = .000001;
```

```
      }

c_face_loop(c, t, n)                       /* loops over all faces of a cell
*/
      {
       f = C_FACE(c,t,n);                  /* identify global face index */
       tf = C_FACE_THREAD(c,t,n);
            if (F_FLUX(f,tf) < 0)
                  {
                  Q += F_FLUX(f,tf);
                  }
      }

a = C_UDSI_M1(c,t,0)/2;
Re_1 = fabs(U_rel)*a/VISCOSITY;           /* Re_drop */
We_2 = RO_AIR*pow(U_rel,2.)*a/SIGMA;      /* We_air */
We_1 = RO_DROP*pow(U_rel,2.)*a/SIGMA;     /* We_drop */
Z = pow(We_1,.5)/Re_1;                    /* Ohnesorge number Z */
T = Z*pow(We_2,.5);                       /* T number */
Lambda = a*9.02*(1.+.45*pow(Z,.5))*(1.+.4*pow(T,.7))/
pow((1.+.87*pow(We_2,1.67)),.6);          /* maximum growrate Lambda */
Omega = (.34+.38*pow(We_2,1.5))*pow((RO_DROP*pow(a,3)/SIGMA),-.5)/((1.+Z)*
(1.+1.4*pow(T,.6)));                      /* wavelength Omega */
Tau = 3.726*B1*a/(Omega*Lambda);
loss = 2.* timestep*(-(a-Lambda*a)/Tau);
vol = C_VOLUME(c,t);

if (loss < 0 && C_UDSI_M1(c,t,0) > 0.0002)
      {
      source = -Q*loss/vol;
      dS[eqn] = 0.0;
      }
else if (C_UDSI_M1(c,t,0) < 0.0002)
      {
      source = 0.0;
      dS[eqn] = 0.0;
      }
else
      {
      source = 0.0;
      dS[eqn] = 0.0;
      }
C_UDMI(c,t,0) = source;                   /* save source to UDM */
return source;
#endif
}


/* definition of droplet diameter based on previous source term */

#include "udf.h"
DEFINE_PROPERTY(diameter,c,t)
{
#if !RP_HOST

real d,d_min,x[ND_ND],d_pot,vof;
int n = 0;
real sum = 0.0;
d_pot = C_UDSI(c,t,0);
vof = C_VOF(c,t);
```

```
begin_c_loop(c,t)
      {
      C_CENTROID(x,c,t);
      if (x[0]<0.0147 && x[0]>0.0073 && x[1]>0.0024 && x[1]<0.0035)
            {
             sum += C_UDSI(c,t,0);
             n += 1;
            }
      }
end_c_loop(c,t)

d_min = sum/n;

if (d_pot > d_min && vof > 0.01)
      {
      d = d_pot;
      }
else
      {
      d = d_min;
      }
return d;
#endif
}
```

APPENDIX X

Coupling between discrete phase model and volume of fluid model

This UDF provides the coupling between discrete phase model and the volume of fluid model. Firstly, the gap between the foot roll and the slab was filled with the water in order to ensure good guess of initial conditions. In other words, there was already the water level at the beginning of simulation. DPM variables namely the flow rate, the velocity were stored in user defined memory every time step and afterwards, DPM particles i.e. droplets were converted into the volume fraction of water via mass source and source of momentum. Finally, DPM particle trajectories were terminated.

```
#include "udf.h"
#include "dpm.h"
#include "surf.h"
#define mu_a 1.789e-05
#define mu_w 0.001003
#define ro_a 1.225
#define ro_w 998
#define jet_Q 0.000279      /* nozzle flow rate [m3/s] */

double inflow;
double vof_tot;
double time_sum;
double outflow;
double outflow_sum;

real drag_coeff(real Re)
{
real Cd;
         Cd = pow(Re,-0.9116) * exp(0.04833 * log10(Re)*log10(Re) + 3.2983);
return Cd;
}

/***************************************************************/
/* UDF that finds the first layer of cells on a impact wall */
/***************************************************************/

DEFINE_ON_DEMAND(impact_cells)
{
int cut_wall_ID,mould_wall_ID,symmetry_wall_ID , zone_ID,n;
cell_t c;
Domain *d;
Thread *t, *tf;
cut_wall_ID = 5;    /* tiny wall creating by cutting edge between roll and slab */
mould_wall_ID = 3;    /* bottom of mould */
symmetry_wall_ID = 6; /* side walls */
zone_ID = 2;
d = Get_Domain(1);
t = Lookup_Thread(d,zone_ID);

        begin_c_loop(c, t)
        {
                c_face_loop(c, t, n)
                {
                tf = C_FACE_THREAD(c,t,n);
                        if ((THREAD_ID(tf) == cut_wall_ID) || (THREAD_ID(tf) == mould_wall_ID) ||
(THREAD_ID(tf) == symmetry_wall_ID))
                        {
                        C_UDMI(c,t,0) = 1.;
                        }
                }
```

```
            }
            end_c_loop(c, t)
}

/****************************************/
/********  reset UDM memory  *********/
/****************************************/

DEFINE_ON_DEMAND(reset_UDM)
{
  Domain *d;
  Thread *t;
  cell_t c;
  d = Get_Domain(1);
  t = Lookup_Thread(d,2);

            begin_c_loop(c, t)
            {
            C_UDMI(c,t,0) = 0.;
    C_UDMI(c,t,1) = 0.;
            C_UDMI(c,t,2) = 0.;
            }
            end_c_loop(c, t)
}

/************************************************************************************************/
/* UDF for computing drag coefficient considering volume fraction in computational cell */
/************************************************************************************************/

DEFINE_DPM_DRAG(droplet_drag_force,Re,p)
{
  int zone_id;
  real drag_force, Cd, mu_m, ro_m, Re_a;
  cell_t c;
  Thread *t;
  Domain *water_domain;
  Thread *water_thread;
  zone_id = 2;
  water_domain = Get_Domain(3);
  water_thread = Lookup_Thread(water_domain,zone_id);
  c = P_CELL(p);
  t = P_CELL_THREAD(p);


  if ((C_VOF(c,water_thread) < 0.5) && (C_VOF(c,water_thread) > 0.0001))    /* lower limit 0.0001
calculated from m-file */
    {
            mu_m = mu_a * (1 - C_VOF(c,water_thread)) + mu_w * C_VOF(c,water_thread);
            ro_m = ro_a * (1 - C_VOF(c,water_thread)) + ro_w * C_VOF(c,water_thread);
    Re_a = Re * mu_m/ro_m * ro_a/ mu_a;
                        if (Re_a <= 1000)
                        {
                        Cd = drag_coeff(Re_a);
                        }
                        else
                        {
                        Cd = 0.424;
                        }
            drag_force = (18./24.) * Cd * Re_a * mu_a/mu_m;
            return (drag_force);
```

```
   }
  else
          {
                  if (Re <= 1000)
                  {
                  Cd = drag_coeff(Re);
                  }
                  else
                  {
                  Cd = 0.424;
                  }
          drag_force = (18./24.) * Cd * Re;
          return (drag_force);
          }
}

/******************************************/
/* DPM macro used to abort particles */
/******************************************/

DEFINE_DPM_SOURCE( name, c, t, S, strength, p)
{
real x[ND_ND];
real lftos = 0.0002;                    /* !!! set lifetime for sources [s], if particles residence time
exceeds
                                                          0.01 s, zero is assigned to particles mass */
                                        /* this precaution is only for preventing particles from
clustering */
Domain *water_domain;
Thread *water_thread;
int zone_id = 2;
water_domain = Get_Domain(3);
water_thread = Lookup_Thread(water_domain,zone_id);

if (FLUID_THREAD_P(t))
{
   P_USER_REAL(p,0) += P_DT(p);
          if ((C_VOF(c,water_thread) >= 0.5) || (P_USER_REAL(p,0) > 0.01) || (C_UDMI(c,t,0) == 1.))
/* either VOF limit or lifitime limit fulfilled */
          {
             if ((C_UDMI(c,t,2) == 0.0) && (C_UDMI(c,t,1) == 0.0) && (P_USER_REAL(p,1) ==0.0))
                                        /* no source imposed so far */
                  {
                   C_UDMI(c,t,2) = P_FLOW_RATE(p)/ (strength * C_VOLUME(c,t) * lftos);
                   C_UDMI(c,t,1) = lftos;
                   P_USER_REAL(p,1) = 1.;
                  }
                  else if ((C_UDMI(c,t,2) > 0.0) && (C_UDMI(c,t,1) < lftos) && (P_USER_REAL(p,1)
==0.0))                          /* source already exists from previous DPM iteration */
                  {
                   C_UDMI(c,t,2) = C_UDMI(c,t,2) * C_UDMI(c,t,1)/lftos;
                   C_UDMI(c,t,2) = C_UDMI(c,t,2) + (P_FLOW_RATE(p)/ (strength * C_VOLUME(c,t) *
lftos));
                   C_UDMI(c,t,1) = lftos;
                   P_USER_REAL(p,1) = 1.;
                  }
                  else if ((C_UDMI(c,t,2) > 0.0) && (C_UDMI(c,t,1) == lftos) && (P_USER_REAL(p,1)
==0.0))                          /* source term already exists from actual DPM iteration */
                  {
```

```
                    C_UDMI(c,t,2) = C_UDMI(c,t,2) + (P_FLOW_RATE(p)/ (strength * C_VOLUME(c,t) *
lftos));
                    P_USER_REAL(p,1) = 1.;
                   }
                   /*else if (P_USER_REAL(p,1) == 1)
      {
                    Message ("for this particle the source already exists\n");
                   }                      */
                   /*else
                   {
                    Message(" Error!!! Consider revising !!! \n");
                   }*/
                   P_MASS(p) = 0.;
                   p->stream_index=-1;
                   /* Message("droplet being aborted \n"); */
          }
}
}


DEFINE_EXECUTE_AT_END( clear_sources)
{
  FILE *fp;
  double flowrate, flowrate_vof;
  real time_interval;
  Domain *d;
  face_t f;
  Thread *t;
  cell_t c;
  Domain *water_domain;
  Thread *water_thread;
  Thread *inlet_thread;
  d = Get_Domain(1);
  t = Lookup_Thread(d,2);
  water_domain = Get_Domain(3);
  water_thread = Lookup_Thread(water_domain,2);
  inlet_thread = Lookup_Thread(water_domain,7);
  fp = fopen("mc_2e-4s_s4x_post.txt","a");
  time_interval = RP_Get_Real("time_interval");


    inflow = 0.0;
          outflow = 0.0;
    begin_c_loop(c, t)
          {
          inflow += C_VOLUME(c,t) * C_VOF(c,water_thread); /* calculates water volume fraction in
whole domain */
          }
          end_c_loop(c, t)


          begin_f_loop(f, inlet_thread)    /* loops over faces of inlet  */
    {
     outflow += F_FLUX(f,inlet_thread);
    }
          end_f_loop(f, inlet_thread)


          time_interval += CURRENT_TIMESTEP; /* time_interval for post processing */
          RP_Set_Real("time_interval", (time_interval));
```

```
        time_sum += CURRENT_TIMESTEP;   /* sum of timesteps  */
        outflow_sum += 0.001 * outflow * CURRENT_TIMESTEP; /* sum outflow [m3] */
        flowrate = jet_Q * time_sum - outflow_sum;   /* difference between inflow by lagrange and
outflow by VOF */
        flowrate_vof = inflow - vof_tot;
        Message("inflow calculated using VOF is %e \n", flowrate_vof);
        Message("inflow by flat jet is %e \n", flowrate);
        Message("time elapsed is %f and total volume originally occupied by water was %e \n",
time_sum, vof_tot);
        Message("outflow through opening is %e", outflow);
        fprintf(fp, "%e %e %e %e \n", time_sum, flowrate, flowrate_vof, outflow);
        fclose(fp);

        begin_c_loop(c, t)
        {
        if (C_UDMI(c,t,2) > 0.0)
        {
                if ((C_UDMI(c,t,1) - CURRENT_TIMESTEP) >= 0.)  /* actual timestep is still larger
than remaining time for particular source */
                {
                C_UDMI(c,t,1) = C_UDMI(c,t,1) - CURRENT_TIMESTEP;
                }
                else    /* last non-zero source when actual timestep is larger than remaining time
C_UDMI(c,t,1) for particular source */
                {
                C_UDMI(c,t,2) = 0.0;
                C_UDMI(c,t,1) = 0.0;
                }
        }
        }
        end_c_loop(c, t)
}

/***************************************************************/
/** UDF that calculates actual volume fraction of water **/
/***************************************************************/

DEFINE_ON_DEMAND(calculate_VOF_water)
{
  Domain *d;
  Thread *t;
  cell_t c;
  Domain *water_domain;
  Thread *water_thread;
  d = Get_Domain(1);
  t = Lookup_Thread(d,2);
  water_domain = Get_Domain(3);
  water_thread = Lookup_Thread(water_domain,2);

    vof_tot = 0.0;
        begin_c_loop(c, t)
        {
        vof_tot += C_VOLUME(c,t) * C_VOF(c,water_thread); /* calculates water volume fraction in
whole doamin */
        }
        end_c_loop(c, t)
        time_sum = 0.0; /* reset the time used for calculation of mass conservation */
        inflow = 0.0;
        Message("Total Volume occupied by water is %f \n", vof_tot);
        Message("Variable for time sum have been initialized \n");
```

```
}


/*********************************************/
/*  mass and momentum source of VOF  */
/*********************************************/

DEFINE_SOURCE(water_mass_source, c, t, dS, eqn)
{
  real source;
  source =  4.0 * C_UDMI(c,t,2);
  /* if (source > 0.)
  {
  Message("source is %f \n", source);
  } */
  dS[eqn] =  0.;
  return source;
}
```

APPENDIX XI

Material properties for heat transfer calculations with DPM

Material properties

	mixture	species		water droplet	steel
		air	vapor		
density [kg/m3]				998	8030
specific heat cp [J/kg-K]	mixing law	1006	2014	4182	502.48
thermal conductivity [W/m-K]	0.0454			0.6	50
viscosity [kg/m-s]	piecewise linear			piecewise linear	
mass diffusivity [m2/s]	kinetic theory				
Thermal diffusion coefficient	kinetic theory				
molecular weight [kg/kg-mol]		28.966	18.01		
reference temperature [°C]		25	25		
latent heat [J/kg]				2263073	
vaporization temperature [°C]				10.85	
boiling point [°C]				99.85	
volatile component fraction [%]				100	
binary diffusivity [m2/s]					
saturation vapor pressure [Pa]				piecewise linear	
droplet surface tension [N/m]				piecewise linear	

Piecewise linear saturated vapor pressure

teplota	**saturated vapor pressure**
[°C]	[Pa]
0	611
50	12288
60	19844
70	31063
80	42273
90	70122
100	101613
120	200501
150	487270
200	1633225
300	9343954
400	30934826
500	74030116
1000	700919169

APPENDIX XII

UDF for film boiling based on heat flux through interface

This UDF identifies the first row of cells on the wall of jet impact. Afterwards, it calculates heat flux through interface if the cell is an interface cell. The heat flux is than used for calculation of vapor mass source. If the cell lies in the first row of cells on the impact wall, than the heat flux from the wall is used for calculation of mass source of vapor.

```
#include "udf.h"
#include "mem.h"
#define T_SAT 373.15
#define T_WALL 400
#define L 2270000
#define zone_ID 2
#define delta 0.002  /* cell height */

/***********************************************************/
/*          UDF for detecting impact wall cells            */
/***********************************************************/

DEFINE_ON_DEMAND(cells_on_wall)
{
#if !RP_HOST
int n = 0;
cell_t c;
Domain *d;
Thread *t, *tf;
d = Get_Domain(3);
t = Lookup_Thread(d,zone_ID);

begin_c_loop_int(c,t)
{
  c_face_loop(c, t, n)
  {
  tf = C_FACE_THREAD(c,t,n);
  if (THREAD_ID(tf) == 7)              /* thread impact wall */
  {C_UDMI(c,t,0) = 1;}
  }
}
end_c_loop_int(c,t)
#endif
}

/***********************************************************/
/*      UDF for calculating HF through interface           */
/***********************************************************/

DEFINE_ADJUST(show_heat_flux, domain)
{
#if !RP_HOST
double k_vapour = 0.0261, k_water = 0.6, conductivity, x[ND_ND], grad, A, A1, A2, h;
int n = 0;
cell_t c;
Domain *d;
Thread *t;
d = Get_Domain(3);
t = Lookup_Thread(d,zone_ID);

begin_c_loop_int(c,t)
{
C_CENTROID(x,c,t);
conductivity = C_VOF(c,t)*k_vapour + (1-C_VOF(c,t))*k_water;
```

```
        if ((C_UDMI(c,t,0) == 0) && (NV_MAG(C_VOF_RG(c,t)) > 0.0) && (C_VOF(c,t) > 0.01) &&
(C_VOF(c,t) < 0.99) && C_T(c,t) > (T_SAT + 1.0))   /* Vypar */
        {
         grad = (C_VOF_RG(c,t) [0] * C_T_RG(c,t) [0] + C_VOF_RG(c,t) [1] * C_T_RG(c,t) [1])/
NV_MAG(C_VOF_RG(c,t));
         A1 = 0.5 * (SQR(x[1] + delta/2) - SQR(x[1] - delta/2));
         h = C_VOLUME(c,t)/A1;
         A2 = x[1] * h;                                  /*  2pi *r * h/ (2pi)       */
         A = (A1 + A2)/2;
         C_UDMI(c,t,1) = A * conductivity * fabs(grad)/ (L*C_VOLUME(c,t));             /* corresponds to
S*k*dT/dn / (L* C_VOLUME) */
        }
        else if ((C_UDMI(c,t,0) == 0) && (NV_MAG(C_VOF_RG(c,t)) > 0.0) && (C_VOF(c,t) > 0.0) &&
C_T(c,t) < (T_SAT - 1.0))     /* condensation */
        {
         grad = (C_VOF_RG(c,t) [0] * C_T_RG(c,t) [0] + C_VOF_RG(c,t) [1] * C_T_RG(c,t) [1])/
NV_MAG(C_VOF_RG(c,t));
         A1 = 0.5 * (SQR(x[1] + delta/2) - SQR(x[1] - delta/2));
         h = C_VOLUME(c,t)/A1;
         A2 = x[1] * h;                                           /*  2pi *r * h/ (2pi)       */
         A = (A1 + A2)/2;
         C_UDMI(c,t,1) = -1.0 * A * conductivity * fabs(grad)/ (L*C_VOLUME(c,t));                  /*
corresponds to S*k*dT/dn / (L* C_VOLUME) */
        }

        else if ((C_UDMI(c,t,0) == 1) && (C_VOF(c,t) < 0.99) && (C_T(c,t) > T_SAT))
/* vypar */
        {
         A1 = 0.5 * (SQR(x[1] + delta/2) - SQR(x[1] - delta/2));            /* area in normal direction */
         h = C_VOLUME(c,t)/A1;
         grad = (T_WALL - C_T(c,t))/(0.5 * h);
         C_UDMI(c,t,1) = conductivity * grad/ (L*h);                        /* corresponds to S*k*dT/dn / (L*
C_VOLUME) */
        }
        else
        {
         C_UDMI(c,t,1) = 0.0;
        }
}
end_c_loop_int(c,t)
#endif
}
DEFINE_SOURCE(vapour, c, t, dS, eqn)
{
double source = 0.0;
source = C_UDMI(c,t,1);
dS[eqn] = 0.0;
return(source);
}
DEFINE_SOURCE(water, c, t, dS, eqn)
{
double source = 0.0;
source = -1.0 * C_UDMI(c,t,1);
dS[eqn] = 0.0;
return(source);
}
DEFINE_SOURCE(energy, c, t, dS, eqn)
{
double source = 0.0;
```

```
source = -L * C_UDMI(c,t,1);
dS[eqn] = 0.0;
return(source);
}
```

APPENDIX XIII

UDF that calculates mean jet velocity of solid jet represented by Lagrange particles

The UDF calculates the jet velocity in each computational cell as a mass-weighted velocity of particles located in this cell

```
#include "udf.h"

DEFINE_ON_DEMAND(cells_on_wall)
{
#if !RP_HOST
int n = 0;
cell_t c;
Domain *d;
Thread *t, *tf;
d = Get_Domain(1);
t = Lookup_Thread(d,2);
begin_c_loop(c,t)
{
  C_UDMI(c,t,0) = 0.;
  c_face_loop(c, t, n)
  {
  tf = C_FACE_THREAD(c,t,n);
  if (THREAD_ID(tf) == 6)               /* thread of impact wall */
  {C_UDMI(c,t,0) = 1.;}
  }
}
end_c_loop(c,t)
#endif
}
DEFINE_ON_DEMAND(calculate_flowrate)
{
real ind[2];
int i;
cell_t c;
Thread *t;
Domain *d;
Particle *p;
d = Get_Domain(1);
t = Lookup_Thread(d,2);
Alloc_Storage_Vars(d, SV_DPM_PARTICLE_BIN, SV_NULL);
bin_particles_in_cells(d);


begin_c_loop(c,t)
{
    C_UDMI(c,t,1) = 0.;
            memset((void*)&ind, 0.0, sizeof(real)*2);
            begin_particle_cell_loop(p,c,t)
            {
                        ind[0] += NV_MAG(P_VEL(p)) * P_MASS(p);
                        ind[1] += P_MASS(p);
            }
            end_particle_cell_loop(p,c,t)


              if (ind[1] > 0.)
              {
              C_UDMI(c,t,1) += ind[0]/ind[1];
              }
}
end_c_loop(c,t)
```

```
Free_Storage_Vars(d, SV_DPM_PARTICLE_BIN, SV_NULL);
}
```

APPENDIX XIV

UDF that calculates mean jet velocity of solid jet represented by Lagrange particles

These concatenated UDFs first identify the first row of cells adjacent to the impact wall. Next, particles located within this row of cells are looped over and their temperature increment is calculated based on HTC, particle time step and droplet diameter. HTCs are calculated according to the Ranz and Marshall correlation as a function of Reynolds and Prandtl number.

```
/**************************************************************************************************************/
/* Concatenated UDFs for the Discrete Phase Model that includes a usage of DPM_SWITCH */
/**************************************************************************************************************/

#include "udf.h"
#include "dpm.h"

/******* reserve UDM ********/

#define NUM_UDM 1
static int udm_offset = UDM_UNRESERVED;

DEFINE_EXECUTE_ON_LOADING(on_loading, libudf)
{
  if (udm_offset == UDM_UNRESERVED) udm_offset =
            Reserve_User_Memory_Vars(NUM_UDM);

  if (udm_offset == UDM_UNRESERVED)
      Message("\nYou need to define up to %d extra UDMs in GUI and "
             "then reload current library %s\n", NUM_UDM, libudf);
  else
    {
             Message("%d UDMs have been reserved by the current "
             "library %s\n",NUM_UDM, libudf);

              Set_User_Memory_Name(udm_offset,"IS_ON_WALL");
            }
  Message("\nUDM Offset for Current Loaded Library = %d",udm_offset);
}

/******* detect impact wall cells ********/

DEFINE_ON_DEMAND(cells_on_wall)
{
#if !RP_HOST
int n = 0;
cell_t c;
Domain *d;
Thread *t, *tf;
d = Get_Domain(1);
t = Lookup_Thread(d,2);

begin_c_loop(c,t)
{
  C_UDMI(c,t,0) = 0.;
  c_face_loop(c, t, n)
  {
  tf = C_FACE_THREAD(c,t,n);
  if (THREAD_ID(tf) == 6)               /* thread impact wall */
  {C_UDMI(c,t,0) = 1.;}
  }
}
end_c_loop(c,t)
#endif
```

```
}

/******* DPM LAW for droplet heating ********/

#define RO_AIR 1.225
#define RO_WATER 998.2
#define MU_AIR 1.7894e-05
#define CP_AIR 1006.43
#define CP_WATER 4182
#define K_AIR 0.0242

DEFINE_DPM_LAW(DropHeatLaw,p,ci)
{
 real area, rel_vel, Re, Pr, HTC, delta_temp;
 cell_t c = P_CELL(p);
 Thread *t = P_CELL_THREAD(p);

  area = 4.0 * M_PI * pow(P_DIAM(p),2.0);
  rel_vel = sqrt(pow((P_VEL(p)[0] - C_U(c,t)),2.0) + pow((P_VEL(p)[1] - C_V(c,t)), 2.0));
  Re = RO_AIR * P_DIAM(p) * rel_vel /MU_AIRů                    /* Reynolds number */
  Pr = CP_AIR*MU_AIR/K_AIR;                                     /* Prandtl number */
  HTC = K_AIR * (2.0 + 0.6*pow(Re,0.5)*pow(Pr,1./3.))/ P_DIAM(p);
  delta_temp = P_DT(p) * area * HTC * (C_T(c,t)-P_T(p))/(P_MASS(p)*CP_WATER);
  P_T(p) = P_T(p) + delta_temp;
}

/******* DPM SOURCE for droplet heating ********/

DEFINE_DPM_SOURCE(dpm_source, c, t, S, strength, p)
{
  real mp;
  Material *sp = P_MATERIAL(p);

  if (P_CURRENT_LAW(p) == DPM_LAW_USER_1)
    {

            S->energy += P_MASS(p) * MATERIAL_PROP(sp,PROP_Cp) * (P_T(p) - P_T0(p));
    }
}

/******* DPM SWITCH for DPM laws switching ********/

DEFINE_DPM_SWITCH(dpm_switch,p,coupled)
{
 cell_t c = P_CELL(p);
 Thread *t = P_CELL_THREAD(p);
  P_CURRENT_LAW(p) = DPM_LAW_USER_1;
}
```