ÉCOLE CENTRALE LYON

N° d'ordre: 2014-45 Décembre 2014

THÈSE

présentée devant

L'ÉCOLE CENTRALE DE LYON
École Doctorale MEGA

en vue d'obtenir le titre de

DOCTEUR

Spécialité: Mécanique

par

Magdalena NEUHAUSER

Development of a coupled SPH-ALE/Finite Volume method for the simulation of transient flows in hydraulic machines

---

Soutenue le 18 décembre 2014 devant la Commission d'Examen

Rapportrice:	Professeur	R. FORTES-PATELLA	Université de Grenoble, France
Rapporteur:	Professeur	T. RUNG	TU Hamburg, Allemagne
Directrice de thèse:	Professeur	I. TRÉBINJAC	Ecole centrale de Lyon, France
Co-Encadrant:	Docteur	J.-C. MARONGIU	ANDRITZ HYDRO, France
Examinateur:	Docteur	S. AUBERT	Ecole centrale de Lyon, France
Examinateur:	Injénieur	L. CASTILLON	ONERA, France
Examinateur:	Professeur	M. LANCE	Ecole centrale de Lyon, France
Examinateur:	HDR	D. VIOLEAU	LNHE & EDF, France

Laboratoire de Méchanique des Fluides et d'Acoustique UMR 5509

Für Mama, Papa, Irene und Nicolas

Acknowledgement

In the past years I had the possibility to work and to discuss with many interesting people and I am grateful to everyone who contributed directly or indirectly to this thesis.

At first I would like to thank the members of the jury Stéphane Aubert, Lionel Castillon, Michel Lance, Damien Violeau and especially the "rapporteurs" Régiane Fortes-Patella and Thomas Rung for their evaluation of this thesis.

This work was a Thèse CIFRE 1120/2010 involving Ecole Centrale de Lyon, the LMFA and ANDRITZ HYDRO. Moreover, it also got financial support from the CIRT (Consortium Industrie-Recherche en Turbomachines). I would like to thank all the partners involved.

It was an honor and a pleasure to work with Francis Leboeuf who was my supervisor at the beginning of the thesis. His ideas and literature suggestions were much appreciated and I found that his enthusiasm for SPH was contagious. His death which came too early was a big shock for everyone who knew him.

I am grateful to Isabelle Trebinjac that she was disposed to become my supervisor for the remaining time of the PhD project and I would like to thank her for her support.

It was an excellent opportunity for me to work in very close collaboration with the R&D department of ANDRITZ HYDRO. Sincere thanks to Etienne Parkinson who offered this opportunity and who was a source of motivation thanks to his innovative ideas. I am equally thankful to Arno Gehrer for presenting me to Etienne and for his interest in this PhD during all these years. In particular, I would like to thank sincerely my PhD co-supervisor from ANDRITZ HYDRO Jean-Christophe Marongiu for his help during the past years. He not only taught me a lot about SPH but he was also available for discussing any aspect of my PhD subject at any moment and he was even proof reading papers or parts of the thesis late in the evening. Merci.

Collaborating with several partners offers the possibility to get to know many different people that I could consult in case of questions on the PhD subject or fluid mechanics in general. Special thanks to my colleagues from ANDRITZ Hydro, in particular to the colleagues from the fourth floor in Vevey, above all Martin but also Hélène, Steven, Dorothée, Nicolas, Florent, Johannes, Franco, Jeremy, Cyril and Pascal and to the colleagues from the office in Lyon, Gilles-Alexis, Julien and Hervé. In particular, I would like to thank Hervé for his help with C++ coding.

The beginning of the time as PhD student is an interesting but also difficult and intense phase for everyone but even more if you move to a new country and if you have to learn a new language like it was the case for me. Luckily I met friends that helped me to learn the language and to discover the region that was my new home now. In particular I would like to thank my Canadian friends, Marie-Eve and Dan, as well as Jaouida and Sanni. Grazie, Chiara for the moments we shared as temporal roommates, for the Italian cooking lessons and the long discussions on our theses and life in general.

Herzlichen Dank an meine "österreichischen" Freundinnen und Freunde und natürlich ganz besonders an meine Eltern und Irene, die mich mein ganzes Leben unterstützt haben.

Merci Nicolas pour tout.

Abstract

The increased use of intermittent forms of renewable energy like wind and solar energy produces fluctuations in the electric grid that have to be compensated. For this reason, hydraulic machines like turbines and pumps are more often operated under non-conventional operating conditions and are submitted to frequent starts and stops. This type of operating conditions has important consequences on the life cycle of the machines. It is thus of paramount importance that transient flows at off-design conditions are properly taken into account in the design phase and numerical simulation is an appropriate way to do so.

The present study aims at developing a flexible coupling method of the meshbased Finite Volume Method (FVM) and the meshless Smoothed Particle Hydrodynamics - Arbitrary Lagrange Euler (SPH-ALE) method, which can be used as a tool for the investigation of transient phenomena in hydraulic machines.

SPH-ALE is very well adapted for the simulation of highly dynamic flows with moving geometries but has difficulties to correctly represent rapidly changing gradients of the field variables. Particle refinement is difficult to implement, especially if particles are refined in an anisotropic way. FV methods are well established in CFD because of their accuracy and stability. However, they can be tedious for simulations with moving geometries and often necessitate an interface between moving and static parts of the mesh which introduces additional errors.

To overcome the shortcomings of both methods, a coupling method is developed that uses a decomposition of the computational domain into regions where the physical field variables are computed by the FV method, regions where they are computed by SPH-ALE and overlapping regions where the information is transferred from the FV domain to the SPH domain and vice versa. In the overlapping regions FV calculation points are used as neighbors for the SPH integration in space. At the boundaries of the FV mesh, velocity and pressure are interpolated from the SPH particles by means of scattered data interpolation techniques, similarly to Chimera methods for overlapping grids.

For this study, an existing SPH-ALE software of the ANDRITZ Group is used. A weakly compressible FV solver is implemented into this software that discretizes the same form of the Navier-Stokes equations than the SPH-ALE solver. Similar to the present SPH-ALE method, Riemann solvers with reconstructed states, obtained by a MUSCL scheme, are employed. Moreover, adaptations and improvements of the SPH-ALE solver itself are made, which are important for the coupling and for the simulation of internal flows in general. Thus, subsonic inlet and outlet conditions are implemented. Furthermore, a correction method of the kernel gradient is presented that ensures zeroth order consistency of the SPH-ALE approximation of the divergence of the convective fluxes. The correction improves greatly the SPH pressure field on non-uniform particle distributions.

The implemented coupled method is successfully validated by means of inviscid academic one-dimensional and two-dimensional testcases like a shock tube case, Taylor-Green vortices and the flow around a symmetric NACA airfoil with particles in Eulerian description. Furthermore,

the coupling provides a possibility to implement outlet boundary conditions to Lagrangian moving SPH particles. It is then applied to the simulation of transient flows in rotor stator systems in 2D with moving particles.

Keywords: SPH-ALE, FVM, coupling, transient flows, hydraulic turbines

Résumé

L'utilisation croissante des sources d'énergie renouvable avec une grande volatilité de production, comme l'énergie éolienne et solaire, conduit à des fluctuations dans le reseau électrique qui doivent être compensées. Pour cette raison les machines hydrauliques, turbines et pompes, sont plus souvent opérées dans les régimes de fonctionnement hors fonctionnement nominal et la fréquence des phases de démarrage et arrêt augmente. Ce type de fonctionnement peut avoir des conséquences importantes sur le cycle de vie des machines. Il est donc essentiel de prendre en compte l'écoulement dans les phases transitoires lors de la conception de la machine et la simulation numérique des écoulements est un outil adapté pour cela.

La présente étude a pour objectif de développer une méthode de couplage flexible qui combine la méthode à maillage volumes finis (VF) et la méthode sans maillage Smoothed Particle Hydrodynamics - Arbitrary Lagrange Euler (SPH-ALE). Cette méthode couplée peut être utilisée comme outil pour l'investigation des phenomènes transitoires dans les machines hydrauliques.

SPH-ALE est particulièrement bien adapté aux simulations des écoulements fortement dynamiques avec des géometries mobiles mais elle a des difficultés pour calculer des forts gradients de pression et vitesse. Un raffinement de particules est difficile à implémenter, surtout si les particules doivent être raffinées de manière anisotrope. Les méthodes volumes finis (VF) sont établies pour les simulations numériques d'écoulements grâce à leur stabilité et précision. Par contre, elles peuvent être lourdes pour les simulations avec des géometries mobiles et demandent souvent une interface entre des parties mobiles et statiques du maillage ce qui génère des erreurs supplémentaires.

Pour combiner les deux approches complémentaires, une méthode de couplage a été développée qui décompose le domaine de calcul en zones où la vitesse et la pression sont calculées par la méthode VF, en zones où elles sont obtenues par SPH-ALE et en zones de recouvrement où les informations sont transférées de la zone VF à la zone SPH et inversement. Dans les zones de recouvrement les points de calcul VF sont utilisés comme voisins pour l'intégration en espace des particules SPH. Aux limites du maillage VF la vitesse et la pression sont interpolées des particules SPH, similairement aux méthodes Chimère des maillages recouvrants.

Un logiciel SPH-ALE existant du groupe ANDRITZ est utilisé pour cette étude. Un solveur VF faiblement compressible est implémenté dans ce logiciel. Le solveur discretise la même forme des équations de Navier-Stokes que le solveur SPH-ALE. Des solveurs de Riemann avec des états reconstuits par la méthode MUSCL sont employés. En outre, le solveur SPH-ALE est amélioré et adapté aux écoulements internes. Pour cette raison des conditions à l'entrée et à la sortie du type subsonique sont implémentées. Du plus, une méthode de correction du gradient de la fonction kernel est présentée qui améliore la précision du champs de pression, notamment si les particules ne sont pas distribuées régulièrement.

La méthode couplée est validée à l'aide des cas test académiques en unidimensionnel et en bidimensionnel, comme le cas de tube à choc, les tourbillons de Taylor-Green et l'écoulement autour d'une aube symétrique du type NACA avec des particules en déscription eulerienne.

En outre, le couplage offre la possibilité d'imposer des conditions à la sortie aux particules lagrangiennes. La méthode est appliquée aux simulations d'écoulement transitoire en 2D avec des particules qui se déplacent en suivant les géometries mobiles.

Mots-clés: SPH-ALE, méthodes volumes finis, couplage, écoulements transitoires, turbines hydrauliques

Kurzfassung

Die zunehmende Verwendung volatiler Arten von erneuerbarer Energie, insbesondere Wind- und Solarenergie, führt zu Schwankungen im Stromnetz, die kompensiert werden müssen. Aus diesem Grund werden hydraulische Maschinen wie Turbinen und Pumpen immer öfters zu nicht konventienellen Betriebspunkten betrieben und die Häufigkeit von An- und Abstellsequenzen nimmt zu. Diese Betriebszustände haben einen direkten Einfluss auf den Lebenszyklus der Maschinen. Es ist daher von grosser Bedeutung, die Strömungen in den transienten Betriebszuständen, von der Designphase der Maschine an, zu berücksichtigen und numerische Simulationen sind gut dafür geeignet.

Das Ziel der vorliegenden Studie ist die Entwicklung einer Methode zur Kopplung der gitterbasierten Finiten Volumen Methode (FVM) mit der gitterlosen Methode, Smoothed Particle Hydrodynamics - Arbitrary Lagrange Euler (SPH-ALE). Die gekoppelte Methode soll ein Werkzeug zur Untersuchung transienter Strömungen in hydraulischen Maschinen darstellen.

SPH-ALE eignet sich sehr gut zur Simulation dynamischer Strömungen mit beweglichen Geometrien, hat aber Schwierigkeiten schnell wachsende Gradienten von Druck und Geschwindigkeit richtig darzustellen. Partikelverfeinerung ist schwierig zu implementieren, insbesondere wenn die Partikel anisotrop verfeinert werden sollten. FV Methoden hingegen sind, wegen ihrer Genauigkeit und Stabilität, die am häufigsten verwendeten CFD Methoden. Für Simulationen mit bewegten Geometrien können sie jedoch umständlich sein und benötigen meist ein Interface zwischen bewegten und statischen Teilen des Rechengitters, das zusätzliche Fehler erzeugt.

Um die Schwachpunkte beider Methoden zu überwinden wird eine Kopplungsstrategie entwickelt, die das Rechengebiet in unterschiedliche Regionen teilt. Nämlich in Regionen, in denen Geschwindigkeit und Druck mit der FV Methode berechnet, Regionen, wo sie von SPH-ALE berechnet und überlappende Regionen, wo die Informationen von dem FV Gebiet ins SPH-ALE Gebiet transferiert werden und umgekehrt. In den überlappenden Regionen werden die FV Berechnungspunkte als Nachbarn für die räumliche Integration der SPH Partikel verwendet. Am Rand des FV Gebietes werden Geschwindigkeit und Druck von den SPH Partikel interpoliert, ähnlich wie es bei Chimera Methoden überlappender Gitter der Fall ist.

Für die vorliegende Studie wird eine existierende SPH-ALE Software der ANDRITZ Gruppe verwendet. In diese wird ein schwach kompressibler FV Löser implementiert, der die selbe Form der Navier-Stokes Gleichungen diskretisiert wie der verwendete SPH-ALE Löser. Ähnlich zur vorliegenden SPH-ALE Methode werden Riemann Löser mit rekonstruierten Anfangszuständen benützt (MUSCL Methode). Des Weiteren wird der SPH-ALE Löser selbst adaptiert und verbessert, damit er für die Kopplung und für interne Strömungssimulationen im Allgemeinen verwendet werden kann. Aus diesem Grund werden Eintritts- und Austrittsrandbedingungen für Unterschallströmungen in die Software hinzugefügt. Darüber hinaus wurde eine Korrektur des Gradienten der SPH Kernel Funktion präsentiert. Die Korrektur verbessert das SPH Druckfeld stark, vor allem wenn die Partikel nicht gleichmässig im Rechengebiet verteilt sind.

Die gekoppelte Methode wird anhand von ein- und zweidimensionalen akademischen Testfällen erfolgreich validiert. Dafür wird unter anderem der eindimensionale Stossrohr-Testfall, der Taylor-Green Wirbel und die Strömung um einen NACA Tragflügel mit Partikel in Eulerscher Beschreibung gerechnet. Ausserdem bietet die Kopplung die Möglichkeit, Austrittrandbedingungen für Lagrangesche Partikel zu implementieren. Die Methode wird zum Schluss für die Simulation von zweidimensionalen Rotor-Stator Systemen mit Partikel, die der Bewegung der Geometrien folgen, verwendet.

Stichwörter: SPH-ALE, FVM, gekoppelte Methode, transiente Strömungen, Wasserturbinen

Contents

Notations

Greek symbols

α	limiter function for the MUSCL scheme
β	Shepard coefficient
ϕ	field variable (pressure, velocity or density)
λ	eigenvalue
μ	dynamic viscosity
ν	kinematic viscosity
θ	non-normalized SPH kernel function
ϑ	parameter for FV MUSCL scheme
ρ	density
σ	Shepard coefficient without self-contribution
ω	particle volume, integration weight
ω^{∂}	area of the surface element, integration weight for the discretized surface integral
$\bar{\omega}$	modified weight
Δx	spatial discretization size
Δt	time discretization size
Ω	computational domain
Ω_{FV}	Finite Volume domain
Ω_{SPH}	SPH domain
$\partial\Omega$	boundary of the computational domain
$\boldsymbol{\Phi}$	vector of conservative variables, state variables
$\boldsymbol{\Phi}^E$	solution vector of Riemann problem
$\boldsymbol{\nu}$	normalized direction
$\boldsymbol{\tau}$	viscous shear stress tensor

Latin characters

c	speed of sound
c_p	pressure coefficient
d	number of space dimensions
h	smoothing length
p	pressure
p_{tot}	total pressure
p_{stag}	stagnation pressure
r	particle size
t	time
u	x-component of the velocity vector $\mathbf{v}$
v	y-component of the velocity vector $\mathbf{v}$
w	z-component of the velocity vector $\mathbf{v}$
u_0	x-component of the ALE velocity vector $\mathbf{v}_0$
v_0	y-component of the ALE velocity vector $\mathbf{v}_0$
w_0	z-component of the ALE velocity vector $\mathbf{v}_0$
x	space coordinate
y	space coordinate
z	space coordinate
C	chord length of NACA hydrofoil
D	SPH kernel support
E	total energy
L	reference length
T	maximum thickness of NACA hydrofoil
V	cell volume
W	SPH kernel function
$\mathbf{b}$	vector of closed box error
$\mathbf{n}$	normal vector
$\mathbf{n}_S$	normal vector times area
$\mathbf{v}$	velocity vector
$\mathbf{v}_0$	ALE particle velocity vector
$\mathbf{x}$	coordinate vector, position
$\bar{\mathbf{x}}$	modified position
$\dot{\mathbf{q}}$	heat flux vector
$\mathbf{B}$	renormalization matrix
$\mathbf{F}$	ALE flux tensor
$\mathbf{F}_c$	convective flux tensor
$\mathbf{F}_\nu$	viscous flux tensor
$\mathbf{I}$	unit tensor
$\mathbf{L}$	vector of left eigenvectors
$\mathbf{R}$	vector of right eigenvectors
$\mathbf{W}$	vector of primitive variables

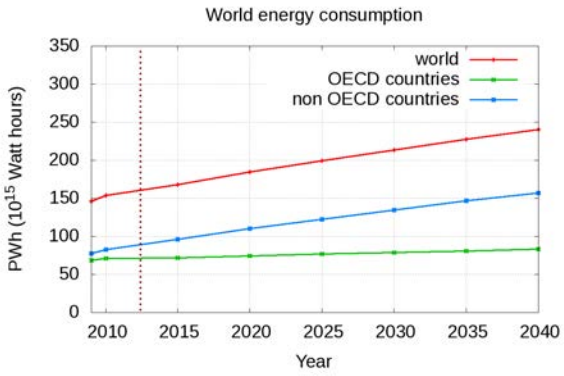
Chapter 1

Introduction

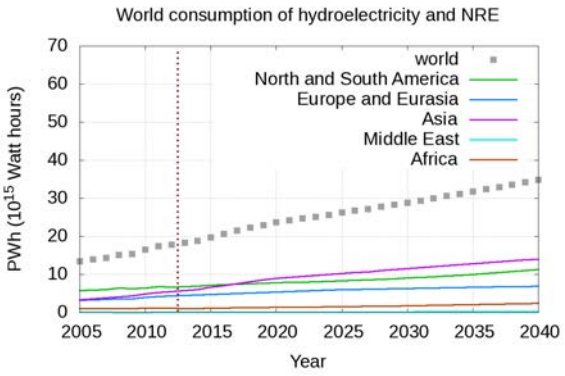
Since the industrial revolution in the 19th century, the worldwide energy consumption increases rapidly with a raise of about fifty percent just between 1990 and 2010. Forecasts published by the United States Energy Information Administration (EIA) [2] do not see a change of this trend and predict a further increase of the energy demand with an annual growth rate of 1.5%. This is mainly driven by non OECD countries, especially China and India, as it is illustrated by Figure 1.1 (a). Most of the sources that are used for energy production are fossil fuels like e.g. petroleum, gas and coal. However, burning them for electricity and heat generation is the largest source of CO_2 emission worldwide [51]. Carbon dioxide CO_2 is one of the greenhouse gases that are responsible for the global climate change that manifests itself as an increase of the globally averaged land and ocean surface temperature. The global warming has many negative impacts on human lives like for example more extreme high-temperature related events, floods and droughts or the rise of the global mean sea level that threatens parts of Bangladesh and other countries that are situated only a few meters above the sea level [50]. These environmental issues together with security concerns against nuclear powerplants lead to an increasing investment in renewable energy sources. Figure 1.1 (b) shows a forecast of the worldwide use of renewable energy that is predicted with an annual growth rate of 2.5% which means that renewable energy consumption is supposed to grow faster than the total energy demand. The largest part of renewable energy for electricity generation comes from traditional hydropower but other renewable energy sources gained in importance recently. New renewable energy sources (NRE) for electricity generation are for example small hydro, wind and solar energy.

The energy policy of the European Commission in the last fifteen years shows a clear preference for renewable sources of energy with the objectives of decreasing the greenhouse emissions and gaining more independence from imported energy. In the directive 2001/77/EC and subsequent documents (2003/30/EC, 2009/28/EC) and *Renewable Energy Roadmap* published in January 2007 a share of 20% of renewable energy in the European energy consumption was defined as target for 2020. The documents define *renewable energy sources* as wind, solar, geothermal, wave, tidal, hydropower, biomass, landfill gas, sewage treatment plant gas and biogases. Other than hydropower, wind and solar are the most important but intermittent energy sources with large variations in production.

As an illustrative example we consider Germany where the national energy policy stimulated the renewable energy sector in a way that more than twenty percent of the electricity consumption is already produced from renewable energy forms. Amongst the renewables, wind and photovoltaic have an extraordinarily high proportion. In 2013 wind turbines contributed 8.4% to the total electricity production in Germany, and photovoltaic systems 5.3%, which was published by the Fraunhofer Institute for Solar Energy Systems (ISE) [33]. At the bottom of



(a) Forecast of the world energy consumption for OECD, non OECD countries and worldwide. An worldwide annual growth rate of 1.5% was assumed.



(b) Forecast of the consumption of hydroelectricity and new renewable energy (NRE) for different regions and worldwide.

Figure 1.1: Outlook of world energy consumption plotted with data from the U.S. Energy Information administration [2] in petawatt hours (PWh) with 1 PWh = 10^{15} Wh, released on the 25.7.2013.

Figure 1.2 the monthly average of the production is shown. As expected, there is more solar energy produced from photovoltaic in summer but the production of energy from wind turbines seems to be completely random. This impression is confirmed by the top of the figure where the daily production averages are plotted. The day of minimum production of wind and solar summed up was on the 16.1.2013 with only 22 GWh, while the day of maximum production took place two weeks later on the 31.1.2013 with more than 25 times more output, namely 580 GWh.

These extreme variations necessitate energy storage and balancing on an hourly as well as weekly and monthly basis. There are several possibilities for energy storage like compressed air, flywheel, gravitational potential and thermal energy storage. But at present the most widely-spread and efficient form is pumped storage hydropower with a pumping-generating cycle efficiency of about 80%. Different aspects about pumped storage hydropower and its importance in the European energy market are discussed in the proceedings of *Pumped storage Powerplants: Challenges and opportunities* [101], a conference organized by *Societé hydraulique de France* (SHF) in Lyon (France) 2011. Only some of the arguments presented there will be mentioned in the following.

Worldwide, more than 380 pumped storage powerplants (PSP) are in operation and others are under construction. Countries that are well-suited for PSP manifest a specific geographical topology with large differences in altitude, like the Alpine countries, Austria and Switzerland, and Norway, some parts of Germany, Spain, Portugal, USA and China amongst others. Pumped storage hydropower is a comparably old technology that experienced a first boom in the 1970s in Europe, where it was used to produce electricity during peak hours. This was possible because it stored the energy produced by nuclear power plants or big thermal units during off-peak hours [64]. In recent years, pumped storage powerplants have often been constructed in combination with large wind parks (China, Scotland) or parks of photovoltaic cells (Morocco). But there are countries like the above cited Germany, that already have a large percentage of intermittent energy and plan to have even more in the near future. However, the natural requirements are not

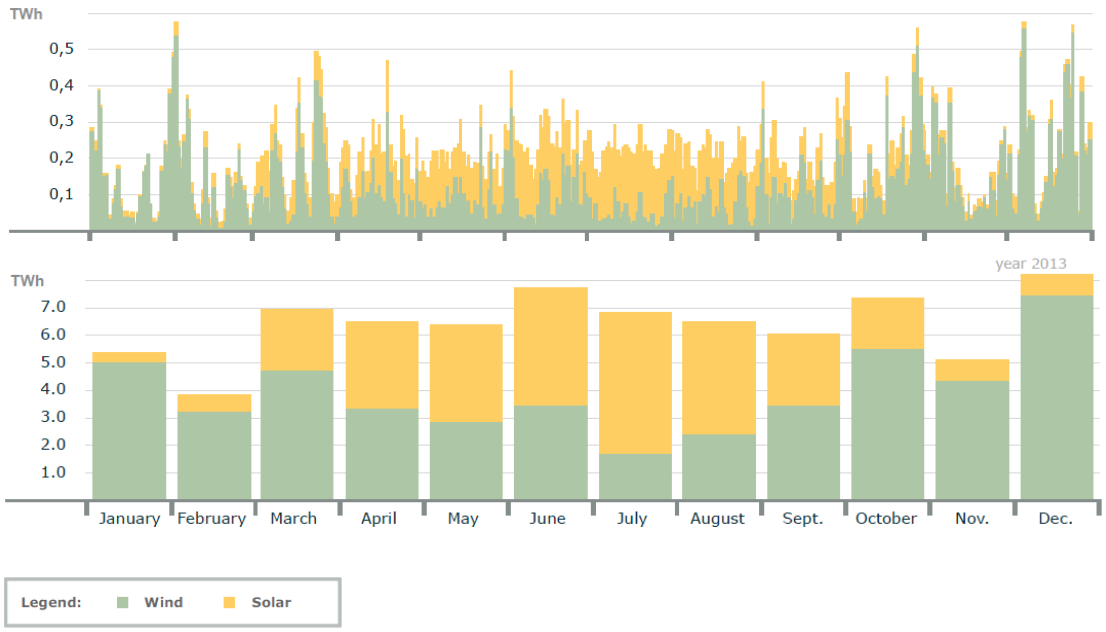


Figure 1.2: Annual energy production by solar and wind in Germany in 2013 taken from [33], in terawatt hour (1 TWh = 10^{12} Wh). At the top, the daily average is plotted and at the bottom, the monthly average. It can be seen that wind and solar are intermittent energy sources with large variations in the production.

optimal for pumped storage hydropower in Germany because there are not enough mountains and another way has to be found to guarantee the energy supply even during a few cloudy days without wind. In this case one option is the import and export of electricity to and from neighbouring countries and the use of the large storage capacities of Switzerland, Austria and in particular Norway that is sometimes called the *blue battery of Europe* [34, 43].

1.1 Changing role of hydropower

Hydropower enjoys continuing popularity. In the past, this was mainly owing to the fact that hydropower is a reliable energy source. It is not only used for the base load of energy production but also as long-term storage possibilities to compensate fluctuations in the energy demand. Fluctuations in the consumption can be of different nature and time scales. For example, there are seasonal differences between winter and summer. Then there is the influence of the industry operating time and the residential demand. Nuclear powerplants and big thermal units are not flexible enough to balance these changing demands because they cannot be easily shut down and so, hydro powerplants are often used for peak-hour production. Nowadays, in addition to the variation of demand, there are extreme supply variations of wind and solar sources which can change every minute. Hydropower plants, in particular pumped storage powerplants, are used to balance these variations with important consequences on the mode of operation of hydropower plants [45, 85, 97, 100]. An overview of the different ways how hydraulic powerplants are used to stabilize the electrical grid is given in [95].

1.1.1 Classification of powerplants and turbines

Hydropower is the largest source of renewable energy worldwide [2] and is generated at least in 150 countries. This is partly due to the fact that hydropower is flexible and can be used under different geographical circumstances like low land, hilly areas or mountainous regions. A classification of hydropower plants is given by Mosonyi [79] as follows (see also [13]).

- Conventional powerplants use the energy of flowing water and of lakes fed naturally. They can be further classified as plants with storage and plants with a small or no storage. The first group, plants with storage, utilizes differences in altitude and store and accumulate water by a dam. This potential energy is transformed into kinetic energy and further into rotational energy by the rotating turbine. The combination of flow rate and difference in altitude determines the amount of power that can be produced.

- The second group, plants with a small or no storage, are so-called *run-of-the-river* powerplants where electricity is produced by the flow rate of the river. The size of the storage, called pondage, determines if the powerplant can regulate its water flow and if it can be used in answer to varying energy demands. Powerplants without pondage are directly affected by seasonal or weather-related changes in the flow rate of the river. In addition, they cannot be used to balance variations in consumption and as a consequence, they can only be used as base-load plants.

- The third category represents pumped storage powerplants (PSP) that consist of a higher and a lower elevation reservoir. Traditionally, they are composed of a water turbine with a generator and a pump with an electrical engine. In case of overcapacity in the electric grid, water is pumped from the lower to the upper reservoir. Then, if electricity is needed, water is passed through the turbine to the lower reservoir. For economical reasons, the pump and the turbine are often replaced by a *reversible pump-turbine*.

- Furthermore, there are other forms of hydro powerplants that are less common but that are increasingly developed. For example, there are tidal powerplants that use the energy of tides, marine current that use currents like e.g. the Gulf stream and small hydro powerplants for small rivers.

In the main, there are three different types of turbines that are utilized depending on the flow rate Q and the net hydraulic head H that are illustrated by Figure 1.3. For more details see e.g. [13].

Kaplan turbines are reaction turbines where the fluid changes the pressure while passing through the turbines. They are used for large flow rates and low head $H < 80m$. The biggest turbines have an output of up to 200 MW and they are typically used in run-of-the-river plants.

Pelton turbines are impulse turbines that are used for high head and low flow rates. Nozzles form jets that impinge on the Pelton buckets that transform the kinetic energy of the jet into mechanical energy. Pelton turbines can be of different size where the biggest have an output of more than 400 MW. Pelton turbines can be used in high-head PSP.

Francis turbines are reaction turbines where the flow passes through the turbine radially from the outside to the inside and leaves it axially. Francis turbines are the most widely used turbine with a broad range of application. They are used for medium head and flow rates situated inbetween the low-head Kaplan turbines and the high-head Pelton turbines. Amongst other, they are used in PSP. A Francis turbine consists of a spiral casing, a runner and a draft tube, see Figure 1.4. The spiral casing has the task of converting parts of the pressure energy into kinetic energy and of distributing the water equally for the runner inlet. It includes stay

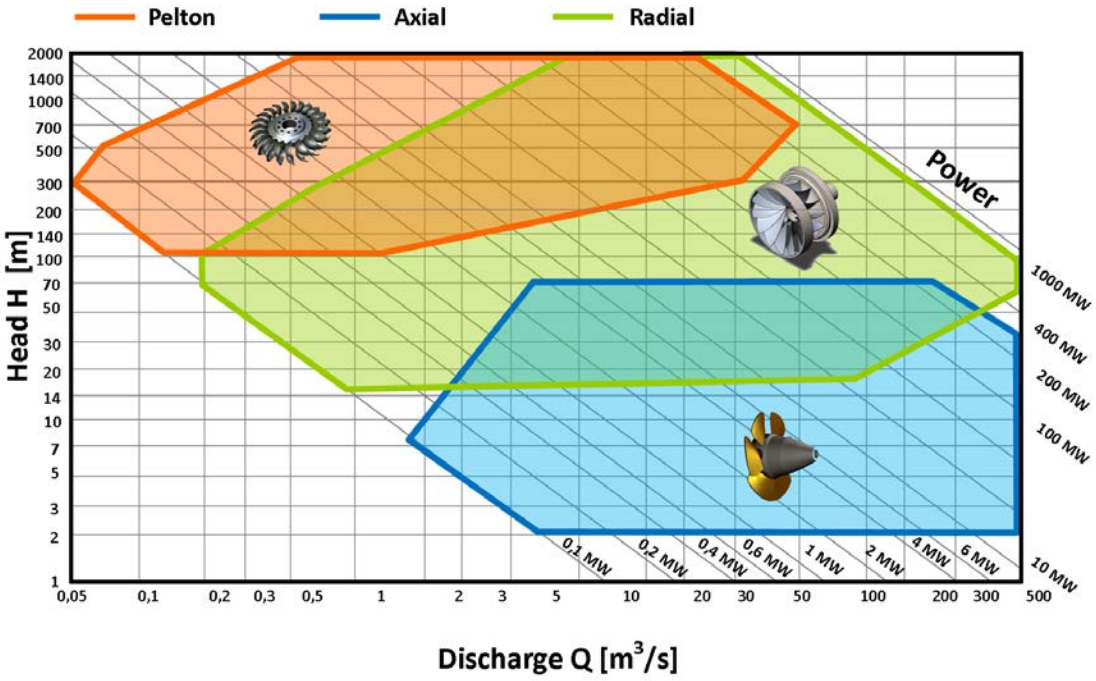


Figure 1.3: Application range of different turbine types.

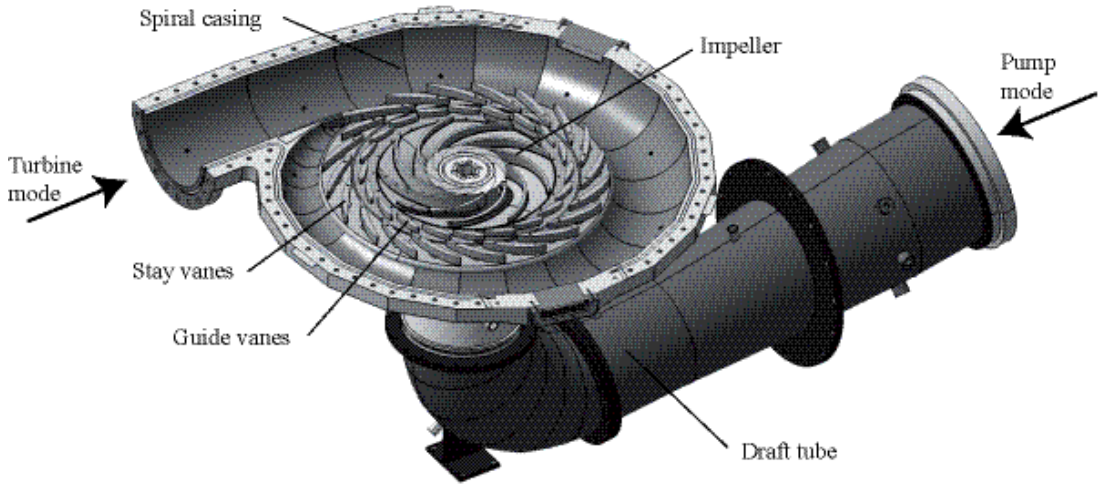


Figure 1.4: Hydrodyna pump-turbine with spiral casing, stay vanes, guide vanes, runner (impeller) and draft tube, taken from [97].

Figure 1.5: Spiral casing produced by the company Escher-Wyss (Zurich) in 1938 for the Sungari powerplant in Manchuria.

vanes and guide vanes where the variable guide vanes regulate the flow discharge. The runner is the most important part of the turbine where the conversion into mechanical energy actually takes place. The draft tube is situated at the runner outlet and recovers part of the kinetic energy.

Pump-turbines are reversible hydraulic machines, mainly of the Francis type. In turbine mode, also called generating mode, the flow passes through the spiral casing, the runner and then the draft tube. In pump mode, the flow direction is reversed, where water passes through the draft tube at first, then the runner and then the spiral casing that becomes the role of a pump diffuser, see Figure 1.4.

1.1.2 Transient operation of hydraulic machines

A pump, a turbine or a pump-turbine is designed for a *best efficiency point* (BEP) that is a fixed flow rate Q and a fixed net hydraulic head H where the highest efficiency is obtained. However, to compensate fluctuations in the electric grid, the hydraulic machines are operated more often under non-conventional conditions. They are operated under off-design conditions like part load ($Q < Q_{BEP}$) and full load condition ($Q > Q_{BEP}$) with numerous changes between the operating conditions. There are several issues connected with that type of operation that will be described in the following.

In general, the operation at off-design condition leads to high cavitation risk in pump and turbine mode [53, 80]. Cavitation is defined as the vaporization at constant temperature because of a decrease in pressure. It is not only a known source of instabilities and but can damage the turbine by erosion of cavitation [80].

Furthermore, centrifugal pumps or pump-turbines in pump mode can show instabilities at part load conditions (low flow rates) that lead to oscillations of the hydraulic system. These instabilities manifest themselves as a positive slope in the $H(Q)$ characteristic, where the head decreases when the flow rate is reduced. They are also called "S-shaped" because of the form

of the characteristic. In addition, this can be linked to a hysteresis, where the onset of the instability occurs at different Q depending on the direction of the change of flow rate. The reason for that is the appearance of recirculation zones in the pump diffuser that are called *alternate stall* if they are fixed in time or *rotating stall* if they propagate from one guide vane channel to the other with a speed that is observably smaller than the rotation speed of the runner [16, 85].

Moreover, the machines are submitted to frequent starts and stops, or in the case of pump-turbines, they are also submitted to a succession of switches from pump mode to generating mode and vice versa. For the start-up of Francis turbines or pump-turbines the guide vanes are opened and the runner is accelerated from rest to *speed no-load* where it is synchronized to the electrical grid. *Speed no-load* is the operating point where the runner is rotating at nominal speed but with small guide vanes opening and a minimal discharge. The whole energy of the incoming flow is dissipated by losses and the generator has no electric load but the unit is ready to be connected to the grid [60]. Then the opening of the guide vanes is continued until the desired operating point is reached while keeping the runner speed constant [117, 81]. Due to the fact that hydraulic machines are more often operated under this condition it is important to evaluate the influence of the transient start-up procedure on the life-time of the turbine.

In general, more care is taken on the design of a pump-turbine in pump mode than in turbine mode because a good pump performance is more difficult to obtain. As a consequence, the runner of the pump-turbine is more similar to a pump impeller than a Francis runner, and instabilities may arise in turbine mode that do not occur for Francis turbines. It is reported that in the start-up procedure of pump-turbines another instability can occur that is not observed for Francis turbines. Oscillations arise near no-load and turbine brake operation that makes it slower to synchronize the pump-turbine with the electrical grid. The observed instabilities are similar to the above described part-load instabilities in pump mode and consist of either stationary vortex formations or rotating stall cells [25, 116, 117, 45].

Another operating point that is related to *speed no-load* but that is caused by a completely different mechanism is the *runaway* point, that is reached after load rejection and with problems of the guide vanes. Again there is zero torque on the shaft but the discharge is very high, i.e. the turbine is at full load, and the runner is accelerated to a very high rotating speed. This is a quite damaging event for the unit but it occurs rarely [60, 81].

The flow in a rotating machine exhibits pressure fluctuations that are caused by the interaction of rotating and stationary parts of the machine, regardless of the considered operating point. The so-called *rotor-stator interactions* (RSI) are the combination of, e.g. in generating mode, the rotating potential flow perturbations of the runner travelling upstream and downstream and the flow perturbations of the wakes of the guide vanes. Amongst others, the RSI depend on the operating point and the size of the gap between rotor and stator. RSI are the source of other effects like guide vane vibrations, impeller vibrations and shaft vibrations [97] and they can induce resonance phenomena with the impeller or even the power house [80], yielding strong vibrations and noise.

The above described type of operating conditions will arise even more frequently in the upcoming years but has not been well analyzed yet. It can lead to strong structural vibrations and is very demanding for the installation, which has a direct consequence on the life-cycle of the machine. In extreme cases, hydrodynamic instabilities can lead to machine failure with consequences on outage of production and safety issues. It is thus of paramount importance that these transient flows are properly taken into account in the design phase. Numerical simulation, together with model tests, is an appropriate way to do so. Unfortunately, classical numerical approaches like mesh based Finite Volume (FV) methods have difficulties to capture and to

predict these phenomena correctly.

1.2 Computational fluid dynamics for transient flows

Computational fluid dynamics (CFD) is a powerful tool that helps to understand and predict flow phenomena in hydraulic machines. Numerical simulations are used throughout the design process of pumps and turbines, complementary to model tests in hydraulic laboratories. In general, it yields reliable results for normal operating points that are close to the best efficiency point but it has difficulties in providing quantitatively correct results at off-design operating points.

Many related simulations found in literature have been performed by the commercial software ANSYS CFX. This is an incompressible meshbased coupled multigrid solver based on a hybrid Finite Element (FE) - Finite Volume (FV) method that uses a high-resolution scheme for the advective fluxes, i.e. a scheme that is between first and second order, mostly second order depending on the flow characteristics. An implicit second-order time integration scheme is used for transient simulations. For turbulence modelling, several turbulence models (SST, k-ε, k-ω) for the Reynolds Averaged Navier-Stokes (RANS) equations and automatic wall functions are implemented as well as large eddy simulation (LES), detached eddy simulation (DES) and the novel scale-adaptive simulation (SAS) model.

In the normal stable operating range steady-state RANS simulations often yield sufficiently accurate results for design purpose. For these simulations only parts of the turbine are simulated and circumferential periodicity conditions are applied. In [53] for example, in turbine mode, a guide vane channel, a runner channel and the draft tube are simulated; in pump mode, the draft tube can be neglected. The stationary components are linked by a mesh interface to the rotor. The mesh interface is a so-called stage interface, also called mixing plane, where the flow variables are circumferentially averaged. This method is not able to capture rotor-stator interactions like for example the effect of the wakes of the guide vanes on the runner in turbine mode. Nevertheless, it yields good results close to the best efficiency operating point and captures the most important flow features correctly [53]. Certainly, unsteady flows at off-design conditions cannot be simulated by steady-state simulations and a transient solver has to be applied. But even transient simulations with a meshbased solver necessitate rotor-stator interfaces between the rotating and the stationary parts of the computational domain. At this mesh interface, which is called general grid interface (GGI), the flow variables often have to be interpolated by the solver in order to be transferred to the other side of the interface. This introduces local interpolation errors at the interface. Furthermore, geometries that are moving arbitrarily can only be simulated if the mesh is deformed or if it is regenerated regularly.

1.2.1 Transient simulations of off-design operating points

Several research projects can be found in literature that have been launched with the aim of analyzing the above described off-design conditions by means of measurements and/or numerical simulations. The HYDRODYNA research project is a cooperation of the Federal Institute of Technology in Lausanne (EPFL) in Switzerland, the Universitat Politecnica de Catalunya in Barcelona (UPC) in Spain and the industrial partners ALSTOM HYDRO, VOITH HYDRO and ANDRITZ HYDRO. The objective is the investigation of the hydrodynamics of reversible pump-turbines. In the frame of this project and a follow-up project, several PhD thesis were published where numerical simulations at off-design conditions have been conducted by the following authors. Braun [16] examined the part-load flow in radial pumps by numerical simula-

tions and experiments. Rotating stall was observed in experiments and could be reproduced by time-dependent RANS simulations launched with ANSYS CFX but the deviation from measurements was considerable. Flow pattern similar to the observed ones were simulated at a sensibly higher flow rate than in the experiment and even there they were quantitatively different from experiments. Hasmatuchi [45] investigated off-design conditions in turbine mode experimentally and numerically. He also used ANSYS CFX but with the SAS turbulence model on 10 million nodes and obtained good qualitative agreement but again the results were not quantitatively accurate enough. Recently, Pacot [85] simulated rotating stall at part-load condition for the first time by a Large Eddy Simulation technique and obtained good agreement of the simulated rotating stall compared to measurement data. An open source second-order Finite Element LES solver called FrontFlow/Blue of the University of Tokyo was used with a mesh of 85 million elements and a reduced Reynolds number. For the simulation more than two thousand cores on a Japanese supercomputer were used. Moreover, the generation of such a mesh is difficult and time-consuming.

1.2.2 Simulation of the start-up of a Francis turbine

It was explained in Section 1.1.2 that due to the changing energy markets, Francis turbines and pump-turbines are submitted more frequently to start and stop procedures. The influence of frequent start-ups on the life-cycle of a turbine is not well investigated yet. Only very few studies can be found in literature that aim at simulating the transient flow in a turbine or a pump-turbine at start-up by numerical simulation. One example is Staubli *et al.* [102] who simulate a pump-turbine at no-load condition but do not simulate the start-up from rest.

The discharge in a Francis turbine or pump-turbine is regulated by the movement of the guide vanes. Hence, transient start-up simulations that simulate the turbine from rest to *speed no-load* have to simulate the opening of the guide vanes. Nicolle *et al.* [81] were to our knowledge the first authors who published such a simulation with a pre-described evolution of the opening angles of the guide vanes that was known from measurements. Because of limited computer resources, they simulate only a portion of the turbine, i.e. one stay vane and two guide vanes, one runner blade and a part of the draft tube cone for the outlet. A transient rotor-stator interface is used between the guide vanes and the runner. Thanks to user defined functions that are used together with ANSYS CFX, it is possible to update the mesh after every time step according to the guide vane opening, while keeping the topology and the number of nodes constant. However, the simulation can not be started from completely closed guide vanes, but has to start at an opening angle of one degree. The opening between zero and one degree that can not be simulated directly is modelled by an annular porous loss region. The simulation compares well to the measured values and helps to analyze the transient flow that occurs during start-up.

1.2.3 The meshless method SPH

When simulating transient flows in hydraulic machines, the meshbased character of standard CFD methods introduces important difficulties. Firstly, mesh interfaces between rotating and stationary parts of the turbine necessitate interpolation procedures in regions that are very important for the development of transient flow phenomena. Secondly, moving geometries like opening guide vanes can only by simulated if mesh deformations are possible without degrading the mesh quality or if the domain, or parts of it, is remeshed regularly. We have seen that it is possible in some cases [81] to launch such a simulation but there are other simulations that are computationally too expensive to launch because they demand a remeshing of the whole

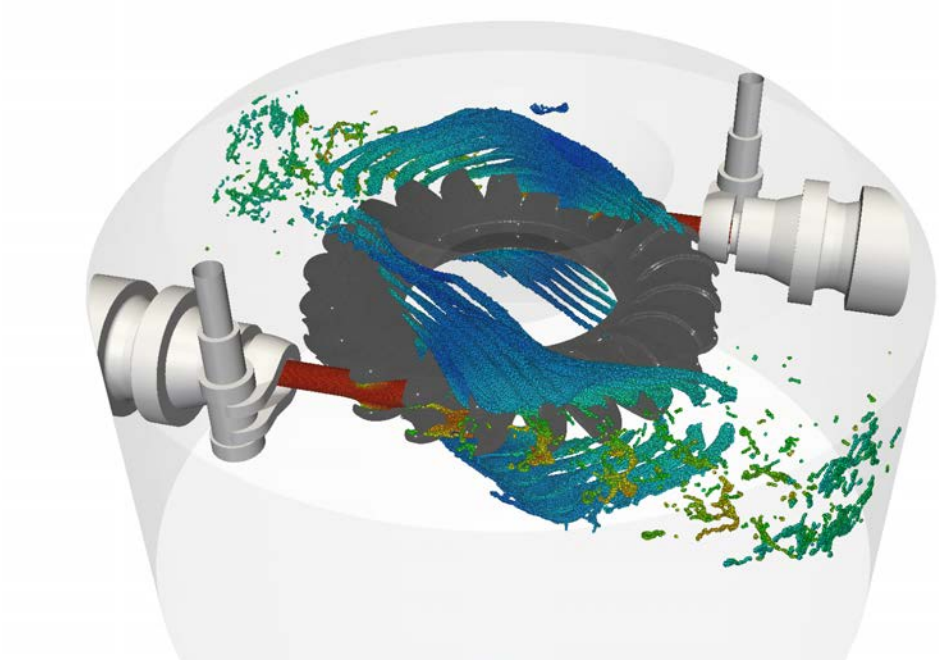


Figure 1.6: Complex free surface flow in the casing of a Pelton turbine obtained by a SPH-ALE software of ANDRITZ HYDRO. High velocity is colored in red and low velocity in blue.

domain at each time step. In addition, mesh generation is often quite cumbersome and needs a skilled and experienced person to do it. Usually automatic mesh generation exists for standard industrial simulations. But in general it does not exist for the kind of meshes that are adapted for simulations of the flows in off-design operating points.

In answer to these inconveniences and because of increased computational power, meshless methods like *Smoothed Particle Hydrodynamics* (SPH) became more popular in the past years. These methods do not need a mesh in the sense that no connectivity information between calculation points, also called particles, has to be known. Furthermore, SPH is a Lagrangian method where the calculation points follow the flow and therefore also capture arbitrarily moving geometries or free surfaces automatically. In the past years, an industrial in-house SPH solver has been developed within the Research & Development Department of ANDRITZ Hydro (Switzerland and France) in collaboration with Ecole Centrale de Lyon (France). The developments were motivated by the need of a CFD method that is able to simulate the complex free surface flow in Pelton runners and casings, see Figure 1.6. Since the position of the complicated free surface of the water sheets is not known beforehand, these simulations are extremely computationally expensive with meshbased methods and not feasible in the time frame of an industrial project. In the contrast, SPH does not need to simulate the air phase that we are not explicitly interested in. As a consequence it needs much less calculation points than meshbased methods for the simulation of free surface flows. However, the standard SPH method suffers from accuracy issues and problems with solid boundary treatment. In the frame of the PhD works of J.-C. Marongiu [67] and J. Leduc [55] a software called ASPHODEL was developed that is based on the more accurate variant SPH-ALE. Amongst others, this variant allows using a flexible solid boundary treatment [67] that is well-suited for industrial simulations of rotating hydraulic machines. It also introduces an additional parameter, the particle velocity v_0 that enables us

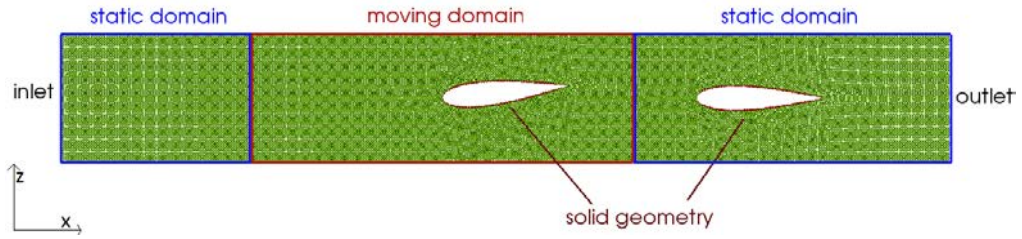


Figure 1.7: Computational domain of a SPH-ALE simulation of two rows of hydrofoils. The first is moving at a constant speed, while the second one is fixed. There are three blocks of particles with different particle motion. The particles in the first and the last block do not move (Eulerian description) but the particles in the block in the middle are translated with the hydrofoil (ALE motion).

to launch Lagrangian simulations, where the calculation points move with the flow, Eulerian simulations where they are fixed and ALE simulations, where the calculation points move with another velocity that is neither Lagrangian nor Eulerian. Moreover, an implemented multiphase solver [55] makes it possible to launch real multi-phase simulations where two phases are computed. Fluid-structure interactions can be simulated thanks to the PhD work of Z. Li [59] who coupled ASPHODEL to the open source software Code-ASTER [27] of EDF.

The ASPHODEL software is used on a daily basis in the Pelton R&D department for several purposes, see e.g. [86]. For example, it is used during the design process of Pelton runners in combination with a FEM stress analysis for stress assessment of the Pelton bucket. Moreover, it helps to understand the complex flow in the Pelton turbine casing and it can be used to simulate the flow at off-design operating conditions as e.g. start-stop sequences of Pelton turbines.

To assess its capabilities in simulating internal flows of rotor-stator systems, first preliminary inviscid SPH simulations have been launched. They show encouraging results but also some weak points of the method as it is illustrated by two examples in the following. A first two-dimensional test case shows two cascades of hydrofoils where one is moving and the other one is stationary, see Figure 1.7. The particles in the different regions move in blocks the same way as the geometry. Figure 1.8 shows the simulated velocity field and shows that the wakes can pass through the different blocks without problems. However, these wakes are artificial because an inviscid simulation was launched! They are caused by the numerical dissipation and should not appear, see Chapter 3 and 4.

Then, an inviscid SPH-ALE simulation of the start-up of a Francis turbine with opening guide vanes is presented. In this simulation the opening of the guide vanes is imposed and the particles move with an ALE velocity in a way that they follow the moving geometry. This type of ALE motion will be explained in detail in Chapter 3, Section 3.9.4. Figure 1.9 shows the velocity field and a detailed view of the position of the particles around the guide vanes. The simulation is started from completely closed guide vanes and the particles distribute themselves in space according to the position of the guide vanes. In addition, no rotor-stator interface, where variables are interpolated, is necessary since the particles follow the geometry automatically. However, the velocity is decelerated near the solid wall which should not happen in inviscid simulations. Again this is the result of the dissipation of the numerical scheme.

Both simulations use particles with constant size because SPH is an intrinsically isotropic method. Space refinement is very difficult to implement and an important topic of current research on SPH, see Chapter 4. However, the gradients of pressure and velocity are changing too

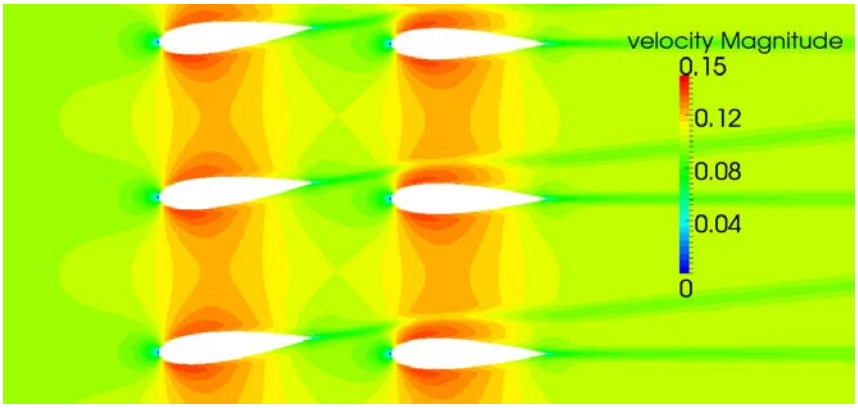
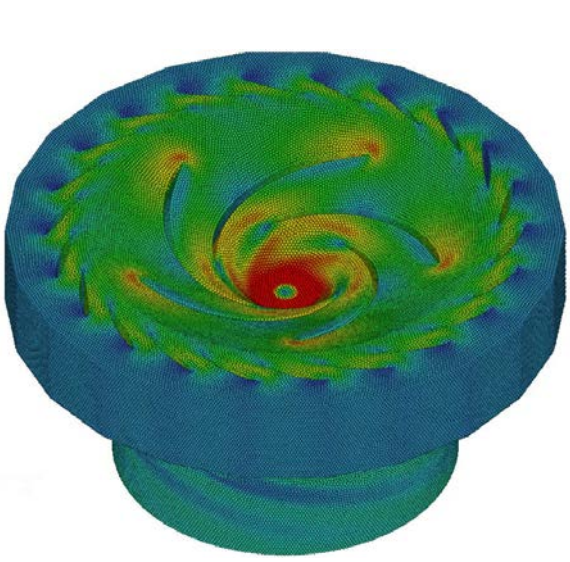
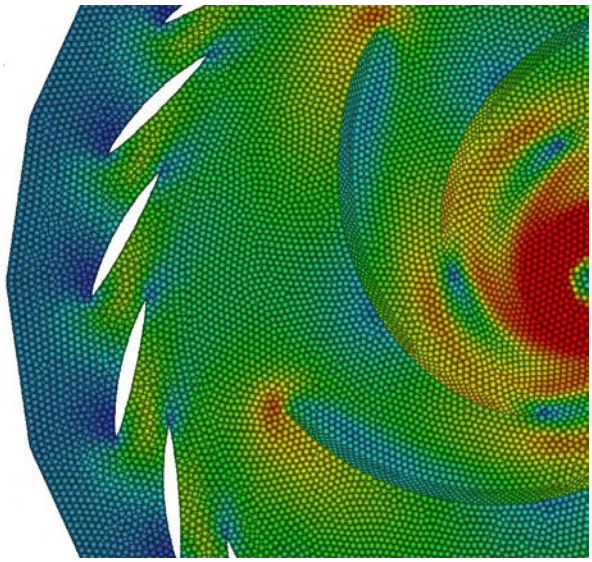


Figure 1.8: Velocity field of two cascades of hydrofoils where the first one is moving and the second one is static.



(a) Overview of the computational domain, where particles are colored by the velocity field with red denoting high velocity and blue low velocity.



(b) Detailed view of the particle distribution around the guide vanes and in the vaneless space between guide vanes and runner.

Figure 1.9: SPH-ALE simulation of the start-up of a Francis turbine. The guide vanes are moving according to an imposed motion. Only the guide vanes, the runner and a cone at the outlet are simulated. The particles adapt themselves to the movement of the geometry and no rotor-stator interface is needed.

rapidly in regions close to wall to be captured by the scheme on this rather coarse discretization. Meshbased methods like FV schemes solve this problem by refining the mesh locally in regions of strong variation, preferably in the direction of the gradient. The constant particle size also imposes itself in regions where such a fine discretization is not necessary, like for example in the spiral casing. Hence, the constant particle size increases the computational cost due to regions where the particle size is too small compared to the size that is imposed by the physical flow phenomena. Furthermore, it deteriorates the results because of regions where a finer spatial discretization is needed to capture the flow features.

1.3 Objective of this thesis

The objective of this thesis is the development of a flexible coupling algorithm of SPH-ALE and a FV method that can be used as a tool for the investigation of transient flows in hydraulic machines like during the previously described start-up procedure, see Sections 1.1.2 and 1.2.2. The SPH method enables the calculation points to move and to adapt themselves to the flow and/or to the solid geometries, while the FV method allows anisotropic refinement in regions of rapidly changing gradients. The algorithm needs to be very flexible and uses a decomposition of the computational domain into regions where the fields are computed by the FVM, regions where they are computed by SPH and overlapping regions where the information is transferred from FV to SPH and vice versa. It should be possible to use the method for fixed SPH particles, particles in Lagrangian motion or particles that move arbitrarily (ALE). FV cells are added in areas where a more accurate (refined) simulation is needed or in regions where there are other advantages to use FVM, as for example at the outlet of the computational domain or in the spiral casing of a Francis turbine. In addition for the coupling, a FV solver has to be implemented into ASPHODEL that discretizes the same form of the weakly-compressible Navier-Stokes equations in ALE formalism than the SPH method.

1.4 Outline of this document

The present document is organizes as follows.

Chapter 2 presents two validation cases, i.e. the inviscid Taylor-Green vortex and the inviscid flow around a symmetric NACA hydrofoil. For the case of the NACA hydrofoil a reference solution is given. These two test cases are used throughout this document for validation purpose.

In Chapter 3 an introduction to the physical model and the numerical methods that are used for the coupling, i.e. the Finite Volume method (FVM) and Smoothed Particle Hydrodynamics (SPH), is given. Furthermore, a novel correction method for the gradient of the SPH kernel function is presented that enforces zeroth order consistency for the computation of the SPH gradients and that improves considerably the computed pressure fields.

The core of the present thesis is Chapter 4 were the developed coupling method of SPH-ALE and FVM is presented. At the beginning, a literature review on variable space discretization in SPH-ALE and FVM and different coupling methods is given. In the second part of this chapter, the coupling is validated by means of academic one-dimensional and two-dimensional inviscid test cases. It is shown that the coupled simulations compare well to analytical solutions or to reference solutions.

Then, the coupling algorithm is applied to two-dimensional simulations in Chapter 5. These show the potential of the method for applications to real three-dimensional simulations of transient flows.

In the end, in Chapter 6, conclusions are given and further perspectives are discussed.

Chapter 2

Validation cases

Our target applications are transient flow simulations in hydraulic machines like Francis turbines and pump-turbines. Therefore, two inviscid two-dimensional validation cases are chosen that are representative for some of the flow phenomena that are found in the target applications. Hence, we choose the Taylor-Green vortex and the flow around a symmetric NACA hydrofoil that both manifest strong gradients of pressure and velocity, once without and once with wall boundaries. They are used several times in this document for different purposes, that is the validation of the FV solver, the validation of a correction method for SPH-ALE and the validation of the coupling of FV and SPH-ALE. For that reason, the test cases themselves as well as the analytical solution of the Taylor-Green vortex and a reference solution of the flow around the NACA hydrofoil are presented in this chapter.

2.1 Taylor-Green vortex

The two-dimensional Taylor-Green vortices are an exact unsteady solution of the incompressible Navier-Stokes equations on a bi-periodic square with a length of L. They are given, for $L = 1$ and for an amplitude of the initial velocity of $U = 1$, by

$$\begin{cases} u = & e^{-8\pi^2 t\nu} \sin(2\pi x)\cos(2\pi y), \\ v = & -e^{-8\pi^2 t\nu} \cos(2\pi x)\sin(2\pi y), \\ p = & \dfrac{\rho}{4} e^{-16\pi^2 t\nu} \left[\cos(4\pi x) + \cos(4\pi y)\right], \end{cases} \tag{2.1}$$

with $(u, v)^T$ denoting the two-dimensional velocity vector, ρ the density, p the pressure and ν the kinematic viscosity. In this work we consider inviscid flows $\nu = 0$ and theoretically Eq. (2.1) describes a steady-state solution of the incompressible Euler equations, see Appendix A.

However, in practice the vortices are decaying, depending on the numerical scheme and the discretization size that are used. Hence, the decay of the maximal velocity is a measure for the dissipation of the numerical scheme that behaves similarly to the physical viscosity [22, 68]. In this study we are not interested in quantifying the dissipation of the scheme but we want to analyze the errors that the fluid solvers, i.e. the FV, the SPH-ALE and the coupled solver, introduce into the pressure and velocity field. The test case is especially useful because a closed-form solution Eq. (2.1) is known. Moreover, it is well adapted to test the behaviour of the solvers for a flow with high gradients without the influence of wall boundary conditions. The analytical solution Eq. (2.1) for inviscid flows that consists of four counter-rotating vortices is plotted in Figures 2.1, 2.2 and 2.3.

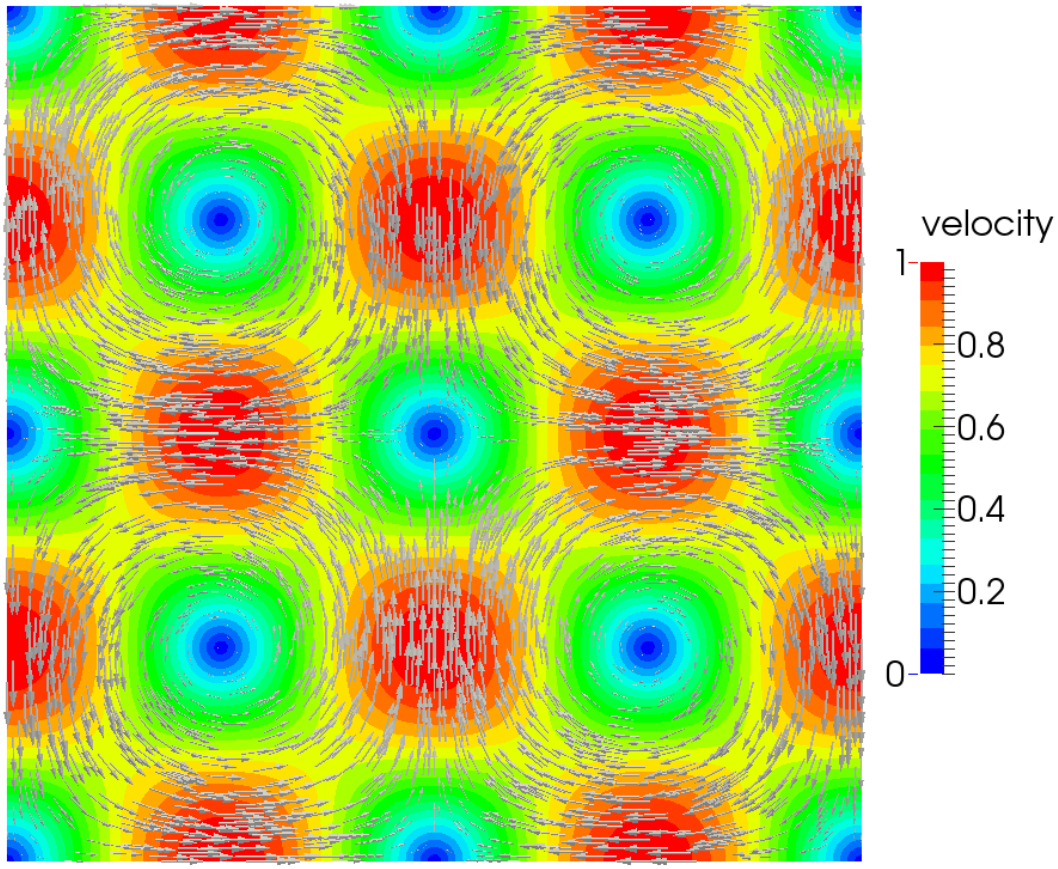


Figure 2.1: Velocity field of the inviscid two-dimensional Taylor-Green vortex. The velocity vectors are scaled by the magnitude of the velocity.

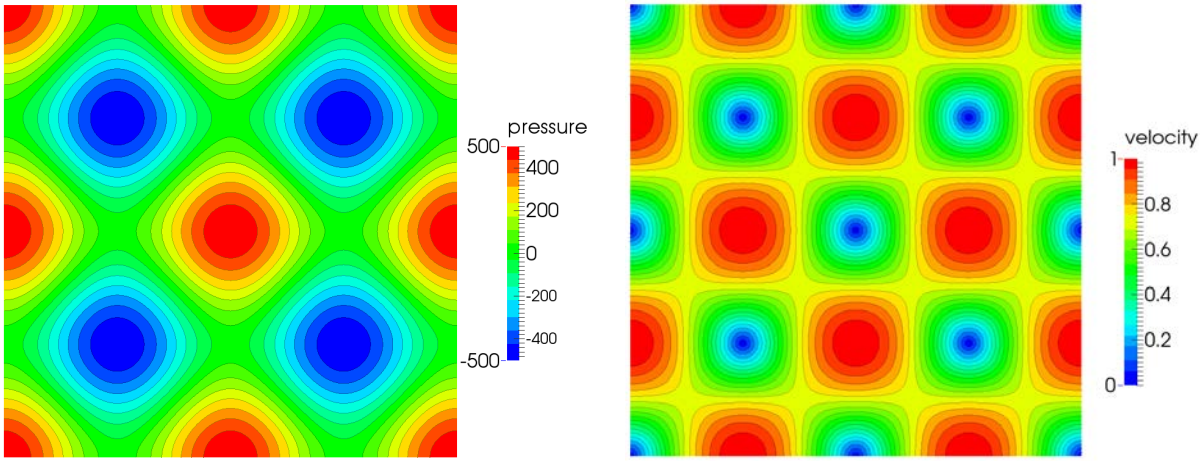
When we use this test case for the validation of a numerical method, the computed pressure and velocity field as well as the derivatives of pressure and velocity are compared to the analytical solution, which is either plotted over the whole computational domain or along a constant coordinate line.

2.2 Flow around a symmetric NACA hydrofoil cascade

An important validation case for our target applications is the inviscid flow around a two-dimensional symmetric hydrofoil with a 4-digit NACA profile as solid geometry. The profile was developed by the National Advisory Committee for Aeronautics (NACA) [1] and is described by

$$\begin{aligned} y &= \frac{T}{0.2}\, C \left[0.2969 \sqrt{\frac{x}{C}} - 0.1260 \, \frac{x}{C} - 0.3516 \left(\frac{x}{C}\right)^2 \right. \\ &+ \left. 0.2843 \left(\frac{x}{C}\right)^3 - 0.1015 \left(\frac{x}{C}\right)^4 \right], \end{aligned} \tag{2.2}$$

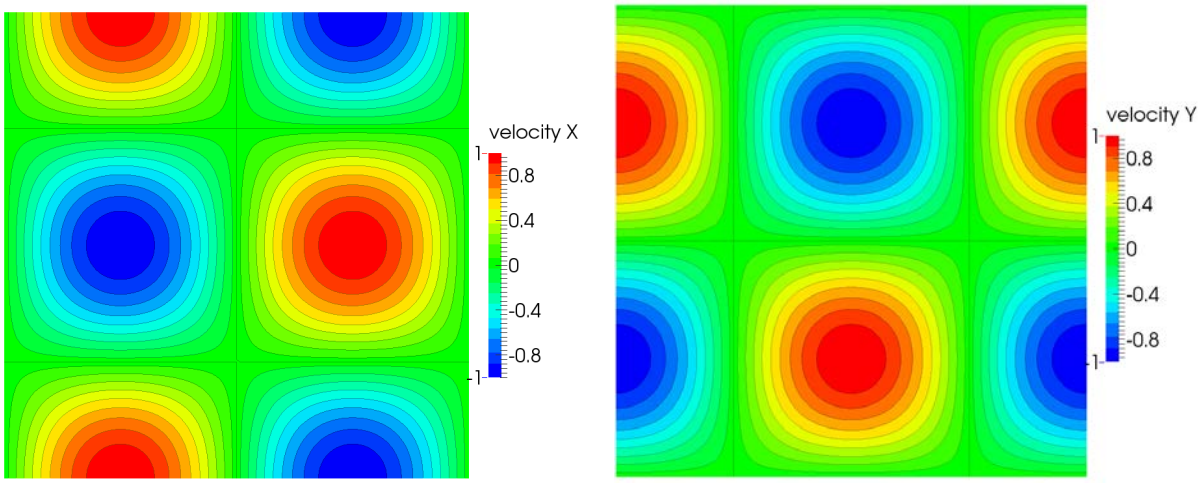
where $C = 0.1$ denotes the chord length and $T = 0.2$ the maximum thickness as a fraction of the chord length. It has a sharp trailing edge. As indicated in Figure 2.4, the computational domain is limited by inlet boundary conditions at the left hand side, outlet boundary conditions



(a) Pressure.

(b) Magnitude of the velocity.

Figure 2.2: Pressure and velocity magnitude of the inviscid two-dimensional Taylor-Green vortex.



(a) The x component of the velocity.

(b) The y component of the velocity.

Figure 2.3: The x and y components of the velocity of the Taylor-Green vortex.

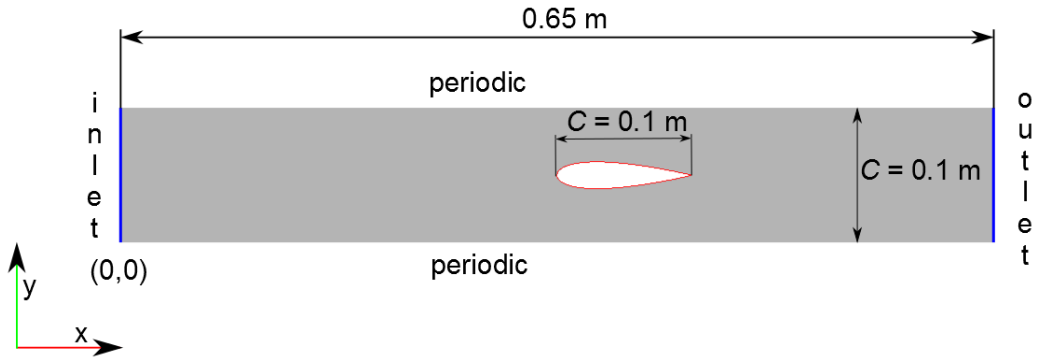


Figure 2.4: The computational domain for the flow simulation around a static symmetric NACA hydrofoil. The inlet is situated at the left hand side and the outlet at the right hand side. Periodicity conditions are imposed elsewhere. The surface of the NACA hydrofoil is indicated by the red line. The leading edge coordinates are $(0.325, 0.05)$.

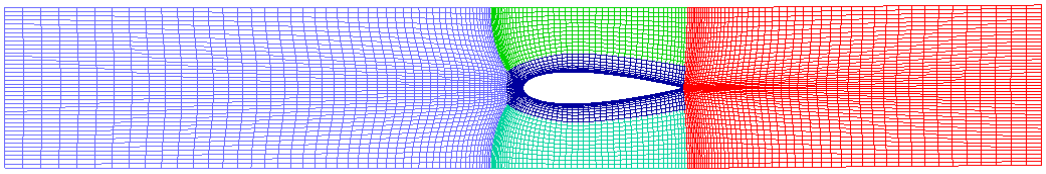


Figure 2.5: A structured multi-block mesh is created for the FV solver. A block of C-grid together with a dove-tail topology is used around the NACA hydrofoil.

at the right hand side and periodic boundary conditions on top and bottom. It is to notice that because of this periodicity condition a cascade of hydrofoils is simulated. Since the channel height is equal to the chord length, the flow solution is expected to be different compared to an isolated profile configuration. At the inlet, constant velocity, $u_{in} = 0.1$ m/s, is imposed, and at the outlet constant zero pressure, $p_{out} = 0$ Pa, is prescribed. The angle of attack is zero.

A reference solution is provided by an inviscid, steady-state, pseudo compressible in-house FV solver of the ANDRITZ group that is based on the work of [35] and [89]. This Euler solver is not the same as the one developed in the frame of this project but it is well-established inside ANDRITZ Hydro for simulations of internal flows. A semi-implicit time integration scheme with local time stepping and a multi-grid algorithm are employed. The pseudo-compressible solver converges to a truly incompressible solution and the numerical speed of sound is only important for the convergence. Figure 2.5 shows the structured multi-block mesh that is used to compute the reference solution. Around the NACA hydrofoil a C-grid with dove-tail topology is used. The mesh blocks are connected by matching block boundaries which means that the mesh lines are continuous through multi-block boundaries and no interpolation is required. The mesh lines are also continuous through the periodicity boundary which makes it easy to impose the periodicity boundary condition. This mesh is generated with an in-house mesh generation software of the ANDRITZ group.

Figure 2.7 shows the pressure field of the reference solution. Since the hydrofoil is symmetric and the angle of attack is zero, the pressure field is also symmetric with a stagnation point

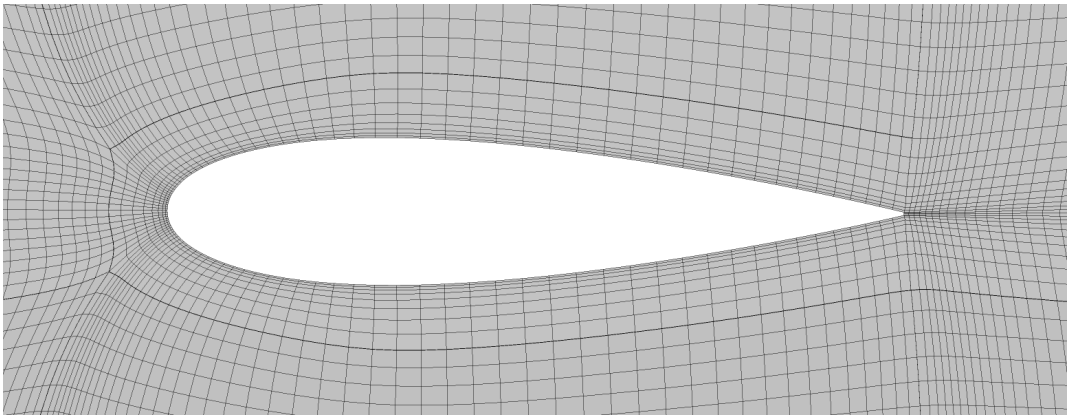


Figure 2.6: Detailed view of the mesh around the NACA hydrofoil.

situated at the leading edge. Because of the short distance between adjacent hydrofoils compared to chord and thickness, the pressure does not drop to zero above and below the profile. All simulations are inviscid which means that no wake should appear in the velocity field downstream of the solid body. Following e.g. [48], we know that the deceleration of the velocity near wall boundaries that manifests itself as a numerical wake is a measure for the numerical dissipation of the method. This numerical dissipation mimics the effects of the viscosity in viscous flows. It is known that the non-physical wake can be reduced by either employing a more accurate solver or using more calculation points in the zone where the wake is created, i.e. close to the solid wall. In the present case a mesh is used that is strongly refined in the direction normal to the surface of the hydrofoil, see Figure 2.6. As a consequence, there is almost no numerical wake, as can be seen in Figure 2.8 that shows the magnitude of the velocity field.

For validation of the implemented FV solver and the coupled solver, the numerical wake, the pressure coefficient and the total pressure loss will be analyzed and compared to the reference solution as it follows. For the comparison of the numerical wake, the x component of the velocity is plotted along two lines defined by $x = 0.45$ m and $x = 0.55$ m, as shown in Figure 2.9(b) for $x = 0.55$.

We define the pressure coefficient on the surface of the NACA hydrofoil as

$$c_p := \frac{p - p_{stag}}{\frac{1}{2}\rho_\infty v_\infty^2}, \tag{2.3}$$

where p_{stag} is the stagnation pressure, i.e. the pressure reached at the stagnation point on the leading edge of the hydrofoil. The density and velocity at infinity are $\rho_\infty = 1000$ kg/m^3 and $v_\infty = 0.1$ m/s and the theoretical value for the pressure at the stagnation point is $p_{stag} = 5$ Pa. Figure 2.9(a) shows the pressure coefficient obtained for the reference solution.

We model an inviscid, incompressible flow even though we use a weakly compressible formalism. Furthermore, we consider a steady flow with uniform inlet, in the present case. According to Bernoulli's principle, the total pressure p_{tot} is constant in incompressible irrotational flows in the whole computational domain. The total pressure is defined as the sum of the static pressure and the dynamic pressure, i.e.

$$p_{tot} := p + \frac{1}{2}\rho \mathbf{v}^2, \tag{2.4}$$

with $\mathbf{v}$ denoting the flow velocity vector. At the stagnation point where $\mathbf{v}_{stag} = 0$ m/s we obtain $p_{stag} = p_{tot}$. That means that for irrotational incompressible flows the total pressure is

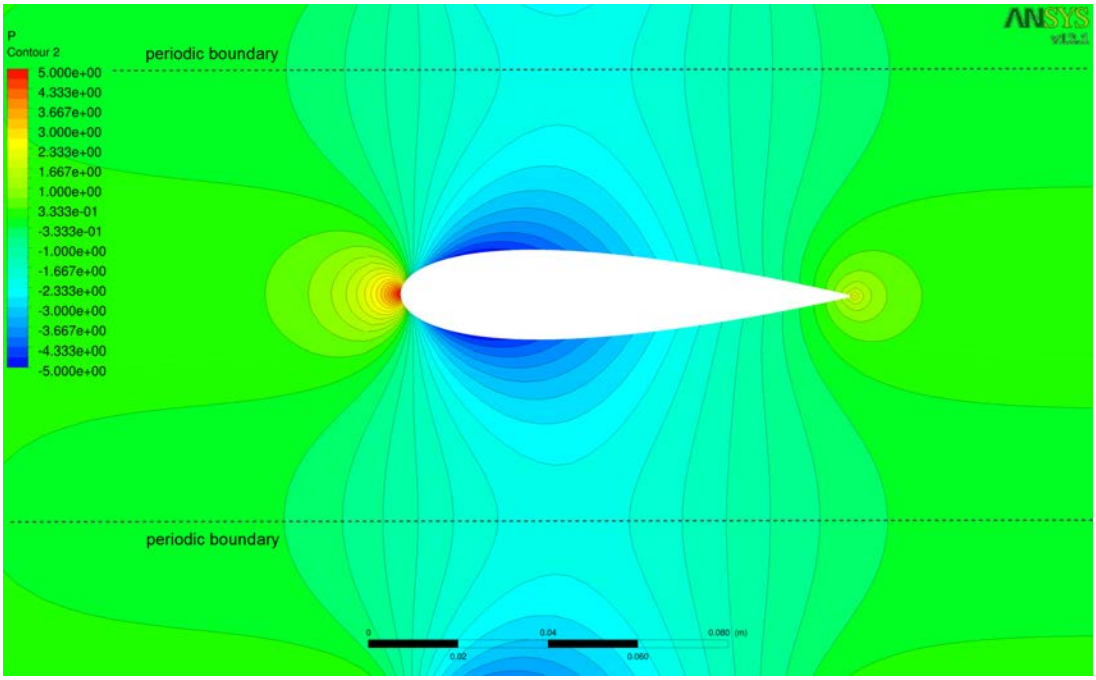


Figure 2.7: Reference pressure field around the NACA hydrofoil.

everywhere equal to the stagnation pressure. This is used in the following to introduce another measure for the numerical dissipation, which is the loss (change) of total pressure, evaluated as the difference between the total pressure area-averaged over a section upstream of the hydrofoil, situated at $x = 0.1$, and a section downstream, situated at $x = 0.6$. The pressure loss obtained by the reference solution is 0.1% and the associated total pressure field is shown in Figure 2.10. This figure shows that the error in total pressure mainly occurs close to wall boundaries where pressure and velocity change rapidly and in the region of the numerical wake.

This kind of analysis provides us with a tool to evaluate the numerical methods that will be presented throughout this document.

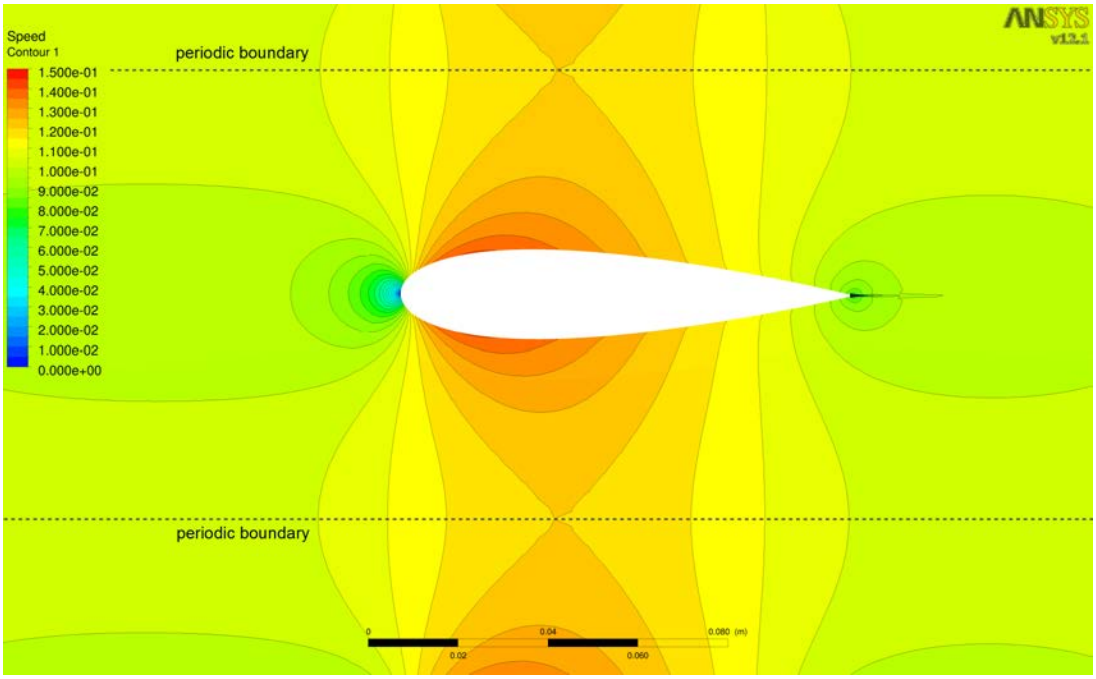
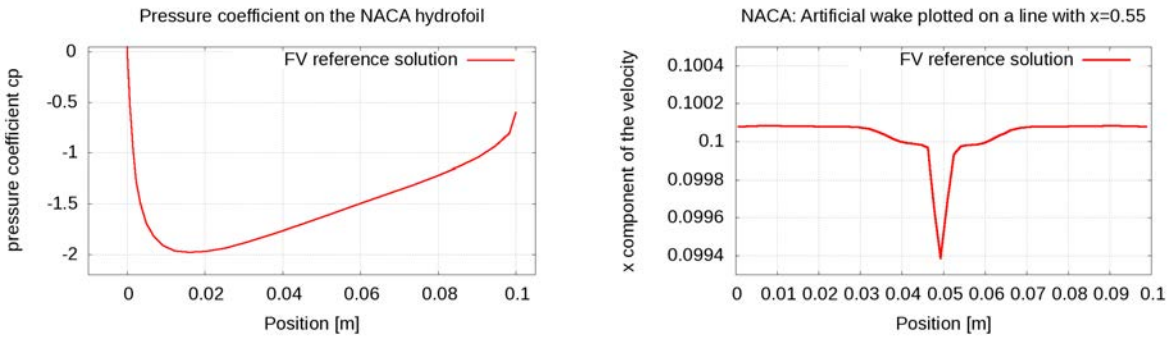


Figure 2.8: Magnitude of the reference velocity field around the NACA hydrofoil.



(a) Pressure coefficient on the surface of the NACA hydrofoil.

(b) The x component of the velocity plotted on a line of $x = 0.55$ m.

Figure 2.9: Pressure coefficient and numerical wake obtained for the reference solution.

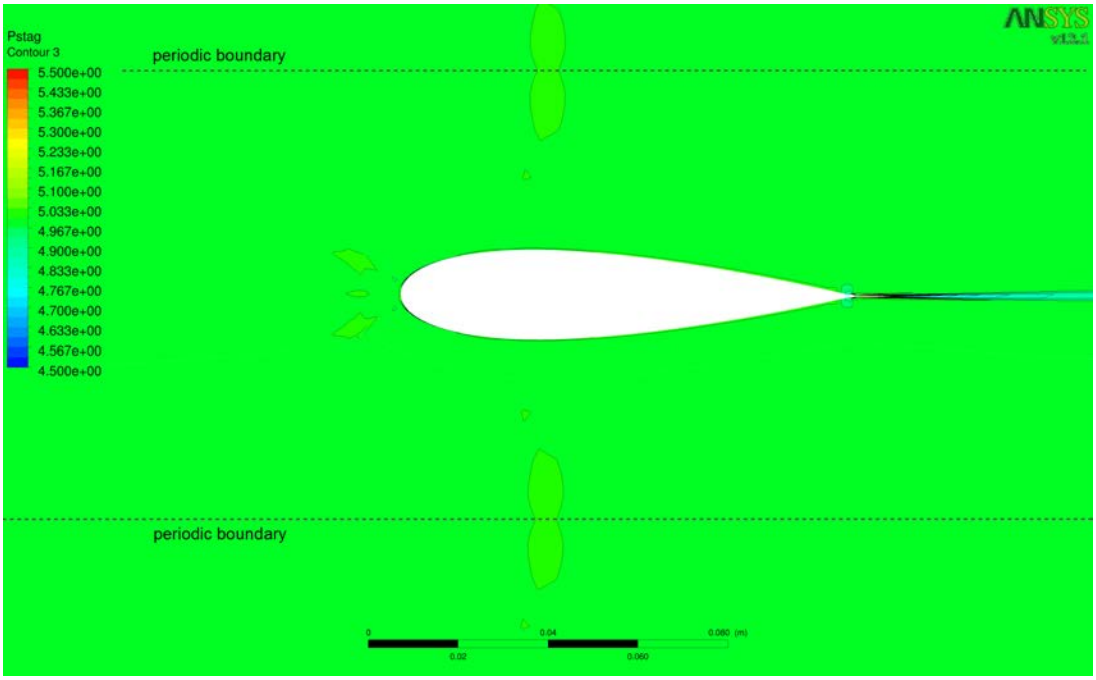


Figure 2.10: Reference total pressure field around the NACA hydrofoil.

Chapter 3

Numerical Methods

In this chapter the numerical methods that are used for the coupling in Chapter 4 are introduced. Both, Smoothed Particle Hydrodynamics (SPH) and Finite Volume (FV) methods, discretize the weakly-compressible Navier-Stokes equations in Arbitrary Lagrange Euler (ALE) form that are presented in Section 3.1. The main characteristics of the methods, as well as their differences and similarities, are analyzed in this chapter. SPH-ALE is a meshless method where fluxes are exchanged between particles, as it is presented in Sections 3.2 and 3.3. The FV method is a meshbased method where fluxes between cells are considered, see Section 3.4. Consequently, they require the computation of numerical fluxes that will be obtained for both methods by Godunov's scheme, explained in Section 3.6. Moreover, a novel correction method for SPH that greatly improves the computed pressure field is developed in Section 3.9. In the end of this chapter in Section 3.10, validations for the corrected SPH-ALE method and for the implemented FV solver are presented.

3.1 Governing equations

Newtonian fluids like water can be described by the Navier-Stokes equations which are a set of non-linear equations based on the physical principles of conservation of mass, momentum and energy. The system of partial differential equations (PDEs) consists of the continuity equation, the momentum equation and the energy equation and is given in the following in conservative differential form, i.e.

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \\ \dfrac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla \cdot (p\,\mathbf{I}) = \mathbf{S} + \nabla \cdot \boldsymbol{\tau}, \\ \dfrac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \mathbf{v} E + p\mathbf{v}) = \mathbf{S} \cdot \mathbf{v} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{v}) - \nabla \cdot \dot{\mathbf{q}} + q_c, \end{cases} \tag{3.1}$$

where $\rho(\mathbf{x}, t)$ is the density, $p(\mathbf{x}, t)$ the pressure, $\mathbf{v}(\mathbf{x}, t)$ the velocity vector, $\boldsymbol{\tau}(\mathbf{x}, t)$ the viscous shear stress tensor, $\mathbf{I}$ the unit tensor, $q_c(\mathbf{x}, t)$ the heat sources, $\mathbf{S}(\mathbf{x}, t)$ the vector of external forces and $\dot{\mathbf{q}}(\mathbf{x}, t)$ the heat flux vector, with the space coordinates $\mathbf{x} \in \mathbb{R}^d$, the physical time $t \in \mathbb{R}_0^+$ and $d = 1, 2, 3$ denoting the space dimension. The total energy $E(\mathbf{x}, t)$ is defined by

$$E := e + \frac{1}{2}\mathbf{v}^2,$$

as sum of the internal energy $e(\mathbf{x}, t)$ and the kinetic energy. The symbol $\otimes$ denotes the outer product that is defined in the following way. Consider two column vectors $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^n$.

Then the outer product $\mathbf{a} \otimes \mathbf{b}$ is equal to the matrix multiplication of $\mathbf{a}$ with the transpose of $\mathbf{b}$, i.e.

$$\mathbf{a} \otimes \mathbf{b} := \mathbf{a}\mathbf{b}^T \in \mathbb{R}^{m \times n}. \tag{3.2}$$

In general, the system of non-linear fully coupled time-dependent equations (3.1) cannot be solved analytically. There are a few special cases with special initial condition and specific assumptions where an analytical solution is known. One famous example is the Taylor-Green vortex, see Section 2.1, that we use for validation purpose throughout this document. In the general case, the solution of Eq. (3.1) is obtained in two steps. The first step is the simplification of the equations by means of physical considerations depending on the fluid or the applications that are studied. The second step is the discretization of space and time in order to compute an approximate solution by numerical methods. In the following paragraph we discuss the first step, while the second one will be covered in Section 3.1.2.

The system of equations (3.1) is closed by an additional equation, a so-called equation of state, that describes the relation between state variables. For example, it defines the internal energy as function of pressure and density, i.e. $e = e(\rho, p)$. Water is usually modelled as incompressible fluid meaning that the density ρ is assumed constant. In that case, the continuity equation becomes

$$\nabla \cdot \mathbf{v} = 0. \tag{3.3}$$

However, we do not model water as an incompressible fluid but as a weakly-compressible one with density variations of less than 1%. We suppose that there are no heat sources and we know from experience that temperature variations are very small. Hence, they will be neglected in the following. A barotropic fluid is modelled which means that the internal energy is constant and that the pressure is a function of the density $p = p(\rho)$. As a consequence, the energy equation does not have to be solved and System (3.1) only consists in the continuity equation and the momentum equation. The new set of equations is closed by a barotropic equation of state that is often referred to as *Tait's equation* (see e.g. [112]),

$$p = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] + p_0, \tag{3.4}$$

where $\gamma = 7$ and ρ_0, p_0 and c_0 denote the reference density, the reference pressure and the reference speed of sound, respectively. Therefrom an expression for the speed of sound $c := \sqrt{\frac{\partial p}{\partial \rho}}$ is derived, i.e.

$$c = c_0 \left(\frac{\rho}{\rho_0} \right)^{\frac{\gamma - 1}{2}}. \tag{3.5}$$

If it is not stated differently, we choose $p_0 = 0$ Pa and $\rho_0 = 1000$ kg/m^3. The consequence of choosing a zero reference pressure is that relative pressures are computed. This is especially important for the free surface treatment in SPH that will be discussed in Section 3.7.2. The equation of state Eq. (3.4) models the compressibility behavior of water well if the real physical speed of sound $c_0 \approx 1500$ m/s is chosen as reference speed of sound. However, this is not what is done in practice since the time step size is connected to the chosen reference speed of sound. In order to speed up the computations, c_0 is most of the time set equal to ten times the maximum flow velocity. This ensures that the flow remains weakly-compressible, i.e. the density does not vary more than one percent, and it allows larger time steps. The Mach number Ma of a flow is defined as the ratio of the fluid velocity and the sound speed,

$$Ma := \frac{\|\mathbf{v}\|}{c},$$

and the weakly-compressibility condition is translated into

$$Ma \leq 0.1.$$

This places the model in the low Mach number limit which influences the accuracy of the numerical methods that we use, see Section 3.6.1 and [41] for details.

If the dynamic viscosity μ is constant, the divergence of the shear stress tensor $\boldsymbol{\tau}$ is given by the following relation depending on the fluid velocity $\mathbf{v}$ and μ,

$$\nabla \cdot \boldsymbol{\tau} = \mu \left(\Delta \mathbf{v} + \frac{1}{3} \nabla \left(\nabla \cdot \mathbf{v} \right) \right). \tag{3.6}$$

Considering incompressible flows, we know that $\nabla \cdot \mathbf{v} = 0$ and Eq. (3.6) further simplifies to

$$\nabla \cdot \boldsymbol{\tau} = \mu \Delta \mathbf{v} = \nabla \cdot \mu \nabla \mathbf{v}. \tag{3.7}$$

Taking all these assumptions into account, the Navier-Stokes equations (3.1) are rewritten in flux vector form,

$$\frac{\partial \boldsymbol{\Phi}}{\partial t} + \nabla \cdot (\mathbf{F}_c - \mathbf{F}_\nu) = \mathbf{Q}, \tag{3.8}$$

with

$$\boldsymbol{\Phi} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \end{pmatrix}, \quad \mathbf{F}_c = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \end{pmatrix}, \quad \mathbf{F}_\nu = \begin{pmatrix} 0 \\ \boldsymbol{\tau} \end{pmatrix} \text{ and } \mathbf{Q} = \begin{pmatrix} 0 \\ \mathbf{S} \end{pmatrix}. \tag{3.9}$$

The vector $\boldsymbol{\Phi}$ is called the vector of conservative variables or state variables and $\mathbf{F}_c$ is the convective flux tensor and the pressure force, and $\mathbf{F}_\nu$ is the viscous flux tensor. In our case, the vector of source terms $\mathbf{Q}$ contains the gravity force which means that $\mathbf{S} = \rho \mathbf{g}$ with $\mathbf{g}$ denoting the gravity vector. The convective flux tensor $\mathbf{F}_c = \mathbf{F}_c(\boldsymbol{\Phi})$ can be expressed as a function of $\boldsymbol{\Phi}$ even if it also depends on the pressure p. But due to Tait's equation, Eq. (3.4), the pressure p itself is a function of the density ρ. The flux tensor $\mathbf{F}_c$ includes the convective flux vectors $\mathbf{F}_c^\alpha = \mathbf{F}_c^\alpha(\boldsymbol{\Phi})$, $\alpha = 1, ..., d$, i.e. $\mathbf{F}_c = (\mathbf{F}_c^1, ..., \mathbf{F}_c^d)^T$.

In the case of inviscid flows, with the dynamic viscosity $\mu = 0$, the viscous fluxes $\mathbf{F}_\nu$ vanish and the *Euler equations* are obtained,

$$\frac{\partial \boldsymbol{\Phi}}{\partial t} + \nabla \cdot \mathbf{F}_c = \mathbf{Q}, \tag{3.10}$$

where $\boldsymbol{\Phi}$ and $\mathbf{F}_c$ are again given by Eq. (3.9).

The mathematical character of the Euler equations differs completely from the Navier-Stokes equations. While the Navier-Stokes equations are a set of second order PDEs, because of the viscous flux vector and Eq. (3.6), the Euler equations are a system of hyperbolic conservation laws for which some mathematical theory exists, see e.g. [58].

In the frame of this thesis only inviscid simulations are carried out, which means that we solve the inviscid Euler equations instead of the viscous Navier-Stokes equations. However, the flows occurring in the target applications that were described in the introduction, Chapter 1, are flows where the viscous effects, especially the viscous boundary layers, play an important role. Even though the coupling algorithm that was developed in the frame of this thesis was implemented and tested for inviscid flows, viscous terms have to be included in the future.

The computational domain is denoted by $\Omega \subset \mathbb{R}^d$ where $\partial\Omega$ is the boundary of the computational domain. Eqs. (3.8) and (3.10) are given in differential form where the divergence of the flux tensor is calculated. As a consequence, only differentiable functions can be a solution of these equations. However in reality, discontinuous fluxes occur in presence of shock waves.

Discontinuous flow variables are only solutions of the integral form of conservation laws, as it is given in the following for the Euler equations,

$$\frac{d}{dt}\int_{\Omega} \mathbf{\Phi}\ dV + \int_{\partial\Omega} \mathbf{F}_c \cdot \mathbf{n}\ dS = \int_{\Omega} \mathbf{Q}\ dV. \tag{3.11}$$

3.1.1 Arbitrary Lagrange Euler

The above equations (3.8) and (3.10) are given in *Eulerian description* meaning that the fluid quantities are analyzed in a fixed frame of reference, where the material fluid particles of the continuum are passing through a fixed region in space. The same equations can also be written in *Lagrangian description*, where the frame of reference follows the material fluid particles in their motion. A generalization of the concept was published in 1974 by Hirt *et al.* [49] where an additional velocity field $\mathbf{v}_0(x,t)$ is introduced that determines the motion of the frame of reference. The approach is called Arbitrary Lagrange Euler (ALE) because $\mathbf{v}_0$ can not only be chosen to be zero to recover the Eulerian equations, or to be equal to the fluid velocity $\mathbf{v}$ to obtain the Lagrangian equations. The transport velocity $\mathbf{v}_0$ can also be used to move the calculation points in a way that suits best the numerical scheme or the considered applications. Figure 3.1 illustrates the differences between Eulerian, Lagrangian and ALE motion.

Using the flux vector notation of Eq. (3.9) and the definition of the outer product Eq. (3.2), we obtain the Euler equations in conservative ALE form with

$$L_{v_0}(\mathbf{\Phi}) + \nabla \cdot (\mathbf{F}_c - \mathbf{v}_0 \otimes \mathbf{\Phi}) = \mathbf{Q}, \tag{3.12}$$

where we define

$$L_{v_0}(\mathbf{\Phi}) := \frac{\partial \mathbf{\Phi}}{\partial t} + \nabla \cdot (\mathbf{v}_0 \otimes \mathbf{\Phi})\,. \tag{3.13}$$

For ease of notation the flux tensor $\mathbf{F}$ is introduced by

$$\mathbf{F}(\mathbf{\Phi}, \mathbf{v}_0) := \mathbf{F}_c(\mathbf{\Phi}) - \mathbf{v}_0 \otimes \mathbf{\Phi}. \tag{3.14}$$

The corresponding integral representation in the moving frame of reference is given by

$$\frac{d}{dt}\int_{\Omega} \mathbf{\Phi}\ dV + \int_{\partial\Omega} \mathbf{F}(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n}\, dS = \int_{\Omega} \mathbf{Q}\ dV, \tag{3.15}$$

where the time derivative has to be considered in the moving system. It is noticed that in this form, the boundary $\partial\Omega$ is moving at speed $\mathbf{v}_0$.

Next, we write the three-dimensional Euler equations explicitly because of their importance in this document. The components of the velocity vector $\mathbf{v}$ are denoted by $\mathbf{v} = (u, v, w)^T$, the components of the transport velocity by $\mathbf{v}_0 = (u_0, v_0, w_0)^T$ and the space coordinates are $\mathbf{x} = (x, y, z)^T$. In 3D, the flux tensor, Eq. (3.14), consists of three flux vectors denoted by $\mathbf{F} = (\mathbf{F}^1, \mathbf{F}^2, \mathbf{F}^3)^T$, that are given by

$$\begin{cases} \mathbf{F}^1(\mathbf{\Phi}, \mathbf{v}_0) = \mathbf{F}_c^1(\mathbf{\Phi}) - u_0\mathbf{\Phi}, \\ \mathbf{F}^2(\mathbf{\Phi}, \mathbf{v}_0) = \mathbf{F}_c^2(\mathbf{\Phi}) - v_0\mathbf{\Phi}, \\ \mathbf{F}^3(\mathbf{\Phi}, \mathbf{v}_0) = \mathbf{F}_c^3(\mathbf{\Phi}) - w_0\mathbf{\Phi}. \end{cases} \tag{3.16}$$

Using the transport operator (3.13) and the vectors

$$\mathbf{\Phi} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{pmatrix}, \quad \mathbf{F}_c^1 = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \end{pmatrix}, \quad \mathbf{F}_c^2 = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \end{pmatrix} \quad \text{and} \quad \mathbf{F}_c^3 = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \end{pmatrix},$$

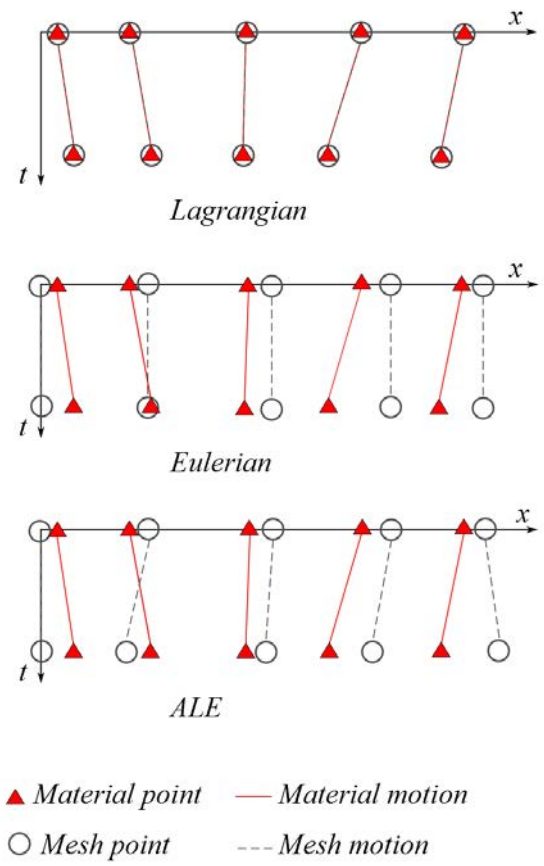
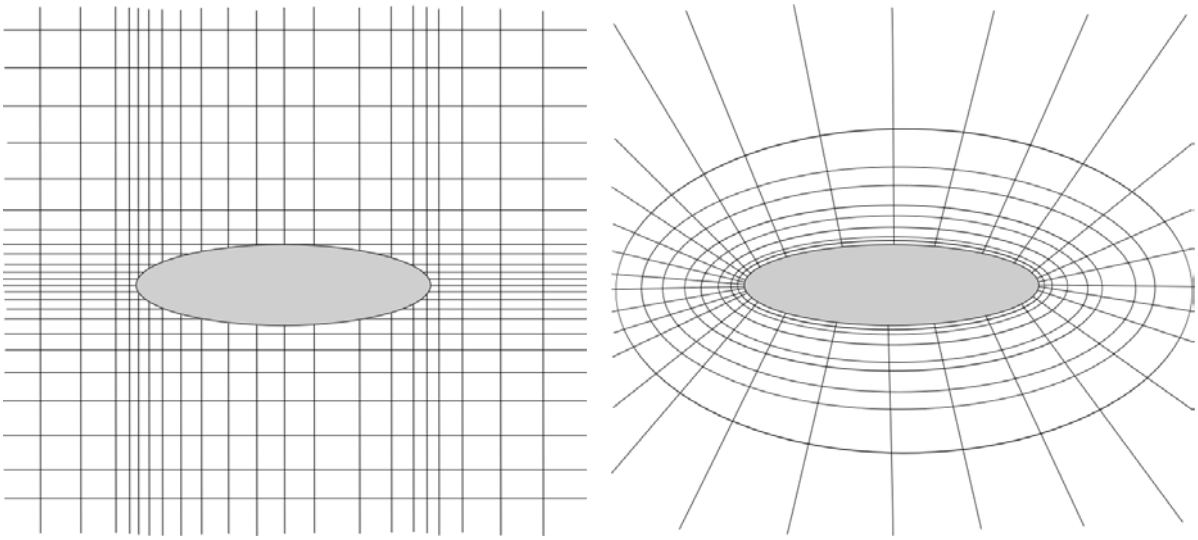


Figure 3.1: Equations can be given in Lagrangian description where the mesh points follow the material points, Eulerian description where the frame of reference is fixed and in the general Arbitrary Lagrange Euler (ALE) form where the mesh points can move arbitrarily. Figure adapted from [24].



(a) Cartesian structured mesh.

(b) Curvilinear structured mesh.

Figure 3.2: Different types of space discretization strategies: structured discretizations.

we obtain the 3D Euler equations in ALE formalism

$$L_{v_0}(\mathbf{\Phi}) + \frac{\partial}{\partial x}\mathbf{F}^1 + \frac{\partial}{\partial y}\mathbf{F}^2 + \frac{\partial}{\partial z}\mathbf{F}^3 = \mathbf{Q}. \tag{3.17}$$

The two-dimensional equations are obtained by setting the third velocity component to zero, $w = 0$, as well as the third flux vector, $\mathbf{F}_c^3 = 0$ and $w_0 = 0$.

3.1.2 Discretizing the continuum

The Euler and the Navier-Stokes equations are a system of non-linear partial differential equations (PDEs) including time and spatial derivatives. For numerical simulations, time and the physical space have to be discretized. A commonly used method is the *method of lines* that discretizes the space to obtain a system of *semi-discrete* ordinary differential equations (ODEs) that then can be discretized in time. It is illustrated as follows,

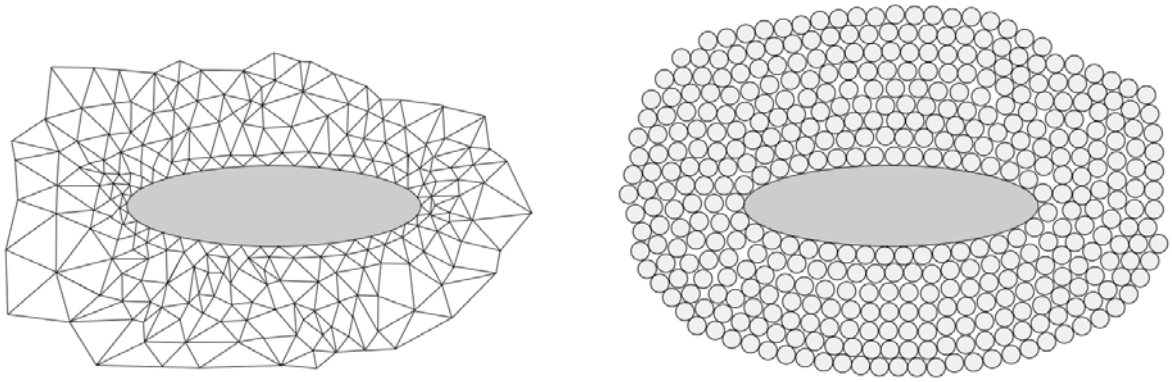
$$\text{PDE} \xrightarrow{\textit{space discretization}} \text{ODE} \xrightarrow{\textit{time integration}} \text{numerical solution.} \tag{3.18}$$

There are also schemes that discretize space and time at once but they will not be used in this document.

Classification of different space discretization strategies

In the main there are three different ways to discretize in space that are closely linked to the solver that will be applied. We distinguish structured meshes, unstructured meshes and meshless discretizations.

- A **structured mesh** can be mapped to the unit square and an index ijk is assigned to every node. The connectivity information can be deduced from the index, i.e. node $(i+1)jk$ is always next to the node with index ijk. Structured meshes can be a Cartesian



(a) Unstructured mesh.

(b) Meshless discretization, particles.

Figure 3.3: Different types of space discretization strategies: unstructured discretizations.

grid (Figure 3.2 (a)) or a curvilinear mesh (Figure 3.2(b)). Compared to Cartesian grids, curvilinear meshes have the advantage that they can be constructed in a way that the mesh lines coincide with the boundaries of the domain. In particular, they are *body-fitted* around solid geometries. Structured meshes are used for Finite Difference methods (FDM) and Finite Volume methods (FVM).

- An **unstructured mesh** is not ordered (Figure 3.3(a)) and the connectivity information has to be stored explicitly by, for example, storing the IDs of the nodes that define a cell or the IDs of the neighbour cells. Unstructured meshes are generated for solvers based on the Finite Element method (FEM) and again Finite Volume methods (FVM).
- **Meshless discretizations** do not store any connectivity information. The numerical stencil consists of clouds of points or particles without any connectivity (Figure 3.3(b)). Methods that do not need any connectivity information are for example Smoothed Particle Hydrodynamics (SPH) or similar methods.

In this document, we describe the implementation of an unstructured Finite Volume solver even though we use blocks of curvilinear meshes. Furthermore, a meshless discretization is inherent to the SPH-ALE method that we employ in the frame of this project.

3.2 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a numerical method to solve partial differential equations on a cloud of calculation points $\mathbf{x}_i$ in a computational domain $\Omega \subset \mathbb{R}^d$. It is a *meshless* or *meshfree* method because there is no connectivity between the calculation points. Traditionally, the SPH operators, that will be introduced below, are used to discretize the Euler or Navier-Stokes equations in Lagrangian form. The calculation points are interpreted as particles that follow the flow in Lagrangian motion.

The method was first published by Lucy in 1977 [65] for astrophysical problems. Then it was further developed for astrophysics by Gingold and Monaghan [37]. In the 1980s, Monaghan applied the method with success to hydrodynamic applications [76], and in particular to free surface applications [78]. It was found that for several reasons SPH is especially well suited to compute dynamic free-surface flows. The fluid particles follow the flow and they are able to capture the position of the free surface. In addition, no mesh has to be created to connect the calculation points that are displaced with the flow. The points can adapt themselves flexibly to changes in the flow field or in the solid geometry.

Introductions to SPH can be found in several publications, e.g. [61, 55, 67, 112]. In the following sections an introduction to SPH is given with a special emphasis on SPH-ALE, the variant of SPH that we actually use.

3.2.1 Kernel approximation

Consider a sufficiently smooth scalar or vector field $f(\mathbf{x})$ in Ω that is written as a spatial convolution product with the Dirac delta function δ,

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}') \; dV'.$$

The Dirac delta function $\delta(\mathbf{x} - \mathbf{x}')$ is zero everywhere but for the case that $\mathbf{x} = \mathbf{x}'$ and it verifies

$$\int_{\Omega} \delta(\mathbf{x} - \mathbf{x}') \; dV' = 1.$$

The so-called *integral representation* or *kernel approximation* of a function f is defined by replacing the *delta* function by a smooth kernel function. That is

$$< f(\mathbf{x}) >_h := \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h) \; dV', \tag{3.19}$$

where $W(\mathbf{x} - \mathbf{x}', h)$ is the *smoothing kernel function* and h is the smoothing length. The kernel function W is a continuous and differentiable function, whose gradient can be calculated analytically. It is defined so that $\frac{1}{W}$ is homogeneous to a volume. Furthermore, the kernel function is chosen in such a way that it satisfies the following conditions.

- *Symmetry*: It is a *symmetric* (even) function, i.e. $W(\mathbf{x} - \mathbf{x}', h) = W(\mathbf{x}' - \mathbf{x}, h)$. A direct consequence from this condition is that

$$\int_{\Omega} (\mathbf{x} - \mathbf{x}')W(\mathbf{x} - \mathbf{x}', h) \; dV' = 0. \tag{3.20}$$

  The left side of the equation is called the *first moment* of the kernel function.

- *Normalization*: The kernel function is *normalized*, i.e.

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) \; dV' = 1. \tag{3.21}$$

- *Compact support*: The kernel function has a compact support that depends on the smoothing length h, i.e.
$$W(\mathbf{x}-\mathbf{x}',h)=0, \text{ if } \|\mathbf{x}-\mathbf{x}'\|>\kappa h, \tag{3.22}$$
where κ is a constant related to the chosen smoothing function, most of the times $\kappa=2$ or $\kappa=3$. The kernel support of $\mathbf{x}$ is defined by
$$D(\mathbf{x}):=\{\mathbf{x}'\in\Omega : W(\mathbf{x}-\mathbf{x}',h)\neq 0\}=\{\mathbf{x}'\in\Omega : \|\mathbf{x}-\mathbf{x}'\|\leq\kappa h\}.$$

- *Delta function property*: The kernel function tends to the Dirac delta function if h tends to zero,
$$\lim_{h\to 0} W(\mathbf{x}-\mathbf{x}',h)=\delta(\mathbf{x}-\mathbf{x}'). \tag{3.23}$$

- *Decay*: The function $W(\mathbf{x}-\mathbf{x}',h)$ is monotonically decreasing when the distance between $\mathbf{x}$ and $\mathbf{x}'$ increases.

- *Positivity*: The function should be positive, i.e.
$$W(\mathbf{x}-\mathbf{x}',h)>0 \text{ for } \mathbf{x}'\in D(\mathbf{x}), \tag{3.24}$$
to avoid non-physical approximations. For example, the integral of a positive function like the density should not become negative.

Since the kernel function has a compact support, the kernel approximation (3.19) is written as
$$< f(\mathbf{x}) >_h = \int_{D(\mathbf{x})} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)\ dV', \tag{3.25}$$
integrating over the support instead of the whole domain. Thanks to that property the approximation becomes a local approximation where $< f(\mathbf{x}) >_h$ does not depend on the whole computational domain but on a compact subdomain depending on the smoothing length h.

Applying Conditions (3.21) and (3.20), directly yields
$$\mathbf{x} = \int_{D(\mathbf{x})} \mathbf{x}'\, W(\mathbf{x}-\mathbf{x}',h)\ dV', \tag{3.26}$$
meaning that $\mathbf{x}$ is situated in the barycenter of the domain $D(\mathbf{x})$ if the normalization and symmetry condition are fulfilled. If we insert the second-order Taylor series approximation of function $f(\mathbf{x})$ in Eq. (3.25) and if we apply Eq. (3.20) and Eq. (3.21), we see that the kernel approximation is equal to the value of the function plus a second order error term,
$$< f(\mathbf{x}) >_h = f(\mathbf{x}) + O(h^2). \tag{3.27}$$
The order of consistency of an approximation is defined as the highest order of a polynomial which can be reproduced exactly. We see from Eq. (3.27) that the kernel approximation has first-order consistency with a second order error term. Higher order kernel approximations of functions of kth order can be constructed but they imply that the kth order moments have to be zero, i.e.
$$M_k := \int_{D(\mathbf{x})} (\mathbf{x}-\mathbf{x}')^k W(\mathbf{x}-\mathbf{x}',h)\ dV' = 0, \quad \forall k>0. \tag{3.28}$$

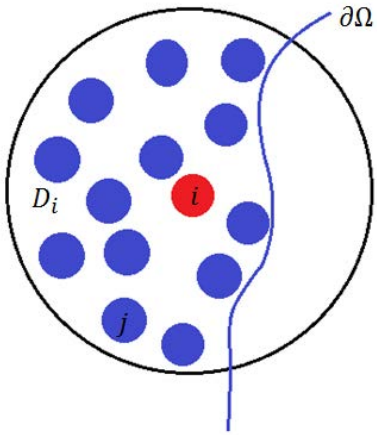


Figure 3.4: The kernel support D_i of particle i is truncated by the boundary of the computational domain $\partial\Omega$.

The zeroing conditions of the moments is in contradiction with the positivity condition of the kernel for moments of even orders with $k \geq 2$. In practice, the positivity of the kernel is preferred over a higher order of consistency in most of the cases.

In Section 3.2.3 we will see that kernel functions can be constructed that fulfil the Conditions (3.20)-(3.24) and constant and linear consistency are obtained. However, this is only true if $\mathbf{x}$ is not situated close to the boundary of the computational domain $\partial\Omega$, see Figure 3.4. If the support $D(\mathbf{x})$ is truncated by the boundary of the domain, the normalization condition is not satisfied any more and the order of consistency is reduced.

The kernel approximation (3.25) can also be applied to the gradient of a field ∇f,

$$< \nabla f(\mathbf{x}) >_h = \int_{D(\mathbf{x})} \nabla f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)\ dV'.$$

The key point of the kernel approximation method is that spatial derivatives of f are computed by putting the derivatives on the kernel function using integration by parts, i.e. the kernel approximation of the gradient of a function is computed as

$$< \nabla f(\mathbf{x}) >_h = \int_{D(\mathbf{x})} f(\mathbf{x}') \nabla W(\mathbf{x} - \mathbf{x}', h)\ dV' + \int_{\partial D(\mathbf{x})} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)\mathbf{n}\ dS, \tag{3.29}$$

where the gradient of the kernel function $\nabla W(\mathbf{x}-\mathbf{x}', h)$ is calculated analytically. The boundary of the support $D(\mathbf{x})$ is denoted by $\partial D(\mathbf{x})$. Far away from the domain boundary $\partial\Omega$, the second integral is zero because of the compactness of the support. However, if the boundary of the domain intersects the kernel support, $D(\mathbf{x}) \cap \partial\Omega \neq \emptyset$, the last integral is non-zero.

It follows from the symmetry of the kernel function that the gradient is antisymmetric,

$$\nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) = -\nabla_{\mathbf{x}'} W \mathbf{x} - \mathbf{x}', h). \tag{3.30}$$

Moreover, it can be also shown by Taylor series expansion that the kernel approximation of the gradient ∇f is first order consistent, far away from boundaries, because a kernel function that fulfils the above conditions also satisfies

$$\int_{D(\mathbf{x})} \nabla W(\mathbf{x} - \mathbf{x}', h)\ dV' = 0, \tag{3.31}$$

and

$$\int_{D(\mathbf{x})} (\mathbf{x}' - \mathbf{x}) \otimes \nabla W(\mathbf{x}' - \mathbf{x}, h)\ dV' = -\mathbf{I}. \tag{3.32}$$

Again this is only true if the kernel support $D(\mathbf{x})$ is not truncated by the boundary of the computational domain $\partial\Omega$.

If $\mathbf{x}$ is located close to the boundary and the kernel support $D(\mathbf{x})$ is truncated, Conditions (3.31) and (3.32) become

$$\int_{D(\mathbf{x})} \nabla W(\mathbf{x} - \mathbf{x}', h)\ dV' + \int_{\partial D(\mathbf{x})} W(\mathbf{x} - \mathbf{x}', h)\mathbf{n}\ dS = 0, \tag{3.33}$$

and

$$\int_{D(\mathbf{x})} (\mathbf{x}' - \mathbf{x}) \otimes \nabla W(\mathbf{x}' - \mathbf{x}, h)\ dV' + \int_{\partial D(\mathbf{x})} (\mathbf{x}' - \mathbf{x}) \otimes W(\mathbf{x}' - \mathbf{x}, h)\mathbf{n}\ dS = -\mathbf{I}, \tag{3.34}$$

which are not fulfilled in general. Equally the normalization condition (3.21) is not verified.

Following De Leffe [22], the kernel function can be corrected by

$$\widetilde{W}(\mathbf{x} - \mathbf{x}', h) = \frac{1}{\gamma(\mathbf{x})} W(\mathbf{x} - \mathbf{x}', h), \tag{3.35}$$

with

$$\gamma(\mathbf{x}) = \int_{D(\mathbf{x})} W(\mathbf{x} - \mathbf{x}', h)\ dV'. \tag{3.36}$$

Then the kernel approximations are redefined as it follows,

$$< f(\mathbf{x}) >_h = \frac{1}{\gamma(\mathbf{x})} \int_{D(\mathbf{x})} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)\ dV', \tag{3.37}$$

and for the gradients

$$\begin{aligned} < \nabla f(\mathbf{x}) >_h = & \frac{1}{\gamma(\mathbf{x})} \int_{D(\mathbf{x})} f(\mathbf{x}') \nabla W(\mathbf{x} - \mathbf{x}', h)\ dV' \\ & + \frac{1}{\gamma(\mathbf{x})} \int_{\partial D(\mathbf{x})} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)\mathbf{n}\ dS. \end{aligned} \tag{3.38}$$

In the interior of the domain $\gamma(\mathbf{x}) = 1$ and it does not modify the kernel approximation. However, close to boundaries it compensates for the truncated kernel support.

3.2.2 Particle approximation

In SPH the continuous space is replaced by a finite set of M calculation points without any connectivity. The reconstructed value of f at the position $\mathbf{x}_i$ of point $i = 0, ..., M-1$ is calculated by the discrete equivalent of Eq. (3.25) that is called *particle approximation*. It is defined by

$$< f_i >:= \sum_{j \in D_i} \omega_j f_j W_{ij}, \tag{3.39}$$

where $D_i := D(\mathbf{x}_i)$ denotes the kernel support of $\mathbf{x}_i$ with boundary ∂D_i, $W_{ij} := W(\mathbf{x}_j - \mathbf{x}_i, h_{ij})$ with

$$h_{ij} := \frac{h_i + h_j}{2}$$

and ω_j the integration weight associated with calculation point j. According to Eq. (3.29) the gradient of f at $\mathbf{x}_i$ is computed by

$$< \nabla f_i >= \sum_{j \in D_i} \omega_j f_j \nabla_i W_{ij} + \sum_{j \in \partial D_i} \omega_j^{\partial} f_j W_{ij} \mathbf{n}_j, \tag{3.40}$$

where $\nabla_i W_{ij}$ denotes the gradient of the kernel function W_{ij} evaluated at the particle position $\mathbf{x}_i$. The second term in Eq. (3.40) is the discretization of the surface integral over the boundary of the support [67]. It is zero far away from domain boundaries but it has to be computed if the kernel support is truncated by the boundary of the computational domain $\partial\Omega$. A consequence of this term is that a discretization of the boundary in surface elements with a surface area ω^{∂} is required.

The physical interpretation of the particle approximation is that the calculation points can be considered as particles and the integration weight ω_i as their volume. The position of the particle is $\mathbf{x}_i$ and each particle carries its physical fields, as pressure p_i and velocity $\mathbf{v}_i$. For the discretization of the surface integral we use surface elements that have a surface area of ω_j^{∂} in 3D. In 2D the surface elements are line segments and in 1D points.

The kernel function is still a symmetric function, i.e. $W_{ij} = W_{ji}$, and the gradient of the kernel function is anti-symmetric,

$$\nabla_i W_{ij} = -\nabla_j W_{ij}. \tag{3.41}$$

The anti-symmetry of the gradient is important to ensure conservation properties as will be seen below in Section 3.3.

Again, we insert the Taylor series expansion of f and ∇f into the formulae of the particle approximation. In contrast to the continuous kernel approximation, we do not even have zeroth order consistency because the following conditions are not satisfied in their discrete form. The first condition is the discrete analogon of the normalization condition Eq. (3.21),

$$\sum_{j \in D_i} \omega_j W_{ij} = 1, \tag{3.42}$$

and the second condition is the discrete form of Eq. (3.33),

$$\sum_{j \in D_i} \omega_j \nabla_i W_{ij} + \sum_{j \in \partial D_i} \omega_j^{\partial} W_{ij} \mathbf{n}_j = 0. \tag{3.43}$$

In practice, the particle approximation Eq. (3.40) is not able even far away from boundaries to compute the gradient of a constant function $c(\mathbf{x}) = c$ correctly because

$$< \nabla c_i >= \sum_{j \in D_i} c_j \omega_j \nabla_i W_{ij} = c \underbrace{\sum_{j \in D_i} \omega_j \nabla_i W_{ij}}_{\neq 0} \neq 0.$$

To enforce zero consistency for the computation of a gradient, the particle approximation is instead of Eq. (3.40) defined by

$$< \nabla f_i >:= \sum_{j \in D_i} \omega_j (f_j - f_i) \nabla_i W_{ij} + \sum_{j \in \partial D_i} \omega_j^{\partial} (f_j - f_i) W_{ij} \mathbf{n}_j. \tag{3.44}$$

This definition is also justified for non-constant functions because the added term is the particle approximation of the gradient of a constant which should be zero in any case. We see that f_i is constant in the sum and that the added term is the particle approximation of the gradient of the unity multiplied by a constant,

$$\begin{aligned} f_i < \nabla 1 > &= f_i \sum_{j \in D_i} \omega_j \nabla_i W_{ij} + f_i \sum_{j \in \partial D_i} \omega_j^{\partial} W_{ij} \mathbf{n}_j \\ &= \sum_{j \in D_i} \omega_j f_i \nabla_i W_{ij} + \sum_{j \in \partial D_i} \omega_j^{\partial} f_i W_{ij} \mathbf{n}_j. \end{aligned}$$

By reusing γ from (3.36), evaluated at $\mathbf{x}_i$, for the definition of the particle approximation, De Leffe [22] defines the SPH operator for the gradient of a function by

$$< \nabla f_i >= \frac{1}{\gamma(\mathbf{x}_i)} \sum_{j \in D_i} \omega_j \left(f_j - f_i\right) \nabla_i W_{ij} + \frac{1}{\gamma(\mathbf{x}_i)} \sum_{j \in \partial D_i} \omega_j^{\partial} \left(f_j - f_i\right) W_{ij} \mathbf{n}_j. \tag{3.45}$$

It would be possible to discretize $\gamma(\mathbf{x}_i)$ by the SPH operator Eq. (3.39) but in literature (e.g. [22]) it is preferred not to discretize the term. In this way the equations are not modified far away from boundaries because $\gamma(\mathbf{x}_i) = 1$ and $\gamma(\mathbf{x}_i)$ only takes effect for boundary treatments. Different ways to evaluate γ as accurate as it is computationally affordable are discussed in [22, 30, 74].

Summarizing, there are three different approaches in literature to compute the gradient of a function by SPH operators in the case where the kernel support is truncated by the boundary of the domain. Either Eq. (3.40) or Eq. (3.44) are used without the boundary term. In this case, the kernel support has to be completed by artificial particles with prescribed physical fields. Or the boundary term in Eq. (3.40) or Eq. (3.44) is directly computed by discretizing the boundary surface. In this case, there is two possibilities: either compute the $\gamma(\mathbf{x}_i)$ term and therefore use Eq. (3.45) or do not compute this term. We decided not to compute $\gamma(\mathbf{x}_i)$ but to compute the boundary term for the following reason. It has been shown that in order to avoid problems at free surfaces, the $\gamma(\mathbf{x}_i)$ term should not be discretized but should be calculated analytically. This is computationally expensive in case of three-dimensional simulations with complex geometries that we are interested in. Hence, we will use the SPH operator from Eq. (3.44) as starting point for the discretization of the equations in Section 3.3.

3.2.3 Choice of a kernel function

In the past, some work has been done to construct sufficiently smooth kernel functions that satisfy the Conditions (3.20)-(3.24). Traditionally in SPH B-spline functions are used as kernel functions ([55, 61]). This family of functions has the form,

$$B(\mathbf{x}, h) = \frac{\sigma_d}{h^d} \theta \left(\frac{\|\mathbf{x}\|}{h} \right),$$

where σ_d is a constant that enforces the normalization condition Eq. (3.21), h is the smoothing length and d the number of dimensions. For the third order B-spline, for example, the function θ is piecewise defined by

$$\theta(q) := \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & \text{if } 0 \leq q < 1, \\ \frac{1}{4}\left(2 - q\right)^2 & \text{if } 1 \leq q \leq 2, \\ 0 & \text{elsewhere,} \end{cases} \tag{3.46}$$

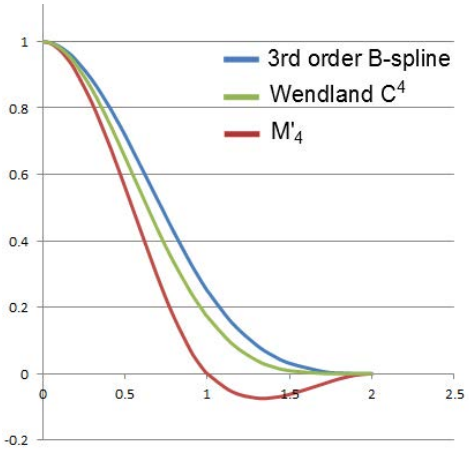


Figure 3.5: Different kernel functions, scaled in a way that $W(0) = 1$.

and the constant σ_d is $\sigma_2 := 10/(7\pi)$ in 2D and $\sigma_3 := 1/\pi$ in 3D.

Wendland [115] introduced another family of functions, the so-called *Wendland kernels* of the following form,

$$W(\mathbf{x}, h) = \frac{\sigma_d}{h^d}\,\theta\left(\frac{\|\mathbf{x}\|}{2h}\right).$$

For the Wendland C^4 function, the function θ is defined by

$$\theta(q) := (1-q)_+^5(8q^2 + 5q + 1), \tag{3.47}$$

with the *truncated power function*

$$x_+^n = \begin{cases} x^n & \text{if } x > 0, \\ 0 & \text{if } x \le 0. \end{cases}$$

The constant σ_d is given by $\sigma_2 := 3/(4\pi)$ in 2D and $\sigma_3 := 165/(256\pi)$ in 3D.

Throughout this thesis we use a third order B-spline or the Wendland C^4 kernel function to solve the SPH flow equations, whereas the Wendland kernel is preferred. However, still another type of kernel, the M_4' kernel is used in some cases for interpolation purpose.

This function was introduced by Monaghan in 1985 [75] and used by Koumoutsakos and his co-workers (see e.g. [20]) for their remeshed particle method. It is a modification of a B-spline function where θ is defined as

$$\theta(q) := \begin{cases} 1 - \frac{5}{2}q^2 + \frac{3}{2}q^3 & \text{if } 0 \le q \le 1, \\ \frac{1}{2}(2-q)^2(1-q) & \text{if } 1 < q \le 2, \\ 0 & \text{elsewhere.} \end{cases} \tag{3.48}$$

The function M_4' verifies all of the above stated properties but one: it is not positve on the whole kernel support area because

$$\frac{1}{2}(2-q)^2(1-q) < 0, \quad \text{if } 1 < q \le 2.$$

It was constructed to fulfil the zeroing condition of its second moment, see Eq. (3.28), which is possible since it is not a positive function on its support. Hence, the kernel approximation using

the M_4' kernel function reproduces exactly a second order polynomial, i.e. it is second order consistent.

Figure 3.5 shows the Wendland C^4, the third order B-spline and the M_4' kernel function, plotted without multiplication by the constant $\frac{\sigma_d}{h}$. As stated above, this constant ensures the normalization property but in this figure we just compare the shape of the functions.

3.2.4 Convergence of the SPH operators

The particle approximations Eq. (3.39) and (3.40) have two different parameters of space discretization, namely the particle size Δx and smoothing length h. On the one hand, the particle size Δx defines an average spacing between the particles and so the number of calculation points in the computational domain. On the other hand, the smoothing length determines the size of the kernel support and so the number of neighbours of a particle. To obtain spatial convergence, both discretization parameters have to tend to zero. Moreover, Raviart [93] showed that for convergence we also need that the number of neighbours increases as both discretization parameters tend to zero,

$$\frac{h}{\Delta x} \to \infty \quad \text{as } \Delta x \to 0, \quad h \to 0. \tag{3.49}$$

It is often not possible to increase the number of neighbours while decreasing the discretization size because of the computational cost. Like most authors we keep a constant ratio of Δx and h, like for example

$$\frac{h}{\Delta x} = 1.2,$$

which works well in practice.

3.2.5 SPH and its variants

The original SPH method for fluid dynamics was mainly developed by Monaghan and is often referred to as standard weakly-compressible SPH (WCSPH). It is a Lagrangian method where the displacement of the particle position is given by

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i. \tag{3.50}$$

The system of SPH equations for fluid dynamics further consists of the SPH discretization of the momentum equation,

$$\frac{D\mathbf{v}_i}{Dt} = \mathbf{f}_S - \sum_{j \in D_i} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij}, \tag{3.51}$$

where $\mathbf{f}_S$ denotes a volumic source term ($\mathbf{S} = \rho\, \mathbf{f}_S$) and $\frac{D}{Dt}$ the total time derivative, and of an equation for the density. Note that m_i denotes the mass of particle i and the particle volume can be computed by

$$\omega_i = \frac{m_i}{\rho_i}.$$

In literature, there are two different approaches to compute the density ρ, that is the *scatter density formulation*, i.e.

$$\rho_i = \sum_{j \in D_i} m_j W_{ij}, \tag{3.52}$$

where the density of a particle is directly computed by summing over its neighbours, and the *continuity density formulation*,

$$\frac{D\rho_i}{Dt} = \sum_{j \in D_i} m_j \left(\mathbf{v}_i - \mathbf{v}_j\right) \nabla_i W_{ij}, \tag{3.53}$$

which is a SPH discretization of the continuity equation. The system is closed by the weakly-compressible state equation (3.4). An artificial viscosity term has to be added to the momentum equation for stability reasons. This treatment adds numerical dissipation to the scheme that is difficult to tune.

Monaghan [77] introduced the *extended SPH* (XSPH) approach where the particles are moved by a smoothed velocity $\mathbf{v}_i^\epsilon$ that depends on the parameter ϵ and replaces the flow velocity $\mathbf{v}_i$ in Eq. (3.50). This modification yields a more regular distribution of the fluid particles.

The standard WCSPH as well as the XSPH approaches are able to capture the position of the free surface quite well but suffer from several weaknesses. In particular, the pressure field often shows non-physical spurious oscillations that pollute the results significantly.

For that reason, several improved SPH schemes have been published. One of them is the δ-SPH approach [4] that adds an artificial diffusion term in the continuity equation to get rid of the non-physical pressure oscillations.

Another weak point of standard WCSPH is the solid boundary treatment. Traditionally the surface integral in Eq. (3.40) is not discretized but is set to zero. In order to compensate for the truncated kernel support, ghost or mirror particles are added that complete the support. The pressure and velocity fields for the added particles are chosen in a way that a solid wall is modelled. This method is accurate for simple geometries but very difficult to implement for complex geometries like they exist in industrial applications. In [78] Monaghan uses repulsive boundary forces that are exerted from added boundary particles on the fluid particles. The boundary forces are given by a Lennard-Jones potential and need a considerable calibration effort before being used. In addition, this method is not accurate enough for our target applications.

In the last years, a new approach for imposing boundary conditions was published, e.g. [67, 22, 30]. Instead of ignoring the surface integral in Eq. (3.40) and instead of completing the kernel support with artificial particles, the surface integral is retained. This enables us to accurately describe the geometrical form of the boundary surface. Some authors [29, 30, 54, 112] compute the surface integral and introduce the $\gamma(\mathbf{x})$ term (3.36) for the discrete standard WCSPH equations to obtain better accuracy close to boundaries and to be able to impose physical boundary conditions.

However, we use another SPH variant throughout this document, which is more accurate and stable. It also allows us to employ an accurate and flexible treatment of boundaries by discretizing the surface integrals and imposing the missing boundary fluxes. The method is called SPH-ALE or Riemann SPH and will be explained in the following section.

3.3 SPH-ALE

In 1999, Vila [111] introduced SPH-ALE (Arbitrary Lagrange Euler) in order to increase accuracy and stability of standard WCSPH. Since then it was used and further developed by several authors (see e.g. [67, 39, 55, 22, 5]) where most of them also show a derivation of the method.

The starting point is the conservative formulation of the Euler equations, written in ALE form, Eq. (3.12). Using a mathematically weak formulation with compact test functions, yields

after doing some maths $\forall i = 0, ..., M-1$:

$$\frac{d(\omega_i \mathbf{\Phi}_i)}{dt} + \omega_i \sum_{j \in D_i} \omega_j \left(\mathbf{F}_i + \mathbf{F}_j\right) \cdot \nabla_i W_{ij} + \omega_i \sum_{j \in \partial D_i} \omega_j^{\partial} \left(\mathbf{F}_i + \mathbf{F}_j\right) \cdot \mathbf{n}_j W_{ij} = \omega_i \mathbf{Q}_i, \tag{3.54}$$

with M denoting the number of particles in the computational domain Ω and $\mathbf{F}_i$ and $\mathbf{F}_j$ denoting the fluxes tensor $\mathbf{F}$ in Eq. (3.14) evaluated at $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. The second sum that is computed for $j \in \partial D_i$ is the boundary term and is zero far away from the boundaries of the domain. A formal demonstration of Eq. (3.54) was obtained by De Leffe [22] using mathematical test functions that are assumed to be compactly supported in space and time. In contrast to [22] we do not compute the γ term of Eq. (3.36).

For simplicity of notation, we will consider in the following the equations without the boundary terms. The boundary terms will be added again when we discuss boundary conditions in SPH-ALE, see Section 3.7.2. Hence, we consider the set of equations,

$$\frac{d(\omega_i \mathbf{\Phi}_i)}{dt} + \omega_i \sum_{j \in D_i} \omega_j \left(\mathbf{F}_i + \mathbf{F}_j\right) \cdot \nabla_i W_{ij} = \omega_i \mathbf{Q}_i. \tag{3.55}$$

This is a conservative formulation because

$$\omega_i \omega_j (\mathbf{F}_i + \mathbf{F}_j) \cdot \nabla_i W_{ij} = -\omega_j \omega_i (\mathbf{F}_j + \mathbf{F}_i) \cdot \nabla_j W_{ij}, \tag{3.56}$$

but it does not conserve mass locally for each particle as it is the case in standard WCSPH. On the contrary, mass fluxes are introduced between particles. That means that particles should be rather considered as moving control volumes than particles, even if we continue to call them particles to underline the meshless character of the method.

It is known from the theory of numerical mathematics of hyperbolic equations that centered discretizations in space of the form of Eq. (3.55) are not stable, see e.g. [58]. A decentered scheme could be obtained by adding an artificial viscosity term like Monaghan did in [78]. However, it is preferred to follow Vila [111] who proposed to replace the term $(\mathbf{F}_i + \mathbf{F}_j)$ by two times a decentered numerical flux computed at $\mathbf{x}_{ij} := \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$, i.e.

$$(\mathbf{F}_i + \mathbf{F}_j) \approx 2\ \hat{\mathbf{F}}_{ij}(\mathbf{\Phi}, \mathbf{v}_0). \tag{3.57}$$

The system of equations (3.55) represents the discretization of the conservation of mass and momentum in ALE form. It has to be completed by the discretization of two additional PDEs for the geometric properties of the particles. The first describes the temporal evolution of the particle positions in ALE form,

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_0(\mathbf{x}_i, t), \tag{3.58}$$

and the second the evolution of the particle volume ω_i,

$$\frac{d\omega_i}{dt} = \omega_i \nabla \cdot (\mathbf{v}_0(\mathbf{x}_i, t)),$$

that is discretized as

$$\frac{d\omega_i}{dt} = \omega_i \sum_{j \in D_i} \omega_j \left(\mathbf{v}_0(\mathbf{x}_j, t) - \mathbf{v}_0(\mathbf{x}_i, t)\right) \cdot \nabla_i W_{ij}. \tag{3.59}$$

Assembling Eqs. (3.58), (3.59), (3.55) and (3.57), we obtain the following semi-discrete SPH-ALE formulation of the Euler equations, $\forall i = 0, ..., M-1$:

$$\begin{cases} \dfrac{d}{dt}(\mathbf{x}_i) = \mathbf{v}_0(\mathbf{x}_i, t), \\ \dfrac{d}{dt}(\omega_i) = \omega_i \sum\limits_{j \in D_i} \omega_j \left(\mathbf{v}_0(\mathbf{x}_j, t) - \mathbf{v}_0(\mathbf{x}_i, t)\right) \cdot \nabla_i W_{ij}, \\ \dfrac{d(\omega_i \mathbf{\Phi}_i)}{dt} + \omega_i \sum\limits_{j \in D_i} \omega_j 2\ \hat{\mathbf{F}}_{ij}(\mathbf{\Phi}, \mathbf{v}_0) \cdot \nabla_i W_{ij} = \omega_i \mathbf{Q}_i. \end{cases} \tag{3.60}$$

Eq. (3.60) is called a semi-discrete equation because the original partial differential equation (PDE) was transformed into an ordinary differential equation (ODE) depending only on time derivatives. The approach was illustrated in Eq. (3.18).

At the beginning of the simulation, at $t = t^0$, initial values

$$\begin{cases} \mathbf{\Phi}_i^0 = \mathbf{\Phi}(\mathbf{x}_i, t^0), \\ \mathbf{v}_{0,i}^0 = \mathbf{v}_0(\mathbf{x}_i, t^0), \\ \omega_i^0 = \omega(\mathbf{x}_i, t^0), \\ \mathbf{x}_i^0 = \mathbf{x}_i(t^0), \end{cases}$$

have to be imposed. The initialization of $\mathbf{\Phi}_i$ and the particle positions $\mathbf{x}_i$ depends on the considered physical problem. The transport velocity $\mathbf{v}_0$ is initialized in Eulerian, Lagrangian or ALE mode. The particle volumes are in general initialized uniformly, i.e. $\omega_i^0 = \omega^0$. For simulations without free surfaces where the whole computational domain is initially filled by water, they are usually determined in a way that the particles discretize the whole computational domain. Hence, we have

$$\sum_{i \in \Omega} \omega_i^0 = M\omega^0 = V_\Omega, \tag{3.61}$$

where V_Ω denotes the total volume that is covered by the computational domain and the initial weights can be easily computed by

$$\omega_i^0 = \omega^0 = \frac{V_\Omega}{M}.$$

If only parts of the computational domain are filled with water and the total volume covered by SPH particles is not known beforehand, the particle volumes are initialised by

$$\omega_i^0 = \omega^0 = (\Delta x)^d.$$

The initial total mass is a result of the prescribed density and the initial particle volumes. It can be computed by summing the mass of the particles, i.e.

$$\sum_{i \in \Omega} m_i = \sum_{i \in \Omega} \omega^0 \rho_i^0.$$

For the computation of the numerical flux $\hat{\mathbf{F}}_{ij}(\mathbf{\Phi}, \mathbf{v}_0)$ Godunov's method and a MUSCL scheme are employed which were originally developed for FV methods. These methods will be introduced in Section 3.6 for SPH-ALE and for the FV method at once.

Geometrical difficulties of SPH

SPH-ALE is a meshless method that does not store or compute any connectivity information between calculation points. Paradoxically, this is the strong point as well as the weak point of the approach. It makes SPH a very flexible method that is able to capture flows with large deformations that a mesh could not support without remeshing. But the lack of connectivity information also has some disadvantages. Assume that the particles are moving in a Lagrangian way. Hence, the evolution of the particle position and the particle volume are described by Eqs. (3.58) and (3.59) but no information is available about the shape of a particle and its size. It is often assumed that a particle is isotropic throughout the whole simulation with a particle size that evolves with the evolution of the volumes, i.e. $\Delta x = \omega^{\frac{1}{d}}$, but experience shows that particles do not always distribute themselves isotropically, e.g. close to stagnation points. In literature, some authors try to cope with this difficulty by introducing anisotropic kernel functions that keep a constant number of neighbours in all directions, see Chapter 4.

Another weak point of the Lagrangian meshless description of SPH is that the total volume is not conserved even if the computational domain is closed. At the beginning of a simulation at $t = t^0$ the particles are usually constructed in a way that (see Eq.(3.61)),

$$\sum_{i \in \Omega} \omega_i^0 = V_\Omega,$$

but after several iterations at $t = t^n$ we obtain

$$\sum_{i \in \Omega} \omega_i^n \neq \sum_{i \in \Omega} \omega_i^0 = V_\Omega.$$

Nevertheless, the total mass is conserved because of Eq. (3.56).

3.4 The Finite Volume method

The Finite Volume method (FVM) is the most widely used method in Computational Fluid Dynamics (CFD) because of its simplicity, flexibility and applicability to a wide range of flow problems. An introduction to Finite Volume discretizations can be found in almost every book about CFD. In the following we use [31, 35, 46, 47, 48, 58, 103, 104].

The derivation of the method starts from a set of conservation laws in a computational domain $\Omega \subset \mathbb{R}^d$ written in integral form, see Eq. (3.15). Again $\mathbf{\Phi} = (\rho, \rho\mathbf{v})^T$ denotes the state vector and $\mathbf{F}$ the flux of Eq. (3.14) through the boundary surface $\partial\Omega$ and $\mathbf{Q}$ the source term. We subdivide the computational domain into N disjoint finite control volumes, also called cells, Ω_l with

$$\bigcup_{l=0}^{N-1} \Omega_l = \Omega.$$

If two cells Ω_{l_1} and Ω_{l_2} ($l_1 \neq l_2$) have a face in common, they are called neighbouring cells. In the case of structured meshes, a cell has two neighbours in 1D, it has four neighbours in 2D and six neighbours in 3D. In this document, a cell-centered FV scheme is considered. Hence in 2D, the mesh lines indicate the faces of the cells and the nodes are the corner points.

For each control volume Ω_l, Eq.(3.15) is valid,

$$\frac{d}{dt} \int_{\Omega_l} \mathbf{\Phi} \, dV + \int_{\partial\Omega_l} \mathbf{F}(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n} dS = \int_{\Omega_l} \mathbf{Q} \, dV.$$

Discretizing the integrals yields the so-called *semi-discrete* flow equations in conservation form for FV schemes $\forall l = 0, ..., N-1$:

$$\frac{d}{dt}(\mathbf{\Phi}_l V_l) + \sum_{faces} \hat{\mathbf{F}}_S(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n}_S = \mathbf{Q}_l V_l, \tag{3.62}$$

where we sum over the flux terms across the faces of cell l. The surface weighted normal vector of a face of area S is denoted by $\mathbf{n}_S := \mathbf{n}S$, where $\|n\| = 1$; V_l is the cell volume. The fluxes evaluated at the faces $\hat{\mathbf{F}}_S$ are called numerical flux vectors.

The vector of state variables $\mathbf{\Phi}_l$ is not associated to a mesh node but is a vector of cell averaged quantities. So $\frac{d}{dt}(\mathbf{\Phi}_l V_l)$ is the time rate of change of the averaged flow variables over the control volume. However, in practice it is often assumed that $\mathbf{\Phi}_l$ represents the value at the center of the control volume.

Even if we use the ALE form of the Euler equations, no mesh deformation is computed in this work and the cell volumes V_l are assumed constant in time. That is why no equation for the evolution of the cell volumes is required.

Due to the hyperbolic character of the Euler equations, centered space discretization schemes are unconditionally unstable. For that reason, the numerical fluxes evaluated on the faces have to be computed by a decentered scheme that will be discused in Section 3.6.

3.5 Similarities of SPH-ALE and FVM

In this chapter the Euler equations in ALE form were discretized by two different approaches, namely the meshbased FVM and the meshless SPH-ALE. It may be surprising that in the end two systems of equations were obtained that resemble each other considerably. Indeed, the similarity of the FV system of semi-discrete equations (Section 3.4) and the one obtained by the SPH-ALE discretization (Section 3.3) is striking and deserves more attention. To analyze the differences and similarities, we compare the SPH-ALE equations of conservation of mass and momentum from Eq. (3.60),

$$\frac{d}{dt}(\omega_i \mathbf{\Phi}_i) + \omega_i \sum_{j \in D_i} \omega_j \, 2 \, \hat{\mathbf{F}}_{ij}(\mathbf{\Phi}, \mathbf{v}_0) \cdot \nabla_i W_{ij} = \omega_i \mathbf{Q}_i, \tag{3.63}$$

to the semi-discrete FV equations, Eq. (3.62),

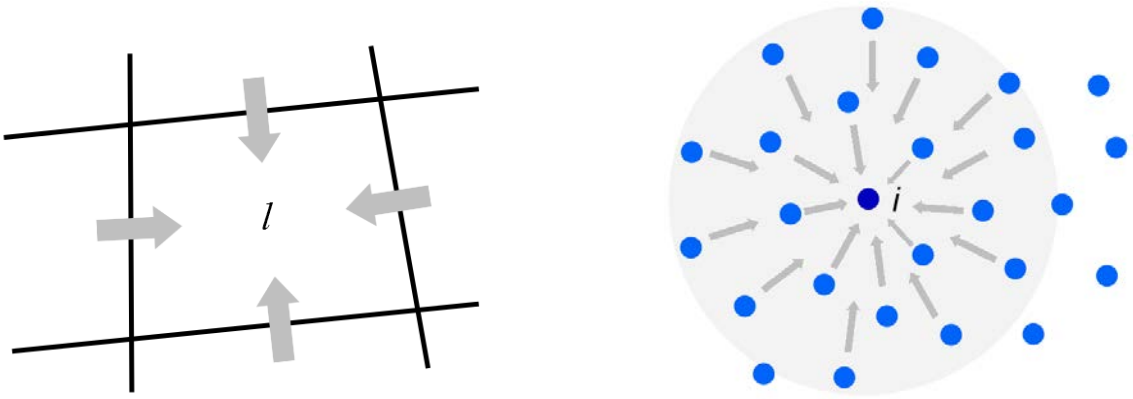
$$\frac{d}{dt}(V_l \mathbf{\Phi}_l) + \sum_{faces} \hat{\mathbf{F}}_S(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n}_S = V_l \, \mathbf{Q}_l.$$

Both equations have the form,

$$\frac{d}{dt}(V\mathbf{\Phi}) + \sum_k \hat{\mathbf{F}}_k(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{A}_k = V\mathbf{Q}. \tag{3.64}$$

The particle weights ω_i are easily identified with the cell volumes V_l, and $\mathbf{A}_k$ is such that $\mathbf{A}_k^{FV} = \mathbf{n}_S = \mathbf{n}S$ and $\mathbf{A}_k^{SPH} = 2\,\omega_i\omega_j\nabla_i W_{ij}$. It is to notice that for FVM the numerical flux vector is evaluated at the face center $\mathbf{x}_S$ and for SPH-ALE at the midpoint between two particles $\mathbf{x}_{ij}$. Moreover, we see that

$$\mathbf{A}_k^{SPH} = 2\,\omega_i\omega_j\|\nabla_i W_{ij}\| \frac{\nabla_i W_{ij}}{\|\nabla_i W_{ij}\|},$$



(a) In FVM fluxes are integrated over the faces of cell l.

(b) In SPH-ALE fluxes are integrated over the kernel support of particle i.

Figure 3.6: The FV discretization and the SPH-ALE discretization of the Euler equations have a similar form. In both methods fluxes are computed from the solution of Riemann problems and then summed.

and using

$$\mathbf{n}_{ij} := \frac{\nabla_i W_{ij}}{\|\nabla_i W_{ij}\|}, \tag{3.65}$$

we obtain

$$\mathbf{A}_k^{SPH} = 2\,\omega_i\omega_j\|\nabla_i W_{ij}\|\mathbf{n}_{ij}.$$

It is noticed that $\mathbf{n}_{ij}$ represents the direction of the connecting line between the cell centers $\mathbf{x}_i$ and $\mathbf{x}_j$.

The $\mathbf{A}_k^{FV}$ term in FVM is the area of the face S multiplied by a normalized direction. Analogously, we can say that the $\mathbf{A}_k^{SPH}$ term is a normalized direction multiplied by $2\omega_i\omega_j\|\nabla_i W_{ij}\|$ which in some way acts as the area of the interface between particles.

Thus, in FVM the $\mathbf{A}_k$ define an area times a direction and they verify

$$\sum_k \mathbf{A}_k^{FV} = \sum_{faces} \mathbf{n}_S = 0.$$

The equivalent statement for SPH-ALE is not true, because

$$\sum_k \mathbf{A}_k^{SPH} = \sum_{j\in D_i} 2\,\omega_i\omega_j\nabla_i W_{ij} = 2\,\omega_i \underbrace{\sum_{j\in D_i} \omega_j\nabla_i W_{ij}}_{\neq 0} \neq 0, \tag{3.66}$$

since the discrete analogon (3.43) of Condition (3.31) is not fulfilled in general.

Summarizing, we see that both methods yield semi-discrete equations that compute the time rate of change of the flow variables averaged over a control volume by summing numerical fluxes over neighbouring cells or particles, see Figure 3.6. However, the way how it is determined if a cell or a particle is a neighbour differs greatly. For FVM, cells are considered as neighbours only if they have a face or parts of a face in common. Fluxes are computed through a known surface that limits one control volume from the other. For SPH-ALE, particles have neighbours that are located within a finite distance to the considered particle i depending on the smoothing

length h. The fluxes are computed between particle i and its neighbours j at the midpoint of the connecting line of the particles i and j. As a consequence, SPH-ALE has a much larger numerical stencil, using about twenty neighbours in 2D, while structured FVM only has four neighbours in 2D. Since the exact geometrical form of a SPH particle is not known, the exact position of the interface between two particles is unknown either, even for two adjacent particles. Furthermore, Eq. (3.66) shows that $\sum_k \mathbf{A}_k$ is not necessarily zero for SPH-ALE. This can be interpreted as using a control volume that is not perfectly closed which has important consequences on the accuracy of the SPH-ALE method. The influence of this error on the consistency of the particle approximation was already noted in Section 3.2.2. A discussion about this error concerning conservation laws and a novel correction method will be given in Section 3.9.3.

3.6 Computation of the numerical fluxes

The semi-discrete SPH-ALE and FV equations (3.60) and (3.62) require the computation of a numerical flux between two particles or between two cells. For both methods Godunov's fluxes are used that will be introduced in the following. For that purpose we consider the system of semi-discrete equations (3.64) of the form,

$$\frac{d}{dt}(V\boldsymbol{\Phi}) + \sum_k \hat{\mathbf{F}}_k(\boldsymbol{\Phi}, \mathbf{v}_0) \cdot \mathbf{A}_k = V\mathbf{Q}, \tag{3.67}$$

with $\mathbf{A}_k = A_k\, \boldsymbol{\nu}_k$ where A_k is an area and $\boldsymbol{\nu}_k$ is a normalized direction.

3.6.1 Godunov's method for the ALE formalism

In 1959 Godunov published a conservative first-order upwind scheme that is applicable to non-linear systems of hyperbolic conservation laws [38] (in Russian). This section gives a short introduction to the method, only highlighting the main steps. The books of Leveque [58] and Toro [103] are recommended to the interested reader. Godunov introduced his method for conservation laws in a fixed frame of reference but the method can directly be extended to conservation laws in ALE form [44]. See also the SPH-ALE literature on moving Riemann problems [111, 67, 55]. In this section, the ALE form of Godunov's method will be presented.

The main idea is to consider the discrete cell values on a mesh as initial data of local one-dimensional Riemann problems. The local Riemann solution is computed in the direction $\boldsymbol{\nu}_k$, that is the normal direction $\mathbf{n}$ to the face between two cells in FVM or the direction $\mathbf{n}_{ij}$ of the connecting line between two particles in SPH-ALE, see Eq. (3.65). It is assumed that the initial discontinuity is situated at a position $\mathbf{x}_k$ that is the face center $\mathbf{x}_S$ in FVM or the midpoint $\mathbf{x}_{ij}$ of the connecting line in SPH-ALE.

We consider a local coordinate system with origin at $\mathbf{x}_k$ and with the $x^{(1)}$-axis parallel to $\boldsymbol{\nu}_k$, our direction of observation. In 2D there is one tangential direction that is parallel to the $x^{(2)}$-axis. In 3D, there are two tangential directions that are parallel to the $x^{(2)}$-axis and the $x^{(3)}$-axis. In the following, only the two-dimensional case is explained but the arguments can be directly extended to 3D. The one-dimensional Riemann problem in direction $\boldsymbol{\nu}_k$ of solution $\boldsymbol{\Phi}(x^{(1)}, t)$, with an initial discontinuity defined at the interface, is given by

$$\begin{cases} \dfrac{\partial}{\partial t}\boldsymbol{\Phi} + \dfrac{\partial}{\partial x^{(1)}}\left(\mathbf{F} \cdot \boldsymbol{\nu}_k\right) = 0, \\ \boldsymbol{\Phi}(x^{(1)}, 0) = \begin{cases} \boldsymbol{\Phi}_L & \text{if } x^{(1)} < 0, \\ \boldsymbol{\Phi}_R & \text{if } x^{(1)} > 0. \end{cases} \end{cases} \tag{3.68}$$

If $\mathbf{F} = \mathbf{F}_c(\mathbf{\Phi})$, the classical Riemann problem in a fixed frame of reference follows. However, if we use the ALE flux of Eq. (3.14), i.e. $\mathbf{F}(\mathbf{\Phi}, \mathbf{v}_0) = \mathbf{F}_c(\mathbf{\Phi}) - \mathbf{v}_0 \otimes \mathbf{\Phi}$, we obtain a moving Riemann problem,

$$\begin{cases} \dfrac{\partial}{\partial t}\mathbf{\Phi} + \dfrac{\partial}{\partial x^{(1)}} \left(\mathbf{F}_c(\mathbf{\Phi}) \cdot \boldsymbol{\nu}_k - \mathbf{v}_0(\mathbf{x}_k, t) \cdot \boldsymbol{\nu}_k \mathbf{\Phi}\right) = 0, \\ \mathbf{\Phi}(x^{(1)}, 0) = \begin{cases} \mathbf{\Phi}_L & \text{if } x^{(1)} < 0, \\ \mathbf{\Phi}_R & \text{if } x^{(1)} > 0. \end{cases} \end{cases} \tag{3.69}$$

In this case, the interface moves with the ALE velocity $\mathbf{v}_0(\mathbf{x}_k, t)$.

The solution of the moving Riemann problem is obtained by using the classical Riemann problem in Eulerian description (denoted by $\mathbf{\Phi}_k^E$) with the following relation,

$$\begin{cases} \lambda_k^0 := \mathbf{v}_0(\mathbf{x}_k, t) \cdot \boldsymbol{\nu}_k, \\ \mathbf{\Phi}_k^E(\lambda_k^0) := \mathbf{\Phi}_k^E(\lambda_k^0, \mathbf{\Phi}_L, \mathbf{\Phi}_R). \end{cases} \tag{3.70}$$

The solution of the moving Riemann problem is then inserted into the flux vector to calculate the numerical flux. Hence,

$$\hat{\mathbf{F}}_k(\mathbf{\Phi}, \mathbf{v}_0) = \mathbf{F}(\mathbf{\Phi}_k^E(\lambda_k^0, \mathbf{\Phi}_L, \mathbf{\Phi}_R)) = \mathbf{F}_c(\mathbf{\Phi}_k^E(\lambda_k^0)) - \mathbf{v}_0(\mathbf{x}_k, t) \otimes \mathbf{\Phi}_k^E(\lambda_k^0), \tag{3.71}$$

where $\mathbf{\Phi}_k^E(0, \mathbf{\Phi}_L, \mathbf{\Phi}_R)$ is the solution of the classical Riemann problem at $\mathbf{x}_k$ in the direction of $x^{(1)}/t = 0$ and $\mathbf{\Phi}_k^E(\lambda_k^0) := (\rho_k^E, \rho_k^E \mathbf{v}_k^E)^T$ is the corresponding solution of the problem in the direction of

$$x^{(1)}/t = \lambda_k^0. \tag{3.72}$$

Hence, the solution $\mathbf{\Phi}_k^E(\lambda_k^0)$ of the moving Riemann problem is obtained in several steps as will be explained in the following.

3.6.1.1 The structure of a classical Riemann problem

The Riemann problem Eq. (3.68) is written in conservative form where the vector $\mathbf{\Phi}$ contains the conserved variables. For the computation of the solution of the Riemann problem we rewrite the equations with $\mathbf{W}$ containing the primitive variables [103]. The new system of equations has the form

$$\frac{\partial}{\partial t}\mathbf{W} + \mathbf{A}(\mathbf{W}) \frac{\partial}{\partial x^{(1)}}\mathbf{W} = 0. \tag{3.73}$$

The vector of primitive variables is defined by $\mathbf{W} := (\rho, v^{(1)}, v^{(2)})^T$ where $v^{(1)}$ denotes the velocity component in direction $\boldsymbol{\nu}_k$ and $v^{(2)}$ denotes the tangential component. Since the weakly-compressible Euler equations Eq. (3.10) are closed by Tait's equation (3.4), the pressure can be expressed as a function of the density. As a consequence, the matrix $\mathbf{A} = \mathbf{A}(\mathbf{W})$ only depends on $\mathbf{W}$ and is given by

$$\mathbf{A} = \begin{pmatrix} v^{(1)} & \rho & 0 \\ \dfrac{c^2}{\rho} & v^{(1)} & 0 \\ 0 & 0 & v^{(1)} \end{pmatrix}. \tag{3.74}$$

The matrix $\mathbf{A}$ has three real distinct eigenvalues, i.e.

$$\begin{cases} \lambda_1 = v^{(1)}, \\ \lambda_2 = v^{(1)} + c, \\ \lambda_3 = v^{(1)} - c. \end{cases} \tag{3.75}$$

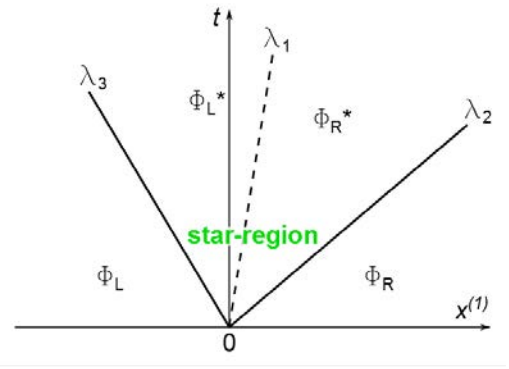


Figure 3.7: The structure of a one-dimensional Riemann problem for the Euler equations along a direction of observation parallel to the $x^{(1)}$-axis.

These eigenvalues determine the general form of the Riemann problem for Euler equations. Figure 3.7 shows the characteristic lines in the $(x^{(1)}, t)$ plane that represent three non-linear waves that are associated to the eigenvalues. The waves separate four different regions of constant states. The central wave that is given by λ_1 is known to be a contact discontinuity. However, the other two waves can in each case either be a shock wave or a rarefaction wave. The region which lies in-between the waves associated to λ_2 and λ_3 is called *star-region* where the state variables are equal to the unknown state $\mathbf{W}^*$. The star-region is divided by the contact discontinuity into a left and a right star-region with states $\mathbf{W}_L^*$ and $\mathbf{W}_R^*$ where only the tangential velocity is discontinuous. Since the tangential velocity does not change through a shock or a rarefaction wave, we obtain

$$\begin{cases} \rho^* = \rho_L^* = \rho_R^*, \\ (v^{(1)})^* = (v^{(1)})_L^* = (v^{(1)})_R^*, \\ (v^{(2)})_L^* = v_L^{(2)}, \\ (v^{(2)})_R^* = v_R^{(2)}. \end{cases} \tag{3.76}$$

In order to get the whole structure of the local Riemann problem, the missing state $(\rho^*, (v^{(1)})^*)$ in the star-region has to be computed. Moreover, the character of the left and the right wave has to be determined and its' speeds to be computed.

Computation of the states in the star-region

The state in the star-region $\mathbf{W}^*$ is reached from the left and the right initial states, $\mathbf{W}_L$ and $\mathbf{W}_R$, through non-linear waves that can be either shock waves or rarefaction waves. A non-linear expression for ρ^* can be found that is based on the Rankine-Hugoniot relation for shocks or on the conservation of Riemann invariants for rarefaction waves [103]. It has the form

$$g(\rho^*) := g_L(\rho^*, \mathbf{W}_L) + g_R(\rho^*, \mathbf{W}_R) + v_R^{(1)} - v_L^{(1)} = 0, \tag{3.77}$$

where g_L and g_R are non-linear functions. Eq. (3.77) has to be solved iteratively, see [67] for details. From the solution of the density ρ^* the normal component of the velocity in the star-region $(v^{(1)})^*$ can be computed by

$$(v^{(1)})^* = \frac{1}{2}\left(v_R^{(1)} + v_L^{(1)}\right) + \frac{1}{2}\left(g_R(\rho^*, \mathbf{W}_R) - g_L(\rho^*, \mathbf{W}_L)\right).$$

It is computationally expensive and for our purpose not necessary to compute $\mathbf{W}^*$ by an exact iterative solver. Instead we prefer to use linearized Riemann solvers for the computation of the numerical fluxes. The fact that the numerical methods compute averaged values that are supposed to be constant or linear on a control volume introduces more errors than an approximate Riemann solver does. Furthermore, in hydraulic applications we do not encounter strong shocks that have to be captured. Validations and comparison of exact and different approximate Riemann solvers can be found in [67, 69]. In the simulations presented in this thesis the approximate *Primitive Variable Riemann Solver* (PVRS) is used exclusively.

The PVRS solver [103, 67, 55] is a linearized Riemann solver where the matrix $\mathbf{A}(\mathbf{W})$ of Eq. (3.73) is approximated by a constant matrix $\overline{\mathbf{A}} := \mathbf{A}(\overline{\mathbf{W}})$, with $\overline{\mathbf{W}}$ denoting a constant averaged state. Solving exactly the Riemann problem of a linear hyperbolic system

$$\frac{\partial}{\partial t}\mathbf{W} + \overline{\mathbf{A}}\,\frac{\partial}{\partial x^{(1)}}\mathbf{W} = 0,$$

we obtain simple relations for ρ^* and $(v^{(1)})^*$, i.e.

$$\begin{cases} \rho^* = \dfrac{1}{2}\left(\rho_L + \rho_R\right) - \dfrac{1}{2}\left(v_R^{(1)} - v_L^{(1)}\right)\dfrac{\bar{\rho}}{\bar{c}}, \\ (v^{(1)})^* = \dfrac{1}{2}\left(v_L^{(1)} + v_R^{(1)}\right) - \dfrac{1}{2}\left(\rho_R - \rho_L\right)\dfrac{\bar{c}}{\bar{\rho}}, \end{cases} \tag{3.78}$$

with

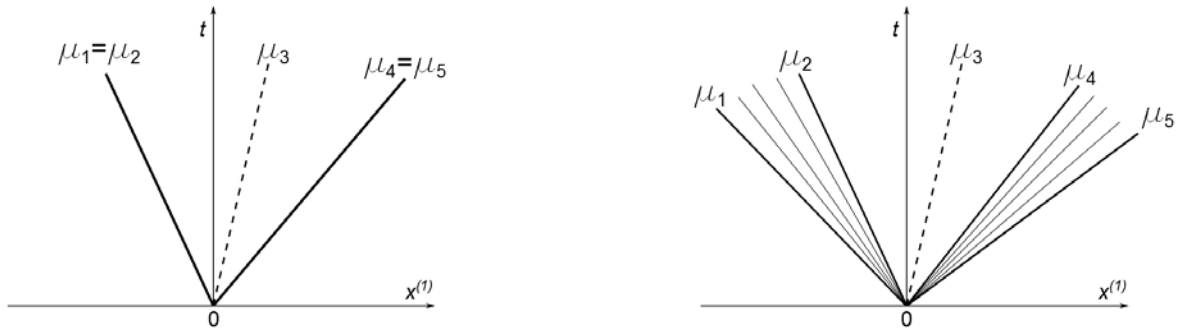
$$\begin{cases} \bar{\rho} = \dfrac{1}{2}\left(\rho_L + \rho_R\right), \\ \bar{c} = \dfrac{1}{2}\left(c_L + c_R\right). \end{cases}$$

The local sound speeds c_L, c_R and c^* are determined by Eq. (3.5). These intermediate states are at first used to determine if the left and right running waves are either shock waves or rarefaction waves. In a next step, they are utilized to get the sought solution that is required for the computation of the numerical fluxes Eq. (3.71).

Determining the nature of the left and the right wave

There are four different configurations of possible combinations of shock and rarefaction waves at each side of the contact discontinuity, see Figures 3.8 and 3.9. If both waves are shocks, there are four regions where the state variables have different values. These regions are delimited by the lines $x^{(1)}/t = \mu_1 = \mu_2$, $x^{(1)}/t = \mu_3$ and $x^{(1)}/t = \mu_4 = \mu_5$. If both waves are rarefaction waves two additional regions are added, one between μ_1 and μ_2 and one between μ_4 and μ_5. In the case of a shock and a rarefaction wave five different regions exist. In the following, we determine the $\mu_\alpha,\ \alpha = 1, ..., 5$.

- The left wave is a rarefaction wave if $\rho^* \leq \rho_L$. This is shown in Figure 3.8(b) and 3.9(a) and we have
$$\begin{cases} \mu_1 = v_L^{(1)} - c_L, \\ \mu_2 = (v^{(1)})^* - c^*. \end{cases}$$
- The left wave is a shock wave if $\rho^* > \rho_L$, see Figure 3.8(a) and 3.9(b). In this case we have
$$\mu_1 = \mu_2 = v_L^{(1)} - \frac{Q_L}{\rho_L},$$



(a) Two shock waves.

(b) Two rarefaction waves.

Figure 3.8: Possible configurations of the Riemann problem where the left and the right waves are the same.

with

$$Q_L := \left(\frac{K[(\rho^*)^\gamma - \rho_L^\gamma]\rho_L\rho^*}{\rho^* - \rho_L} \right)^{\frac{1}{2}},$$

and with

$$K := \frac{c_0^2}{\gamma \rho_0^{\gamma-1}}, \tag{3.79}$$

where γ, c_0, ρ_0 are defined as in Section 3.1.

- We know that the wave associated to λ_1 is a contact continuity with

$$\mu_3 = (v^{(1)})^*.$$

- The right wave is a rarefaction wave if $\rho^* \leq \rho_R$, see Figure 3.8(b) and 3.9(b), and we obtain analogously to the left rarefaction wave

$$\begin{cases} \mu_4 = (v^{(1)})^* - c^*, \\ \mu_5 = v_R^{(1)} - c_R. \end{cases}$$

- The right wave is a shock wave if $\rho^* > \rho_R$, see Figure 3.8(a) and 3.9(a). Similarly to the left shock wave, we have

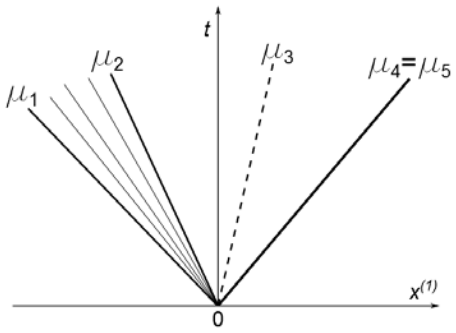
$$\mu_4 = \mu_5 = v_R^{(1)} + \frac{Q_R}{\rho_R},$$

with

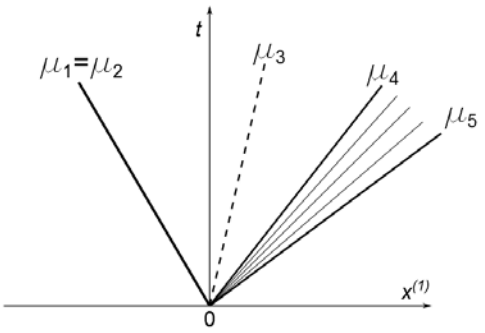
$$Q_R := \left(\frac{K[(\rho^*)^\gamma - \rho_R^\gamma]\rho_R\rho^*}{\rho^* - \rho_R} \right)^{\frac{1}{2}},$$

and K defined by Eq. (3.79).

Knowing $\mathbf{W}^*$ from Eq. (3.78) and the nature of the left and right waves, the solution state $\mathbf{\Phi}_k^E(\lambda_k^0)$ that we need for the computation of the numerical fluxes will be sought in the direction given by Eq. (3.72).



(a) A rarefaction wave at the left hand side and a shock wave at the right hand side.



(b) A shock wave at the left hand side and a rarefaction wave at the right hand side.

Figure 3.9: Possible configurations of the Riemann problem where the left and the right waves are different.

3.6.1.2 Computation of the solution state for the numerical fluxes

The next step is to determine in which region the sought solution $\mathbf{\Phi}_k^E(\lambda_k^0)$ lies. We simply check between which $\mu_\alpha,\ \alpha = 1, ..., 5$ the line

$$x^{(1)}/t = \lambda_k^0 = \mathbf{v}_0(\mathbf{x}_k, t) \cdot \boldsymbol{\nu}_k$$

is situated. This is the first time that considering a moving Riemann problem instead of a classical Riemann problem in a fixed frame of reference intervenes in our considerations. In the case of the classical Godunov method the solution is sought in the direction $x^{(1)}/t = 0$ which corresponds to the t-axis in the $(x^{(1)}, t)$-diagram.

1. If $\lambda_k^0 \leq \mu_1$, the solution state is equal to the left initial state, i.e. $\mathbf{\Phi}_k^E(\lambda_k^0) = \mathbf{\Phi}_L$.

2. If $\mu_1 < \lambda_k^0 \leq \mu_2$, the solution is situated inside the left rarefaction wave. In this case, the flow variables in the local coordinate system are given by

$$\begin{cases} (v^{(1)})_k^E = \dfrac{2}{\gamma+1}\left(c_L + \dfrac{1}{2}v_L^{(1)}(\gamma-1) + \lambda_k^0\right), \\ \rho_k^E = \left(\dfrac{((v^{(1)})_k^E - \lambda_k^0)^2}{\gamma\ K}\right)^{\frac{1}{\gamma-1}}, \\ (v^{(2)})_k^E = v_L^{(2)}. \end{cases}$$

3. If $\mu_2 < \lambda_k^0 \leq \mu_3$, the solution state $\mathbf{\Phi}_k^E(\lambda_k^0)$ is equal to $\mathbf{\Phi}_L^*$. This means that

$$\begin{cases} \rho_k^E = \rho^*, \\ (v^{(1)})_k^E = (v^{(1)})^*, \\ (v^{(2)})_k^E = v_L^{(2)}. \end{cases}$$

4. If $\mu_3 < \lambda_k^0 \leq \mu_4$, the solution state is given by $\mathbf{\Phi}_k^E(\lambda_k^0) = \mathbf{\Phi}_R^*$, i.e.

$$\begin{cases} \rho_k^E = \rho^*, \\ (v^{(1)})_k^E = (v^{(1)})^*, \\ (v^{(2)})_k^E = v_R^{(2)}. \end{cases}$$

5. If $\mu_4 < \lambda_k^0 \leq \mu_5$, the solution is situated inside the right rarefaction wave. In this case, the flow variables are given by

$$\begin{cases} (v^{(1)})_k^E = \dfrac{2}{\gamma+1}\left(-c_R + \dfrac{1}{2}v_R^{(1)}(\gamma-1) + \lambda_k^0\right), \\ \rho_k^E = \left(\dfrac{(\lambda_k^0 - (v^{(1)})_k^E)^2}{\gamma\ K}\right)^{\frac{1}{\gamma-1}}, \\ (v^{(2)})_k^E = v_R^{(2)}. \end{cases}$$

6. If $\lambda_k^0 > \mu_5$, the solution state is equal to the right initial state, i.e. $\mathbf{\Phi}_k^E(\lambda_k^0) = \mathbf{\Phi}_R$.

Summarizing, several steps are necessary to obtain the solutions that are used to compute the numerical fluxes. At first the classical Riemann problem, Eq. (3.68) is written in primitive variables to obtain the eigenvalues of the hyperbolic system. Then we compute the variables in the star-region in order to determine the whole structure of the Riemann problem. After detecting if the left and right waves are shock or rarefaction waves, the required solution is obtained by searching in the direction of Eq. (3.72).

3.6.2 The semi-discrete SPH-ALE and FV equations

The computed solution states $\mathbf{\Phi}_k^E(\lambda_k^0)$ are used to get the numerical fluxes for the systems of semi-discrete equations (3.60) and (3.62). Hence, the resulting system of semi-discrete SPH-ALE equations is

$$\begin{cases} \dfrac{d}{dt}(\mathbf{x}_i) = \mathbf{v}_0(\mathbf{x}_i, t), \\ \dfrac{d}{dt}(\omega_i) = \omega_i \displaystyle\sum_{j \in D_i} \omega_j \left(\mathbf{v}_0(\mathbf{x}_j, t) - \mathbf{v}_0(\mathbf{x}_i, t)\right) \cdot \nabla_i W_{ij}, \\ \dfrac{d}{dt}(\omega_i \rho_i) + \omega_i \displaystyle\sum_{j \in D_i} \omega_j 2\rho_{ij}^E \left(\mathbf{v}_{ij}^E - \mathbf{v}_0(\mathbf{x}_{ij}, t)\right) \cdot \nabla_i W_{ij} = 0, \\ \dfrac{d}{dt}(\omega_i \rho_i \mathbf{v}_i) + \omega_i \displaystyle\sum_{j \in D_i} \omega_j 2\left[\rho_{ij}^E \mathbf{v}_{ij}^E \otimes (\mathbf{v}_{ij}^E - \mathbf{v}_0(\mathbf{x}_{ij}, t)) + p_{ij}^E \mathbf{I}\right] \cdot \nabla_i W_{ij} = \omega_i \rho_i \mathbf{g}, \end{cases} \tag{3.80}$$

and analogously the system of semi-discrete FV equations in conservative ALE form reads as,

$$\begin{cases} \dfrac{d}{dt}(\rho_l V_l) + \displaystyle\sum_{faces} \rho_S^E \left(\mathbf{v}_S^E - \mathbf{v}_0(\mathbf{x}_S, t)\right) \cdot \mathbf{n}_S = 0, \\ \dfrac{d}{dt}(\rho_l \mathbf{v}_l V_l) + \displaystyle\sum_{faces} \left[\rho_S^E \mathbf{v}_S^E \otimes \left(\mathbf{v}_S^E - \mathbf{v}_0(\mathbf{x}_S, t)\right) + p_S^E \mathbf{I}\right] \cdot \mathbf{n}_S = V_l \rho_l \mathbf{g}. \end{cases} \tag{3.81}$$

The weakly-compressible state equation (3.4) is used to close both systems. Note that the standard Godunov equations for FVM in Eulerian description are recovered if $\mathbf{v}_0 = 0$ is chosen in Eq. (3.81).

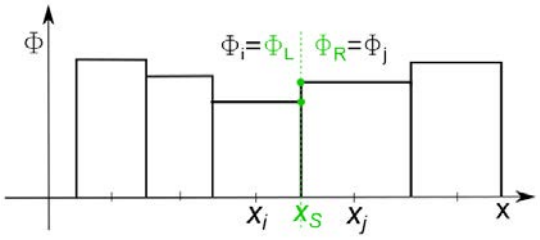


Figure 3.10: Initial states $\mathbf{\Phi}_L$ and $\mathbf{\Phi}_R$ at the interface between two cells i and j for the original Godunov method in the case of a 1D FV solver. Originally in Godunov's method the states were assumed to be constant in a cell with $\mathbf{\Phi}_L = \mathbf{\Phi}_i$ and $\mathbf{\Phi}_R = \mathbf{\Phi}_j$.

3.6.3 The MUSCL scheme

In the previous section 3.6.1 Godunov's method was described without specifying how the initial states $\mathbf{\Phi}_L$ and $\mathbf{\Phi}_R$ at the interface of the Riemann problem Eq. (3.69) are chosen. The original Godunov scheme is a first order conservative upwind scheme where the initial states $\mathbf{\Phi}_L$ and $\mathbf{\Phi}_R$ are taken equal to the states $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_j$ of the considered cells or particles. That means that it is supposed that the states are constant in each control volume. Figure 3.10 illustrates the concept for one space dimension and FVM. In 1979, Van Leer [110] published his second-order sequel of the Godunov method where the states $\mathbf{\Phi}_L$ and $\mathbf{\Phi}_R$ are supposed to be linear inside a control volume. Hence, they are reconstructed from the center of the control volume to the interfaces by linear interpolation. In the original paper the method is called *Monotonic Upstream-centered Scheme for Conservation Laws* (MUSCL).

3.6.3.1 The MUSCL approach adapted to unstructured FVM

We use a version of the method for unstructured grids, see [26]. We consider a face with center $\mathbf{x}_S$ that connects cells i and j with cell centers $\mathbf{x}_i$ and $\mathbf{x}_j$.

The idea of the MUSCL scheme is to interpolate the interface states by an expression that ideally would be

$$\begin{cases} \mathbf{\Phi}_L = \mathbf{\Phi}_i + \nabla_i \mathbf{\Phi} \cdot (\hat{\mathbf{x}}_\vartheta - \mathbf{x}_i), \\ \mathbf{\Phi}_R = \mathbf{\Phi}_j + \nabla_j \mathbf{\Phi} \cdot (\hat{\mathbf{x}}_\vartheta - \mathbf{x}_j), \end{cases} \tag{3.82}$$

where $\nabla_i \mathbf{\Phi}$ denotes the gradient of $\mathbf{\Phi}$ evaluated at the cell center $\mathbf{x}_i$. The $\hat{\mathbf{x}}_\vartheta$ is the point where the connecting line $\overline{\mathbf{x}_i \mathbf{x}_j}$ intersects the face. In general, it is not equal to the face center $\mathbf{x}_S$. Hence, there is a parameter $\vartheta_S \in [0, 1]$, such that

$$\hat{\mathbf{x}}_\vartheta = (1 - \vartheta_S)\mathbf{x}_i + \vartheta_S \mathbf{x}_j, \tag{3.83}$$

where the parameter is computed by

$$\vartheta_S := \frac{\|\mathbf{x}_S - \mathbf{x}_i\|}{\|\mathbf{x}_j - \mathbf{x}_i\|}. \tag{3.84}$$

But expression (3.82) for the interface states cannot be used in practice and a non-linear limiter function α is needed to prevent spurious oscillations to arise. A famous limiter is the *minmod* function [58]. Thus, we obtain with $\alpha \in [0, 1]$,

$$\begin{cases} \mathbf{\Phi}_L = \mathbf{\Phi}_i + \alpha(\mathbf{\Phi}_i, \mathbf{\Phi}_j, \nabla_i \mathbf{\Phi}) \nabla_i \mathbf{\Phi} \cdot (\hat{\mathbf{x}}_\vartheta - \mathbf{x}_i), \\ \mathbf{\Phi}_R = \mathbf{\Phi}_j + \alpha(\mathbf{\Phi}_i, \mathbf{\Phi}_j, \nabla_j \mathbf{\Phi}) \nabla_j \mathbf{\Phi} \cdot (\hat{\mathbf{x}}_\vartheta - \mathbf{x}_j). \end{cases} \tag{3.85}$$

Hence, the reconstruction of the interface states demands the computation of the gradients of the state variables that are not needed for the Godunov scheme. In unstructured FVM the gradients of a variable ϕ can be computed as

$$\nabla_i \phi = \frac{1}{V_i} \sum_{faces} \phi_{i,S} \, \mathbf{n}_S, \tag{3.86}$$

where

$$\phi_{i,S} := (1 - \vartheta_S)\phi_i + \vartheta_S \, \phi_j,$$

see [26, 31]

3.6.3.2 The MUSCL approach adapted to SPH-ALE

The same way, to improve the accuracy of the SPH-ALE scheme, the MUSCL scheme was adapted to SPH-ALE by [111, 67]. In contrast to FVM, the exact position of the interface between two particles is not known but it is assumed that it is situated at the midpoint between them. So we obtain

$$\begin{cases} \mathbf{\Phi}_L = \mathbf{\Phi}_i + \alpha(\mathbf{\Phi}_i, \mathbf{\Phi}_j, \nabla_i \mathbf{\Phi}) \nabla_i \mathbf{\Phi} \cdot \dfrac{\mathbf{x}_j - \mathbf{x}_i}{2}, \\ \mathbf{\Phi}_R = \mathbf{\Phi}_j - \alpha(\mathbf{\Phi}_i, \mathbf{\Phi}_j, \nabla_j \mathbf{\Phi}) \nabla_j \mathbf{\Phi} \cdot \dfrac{\mathbf{x}_j - \mathbf{x}_i}{2}, \end{cases} \tag{3.87}$$

where α again denotes a limiter function. The gradients $\nabla_i \mathbf{\Phi}$ are computed by the SPH estimation, Eq. (3.44).

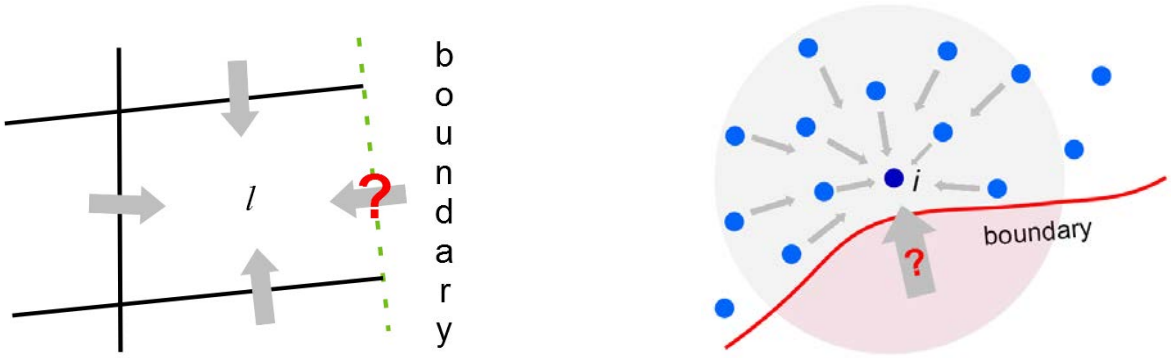
3.7 Boundary conditions

In practical computations of flows, a finite computational domain Ω with boundary $\partial\Omega$ is considered. At the boundary, conditions have to be imposed according to several considerations. First, the numerical stencil is not complete close to boundaries and a way has to be found such as the missing calculation points are compensated for. Second, we want to impose physical conditions on the boundary that depend on the boundary type, i.e. wall, inlet or outlet, periodicity or free surface. Third, the physical conditions have to be in accordance with the hyperbolic character of the Euler equations. Solutions of the Euler equations describe convective phenomena and have a wave-like character. The boundary conditions have to be compatible with the direction of propagation of the waves. The first aspect, the truncation of the numerical stencil, depends on the numerical method that is used, in our case, SPH-ALE or FVM. The other two aspects, the physical and mathematical constraints, are independent of the employed method.

In this section, boundary treatment for SPH-ALE and FVM for different types of boundaries will be explained with a special emphasis on the similarities and differences between the two methods.

3.7.1 Boundary conditions for FVM

The FV method computes the fluxes between two cells that are connected by a face. In the interior of the domain the fluxes are calculated from the states of both neighbouring cells. At the boundary there is a face with only one neighbour cell and a way has to be found to compute the missing flux, see Figure 3.11 (a). Basically there are two strategies, either we add artificial cells to complete the numerical stencil or we directly compute the boundary flux. Each of them has advantages and disadvantages. The first necessitates the construction of artificial cells where



(a) In FVM fluxes are integrated over the faces of cell l.

(b) In SPH-ALE fluxes are integrated over the kernel support of particle i.

Figure 3.11: The fluxes between two cells or two particles are determined by solving Riemann problems. However, at the boundaries of the computational domain it is not obvious how to compute the fluxes that are indicated by a red question mark.

the field variables are set in a way that the physical boundary conditions are satisfied. The second one demands a modification of the numerical stencil at the boundary. Furthermore, the boundary flux has to be determined in such a way that the hyperbolic character of the equations is not violated. We will use both strategies depending on the type of boundaries that is considered.

If the numerical stencil is not complete, the semi-discrete FV equations (3.81) for a cell l at the boundary become

$$\frac{d(V_l \mathbf{\Phi}_l)}{dt} + \sum_{IF} \hat{\mathbf{F}}_S(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n}_S + \sum_{BF} \hat{\mathbf{F}}_S(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n}_S = V_l \mathbf{Q}_l, \tag{3.88}$$

where the sum over IF is over the internal faces and the sum over BF is over the boundary faces. The boundary faces represent a discretization of the boundary with a surface area S and a normalized direction $\mathbf{n}$ that is normal to the surface element.

In this project, the FV domain can be limited by wall boundaries, inlet and outlet boundaries and periodic boundaries. However, the implemented FV solver does not treat free surfaces or multi-phase flows.

3.7.2 Boundary conditions in SPH-ALE

A particle that is situated close to the boundary of the computational domain has a kernel support that is truncated like it is illustrated in Figure 3.11 (b). In general, the surface integral in the kernel approximation of gradients Eq. (3.29) is not zero at the boundaries. In literature, two strategies can be found for boundary treatment. Either the truncated kernel is completed by adding particles, similar to the virtual cells in FVM, or the surface integral is directly computed. In this document, we complete the kernel support by adding virtual particles only in order to impose periodic boundary and plane symmetry conditions. In all other cases, we prefer to compute the boundary integral. For that purpose, the boundaries are discretized in surface elements j with normal vector $\mathbf{n}_j$ and integration weight ω_j^∂. Note that ω_j^∂ is equal to the area of the surface element j. The semi-discrete SPH-ALE equations, Eq. (3.60), with boundary

terms have the form,

$$\frac{d(\omega_i \mathbf{\Phi}_i)}{dt} + \omega_i \sum_{j \in D_i} \omega_j \, 2 \, \hat{\mathbf{F}}_{ij}(\mathbf{\Phi}, \mathbf{v}_0) \cdot \nabla_i W_{ij} + \omega_i \sum_{j \in \partial D_i} \omega_j^{\partial} \, 2 \, \hat{\mathbf{F}}_{ij}(\mathbf{\Phi}, \mathbf{v}_0) \cdot \mathbf{n}_j W_{ij} = \omega_i \mathbf{Q}_i. \quad (3.89)$$

The boundary flux has to be determined in a way that satisfies the physical conditions that we want to impose and that respects the hyperbolic nature of the system. Analogously to FVM, different strategies are applied, depending on the type of the boundary, i.e. wall, inlet or outlet.

In the following, similar to Section 3.6 we use the index k to describe the boundary contributions that appear for the SPH-ALE or the FV equations. Hence, the index k denotes the index of the boundary faces if FVM is considered and the index of the surface elements that discretize the boundary of the kernel support ∂D_i in the case of SPH-ALE. The direction of observation is in both cases the normalized normal $\mathbf{n}_k$ of the boundary face or surface element.

3.7.3 Solid wall boundary conditions

In this document we want to impose inviscid wall boundary conditions, so-called *free slip* conditions, that are translated into the equation

$$(\mathbf{v}(\mathbf{x}_k, t) - \mathbf{v}_w(\mathbf{x}_k, t)) \cdot \mathbf{n}_k = 0, \quad (3.90)$$

with $\mathbf{x}_k$ denoting a point at the wall and $\mathbf{v}_w$ the velocity of the solid. For our target applications we need a way to impose wall boundary conditions on complex geometries in an accurate way. The approach that we choose is closely linked to our experience with SPH-ALE where partial Riemann problems on surface elements are solved. For SPH-ALE the technique is based on the work of Marongiu [67]. Originally, the notion of partial Riemann problems was developed for unstructured FVM by Dubois [26] and can be directly applied to the FV solver that was implemented in the frame of this thesis.

In Section 3.6.1 we have seen that Godunov's method reinterprets faces as interfaces between discontinuous states that are initial conditions for Riemann problems. The idea of Dubois [26] is to stick to that interpretation, considering Riemann problems also on boundaries, but with some necessary modifications. Again we consider a Riemann problem like in Eq. (3.69) but this time the interface is situated at the solid wall at $\mathbf{x}_k$. At the boundary, one initial state is missing because there is no control volume on the other side of the face. We assume that the missing state that comes from the boundary is $\mathbf{\Phi}_R$ and that $\mathbf{\Phi}_L$ denotes the fluid state. The missing information is replaced by the information that comes from the physical boundary condition, Eq. (3.90). Since the velocity of the interface is equal to the velocity of the solid, i.e. $\mathbf{v}_0(\mathbf{x}_k, t) = \mathbf{v}_w(\mathbf{x}_k, t)$, Eq. (3.90) gives

$$\left(\mathbf{v}_k^E - \mathbf{v}_0(\mathbf{x}_k, t)\right) \cdot \mathbf{n}_k = 0.$$

As a consequence, the convective part of the flux cancels and the formulation guarantees that no mass is transfered between the solid wall and the fluid. The numerical boundary flux in Eq. (3.80) reduces to

$$\hat{\mathbf{F}}_k = (0, p_k^E \mathbf{I})^T.$$

According to [26] it is an acceptable simplification to seek the solution state p_k^E in the star region. Applying a linearization around the interface yields for the density

$$\rho_k^E = \rho^* = \rho_L - (\mathbf{v}_0 - \mathbf{v}_L) \cdot \mathbf{n}_k \frac{\rho_L}{c_L},$$

and the pressure p_k^E is again obtained by Eq. (3.4).

Hence, the ALE formalism enables us to consider solids that are moving with an arbitrary velocity $\mathbf{v}_w$ that can be predefined or imposed by the fluid [96]. This property is of paramount importance for the applications of flows in hydraulic machines that we are interested in. Note that in the frame of this project, we do not use solid walls that move relatively to the border of the FV domain because we do not consider changes in the connectivity of the FV mesh. In Chapter 5 it will be shown that the SPH domain is used to handle this feature.

3.7.4 Inlet and outlet conditions

It is of crucial importance to impose inlet and outlet conditions correctly. Since we are considering hydraulic applications, subsonic conditions are specified because the fluid velocity is smaller than the speed of sound, $\|\mathbf{v}\| < c$. For simplicity, we discuss the two-dimensional case with $\mathbf{v}_0 = 0$ and assume that the normal vectors $\mathbf{n}_k$ at the inlet and at the outlet are pointing inside the fluid domain. At the inlet, $\mathbf{v} \cdot \mathbf{n}_k > 0$, so the eigenvalues from Eq. (3.75) fulfil

$$\lambda_3 < 0 < \lambda_1 < \lambda_2,$$

with

$$\begin{cases} \lambda_1 = \mathbf{v} \cdot \mathbf{n}_k, \\ \lambda_2 = \mathbf{v} \cdot \mathbf{n}_k + c, \\ \lambda_3 = \mathbf{v} \cdot \mathbf{n}_k - c. \end{cases}$$

Hence, there are two positive eigenvalues that correspond to information that comes from the outside of the domain and one negative eigenvalue associated to information leaving the domain. At the outlet $\mathbf{v} \cdot \mathbf{n}_k < 0$ and we obtain

$$\lambda_3 < \lambda_1 < 0 < \lambda_2,$$

which means that there is one positive and two negative eigenvalues corresponding to one information from outside and two from inside. As a consequence, there are two components of field variables $\mathbf{\Phi}$ that have to be imposed at the inlet and one at the outlet. In many cases it is a good choice to impose velocity (two components in 2D, and three in 3D) at the inlet and static pressure at the outlet.

Launching simulations of internal flows with the weakly-compressible formalism shows reflections at the inlet and the outlet that pollute the results if no special treatment is applied. A first attempt to get rid of the reflections was made by implementing non-reflecting boundary conditions at infinity [36] that were first adapted to SPH-ALE by [22]. Boundary conditions at infinity suppose that a state at infinity $\mathbf{\Phi}_\infty$, also called a free-stream state, is imposed and that this state is not influenced by the flow inside the computational domain. In this approach a numerical flux $\hat{\mathbf{F}}$ at the boundary is defined by

$$\begin{aligned} \hat{\mathbf{F}} := & \sum_{\alpha:\lambda_\alpha \geq 0} \mathbf{L}_\alpha(\mathbf{\Phi}_\infty, \mathbf{n}_k) \cdot [(\mathbf{F}(\mathbf{\Phi}_\infty) \cdot \mathbf{n}_k)\mathbf{R}_\alpha(\mathbf{\Phi}_\infty, \mathbf{n}_k)] \\ & + \sum_{\alpha:\lambda_\alpha < 0} \mathbf{L}_\alpha(\mathbf{\Phi}_\infty, \mathbf{n}_k) \cdot [(\mathbf{F}(\mathbf{\Phi}_i) \cdot \mathbf{n}_k)\mathbf{R}_\alpha(\mathbf{\Phi}_\infty, \mathbf{n}_k)], \end{aligned} \tag{3.91}$$

where $\mathbf{L}_\alpha$ and $\mathbf{R}_\alpha$ are the left and right eigenvectors of the matrix $\mathbf{A}$ in Eq. (3.74) associated to λ_α with $\alpha = 1, 2, 3$. The eigenvectors are evaluated at the imposed state $\mathbf{\Phi}_\infty$. Analogously, $\mathbf{F}(\mathbf{\Phi}_\infty)$ denotes the flux tensor given by Eq. (3.14) evaluated at the imposed state $\mathbf{\Phi}_\infty$. The

flux tensor $\mathbf{F}(\mathbf{\Phi}_i)$ is evaluated at $\mathbf{\Phi}_i$ which is the state of the particle or the cell i that is the neighbour of the surface element k. Note that the first term is a sum over the eigenvectors that are associated to the positive eigenvalues, i.e. the eigenvectors that correspond to the information that is propagating into the domain and that has to be imposed. The other term corresponds to the information that leaves the domain. Thus, the hyperbolic characteristics of the Euler equations are well respected.

These boundary conditions are very effective in eliminating the reflected waves but do not allow us to impose the above introduced subsonic conditions. They were developed for simulations of unbounded physical domains where the states at infinity are known and where the imposed velocity and pressure at inlet and outlet are the same. This is not the case in the internal flow applications that are our target applications where typically a pressure gradient between inlet and outlet is established. Nevertheless, these conditions allow gradients of the flow variables between inlet and outlet but the subsonic conditions cannot be imposed explictly, especially the pressure at the outlet cannot be controlled directly.

For that reason, another approach of non-reflecting boundary conditions called *Navier-Stokes Characteristic Boundary Conditions* based on the publications [82, 90] is available in our solver. The idea is to write the conservative Euler equations in the direction normal to the inlet or outlet. The conservative variables are transformed to primitive and then to characteristic variables which are used to obtain the compatibility relations that allow us to identify the ingoing and outgoing waves. This gives a relation between the primitive and the characteristic variables that is used to understand why imposing pressure at a subsonic outlet introduces necessarily reflections. In the end, a relaxed boundary value is used that is able to evacuate the unwanted perturbations and that maintains a target value that we want to impose, e.g. the pressure at the outlet. The difficulty is that a relaxation coefficient has to be determined depending on geometrical properties of the computational domain.

Difficulties at inlets and outlets with SPH-ALE

When imposing inlet and outlet conditions to SPH particles in Eulerian description the boundary treatment simply consists of inserting the computed boundary fluxes, e.g. Eq. (3.91), in the semi-discrete system (3.89) like it is done for FVM. However, when condering moving SPH particles additional difficulties arise that are due to geometrical problems. Moving particles have to be created when they enter the computational domain at the inlet and they have to be destroyed when they leave the domain at the outlet. Special care has to be taken to insert or remove particles without introducing perturbations into the flow fields. At the outlet, it is especially tricky to impose pressure correctly before deleting the particles that leave the computational domain. Whenever it is possible, we use particles in Eulerian motion to avoid this problem.

3.7.5 Periodic boundary conditions

In order to impose periodic boundary conditions the strategy where cells or particles are added to complete the numerical stencil is employed. In the case of FVM, Figure 3.12 shows that two layers of virtual cells are added for the periodic boundaries by copying them from the fluid cells. For that purpose, care has been taken during mesh generation that the part of the fluid mesh indicated by a red frame has coincident nodes with the part indicated by a black frame if they are translated by the domain width. If this is the case, the cells next to the boundaries of one block can be added at the other block with continuous mesh lines. In the case of the MUSCL scheme, two layers of cells are needed due to the computation of the gradients. Using Godunov's scheme with states that are supposed to be constant in a cell, necessitates only one layer of cells.

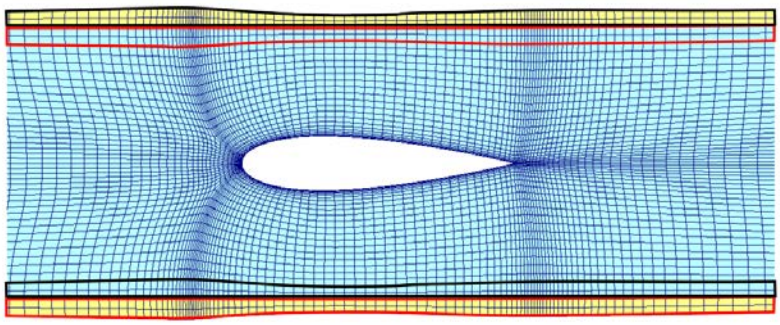


Figure 3.12: A block-structured mesh around a symmetric NACA hydrofoil that is used for the FV solver. The blue mesh is the normal fluid mesh while the yellow mesh are virtual cells that are copied for the periodicity conditions. Note that the mesh lines are continuous everywhere even between the original mesh (blue) and the virtual cells (yellow). The virtual cells are constructed by copying the cells in the black frame (bottom) to the black framed virtual cells (top) and the red ones (top) to the red framed virtual cells (bottom).

The same technique is applied for SPH-ALE where, in the case of the MUSCL scheme, particles within two diameters of the kernel support are copied and translated. These fictitious particles are also called *ghost particles* or *phantom particles*.

3.7.6 Free surface condition

Free surfaces are particular boundaries that in reality represent interfaces between water and air. They are characterized by dynamic and kinematic conditions. For the inviscid Euler equations the dynamic condition says that the pressure is continuous through the interface, i.e. the pressure of the water phase at the free surface is equal to the pressure of the air phase. The kinematic condition states that the velocity component of the water phase normal to the interface is equal to the normal velocity of the air phase and that the interface moves according to that velocity field.

In SPH it is possible to compute free surface flows without modelling the air phase. It is assumed that particles in Lagrangian motion automatically satisfy the kinematic condition. Also the dynamic condition seems to be fulfilled without any special treatment or modification of the equations. It was said before that we usually choose a zero reference pressure, $p_0 = 0$, in the weakly-compressible equation of state (3.4). The reason for that is that the pressure at the free surface has to be equal to the ambient pressure and in our case it is equal to zero. Hence, we interpret the missing particles in the kernel support as air particles with zero pressure that are taken into account automatically. That means that the surface integral does not have to be computed because the kernel support should not be seen as truncated at the free surface. A detailed explanation for free surface treatment in SPH-ALE can be found in [22].

3.8 Time integration

From the FV and the SPH-ALE space discretization we obtain two similar sets of semi-discrete differential equations (3.60) and (3.62) of the form,

$$\frac{d\mathbf{\Psi}}{dt} = \mathcal{H}(\mathbf{\Psi}, \mathbf{x}, t).$$

The same time integration scheme is used for both approaches, that is an explicit Runge-Kutta method. In particular, we use the explicit Euler scheme, the Heun scheme (a second order Runge-Kutta scheme) and the explicit forth-order Runge-Kutta time integration scheme. They can be found in books about numerical mathematics in general or CFD in particular, e.g. [46].

The idea of an explicit scheme is that the update of the state variables $\mathbf{\Psi}$ from time $t = t^n$ to $t = t^{n+1}$ is done by using the known values of the function $\mathcal{H}^n := \mathcal{H}(\mathbf{\Psi}^n, \mathbf{x}^n, t^n)$ at time t^n. In the contrast, an implicit method would use $\mathcal{H}^{n+1}$ to evaluate $\mathbf{\Psi}^{n+1}$.

The explicit Euler scheme is defined by

$$\mathbf{\Psi}^{n+1} = \mathbf{\Psi}^n + \Delta t\, \mathcal{H}^n. \tag{3.92}$$

The higher-order Runge-Kutta schemes include s intermediate steps called stages in the interval $[t^n, t^{n+1}]$. The time update for the Heun scheme is then given by

$$\begin{cases} \overline{\mathbf{\Psi}}^{n+1} = \mathbf{\Psi}^n + \Delta t\, \mathcal{H}^n, \\ \mathbf{\Psi}^{n+1} = \mathbf{\Psi}^n + \dfrac{1}{2}\Delta t\, \left(\mathcal{H}(\overline{\mathbf{\Psi}}^{n+1}) + \mathcal{H}^n \right). \end{cases} \tag{3.93}$$

Similarily, the fourth-order Runge-Kutta scheme is defined by

$$\begin{cases} \mathbf{\Psi}^{(1)} = \mathbf{\Psi}^n \\ \mathbf{\Psi}^{(2)} = \mathbf{\Psi}^n + \dfrac{1}{2}\, \Delta t\, \mathcal{H}^{(1)}, \\ \mathbf{\Psi}^{(3)} = \mathbf{\Psi}^n + \dfrac{1}{2}\, \Delta t\, \mathcal{H}^{(2)}, \\ \mathbf{\Psi}^{(4)} = \mathbf{\Psi}^n + \Delta t\, \mathcal{H}^{(3)}, \\ \mathbf{\Psi}^{n+1} = \mathbf{\Psi}^n + \dfrac{1}{6}\, \Delta t\, \left(\mathcal{H}^n + 2\mathcal{H}^{(2)} + 2\mathcal{H}^{(3)} + \mathcal{H}^{(4)} \right), \end{cases} \tag{3.94}$$

where the $\mathcal{H}^{(k)} := \mathcal{H}(\mathbf{\Psi}^{(k)})$ with $k = 1, ..., 4$.

The discretization sizes of time Δt and space Δx are linked for hyperbolic equations by the so-called *Courant-Friedrichs-Lewy* (CFL) condition,

$$\Delta t = K_{CFL} \min_{i \in \Omega} \frac{\Delta x_i}{\|\mathbf{v}_i\| + c_i}, \tag{3.95}$$

where c_i denotes the local sound speed and K_{CFL} the CFL number that is $K_{CFL} < 1$ for explicit time integration schemes. Basically, the condition says that the time step size has to be chosen in a way that a wave does not leave a considered cell with size Δx_i during one time step. Hence, decreasing the spatial discretization size also yields a smaller time step size. Another consequence of this relation is that, it is computationally too expensive to use the real physical speed of sound for water and a numerical sound speed is used instead.

3.9 Correction methods for the SPH kernel function and its gradient

It has been discussed in Section 3.2.2 that SPH suffers from a lack of consistency due to the discretization of the kernel approximation that does not satisfy the continuous consistency conditions anymore. In literature, different methods can be found that correct the kernel function and/or its gradient in a way that the consistency conditions for the particle approximation of a

function and/or its gradient are satisfied. This section does not give an exhaustive summary of these approaches but only presents the ones that we actually use in this work. For more details on different correction methods the reader is referred to [14, 62, 57].

In the following, two well-known correction methods for the kernel function and its gradient, respectively, are presented. Then, a novel correction method that ensures zeroth order consistency for the computation of the divergence of the flux vector is introduced and its application to the SPH-ALE flux equations is discussed.

3.9.1 Shepard correction

It was explained in Section 3.2.2 that in general the discrete kernel function does not satisfy the normalization condition (3.42). The Shepard correction [99] is a technique that enforces the discrete normalization condition by defining a new kernel function as,

$$\widetilde{W}_{ij} = \frac{W_{ij}}{\sum\limits_{j \in D_i} \omega_j W_{ij}}. \tag{3.96}$$

The Shepard coefficient $\beta_i := \sum_{j \in D_i} \omega_j W_{ij}$ has to be computed in an additional loop over the particles. The correction is a zeroth-order moving least square (MLS) method, see [7]. It should not be confused with the function $\gamma(\mathbf{x}_i) = \int_{D_i} W(\mathbf{x}' - \mathbf{x}_i, h) dV'$ (Eq. (3.36)) that is evaluated analytically. Indeed, the Shepard coefficient β is not equal to one in the interior of the domain where D_i is not truncated. Hence, applying the corrected gradient of the kernel function that is computed by

$$\widetilde{\nabla_i W}_{ij} = \frac{\nabla_i W_{ij}}{\sum\limits_{j \in D_i} \omega_j W_{ij}},$$

to the SPH-ALE equations, Eq. (3.60), would change the SPH equations everywhere.

For that reason, it is only used for interpolation purpose in the coupling technique of SPH-ALE and FVM presented in the Chapter 4.

3.9.2 Renormalization

Renormalization is a correction approach that increases the order of consistency of the particle approximation of the gradient of a function by correcting the gradient of the kernel function [92]. We usually apply the renormalization technique to the computation of the gradients for the MUSCL scheme (Section 3.6.3.2).

We consider a linear function $f(\mathbf{x}) = a_1 + a_2 \mathbf{x}$, with $a_1, a_2 \in \mathbb{R}$ denoting two constants. A linearly consistent particle approximation of the gradient of f (Eq. (3.44)) verifies in the interior of the domain,

$$a_2 \left[\left(\sum_{j \in D_i} \omega_j (\mathbf{x}_j - \mathbf{x}_i) \otimes \nabla_i W_{ij} \right) \mathbf{B}_i \right] = a_2 \mathbf{I}.$$

Accordingly, $\mathbf{B}_i$ the renormalization matrix of particle i, is defined by

$$\mathbf{B}_i := \left(\sum_{j \in D_i} \omega_j (\mathbf{x}_j - \mathbf{x}_i) \otimes \nabla_i W_{ij} \right)^{-1}, \tag{3.97}$$

which means that a matrix inversion has to be computed for every particle at every time step if particles move.

Both renormalization and Shepard's correction destroy the anti-symmetry of the gradient because the corrections depend on the neighbourhood of a particle i which is not the same as for particle j. Hence, $\mathbf{B}_i \neq \mathbf{B}_j$ and $\beta_i \neq \beta_j$. If they are applied to the SPH-ALE equations, Eq. (3.60), the method is not exactly conservative anymore. For that reason, Vila [111] proposed to use a symmetrization of the renormalization matrix, i.e.

$$\mathbf{B}_{ij} := \frac{\mathbf{B}_i + \mathbf{B}_j}{2}$$

when it is applied to the gradient of the kernel function in the SPH-ALE equations.

3.9.3 Closed box correction

In the following, a novel correction method for the kernel gradient will be presented. It was developed in the frame of this work and it is applied to all simulations in this document. The idea is to correct the gradient of the kernel function in a way that the condition for zeroth consistency, Eq. (3.43), is fulfilled,

$$\sum_{j \in D_i} \omega_j \widetilde{\nabla_i W}_{ij} + \sum_{j \in \partial D_i} \omega_j^{\partial} W_{ij} \mathbf{n}_j = 0, \tag{3.98}$$

where $\widetilde{\nabla W}_{ij}$ is the corrected gradient of the kernel function. We call this correction the *closed box correction* and we define the closed box error $\mathbf{b}_i$ as

$$\mathbf{b}_i := \sum_{j \in D_i} \omega_j \nabla_i W_{ij} + \sum_{j \in \partial D_i} \omega_j^{\partial} W_{ij} \mathbf{n}_j \neq 0. \tag{3.99}$$

It was seen in Section 3.5, comparing SPH-ALE with FVM that the failure of the particle approximation to fulfil Eq. (3.43) can be interpreted as summing the fluxes through a control volume that is not completely closed. As illustration of the consequences, let us consider a fluid at rest in a periodic box (no boundary terms) where constant pressure and zero velocity are initially imposed. Theoretically no velocity should be created. If we take a look at the SPH-ALE equations (3.80) and if we insert $\mathbf{v} = \mathbf{v}_0 = 0$ and $p = p_c = cst$ at $t = 0$, we obtain from the momentum equation

$$\frac{d}{dt}(\omega_i \rho_i \mathbf{v}_i) = -2\,\omega_i\, p_c \underbrace{\sum_{j \in D_i} \omega_j \nabla_i W_{ij}}_{\neq 0} \neq 0.$$

As a consequence, momentum is created non-physically and hence, it is not even possible to simulate a fluid at rest correctly. Figure 3.13 shows the velocity field after only hundred iterations, corresponding to $t = 0.01242s$. There are 80×80 particles distributed on a Cartesian grid that were shifted by random numbers. The simulation is launched in Lagrangian particle motion. The left image was obtained without gradient correction and shows a maximum velocity of 0.0127 m/s. The right image was obtained with the corrected gradient and the velocity is smaller than 10^{-15}.

Price [91] interprets the artificial velocity that is created as an intrinsic re-meshing of the particles. If the distribution of the particles is bad for the SPH operators, the error term from Eq. (3.43) is relatively big which creates a movement of the particles. In the case of a positive uniform pressure field, the movement fills the holes and distributes the particles isotropically. This is exactly what can be observed if the above example is ran without a corrected kernel gradient. After 1000 iterations ($t = 0.59s$) the particles have distributed themselves regularly

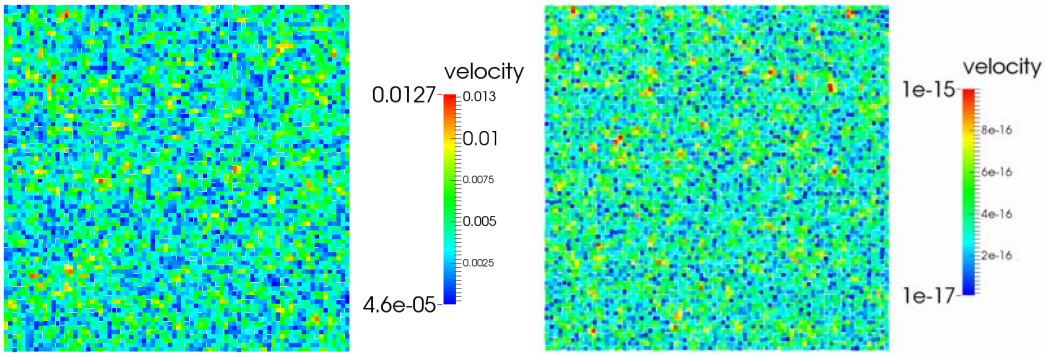


Figure 3.13: Artificial velocity created after 100 iterations ($t = 0.01242s$) without the closed box correction (left) and with the closed box correction (right). Note the different scales.

and the maximum of the error term $\mathbf{b}_i$ is more than 15 times smaller. To reach this optimum distribution, which is not a Cartesian distribution, the particles have to move. Since the particle velocity is equal to the flow velocity in Lagrangian formulation, errors are introduced into the physical field variables.

In meshbased ALE methods, like FVM and FEM with moving and deforming meshes, the *geometric conservation law* describes the capability of the ALE method to exactly compute fields that are constant in space and in time (see [40, 73]). If the geometric properties of the mesh are computed following this conservation law, the computed constant fields are not influenced by the motion and deformation of the grid. This concept is thus equivalent to the problem described in this section.

It is to notice that in the astrophysics SPH community the so-called $\mathbf{E}_0$ error which is very similar to the *closed box error* was analysed by several authors (see [94, 21]). It is defined by

$$\mathbf{E}_{0,i} := \sum_{j \in D_i} \omega_j \left[g_{ij} + g_{ij}^{-1} \right] \nabla_i W_{ij},$$

with

$$g_{ij} := \frac{\rho_j}{\rho_i} \frac{\phi_i}{\phi_j}.$$

The $\mathbf{E}_0$ error is the leading error term in their formulation of the momentum equation and the free function ϕ_i can be chosen in a way that the error is minimized. It is interesting that they find that choosing $\phi_i = \rho_i$ improves the results. With that choice $\mathbf{E}_0$ corresponds to our definition of the closed box error Eq. (3.99). It is established that $\mathbf{E}_0$ decreases by increasing the number of neighbours for each particle.

Nevertheless, we do not choose to increase the number of neighbours because that increases the computational cost of the method further. Instead we present a correction to enforce the closed box error condition. The corrected gradient of the kernel function has the form

$$\widetilde{\nabla_i W}_{ij} := \nabla_i W_{ij} - c_{ij} \mathbf{b}_i, \tag{3.100}$$

where the coefficients c_{ij} have the property that

$$\sum_{j \in D_i} \omega_j c_{ij} = 1. \tag{3.101}$$

The kernel gradient verifies $\nabla_i W_{ii} = 0$, which means that it does not introduce any self-contribution of particles into the flow equations. We want to keep this property for the corrected kernel gradient. Therefore, the coefficients have to be chosen in a way that

$$c_{ii} = 0. \tag{3.102}$$

Corrected gradients of the form (3.100) satisfy the discrete closed box condition (3.98) because

$$\begin{aligned}
\sum_{j\in D_i} \omega_j \widetilde{\nabla_i W}_{ij} + \sum_{j\in\partial D_i} \omega_j^{\partial} W_{ij}\mathbf{n}_j &= \sum_{j\in D_i} \omega_j \left(\nabla_i W_{ij} - c_{ij}\mathbf{b}_i\right) + \sum_{j\in\partial D_i} \omega_j^{\partial} W_{ij}\mathbf{n}_j, \\
&= \sum_{j\in D_i} \omega_j \nabla_i W_{ij} - \mathbf{b}_i \underbrace{\sum_{j\in D_i} \omega_j c_{ij}}_{=1} + \sum_{j\in\partial D_i} \omega_j^{\partial} W_{ij}\mathbf{n}_j, \\
&= \sum_{j\in D_i} \omega_j \nabla_i W_{ij} + \sum_{j\in\partial D_i} \omega_j^{\partial} W_{ij}\mathbf{n}_j - \mathbf{b}_i, \\
&= \mathbf{b}_i - \mathbf{b}_i = 0.
\end{aligned}$$

Hence, the error $\mathbf{b}_i$ of a particle i is redistributed on its fluid neighbours by changing the kernel gradient of the particle interactions. But the gradient of the boundary contributions is not changed even though the boundary contributions are considered in $\mathbf{b}_i$ and in Eq. (3.98). The coefficients c_{ij} can be chosen freely, e.g.

$$c_{ij} = \frac{1}{\omega_j N_i}, \text{ and } c_{ii} = 0,$$

where N_i denotes the number of neighbours of particle i. However, this is not what we do. Instead we choose the coefficients in a way that corresponds better to the character of the SPH method. We distribute the error on the neighbours weighted by the kernel function W_{ij}, i.e.

$$c_{ij} = \frac{W_{ij}}{\omega_j s_i}, \text{ with } s_i := \sum_{\substack{k\in D_i \\ k\neq i}} W_{ik}, \text{ and } c_{ii} = 0, \tag{3.103}$$

or also weighted by the volumes ω_j

$$c_{ij} = \frac{W_{ij}}{\sigma_i}, \text{ with } \sigma_i := \sum_{\substack{k\in D_i \\ k\neq i}} \omega_k W_{ik}, \text{ and } c_{ii} = 0. \tag{3.104}$$

Note that σ_i is not the Shepard coefficient β_i of the Shepard correction, Eq. (3.96), because σ_i does not include the self-contribution term $\omega_i W_{ii}$. If the ω_j are constant in the whole neighborhood of i, the sums s_i and σ_i reduce to the same expression. In this document both expressions (3.103) and (3.104) are used where (3.104) is preferred if the ω_j are varying strongly like in the case of the coupling method presented in the next chapter. The sums $\mathbf{b}_i$, s_i or σ_i have to be computed in an additional loop over the neighbours before summing the fluxes.

The formula (3.104) using σ_i is similar to the first step of the correction method introduced by Leonardi *et al.* [57] without boundary terms. In their paper, that was presented at the SPHERIC 2014 workshop, explicit strategies for consistent kernel approximations are introduced. In particular, they use a kernel gradient correction that is of first-order consistency and uses two steps. The first step derives a kernel gradient $\hat{W}'_{ij}$ that ensures zeroth order consistency. The second step is a normalization step that enforces the condition for linear consistency.

However, the corrected kernel gradient $\hat{W}'_{ij}$ has an important difference compared to $\widetilde{\nabla_i W}_{ij}$. It includes self-contributions W_{ii} that we preferred not to include. That means that $\hat{W}'_{ii} \neq 0$ while we enforce $\widetilde{\nabla_i W}_{ii} = 0$ by choosing $c_{ii} = 0$.

The corrected gradient $\widetilde{\nabla_i W}_{ij}$ is not anti-symmetric anymore (see Eq. (3.41)) and therefore the SPH-ALE equations together with $\widetilde{\nabla_i W}_{ij}$ are not exactly conservative because Eq. (3.56) is not fulfilled. This problem is known in literature and arises for all kernel gradient correction methods. The correction terms are sometimes symmetrized (see Section 3.9.2) in order to keep an anti-symmetric gradient. In this way the conservation properties are maintained but the conditions that should be enforced are not fulfilled exactly anymore. In the case of the closed box correction it is important to enforce $\mathbf{b}_i = 0$ exactly for each particle and we do not symmetrize the correction.

Validation cases that illustrate the improvement of the results obtained by this method are given in the end of this chapter, see Section 3.10.

3.9.4 Correction of the particle motion

A method that aims at correcting the closed box error by a different approach has been developed in parallel to this project by J.-C. Marongiu and J. Leduc. Instead of correcting the gradient of the kernel function, the particle velocity is modified in a way that the closed box error is reduced. This is possible because the particle velocity in SPH-ALE is a free parameter that can be chosen arbitrarily. Hence, the errors that are introduced by a poor particle distribution can be reduced by adapting the particle distribution itself through a corrected particle velocity.

The main idea is to use the intrinsic re-meshing of SPH, that was described in Section 3.9.3 for the fluid at rest case, but without modifying the physical field variables, i.e. without creating an artificial fluid velocity. To our knowledge, the intrinsic re-meshing of SPH was first used explicitly by Colagrossi *et al.* [19] who developed a particle packing algorithm that makes use of the fact that the vector $\mathbf{w}_i := -\mathbf{b}_i$ always points in the direction of the holes in the particle distribution. The packing algorithm utilizes this feature of SPH in order to obtain an initial particle distribution that minimizes the closed box error, while we want to use it to correct the particle velocity in every time step throughout the simulation.

Following [19] we want to shift the particles in the direction of $\mathbf{b}_i$, Eq. (3.99). For that purpose, a uniform pressure field is considered that is applied to all particles with

$$\forall i \in \Omega : \; p_i = p_{ref} = \frac{\rho_0 c_0^2}{\gamma}. \tag{3.105}$$

This constant pressure only acts on the kinematics of the particles, i.e. on the particle velocity $\mathbf{v}_0(\mathbf{x}_i)$ and the particle position $\mathbf{x}_i$. Furthermore, we assume that the system that couples the position, the particle velocity and the constant pressure p_{ref} is a hyperbolic system that necessitates a decentered treatment of the same type as it is used in Riemann solvers. This is due to the fact that numerical experiments have shown that the system is unstable if no decentered treatment is employed. Hence, we compute

$$\begin{aligned}\left(\frac{d\mathbf{v}_0(\mathbf{x}_i)}{dt}\right)_{corr} := &\sum_{j\in D_i} \omega_j \left[p_{ref} + \frac{c_0}{2}\left(\mathbf{v}_0(\mathbf{x}_i) - \mathbf{v}_0(\mathbf{x}_j)\right)\cdot \mathbf{n}_{ij}\right] \nabla_i W_{ij} \\ &+ \sum_{j\in \partial D_i} \omega_j^{\partial} \left[p_{ref} - \frac{c_0}{2}\left(\mathbf{v}_0(\mathbf{x}_i) - \mathbf{v}_0(\mathbf{x}_j)\right)\cdot \mathbf{n}_{j}\right] W_{ij}\mathbf{n}_j,\end{aligned}$$

where the direction

$$\mathbf{n}_{ij} = \frac{\nabla_i W_{ij}}{\|\nabla_i W_{ij}\|},$$

is defined as in Eq. (3.65) and $\mathbf{n}_j$ denotes the normal of the surface element j. Then, an additional term is employed in order to smooth the transport field,

$$\left(\frac{d\mathbf{v}_0(\mathbf{x}_i)}{dt}\right)_{smooth} := \left(\frac{d\mathbf{v}_0(\mathbf{x}_i)}{dt}\right)_{corr} - \alpha \sum_{j \in D_i} \frac{c_0}{h_i}\, \omega_j \left(\mathbf{v}_0(\mathbf{x}_j) - \mathbf{v}_0(\mathbf{x}_i)\right) W_{ij}, \tag{3.106}$$

with α denoting a numerical parameter.

Expression (3.106) is used to define two different types of corrected particle motion, a corrected Lagrangian motion and a corrected Eulerian motion, as shown in the following.

- Eq. (3.106) is applied to correct particles in Lagrangian motion in a way that the closed box error is reduced. For explicit Euler time integration scheme (see Eq. (3.92)), we obtain

$$\mathbf{v}_0^{n+1}(\mathbf{x}_i) = \mathbf{v}^{n+1}(\mathbf{x}_i) - \Delta t \left(\frac{d\mathbf{v}_0(\mathbf{x}_i)}{dt}\right)_{smooth}. \tag{3.107}$$

  De Leffe [22] already introduced an ALE correction of the particle velocity based on the closed box error but with a different formulation.

- We use Eq. (3.106) to define another type of particle motion where the particles are not following the flow in a Lagrangian way but where they fill the holes in the distribution. In particular, it can be used to obtain particles that follow moving boundaries in a way that the closed box error is small. In the case of explicit Euler time integration (see Eq. (3.92)), we have

$$\mathbf{v}_0^{n+1}(\mathbf{x}_i) = \mathbf{v}_0^{n}(\mathbf{x}_i) - \Delta t \left[\left(\frac{d\mathbf{v}_0(\mathbf{x}_i)}{dt}\right)_{smooth} + \alpha \frac{c_0}{h_i} \mathbf{v}_0^n(\mathbf{x}_i)\right], \tag{3.108}$$

  and $\mathbf{v}_0^0(\mathbf{x}_i) = 0$. The term

$$\mathbf{T}_i := \alpha \frac{c_0}{h_i} \mathbf{v}_0^n(\mathbf{x}_i)\ \Delta t$$

  is added to obtain convergence of the particle positions. A similar damping term is also used for the packing algorithm [19]. This type of particle motion was used in the simulation of the Francis turbine start-up that was rapidly presented in Chapter 1. Furthermore, it will be employed in Chapter 5.

In practice, it turned out that it is sometimes better to directly correct the positions of the particles by *shifting* them instead of modifying the particle velocity, i.e.

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_0^n(\mathbf{x}_i)\, \Delta t - \frac{1}{2}(\Delta t)^2 \left(\frac{d\mathbf{v}_0(\mathbf{x}_i)}{dt}\right)_{smooth}. \tag{3.109}$$

It is important to note that the correction of the particle motion and the closed box correction are not mutually exclusive methods. On the contrary, they work best when they are used together. The correction of the particle motion reduces the closed box error but does not eliminate it completely and best results are obtained when using both approaches.

3.10 Implementation and validation

The in-house SPH-ALE solver of ANDRITZ Hydro is called ASPHODEL and is built on the PhD theses of J.-C. Marongiu [67] and J. Leduc [55]. It is well validated for free surface flows and for flows in hydraulic machines like Pelton turbines. Validation cases on this topic and on an implemented multiphase solver can be found in the publications of Marongiu and Leduc, e.g. [71], [56]. The extension of the solver to internal flows especially the implementation of the closed box correction was amongst others done in the frame of this PhD project and some validations are presented.

The weakly-compressible FV solver, based on Section 3.4, was on the whole implemented in the frame of this PhD project and some validation cases are given in Section 3.10.2. However, we want to remind the reader that the purpose of this thesis is not the development of another FV solver but the coupling of this solver with the already existing SPH-ALE solver. This coupling will be presented in detail in Chapter 4.

3.10.1 SPH-ALE with closed box correction

The closed box correction that was introduced in Section 3.9 was applied to all SPH-ALE simulations presented in this document. It cannot be directly applied to free surface flows but these flows are not subject of this thesis. In the following, the Taylor-Green vortices from Chapter 2 are used for the validation. We compare the results obtained with and without the closed box correction on two different particle distributions. However, at first we take a closer look at the fluid at rest case mentionned in Section 3.9.3.

Every time a computed value for the magnitude of the closed error is plotted or explicitly given, the closed box error is divided by a reference value. The reference value represents the closed box error of a kernel support that is truncated by 50% and depends on the considered kernel function and on the smoothing length. Using this scaling allows us to compare the closed box error of different simulations with different discretization sizes to each other.

Fluid at rest in a periodic box

We start from an initial distribution with strongly disordered particles that was generated in the following way. Particles with size $\Delta x = 0.0125$m situated on a Cartesian grid (with grid spacing Δx) are displaced randomly from their initial position where the displacement is smaller than $0.25\,\Delta x$ in each coordinate direction. A first simulation is launched in Lagrangian description with a positive uniform initial pressure of 2000 Pa and $\mathbf{v} = 0$ m/s and $\mathbf{v}_0 = 0$ m/s everywhere. As Price [91] explained, the particles reorder themselves in a way that minimizes the closed box error. After 2000 iterations, $t = 1.1837s$, the maximum error is about 0.04, see Figure 3.14 and Figure 3.16 (b). Also the artificial velocity reduced itself to less than 0.0085 m/s and accordingly the total kinetic energy is very small, see Figure 3.15 (a). However, the initial pressure was not maintained and the pressure is fluctuating between $p \in [535, 597]$ Pa, see Figure 3.16 (a). Moreover, Figure 3.15 (b) shows that the total volume increased of about $1 - 1.5\%$.

We see that the particles tend to reorder themselves in order to minimize the closed box error but in more complex flows there are physical mechanisms that work against this tendency. In the point of view of the numerical scheme and of the closed box error, particles are disordered close to stagnation points or vortices that occur in more complicated flows.

Applying the closed box correction to this simple case prevents the creation of an artificial velocity. The particle distribution stays constant because $\mathbf{v}_0 = \mathbf{v} \approx 0$ and the initial pressure is

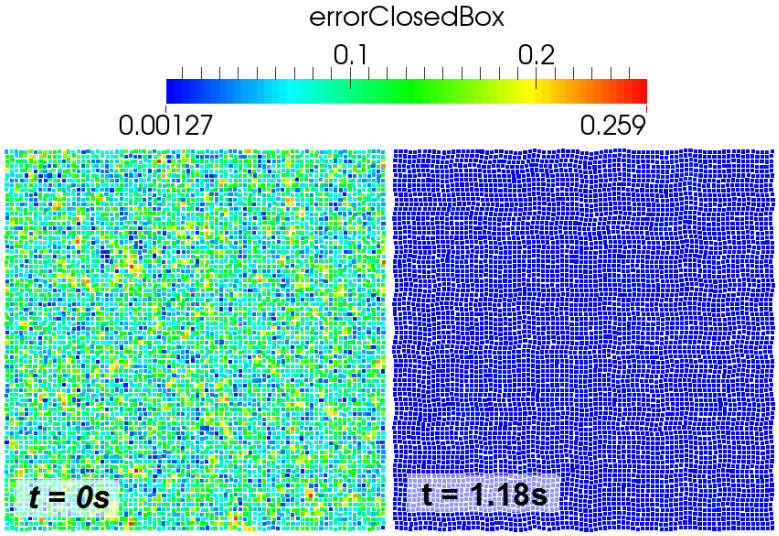
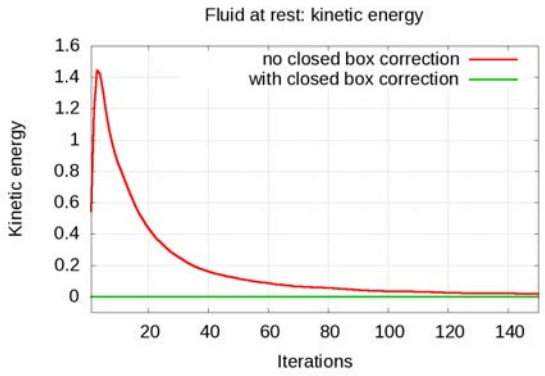
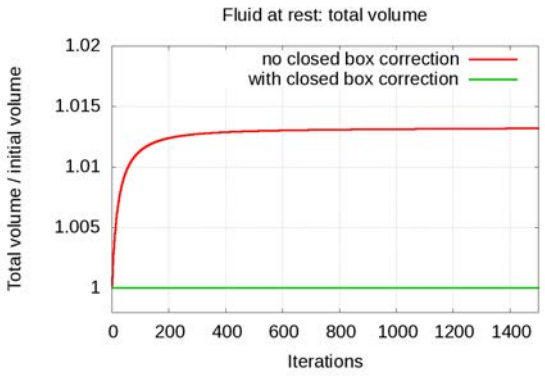


Figure 3.14: The magnitude of the closed box error for the fluid at rest case without closed box correction. At the beginning the closed box error is high (left) but the movement of the particles reduces the error (right) because the particles reorganize themselves.

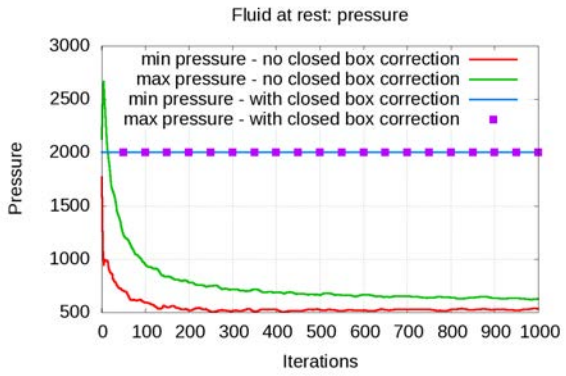


(a) Total kinetic energy plotted over the number of iterations. If no correction is employed, the kinetic energy increases at the beginning when the particles redistribute themselves and decreases when a stable configuration is reached.

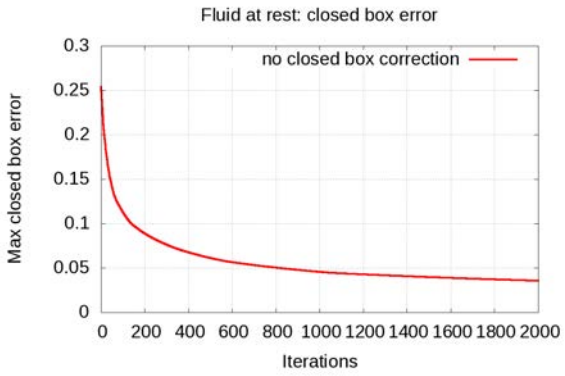


(b) The total volume is not maintained with particles that move in Lagrangian motion. In the case without correction it increases at first when the velocity increases and reaches a stable value when the particles found an adjusted distribution.

Figure 3.15: The evolution of the kinetic energy and the total volume plotted over the number of iterations for the fluid at rest case with and without closed box correction.



(a) The minimum and maximum pressure plotted over the number of iterations. If no correction is employed the pressure decreases, while it stays constant at the correct value if the correction is used.



(b) In the case without correction, the maximum closed box error in the fluid domain decreases because the particles redistribute themselves in a way that the error is reduced.

Figure 3.16: The evolution of the pressure with and without closed box correction and the closed box error plotted over the number of iterations for the fluid at rest case.

maintained, see Figure 3.16 (a). No variation of the global mass has been detected after 2000 iterations with and without closed box correction.

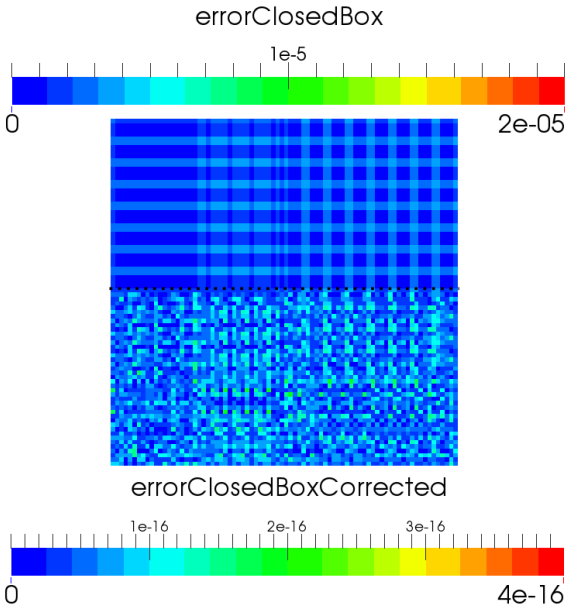
Taylor-Green vortex

We have seen in Chapter 2, Section 2.1, that the bi-periodic Taylor-Green vortex is an analytical steady-state solution of the Euler equations. In the following, this case is used to illustrate the effect of the closed box correction on the flow fields.

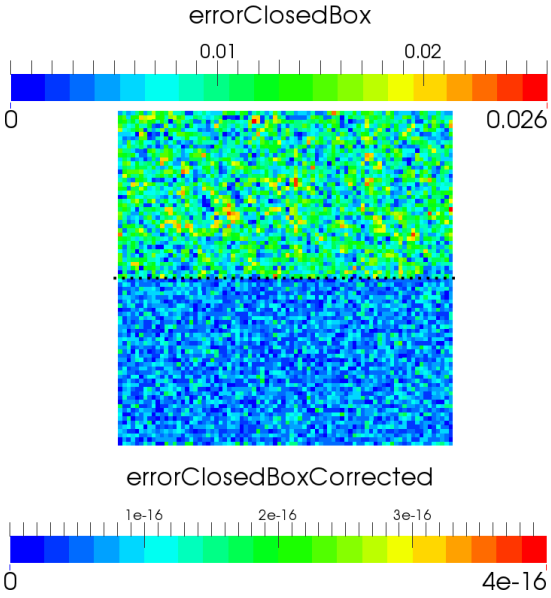
We launch simulations in Eulerian description without applying renormalization on the gradients of the MUSCL scheme. Two different distributions of the particles are considered. The first one, *Distribution 1*, is a Cartesian distribution where 80×80 particles are situated on a Cartesian grid in the unit square $[0, 1] \times [0, 1]$. The particle size is $\Delta x = 0.0125$m. We assume that this distribution leads to very small closed box errors which is confirmed by the simulation. For all particles the closed box error is small, $\|\mathbf{b}_i\| < 5 \cdot 10^{-6}$. The second particle distribution, *Distribution 2*, is modified from the Cartesian one, where the original positions are randomly modified by a displacement that is smaller than $\frac{1}{40}\Delta x$ in each coordinate direction. Note that this randomly disturbed mesh is not a random mesh but a mesh that only slightly differs randomly from a Cartesian mesh.

Figure 3.17 shows the closed box error $\mathbf{b}_i$ for both particle distributions with and without the correction. *Distribution 1* shows a closed box error that is three magnitudes smaller than *Distribution 2*. With the correction the error term is almost zero ($\|\mathbf{b}_i\|$ smaller than $4 \cdot 10^{-16}$) for both distributions. Hence, this confirms that the error can be interpreted as a measure for the particle disorder.

If the error is too big, i.e. if the particles are disordered, it is directly translated into a very noisy pressure field. Figure 3.18 shows a comparison of the computed pressure field with both particle distributions with and without closed box correction after a relatively short physical time of $t = 0.1s$. *Distribution 1* shows a smooth pressure field with and without correction because the closed box error is initially small. *Distribution 2* on the contrary shows big errors

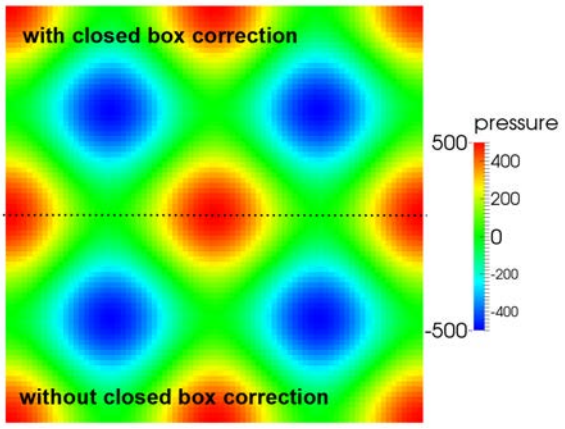


(a) The particles are situated on a Cartesian grid. The error closed box is shown with and without correction.

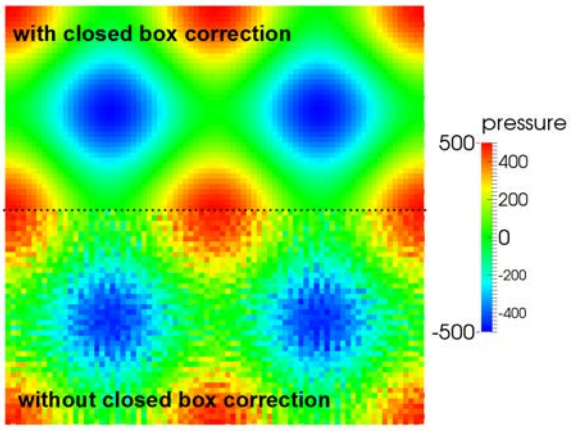


(b) The particles are situated on a grid that was randomly disturbed. The error closed box is shown with and without correction.

Figure 3.17: Comparison of the closed box error on a Cartesian and on a slightly disturbed particle distribution. The closed box error for the disturbed distribution is three orders of magnitude bigger than on the Cartesian distribution.



(a) The particles are situated on a Cartesian grid and the closed box error is smaller than $5 \cdot 10^{-6}$ for every particle. The pressure is very smooth and the correction has no visible effect.



(b) The particles are situated on a grid that was randomly disturbed. The maximum closed box error is 0.0267 and the pressure for the non-corrected simulation is extremely noisy.

Figure 3.18: Pressure field for the Taylor-Green vortices after $t = 0.1s$ for two different particle distributions with and without closed box correction.

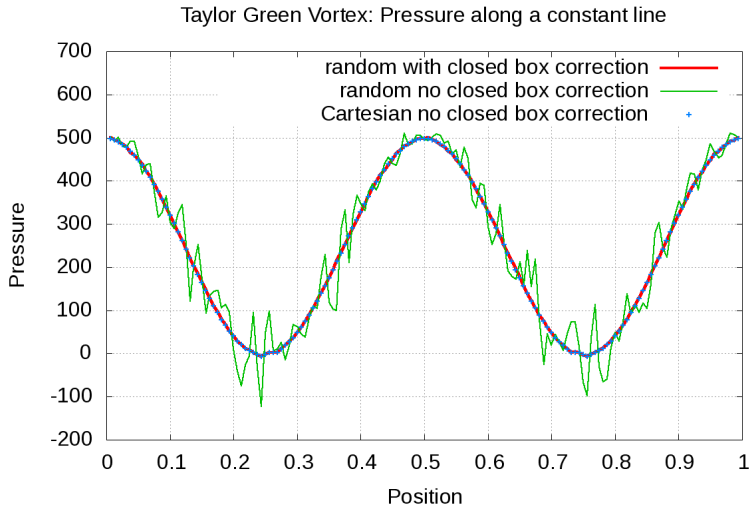


Figure 3.19: Pressure on a line at $y = 0.5$m for *Distribution 1* and *Distribution 2* with and without correction. The position of the line is indicated in Figure 3.18 by a black dotted line.

and the pressure field without correction is disturbed. However, applying the correction allows us to recover the smooth pressure field of the Cartesian grid. A more detailed analysis is given by Figure 3.19 that shows the pressure on a line at $y = 0.5$m. It confirms that the non-corrected pressure on the Cartesian distribution and the corrected pressure on the "random" distribution correspond very well while it is not possible to obtain correct results on Distribution 2 without the correction.

The same analysis is carried out for the velocity field where the effect of the error is less visible. Figure 3.20 shows the magnitude of the velocity field on *Distribution 2* with and without correction. The x component of the velocity on a line is plotted in Figure 3.21 where a detailed view in the zone around the maximum velocity shows that the correction also improves the velocity field.

3.10.2 Finite Volume solver

The implemented FV solver is an unstructured solver based on the numerical schemes introduced in Sections 3.4, 3.6, 3.7 and 3.8. We have seen in Section 3.5 that the numerical fluxes that are summed over the neighbour cells can be computed in the same way as in the SPH-ALE solver and the same time integration schemes can be used. In practice, this means that Riemann solvers and time integration schemes, that were already implemented in ASPHODEL, can be reused for the FV solver.

Even though the solver is unstructured, we use structured multiblock grids because they are advantageous close to boundaries. These meshes are either simple regular grids or they are obtained by an in-house meshing software of ANDRITZ. In the following, the validation cases are utilized that were presented in Chapter 2. At first the results for the Taylor-Green vortex is presented, then the inviscid flow around symmetric NACA hydrofoil.

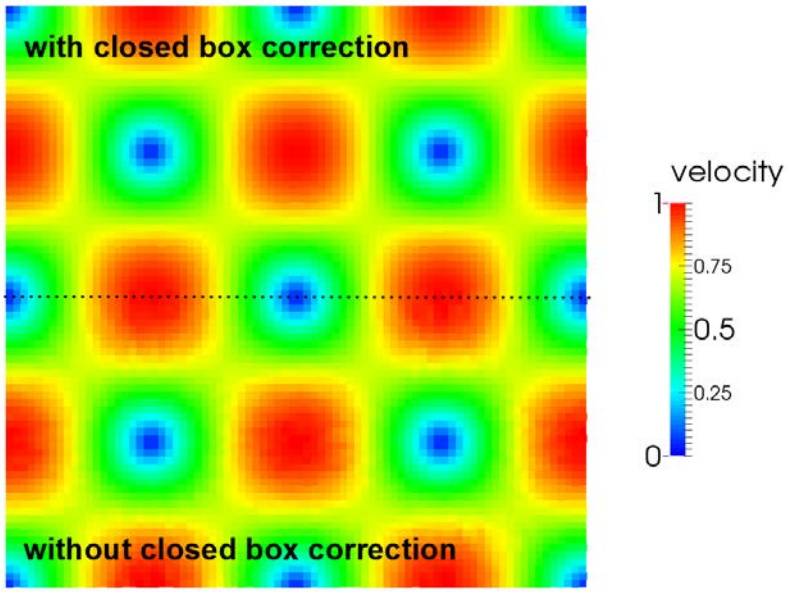
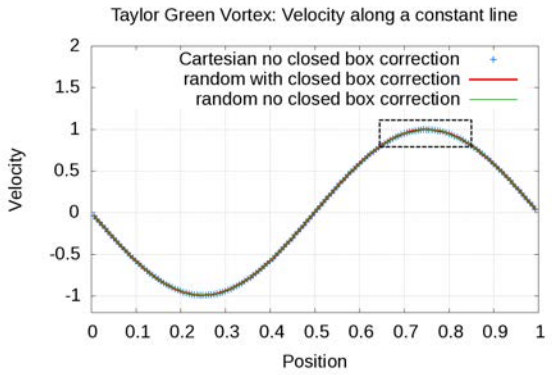
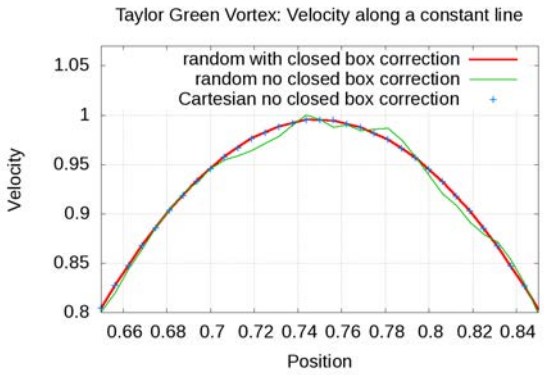


Figure 3.20: Velocity field on *Distribution 2* with and without closed box correction. The velocity without correction is less smooth but the difference is less visible than for the pressure field.



(a) Velocity on a constant line with and without correction. The differences are less visible than for the pressure field. The black box indicated the view of the picture on the right hand side.



(b) Detailed view in the zone of maximum velocity. The velocity is less smooth without correction but the difference is less important than for the pressure.

Figure 3.21: The x component of the velocity plotted on a line $y = 0.5$m for the Taylor Green vortices after $t = 0.1s$ for two different particle distributions with and without closed box correction.

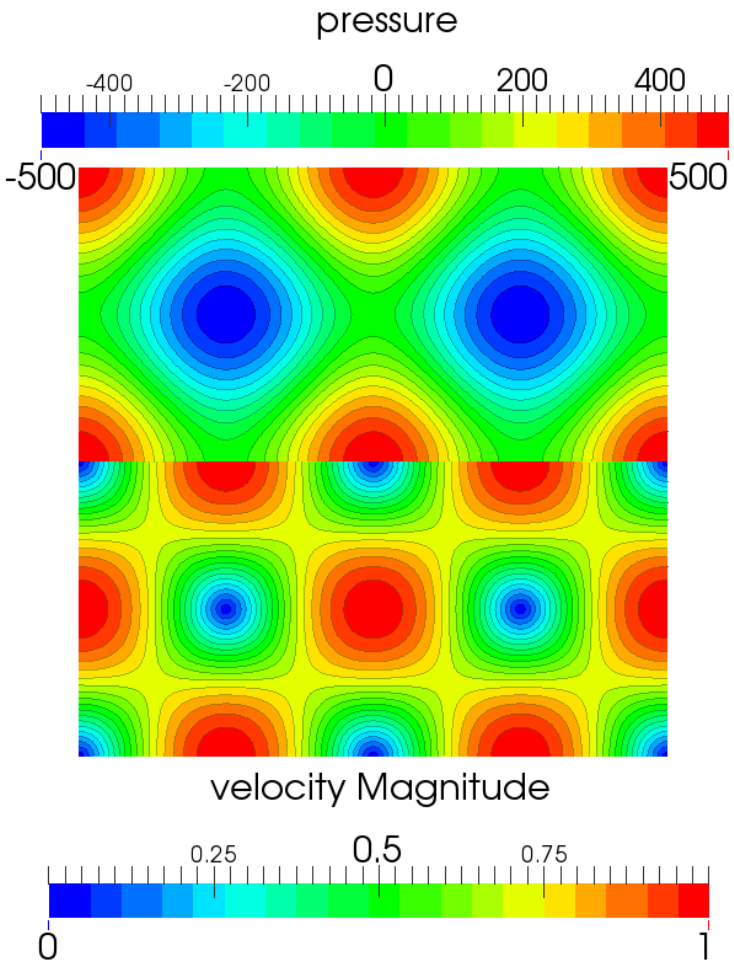


Figure 3.22: Pressure and magnitude of the velocity field of the Taylor-Green vortex computed with the weakly-compressible FV solver on 150×200 cells on the unit square. The results are given after $t = 10s$.

Taylor-Green vortex

We have seen in Section 2.1 that the inviscid Taylor-Green vortices are given by a closed-form solution of sine and cosine functions, Eq. (2.1). The pressure and velocity fields show the same length scales along the x axis and the y axis. As a consequence, using a uniform Cartesian grid would be a good choice for this test case. However, we consider a regular mesh with 150×200 cells in the unit square in order to stress the solver. Figure 3.22 shows the pressure and the magnitude of the velocity computed with the MUSCL scheme (3.85). The results are shown for a physical time of $t = 10s$.

Furthermore, Figure 3.23 shows the velocity gradients plotted on the line $y = 0.5$m, where only $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial y}$ are non-zero. They are computed by the solver from the analytical solution of the velocity field. The computed gradients, Eq. (3.86), correspond very well to the gradients that are obtained analytically, Eqs. (A.3) and (A.6).

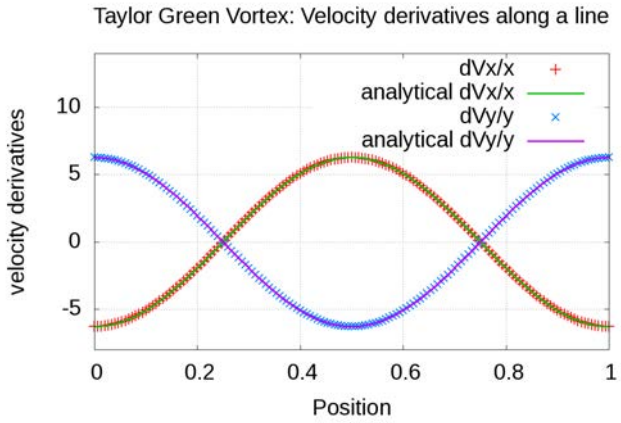


Figure 3.23: Derivatives of the velocity computed by the solver from the analytical velocity field. Only the derivatives $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial y}$ are non-zero on the line $y = 0.5$m.

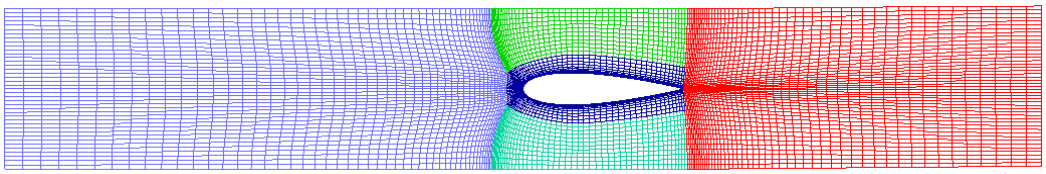


Figure 3.24: A structured multi-block mesh is created for the FV reference solution. A block of C-grid together with a dove-tail topology is used around the NACA hydrofoil.

Inviscid flow around a NACA hydrofoil

The testcase of the inviscid flow around a symmetric NACA hydrofoil that was introduced in Section 2.2 is reused in this section for the validation of the FV solver. Figure 3.24 shows the mesh that used for the simulation. It consists of several blocks and it was already presented in Chapter 2, Section 2.2.

Figure 3.25 compares the magnitude of the velocity field obtained with the FV solver to the reference solution. The obtained velocity field corresponds well to the reference solution. It shows the same shape and the same range of values. Moreover, it shows almost no numerical wake because of the mesh refinement close to the hydrofoil in the normal direction to the solid boundary.

In Figure 3.26 the pressure field can be seen that compares equally well to the reference solution. A more quantitative analysis is given by Figure 3.27 that shows the pressure coefficient, Eq. (2.3), at the hydrofoil, obtained with the FV solver and the FV reference solution. Again good agreement is found.

3.11 Summary and discussion

In this chapter two numerical methods, the meshless method SPH-ALE and the meshbased FVM, were introduced. It was found that discretizing the Euler equations in ALE form by FV operators or by SPH operators yields two systems of semi-discrete equations that manifest

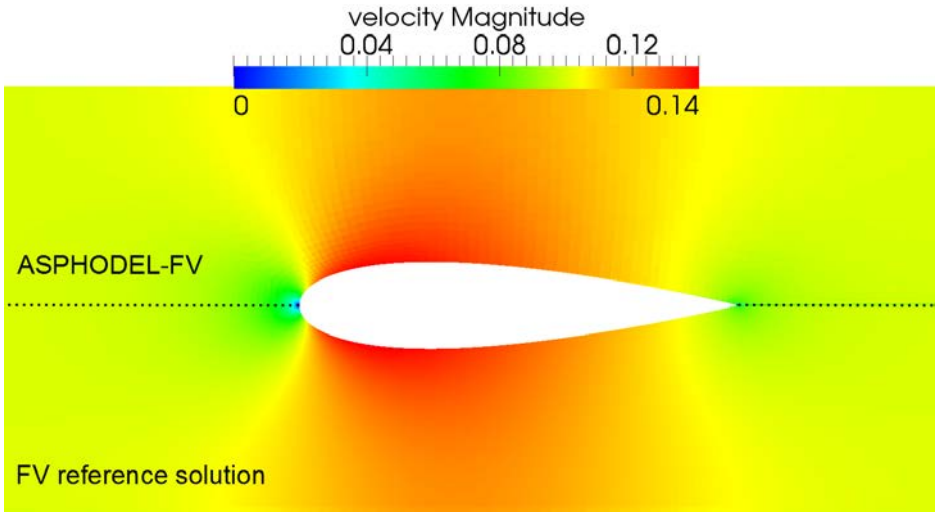


Figure 3.25: Magnitude of the velocity field around a static NACA hydrofoil. The result obtained with the implemented weakly-compressible FV solver (ASPHODEL-FV) is compared to the FV reference solution.

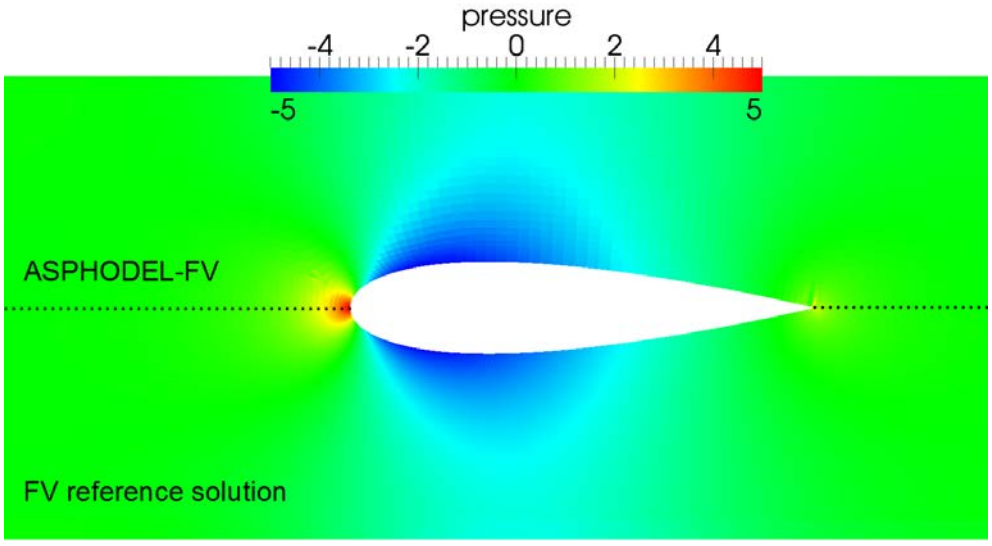


Figure 3.26: Pressure field around a static NACA hydrofoil. The result obtained with the implemented weakly-compressible FV solver (ASPHODEL-FV) is compared to the FV reference solution.

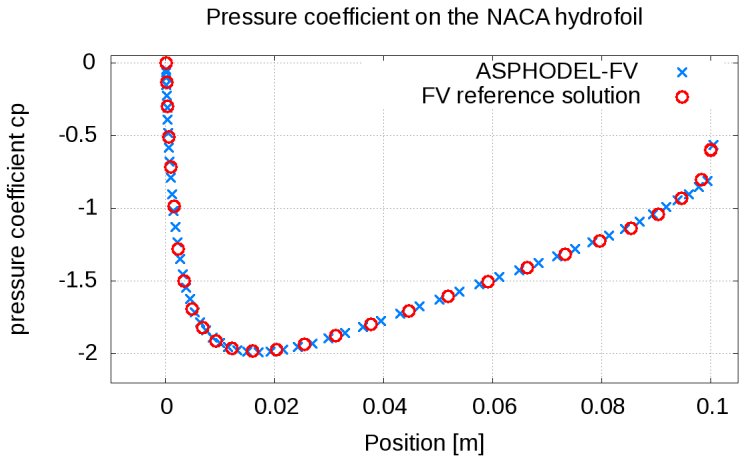


Figure 3.27: Pressure coefficient plotted at the hydrofoil. The implemented weakly-compressible FV solver (ASPHODEL-FV) is compared to the FV reference solution.

important similarities. First, both methods sum numerical fluxes over neighbours. Second, these numerical fluxes can be determined in the same way, i.e. by local one-dimensional Riemann problems with reconstructed initial states obtained by the MUSCL scheme. Third, the same time integration schemes can be used to advance the equations in time. The main difference between these two methods lies in the way how neighbours are defined. FVM is a meshbased method where the neighbours are the adjacent cells, while SPH-ALE is a meshless method where the neighbours are particles that lie in a compact region around the considered particle, depending on the so-called smoothing length.

The meshless character of SPH-ALE is the reason for its strong points and its weaknesses. On the one hand, SPH-ALE is a flexible methods where the particles can move in a Lagrangian way, in an arbitrary way or they can be fixed in space (Eulerian description) since no connectivity information between particles is necessary. On the other hand, a SPH particle in 2D has about five times more neighbours than a cell in a structured FV mesh which makes SPH computationally expensive. Moreover, the exact geometrical shape of a particle and the exact position of the interface between two particles is unknown.

As a consequence, SPH suffers from a lack of consistency and not even zeroth order consistency is obtained for the semi-discrete SPH-ALE equations. This introduces errors into the pressure field that pollute the results considerably. Therefore, a novel correction method was proposed in this work to correct the gradient of the kernel function in such a way that zeroth order of consistency for the divergence of the flux vector is recovered. The benefit of this correction method, that was named *closed box correction*, was shown by means of a fluid at rest case and the Taylor-Green vortex on two different particle distributions.

In the frame of the thesis, an unstructured FV solver was implemented in the in-house SPH-ALE software ASPHODEL of ANDRITZ Hydro. Validation of the solver by the Taylor-Green vortex and the flow around a static symmetric NACA hydrofoil was shown in the last part of this chapter.

This FV solver together with the corrected SPH-ALE method constitutes the basis for the developed coupling strategy in Chapter 4.

Chapter 4

Coupling of SPH-ALE with a Finite Volume method

We have seen in Chapter 1 that SPH is more and more used for industrial applications in the last years. Especially for free surface flows with moving geometries like for example the flow in the runner of hydraulic Pelton turbines [71], the use of SPH or variants of SPH is advantageous compared to traditional meshbased numerical methods. But also for internal flows without free surfaces there are applications where SPH has great potential because of its meshless nature that treats geometries moving in a complex way without remeshing or mesh interfaces.

However, there is one important drawback of the SPH method. It is difficult to have non-constant particle sizes, i.e. particle refinement is complicated for particles in Lagrangian motion, in particular, anisotropic refinements. SPH is an intrinsically isotropic method with isotropic particles and a kernel function with an isotropic support but flow phenomena often necessitate fine space discretization only in one direction. Close to solid walls for example many particles are needed in direction normal to the wall but not tangential to the wall. In Finite Volume methods (FVM) long thin cells are used in this region that can feature aspect ratios of ten or more.

In the past, some work has been done to develop anisotropic kernel functions for SPH (see Section 4.1.1.2) but this is not the approach that this work takes as will be explained below.

We present a flexible coupling method of SPH-ALE with FVM that allows us to refine anisotropically in space and to benefit of the other advantages of mesh based methods like imposing open boundary conditions more easily. Section 4.2 describes the coupling algorithm that is validated by means of 1D and 2D academical testcases in the subsequent section.

We have seen in Chapter 3 that even though SPH-ALE is a meshless method, there are important similarities with FVM. Hence, it suggests itself to adapt approaches to SPH that were first developed for FVM. In the 1980s the bottleneck of meshbased methods applied to industrial problems was the mesh generation and multi-domain methods were developed to make it possible to subdivide the computational domain into subdomains where the mesh can be created independently. These methods can be reinterpreted for the coupling of FVM with other methods by applying different solvers to the subdomains. For that reason multi-domain methods for FVM are reviewed in Section 4.1.2.1. Emphasis is placed on one family of these methods that is called Chimera methods and that was the inspiration for the development of the present coupling method. It uses overlapping meshes where information is transferred from one mesh to the other by interpolated boundary cell values [9]. In general, these methods are very robust but not conservative but for the case where conservative interpolation methods are employed. The disadvantages of non-conservative methods and the difficulties of developing

conservative methods on overlapping domains are discussed in the following, as well as the different approaches found in literature to overcome that problem are presented.

4.1 Literature review

In the first part of this section, Section 4.1.1, we discuss how several authors realized non-uniform space discretization in SPH. Then some developments on an anisotropic SPH method with kernel functions with an elliptic support are reviewed in Section 4.1.1.2. In the second part, Section 4.1.2, multi-domain approaches in FVM are discussed with regard to their applicability to the coupling of SPH and FVM. In the third part of this section, Section 4.1.3, different techniques for coupling SPH with other mesh-based methods are cited and compared to the present method.

4.1.1 Variable resolution methods for SPH

The review of particle refinement techniques includes variable resolution schemes and dynamic refinement and de-refinement strategies. These approaches are discussed in quite some detail because they show us not only how refinement in space can be implemented without coupling SPH to an other method, but also because these techniques have some similarities with the present coupling method. This is especially true for Barcarolo's method of refinement and de-refinement [5] that can be interpreted as a method for overlapping domains with different sized particles.

4.1.1.1 Isotropic refinement of SPH particles and kernels

It was mentioned previously that traditionally all SPH particles in the computational domain have the same size and the same smoothing length h. We have seen in Section 3.2.4 that the smoothing length is in practice often chosen by

$$h = 1.2\,\Delta x. \tag{4.1}$$

However, physical flow phenomena do not necessitate the same space discretization size everywhere in the computational domain and it is computationally expensive to launch simulations with a fine constant particle size everywhere. For that reason, many authors work on implementing SPH schemes with varying particle sizes. But not only the particle size can be reduced, also the smoothing length might be adapted because the number of neighbours per particle varies strongly if the particles are refined without changing the smoothing length. If the number of neighbours should be kept approximatively constant, the smoothing length h has to be adapted according to the particle size by Eq. (4.1) or a similar relation. However, varying smoothing length implies non-zero gradients of h which means that the ∇h terms should be included into the computation of the gradient of the kernel function ∇W,

$$\nabla W\left(q = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{h}\right) = \frac{\partial W}{\partial q}\left[\frac{\partial W}{\partial \|\mathbf{x}_i - \mathbf{x}_j\|}\nabla\|\mathbf{x}_i - \mathbf{x}_j\| + \frac{\partial W}{\partial h}\nabla h\right]. \tag{4.2}$$

It is not obvious how to compute ∇h and some possibilities and ideas are discussed below.

The different approaches for variable resolution schemes found in literature can be classified in three main categories that will be discussed in the following paragraphs.

- The first one initialises particles with varying sizes and smoothing length that then move in a Lagrangian way without any special treatment [83, 72]. In this case the particle size depends on the initial position of the particle and Ulrich *et al.* [106] call it a *Lagrangian variable-resolution approach.*

- The second one is an *Eulerian variable-resolution strategy* introduced by Ulrich [105] that prescribes a spacing between particles depending on their instantaneous position.
- The third one is a *dynamic refinement (and de-refinement) strategy* that splits particle according to certain criteria that can be physical or geometrical [28, 108, 5].

Lagrangian variable-resolution approach

Lagrangian variable-resolution approaches initialize particles non-uniformly according to predefined refinement zones. Since the particle distribution is not adapted during the simulation, these techniques are only appropriate if particles stay near their initial position. Either they are in Eulerian motion (only possible for SPH-ALE), or the flow has no high mean flow velocity.

Omidvar [84] uses a variable particle mass approach with predefined zones where at the beginning of the simulation one particle is replaced by four lighter particles. The mass is equally distributed and the sum of the new particles has the same mass as the original particle. The smoothing length is kept constant and is determined by the heavier particles and there is a sharp transition between lighter and heavier particles. In addition, lighter and heavier particles might mix in the case of dynamically moving flows which can lead to numerical instabilities.

Starting from a refined zone of constant h_0 in the form of a semicircle or a rectangle, Oger *et al.* [83] de-refine the particles by defining the smoothing length of the ith particle counted in rows from the boundary of the constant h_0 area as $h_i = \delta^i h_0$. The ∇h term is not included in the computation of $\nabla_i W$ and the constant $\delta = 1.03$ is chosen in a way that the absence of the ∇h term does not introduce visible errors. Marsh *et al.* [72] improve this method by adding an approximation for the ∇h term. Given that far away from boundaries the SPH approximation of ∇h is

$$< \nabla h_i >= -\sum_j \omega_j h_j \nabla_i W(\mathbf{x}_i - \mathbf{x}_j, h_i),$$

it is proposed to approximate

$$\nabla h_i \approx -\omega_j h_j \nabla_i W(\mathbf{x}_i - \mathbf{x}_j, h_i),$$

where the gradient of the kernel function $\nabla_i W(\mathbf{x}_i - \mathbf{x}_j, h_i)$ is computed without taking into account the contributions from ∇h. The new gradient of the kernel function is then obtained by symmetrization, i.e.

$$\begin{aligned}\widetilde{\nabla_i W}_{ij} := &\frac{1}{2}\left(\nabla_i W(\mathbf{x}_i - \mathbf{x}_j, h_i) + \nabla_i W(\mathbf{x}_i - \mathbf{x}_j, h_j)\right) \\ &+ \frac{1}{2}\left(\frac{\partial W(\mathbf{x}_i - \mathbf{x}_j, h_i)}{\partial h}\nabla h_i + \frac{\partial W(\mathbf{x}_i - \mathbf{x}_j, h_j)}{\partial h}\nabla h_j\right). \end{aligned} \tag{4.3}$$

The gradient $\widetilde{\nabla_i W}_{ij}$ of Eq. (4.3) is inserted into the SPH-ALE equations, Eq. (3.80), in order to compute the physical fields. According to [72], this formulation also works if the smoothing length is approximately constant and has the nice side effect of distributing the particles more regularly than in the case of standard Lagrangian simulations.

Eulerian variable-resolution approach

Ulrich *et al.* [105, 106, 107] develop an approach of particle refinement applied to XSPH (see Section 3.2.5) called Eulerian variable-resolution technique that prescribes a particle spacing with constant gradient $\nabla(\Delta x)$ and then computes the gradients of the smoothing length ∇h

and of the particle mass ∇m analytically there-from. The changing particle mass introduces source terms in the continuity equation and the momentum equation. In [105] it is shown by means of a hydrostatic tank simulation that it is necessary to include the ∇h term into the computation of the gradient of the kernel function, see Eq. (4.2), in order to obtain a stable hydrostatic pressure distribution. In the case of this Eulerian variable resolution approach, ∇h is evaluated analytically.

Dynamic refinement (and de-refinement) strategy

Feldman [28] proposes a particle splitting scheme for standard WCSPH (see Section 3.2.5) where bigger "mother" particles are split into several smaller "daughter" particles according to a predefined splitting pattern. The new particle size and the new smoothing length are given by a set of splitting parameters $(\varepsilon, \alpha) \in [0,1] \times [0,1]$ with $\Delta x_d = \varepsilon \Delta x_n$ and $h_d = \alpha h_n$, where the subscript d stands for the daughter particle and n for the mother particle. The mass of the daughter particles is chosen in order to minimize the global density error introduced by the splitting and so that global mass is conserved. Feldman uses the *scatter density formulation*, see Eq. (3.52), to compute the density and a local density refinement error is defined as the change of density due to the introduction of new particles. The velocity of the mother particles is assigned to the daughter particles $\mathbf{v}_d := \mathbf{v}_n$ which represents the unique fully conservative velocity distribution. This method can be applied *dynamically* where particles are split during simulations according to certain refinement criteria that can be physical criteria like strong gradients of the physical field variable, or geometrical criteria where particles are split when entering refinement zones.

Vacondio *et al.* [108] apply Feldman's splitting scheme to shallow water SPH in order to simulate flooding. The method is further extended to a de-refinement procedure that consists of coalescing (merging) two particles in a conservative way without introducing density errors. The particle splitting and merging is extended to a 3D SPH scheme in Vacondio *et al.* [109]. It is also shown that the optimal splitting pattern replaces one mother particle by 13 daughter particles, where one is situated at the position of the mother particle and the other 12 particles are located at the vertices of a icosahedron. According to the analysis in the paper, the refinement pattern of the form of a cube behaves less well than the icosahedron and should be avoided. No matter what refinement pattern is chosen, they show that a particle has to be kept at the position of the original mother particle.

Nevertheless, López *et al.* [63] proposed a two-dimensional splitting pattern where the daughter particles are situated on a square centered around the mother particle and mass is distributed uniformly, i.e. $m_d = 0.25\, m_n$. This pattern does not include a particle at the position of the original mother particle. In contrast to Feldman [28], they use the *continuity density formulation* and cannot directly apply the same definition of the density error. A more general definition of the refinement error is given instead. That is the error introduced by the splitting to the SPH approximation of the gradient of any function f. That makes the method applicable to all SPH variants even to those that do not use the scatter density formulation. The density refinement error can be defined corresponding to the SPH method that is employed and in particular to the way the density is calculated.

Barcarolo [5, 6] validates this method for SPH-ALE (what they call Riemann SPH) and δ-SPH and propose a novel de-refinement procedure that does not coalesce particles but keeps the mother particles throughout the whole simulation and simply deletes the daughter particles when they leave the refinement zone. If there are daughter particles, the mother particle is advanced passively in time. That means that the flow equations are solved for the particle but that it does not contribute itself to the integration of the others. For that purpose, Barcarolo

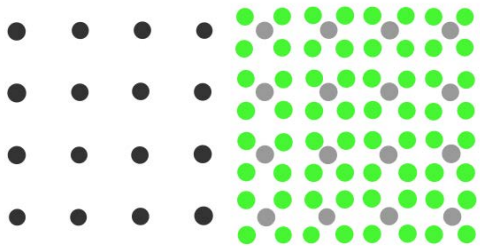


Figure 4.1: Adaptive particle refinement [5] where coarse mother particles (black points) that enter a predefined region are split into four daughter particles (green points) but the mother particles are still kept and moved passively (grey points). The daughter particles that leave the refinement zone are deleted and the mother particles are active again (black points).

introduces a flag γ that is $\gamma = 0$ if the particle is passive (OFF) and $\gamma = 1$ if the particle is active (ON). The SPH approximations become for a function f and its gradient ∇f

$$f_i = \sum_j \omega_j \gamma_j \, f_j W_{ij}, \tag{4.4}$$

and

$$\nabla f_i = \sum_j \omega_j \gamma_j \, f_j \nabla_i W_{ij}. \tag{4.5}$$

Barcarolo points out that the sharp transition between $\gamma = 0$ and $\gamma = 1$ can introduce additional errors in the pressure field in some simulations, especially in cases of slow flows. Therefore, a transition zone is added where γ follows a smooth function defined by geometrical properties depending on the position of the particle in the transition zone. For the mother particle, the parameter $\gamma_n \in [0, 1]$ is defined by the ratio of the distance of the particle to the zone of constant γ and the size of the transition zone. For the daughter particle $\gamma_d = 1 - \gamma_n$ is used.

Barcarolo calls this method *adaptive particle refinement* (APR) in correspondence to the adaptive mesh refinement (AMR) method that is discussed in Section 4.1.2. Figure 4.1 shows that particles are refined in a similar way than cells that are refined in AMR, see Figure 4.4. He points out that AMR refines space and time equally which is not yet the case for APR. In addition, AMR is a conservative method in contrast to APR.

In the context of the present thesis, we are especially interested in Bacarolo's method because it can be interpreted as a method of overlapping domains where a domain with a finer discretization as the other (green points in Figure 4.1) is completely overlapped by the domain with the coarse discretization size (black and grey points in Figure 4.1). (This is also true for AMR, see Section 4.1.2.) As a consequence, similar questions arise as in the present coupling method. In particular, the SPH particles that are overlapped by the FV domain during the coupling are treated similarly to the passive "mother" particles in the refinement zone, since their fields are computed but they do not contribute themselves to the integration of the FV cells.

4.1.1.2 Anisotropic kernels

The methods that were introduced in the above Section 4.1.1.1 refine particles isotropically and only employ kernel functions with an isotropic support. Schick [98] develops in his diploma

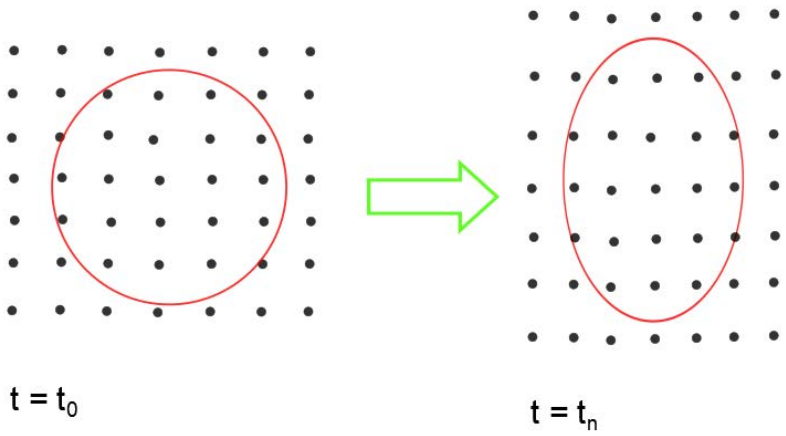


Figure 4.2: Evolution of the kernel support area in time [98, 39]. The initially circular kernel support area follows the deformation of the particle distribution and becomes anisotropic of the form of an ellipse. The number of neighbours stays approximately the same in every direction along the axes of the ellipse.

thesis adaptive anisotropic kernels that are then further improved by Guilcher [39] and applied to the SPH modelling of shallow-water flows by de Leffe [23]. In their approach the smoothing length h is replaced by a symmetric positive definite matrix $\mathbf{H} \in \mathbb{R}^{d\times d}$ that defines a norm,

$$\|\mathbf{x}\|_H = \sqrt{\mathbf{x}^T \cdot \mathbf{H} \cdot \mathbf{x}}.$$

All $\mathbf{x} \in \mathbb{R}^d$ with

$$\|\mathbf{x}\|_H = \text{const}$$

lie on an ellipse with center 0 and axes h_1 and h_2 in 2D and on an ellipsoidal in 3D. The anisotropic kernel function W_i of $\mathbf{x}_i$ is given by

$$W_i(\mathbf{y}, \mathbf{H}) := \frac{1}{vol(P_i)} \theta\left(\|\mathbf{y} - \mathbf{x}_i\|_{H_i}\right),$$

where $vol(P_i)$ is the volume of the elliptic support

$$P_i(t) := \{\mathbf{y} \in \mathbb{R}^d : \|\mathbf{y} - \mathbf{x}_i\|_{H_i} \leq 1\},$$

with $\mathbf{H}_i := \mathbf{H}(t, \mathbf{x}_i)$. The function θ is the same as in the case of the circular kernel function defined in Chapter 3, Section 3.2.3, for the 3rd order B-Spline or the Wendland kernels. If $h_1 = h_2$, the kernel function with spherical support is recovered. The matrix $\mathbf{H}$ depends on the time t and changes in a way that not only the number of neighbour particles is kept approximately constant but also the number of neighbours in each direction varies little. In [39] the evolution of $\mathbf{H}$ is given by

$$\frac{d\mathbf{H}_i}{dt} = -\left(\mathbf{G}_i^T \cdot \mathbf{H}_i + \mathbf{H}_i \cdot \mathbf{G}_i\right), \quad \mathbf{G}_i = \nabla \mathbf{v}_0(\mathbf{x}_i),$$

where $\mathbf{v}_0$ denotes the transport velocity. Figure 4.2 shows an example where an initially circular kernel support area deforms into an ellipse during the simulation.

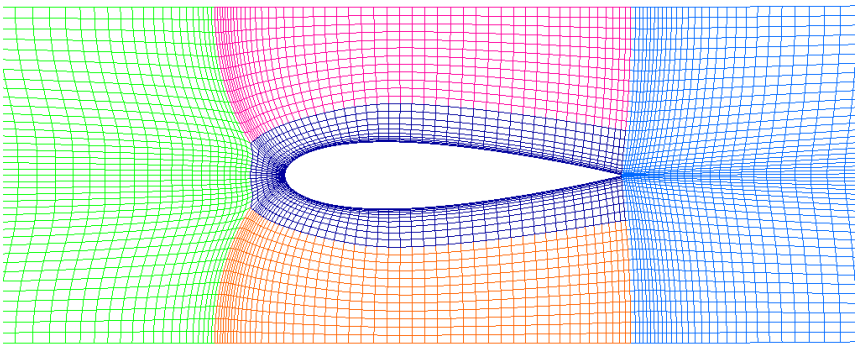


Figure 4.3: Example of a multi-block mesh around a symmetric NACA hydrofoil. There are five different blocks colored differently that have matching boundaries, i.e. all mesh lines are continuous at the block boundaries.

Even though this is an interesting approach, it will not be further investigated in the following. The industrial applications, that we are interested in, necessitate many points in direction normal to solid walls and much fewer in tangential direction. This would lead to very long and thin elliptical kernel support areas with high aspect ratios whose numerical behavior, to our knowledge, has not been investigated yet in literature. In FVM block structured meshes for example allow us to use cells with high aspect ratios. Furthermore, it is possible in FVM to refine quite abruptly with a high refinement ratio from one cell to the other according to well-established best practice guidelines for mesh construction. Refining an SPH domain in an equal way would lead to strong variations of the minor or major axes of the ellipses and therefore to strong gradients of the smoothing length $\mathbf{H}$. For that purpose Eq. (4.2) would have to be adapted to the anisotropic formalism in order to be taken into account for the computation of the gradient of the kernel function with all the connected difficulties of variable h formulations that were discussed above, in Section 4.1.1.1.

Instead of a pure SPH solution, anisotropic refinement will be added by a coupling with a FV method that uses meshes with anisotropic hexahedral cells because we felt more confident in obtaining an efficient coupling method in the time frame of this PhD project than an equally efficient anisotropic SPH solver.

4.1.2 Variable resolution and multi-domain methods for FVM

In Chapter 3 SPH-ALE was introduced and it was explained that SPH-ALE has many similarities with FV formalisms. Hence, it is not surprising that techniques that were first developed for FVM are often adapted successfully for SPH. In the following, different multi-domain approaches for FVM are discussed with a special emphasis on methods for overlapping grids. At the beginning of the 1980s increasing computational power and advanced numerical and physical models made it possible to launch more and more complicated simulations. So, the mesh generation became the bottleneck of meshbased methods applied to industrial problems. As a consequence multi-domain methods were developed to make it possible to subdivide the computational domain into subdomains where the mesh can be created independently. In particular, there were two types of methods that appeared in that time, i.e.

- *grid patching*: The computational domain is divided into a set of disjoint subdomains with common boundaries. The mesh of each subdomain is easier to generate than one mesh

for the whole domain. In general, mesh lines through the boundaries between subdomains are discontinuous and interpolation procedures are needed for the transfer of information from one subgrid to the other. These interpolations are conservative. A special case of patched grids are multi-block grids that have several blocks of curvilinear bodyfitted meshes with matching boundaries, i.e. continuous mesh lines. In this case no interpolation between the subdomains is necessary but the mesh generation is more demanding than for the non-matching case. In Figure 4.3 a multi-block mesh around a symmetric static NACA hydrofoil is shown. There are five different blocks with matching block boundaries. Multi-block meshes are often used for turbomachinery applications.

- *grid embedding* or *overset grids* or *overlapping grids*: Again the computational domain is divided into subdomains but this time the domains are overlapping. Grid generation is even easier than for patched grids because each grid can be generated completely independently from the other and no matching boundaries are needed. The only constraint is the size of the overlapping region that has to be large enough to guarantee information transfer from one grid to the other and vice versa which is done by interpolation. This approach will be discussed in detail below.

Another important approach originates in that time that is strictly speaking no multi-domain method but is still connected to these approaches,

- *grid adaptation* or *adaptive mesh refinement* (AMR) [12]: In the main, there are two different types of adaptive refinement. There is the *moving grid points methods* that adapts the grid distribution by moving mesh lines into one region and leaving a coarser region somewhere else. This method tries to get the most accurate solution by a fixed number of mesh points. However, it is not easy to maintain a smooth grid and the number of necessary grid points has to be guessed at the beginning of the simulation. The second method is a *local grid refinement* that adds new points in regions where a local error estimation is too big or where a physical criteria is fulfilled and removes them if necessary. This method tries to reach a fixed accuracy by a minimum number of mesh points. A drawback is that the equations have to modified at the interfaces between the coarser and refined zones in order to preserve conservation. The grids are also refined in time by the the same refinement ration which means that

$$\frac{\Delta t_{coarse}}{\Delta x_{coarse}} = \frac{\Delta t_{fine}}{\Delta x_{fine}},$$

and the same explicit difference schemes are stable on all grids. In practice, this implies that more time steps are taken on the finer grids than on the coarser ones. In [11] only two-dimensional rectangular grids are considered that are refined by other rectangular grids but the method has been generalised afterwards. Figure 4.4 shows coarse cells that are refined by rectangular cells with a constant refinement factor

$$R = \frac{\Delta x_{coarse}}{\Delta x_{fine}}.$$

The solution ϕ_{ij} on the coarse cell with index (i, j) is computed by

$$\phi_{ij}^{coarse} = \frac{1}{R^2} \sum_{p=0}^{R-1} \sum_{q=0}^{R-1} \phi_{k+p,m+q}^{fine},$$

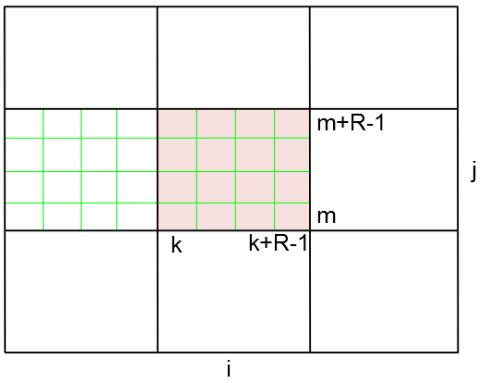


Figure 4.4: Coarse rectangular cells are refined by fine rectangular cells [12]. The cell with index (i, j), colored in red, is refined by the cells of indices $k, ..., k + R - 1$ and $m, ..., m + R - 1$, where R denotes the refinement factor. The physical fields of cell (i, j) are computed by Eq. (4.6).

where the indices are taken from Figure 4.4. This formula is written for two-dimensional rectangular grids, but it can be rewritten in a more general manner

$$\phi_{ij}^{coarse} = \sum_{p=0}^{R-1} \sum_{q=0}^{R-1} \frac{V_{fine}}{V_{coarse}} \phi_{k+p,m+q}^{fine}, \tag{4.6}$$

which is a formula that we will find again in several other methods including in the present coupling method for particles covered by FV cells that are close to solid walls.

Berger [11] points out that AMR can be interpreted as a special case of overlapping grids with aligned boundaries. The fine grid with green grid lines in Figure 4.4 is overlapping the coarse grids with black mesh lines. However, it can equally be interpreted as a special case of patched grids where the fine grid with green mesh lines is patched into the coarse grid. Hence, AMR can be considered as one of the multi-domain approaches.

In [10] a combination of AMR and the Chimera method was developed. The background grids are refined by moving overlapping grids that are automatically generated by solving an optimization problem. The improvement compared to AMR is that the overlapping grids do not have to be aligned with the background grid but can be oriented in a way that physical phenomena are better captured. A successful application to blade vortex interactions was shown by the author.

4.1.2.1 Overlapping grids

In this document, FVM for overlapping grids are discussed in more detail because it is important for the understanding of the presented coupling algorithm. In the following, we consider a computational domain $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, 3$ denotes the space dimension. The domain Ω is subdivided into two (or more) subdomains Ω_1 and Ω_2 that are overlapping each other, i.e.

$$\Omega_1 \cap \Omega_2 = \Omega_{12} \neq \emptyset. \tag{4.7}$$

The Chimera method

In 1983, Benek *et al.* [8, 9] published a method for overlapping grids applied to the solution of the Euler equations for aerodynamics in 2D and 3D. In order to simplify mesh generation, component parts are meshed independently. Then the component grids are put together by an underlying background grid in which they are embedded. Figure 4.5 shows the example of a curvilinear grid G_2 around a cylinder that is embedded into a rectangular grid G_1. The equations are solved independently on each grid and communication between them is done by interpolating boundary cell values from one grid to the other. The Chimera boundary cells are indicated in grey in Figure 4.5. Hence, only the minimum overlap that is necessary for the interpolation method is used. Other overlapped cells, that are in the interior of the embedded grid and that are not necessary for the interpolation of the boundary value, are disactivated. That means that the component grids introduce "holes" into the background grids. In [8] a grid hierarchy and an algorithm for hole generation is described.

The authors named their method *Chimera method* after the legendary creature of Greek mythology that is composed of different animal parts. The Chimera method is very flexible and can bring together different types of grids that were generated independently. In the original method only structured grids were used because the aim was the flexible refinement without loosing the advantages of structured grid. Nevertheless, extensions to mixed grids, unstructured and structured, were published thereafter [113].

Due to the fact that the method was extended from the compressible Euler equations to conservation laws in general [11] and that the individual grids are independent, it is not necessary to solve the same set of equations on each subdomain. Already in [9] it was proposed to solve the Navier-Stokes equations on body fitting meshes around the solid geometries and the Euler equations in the interior of the fluid domain. Hence, viscous effects are only taken into account in the boundary layer where they are indispensable for many flow simulations. Away from these regions viscous effects are less important and computation time can be saved by solving the inviscid Euler equations. Pärt-Enander [87] successfully applies a Navier-Stokes/Euler coupling to supersonic and hypersonic compressible flows on overlapping grids.

Almost immediately after the publication of the Chimera method, some discussion started about the interpolation technique to employ. In the original paper [9] a Taylor series approximation was proposed, then a bi-linear for 2D (and tri-linear in 3D) interpolation was found to be better [8]. However, both methods are not conservative. This presents a problem especially if shocks appear in the simulation because the shock speed is not computed correctly at the interface if a non-conservative interpolation is used. In the case of slowly travelling shocks that are parallel to the mesh boundary, shocks might not even pass through the interface. That is the reason why conservative interpolation methods were developed by several authors that are discussed below.

Conservative interpolation and stability

In this section we follow [87, 88] for the classification of different interpolation methods. For simplicity we consider a scalar one-dimensional model problem,

$$\begin{cases} \frac{\partial}{\partial t}\varphi + \frac{\partial}{\partial x} f(\varphi) = 0, \quad -\infty < x < \infty, \ t \geq 0, \\ \varphi(x,0) = \varphi_0(x), \end{cases} \tag{4.8}$$

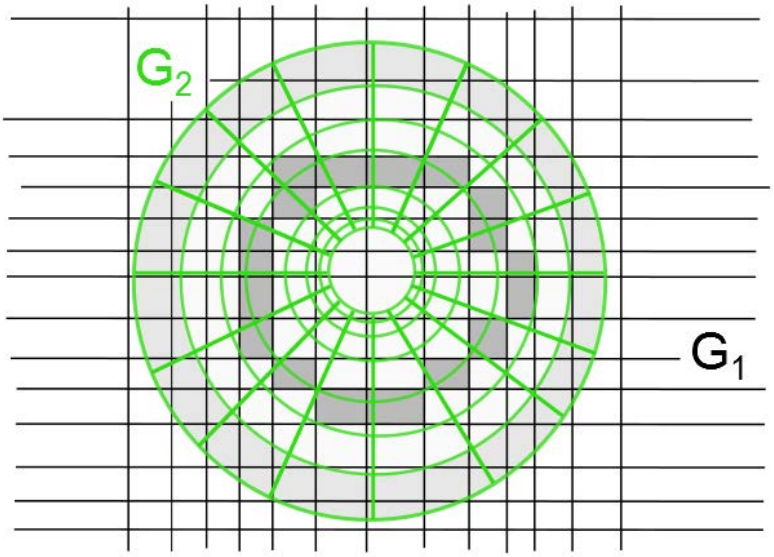


Figure 4.5: Two 2D grids that are overlapping each other. The rectangular grid G_1 with the black mesh lines is the grid that envelops the curvilinear component grid G_2 with green mesh lines. The original Chimera algorithm detects which cells of G_1 are covered by G_2, determines the minimum overlap necessary for the interpolation and excludes the mesh points of G_1 that lie in the interior of G_2. That means that the embedded grid G_2 (green) introduces a hole into G_1 and only boundary cell values are interpolated. The dark grey cells are interpolated from G_2 while the light grey cells are the boundary cells of G_2 that are interpolated from G_1. The cells of G_1 that are inside the zone that is limited by dark grey boundary cells are disactivated.

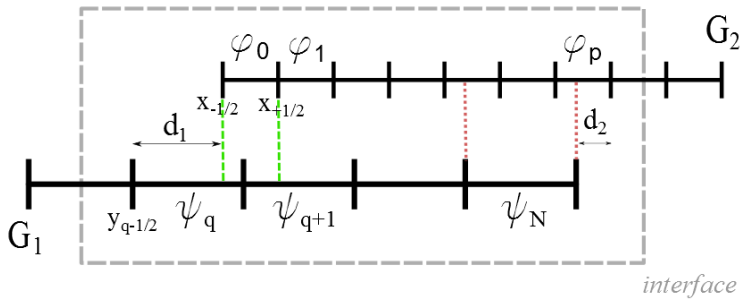


Figure 4.6: Two 1D grids that are overlapping each other [87]. Grid G_1 has uniform cells of size h_1 and G_2 has uniform cells of size h_2. The cell with center y_N is the boundary cell of G_1 and the cell with center x_0 is the boundary cell of G_2. This is where information between the grids has to be exchanged.

discretized on two uniform overlapping grids G_1 and G_2 with a discretization size of h_1 and h_2. For the inner cells we apply a conservative numerical scheme of the form

$$\begin{aligned} \psi_j^{n+1} &= \psi_j^n - \tfrac{\Delta t}{h_1}\left(f_{j+\frac{1}{2}}^n - f_{j-\frac{1}{2}}^n\right), \quad y_j \in G_1, \\ \varphi_j^{n+1} &= \varphi_j^n - \tfrac{\Delta t}{h_2}\left(g_{j+\frac{1}{2}}^n - g_{j-\frac{1}{2}}^n\right), \quad x_j \in G_2. \end{aligned} \tag{4.9}$$

We denote the numerical flux on G_1 with $f_{j+\frac{1}{2}}^n = f(\psi_j^n, \psi_{j+1}^n)$ and the numerical flux on G_2 with $g_{j+\frac{1}{2}}^n = g(\varphi_j, \varphi_{j+1}^n)$. Figure 4.6 shows a sketch of two 1D overlapping grids and the notation that we use. The cells with centers $x_0 \in G_2$ and $y_N \in G_1$ are the two boundary cells. In this example, only one layer of cells is interpolated for each grid. Depending on the stencil of the numerical scheme that is used, two or more layers of interpolated cells at the boundary might be needed.

In the main, three different approaches of interpolation are distinguished as follows.

- Most industrial applications use a *non-conservative method* that interpolates the cell values. The cell value of the cell with center x_0 is determined by

$$\varphi_0^{n+1} = (1-\omega_1)\psi_q^{n+1} + \omega_1 \psi_{q+1}^{n+1}, \tag{4.10}$$

  with the interpolation weight $\omega_1 = \frac{x_0 - y_q}{h_1}$ that are chosen in a way that the cell in G_1 where the center y_q or y_{q+1} is closer to x_0 contributes more to φ_0^{n+1} than the other cell.

- Berger [11] proposed a *conservative method* that interpolates the fluxes by

$$\tilde{g}_{-\frac{1}{2}}^n = (1-\omega_1) f_{q-\frac{1}{2}}^n + \omega_1 f_{q+\frac{1}{2}}^n, \tag{4.11}$$

  with the interpolation weight $w_1 = \frac{d_1}{h_1}$. The interpolated flux $\tilde{g}_{-\frac{1}{2}}^n$ replaces $g_{-\frac{1}{2}}^n$ in Eq. (4.9). This method reproduces the correct shock speed but Pärt-Enander [87, 88] showed that it is not stable and the solution blows up if the interface is an outflow boundary. Hence, it depends on the orientation of the characteristics at the interface. Therefore, a characteristic decomposition of the interpolated fluxes was proposed depending on the flow direction at the boundary. In addition, a filter has to be applied that removes non-physical oscillations in front of shocks.

- Another non-conservative method is the *reconstruction-integration method* that uses a piecewise linear reconstruction of the solution, known from the MUSCL scheme, see Section 3.6.3. The reconstructed function on G_1 is given by

$$\psi(x) = \psi_j^n + \frac{s_j}{h_1}(x - x_j), \quad x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}},$$

where the slope s_j of cell j is limited by a well established slope limiter like the *minmod* function. The boundary cell value φ_0^{n+1} is evaluated by integrating over the reconstruction,

$$\varphi_0^{n+1} = \frac{1}{h_2} \int_{x_{-\frac{1}{2}}}^{x_{\frac{1}{2}}} \psi(x) dx. \tag{4.12}$$

The French aerospace lab ONERA implemented the non-conservative Chimera method to their structured FV solver *elsA*. They first applied it to helicopter simulations with curvilinear grids around the rotor blades and the fuselage and a Cartesian background grid [52]. Then they used it for complex simulations of turbomachinery including technology effects like cooling holes, gaps separating fixed and rotating walls and many more [17, 18]. The Chimera method is particularly well suited to study the influence of these geometrical components. Adding one of them to the simulation does not necessitate a remeshing of the whole computational domain, but simply adds another component grid that can be modified independently. ONERA uses a non-conservative interpolation because of its flexibility and simplicity. However, mass flow is a very important parameter for the analysis of flows in turbomachines. Therefore, the introduced conservation losses have to be smaller than a prescribed upper bound. If this is not the case, the meshes are refined.

Hadžić [42] applies the Chimera method to the computation of the incompressible Navier-Stokes equations for flows around moving bodies. Again a non-conservative interpolation is used for the interpolated Chimera cells because the lack of conservation of momentum does neither introduce computational difficulties nor degrade the solution, according to [42]. However, mass conservation is indispensable for the correct solution of the pressure-correction equation that has to be solved for the used incompressible algorithm. For that reason, a correction method is introduced that enforces global mass conservation by correcting the mass fluxes at the interface between overlapping grids in a way that the sum of the mass fluxes is zero. The interface is defined as the faces between the Chimera cells that are interpolated and the fluid cells that are updated by the FV method. We define $\dot{m}_i$ as the mass flux between a Chimera cell and a fluid cell, where $i = 0, ..., n-1$ with $n \in \mathbb{N}_0$ denoting the number of faces between Chimera cells and fluid cells. The total mass flux over the interface

$$\Delta\dot{M} := \sum_{i=0}^{n-1} \dot{m}_i \tag{4.13}$$

should be zero and is used to correct the fluxes by

$$\dot{m}_i^{cor} = \dot{m}_i - \beta_i \Delta\dot{M}, \tag{4.14}$$

where β_i is defined as

$$\beta_i := \frac{|\dot{m}_i|}{\dot{M}} \quad \text{with} \quad \dot{M} := \sum_{i=0}^{n-1} |\dot{m}_i|.$$

That means that the mass conservation error $\Delta\dot{M}$ is redistributed among the Chimera-fluid faces, weighted by the magnitude of the uncorrected mass flux of each face. The sum of the

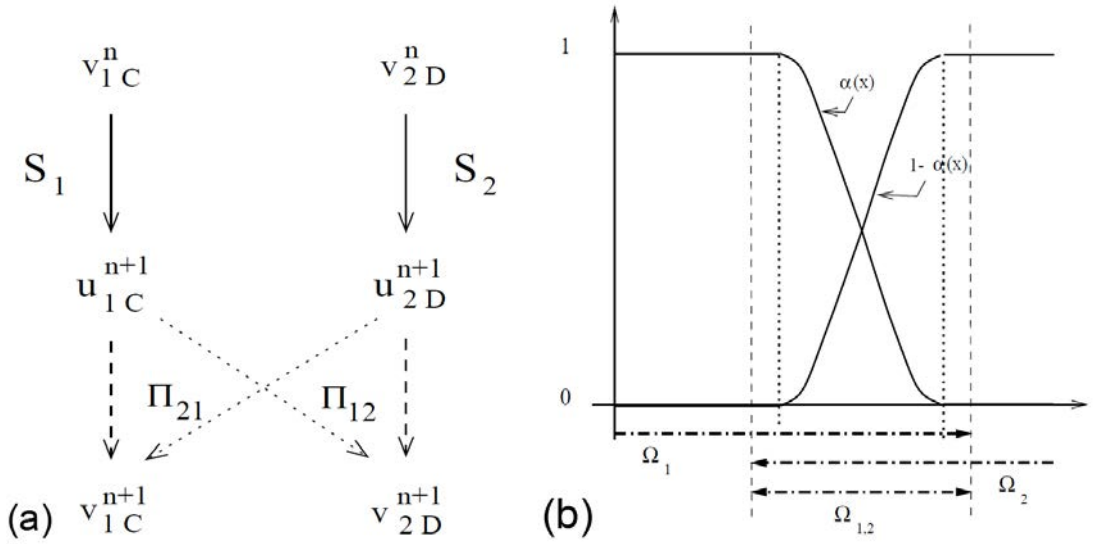


Figure 4.7: Mancip's coupling method [66] on overlapping domains Ω_1 and Ω_2 where the solution from Ω_1 at time t^{n+1} denoted by u_{1C}^{n+1} is projected on Ω_2 and vice versa in Ω_{12} and weighted by a smooth coupling function $\alpha(x)$. The method is conservative. (Both figures taken from [66].)

β_i is equal to one and hence, a zero total mass flux is enforced. Note that the idea behind the correction approach is similar to the closed box correction introduced in Chapter 3 even though the corrected error is of a different nature.

A different method to obtain conservation is proposed by Wang [114, 113]. Instead of imposing boundary conditions on the overlap, a common boundary is detected by the algorithm. In Figure 4.5 this would be the external green mesh line, i.e. the boundary of G_2. At these interfaces Riemann problems are solved similarly to the interior of a grid. The physical fields are linearly reconstructed and limited like for the MUSCL scheme, see Eq. (4.12), but here $\mathbf{\Phi}_L$ is reconstructed from a cell in G_1 and $\mathbf{\Phi}_R$ from an adjacent cell in G_2 or vice versa. This technique combines the strong points of overset method with the advantages of patched grids. On the one hand, grids can be generated independently from each other without special constraints on the boundaries which means that Chimera grids can still be used. On the other hand, the method is fully conservative like it is usually the case for patched grids.

In 2001, Mancip presented yet an other approach to obtain a conservative method for overlapping grids that he also applied to the coupling of SPH and FVM. To our knowledge this is the first publication about the coupling of SPH and FVM.

Mancip's coupling method

The starting point of Mancip's thesis [66] is the Chimera method and the work that has been done by Berger [11] and Pärt-Enander [87, 88] on conservative interpolation and the analysis of its stability (see the section above). On this basis Mancip develops a stable and conservative coupling algorithm on overlapping domains where the coupling no longer consists of imposing boundary values for each domain. Instead of a local interpolation, a projection is used on Ω_{12} (defined by Eq. 4.7) and the approximative solution is constructed by means of a smooth

coupling function defined over the whole computational domain $\alpha(\mathbf{x}) \in C^1\left(\mathbb{R}^d\right)$ with

$$\alpha(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1 \setminus \Omega_{12}, \\ 0 \leq \alpha(\mathbf{x}) \leq 1 & \text{if } \mathbf{x} \in \Omega_{12}, \\ 0 & \text{if } \mathbf{x} \in \Omega_2 \setminus \Omega_{12}. \end{cases} \tag{4.15}$$

A one dimensional example of $\alpha(\mathbf{x})$ is shown in Figure 4.7 (b). Figure 4.7 (a) shows the algorithm of the coupling method. Consider two triangulations (grids) G_1, G_2 with $C \in G_1$ in Ω_1 and $D \in G_2$ in Ω_2. Assuming that the approximate solutions on each domain at time t^n have been computed, the numerical schemes S_1 and S_2 are applied separately on both domains to obtain intermediate approximate solutions at time t^{n+1}, i.e. u_{1C}^{n+1} and u_{2D}^{n+1}. Then projections Π_{12}, Π_{21} and the coupling function $\alpha(\mathbf{x})$ are used to obtain a coupled solution for both domains, hence

$$v_{1C}^{n+1} = \alpha_C \; u_{1C}^{n+1} + (1 - \alpha_C) \; \Pi_{21}(u_2^{n+1})_C, \tag{4.16}$$

with

$$\Pi_{21}(u_2^{n+1})_C = \sum_{C : D \cap C \neq \emptyset} \frac{V_{D \cap C}}{V_C} \; u_{2D}^{n+1}, \tag{4.17}$$

and

$$v_{2D}^{n+1} = (1 - \alpha_D) \; u_{2D}^{n+1} + \alpha_D \Pi_{12}(u_1^{n+1})_D, \tag{4.18}$$

with

$$\Pi_{12}(u_1^{n+1})_D = \sum_{D : D \cap C \neq \emptyset} \frac{V_{D \cap C}}{V_D} \; u_{1C}^{n+1}. \tag{4.19}$$

The $\alpha_C, \alpha_D \in [0, 1]$ are the projection of $\alpha(\mathbf{x})$ on cells C and D, V_C, V_D are the volumes of cell C and D and $V_{D \cap C}$ denotes the volume of the intersection of cell C and D. That means that the intersection between the two triangulations has to be computed for the projection. If the meshes are static, the intersection is computed once at the beginning of the simulation. However, if the mesh is moving, the intersection has to be computed in every time step. In 1D the intersection can be easily computed but in 2D/3D it is not obvious if the mesh is not regular. Therefore an approximation based on a quadrature formula is employed, similar to what we propose for the computation of the modified weights (see Section 4.2.2). Note that Eq. (4.17) and (4.19) correspond to Eq. (4.6), that Berger *et al.* [12] call the conservative average of finer cells that replaces the physical fields of the coarse cells that are refined in AMR.

Mancip shows the convergence of the coupling algorithm in the scalar case and gives an error estimation. Then he shows 1D and 2D applications of the method for the coupling of two FV domains where the measured convergence rate is better than the one obtained by the theoretical analysis. The general construction of a multidimensional coupling function α is also explained. In the last chapter he adapts the method to couple SPH and FVM for an injection problem. SPH is a meshless method, in the sense that no mesh information is explicitly available since no connectivity information between the points is stored. Only the position $\mathbf{x}_j$ and the weight ω_j of particle j is known at any moment t^n. For Mancip's algorithm mesh information about the position of the corner points is needed and therefore he computes what he calls an *approximative and deformed* mesh where he moves the corner points in the same way as the particle positions. The resulting mesh cells have volumes that are very close to the ω_j. The computation of a mesh and the computation of the intersection in every time step increases the computational cost of the method. In addition, the Lagrangian velocity in the coupling zone has to be smooth enough to make the computation of the mesh possible without too strong deformations which means that the coupling zone has to be chosen accordingly.

4.1.3 Coupling of SPH with meshbased methods

Recently, several authors developed techniques to couple SPH with different meshbased methods. Amongst others, there are two main motivations to couple SPH with another method. The first is to discretize the fluid domain by SPH and couple it to a solid domain that is discretized by the Finite Element method (FEM). The coupling aims at simulating fluid-structure interaction. The other reason to couple SPH with a meshbased method like FVM is to obtain more information about the fluid by using more accurate operators, (an-)isotropic refinement or a better description of the physical boundary conditions.

4.1.3.1 Coupling of SPH with Finite Element methods

In structural mechanics the Finite Element method (FEM) is well established and used to compute stresses and deformations of structures but in many applications the influence of the fluid domain on the structure has to be considered. The so-called *fluid-structure interactions* (FSI) are the dynamic interactions of a structure with the fluid flow. Their computation needs an exchange of information from the fluid solver to the solid domain and vice versa in the course of the simulation. Some work has been done on the coupling of SPH for the fluid domain and FEM for the solid domain, where only two examples are cited here. In [32] the field variables at the interface between the solvers are directly exchanged, where the pressure is computed from the fluid for the structure by means of ghost particles. Another approach that conserves the energy at the interface exactly can be found in [59]. Due to the conservation of energy, different time integration schemes can be used without degenerating the order of the scheme in time or introducing numerical instabilities. Instead of ghost particles, surface elements and partial Riemann problems are used for wall boundaries [67].

4.1.3.2 Coupling of SPH with FV methods

To our knowledge the first published coupling method for SPH and FVM was presented by Mancip [66] in 2001. As it was explained above, it adapts the Chimera method to a conservative approach with a smooth coupling functions but necessitates the computation of an auxiliary grid for the SPH particles in the overlapping zone. This is a limitation of the method that makes it difficult to apply it to dynamic flows where the auxiliary grid would be very distorted or even impossible to construct because the particles moved away from their initial neighbours.

Then at the 8th SPHERIC Workshop in Trondheim 2013, where the coupling method proposed in this thesis was presented for the first time, two other contributions dealt with the same subject. They will be discussed in the following paragraphs.

In [15] a multi-domain method is presented that makes it possible to couple SPH to any other external solver or analytical solution. An interface with ghost particles in Lagrangian motion imposes boundary conditions on the SPH domain. Like in Section 4.1.2.1 we consider two subdomains Ω_1 and Ω_2 that overlap each other, see Eq. 4.7. The domain Ω_1 is discretized by SPH particles. The other domain Ω_2 can be discretized by SPH particles as well but with a different discretization size, or it can be discretized by a FV mesh or the physical fields can be given by an analytical solution. The physical variables of the ghost particles in the interface region are interpolated from the interior of Ω_2, using an interpolation method that is adapted to the way Ω_2 is discretized. The multi-purpose interface is successfully applied to a coupling with an implicit FV solver.

A coupling of SPH-ALE and Voronoi-FVM is developed in [5] that is called *Hybrid-SPH*. The idea is to combine the more precise Voronoi-FVM operators with the surface treatment of SPH. For that purpose, a Voronoi tesselation is constructed on the whole fluid domain, the free

surface is detected and the SPH domain is defined close to the free surface. Away from the free surface, Voronoi FVM is employed. Similar to what is done for the coupling algorithm presented in this thesis, the same weakly-compressible Euler equations are discretized by FV operators and Riemann problems are solved between Voronoi cells using the same MUSCL reconstruction to obtain $\mathbf{\Phi}_L$ and $\mathbf{\Phi}_R$ at the cell faces as between particles. Thanks to the Voronoi tesselation the interface between the SPH domain and the FV domain is known. However, no special interface treatment is used but SPH particles (with its known tesselation) are used as neighbours for the Voronoi FVM integration and the Voronoi cells included in the kernel support area are themselves neighbours for the SPH particles. Due to the different operators on the SPH domain and the FV domain, the method is not conservative.

4.1.4 Summary and discussion

In this section different approaches from literature concerning spatial refinement in SPH and FVM were discussed. Special emphasis was placed on multi-domain methods like the Chimera method for overlapping grids because we identify a multi-domain strategy as a possible candidate for the coupling of SPH and FVM that will be presented in the next section. These methods are known to be flexible and there are good experiences with industrial applications. However, in general they are not conservative and mass conservation has to be checked after each simulation. Patched grid methods or improved Chimera methods like the one proposed by [114] do not have this problem. But its application to the coupling of SPH and FVM might be tricky because the movement of the particles in Lagrangian motion has to be managed explicitly at interfaces between the FV domain and the SPH domain where particles cannot penetrate.

Recently, three different coupling methods of SPH and FVM were published. The first one by Mancip [66] is conservative and uses a smooth coupling function defined on the whole overlapping region but needs an auxiliary mesh that is computed by evolving the corner points of the initial grid in the same way as the particle centers. This is not adapted to our target applications where we want to use SPH in particular for very dynamic flows that cannot be easily computed with meshbased methods.

The second method published by Bouscasse *et al.* [15] is a multi-domain method that uses Lagrangian ghost particles to impose boundary conditions to the SPH domain. The domains can be overlapping but are not necessarily overlapping. The region of ghost particles has to be larger than the radius of the smoothing length. The authors use ghost particles for every boundary of the SPH domain, the solid boundaries, the inflow and outflow boundaries and it is consistent to use them also for multi-domain boundaries. The SPH-ALE method used in this thesis employs partial Riemann solvers and surface elements for solid wall boundaries and no ghost particles are constructed for wall boundary treatment. Nevertheless, this method is a possible candidate for the coupling of SPH and FVM for our target applications but it is not the one that we employ.

The third method proposed by [5] couples SPH-ALE to Voronoi-FVM. The domains do not overlap and there is a known interface between the SPH particles and the Voronoi cells. The Voronoi tesselation has to be computed for the cells and for the SPH particles and information is transferred through the interface by using the particles as neighbours for the FVM and the Voronoi cells as neighbours for the SPH integration. In the method that we present in the next section, FV cells are used as neighbours for the SPH integration similar to what is done by [5]. However, no tesselation for the SPH particles will be computed and the FV mesh is an anisotropic block structured mesh and not a Voronoi tesselation. In [5] it is explained that the Voronoi cells should be kept as isotropic as possible in order to conserve accuracy. This is not what we are searching for our target applications.

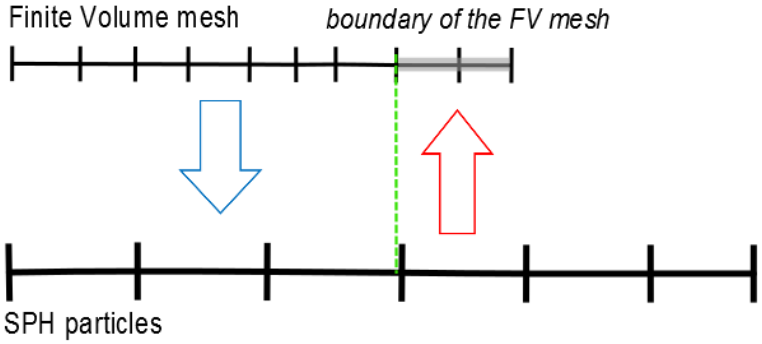


Figure 4.8: Sketch of the coupling strategy for overlapping domains, where communication works in two ways. On the one hand, FV calculation points are used as SPH neighbours (indicated by a blue arrow), while on the other hand, FV boundary values are interpolated from SPH particles (indicated by a red arrow). The FV cells that obtain interpolated fields, that we call *Chimera cells*, are shown by a grey box. The green line denotes the limit of the FV mesh (excluding the Chimera cells).

In the following, a flexible coupling method is presented that does not require the construction of a mesh for the SPH particles or the construction of ghost particles. It is based on overlapping domains where not only boundary conditions are applied, but the refined and more accurate FV cells of the overlapping meshes are used as neighbours for the SPH integration. The method is applicable to all kinds of FV meshes even though anisotropic block structured meshes are best adapted for our target applications.

4.2 The proposed coupling methodology

The coupling strategy that we propose in this thesis is based on a multi-domain technique of overlapping domains of SPH particles $\Omega_{SPH} \subset \mathbb{R}^d$ and FV meshes $\Omega_{FV} \subset \mathbb{R}^d$, $d = 1, 2, 3$, with

$$\Omega_I = \Omega_{SPH} \cap \Omega_{FV} \neq \emptyset.$$

Domains are completely or partially overlapped by others in order to enable communication between them. A different technique is applied for the transfer of information from the SPH to the FV domain than for the transfer from the FV to the SPH domain. On the one hand, boundary values are interpolated from the SPH particles to the FV mesh and on the other hand, FV cells are used as SPH neighbours for the SPH space integration in Ω_I. In Figure 4.8 the idea of the coupling is illustrated in one space dimension.

In the following, we assume that the field variables computed on the FV domain are more accurate than the ones computed by the SPH method because we use FV meshes that have smaller discretization sizes on the overlapping domain than the SPH particles. Hence, the condition

$$\max(\Delta\xi_l, \Delta\psi_l, \Delta\zeta_l) \leq r \quad \forall l \in [0, N-1] \tag{4.20}$$

has to be respected in the overlapping region Ω_I when constructing the mesh. Here the number of cells in Ω_I is denoted by $N \in \mathbb{N}_0$ and the notation for the mesh lines is described in Figure 4.9. In this chapter the particle size of the SPH particles is denoted by r instead of Δx to avoid confusion with the discretization size of the FV mesh. Note that for most applications this

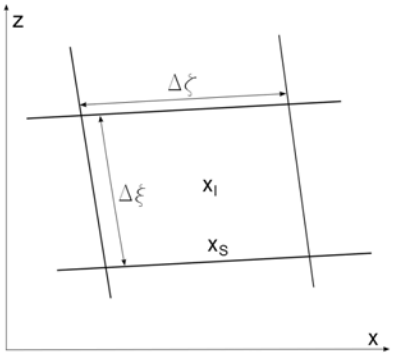


Figure 4.9: Notation for a twodimensional cell l with center $\mathbf{x}_l$. Each side has a center $\mathbf{x}_S$, a surface area S and a normal $\mathbf{n}$ pointing from cell l to its neighbour cell. The normal multiplied by the surface area is denoted by $\mathbf{n}_S = \mathbf{n}\,S$ and the cell volume by V_l.

condition is not a real restriction because the mesh cells can be bigger than SPH particles on $\Omega_{FV} \setminus \Omega_I$ and we use the FV mesh for local refinement most of the time. However, in the case of boundary layers long and thin cells are traditionally used for FVM. In this case, care has to be taken to fulfil the above condition in all directions of the mesh lines.

4.2.1 Interpolation of boundary cell values to the FV domain

Since the FV domain is the more refined and therefore the more accurate one, only boundary values are interpolated from the SPH particles to the FV mesh. In analogy to the Chimera method density and velocity are interpolated for two layers of boundary cells. For the interpolation scattered data approximation techniques are well adapted as the SPH particles are an unstructured cloud of points. Moving least square (MLS) approximation of any order can be chosen. In the testcases presented below, Shepard interpolation [99] is used. That means that we interpolate the fields ϕ_l for the FV boundary cells, i.e. density and velocity, as follows,

$$\phi_l = \sum_{j \in D_l^{SPH}} \phi_j \frac{\omega_j W_{lj}}{\sum_k \omega_k W_{lk}}.$$

The size of the kernel support for the interpolation is determined by the SPH discretization size in order to have enough neighbours for the interpolation. Analogously to what was said for Chimera methods in Section 4.1.2.1, the interpolation is not conservative. That means that the mass unbalance has to be evaluated at the end of simulations.

4.2.2 FV calculation points as neighbours for the SPH space integration

The main idea of the coupling approach is to use FV cells as neighbours for the SPH space integration if FV cells as well as SPH particles discretize the same space. If there are no FV cells overlapping the kernel support area of a certain particle, the other SPH particles in the neighbourhood are used, like in non-coupled standard SPH. However, a special treatment has to be applied for overlapped particles close to solid walls that will be described below. Figure 4.10 shows a 1D mesh overlapping SPH particles, where the considered SPH particle at x_i uses FV cells as well as other SPH particles as neighbours. In the kernel support there are particles that

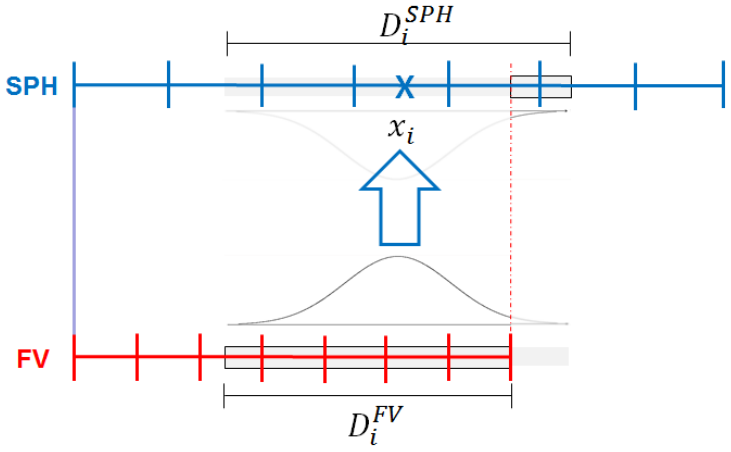


Figure 4.10: A 1D mesh is overlapping SPH particles and the SPH space integration of the particle at x_i is considered. Only parts of the kernel support area are covered by FV cells and there are FV calculations points as well as other SPH particles acting as SPH neighbours.

are themselves only partially overlapped by the FV mesh and their weight has to be adjusted in a way that only the part of the particle that is not covered by the FV domain is taken into account for the SPH integration. We call the new weight of the cut particle *modified weight*.

Far away from solid walls, we obtain the coupled SPH-ALE flow equations that are written as

$$\frac{d(\omega_i \mathbf{\Phi}_i)}{dt} = \omega_i \left(\sum_{k \in D_i^{FV}} V_k \hat{\mathbf{F}}_{ik} \nabla_i W_{ik} + \sum_{j \in D_i^{SPH}} \bar{\omega}_j \overline{\mathbf{F}}_{ij} \overline{\nabla_i W}_{ij} \right), \tag{4.21}$$

where $\bar{\omega}_j$ denotes the modified weights, $\mathbf{\Phi}_i$ the vector of state variables of particle i and $\hat{\mathbf{F}}_{ij}$ the numerical flux between calculation points i and j. Note that $\bar{\omega}_j = 0$ if particle j is fully overlapped by FV cells. Not only the weight of a partially covered particle has to be modified, also its position. A particle with a new weight should again be situated close to its barycenter and the position has to be modified accordingly. If the particle is not overlapped at all, the position should not change.

Since the positions are modified, the kernel function and its gradient $\nabla_i W_{ij}$ change as well, and is denoted by $\overline{\nabla_i W}_{ij}$. The modified position equally alters the direction of the vector connecting the particle with its' neighbours and therefore also the numerical flux $\hat{\mathbf{F}}_{ij}$ that is obtained by solving Riemann problems. We denote the modified numerical flux by $\overline{\mathbf{F}}_{ij}$, to emphasize that it is not the same as for non-coupled simulations.

Figure 4.11 shows a 2D mesh (coloured in red) that is overlapping particles. The particles that have to adjust their weight and position are coloured in green. Figure 4.12 summarizes the three different types of neighbourhoods that an SPH particle can have for the coupling. There can be other SPH particles as neighbours (Zone 1), other unchanged SPH particles, modified SPH particles and FV cells (Zone 2) or exclusively FV cells (Zone 3).

Computation of modified weights and modified positions

In Zone 2 in Figure 4.12 and 4.16 every SPH particle has other SPH particles as well as FV cells as neighbours. If the neighbour particle is situated at the boundary of the FV domain,

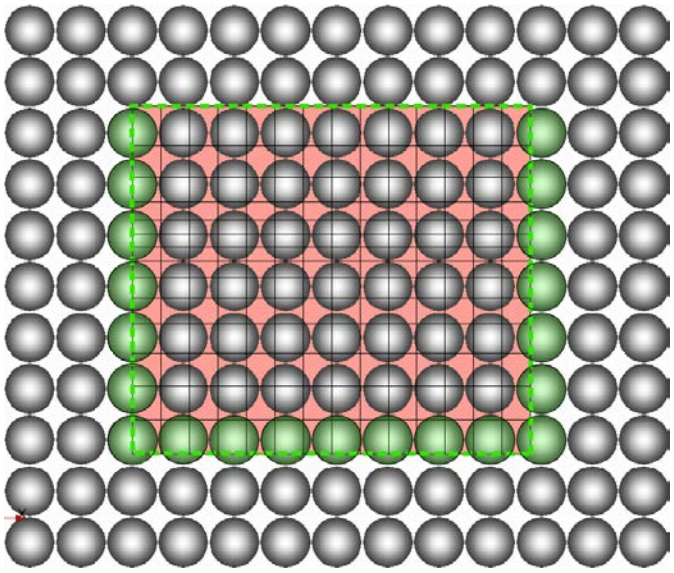


Figure 4.11: A FV mesh (colored in red) is overlapping SPH particles where a green dashed line indicates the boundary of the FV domain. The weight of the particles that are colored in green has to be adjusted for the coupling to obtain a correct space discretization, while the particles that are completely overlapped by the mesh are not taken into account as neighbours. Note that the Chimera cells are not indicated in this figure.

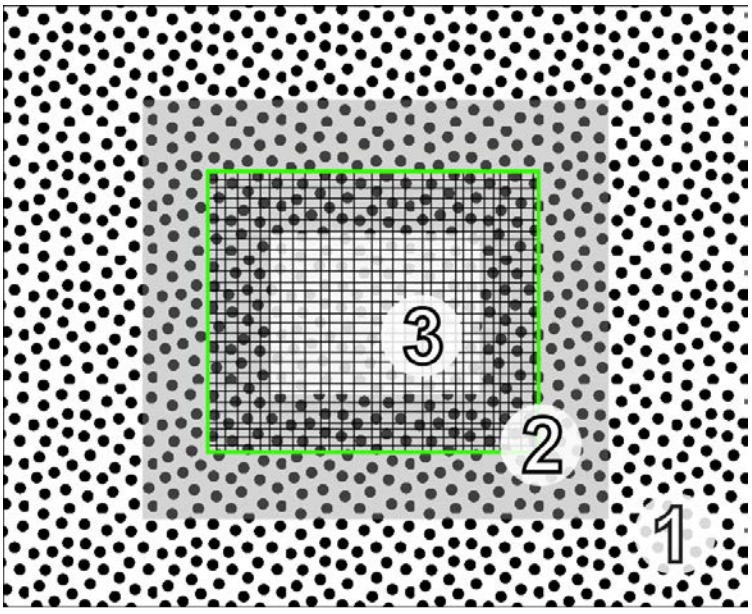


Figure 4.12: Distinction of three different zones that occur for the coupling far away from solid walls. In Zone 1 the particles exclusively have other SPH particles as neighbours, in Zone 2 (colored in grey) there are particles as well as cells in the neighbourhood of each particle, while in Zone 3 there are only FV neighbours. The green line indicates the boundary of the FV mesh (without indicating the Chimera cells). The width of Zone 2 corresponds to the diameter of the kernel support.

the weight of the neighbour particle has to be adjusted in accordance with the volume that is overlapped by the FV mesh. We define the *modified weight*, $\bar{\omega}_j$, of a particle j as the weight that is taken into account for the space integration in the coupling. The modified weight is equal to the original weight ω_j if the particle is not covered at all by a FV cell, i.e. $\bar{\omega}_j = \omega_j$. It is zero if the particle is completely overlapped, i.e. $\bar{\omega}_j = 0$, and it is between zero and the original weight, $\bar{\omega}_j \in (0, \omega_j)$, if the particle is partially overlapped like the ones coloured in green in the example of Figure 4.11. Hence, we define the modified weight by

$$\bar{\omega}_j := \omega_j - \tilde{V}_j, \tag{4.22}$$

where $\tilde{V}_j$ is the exact volume of the FV mesh that is overlapping the SPH particle j. In the general case, it is not possible to compute $\tilde{V}_j$ exactly but an approximate method has to be employed. One possible method to compute approximately the modified weight $\bar{\omega}_j$ of particle j is given by

$$\bar{\omega}_j \approx \begin{cases} \omega_j - \sum_l V_l & \text{if } (\omega_j - \sum_l V_l) > 0, \text{ with } l : \|\mathbf{x}_l - \mathbf{x}_j\| < \frac{1}{2}\,\omega_j^{\frac{1}{d}} \\ 0 & \text{elsewhere} \end{cases}, \tag{4.23}$$

where d denotes the space dimension. The size of the particle is approximated by $\omega_j^{\frac{1}{d}}$ in this expression and $\tilde{V}_j$ is approximated by the sum of the cell volumes whose cell center is situated inside the isotropic particle. Note that in SPH only the position of a particle and its volume are known but its exact extent in space is not known. Supposing that the particle is a sphere in 3D or a circle in 2D, like we do in Eq. 4.23, is an approximation of the real form of the particle that is not computed in SPH.

If the SPH particles are distributed homogeneously, their position is situated at the barycentre of the neighbouring particles, i.e. $\mathbf{x}_i = \sum_{j \in D_i} \omega_j \mathbf{x}_j W_{ij}$. In [3] it is shown that truncation errors increase significantly if the particle is not situated in its barycentre. Hence, it is necessary not only to modify the weights but also the particle positions. From the definition of the barycentre it is known that the barycentre of the cut particle can be written as

$$\bar{\mathbf{x}}_j \bar{\omega}_j = \mathbf{x}_j \omega_j - \tilde{\mathbf{X}}_j \tilde{V}_j, \tag{4.24}$$

where $\tilde{\mathbf{X}}_j$ is the barycentre of the part of the FV mesh that overlaps the particle. We assume that the particles are initially situated at their barycentre. Then we can compute the modified position as

$$\bar{\mathbf{x}}_i \approx \begin{cases} \frac{1}{\bar{\omega}_j}\,(\mathbf{x}_j \omega_j - \sum_l \mathbf{x}_l V_l) & \text{if } \bar{\omega}_j > 0, \\ \mathbf{x}_j & \text{if } \bar{\omega}_j = 0, \end{cases} \tag{4.25}$$

where the sums are again calculated over

$$l \in [0, N-1] : \|\mathbf{x}_l - \mathbf{x}_j\| < \frac{1}{2}\,\omega_j^{\frac{1}{d}}, \tag{4.26}$$

and d denotes the space dimension.

Figures 4.13 and 4.14 show a one dimensional sketch of a SPH particle that is overlapped by FV cells. In the case of Figure 4.13 we obtain the exact $\bar{\omega}_{j_1}$ of the "cut" particle j_1 by computing $\bar{\omega}_{j_1} = \omega_{j_1} - (l_1 + l_2 + l_3)$. Also the modified weight of the completely overlapped particle j_2 is computed correctly, since we obtain $\bar{\omega}_{j_2} = 0$. This is exact in this case because the left particle interface and the cell interface of cell l_1 are aligned. However, in general this is not true. Considering the example of Figure 4.14 we again obtain $\bar{\omega}_{j_2} = 0$ for particle j_2 but we

do not obtain the correct modified weight for particle j_1 because the volume indicated by a red rectangle is not seen by the formula and therefore the computed $\bar{\omega}_{j_1}$ is too big. Even though one part of the FV cell l_1 covers the considered SPH particle, the volume of the cell will not be taken into account in the sum $\sum_l V_l$ because the cell center does not verify the condition (4.26), i.e. $l : \|\mathbf{x}_l - \mathbf{x}_j\| < \frac{1}{2}\,\omega_j^{\frac{1}{d}}$. The modified weight that is computed by Eq. (4.23) also contains the volume indicated by the red square which is not correct.

Figure 4.15 shows that in general it is not even possible to compute correctly the modified weights of particles that are completely overlapped by FV cells and that should be zero to exclude them from space integration.

Analogously to what was said for Eq. (4.23), Eq. (4.25) does not yield the exact barycentre for the modified particle in general, even if the particles are initially situated at their barycentre.

The errors that are introduced by the computation of the modified weights by Eq. (4.23) and the positions by Eq. (4.25) considerably deteriorate the quality of the results, especially if a particle that is completely overlapped by FV cells does not have a zero modified weight which means that a contribution of a SPH particle is added to the integration that is wrong.

To improve the accuracy of Eqs.(4.23) and (4.25) quadrature points are added, similar to Newton-Cotes formulae for numerical integration. In the following the approach is outlined for quadrilateral/hexahedral meshes that are used in this work but it can be easily adapted to triangle/tetrahedral meshes. For simplicity the two-dimensional case is explained but the extension to 3D is straight forward.

A reference length L is defined by

$$L := \alpha\ \omega_j^{\frac{1}{d}}, \text{with } \alpha \in (0, 1]. \tag{4.27}$$

For each direction of the mesh lines ζ, ξ we compute

$$n_\zeta(l) = \left\lceil \frac{\Delta\zeta(l)}{L} \right\rceil \text{ and } n_\xi(l) = \left\lceil \frac{\Delta\xi(l)}{L} \right\rceil$$

using a ceiling function $\lceil\cdot\rceil$ and constructing $n_\zeta(l) \times n_\xi(l)$ refined quadrilateral cells for each cell l with edges smaller or equal to L. Note that L does not depend on the FV mesh but only on the discretization size of the SPH domain. The number of auxiliary subcells for each cell, $n_\zeta(l) \times n_\xi(l)$, depends on the discretization size in the direction of each mesh line, $\Delta\zeta(l)$, $\Delta\xi(l)$, that varies for each cell l.

The auxiliary cells are then used for the sums in (4.23) and (4.25) that become

$$\tilde{V}_j \approx \sum_l \sum_{k(l)} V_k \tag{4.28}$$

and

$$\tilde{\mathbf{X}}_j \tilde{V}_j \approx \sum_l \sum_{k(l)} \mathbf{x}_k V_k, \tag{4.29}$$

where

$$\begin{gathered} l \in [0, N-1],\ 0 \le k < (n_\zeta(l) \times n_\xi(l)) : \\ \|\mathbf{x}_k - \mathbf{x}_j\| < \tfrac{1}{2}\,\omega_j^{\frac{1}{d}}. \end{gathered} \tag{4.30}$$

In Section 4.3.2 the influence of the choice of the refinement factor α on the accuracy of the computed modified weights and on the resulting pressure field is shown by means of the two-dimensional Taylor Green vortices.

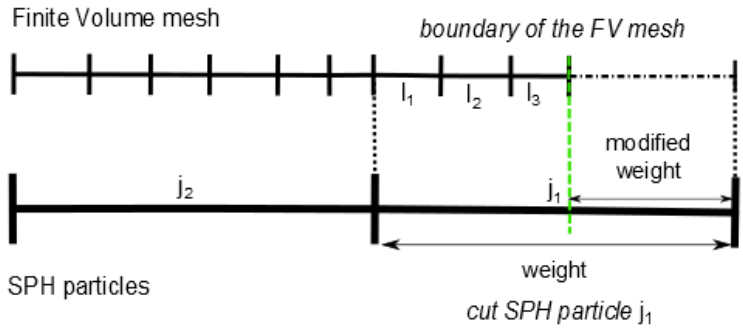


Figure 4.13: A FV mesh is overlapping SPH particles, where the black lines indicate the interfaces between the cells and between the particles, respectively. The green line shows the boundary of the FV domain and the SPH particles are covered by FV cells in a way that the particle j_1 at the right side is "cut" by the boundary of the FV mesh. The modified weight of this partially overlapped particle is computed by Eq. (4.23). This is exact because the sum of the volumes of cell l_1, l_2 and l_3 represents exactly the volume that is overlapping particle j_1.

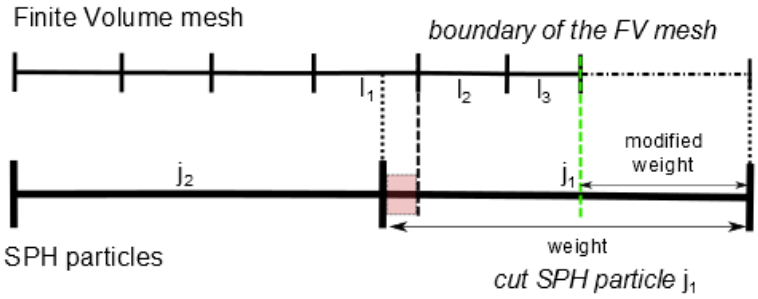


Figure 4.14: SPH particles are covered by FV cells in a way that the particle j_1 at the right side is "cut" by the boundary of the FV mesh. The modified weight of this partially overlapped particle is computed by Eq. (4.23). In contrast to what is shown in Figure 4.13, the volume of the considered SPH particle (indicated by a dotted line) extends beyond the volume that is covered by the sum of cells l_2 and l_3 and is partially covered by cell l_1. Since the center of cell l_1 lies outside the limits of the particles, its volume is not taken into account. The volume indicated by the red square is wrongly added to the modified weight.

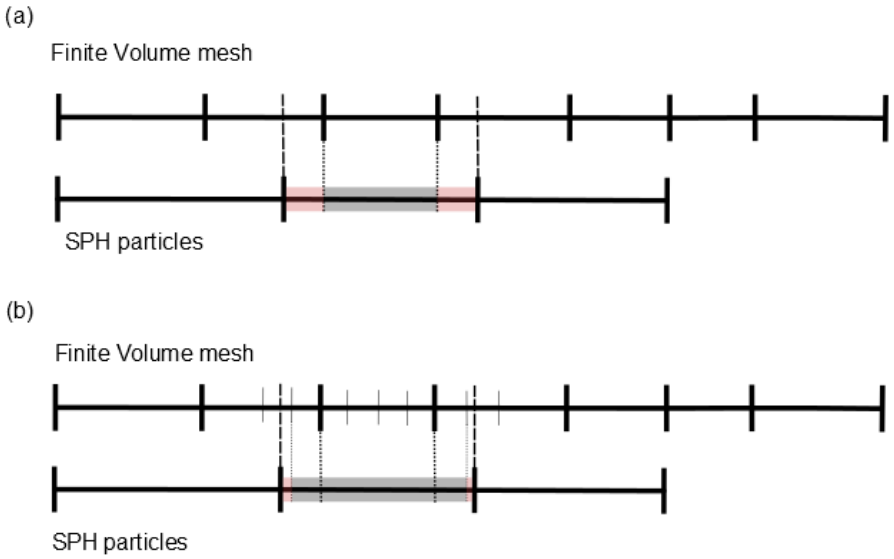


Figure 4.15: Finite volume cells (above) that are overlapping SPH particles (below). In (a) the error that is made by computing the modified weight of the considered particle by Eq. (4.23) is colored in red. In (b) it is shown how the error can be reduced by applying Eq. 4.23 with refined FV cells.

Particles close to solid walls

In SPH-ALE solid wall boundaries are taken into consideration by integrating over surface elements and solving partial Riemann problems at these surface elements, see Section 3.7.2. For the coupling we detect the particles whose kernel support area is truncated by the surface elements and then interpolate density and velocity directly from the underlying FV cells using the same quadrature points as for the computation of the modified weights. We compute

$$\phi_j = \frac{1}{\tilde{V}_j} \sum_l \sum_{k(l)} \phi_k V_k, \tag{4.31}$$

for ϕ_j denoting density and velocity of particle j and $\phi_k = \phi_l$ with l and k as in (4.30) and $\tilde{V}_j$ defined by (4.28). Note that Eq. (4.31) is very similar to the projections introduced by Mancip, Eq. (4.19), (4.17), and the conservative average from AMR, Eq. (4.6). An important difference between literature and Eq. (4.31) is that $\tilde{V}_j$ is only an approximation of the volume of the FV domain that is covering particle j. That means that even for completely overlapping domains $\Omega_{SPH} \cap \Omega_{FV} = \Omega_{FV} = \Omega_{SPH}$,

$$\sum_{j \in \Omega_{SPH}} \tilde{V}_j \neq \sum_{j \in \Omega_{SPH}} \omega_j \neq \sum_{l \in \Omega_{FV}} V_l,$$

for several reasons. We denote the total volume that is coverd by $\Omega_{FV} = \Omega_{SPH}$ by V_Ω. FV cells always cover the exact volume that they discretize, independent of the discretization size, i.e. $\sum_{l \in \Omega_{FV}} V_l = V_\Omega$. However, in SPH there is no conservation of volume and it is not guaranteed

that the sum of all SPH weights is equal to V_Ω. Hence, even for completely overlapping domains it is not certain that the total volume covered by the SPH particles is equal to the total volume covered by the FV cells. Moreover, the $\tilde{V}_j$ are computed approximately to approximate the SPH weights and additional errors are introduced. These errors decrease by decreasing α and hence, by decreasing the reference length L of the auxiliary cells.

A constraint on the meshes that are used for the coupling around solid geometries follows from the interpolation of the physical field variables of the covered particles close to walls, using Eq. (4.31). It means that if there is a FV mesh around a solid geometry, it has to be at least as large as the radius of the kernel support even if the interpolation itself only requires cells covering the volume of the considered particle. However, the field variables of every particle, whose kernel support is truncated by the wall and that has cells covering parts of its kernel support, are interpolated by Eq. (4.31).

Choice of the smoothing length

In the coupled solver, the smoothing length h that determines the size of the kernel support area for the SPH space integration is given by the SPH particles. That means that the smoothing length is not adapted to the discretization size of the FV mesh. As a consequence, the number of neighbours varies strongly when coupled to a fine FV mesh. In Section 4.1 we saw that variable h introduces ∇h terms in the computation of ∇W that are not easy to add. In particular, the FV meshes are anisotropic, often strongly refined in one direction, e.g. normal to solid walls. Anisotropic kernel functions (see Section 4.1.1.2) would have to be used if a constant number of neighbours had to be kept in all directions of the mesh lines. This is very complicated to implement for arbitrary geometries like they arise in industrial applications. For these reasons, the smoothing length is not adapted to the FV mesh in the coupled solver and a higher computational cost arising from a higher number of neighbours is accepted.

4.2.3 Practical implementation

A description of the sequential SPH-ALE algorithm used in ASPHODEL is given in [70]. In this thesis, only the three main parts, that are run for each time step, are mentioned. That are the *preliminary loops* (pre-loops) where the gradients for the MUSCL reconstruction are computed, the *flux balance loop* where the numerical fluxes are computed and accumulated, and the *time integration loop* where the variables are advanced in time. The FV solver was directly implemented into ASPHODEL following a similar structure containing the same three main parts. In the same way, the coupling algorithm is implemented directly into ASPHODEL, see Algorithm 1.

Algorithm 1 Coupled SPH-ALE and FV solver

1: **while** $t < t_{end}$ **do**
2: Interpolate the physical fields for Chimera cells
3: Compute modified weights and positions for SPH particles
4: Interpolate the fields from the FV cells to the covered SPH particles close to walls
5: Execute the pre-loops SPH and SPH-FV
6: Execute the pre-loops FV: compute gradients
7: Execute the flux balance loop SPH and SPH-FV
8: Execute the flux balance loop FV
9: Perform the time integration for SPH and FV domains
10: **end while**

The loops over the FV cells as SPH neighbours are denoted by pre-loops SPH-FV and flux balance loop SPH-FV. The loops pre-loops SPH and flux balance loop SPH use the modified weights and positions. That means that also the gradients for the SPH particles are computed by the coupled method, i.e. SPH neighbours and FV neighbours are used.

4.2.4 Properties of the coupling

The coupling algorithm is very flexible and allows us to add FV meshes for anisotropic local refinement anywhere in the SPH domain. The SPH particles can be in Eulerian or Lagrangian motion or something in-between (ALE) and their motion is not disturbed by the meshes. One advantage of the present method is that the creation and destruction of particles is handled by the SPH solver and no additional interfaces where particles have to be added or deleted are needed. Moreover, FV meshes can be easily used to impose inlet or outlet conditions to the SPH domain. This will be further discussed in Chapter 5.

However, a drawback of overlapping domain techniques is the loss of conservation (see Section 4.1.2.1). Both, the FV scheme and the SPH-ALE scheme, are written in conservative form but the coupling algorithm does not guarantee conservation of mass and momentum. That means that FV solver and the SPH solver are conservative but the coupling is not. If particles are used as SPH neighbours interactions are still symmetric but in the overlapping region FV cells are taken as neighbours and the particles are passive without contributing to the integration of the other particles. Hence, conservation is not verified automatically any more. Moreover, for the interpolation of the FV boundary values no conservative interpolation technique has been used. Analogously to the Chimera technique, conservation has to be checked carefully for each simulation. However, the validation cases in the section below show that conservation losses are very small and no big issue for practical applications. It is known from literature that non-conservative methods can introduce additional problems like non-physical shock speeds for shock simulations. We did not observe that for the validation cases presented below. Furthermore, shock simulations are not part of the target applications that we are interested in but it is still something that has to be kept in mind when analysing the results.

The same time step size is used for the FV domain as well as the SPH domain. However, cells can be much smaller than particles, especially close to solid walls. The small cells reduce the timestep considerably due to the CFL condition that has to be verified. Coupling methods with different time integration exist in meshbased methods. Implementing a coupling method for SPH with different timestep sizes or even different time integration schemes is left to future research.

In Section 4.1 refinement and de-refinement strategies were discussed including Barcarolo's de-refinement method with passive mother particles. Eq. (4.4) and Eq. (4.5) define the SPH operators for a function and its gradient for a computational domain including mother and daughter particles. It was already stated above that the passive mother particles are treated similarly to SPH particles in the present coupling method that are completely covered by FV cells. If we define $\gamma_i := \frac{\bar{\omega_i}}{\omega_i}$, we obtain similar SPH operators as [5] with $\gamma \in [0, 1]$, depending on the position of the particle to the FV domain. We can write the coupled SPH operators as

$$f_i = \sum_{j \in D_i^{SPH}} f_j \overline{W}_{ij} \omega_j \gamma_j + \sum_{j \in D_i^{FV}} f_j W_{ij} V_j,$$

for a function f and

$$\nabla f_i = \sum_{j \in D_i^{SPH}} f_j \overline{\nabla_i W}_{ij} \omega_j \gamma_j + \sum_{j \in D_i^{FV}} f_j \nabla_i W_{ij} V_j,$$

for its gradient $\nabla_i f$, using $\overline{W}_{ij}$ and $\overline{\nabla_i W}_{ij}$ to remind the reader that also the positions are modified for the cut particles.

4.2.5 Summary

A coupling algorithm for SPH-ALE and FVM was developed that is based on communication on overlapping domains where information is transferred in two ways.

For the coupled SPH integration, we distinguish between the following four zones, as it is shown in Figure 4.16, depending on the neighbours of the considered particle.

- In *Zone 1* there are only SPH particles and no FV mesh. Every SPH particle has exclusively other particles as neighbours in its kernel support.
- *Zone 2* is an intermediate region where the particles have other particles as neighbours as well as FV cells. Particles that are completely overlapped by FV cells are excluded from the integration to avoid integrating twice over a control volume, once by integrating over the particles and once over the cells. Some particles are only partially overlapped by the FV mesh, they are at the boundary of the FV domain and their weights and positions have to be adjusted to guarantee a correct space discretization. In Figure 4.11 the particles whose weight has to be adjusted are coloured in green.
- In *Zone 3* the particles are situated in the interior of the FV mesh and the whole kernel support area lies within the FV domain. All SPH particles in the neighbourhood are completely overlapped by FV cells and do not contribute to the SPH space integration. They are passive and only FV cells are used as neighbours. The modified weights are set to zero.
- In *Zone 4* the particle is situated close to a solid wall and its kernel support area is intercepted by the wall. In this region we do not perform a SPH space integration but interpolate velocity and density directly from the FV cells, see Eq. (4.31). The SPH particles in this zone do not contribute to the space integration of its' neighbours.

The coupled FV solver receives two layers of interpolated boundary cell values, the so-called Chimera cells, from the SPH particles in every timestep. The rest of the FV solver is carried out as usual.

4.3 Validation of the coupling strategy

The coupling method is validated by academic testcases in one and two space dimensions where either an analytical solution is known or a reference solution from another well-established CFD solver is available. All validation cases are inviscid flow simulations. In 1D we present a shock tube case and a Lagrangian case with a sinusoidal initial velocity. In 2D the bi-periodic Taylor-Green vortices and the flow around a symmetric NACA hydrofoil that were introduced in Chapter 2 are reused.

The closed box correction that was presented in Section 3.9.3 is applied to all simulations for the following reason. For the coupling algorithm FV cells are taken as SPH neighbours. Usually the SPH neighbourhood consists of neighbouring particles and the considered particle itself. Since the gradient of the kernel function at the position of the considered particle is zero, no self-contributions are added. But because of the particle in the center, the other particles are distributed around it. This is not the case if FV cells are the neighbours. They are not situated

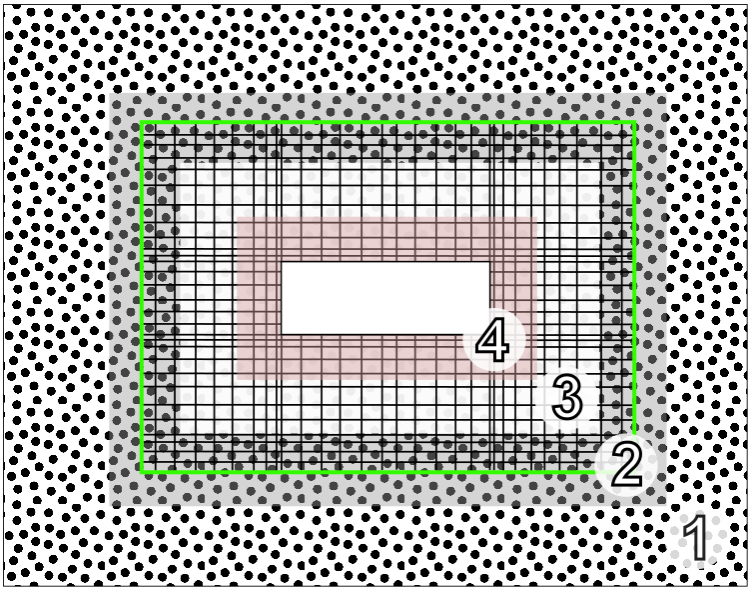


Figure 4.16: Distinction of four different zones that occur for the coupling. In Zone 1 the particles exclusively have other SPH particles as neighbours, in Zone 2 (colored in grey) there are particles as well as cells in the neighbourhood of each particle, while in Zone 3 there are only FV neighbours. In Zone 4 close to solid walls (colored in red) the physical fields are directly interpolated from the underlying FV mesh. The green line indicates the boundary of the FV mesh.

around the considered particle and in general there is no cell situated in the middle with a center at the position of considered particle. Hence, there is no symmetry of the cells regarding the position of a considered particle. In addition, the cells are not isotropic and they are not able to redistribute themselves because they are part of a mesh. It is therefore important to apply the closed box correction to the SPH integration in order to obtain good results for the coupling.

4.3.1 One-dimensional testcases

The following one-dimensional testcases are inviscid, i.e. $\nu = 0$ and all SPH particles are in Lagrangian motion, i.e. $u_0(x,t) = u(x,t)$. For the SPH and FV domain an explicit forth order Runge-Kutta time integration scheme is employed with the same time step size. The Riemann problems in the FV solver and the SPH solver were solved by approximative Riemann solvers and MUSCL reconstruction with minmod-limiters, see e.g. [68].

4.3.1.1 Shocktube testcase

We consider the one dimensional shock tube testcase presented in [55] and [67]. The initial data was chosen according to [55] with a density discontinuity in the middle of the domain at $x = 0.5$,

$$\begin{cases} \rho_L = 1100, & u_L = 0, \\ \rho_R = 1000, & u_R = 0, \end{cases}$$

with the reference density $\rho_0 = 1000$ kg/m^3 and the reference speed of sound $c_0 = 1450$ m/s. The analytical solution consists of a rarefaction wave travelling from the discontinuity to the

left and a shock wave travelling to the right. This testcase is chosen to illustrate the following two aspects. First, it shows that the coupling does not introduce any significant errors if the initial discontinuity is situated exactly at the interface. In this case the discretization size of both domains is the same. Then, we show how the FV domain can be used to refine a certain region of interest, where of course different discretization sizes are used.

Coupling with equal discretizaton sizes

We use 5000 SPH particles distributed in the whole computational domain and 500 FV cells distributed in $x_l \in [0.5, 0.6]$. That means that the discretization size for the SPH domain is the same as for the FV domain, i.e. $r = \Delta x = 0.0002$m. Figure 4.17 shows the pressure and velocity at time $t = 10^{-5}$s. It can be seen that the discontinuity is propagated correctly through both domains. No variation of the global mass has been detected until the end of the simulation.

Coupling with refinement in a zone of interest

The idea of this case is to show that using a coupled simulation, where certain regions are refined, can yield a comparable result for a lower cost than a refined SPH-ALE simulation. We have a coarse (1000 particles) and a fine (5000 particles) SPH-ALE reference simulation without coupling. We compare a coupled simulation to them, where the SPH particles have the same size as for the coarse simulation and the FV cells the same discretization size as the fine SPH particles. Therefore, we use 1000 SPH particles that are distributed in the whole computational domain and 200 FV cells distributed around the initial discontinuity, i.e. $x_l \in [0.48, 0.52]$m. Starting from a rather coarse discretization for the SPH domain, we use the FV domain to refine in a region of interest around the initial pressure discontinuity. Figure 4.18 compares the coupled simulation to the coarse and the fine SPH-ALE simulation. The coupled simulation is very close to the fine SPH-ALE simulation, while only 1200 calculation points were used instead of 5000. The total mass increases of about $2 \cdot 10^{-3}\%$ until the end of the simulation.

4.3.1.2 Sinusoidal inlet velocity

Figure 4.19 shows the velocity and the pressure of a 1D coupled simulation at the same physical time t^n, where a sinusoidal velocity is imposed at the inlet of the domain and constant pressure at the outlet. We use 50 SPH particles distributed in the whole computational domain and 50 FV cells distributed in $x_l \in [0.45, 0.55]$m. That means that $r = 0.02$m and $\Delta x = 0.002$m. The discretization size of the FV cells is ten times smaller than the discretization size of the particles. This corresponds to the fact that we want to use the FV mesh to obtain a refined solution. In our target applications, the refined FV mesh will be often used to simulate the region close to solid wall boundaries where pressure and velocity change rapidly. In this case the mesh size will be much smaller than the particle size.

In Figure 4.19 it can be seen that the sinusoidal velocity and the pressure are transferred without any problems from the SPH to FV domain and back again to the SPH particles.

4.3.2 Two-dimensional testcases

The two-dimensional testcases are inviscid flow simulations. In this section, all SPH particles are in Eulerian motion but Lagrangian applications are presented in the next chapter. An explicit second order Runge-Kutta time integration scheme, also called Heun scheme Eq. (3.93), was used with the same time step sizes for the FV and the SPH domain. The Wendland C^4 kernel,

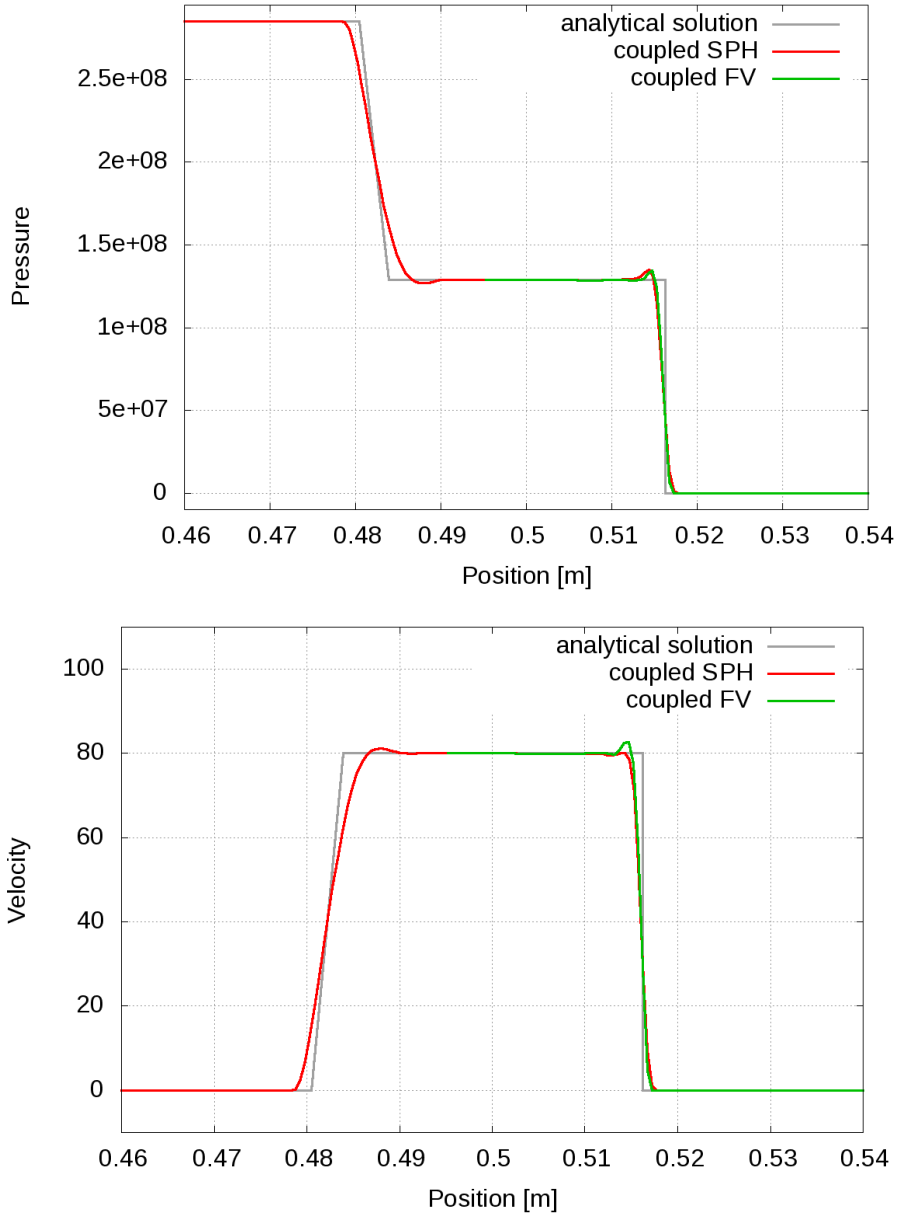


Figure 4.17: Pressure (top) and velocity (bottom) of the coupled shock tube simulation. SPH particles are situated everywhere in the domain while the FV domain extends from $x \in [0.5, 0.6]$m. The analytical solution is indicated by a thin grey line, the coupled SPH simulation by a red line and the solution on the FV domain by a green line.

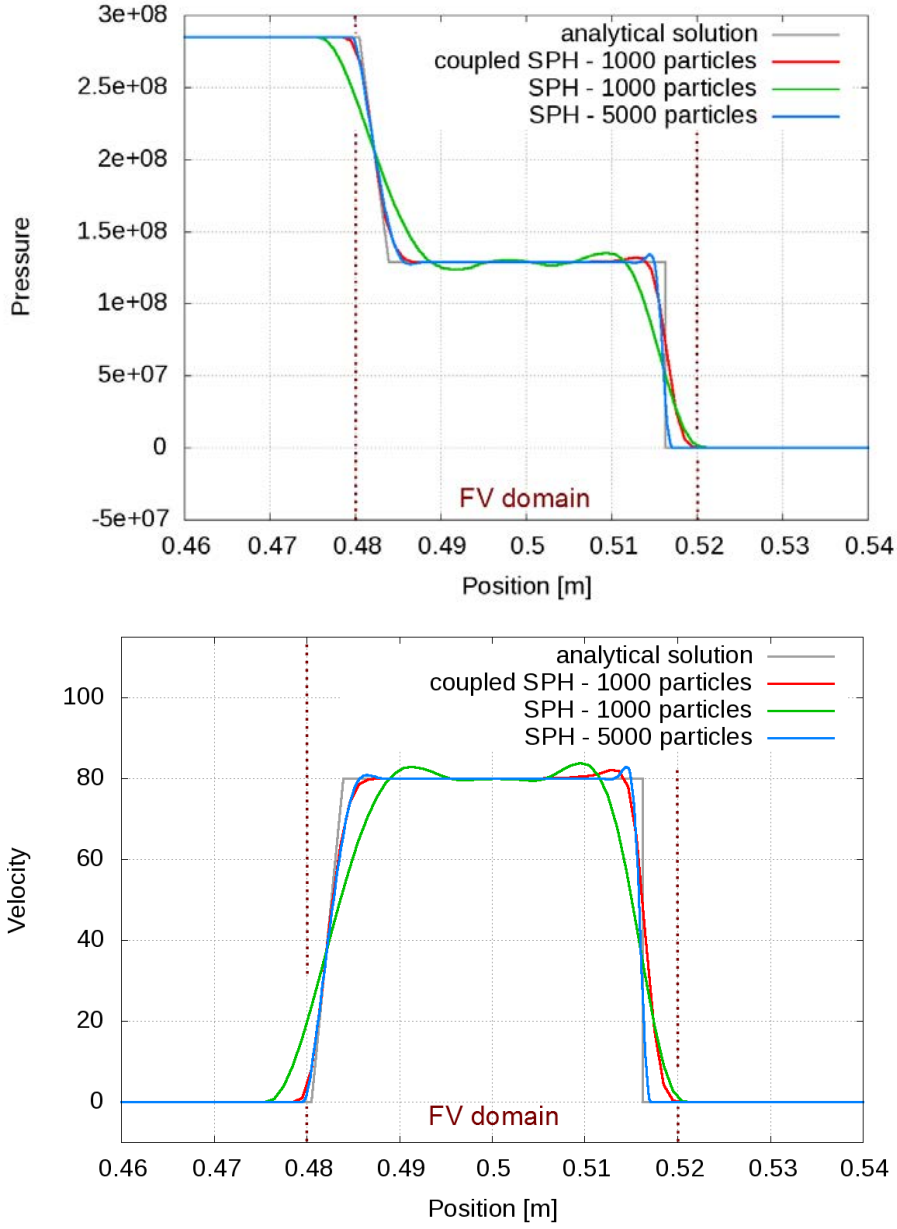


Figure 4.18: Pressure (top) and velocity (bottom) of the coupled shock tube simulation. SPH particles are situated everywhere in the domain while the FV domain extends from $x \in [0.48, 0.52]$m. The analytical solution is indicated by a thin grey line, the coupled SPH simulation by a red line with the corresponding FV domain indicated by a dashed red line. This is compared to a fine (blue line) and a coarse (green line) SPH simulation (without any coupling).

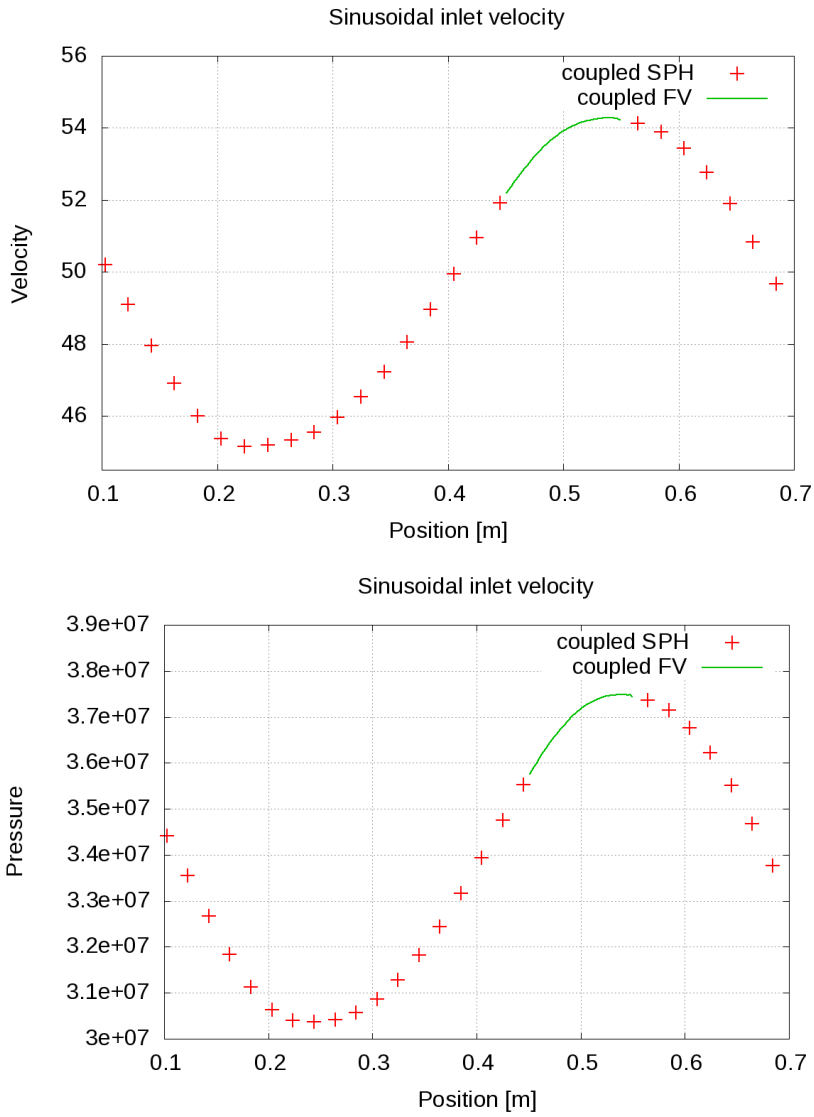


Figure 4.19: Velocity (above) and pressure (below) of a coupled 1D simulation. A sinusoidal velocity is imposed at the inlet situated on the left hand side ($x = 0$m), while constant pressure is imposed at the outlet ($x = 1$m). The green line indicates the solution obtained by the Finite Volume methods, while the red points represent the SPH particles in Lagrangian motion. The particles that are covered by the FV cells are not shown.

Eq. (3.47), was employed for the SPH space integration and the closed box correction (see Section 3.9.3) was applied.

4.3.2.1 Taylor-Green vortices

The Taylor-Green vortices are an exact steady-state solution of the Euler equations on the bi-periodic 2D unit square. They were presented in detail in Chapter 2, Section 2.1. This test case is reused to validate the coupling method.

For this purpose, a simulation was launched to analyze in particular the physical field variables in Zone 2 (see Figure 4.16) at the boundary of the FV domain. The SPH domain consists of 160×160 particles in the unit square $[0, 1] \times [0, 1]$. A regular FV mesh is added in the middle of the SPH domain with 147×147 uniform cells in $[0.257, 0.743] \times [0.257, 0.743]$ plus two layers of Chimera cells. Figure 4.20 shows the pressure and the magnitude of the velocity field of the Taylor Green vortices after $t = 4s$, plotted on the SPH particles. The contour lines are very smooth and no visible errors are introduced by the coupling. Figure 4.21 confirms this by showing the velocity plotted on a coordinate line in a region around the boundary of the FV domain. As mentioned above the coupling algorithm is not conservative. In this case the loss of global mass is smaller than 0.005% after $4s$ and therefore insignificant.

Influence of the refinement factor on the computation of the modified weights

In Section 4.2 it was explained that additional integration points are used for the computation of the modified weights and positions, similar to Newton-Cotes quadrature formulae. The number of points is determined by a refinement factor α that defines a reference length $L := \alpha\, \omega_j^{\frac{1}{d}}$, see Eq. (4.27). The reference length gives a target edge size for the auxiliary cells. A FV cell l is subdivided into smaller cells with an edge size close to L. Note that L does not depend on the FV mesh but on the SPH discretization size.

In the following, we use the Taylor-Green vortices to study the influence of α on the error of the modified weights and on the resulting pressure field. For that purpose, three different regular FV meshes are used, each of them covering the whole computational domain. *Mesh 1* is a non-uniform mesh with 200×250 cells, *Mesh 2* is a coarse uniform mesh with 150×150 cells and *Mesh 3* is a fine uniform mesh with 250×250 cells. Bi-periodic boundary conditions are imposed on the FV domain. A domain of 59×59 static SPH particles is added in the middle, see Figure 4.22 (left). This set up is chosen because the analytical value of all modified weights $\bar{\omega}_j$ is known. All particles are covered by FV cells and $\bar{\omega}_j$ should be zero. That means that all particles have FV neighbours exclusively and the particles are passive, i.e. their fields are updated in time but they do not contribute to the field variables of the other particles or cells.

Figure 4.22 (right) shows the modified weight, accumulated over the whole SPH domain, as a function of the refinement factor α for all three meshes. If we choose $\alpha = 1$, the errors are quite big for all meshes, especially for the coarse one as expected. The error decreases rapidly by decreasing α and for $\alpha = \frac{1}{8}$ all meshes lead to the analytical value of $\bar{\omega}_j = 0,\ \forall j \in \Omega_{SPH}$. The meshes used in the example are regular which is not always the case in other validation cases or applications. In the following, $\alpha = \frac{1}{16}$ will be chosen, if not stated differently, to ensure that no wrong contributions are added for the SPH space integration.

Figure 4.23 shows the modified weights for *Mesh 1*. If $\alpha = \frac{1}{4}$ all modified weights are zero for this mesh. In addition, it can be seen in the figure how the resulting pressure field on the SPH particles is strongly perturbated if the modified weights are not computed correctly. The pressure is plotted after a very short physical time, $t = 5 \cdot 10^{-3}s$, corresponding to thirty iterations.

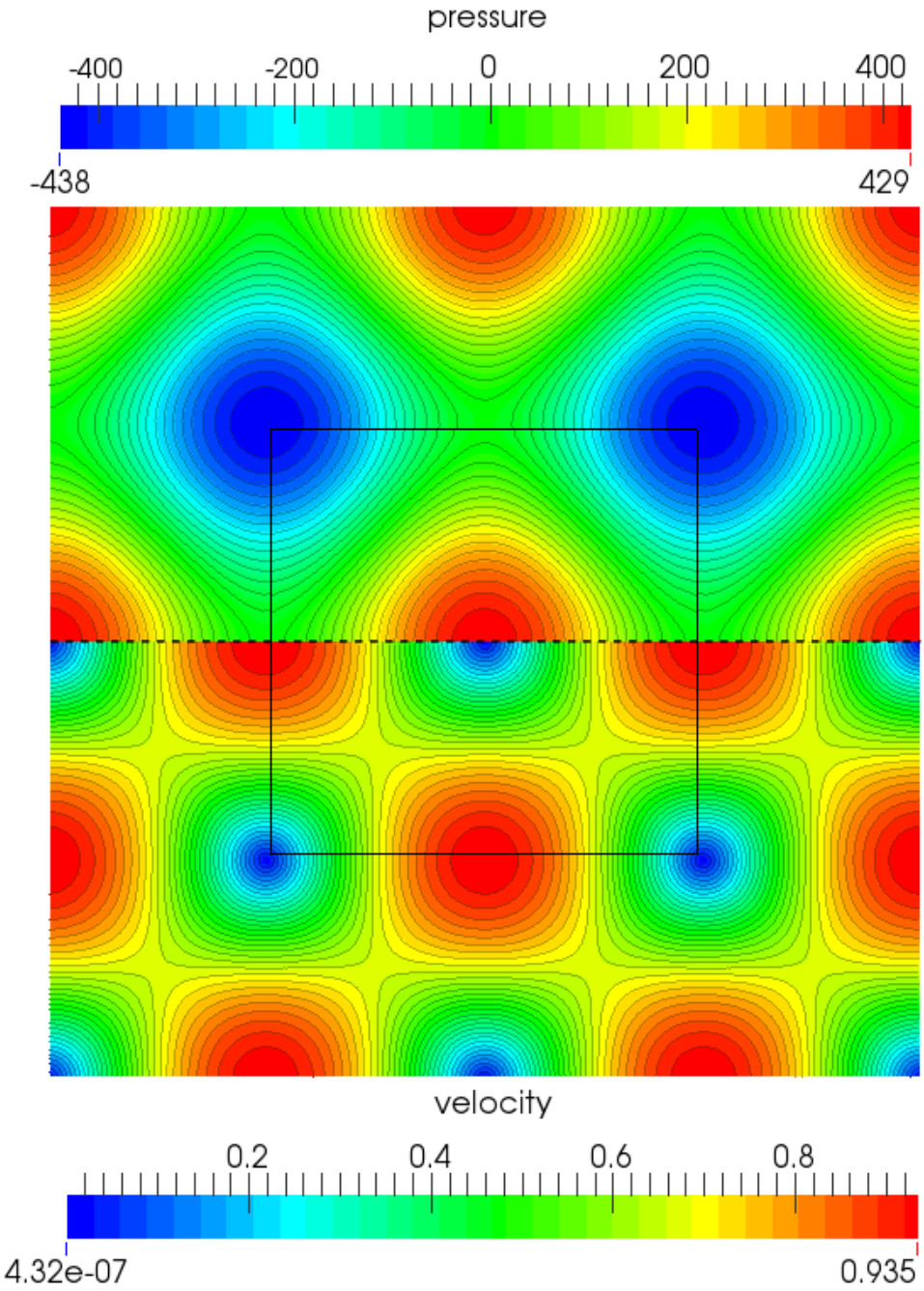


Figure 4.20: Pressure (above) and magnitude of the velocity field (below) after $t = 4s$. The particles inside the black square are covered by a uniform FV mesh. There are 160×160 particles in a square $[0, 1] \times [0, 1]$ and 147×147 uniform cells in $[0.257, 0.743] \times [0.257, 0.743]$ plus two layers of Chimera cells.

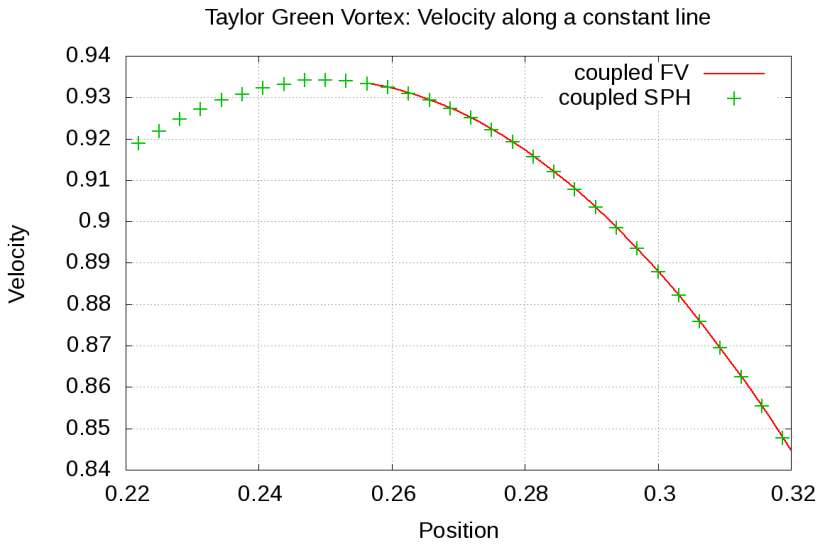


Figure 4.21: The z component of the velocity plotted at a constant line at $t = 4s$ in a region around the boundary of the FV domain.

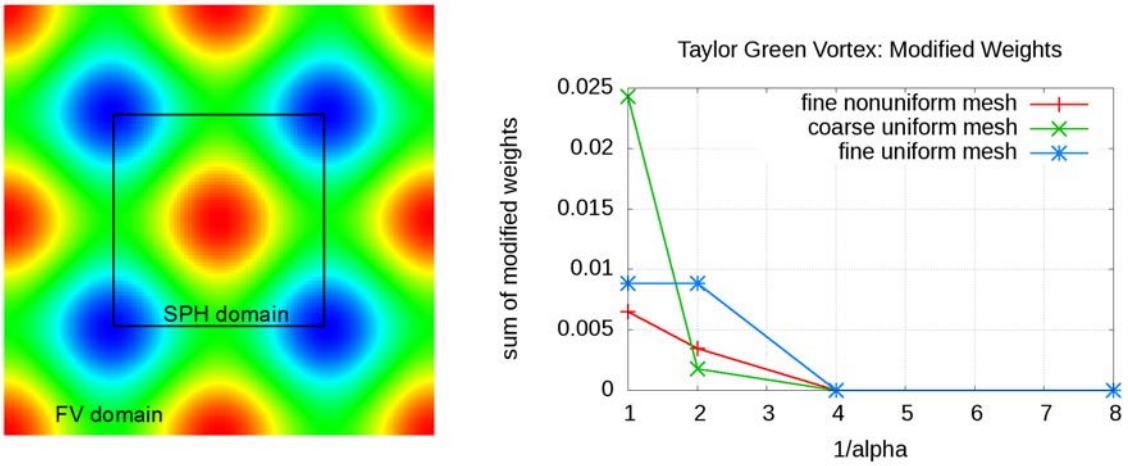


Figure 4.22: The whole computational domain is covered by FV cells and SPH particles are only added in the middle of the domain (left). Every SPH particle is completely overlapped by the FV mesh and has only FV neighbours. All modified weights ω_j should be zero. On the right side, $\sum_{j \in \Omega_{SPH}} \bar{\omega}_j$ is plotted as a function of the refinement factor α.

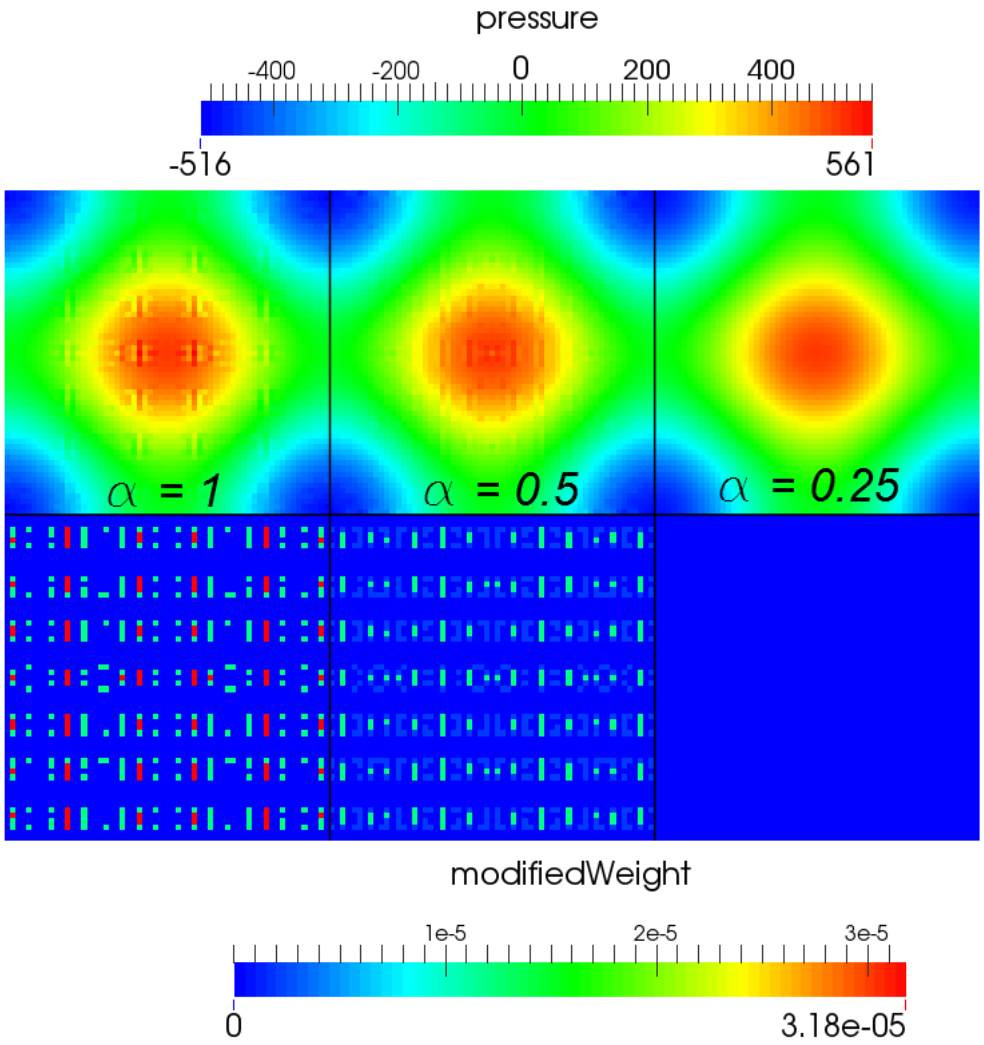


Figure 4.23: Influence of the refinement factor α on the accuracy of the computation of the modified weights $\bar{\omega}_j$. The whole SPH domain shown here is covered by FV cells (see Figure 4.22) and all modified weights should be zero. The errors in the computation of the modified weights deteriorate considerably the pressure field computed on the SPH particles by the coupling, shown here after 30 iterations ($t = 0.005s$).

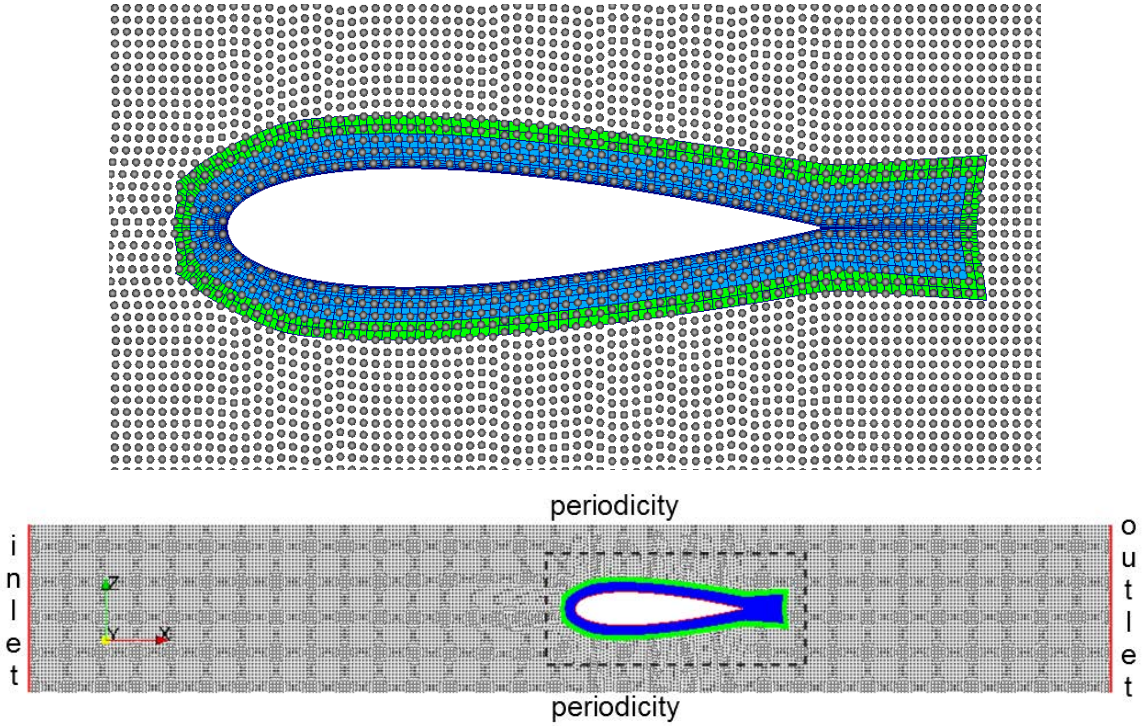


Figure 4.24: The computational domain for the flow simulation around a static symmetric NACA airfoil. The inlet is situated at the left hand side and the outlet at the right hand side. Periodicity conditions are imposed elsewhere. A FV mesh covers the SPH particles around the solid geometry that is indicated by a red line. The Chimera cells of the mesh are coloured in green and the fluid cells in blue. The above figure shows a zoom of the zone around the NACA profile (indicated by a black dashed box in the figure below). SPH particles are coloured in grey.

4.3.2.2 Flow around a symmetric NACA hydrofoil

It was explained in Chapter 1 that our target applications are transient flow simulations in hydraulic machines like Francis turbines and pump-turbines. An important validation case for these applications is the flow around a symmetric NACA hydrofoil that was introduced in Chapter 2, Section 2.2. For the coupled simulations, a block-structured mesh is overlapping the SPH particles around the hydrofoil. Figure 4.24 shows the computational domain with SPH particles, colored in grey, the FV mesh colored in blue and two layers of Chimera cells that are colored in green. At the inlet, constant velocity $u_{in} = 0.1$ m/s is imposed, at the outlet constant zero pressure, $p_{out} = 0$ Pa, and the numerical sound speed is chosen equal to ten times the inlet velocity.

The SPH particles are initially distributed by using the packing algorithm of Colagrossi *et al.* [19] and are static throughout the simulation (Eulerian description).

A reference solution is provided by an inviscid, steady-state, pseudo compressible in-house FV solver of ANDRITZ that was presented in Section 2.2. It is important to note that this Euler solver is not the same as the one used for the coupling but a solver that is well-established inside the ANDRITZ group for simulations of internal flows.

Several simulations with and without the coupling have been launched, compared and analyzed. Table 4.1 gives an overview over the ran simulations. All simulations are inviscid which

Table 4.1: Different simulations launched for the NACA validation case

	Simulation type	SPH	SPH coupled	FV	No of particles/cells
1	SPH coarse, $r = 2mm$	YES	NO	NO	15808/-
2	SPH intermediate, $r = 1mm$	YES	NO	NO	63728/-
3	SPH fine, $r = 0.5mm$	YES	NO	NO	254714/-
4	SPH coupled coarse, $r = 2mm$	NO	YES	YES	15808/2356
5	SPH coupled intermediate, $r = 1mm$	NO	YES	YES	63728/10448
6	FV reference solution	NO	NO	YES	-/24320
7	FV solution, no wall refinement	NO	NO	YES	-/24320

means that no wake should appear downstream of the solid body. However, launching a coarse non-coupled SPH simulation shows a very strong wake. This non-physical wake can be reduced by either employing a more accurate solver or using more calculation points in the zone where the wake is created, i.e. close to the solid wall.

The first three non-coupled SPH simulations in Table 4.1 are used to show the influence of the particle size r to the artificial wake. Figure 4.25 confirms that reducing the particle size decreases the numerical dissipation of the scheme. It shows the velocity field around the NACA hydrofoil as well as the x component of the velocity u plotted on a line with $x = 0.55$m downstream of the NACA geometry. For these simulations no special visibility criterion was used at the trailing edge which intensifies the wake. A particle situated near the trailing edge on one side of the NACA "sees" the particles on the other side of the NACA which is not physical and does not occur in FVM. To test the influence of it in this particular case, a simple visibility criterion was implemented. The visibility criterion excludes from space integration the particles "on the other side" of the NACA and the surface elements with normals that are not oriented in the direction of the considered particle. This criterion greatly improves the results but we are still not able to reproduce the velocity field of the reference solution even though the reference solution uses more than ten times fewer cells than we use particles, see Figure 4.26.

In order to confirm that the large errors in the velocity field occur because of a lack of spatial resolution and not because of the SPH scheme itself, another FV simulation is launched. For that purpose, the mesh for the FV reference simulation is modified in a way that the number of cells stays the same, see Table 4.1, but the distribution of the nodes changes. The new mesh shows almost no refinement in direction normal to the NACA solid geometry. Figure 4.27 compares the two meshes, one refined and one with almost no refinement close to the wall. A comparison of the velocity field of the two FV simulations is given in Figure 4.28. As we expected the velocity field for the unrefined FV simulation is strongly decelerated at the solid wall because the spatial resolution together with the precision of the scheme are not able to capture the rapidly changing gradients. We conclude that the SPH particle size has to be adapted according to the discretization size of the FV mesh of the reference solution in order to obtain similar results.

Taking a look at the mesh in Figure 2.5 and Figure 4.27, it becomes evident that it would be much too computationally expensive to have particles with a constant size everywhere that is equal to the smallest edge of a cell. The smallest edge in the mesh that was used to obtain

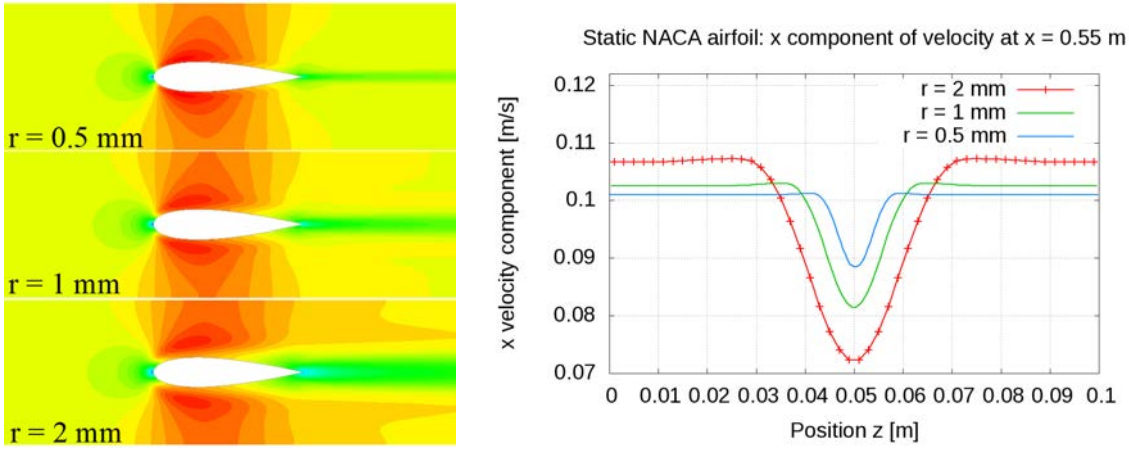


Figure 4.25: Velocity field around a static symmetric NACA hydrofoil computed with different particle sizes r (left). The numerical wake is reduced by reducing the particle size r (right) because the numerical dissipations reduces. No visibility criterion was used for this comparison.

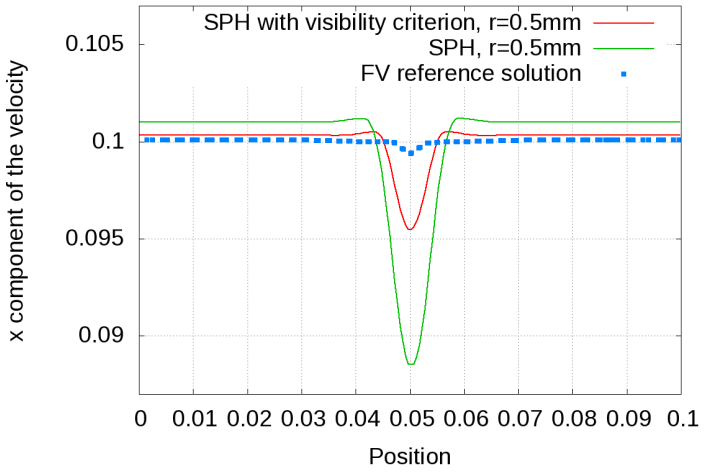


Figure 4.26: The numerical wake plotted for a refined non-coupled SPH simulation with $r = 0.5mm$ is further reduced by employing a visibility criterion. However, even though ten times as many particles are used as cells for the FV reference solution, the numerical wake is still too important.

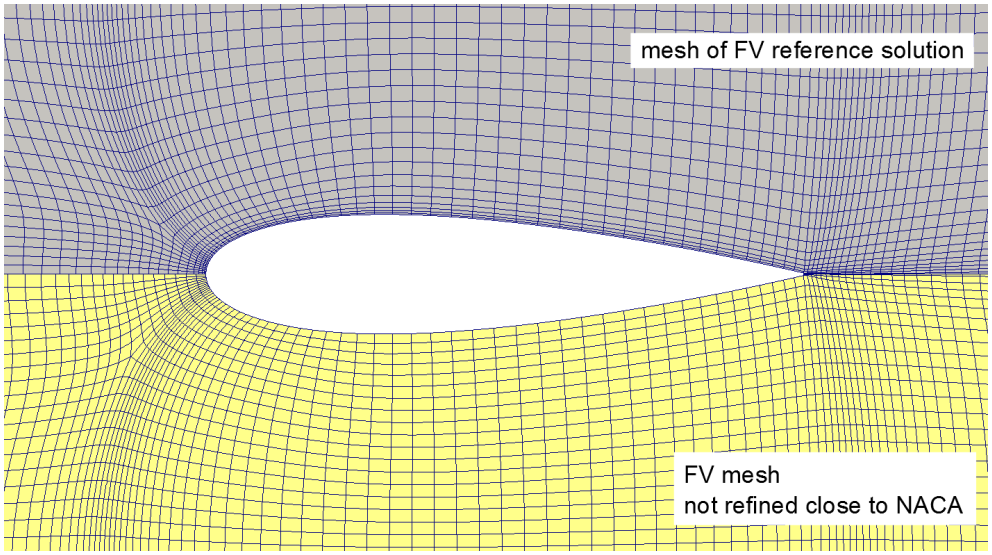


Figure 4.27: Comparison of a mesh with refinement close to the solid wall boundary (above) and a mesh without refinement close to the solid (below). Both meshes contain the same number of mesh points.

the reference solution is $\Delta x = 0.1$mm and the second smallest $\Delta x = 0.25$mm. Even if we use the second smallest size $\Delta x = 0.25$mm we would end up with four times as many particles as in the fine simulation, which would be fourty times more than in the FV reference simulation.

In contrast, the coupled simulations allow us to refine in an anisotropic way. We launch simulations with the coarse and the intermediate particle size and couple them to a mesh that is constructed according to Condition (4.20). That means that the mesh used for the intermediate SPH particle size is more refined than the one used for the coarse particle size. Both meshes do not correspond to the one used for the reference solution because this mesh contains cells with edges that are even bigger than the coarse particle size. But all of the used meshes are block structured meshes generated with the same in-house mesh tool.

Figure 4.29 shows the modified weights for the coarse coupled simulation, computed by (4.23) with additional quadrature points. In Figure 4.30 the particles are colored by the closed box error $\mathbf{b}_i$ that was presented in Chapter 3, Section 3.9.3. This error is a measure for the disorder of the particles in the point of view of the discrete SPH operators.

In Figure 4.31 the effect of the coupling is illustrated by showing the velocity field for two simulations with both particles of the coarse size of $r = 2$mm, one simulation without coupling and one coupled simulation. The SPH simulation shows a very strong numerical wake as it was discussed above. In the coupled simulation the numerical wake almost disappears because the FV mesh locally refines the simulation anisotropically with long and thin cells close to the wall. Again the x component of the velocity is plotted on a line of constant x downstream of the NACA in Figure 4.32. On the left hand side, the numerical wake of the coupled simulation is compared to the one obtained by the non-coupled SPH simulation. On the right hand side, a detailed view shows the numerical wake of the coarse and the intermediate coupled simulations. For both the velocity is varying less than 0.5% but reducing the discretization size further decreases the numerical dissipation as it is expected.

In Figure 4.33 the comparison between the velocity field of the intermediate coupled simulation $r = 1$mm and the FV reference solution can be seen. It compares very well, but the

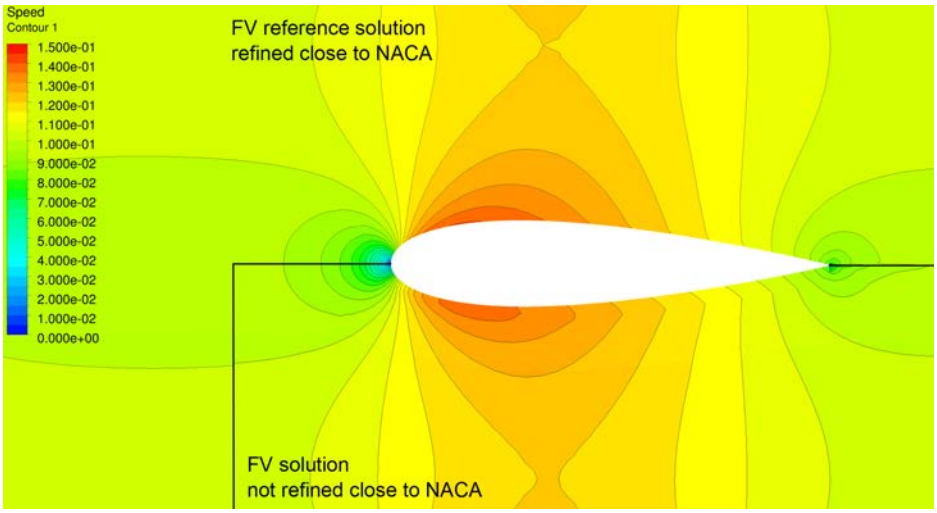


Figure 4.28: Comparison of the magnitude of the velocity field computed on a mesh with refinement close to the solid boundary (above) and without refinement close to the solid boundary (below). The velocity is strongly decelerated close to the wall if no mesh refinement is applied. This deceleration and the resulting numerical wake are not physical.

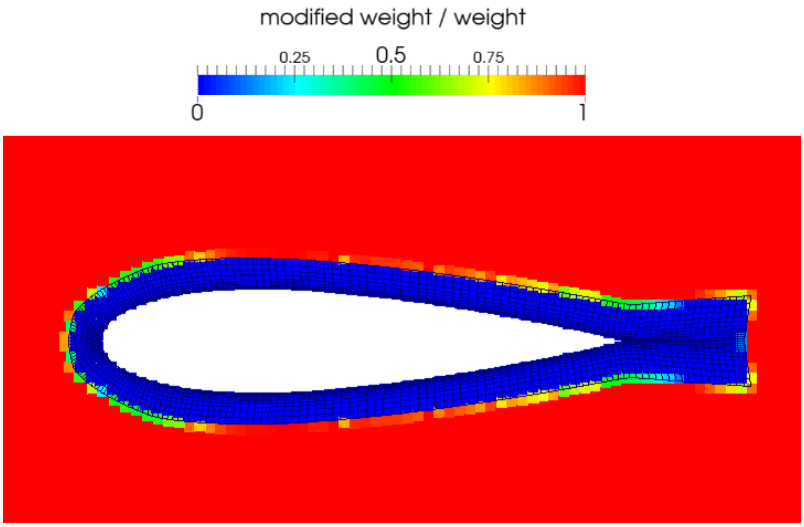


Figure 4.29: FV mesh around a symmetric NACA hydrofoil. The particle size is $r = 2mm$ and the mesh is constructed accordingly, verifying Condition (4.20). It can be seen that the modified weights $\bar{\omega}$ are equal to ω if they are not covered by the mesh, and zero if they are completely overlapped.

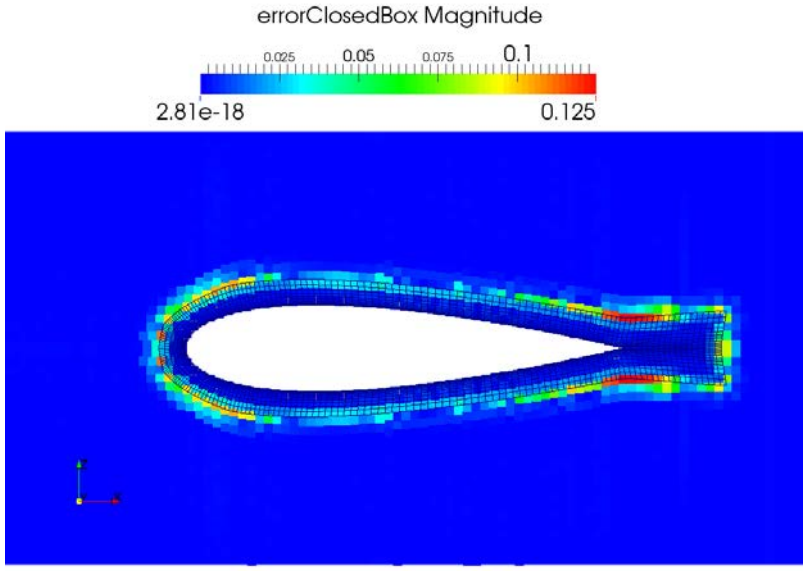


Figure 4.30: FV mesh around a symmetric NACA hydrofoil. The particles are colored by the closed box error that was introduced in Chapter 3, Section 3.9.3.

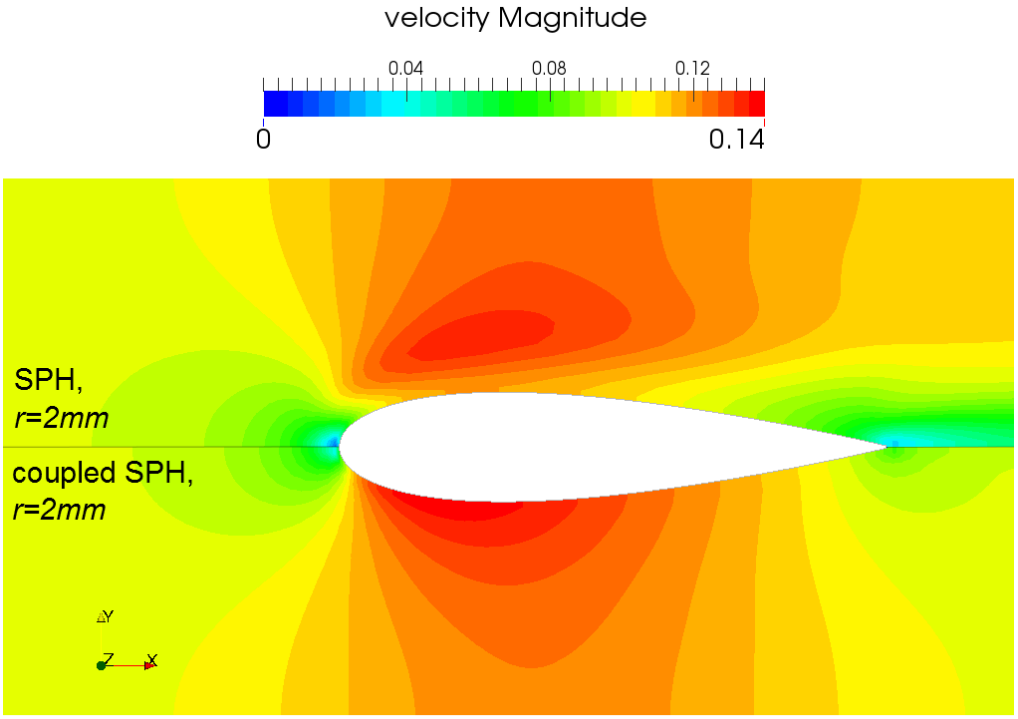


Figure 4.31: Magnitude of the velocity field around a symmetric NACA airfoil with particles of $r = 2$mm, only SPH above and a coupled simulation below. Since this is an inviscid simulation the wake that appears for the SPH simulation without coupling is purely numerical and nonphysical. The coupling leads to an almost complete disappearance of the artificial wake.

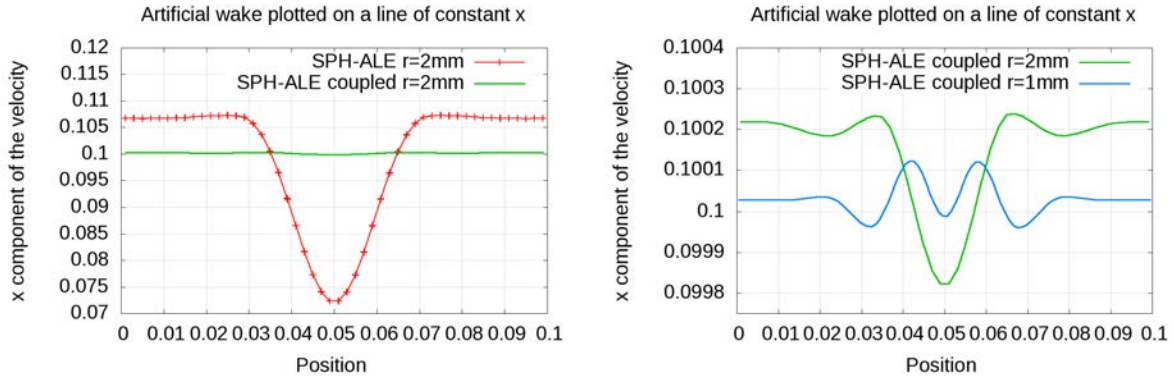


Figure 4.32: The x component of the velocity plotted on a constant line downstream of the NACA. The wake that is seen for the SPH simulation without coupling (left) is purely numerical and non-physical. A detailed view (right) shows that the artificial wake still exists for the coupled simulation but it is very small and reduces if the particles and the mesh are refined.

velocity contours differ a little far away from the NACA geometry. It is in this region where the coupled SPH simulation is more refined than the FV reference simulation because of the anisotropic mesh that has larger cells far away from solid boundaries and that is not restricted by the SPH particle size as the coupled simulation.

Figure 4.34 shows the pressure and the velocity field for the refined coupled simulation plotted on the SPH particles. It confirms that no visible errors are introduced by the coupling between the overlapping domain and $\Omega_{SPH} \setminus \Omega_{FV}$.

We define the pressure coefficient as in Eq. (2.3). The density and velocity at infinity are $\rho_\infty = 1000$ kg/m^3 and $v_\infty = 0.1$ m/s and the analytical value for the pressure at the stagnation point is $p_{stag} = 5$. Figure 4.35 shows that the pressure coefficient computed at the wall of the FV domain obtained by the coupled simulation with $r = 1$mm and the FV reference solution compare very well. Note that the pressure computed at the wall is obtained from the FV solver that receives its' boundary conditions from the SPH domain.

In this validation case we model an inviscid, incompressible flow even though we use a weakly compressible formalism. According to Bernoulli's principle, the total pressure p_{tot} is constant in incompressible irrotational flows throughout the whole simulation in the whole computational domain. This is used in the following to introduce another measure for the numerical dissipation, which is the loss (change) of total pressure, evaluated as the difference between the total pressure integrated over a section upstream of the NACA and a section downstream, defined in Section 2.2. Table 4.2 shows the error of the integrated total pressure on the section next to the inlet and next to the outlet compared to the analytical value. In the last column the change of total pressure between the two sections is given, independently of the analytical value.

For the non-coupled simulations the total pressure loss reduces with the particle size but it does not reach the same order of magnitude as in the reference simulation. In order to reach the same level of accuracy as for the FV reference solution, the coupling is needed, similar to what was found above concerning the numerical wake. The coupled simulation with $r = 1$mm performs even better than the FV reference solution but this is not surprising because it is more refined than the FV reference solution. Recall that the FV reference solutions is obtained using more cells than the coarse coupled simulation ($r = 2$mm) uses cells and particles together. But the total number of particles and cells for the intermediate coupled simulation ($r = 1$mm) is

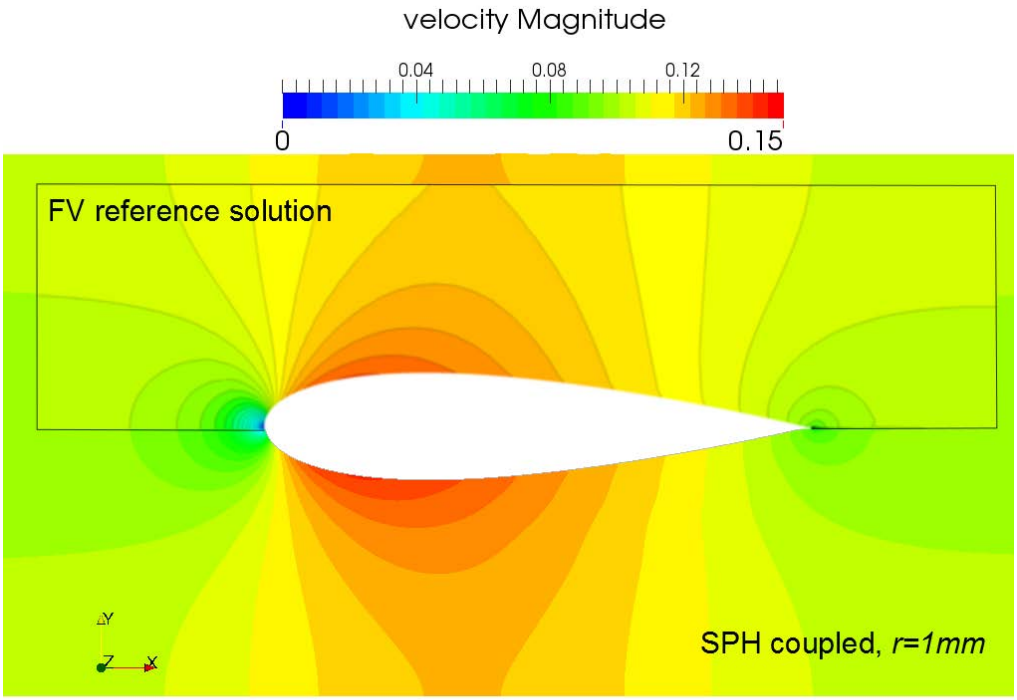


Figure 4.33: Comparison of the magnitude of the velocity field obtained by the intermediate coupled simulation and the FV reference solution. Both of them do not a visible numerical wake and the velocity contours compare well.

Table 4.2: Loss of total pressure and error at sections upstream and downstream

	Simulation type	p_{tot} error [%] upstream	p_{tot} error [%] downstream	p_{tot} change [%]
1	SPH coarse	7.98	−4.5	−11.55
2	SPH intermediate	4.6	−0.28	−4.67
3	SPH fine	2.57	0.71	−1.81
4	SPH coupled coarse	0.42	0.31	−0.12
5	SPH coupled intermediate	0.05	0.06	0.011
6	FV reference solution	−0.03	−0.13	−0.1

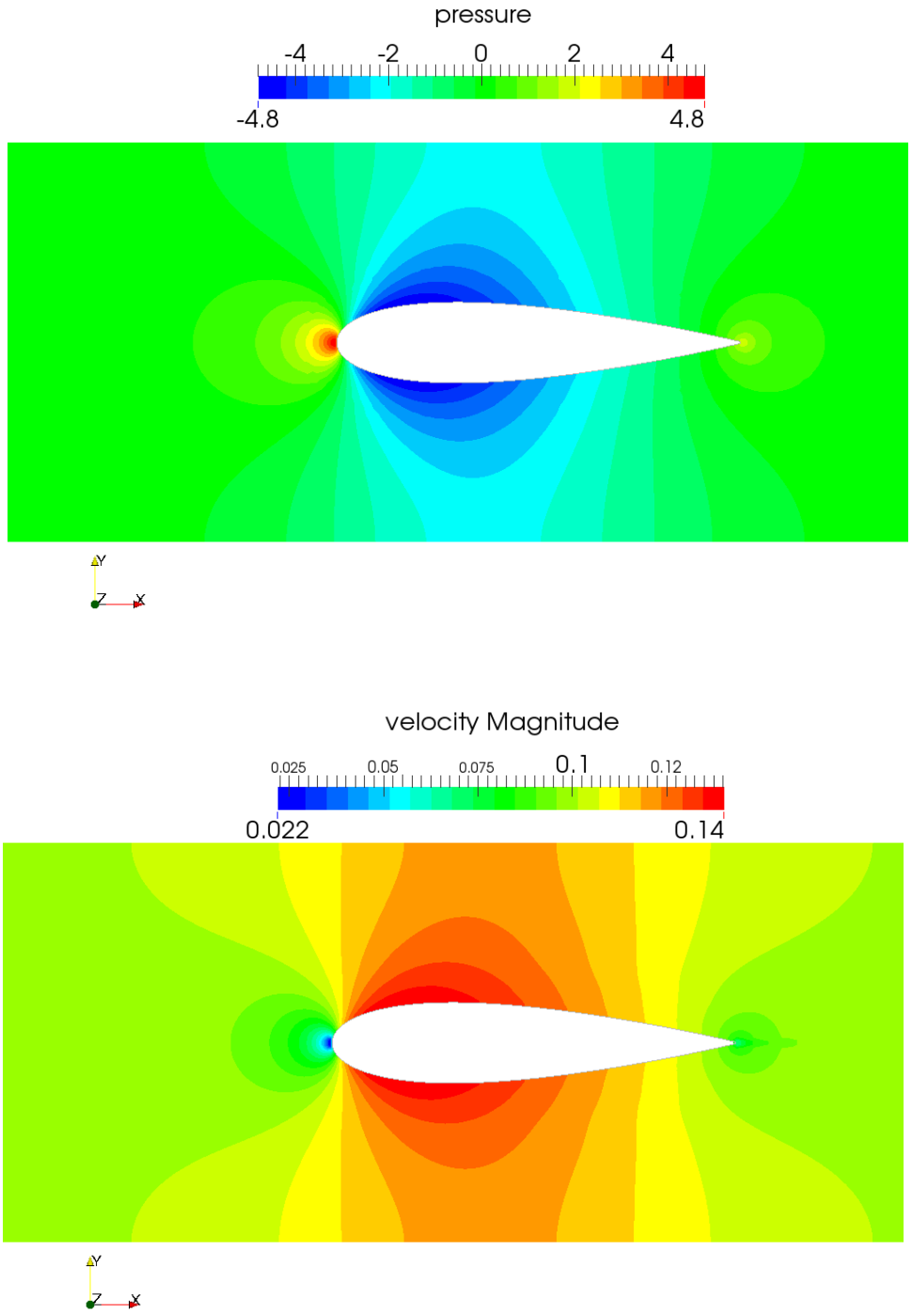


Figure 4.34: Pressure and the magnitude of the velocity field around a static NACA hydrofoil obtained by a coupled simulation with a fine discretization size. The particles have the size $r = 1$mm and the mesh is adapted according to Condition (4.20).

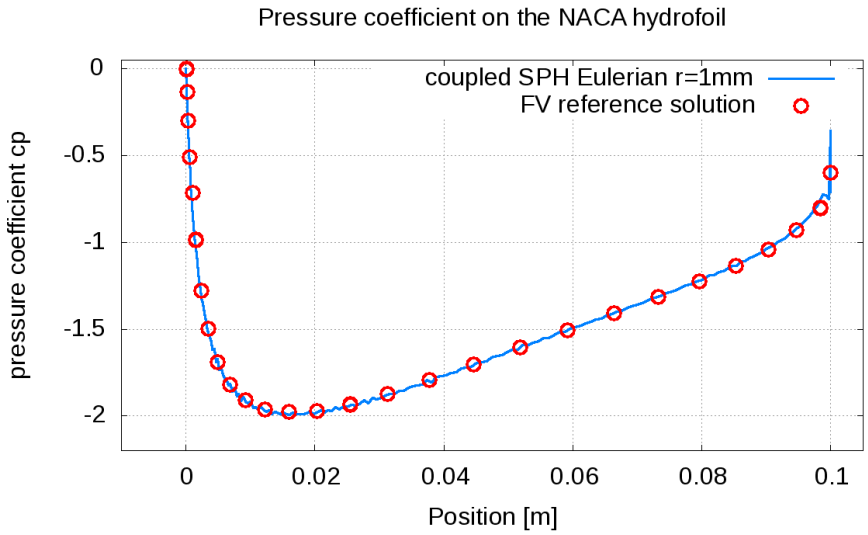


Figure 4.35: Comparison of the pressure coefficient c_p computed with a coupled SPH simulation with particles of size of $r = 1$mm and a FV reference solution. The SPH particles are in Eulerian motion.

larger than for the FV reference simulation, see Table 4.1.

4.4 Summary and discussion

A flexible coupling algorithm for SPH-ALE and FVM was developed that is based on a multi-domain approach of overlapping SPH and FV domains. Information is transferred in two ways. On the one hand, boundary cell values are interpolated from the SPH particles to the FV domain, while on the other hand, the FV cells are used as neighbours for the SPH integration. Particles that are completely covered by FV cells are moving passively through the FV mesh, having their field variables updated in each time step without contributing themselves to the integration of the other particles or cells.

The method was validated by means of well known academic inviscid test cases in one or two space dimensions. It was shown that the coupling method performs well for shock tube simulations, dynamic 1D Lagrangian simulations as well as the 2D Taylor Green vortices. The validation case that is the most significant for our target applications is the flow around a 2D static symmetric NACA hydrofoil. This testcase not only showed that the coupling works but it showed that the coupling is necessary to obtain good results for a reasonable computational cost. At first non coupled SPH simulations were launched and we have seen that using a constant coarse discretization size in the whole domain leads to the creation of large numerical errors around the solid geometry, even in the case of inviscid simulations. The numerical errors manifest themselves in the form of an artificial wake. Coupling a SPH simulation with a strongly refined FV mesh, reduces the wake drastically. The coupled simulation compares very well to a reference simulation obtained by an in-house FV solver. Both, velocity and pressure field show a very good agreement, just as good as the pressure coefficient at the solid wall.

In the next chapter, Chapter 5, the coupling method will be applied to SPH particles in Lagrangian motion. Adding a FV mesh in order to impose outlet boundary conditions to

Lagrangian moving particles, opens the door to many interesting applications.

Chapter 5

Applications

The objective of this thesis is the development of a flexible coupled SPH-ALE/FV method that can be employed for the investigation of transient flows in hydraulic machines. For that reason, the following properties are important. The method combines the strong points of SPH and FVM. On the one hand, it is applicable for particles in Eulerian, Lagrangian and ALE motion, while on the other hand all kind of meshes, unstructured or structured, moving or static are usable. Note that we do not investigate free surface flows in the frame of this project and no free surface treatment was implemented in the FV solver. Until now, we only considered SPH particles in Eulerian motion and fixed meshes for the validation of the coupling algorithm. This is not due to a limitation of the coupling algorithm but due to the difficulties of the SPH method to impose subsonic boundary conditions to Lagrangian moving particles. Hence, a novel approach for outlet boundary conditions is proposed in the following. The main idea is to apply the coupling method at the outlet, where a partially overlapping FV mesh is used to impose outlet boundary conditions to the SPH domain.

In Chapter 1 the start-up of a Francis turbine with imposed guide vane opening angles was identified as target application for the method. We saw that SPH-ALE has interesting advantages for this kind of applications but that local refinement and de-refinement is indispensable for a successful simulation of dynamic internal flows with stagnation points. In order to show the potential of the coupled method for these applications, a preliminary two-dimensional test is presented. For that purpose, we consider a NACA hydrofoil that is rotating about its center. The mesh around the NACA hydrofoil follows the motion of the solid geometry and guarantees a refined space discretization close to the solid during the whole simulation. Instead of Lagrangian moving particles, an ALE motion, see Chapter 3, Section 3.9.4, is employed that follows the moving geometry. Comparison with the pressure coefficient of FV reference solutions shows good agreement.

5.1 Outlet boundary conditions for particles in Lagrangian motion

In SPH it is especially difficult to impose subsonic outlet boundary conditions to Lagrangian moving particles because of the following reasons. Particles follow the flow and usually leave the domain at the outlet. When they are leaving the computational domain, they are deleted. But before their destruction, boundary values should be imposed, i.e. a constant static pressure. This has to be done in a way that the imposed boundary value can be transferred upstream. In addition, care has to be taken that the destruction of particles does not introduce errors. The particles have a certain volume and it is not obvious at what moment they should be deleted.

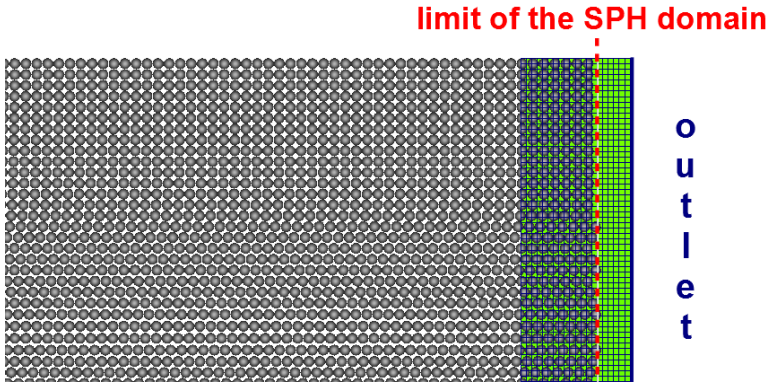


Figure 5.1: A block of FV cells is overlapping the SPH particles at the outlet. The particles are deleted before reaching the outlet and the boundary conditions are imposed by the FV mesh by means of the coupling.

Ideally, the particles would be deleted continuously, where the particle weight is reduced in correspondence to the volume that actually leaves the computational domain. However, the exact extent in space of a particle is not known and it might be quite complicated in practice to delete a particle gradually.

Instead of dealing with these problems that are due to our choice to use Lagrangian moving particles, we propose a different way of outlet boundary treatment. The idea is to use the coupling to impose boundary conditions. A block of FV cells is added in a way that it is overlapping the SPH particles at the outlet. The mesh extends further than the particles that are deleted before reaching the outlet, as it is illustrated in Figure 5.1. In this way, the boundary information is transferred from the FV mesh to the SPH particles by the coupling. As it was explained in Chapter 3, Section 3.7.4, boundary conditions are easier to implement for fixed FV meshes than moving particles because the geometrical properties do not change. From the moment on that the particles are completely overlapped by the FV mesh, they are moving passively through the mesh and do not contribute themselves to the computation of the physical fields. As a consequence, they can be deleted without any influence on the simulation. Note that it is true in general for the coupled method, that particles in Lagrangian motion can be deleted if they are completely overlapped by a mesh and if they are not needed any more downstream of their current position.

As illustrative test case, the flow around the NACA hydrofoil is chosen, that was considered in Chapter 4 for validation purpose with Eulerian particles. This is an example where it is indispensable to impose subsonic non-reflecting boundary conditions. Due to the use of the above described boundary treatment, it can be relaunched with particles in Lagrangian motion.

Flow around a static NACA hydrofoil cascade with Lagrangian moving particles

So we reconsider the test case of the flow around a static NACA hydrofoil from Chapters 2, Section 2.2, and Chapter 4 but this time with particles in Lagrangian motion, together with the previously described boundary treatment. Figure 5.2 shows the computational domain that corresponds to the one that was used in Chapter 4 but for the block of FV cells that was added at the outlet. At the inlet, once again constant velocity is imposed, directly to the SPH particles,

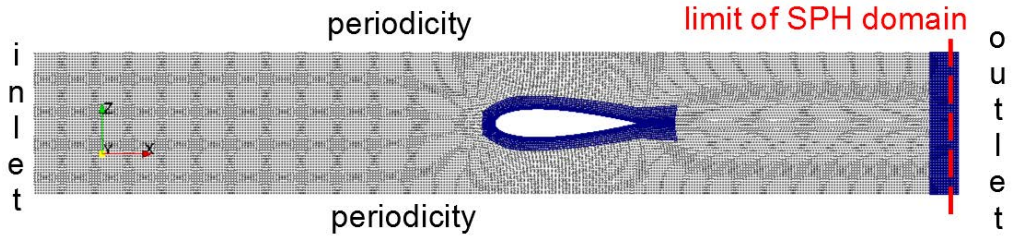


Figure 5.2: Computational domain of the simulation of a NACA hydrofoil with Lagrangian moving particles. A block of FV mesh is added at the outlet to impose outlet boundary conditions to the SPH particles.

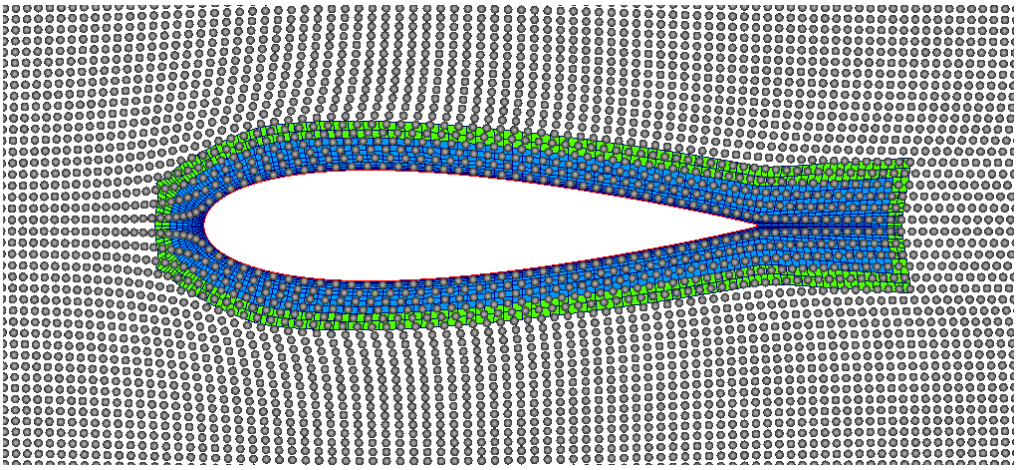


Figure 5.3: Detailed view of the distribution of the particles in Lagrangian motion around the NACA hydrofoil and the FV mesh that is used around the solid geometry. The surface elements are colored in red, the normal fluid cells in blue and the Chimera cells in green.

and at the outlet, constant pressure is imposed by the coupling

Figure 5.3 shows the Lagrangian particle distribution around the NACA and the mesh including the Chimera cells. It can be seen that the particles follow the flow and pass through the mesh without being disturbed by the mesh. The particles move according to the velocity field along lines and they are not isotropically distributed anymore, see Figure 5.4. Furthermore as a result of the Lagrangian particle motion, there are no particles at the stagnation point at the leading edge of the hydrofoil. In the case of moving particles, the modified weights $\bar{\omega}$ of the particles change at every time step since particles are moving into the mesh and out again. Figure 5.5 shows $\frac{\bar{\omega}}{\omega} \in [0, 1]$ that is again computed by the approximate method of Section 4.2.2 where it is assumed that particles are spherical.

In Figure 5.6 the velocity field obtained by Lagrangian particles is compared to the one obtained by Eulerian particles. The velocity field compares well, however, the pressure field has a slightly different form, see Figure 5.7. This is also confirmed by the pressure coefficient Eq. (2.3) in Figure 5.8 that compares less well to the reference solution than the one obtained by the Eulerian simulation.

In the case of Eulerian particles the initial particle distribution is prepared with Colagrossi's packing algorithm [19] that distributes the particles isotropically by reducing the closed box

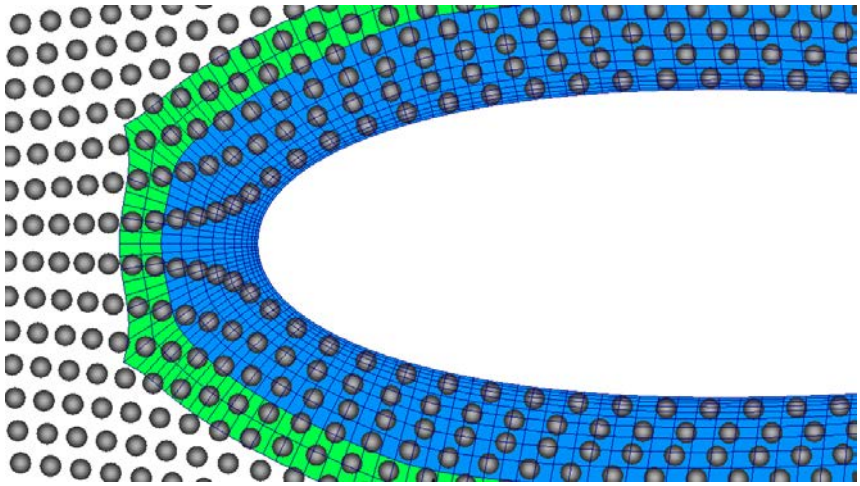


Figure 5.4: Detailed view of the distribution of the particles in Lagrangian motion close to the stagnation point. The interior FV cells are colored in blue and the Chimera cells in green.

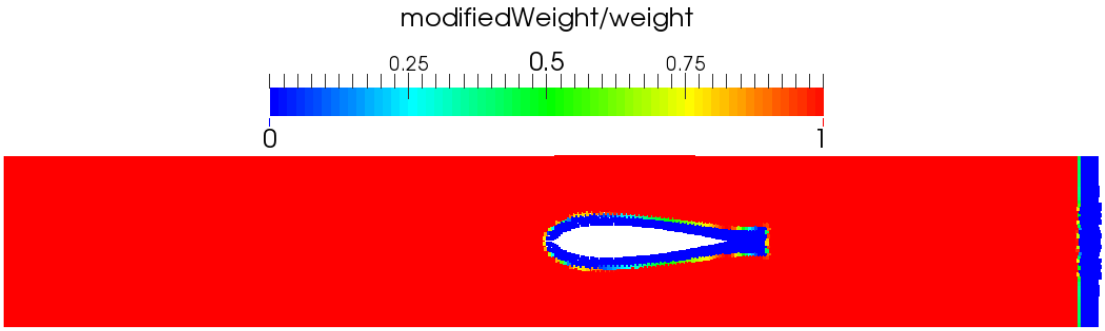


Figure 5.5: The modified weight $\bar{\omega}$ is equal to the original weight ω if the particle is not overlapped by the FV mesh. If it is completely covered, the modified weight is zero. The particles are moving in Lagrangian motion and the modified weights change at every time step.

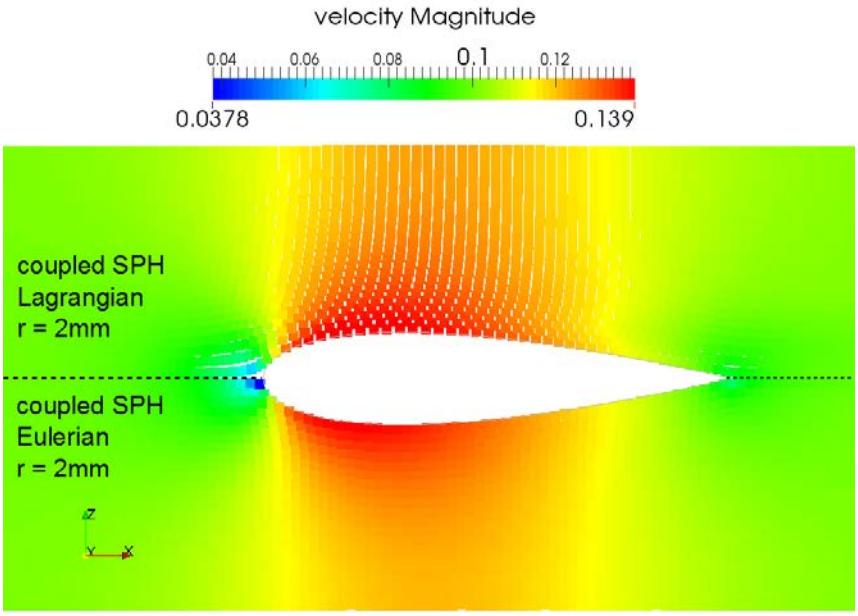


Figure 5.6: Comparison of the velocity field computed by the coupled algorithm around a NACA hydrofoil. At the top the particles are in Lagrangian motion, while they are in Eulerian motion at the bottom.

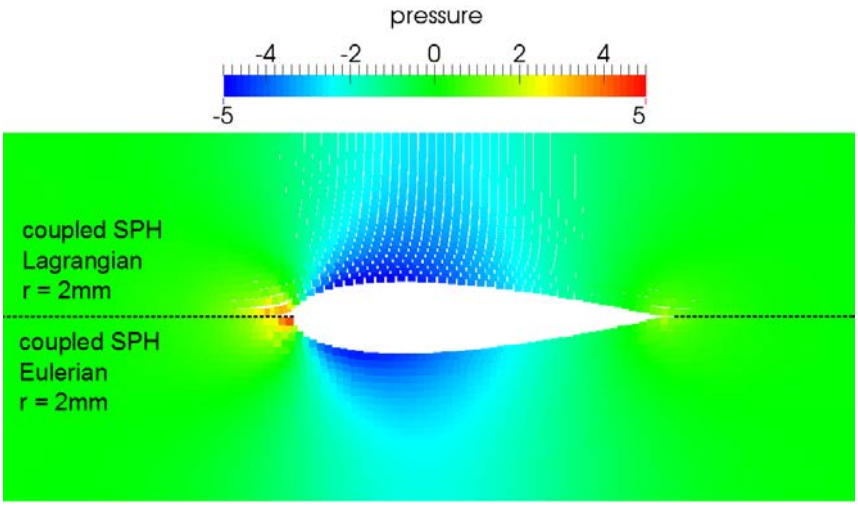


Figure 5.7: Comparison of the pressure field computed by the coupled algorithm around a NACA hydrofoil. At the top the particles are in Lagrangian motion, while they are in Eulerian motion at the bottom.

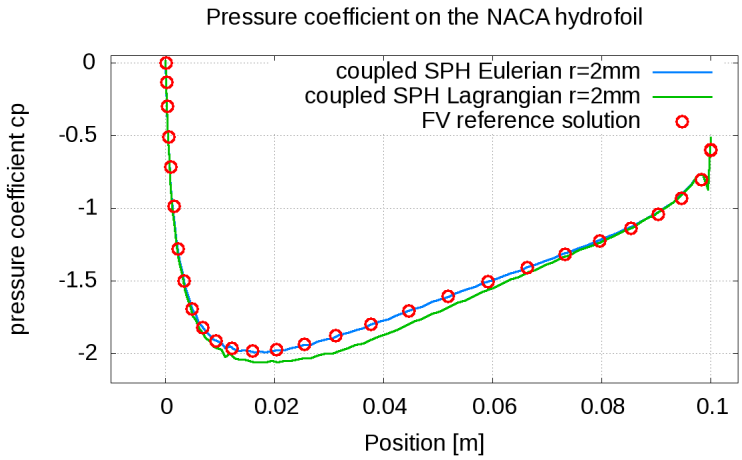


Figure 5.8: Comparison of the pressure coefficient computed by the coupled algorithm with particles in Lagrangian motion and by a FV reference solution. The Lagrangian simulation (green line) yields less good results than the Eulerian simulation (blue line).

error. In this way there are always enough particles in the neighborhood for the interpolation of pressure and velocity for the boundary values of the Chimera cells. Moreover, the approximate method that computes the intersection of a cell and a particle assumes that particles are of spherical shape which is an acceptable simplification for uniformly distributed particles but it is less true for the present particle distribution obtained by Lagrangian moving particles.

5.2 Rotating NACA hydrofoil cascade

In all the previously described coupled simulations the hydrofoil was static and the mesh was fixed. But if flows in rotating machines are considered, it suggests itself to use meshes around the solid geometry that move in the same way as the solid that they are attached to. In this way, a fine space discretization close to the solid wall boundary is guaranteed throughout the whole simulation. Entire blocks of the mesh are moved and the connectivity inside the mesh as well as the geometrical properties like volumes and surfaces do not change.

As an two-dimensional example we reconsider the symmetric NACA hydrofoil cascade from Figure 5.2. However, this time it is not static but the hydrofoil rotates with a predefined rotation speed about its center. The motion of the geometry is imposed similarly to what is done for the opening guide vanes in the Francis start-up simulations.

A first simulation was launched where the particles move in a Lagrangian way and the hydrofoil rotates with a rotation speed of $\omega = 0.03$ rad/s. As expected, the particles follow the flow and the motion of the geometry, see the particle distribution at $t = 3s$ in Figure 5.9. Figure 5.10 shows the velocity field at the same physical time. Launching the simulation with particles in Lagrangian motion, works in this case for small angles. However, since the particles follow the flow, they are not uniformly distributed anymore and might even manifest holes which can increase numerical errors drastically. As long as the particles are completely overlapped by a mesh, the particle distribution does not pose a problem for the SPH domain. However, if there are not enough particles in the vicinity of the Chimera cells, the simulation crashes because no correct boundary values can be interpolated for the FV solver. In our target application

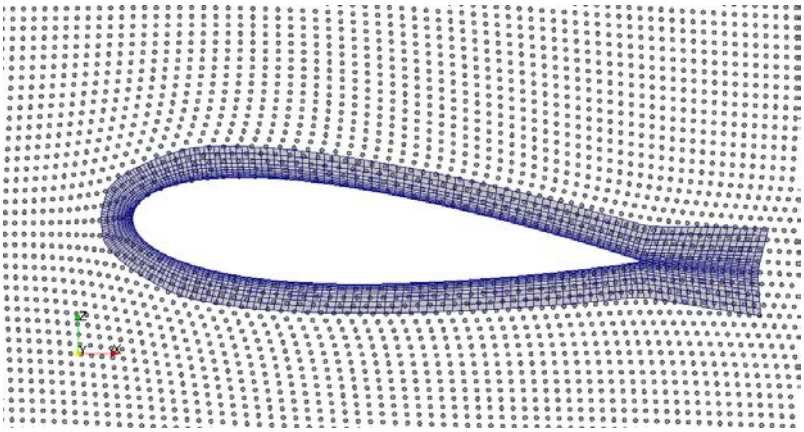


Figure 5.9: Distribution of Lagrangian particles around a rotating hydrofoil with a rotation speed of $\omega = 0.03$ rad/s at $t = 3s$.

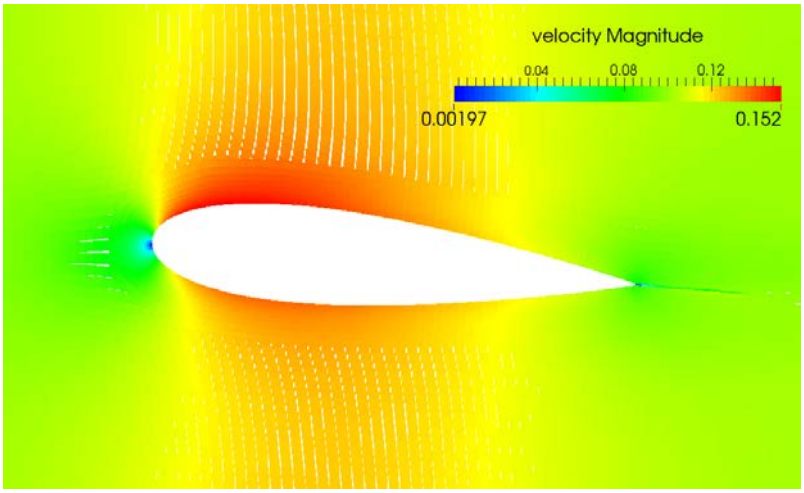
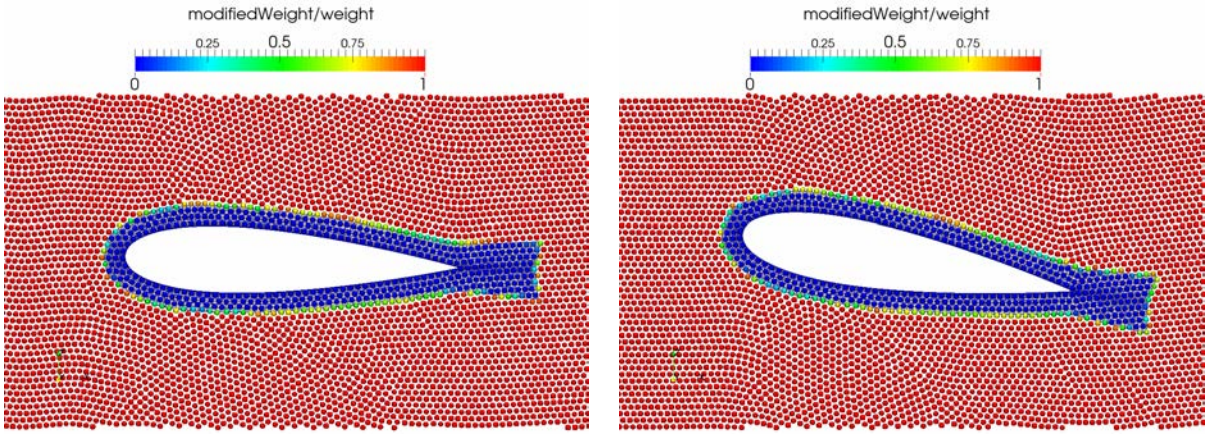


Figure 5.10: Magnitude of the velocity field plotted on Lagrangian particles and on the FV mesh that is situated around a rotating hydrofoil with a rotation speed of $\omega = 0.03$ rad/s at $t = 3s$.



(a) Position of the hydrofoil that is rotated by 2°.

(b) Position of the hydrofoil that is rotated by 10°.

Figure 5.11: Particle distribution and modified weight around the rotating NACA hydrofoil.

of transient flows, large areas of flow separation may occur which might generate holes in the resulting particle distribution. It would not be possible to predict the position of the holes and add meshes that are large enough in these regions. Other solutions to overcome this problem would be to use particle shifting that alters the position of the particles, or interpolation methods that need less calculation points. But these are not the approaches that we take.

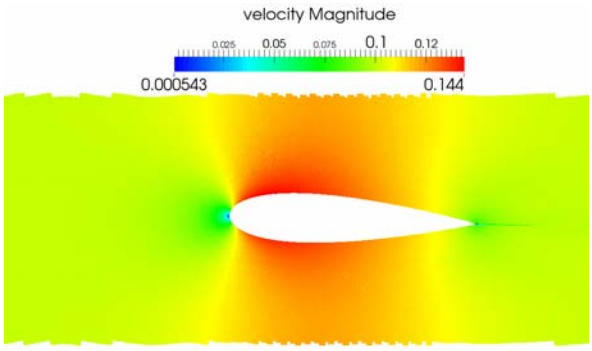
Given that we do not consider free surface or multiphase flows, the only motivation to use Lagrangian moving particles is that geometries are moving. Thus, instead of Lagrangian particles, we use a ALE particle motion that fills the holes in the SPH domain and therefore follows the motion of the hydrofoil, see Section 3.9.4. As a result, the particles are distributed very isotropically in a way that reduces the errors of the SPH operators.

For validation, a simulation is considered where the hydrofoil rotates with a very low rotation speed of $\omega = 0.01$ rad/s about its center. This low rotation speed is chosen because we want to compare the pressure coefficient at two degrees, ten degrees and thirty degrees to reference results of a static simulation with an angle of attack of two, ten and thirty degrees, respectively. A higher rotation speed would modify the relative velocity of the geometry too strongly to allow such a comparison.

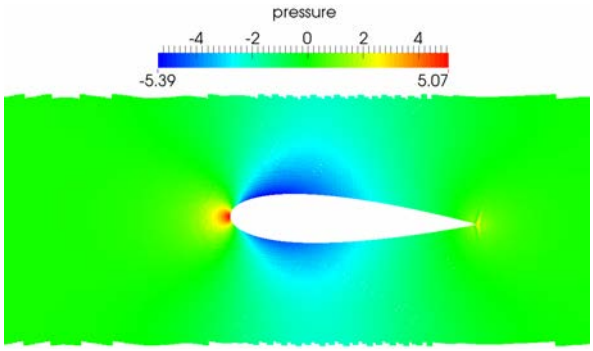
Figure 5.11 shows the particle distribution after a rotation of 2° and 10° where the particles are colored by modified weight over the original weight, i.e. $\frac{\bar{\omega}_i}{\omega_i} \in [0, 1]$. It can be seen that the particles are very isotropically distributed around the hydrofoil without showing any holes. Moreover, it shows the mesh that moves with the NACA and the computed modified weights that detect the movement of the mesh correctly. Figures 5.12, 5.13 and 5.14 show the position of the NACA at three different instances (2°, 10° and 30°) with the corresponding pressure and velocity fields.

For validation, the pressure coefficient on the NACA, computed from the FV domain, is compared to static FV reference solutions and good agreement is found, see Figures 5.15, 5.16 and 5.17.

After comparing of the simulation with the low rotation speed at three instances to static reference solutions with different angle of attack, the same simulation is launched with a larger rotation speed of $\omega = 0.03$ rad/s. Again the particles follow the geometries by the ALE motion from Section 3.9.4. Figures 5.18 and 5.19 show the magnitude of the velocity field and the

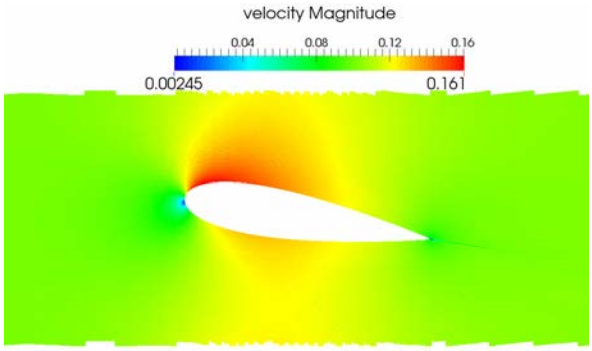


(a) Magnitude of the velocity field plotted on the FV mesh and the SPH particles.

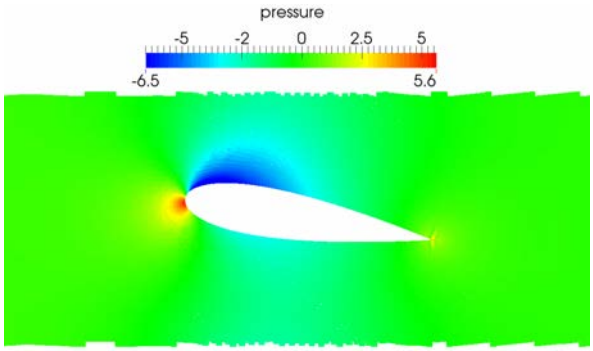


(b) Pressure field plotted on the FV mesh and the SPH particles.

Figure 5.12: NACA hydrofoil at a rotation of 2° where the FV mesh is plotted above the SPH particles.

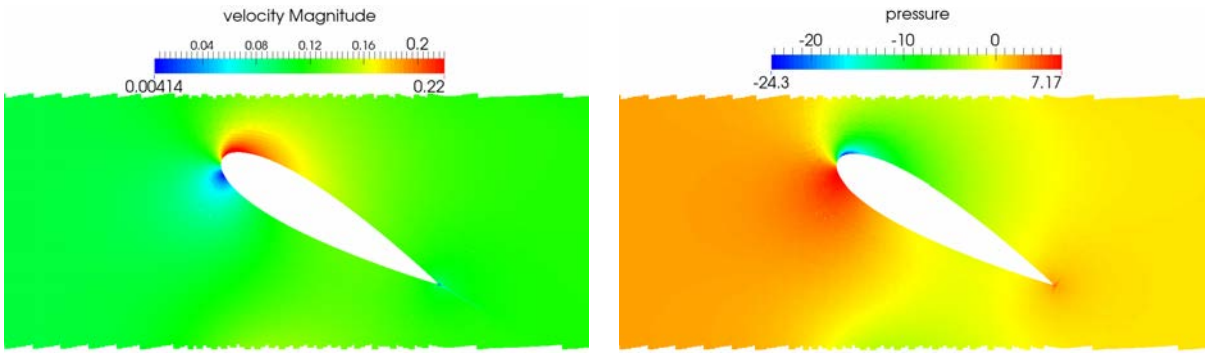


(a) Magnitude of the velocity field plotted on the FV mesh and the SPH particles.



(b) Pressure field plotted on the FV mesh and the SPH particles.

Figure 5.13: NACA hydrofoil at a rotation of 10° where the FV mesh is plotted above the SPH particles.



(a) Magnitude of the velocity field plotted on the FV mesh and the SPH particles.

(b) Pressure field plotted on the FV mesh and the SPH particles.

Figure 5.14: NACA hydrofoil at a rotation of 30° where the FV mesh is plotted above the SPH particles.

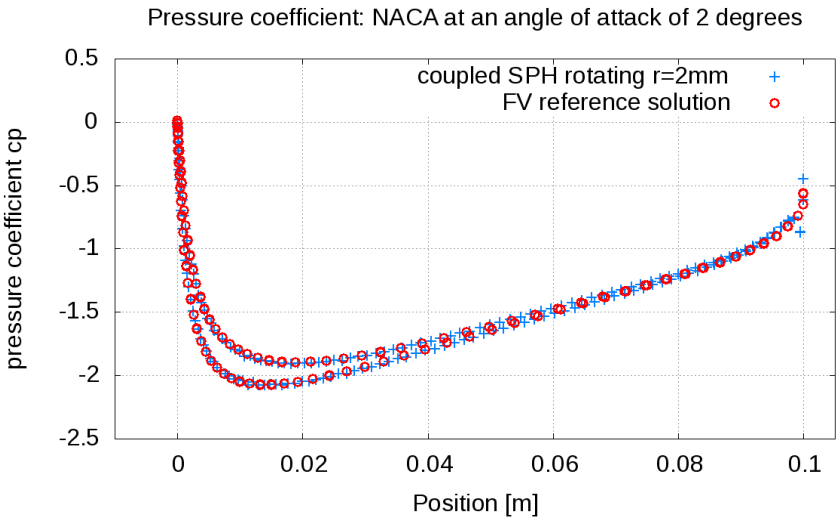


Figure 5.15: Comparison of the pressure coefficient at the NACA rotated by two degrees obtained by a transient simulation with the coupled method and a static FV reference solution.

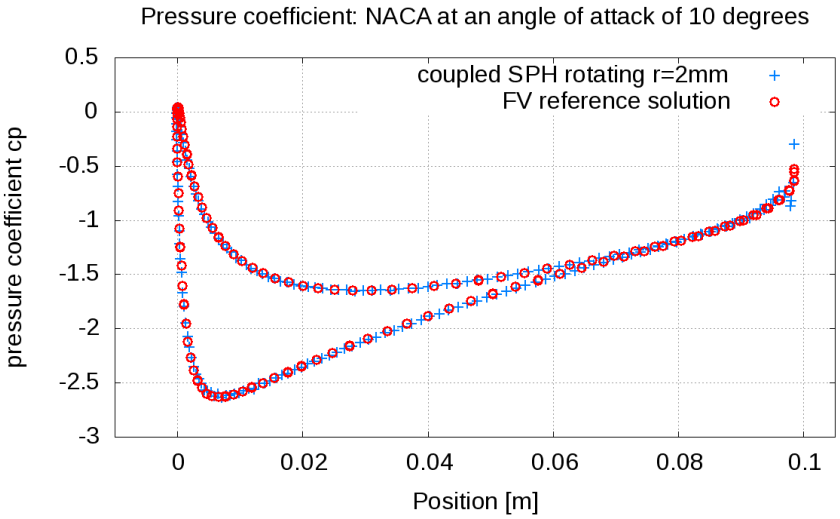


Figure 5.16: Comparison of the pressure coefficient at the NACA rotated by ten degrees obtained by a transient simulation with the coupled method and a static FV reference solution.

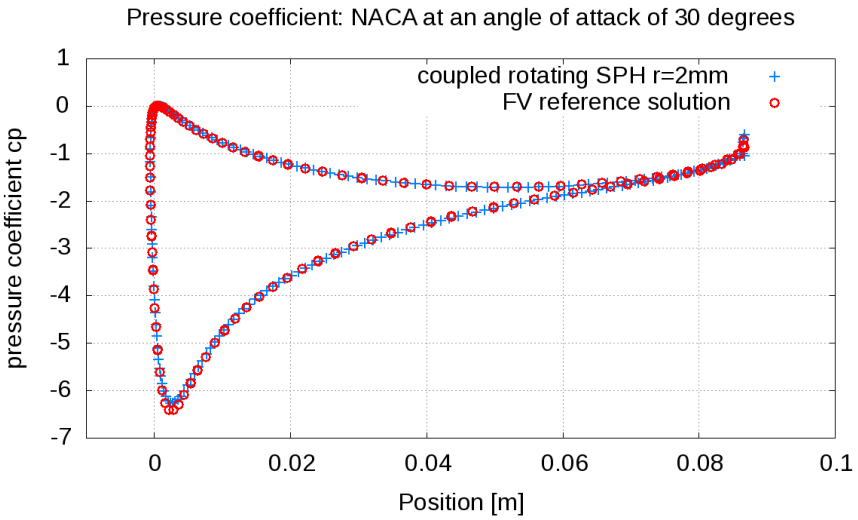


Figure 5.17: Comparison of the pressure coefficient at the NACA rotated by thirty degrees obtained by a transient simulation with the coupled method and a static FV reference solution.

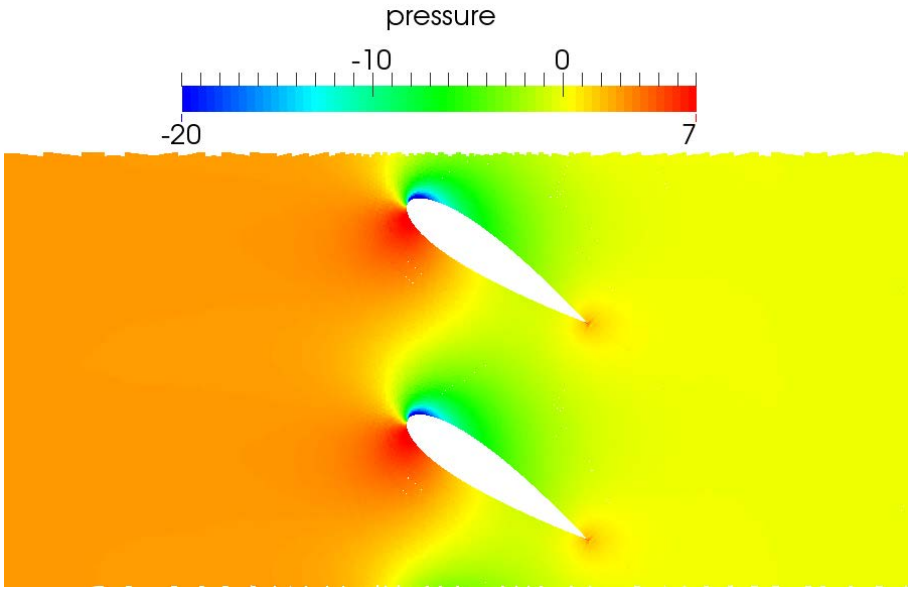


Figure 5.18: Pressure field of the rotating NACA after $t = 11.8s$ plotted on the SPH particles and the FV mesh. A rotation speed of $\omega = 0.03$ rad/s is imposed. This instance corresponds to an angle of attack of 33.4°.

pressure field at $t = 11.8s$ which corresponds to an angle of attack of 33.4°.

5.3 Summary and discussion

It was shown by means of two-dimensional simulations that the coupling algorithm is capable of treating Lagrangian moving particles as well as particles moving arbitrarily. Outlet boundary conditions can be applied to Lagrangian moving particles by adding a block of overlapping FV cells next to the outlet in a way that the boundary condition is imposed by the coupling from FV domain to the SPH domain. This was used to simulate the flow around a static hydrofoil with Lagrangian moving particles.

Next, the test case was readopted to the simulation of rotating hydrofoils with a prescribed rotation about the center of the NACA. The mesh that refines the region close to the solid geometry moves with the solid body. The SPH particles move either in a Lagrangian way or in an ALE way that fills the holes and distributes the particles isotropically. The simulation with a very low rotation speed was compared to FV reference solutions which show good agreement for the pressure coefficient on the NACA hydrofoil. Then, the results of a simulation with a higher rotation speed were shown.

These simple two-dimensional test cases show the potential of the coupled method for its application to real three-dimensional transient simulations like the start-up of a Francis turbine. The meshes can follow a prescribed motion that is for example determined by the moving solid geometry. The particle positions evolve without being disturbed by the motion of the mesh, following the flow (Lagrangian motion), being fixed (Eulerian motion) or moving in an arbitrary way (ALE).

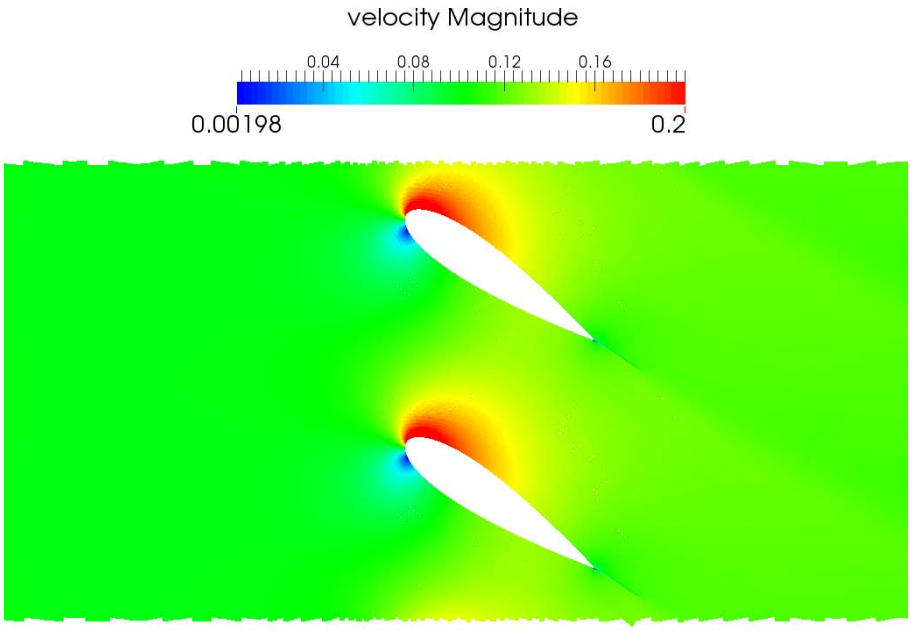


Figure 5.19: Magnitude of the velocity field of the rotating NACA after $t = 11.8s$ plotted on the SPH particles and the FV mesh. A rotation speed of $\omega = 0.03$ rad/s is imposed. This instance corresponds to an angle of attack of 33.4°.

Chapter 6

Conclusions and perspectives

6.1 Conclusions

The objective of this thesis was the development of a flexible coupling methodology that combines the meshless Smoothed Particle Hydrodynamics - Arbitrary Lagrange Euler (SPH-ALE) with the meshbased Finite Volume method (FVM). The coupled approach was designed as a tool for the investigation of transient flows in hydraulic turbines and pumps, like for example the flows that occur during the start-up procedure of a Francis turbine. Traditional numerical methods like FVM have difficulties to correctly simulate these transient flows because of their meshbased character. A mesh necessitates interfaces between rotating and static parts of the computational domain. In some cases it might even require the regeneration of parts of the mesh at every time step.

In contrast, SPH-ALE is a meshless method without any connectivity between particles that can follow the flow in Lagrangian motion, or that can be fixed in space (Eulerian description), or that move in an arbitrary way, e.g. following the motion of solid bodies. However, SPH suffers from several weaknesses, where some have been addressed in this thesis. First, the SPH operators used for the discretization of the Euler equations show a lack of consistency. As a consequence, in many simulations the computed pressure field is very noisy. Second, SPH does not easily allow anisotropic refinement because it is an intrinsically isotropic method. But in real applications different length scales occur. Thus, some zones and some directions require a smaller space discretization size than others. Third, subsonic outlet boundary conditions are difficult to impose on Lagrangian moving particles, even though they are indispensable for internal flow simulations.

In the frame of this thesis, an improvement of the SPH-ALE method was achieved by correcting the gradient of the SPH kernel function in order to increase accuracy. As a result of the correction, the sum over a particle's neighborhood of the modified gradient of the kernel function vanishes identically for each particle. In this way, zeroth order consistency for the SPH-ALE discretization of the divergence of the flux vector is enforced. The correction improves the smoothness of the computed pressure field considerably, especially if particles are disordered.

Then, an unstructured FV solver was implemented into the SPH-ALE software that discretizes the same weakly-compressible Euler equations in ALE formalism than the SPH-ALE method. The obtained semi-discrete FV equations are very similar to the SPH-ALE equations. This allows us to compute the numerical fluxes analogously using the same Riemann solvers with initial states reconstructed by the MUSCL scheme. In addition, the same explicit Runge-Kutta time integration schemes were employed.

The corrected SPH-ALE method was then coupled to FVM by a flexible multi-domain ap-

proach that was inspired by the Chimera method of overlapping grids. The proposed methodology uses regions of SPH particles, regions of FV cells and overlapping regions where the pressure and velocity are transferred. Communication on the overlapping regions is done by interpolating boundary cell values from the SPH domain to the so-called Chimera cells of the FV domain and by using FV cells as neighbours for the SPH space integration. This allows us to use anisotropic meshes that are strongly refined in zones where velocity and pressure fields manifest strong gradients, in particular close to solid wall boundaries.

Contrary to variable resolution SPH methods, the discretization size of the SPH particles does not vary much and the smoothing length stays approximately constant since the refinement is handled by the FV mesh. Consequently, the difficulties linked to variable smoothing length SPH formulations are avoided. It is not necessary to include the gradient of the smoothing length in the computation of the gradient of the kernel function. The refinement ratios of the mesh, which are chosen according to best practice in structured FV meshes, can be quite large from one cell to the other and high aspect ratios are possible.

Compared to other coupled approaches in literature the present coupling strategy has the advantage that no additional geometrical information for the SPH particles has to be computed, i.e. no Voronoi tesselation or auxiliary mesh. The intersection of an SPH particle and a FV cell is computed approximately by adding quadrature points that divide the considered cell into subcells with egdes smaller than a reference length that depends on the particle size. Furthermore, it is assumed that particles are of spherical shape. This approximate computation of the intersection is relatively fast and simple and it is applicable to two- and three-dimensional simulations. Knowing an approximation of the intersection between particles and cells, the physical fields of the completely overlapped particles can be either computed by taking the FV cells as neighbors or by directly interpolating velocity and pressure from the overlapping cells. The latter approach is used for particles close to solid wall boundaries. In both cases, completely overlapped SPH particles do not contribute to the evolution of the flow fields of its' neighbors.

For validation, the Taylor-Green vortex and the flow around a static hydrofoil was presented. It was shown that FV meshes can be added flexibly to the compuational domain without introducing artificial discontinuities in the flow fields. Furthermore, the latter test case also underlined that the coupling is necessary to obtain accurate results with a reasonable number of calculation points.

In the last chapter, the overlapping FV meshes were also used to impose subsonic outlet conditions to moving particles. This enables us to perform internal flow simulations with Lagrangian moving particles. Then, a two-dimensional illustrative example of a rotating hydrofoil was presented. The FV mesh rotates with the solid body and guarantees a fine space discretization close to the wall throughout the whole simulation.

6.2 Perspectives

The developed coupled SPH-FV method possesses the features that are important for the investigation of transient flows in hydraulic machines. Amongst others, it offers the usage of variable anisotropic space resolution and is able to handle arbitrarily moving solid bodies.

However, real flows in hydraulic machines are viscous and turbulent with high Reynolds number. In the future, the coupled method has to be extended to the simulations of viscous fluids. Hence, the discretized viscous terms have to be added to the semi-discrete SPH-ALE and FV equations and a turbulence model is necessary for both solvers. It suggests itself to use the FV solver close to solid wall for the computation of the viscous boundary layer. For that purpose, correct viscous boundary layer treatment has to be implemented into the FV solver.

In literature, Chimera methods were applied to a Euler/Navier-Stokes coupling where boundary layers were computed with a Navier-Stokes solver and the flow in the interior of the domain by an Euler solver. As a first step, this approach could be applied to the coupled method, where the FV solver could be used to compute the Navier-Stokes equations while the SPH solver could be used unchanged.

The coupled method was designed for simulations of transient flows with moving geometries where standard CFD simulations have difficulties to correctly capture the important flow phenomena. It was not developed to compete with standard steady-state FV simulations where FVM are well-established, stable and accurate. Due to its larger numerical stencil SPH is more computationally expensive than FVM. Furthermore, traditionally SPH-ALE is an explicit method in time that demands small time steps for stability reasons, which makes the method even more expensive compared to FVM with implicit time integration. But then, the explicit time integration has the advantage of making it very well adapted for parallelization on multiple CPUs or GPUs that are designed for the purpose of high performance computing. The coupled method uses the same time integration and the same time step size as the SPH-ALE method. Since we use the FV domain for refinement, the cells can be much smaller than the particles, especially close to walls. These small cells decrease the time step size on the entire coupled domain and slow down the whole simulation. Interesting questions for future research are connected to the problem of different time steps and even different time integration schemes for the FV domain and the SPH domain. Once again some inspiration can be found in literature on adaptive mesh refinement and Chimera methods for FVM where different time step sizes on different parts of the computational domain are successfully used.

In the last years, multi-physics simulations gained in importance. The increasing computational power made it possible to couple CFD simulations to structural FEM calculation in order to compute fluid-structure interactions. In the future, the present coupling method should be combined with the already existing coupling of SPH-ALE and FEM to better understand the consequences of transient operation modes on the life cycle of hydraulic machines. Hydraulic machines are rigid and only small deformations occur in general. The FV solver that is already in ALE form could be adapted to slightly deformable meshes where the volumes and other geometrical information of the cells are recomputed regularly.

Appendices

Appendix A

Taylor-Green vortex for incompressible Euler equations

The two-dimensional incompressible Euler equations in non-conservative form are given by

$$\begin{cases} \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} = 0, \\ \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + \dfrac{1}{\rho}\dfrac{\partial p}{\partial x} = 0, \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + \dfrac{1}{\rho}\dfrac{\partial p}{\partial y} = 0, \end{cases} \tag{A.1}$$

where $(u, v)^T$ denotes the velocity vector, p the pressure, ρ the density and x and y the space coordinates. The first equation of System (A.1) is called continuity equation and the second and the third are the momentum equations. The Taylor-Green vortex on the bi-periodic unit square,

$$(x, y) \in [0, 1] \times [0, 1],$$

for the incompressible Euler equations is given by the initial condition

$$\begin{cases} u = \sin(2\pi x)\cos(2\pi y), \\ v = -\cos(2\pi x)\sin(2\pi y), \\ p = \dfrac{\rho}{4}\left(\cos(4\pi x) + \cos(4\pi y)\right). \end{cases} \tag{A.2}$$

This is actually a steady-state solution of the incompressible Euler equations (A.1), which can be seen as follows. The derivatives of u and v are

$$\frac{\partial u}{\partial x} = 2\pi\cos(2\pi x)\cos(2\pi y), \tag{A.3}$$

$$\frac{\partial u}{\partial y} = -2\pi\sin(2\pi x)\sin(2\pi y), \tag{A.4}$$

$$\frac{\partial v}{\partial x} = 2\pi\sin(2\pi x)\sin(2\pi y), \tag{A.5}$$

$$\frac{\partial v}{\partial y} = -2\pi\cos(2\pi x)\cos(2\pi y), \tag{A.6}$$

and the derivatives of p are

$$\frac{\partial p}{\partial x} = -\rho\pi \sin(4\pi x), \tag{A.7}$$

$$\frac{\partial p}{\partial y} = -\rho\pi \sin(4\pi y). \tag{A.8}$$

Inserting Eq. (A.3) and Eq. (A.6) into the incompressible continuity equation yields directly that u and v defined by Eq. (A.2) are solutions of the continuity equation. In order to show that u, v and p are also steady-state solutions of the momentum equations some properties of the sine and cosine function are used. In particular, we can write

$$\frac{\partial p}{\partial x} = -\rho\pi \sin(4\pi x) = -2\rho\pi \ \sin(2\pi x)\cos(2\pi x), \tag{A.9}$$

$$\frac{\partial p}{\partial y} = -\rho\pi \sin(4\pi y) = -2\rho\pi \ \sin(2\pi y)\cos(2\pi y). \tag{A.10}$$

If we insert Eq. (A.3), Eq. (A.4) and Eq. (A.9), we obtain

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} - \frac{1}{\rho}\frac{\partial p}{\partial x} \\
&= -2\pi \ \sin(2\pi x)\cos(2\pi y)\cos(2\pi x)\cos(2\pi y) \\
&\quad - 2\pi \ \cos(2\pi x)\sin(2\pi y)\sin(2\pi x)\sin(2\pi y) \\
&\quad + 2\pi \ \sin(2\pi x)\cos(2\pi x) \\
&= -2\pi \ \sin(2\pi x)\cos(2\pi x)\underbrace{\left[\cos^2(2\pi y) + \sin^2(2\pi y)\right]}_{=1} \\
&\quad + 2\pi \ \sin(2\pi x)\cos(2\pi x) = 0.
\end{aligned}$$

Analogously it can be shown that

$$\frac{\partial v}{\partial t} = -u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} - \frac{1}{\rho}\frac{\partial p}{\partial y} = 0,$$

by inserting Eq. (A.5), Eq. (A.6) and Eq. (A.10).

Summarizing, we have seen that Eq. (A.2) are steady-state solutions of the incompressible Euler equations Eq. (A.1).

Bibliography

[1] I. Abbot and A. von Doenhoff. *Theory of wing sections. Including a summary of Airfoil Data*. Dover Publications, 1959.

[2] United States Energy Information Administration. The international energy outlook 2013. Available at: `www.eia.gov/forecasts/ieo/pdf/0484(2013).pdf`. Online, released on 25.07.2013.

[3] A. Amicarelli, J.-C. Marongiu, F. Leboeuf, J. Leduc, M. Neuhauser, L. Fang, and J. Caro. SPH truncation error in estimating a 3D derivative. *International Journal for Numerical Methods in Engineering*, 87:677–700, 2011.

[4] M. Antuono, A. Colagrossi, S. Marrone, and D. Molteni. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Computer Physics Communications*, 181:532–549, 2010.

[5] D.A. Barcarolo. *Improvement of the precision and the efficiency of the SPH method: Theoretical and numerical study.* PhD thesis, EC Nantes, France, 2013.

[6] D.A. Barcarolo, D. LeTouzé, G. Oger, and F. DeVuyst. Particle refinement and derefinement procedure applied to the Smoothed Particle Hydrodynamics method. In *8th Int. SPHERIC Workshop, Trondheim (Norway)*, 2013.

[7] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg*, 139:3–47, 1996.

[8] J.A. Benek, P.G. Buning, and J.L. Steger. A 3D chimera grid embedding technique. *AIAA Papers*, 85-1523:322–331, 1985.

[9] J.A. Benek, J.L. Steger, and F.C. Dougherty. A flexible grid embedding technique with application to the Euler equations. *AIAA Papers*, 83-1944:373–382, 1983.

[10] C. Benoit. *Méthode d'adaptation de maillages au moyen d'algorithmes génétiques pour le calcul d'écoulements compressibles.* PhD thesis, ENSAM Paris (France), 1999.

[11] M.J. Berger. On conservation at grid interfaces. *SIAM Journal on Numerical Analysis*, 24:967–984, 1987.

[12] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, 1989.

[13] W. Bohl and W. Elmendorf. *Strömungsmachinen 1. Aufbau und Wirkungsweise.* Vogel Buchverlag, 2008.

[14] J. Bonet and T.-S.L. Lok. Variational and momentum preservation aspects of Smooth Particle Hydrodynamics formulations. *Computer Methods in applied mechanics and engineering*, 180:97–115, 1999.

[15] B. Bouscasse, S. Marrone, A. Colagrossi, and A. DiMascio. Multi-purpose interfaces for coupling SPH with other solvers. In *8th Int. SPHERIC Workshop, Trondheim (Norway)*, 2013.

[16] O. Braun. *Part Load Flow in Radial Centrifugal Pumps*. PhD thesis, EPFL, 2009.

[17] L. Castillon, G. Billonnet, S. Péron, and C. Benoit. Numerical simulations of technological effects encountered on turbomachinery configurations with the Chimera technique. In *ICAS*, 2010.

[18] L. Castillon and G. Legras. An unsteady overset grid method for the simulation of compressors with non circumferential casing treatments. In *20th International Symposium of Air Breathing Engines (ISABE11)*, 2011.

[19] A. Colagrossi, B. Bouscasse, M. Antuono, and S. Marrone. Particle packing algorithm for SPH schemes. *Computer Physics Communications*, 183:1641–1653, 2012.

[20] G.H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice*. Cambridge University Press, 2000.

[21] W. Dehnen and H. Aly. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Mon. Not. R. Astron. Soc.*, 000:1–15, 2012.

[22] M. DeLeffe. *Modélisation d'écoulement visqueux par méthode SPH en vue d'application à l'hydrodynamique navale*. PhD thesis, EC Nantes (Frances), 2011.

[23] M. DeLeffe, D. LeTouzé, and B. Alessandrini. SPH modelling of shallow-water coastal flows. *Journal of Hydraulic Research*, 48:118–125, 2010.

[24] J. Donea, A. Huerta, J.Ph. Ponthot, and A. Rodrígues-Ferran. Arbitrary Lagrangian-Eulerian methods. In E. Stein, R. deBorst, and T. Hughes, editors, *Encyclopedia of Computational Mechanics*. John Wiley, 2004.

[25] P. Dörfler, M. Sick, and A. Coutu. *Flow-induced pulsation and vibration in hydroelectric machinery. Engineer's guidebook for planing, design and troubleshooting*. Springer-Verlag, 2013.

[26] F. Dubois. Partial Riemann problem, boundary conditions, and gas dynamics. In *Absorbing Boundaries and Layers, Domain decomposition methods: Applications to large scale computations.*, pages 16–77, New York, 2001. Nora Science Publishers, Inc.

[27] EDF. Code-ASTER. Available at: `www.code-aster.org`. Online.

[28] J. Feldman. *Dynamic refinement and boundary contact forces in Smoothed Particle Hydrodynamics with applications in fluid flow problems*. PhD thesis, University of Wales Swansea, 2006.

[29] J. Feldman and J. Bonet. Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. *International Journal for numerical methods in engineering*, 72:295–324, 2007.

[30] M. Ferrand, D. Laurence, B.D. Rogers, D. Violeau, and C. Kassiotis. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids*, 71:446–472, 2012.

[31] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer-Verlag, 1999.

[32] G. Fourey, G. Oger, D. LeTouzé, and B. Alessandrini. Stratégie de couplage faible entre les méthodes FEM et SPH pour les simulations d'IFS. In *21ème Congrès Francais de Méchanique, Bordeaux (France)*, 2013.

[33] Fraunhofer Institute for Solar Energy Systems ISE. Electricity production from solar and wind in Germany in 2013. `www.ise.fraunhofer.de/en/downloads-englisch/pdf-files-englisch/news`. Online, released on 09.01.2014.

[34] H.-C. Funke, O. Heil, J. Bruhn, and E. Riedel. Options for more flexibility by large scale storage. In *SHF: Pumped storage powerplants, Lyon (France)*, 2011.

[35] A. Gehrer. *Entwicklung eines 3D-Navier-Stokes Codes zur numerischen Berechnung der Turbomaschinenströmung.* PhD thesis, Fakultät für Maschinenbau an der TU Graz, 1998.

[36] J.-M. Ghidaglia and F. Pascal. The normal flux method at the boundary for multidimensional finite volume approximations in CFD. *European Journal of Mechanics BFluid*, 24:1–17, 2005.

[37] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Mon. Not. R. astr. Soc.*, 181:375–389, 1977.

[38] S.K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb. (N.S.)*, 47:271–306, 1959.

[39] P.M. Guilcher. *Contribution au développement d'une méthode SPH pour la simulation numérique des interaction houle-structure.* PhD thesis, EC Nantes, France, 2008.

[40] H. Guillard and C. Farhat. On the significance of the geometric conservation law for flow computations on moving meshes. *Computer Methods in applied mechanics and engineering*, 190:1467–1482, 2000.

[41] H. Guillard and A. Murrone. On the behavior of upwind schemes in the low mach number limit: II. Godunov type schemes. *Computer & Fluids*, 33:655–675, 2004.

[42] H. Hadžić. *Development and Application of a Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructured, Overlapping Grids.* PhD thesis, TU Hamburg (Germany), 2005.

[43] A. Harby, A. Killingtveit, E. Solvang, J. Sauterleute, A. Ruud, and M. Belsnes. Opportunities and challenges for hydro-balancing from norwegian reservoirs. In *SHF: Pumped storage powerplants, Lyon (France)*, 2011.

[44] A. Harten and J. M. Hyman. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50:235–269, 1983.

[45] V. Hasmatuchi. *Hydrodynamics of a Pump-Turbine Operating at Off-Design Conditions in Generating Mode.* PhD thesis, EPFL, 2014.

[46] C. Hirsch. *Numerical Computation of Internal and External Flows. Volume 1. Fundamentals of Numerical Discretization.* John Wiley and Sons Ltd., 1988.

[47] C. Hirsch. *Numerical Computation of Internal and External Flows. Volume 2. Computational Methods for Inviscid and Viscous Flows.* John Wiley and Sons Ltd., 1988.

[48] C. Hirsch. *Numerical Computation of Internal and External Flows. Second Edition. Fundamentals of computational fluid dynamics.* Butterworth-Heinemann, 2009.

[49] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227–253, 1974.

[50] Intergovernmental Panel on Climate Change. IPCC special report. The regional impacts of climate change: An assessment of vulnerability. Summary for policymakers. Available at: `www.ipcc.ch/pdf/special-reports/spm/region-en.pdf`. Online, released in 1997.

[51] Intergovernmental Panel on Climate Change. Special report on renewable energy sources and climate change mitigation. Summary for policymakers. Available at: `www.ipcc.ch/pdf/special-reports/srren/SRREN_FD_SPM_final.pdf`. Online, released in 2012.

[52] G. Jeanfaivre, C. Benoit, and M.C. LePape. Improvement of the robustness of the Chimera method. *AIAA Papers*, 2002-3290:1–11, 2002.

[53] P. Kerschberger and A. Gehrer. Hydraulic development of high specific-speed pump-turbines by means of an inverse design method, numerical flow-simulation (cfd) and model testing. In *25th IAHR symposium on hydraulic machinery and systems*, 2010.

[54] S. Kulasegaram, J. Bonet, R.W. Lewis, and M. Profit. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechnics*, 33:316–325, 2004.

[55] J. Leduc. *Etude physique et numérique de l'écoulement dans un système d'injection de turbine Pelton.* PhD thesis, LMFA, EC Lyon (France), 2010.

[56] J. Leduc, F. Leboeuf, M. Lance, E. Parkinson, and J.-C. Marongiu. Improvement of multiphase model using preconditioned Riemann solvers. In *5th Int. SPHERIC Workshop, Manchester (England)*, 2010.

[57] M. Leonardi and T. Rung. Explicit strategies for consistent kernel approximations. In *9th Int. SPHERIC Workshop, Paris (France)*, 2014.

[58] R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser, 1992.

[59] Zhe Li. *Développement d'une méthode de simulation de couplage fluide-structure à l'aide de la méthode SPH.* PhD thesis, EC Lyon (France), 2013.

[60] Q. Liang and M. Keller. Behavior of pump turbines operating at speed no load condition in turbine mode. In *HydroVision 2010, Charlotte (USA)*, 2010.

[61] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics. A meshfree particle method.* World Science, New Jersey, 2003.

[62] M.B. Liu and G.R. Liu. Restoring particle consistency in smoothed particle hydrodynamics. *Applied numerical mathematics*, 56:19–36, 2006.

[63] Y.R. López, D. Roose, and C.R. Morfa. Dynamic particle refinement in SPH: application to free surface flow and non-cohesive soil simulations. *Comput Mech*, 51:731–741, 2013.

[64] F. Louis. Pumped storage worldwide overview. In *SHF: Pumped storage powerplants, Lyon (France)*, 2011.

[65] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

[66] M. Mancip. *Couplage de méthodes numériques pour les lois de conservation. Application au cas de l'injection.* PhD thesis, INSA Toulouse (France), 2001.

[67] J.-C. Marongiu. *Méthode numérique lagrangienne pour la simulation d'écoulements à surface libre - Application aux turbines Pelton.* PhD thesis, LMFA, Ecole Centrale de Lyon (France), 2007.

[68] J.-C. Marongiu, F. Leboeuf, J. Caro, and E. Parkinson. Low mach number numerical schemes for the SPH-ALE method. Application in free surface flows in Pelton turbines. In *4th Int. SPHERIC Workshop, Nantes (France)*, 2009.

[69] J.-C. Marongiu, F. Leboeuf, and E. Parkinson. Riemann solvers and efficient boundary treatments: an hybrid SPH-finite volume numerical method. In *3rd Int. SPHERIC Workshop, Lausanne (Switzerland)*, 2008.

[70] J.-C. Marongiu, J. Leduc, and M. Schaller. GPU implementation of a SPH-ALE fluid dynamics solver. In *6th Int. SPHERIC Workshop, Hamburg (Germany)*, 2011.

[71] J.-C. Marongiu, E. Parkinson, S. Lais, F. Leboeuf, and J. Leduc. Application of SPH-ALE method to Pelton hydraulic turbines. In *5th Int. SPHERIC Workshop, Manchester (UK)*, 2010.

[72] A. Marsh, G. Oger, D. LeTouzé, and D. Guibert. Validation of a conservative variable-resolution SPH scheme including ∇h terms. In *6th Int. SPHERIC Workshop, Hamburg (Germany)*, 2011.

[73] D. J. Mavriplis and C. R. Nastase. On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes. *Journal of Computational Physics*, 230:4285–4300, 2011.

[74] A. Mayrhofer, M. Ferrand, C. Kassiotis, D.Violeau, and F.-X. Morel. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3D. *Numerical Algorithms*, pages 1–20, February 2014.

[75] J.J. Monaghan. Extrapolating B splines for interpolation. *Journal of Computational Physics*, 60:253–262, 1985.

[76] J.J. Monaghan. Particle methods for hydrodynamics. *Computer Physics Reports*, 3:71–124, 1985.

[77] J.J. Monaghan. On the problem of penetration in particle methods. *Journal of Computational Physics*, 82:1–15, 1989.

[78] J.J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399–406, 1994.

[79] E. Mosonyi. *Low-head power plants*. Akademiai Kiado, Budapest, 1987.

[80] C. Nicolet. *Hydroacoustic modelling and numerical simulation of unsteady operation of hydroelectric systems*. PhD thesis, EPFL, 2007.

[81] J. Nicolle, J. F. Morisette, and A. M. Giroux. Transient CFD simulation of a francis turbine startup. In *26th IAHR symposium on hydraulic machinery and systems*, 2012.

[82] F. Nicoud. Unsteady flows modeling and computation. *VKI Lecture Notes*, pages 1–65, 2007.

[83] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant. Two-dimensional SPH simulations of wedge water entries. *Journal of Computational Physics*, 213:803–822, 2006.

[84] P. Omidvar. *Wave loading on bodies in the free surface using Smoothed Particle Hydrodynamics (SPH)*. PhD thesis, University of Manchester, 2010.

[85] O. Pacot. *Large Scale Computation of the Rotating Stall in a Pump-Turbine using an Overset Finite Element Large Eddy Simulation Numerical Code*. PhD thesis, EPFL, 2014.

[86] E. Parkinson, A. Karakolcu, N. Gervais, M. Rentschler, and K. Winkler. Recent development in the hydraulics of Pelton turbines. In *HYDROVISION India*, 2012.

[87] E. Pärt-Enander. *Overlapping Grids and Applications in Gas Dynamics*. PhD thesis, Uppsala University, Department of Scientific Computing (Sweden), 1985.

[88] E. Pärt-Enander and B. Sjögreen. Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems. *Computer & Fluids*, 23:551–574, 1994.

[89] P. Pieringer. *Methoden zur instationären Strömungsrechnung in Thermischen Turbomaschinen.* PhD thesis, Fakultät für Maschinenbau und Wirtschaftswissenschaften. TU Graz, 2008.

[90] T.J. Poinsot and S.K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics*, 101:104–129, 1992.

[91] D. J. Price. Smoothed particle hydrodynamcs and magnetohydrodynamics. *Journal of Computational Physics*, 231:759–794, 2012.

[92] P.W. Randles and L.D. Libersky. Smoothed particle hydrodynamics: Some recent improvements and applications. *Comput. Methods Appl. Mech. Engrg.*, 139:375–408, 1996.

[93] P.A. Raviart. An analysis of particle methods. In *Lecture notes in mathematics*. Springer, 1985.

[94] J.I. Read, T. Hayfield, and O. Agertz. Resolving mixing in smoothed particle hydrodynamics. *Mon. Not. R. Astron. Soc.*, 405:1513–1530, 2010.

[95] J.-C. Riesterer, M. Sick, M. Sallaberger, R. Schürhuber, and P. Nowicki. Wasserkraftanlagen - flexibel und dynamisch. *VGB PowerTech Journal*, 9:45–53, 2013.

[96] D. Robb, S. Gaskin, J.-C. Marongiu, and M. Villeneuve. Smoothed particle hydrodynamics simulations of freely moving solid objects in a free-surface flow with applications to river ice dynamics. In *21st Canadian Hydrotechnical Conference, Banff (Alberta)*, 2013.

[97] S. Roth. *Fluid-structure coupling effects on the dynamic response of pump-turbine guide vanes*. PhD thesis, EPFL, 2012.

[98] C. Schick. Adaptivity for Particle Methods in Fluid Dynamics. Master's thesis, University of Kaiserslautern (Germany), 2000.

[99] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings ACM National Conference*, 23:517–524, 1968.

[100] M. Sick, C. Oram, O. Braun, B. Nennemann, and A. Coutu. Hydro projects delivering regulating power: Technical challenges and cost of operation. In *Hydro 2013, Innsbruck (Austria)*, 2013.

[101] Societé hydraulique de France. *Pumped storage powerplants: Challenges and opportunities, Lyon (France)*, 2011.

[102] T. Staubli, F. Senn, and M. Sallaberger. Instability of pump-turbines during start-up in turbine mode. In *Hydro 2008, Ljubljana (Slovenia)*, 2008.

[103] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction*. Springer-Verlag, 1999.

[104] John A. Trangenstein. *Numerical Solution of Hyperbolic Partial Differential Equations*. Cambridge University Press, 2009.

[105] C. Ulrich. *Smoothed-Particle-Hydrodynamics Simulation of Port Hydrodynamic Problems*. PhD thesis, TUHH Germany, 2013.

[106] C. Ulrich, N. Koliha, and T. Rung. SPH modelling of water/soil-flows using a variable-resolution scheme. In *6th Int. SPHERIC Workshop, Hamburg (Germany)*, 2011.

[107] C. Ulrich, M. Leonardi, and T. Rung. Multi-physics SPH simulation of complex marine-engineering hydrodynamic problems. *Ocean Engineering*, 64:109–121, 2013.

[108] R. Vacondio, B.D. Rogers, P.K. Stansby, and P. Mignosa. Shallow water SPH for flooding with dynamic particle coalescing and splitting. *Advances in Water Resources*, 58:10–23, 2013.

[109] R. Vacondio, B.D. Rogers, P.K. Stansby, and P. Mignosa. 3-D SPH scheme with variable resolution: assessment of the optimal splitting refinement pattern. In *9th Int. SPHERIC Workshop, Paris (France)*, 2014.

[110] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

[111] J.P. Vila. On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences*, 9:161–209, 1999.

[112] D. Violeau. *Fluid Mechanics and the SPH Method. Theory and Applications.* Oxford University Press, 2012.

[113] Z.J. Wang. A fully conservative structuredunstructured Chimera grid scheme. *AIAA Papers*, 95-0671:1–10, 1995.

[114] Z.J. Wang and H.Q. Yang. A unified conservative zonal interface treatment for arbitrarily patched and overlapped meshes. *AIAA Papers*, 94-0320:1–10, 1994.

[115] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.

[116] C. Widmer, T. Staubli, and N. Ledergerber. Unstable characteristics and rotating stall in turbine brake operation of pump-turbines. *Journal of Fluids Engineering*, pages 1–9, 2011.

[117] C. Widmer, T. Staubli, T. Tresch, and M. Sallaberger. Unstable pump-turbine characteristics and their interaction with hydraulic systems. In *HydroVision 2010, Charlotte (USA)*, 2010.