Technische Universität München
TUM School of Engineering and Design

A Modular Parallel-Computing Strategy for Multiresolution Simulations of Complex Flows on High-Performance Computers

Nils Hoppe

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Michael W. Gee

Prüfer*innen der Dissertation:

1. Prof. Dr.-Ing. Nikolaus A. Adams
2. Prof. Dr. Petros Koumoutsakos
3. Prof. Dr. rer. nat. Michael Georg Bader

Die Dissertation wurde am 15.09.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 27.02.2023 angenommen.

Sandra and Konrad: I love you to pieces, to you I dedicate this thesis.

Typesetting LaTeX

München, September 15, 2022 *Nils Hoppe*

Acknowledgement

Many people have contributed to this thesis, some directly with advise on its subject, others indirectly through continuous moral support. Some have helped knowingly, others may not even know me. No matter the way you influenced this thesis for the better, my deepest gratitude to you!

In particular, I want to thank my supervisor Prof. Nikolaus A. Adams for the opportunity to work in a stimulating environment with large thematic freedom. Thank you also for the support and guidance throughout the years and last but not least for the examination of this work.

I further want to express my gratitude towards Prof. Petros Koumoutsakos for taking the time and effort to evaluate this thesis. The same applies to Prof. Michael Bader, whom I also thank for his challenging but inspiring lectures and the good times at the Bavarian Graduate School of Computational Engineering. Similarly, I am thankful to Prof. Gee whose lectures sparked the interest for numerical mechanics in my first undergraduate semesters and who volunteered as chairmen of the examination board of this thesis.

My thanks expand to Dr. Stefan Adami an extraordinary supervisor and reviewer, who spent hours and days elevating my notes into reports, my slides into presentations and my drafts into papers. Thank you for keeping the Nanoshock group on track and for the memorable times at the workshops.

I thank Prof. Bungartz for his tremendous support during my master study program, which enabled and encouraged me to enroll in a doctoral program to begin with. Your taking on the mentorship for my dissertation project was much appreciated, and doing so even after I rejected your offer to work at your chair speaks for itself.

Thanks is due to Prof. Christian Stemmer not only for the countless technical assistance provided, but for the constructive work and the high levels of trust you put in me during the NFDI4Ing project.

I want to thank Angela Grygier and Hua Liu for their help with the bureaucracy. Even though I never manged to fill any form correctly on the first try, you never lost your patience nor your good attitude towards me.

I thank Josef for the fun and effort in our joint publication, his high work and coding ethics and his friendship outside the office. I appreciate that the words “No time” and “I cannot help” do not exist in your vocabulary. Further, I thank Vladimir for the many joyful hours in our office room. It was a pleasure learning about Russian and German language and culture together. Thanks go out to Nico for his physics and numerics explanations for dummies without which I’d still be lost in the forest of papers.

Although your mountaineering is on a different level than mine, you made sure everyone enjoyed the joint hikes. I thank you, Jakob, for sharing your TUM intel with us. For your sarcasm, cynicism and the vodka: thank you, Alexandr. Naeimeh, thanks for keeping up the good spirit despite being all alone - culture, gender and topic wise. I thank Thomas for his commitment to the shared code base and for keeping a smile on his face and a calm hand even in rough times. For keeping the ALPACA legacy alive and rejuvenating it, I thank you, Alexander.

My deep gratitude to Benjamin without you I would have constantly drowned in NFDI4Ing organizational tasks. Thanks also to Vasiliki for your good mood and for the constant plant-watering efforts. Further, I would like to thank Steffen and Marcus for intense physics debates after hours and the peeks behind the scenes of university politics. Christopher, Christian, Deniz, Rim, Fabian, Aaron, Ludger, and Vladislav, thanks for the great lunch times and the long nights at the Italian restaurant. Thanks to Patrick, Stefan, Aleksandra, Felix, Andrei and all the other colleagues who made the time in-between work more enjoyable.

Outside the chair I want to thank my family. Without the continuous support of my parents and my brother I would not have made it to the finish line. I want to thank my in-laws not only, but in particular for the countless hours of baby-sitting allowing me working on this very manuscript. Thank you Martin and Haosi for continuous friendship throughout the years. And thanks to Nicolas, Felix and Elena: it is a privilege calling you my friends.

And finally, I want to thank Scott Adams, Charles M. Schulz, Bill Watterson, Thomas 'Tom' Körner, and Jorge Cham for keeping me entertained throughout the years.

Abstract

Today, resolving all relevant scales and phenomena of practical compressible multiphase flow applications is infeasible for direct numerical simulation (DNS) without subgrid modelling. This is due to the complex flow features such as shock waves, the inherent three-dimensionality of the problem, and the need to resolve a wide range of physical scales. These hindrances stem from a lack of computational power as well as the complexity of the required numerical methods. Nevertheless, a decent range of scales are already representable if compression algorithms and high-performance computing (HPC) systems are utilized. Growing computational power translates hence directly to more powerful DNS as long as the numerical methods and algorithms use (all the features of) the provided hardware efficiently. This, however, is rarely the case.

This thesis presents concepts and implementations for the parallelization of a modular DNS finite-volume method (FVM) compressible multiphase flow solver. The parallelization strategy takes into account the requirements of multiresolution (MR) for spatial and adaptive local-time stepping (LTS) for temporal compression. Additionally, techniques to efficiently incorporate the level-set (LS) algorithm for the representation of multiple phases as well as free surfaces or complex solid geometries are given. The described concepts allow a modular implementation capable of maintaining the parallel performance across numerical schemes of different nominal convergence order. In turn, this modularity eases integration of new schemes speeding up method development and eases benchmarking new methods on large-scale three-dimensional (3D) problems. The performance portability is demonstrated on homogenous supercomputing hardware, with special focus on improved auto-vectorization. Besides thorough verification and validation, various large-scale single- and two-phase simulations of 3D flows demonstrate the capabilities of the solver.

Zusammenfassung

Mit der heutigen Rechenkapazität ist eine vollständige Abbildung praktischer Anwendungen kompressibler Mehrphasenströmungen mittels direkter numerischer Simulation (DNS) unmöglich. Dies liegt insbesondere an komplexen Strukturen wie etwa Verdichtungsstößen, der dreidimensionalen (3D) Natur dieser Strömungen und des großen Bereichs relevanter physikalischer Skalen. Bereits heute kann jedoch schon ein weiterer Skalenbereich abgedeckt und somit wichtige physikalische Effekte in den Anwendungen untersucht werden, sofern Komprimierungsalgorithmen und Hochstleistungsrechner (HPC) verwendet werden. Die Zunahme an Rechenkapazität lässt die Möglichkeiten von DNS direkt anwachsen, falls die verwendeten numerischen Methoden und Algorithmen die spezielle Hardware effizient verwenden können. Dies ist allerdings selten der Fall.

Diese Arbeit legt Konzepte für die Parallelisierung eines DNS Finite-Volumen Mehrskalenlösers für kompressible Strömungen dar. Die vorgestellte Parallelisierung berücksichtigt die speziellen Anforderungen räumlicher und zeitlicher Komprimierung. Ferner erlaubt sie die effiziente Einbindung des Niveaumengenalgorithmus (“level set”) zur Darstellung mehrerer Fluide, fester komplexer Geometrien oder freier Oberflächen. Die theoretischen Erkenntnisse wurden in einem modularen Framework implementiert. Dieses erreicht gleichbleibende Performanz bei parallelisierter Berechnung über verschiedene numerische Verfahren mit unterschiedlichen Konvergenzordnungen hinweg. Dies wird durch Rechnungen auf homogenen Hochleistungsrechnern demonstriert. Die Modularität des Framewokrs vereinfacht das Einbauen neuer Methoden und verkürzt damit die Entwicklungszeit. Ferner ist die direkte Anwendung neuer Methoden auf hochaufgelösten, mehrdimensionalen Gittern möglich. Ein spezieller Fokus liegt dabei auf der effizienten Auslastung der Vektorregister. Neben ausführlichen Validierungsrechnungen zeigen eine Vielzahl anspruchsvoller hochaufgelöster Ein- und Zweiphasensimulationen die Vorzüge der Parallelsierungsstrategie und ihrer Implementierung auf.

Contents

List of Figures

Acronyms

3D three-dimensional

AGU address generation unit

ALPACA Adaptive Level-set PArallel Code ALPACA

ALTS adaptive LTS

ALU arithmetic logic unit

AMR adaptive mesh refinement

API application programming interface

CFD computational fluid dynamics

CFL Courant-Friedrichs-Lewy

CPU central processing unit

CRTP curiously recurring template pattern

DGM Discontinuous Galerkin method

DNS direct numerical simulation

DOF degree of freedom

ENO essentially non-oscillatory

EOS equation of state

FDM finite-difference method

FEM finite-element method

FLAC Free Lossless Audio Codec

FLOP floating-point operations

FPU floating-point unit

FVM finite-volume method

GFM ghost-fluid method

GPU graphics processing unit

HLLC Harten-Lax-van Leer with restored contact surface

HPC high-performance computing

I/O input/output

ID identifier

JPEG Joint Photographic Experts Group

LAN local area network

LRZ Leibniz Supercomputing Centre

LS level-set

LTS local-time stepping

MP3 Moving Picture Experts Group Audio Layer III

MPI Message Passing Interface

MR multiresolution

MUSCL Monotonic Upstream-centered Scheme for Conservation Laws

NSE Navier-Stokes equations

NUMA non-uniform memory access

OS operating system

PC personal computer

PNG Portable Network Graphics

RAM random-access memory

RANS Reynolds-averaged Navier-Stokes

RK Runge-Kutta

RTI Rayleigh-Taylor instability

SFC space-filling curve

SIMD single instruction multiple data

SMT simultaneous multithreading

TENO targeted ENO

TPU tensor processing unit

TVD total variation diminishing

VOF volume of fluid

WENO weighted ENO

Nomenclature

Thermo-Fluid Dynamics

ρ	Density	$\frac{\mathrm{kg}}{\mathrm{m}^3}$
$\mathbf{v}$	Velocity vector	$\frac{\mathrm{m}}{\mathrm{s}}$
p	Pressure	$\frac{\mathrm{kg}}{\mathrm{m}\,\mathrm{s}^2}$
e	Specific internal energy	$\frac{\mathrm{m}^2}{\mathrm{s}^2}$
E	Total specific energy	$\frac{\mathrm{m}^2}{\mathrm{s}^2}$
Π	Stress tensor	$\frac{\mathrm{kg}}{\mathrm{m}\,\mathrm{s}^2}$
Υ	Viscous stress tensor	$\frac{\mathrm{kg}}{\mathrm{m}\,\mathrm{s}^2}$
q	Heat flux density	$\frac{\mathrm{kg}}{\mathrm{s}^3}$
g	Gravitational pull	$\frac{\mathrm{m}}{\mathrm{s}^2}$
T	Temperature	K
k	Thermal conductivity	$\frac{\mathrm{kg}\,\mathrm{m}}{\mathrm{s}^3\,\mathrm{K}}$
μ	Dynamic viscosity	$\frac{\mathrm{kg}}{\mathrm{m}\,\mathrm{s}}$
$\mathbf{U}$	State vector	$\left[\frac{\mathrm{kg}}{\mathrm{m}^3}, \frac{\mathrm{kg}}{\mathrm{m}^2\,\mathrm{s}}, \frac{\mathrm{kg}}{\mathrm{m}\,\mathrm{s}^2}\right]^T$
$\mathbf{f}$	Volume-force vector	$\left[\frac{\mathrm{kg}}{\mathrm{m}^3}, \frac{\mathrm{kg}}{\mathrm{m}^2\,\mathrm{s}}, \frac{\mathrm{kg}}{\mathrm{m}\,\mathrm{s}^2}\right]^T$
γ	Heat capacity ratio	—
σ	Surface tension coefficient	$\frac{\mathrm{kg}}{\mathrm{s}^2}$
a	(Local) speed of sound	$\frac{\mathrm{m}}{\mathrm{s}}$

Numerics

F	Flux	$\left[\frac{\mathrm{kg}}{\mathrm{m}^5}, \frac{\mathrm{kg}}{\mathrm{m}^4\,\mathrm{s}}, \frac{\mathrm{kg}\,\mathrm{m}^3}{\mathrm{s}^2}\right]^T$
$\mathbf{R}$	Roe eigenvector	—
Λ	Roe eigenvalue matrix	—

$\mathbb{S}$	Signal speed estimate	$\frac{\mathrm{m}}{\mathrm{s}}$
β	Velocity component normal to Riemann problem	$\frac{\mathrm{m}}{\mathrm{s}}$
ζ	Permutated velocity components of HLLC-input state	$\frac{\mathrm{m}}{\mathrm{s}}$
Θ	Stencil	—
w, a, b, d	WENO weights	—
ϵ	Small positive quantity	—

Time

t	Time	s
Δt	Time step	s
C	CFL constant	—
τ	Pseudo time	—

Geometric Quantities

V	Volume	m^3
A	Aperture	m
α	Volume fraction	m^3
$\mathbf{n}$	Normal vector	m
Δx	Cell size	m

Multiphase

ϕ	Level set	—
ξ	Fluid identifier	—
Γ	Ideal-gas variable for diffusive interface model	—
Ω	Background-pressure variable for diffusive interface model	$\frac{\mathrm{kg}}{\mathrm{m\,s}^2}$

Multiresolution

l	Level of refinement	—
m	Level index	—
c	Scaling coefficient	—
θ	Scaling Function	—

d	Detail	–
Ψ	Wavelet function	–
$\mathcal{A}$	Averaging operator	–
$\mathcal{P}$	Prediction operator	–
Q	Q-terms of the prediction operator	–
ω	Prediction coefficients	–

Mathematical Symbols

$\otimes$	Tensor product	–
$\cdot$	Dot product	–
∇	Gradient	–
∂	Partial derivative	–

Miscellaneous

$\mathbf{I}$	Identity matrix	–
i, j, k	Cell indices	–
p, q, r, s	Arbitrary indices	–
N	Arbitrary natural number	–
$\mathbb{N}$	Set of all natural numbers	–
$\mathbb{Z}$	Set of all integer numbers	–

Chapter 1

Introduction

Compressible multiphase flows govern a wide variety of engineering applications. Cavitation bubbles damage, e. g., ship propellers [34] by emitting shockwaves when collapsing. Bubble dynamics are also relevant within nozzles such as fuel injectors or for flows around airfoils, submarines and alike [23, 169, 172, 173, 194, 195]. Breakup and atomization of liquid droplets is relevant for combustion engines [49, 151, 152, 196, 197, 237, 241], supersonic aircraft design [54, 111, 122, 217], and nuclear safety [36, 37]. Nanoparticles can be generated using plasma-based laser ablation [116, 153], but also biomedical procedures rely on compressible flow phenomena including shockwaves, phase-separation and phase change. Examples include sonoporation [164], lithotripsy [133, 147, 150, 176], histotripsy [131], artificial heart valves and pumps [30], blast trauma care [55, 135, 178], or manufacturing [235] and delivery [210, 233] of medical drugs. Further applications range from boiling flows [38, 138, 141] and liquid jets [132, 184, 215, 216, 244] to blast effects on structures [57, 75, 91, 92, 159, 160].

Despite successful practical application of complex flows in the aforementioned areas, not all underlying physical processes are understood. Hence, to make these methods effective or at least more efficient a better understanding is required. Experimental investigations have revealed many details of compressible multiphase flows: From singular-bubble collapse and rebound behavior [181] and jetting-behavior of bubbles [21, 32, 87, 143, 163, 193] to fragmentation and atomization of droplets [102, 235] to name, but a few. Yet, they cannot produce complete field solutions and are often limited by measurement equipment, e. g. exposure times of cameras or measurements probes altering the observed flow field. The more complex the investigated flow becomes the more cumbersome, expensive and potentially dangerous experiments become [228, 236].

Numerical investigations do not suffer from these shortcomings, but bring their own challenges. For the simulation of multiphase flows, one such challenge lies in correctly representing the fluid interface(s). While the Navier-Stokes equations (NSE) accurately describe the flow of a single phase, the interplay between two or more immiscible fluids is not described by them. Hence, one needs to introduce additional models for the fluid interface and the transport across it. Such interface models usually fall into the categories of interface capturing with smeared interfaces or interface tracking schemes with a sharp interface representation. The volume of fluid (VOF) methods [101, 161] or the diffuse interface method [1, 4] are prominent examples for the first and the free-Lagrange [11, 227], the front-tracking [78, 79, 225, 229], or the level-set (LS) algorithm [47, 167, 213] for the latter.

Numerical discretization schemes for (compressible) flows should ideally provide vanishing smearing of flow states, prevent oscillations near discontinuities, maintain conservation and ideally provide high-order convergence. The (cell-averaged) finite-volume method (FVM) Godunov's approach [80] achieves these goals and is hence typically employed. As for flux limiting procedures [142, 221], it ensures discrete conservation. Similar to the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) approach [230], it allows combinations of approximate Riemann solvers [64, 97, 186] but enables application of

higher-order methods than MUSCL. To reach theses higher-orders, weighted ENO (WENO) type reconstruction stencils [118] are typically chosen [16, 119, 170, 219] as they allow straightforward construction of stencils with arbitrary order while introducing limited numerical diffusion. However, their computational cost is high and for certain problem types commonly selected combinations of Riemann solvers and high-order stencils may fail [65, 66, 70]. Hence, support for a wide range of numerical schemes is beneficial in a simulation framework for compressible flow research. It shall be noted, that also finite-difference method (FDM) [146] and Discontinuous Galerkin methods (DGMs) are applied to compressible flows [40]. For these methods, however, it is more challenging to meet all the mentioned desired properties [62, 158, 174, 182]. Hence, they are not further discussed within this thesis.

For Godunov-type schemes, in addition to the numerically expensive discretization, the wide range of scales poses high computational demands. Physically, all scales from the domain size to the Kolmogorov scale and in multiphase flows additonally the scale of the smallest interface waves need to be resolved or modelled appropriately. Although subgrid-modelling methods [73, 208, 223] are commonly used the focus here lies on direct numerical simulation (DNS), i. e. the aim to ultimately resolve all relevant scales. While this eliminates modelling errors it makes the application of compression algorithms and the usage of high-performance computing (HPC) systems inevitable. A variety of compression algorithms exists. While the most prominent ones stem from media compression such as the lossy Joint Photographic Experts Group (JPEG) [112] [112] or Moving Picture Experts Group Audio Layer III (MP3) [113] and the lossless Portable Network Graphics (PNG) [114] or Free Lossless Audio Codec (FLAC) [68] for image and audio compression, respectively. For the application of compression algorithms to complex flows, however, the aforementioned properties of the numerical methods must stay intact. Additionally, the compression algorithms should compress both number of floating-point operations (FLOP) and the memory footprint of the simulations. While multigrid [60], sparse grids [204], adaptive DGM and wavelet-collocation [231] have been applied to complex flows [5, 98, 232, 245], the commonly applied techniques within the simulations of physical application are the gradient-based adaptive mesh refinement (AMR) [17] and the wavelet-based multiresolution (MR) [95] algorithms [48, 90, 106, 162, 198, 207, 209, 239]. Utilization of such mesh compression, however, introduces additional algorithmic complexity and may affect the performance of the computation on HPC hardware. In order to efficiently use such hardware, all its layers of parallelism from distributed-memory to single instruction multiple data (SIMD) parallelism need to be addressed [89, 236].

This thesis, hence, addresses the aforementioned challenges for numerical simulation of compressible multiphase flows. To improve the parallel performance of MR compression on modern HPC systems a block-based variant is proposed. The scheme is the algorithmic backbone of the open-source C++ framework Adaptive Level-set PArallel Code ALPACA (ALPACA)[3]. For the algorithm design a modular built-up is proposed to allow exchanging numerical schemes and parameters, material models, equation of states (EOSs), and partitioning methods. The modular design even allows exchanging of the underlying equations at compile time, yielding straightforward extension to different governing equations. Beyond that, the narrowband algorithm [2, 39, 171] for LS simulations is modified to naturally fit into the block-based MR scheme allowing for large scale three-dimensional (3D) simulations using the sharp-interface method. The implementation focuses on modern programming patterns [71, 81, 149, 155] and the use of C++20 features to enable the algorithmic flexibility. For increased usability clean-coding techniques and high-level source code optimizations are employed. Therein, the flexibility and maintainability of the code base is also taken into account. The aim is to obtain steady parallel performance across the mod-

ules while providing a compute environment that fosters rapid large-scale prototyping of new numerical methods and novel physical setups alike allowing a more agile approach to computational fluid dynamics (CFD).

The remainder of this publication-based thesis is structured as follows. In chapter 2 the governing equations and their numerical discretization are stated. Details on mesh-compression algorithms in general and MR in particular are given. Furthermore, the parallelization concepts employed in HPC hardware are stated and their implications for software implementations are discussed. A discussion of competing compute frameworks for complex flows is given. Chapter 3 lists the main findings of this thesis and chapter 4 links the respective publications. A conclusion including a review on relevant literature is drawn in chapter 5.

Chapter 2

Fundamentals

2.1 Governing Equations

Under continuum assumptions the compressible NSE

$$\frac{\partial \rho}{\partial t} = -\operatorname{div} \rho \mathbf{v}, \tag{2.1}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} = -\operatorname{div} (\rho \mathbf{v} \otimes \mathbf{v} + \Pi) \quad + \rho \mathbf{g}, \tag{2.2}$$

$$\frac{\partial \rho E}{\partial t} = -\operatorname{div} (\rho E \mathbf{v} + \Pi \mathbf{v} - \mathbf{q}) + \rho \mathbf{g} \cdot \mathbf{v}, \tag{2.3}$$

describe the flow of a fluid in 3D space $\mathbf{x} = \left[x^1, x^2, x^3\right]^T$ and time t by stating the conservation of mass, momentum and energy, respectively. In the above equations, density $\rho(\mathbf{x}, t)$, the velocity vector $\mathbf{v}(\mathbf{x}, t)$, the gravitational acceleration vector $\mathbf{g}(\mathbf{x}, t)$, and the total specific energy $E = e + \mathbf{v}^2/2$ which is the sum of the specific internal energy $e(\mathbf{x}, t)$ and the specific kinetic energy are used. Note, throughout this thesis the spatial $\mathbf{x}$ and temporal t variables are skipped unless they are crucial for the understanding of the described phenomenon. Heat fluxes are considered by

$$\mathbf{q} = -k \nabla T \tag{2.4}$$

with the thermal conductivity $k(\mathbf{x}, t)$ and temperature $T(\mathbf{x}, t)$. The stress tensor for Newtonian fluids is given by

$$\Pi = -p\mathbf{I} + \Upsilon = -p\mathbf{I} + \mu \left(\nabla \otimes \mathbf{v} + (\nabla \otimes \mathbf{v})^T - \frac{2}{3} \operatorname{div}(\mathbf{v}) \mathbf{I} \right) \tag{2.5}$$

with pressure $p(\mathbf{x}, t)$, dynamic viscosity μ, identity matrix $\mathbf{I}$ and viscous stress tensor $\Upsilon(\mathbf{x}, t)$. The integral formulation of eqs. (2.1) to (2.3) reads

$$\int_V \frac{\partial \rho}{\partial t} \, dV = -\oint_{\partial V} (\rho \mathbf{v}) \cdot \mathbf{n} \, d\partial V \tag{2.6}$$

$$\int_V \frac{\partial \rho \mathbf{v}}{\partial t} \, dV = -\oint_{\partial V} (\rho \mathbf{v} \otimes \mathbf{v} + p\mathbf{I} + \Upsilon) \cdot \mathbf{n} \, d\partial V \quad + \int_V \rho \mathbf{g} \, dV \tag{2.7}$$

$$\int_V \frac{\partial \rho E}{\partial t} \, dV = -\oint_{\partial V} (\rho E \mathbf{v} + p\mathbf{I}\mathbf{v} + \Upsilon \mathbf{v} + \mathbf{q}) \cdot \mathbf{n} \, d\partial V + \int_V \rho \mathbf{g} \cdot \mathbf{v} \, dV, \tag{2.8}$$

wherein V denotes a material volume. By rearranging and summarizing terms in this set of equations the flux-based formulation gives

$$\int_V \frac{\partial \mathbf{U}}{\partial t} \, dV = -\oint_{\partial V} \left(\mathbf{F}^{\mathrm{c}} + \mathbf{F}^{\mu} + \mathbf{F}^{\mathrm{q}} \right) d\partial V + \int_V \mathbf{f} \, dV \tag{2.9}$$

with the state vector $\mathbf{U} = \begin{bmatrix} \rho & \rho\mathbf{v} & \rho E \end{bmatrix}^T$, the convective contribution

$$\mathbf{F}^{\mathrm{c}} = \begin{bmatrix} (\rho\mathbf{v})\cdot\mathbf{n} \\ (\rho\mathbf{v}\otimes\mathbf{v} - p\mathbf{I})\cdot\mathbf{n} \\ (\rho E\mathbf{v} - p\mathbf{I})\cdot\mathbf{n} \end{bmatrix}, \tag{2.10}$$

the viscous contribution

$$\mathbf{F}^{\mu} = \begin{bmatrix} 0\cdot\mathbf{n} \\ -\Upsilon\cdot\mathbf{n} \\ -\Upsilon\mathbf{v}\cdot\mathbf{n} \end{bmatrix}, \tag{2.11}$$

heat-flux densities

$$\mathbf{F}^{\mathrm{q}} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{q}\cdot\mathbf{n} \end{bmatrix}, \tag{2.12}$$

and the volume-force vector

$$\mathbf{f} = \begin{bmatrix} 0\cdot\mathbf{n} \\ (\rho\mathbf{g})\cdot\mathbf{n} \\ (\rho\mathbf{g}\cdot\mathbf{v})\cdot\mathbf{n} \end{bmatrix}. \tag{2.13}$$

Thermodynamic closure of the NSE is obtained by an EOS in the form

$$p = f(\rho, e), \tag{2.14}$$

which relates pressure, density, and energy.

2.1.1 Multiphase Flows

The interaction between immiscible phases is typically modelled with either interface tracking or interface capturing schemes. The schemes relevant to this thesis of either group define the interface with respect to a material or numerical property. E. g. in the diffusive interface model [1, 6] the ratio of the specific heats $\gamma(\mathbf{x}, t)$ and the background pressure $p_\infty(\mathbf{x}, t)$ of the stiffened gas EOS [93] are used to define $\Gamma(\mathbf{x}, t) = \frac{1}{\gamma - 1}$ and $\Omega(\mathbf{x}, t) = \frac{\gamma p_\infty}{\gamma - 1}$. For these two properties the additional advection equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\Gamma \\ \Omega\end{pmatrix} + \frac{\partial}{\partial x}u\begin{pmatrix}\Gamma \\ \Omega\end{pmatrix} + \frac{\partial}{\partial y}v\begin{pmatrix}\Gamma \\ \Omega\end{pmatrix} + \frac{\partial}{\partial z}w\begin{pmatrix}\Gamma \\ \Omega\end{pmatrix} = 0, \tag{2.15}$$

are solved alongside eqs. (2.1) to (2.3). The interface position is then defined with respect to transition in γ. Note, for an ideal gas law only the first line of eq. (2.15) is required. Consequently, more complex EOSs might require solving even more equations. In VOF formulations [101, 161], a volume fraction function $\alpha(\mathbf{x}, t)$ is advected in addition to the NSE, as

$$\frac{\partial \alpha}{\partial t} + (\mathbf{v}\cdot\nabla)\alpha = 0. \tag{2.16}$$

Therein, $\alpha = 1$ in one and $\alpha = 0$ in the other fluid, respectively. For immiscible fluids the interface is defined by the $\alpha = 0.5$ contour. Despite this precise definition for the interface location, the actual state change from one fluid to the other is smeared out over all cells in

which $0 < \alpha < 1$. The volume-fraction approach is typically coupled with mixture models for the fluid states [4, 130]. For example the mass conservation eq. (2.1) is split for each phase to read

$$\frac{\partial \alpha \rho_1}{\partial t} = -\operatorname{div}(\alpha \rho_1 \mathbf{v}), \tag{2.17}$$

$$\frac{\partial (1-\alpha) \rho_2}{\partial t} = -\operatorname{div}((1-\alpha) \rho_2 \mathbf{v}). \tag{2.18}$$

Therein, the overall density is given by the weighted sum of the partial densities. Similarly, eqs. (2.2) and (2.3) may be split and enriched by additional terms to model more complex thermodynamic phenomena [185].

In contrast, interface tracking schemes such as free-Lagrange [11, 227], front-tracking [78, 79, 225, 229], or the LS algorithm [47, 167, 213] represent the interface sharply. Here, the focus lies on the LS approach as it locates the interface accurately and simplifies imposing arbitrary coupling conditions between the phases. Furthermore, it handles topological changes [77] naturally and gives geometric quantities via straightforward calculations [201]. In addition, it allows prescribing interfacial physics directly [33]. By adjusting the exchange terms across the interface the LS model can also be used to model free surfaces or solid boundaries with complex geometries [154]. For a more extensive overview about LS methods and comparison to other interface modeling approaches the interested reader is referred to [167, 201].

In the LS method, also, the additional advection equation

$$\frac{\partial \phi}{\partial t} = -u_S \mathbf{n}_S^{\xi} \cdot \nabla \phi = -u_S |\nabla \phi|, \tag{2.19}$$

with the interface velocity $u_S(\mathbf{x}, t)$ obtained from solving the two-material Riemann problem for the stress balance

$$-\mathbf{n}_S \cdot \mathbf{\Pi}(\mathbf{U}_S^1) \cdot \mathbf{n}_S + \mathbf{n}_S \cdot \mathbf{\Pi}(\mathbf{U}_S^2) \cdot \mathbf{n}_S = -\sigma \mathbf{n}_S (\nabla^T \cdot \mathbf{n}_S) \cdot \mathbf{n}_S, \tag{2.20}$$

is solved for the signed-distance function $\phi(\mathbf{x}, t)$. Therein,

$$\mathbf{n}_S^{\xi}(\mathbf{x}) = \frac{\nabla \phi(\mathbf{x})}{\|\nabla \phi(\mathbf{x})\|}, \tag{2.21}$$

is the interface normal pointing towards the respective fluid, indicated by $\xi \in 1, 2$. By defining the interface as

$$S(\mathbf{x}, t) = \{\mathbf{x} \,:\, \phi(\mathbf{x}, t) = 0\}, \tag{2.22}$$

an exact representation of the interface is obtained. Hence, this method is called a *sharp-interface method*. Away from the interface, the sign of ϕ indicates which of the two fluids is present. Advecting the interface according to eq. (2.19) might, however, violate the signed-distance property of ϕ. To recover this property the steady-state solution of the *reinitalization* equation

$$\frac{\partial \phi}{\partial \tau} = \operatorname{sign}(\phi)(1 - |\nabla \phi|) \tag{2.23}$$

needs to be found in pseudo-time τ [213]. However, finding such a solution while maintaining conservation, i. e. leaving the zero contour intact, is challenging. Numerous types of reinitalization techniques have been developed addressing this issue [156, 166]. Evaluating fluxes at the interface across two phases may produce unphysical oscillations. Therefore,

the *ghost-fluid method (GFM)* [59] treats the phases separately. The states of each fluid are extrapolated by the steady-state solution of

$$\frac{\partial \mathbf{U}(\mathbf{x})}{\partial \tau} = \mathbf{n}_S(\mathbf{x}) \cdot \nabla \mathbf{U}(\mathbf{x}). \tag{2.24}$$

Interactions between two fluids are considered by adding

$$-\oint_S \left(\mathbf{F}_S^{\mathrm{c}} + \mathbf{F}_S^{\mu} + \mathbf{F}_S^{\mathrm{q}}\right) d\partial V \tag{2.25}$$

on the right-hand side of eq. (2.9). Where the interface-flux densities are defined as

$$\mathbf{F}_S^{\mathrm{c}} = \begin{bmatrix} (\rho \mathbf{v}) \cdot \mathbf{n}_S \\ (\rho \mathbf{v} \otimes \mathbf{v} - p\mathbf{I}) \cdot \mathbf{n}_S \\ (\rho e \mathbf{v} - p\mathbf{I}) \cdot \mathbf{n}_S \end{bmatrix} \tag{2.26}$$

for convection,

$$\mathbf{F}_S^{\mu} = \begin{bmatrix} 0 \cdot \mathbf{n}_S \\ -\Upsilon \cdot \mathbf{n}_S \\ -\Upsilon \mathbf{v} \cdot \mathbf{n}_S \end{bmatrix} \tag{2.27}$$

for viscosity and

$$\mathbf{F}_S^{\mathrm{q}} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{q} \cdot \mathbf{n}_S \end{bmatrix}, \tag{2.28}$$

for heat-transfer, respectively. The LS algorithm allows using distinct EOSs for each fluid as long as the interface states are compatible.

2.2 Numerical Model

The aforementioned equations are discretized to be solved numerically. A DNS approach is followed without any models for under-resolved physical effects in contrast to e. g. large-eddy or Reynolds-averaged Navier-Stokes (RANS) simulations. While this eliminates modeling errors it requires finer discretization and hence more compute power as discussed later in this thesis.

The domain V is discretized by partitioning it into a disjunct set of cuboid finite volumes

$$V_{i,j,k} \leq V \,|\, \Sigma V_{i,j,k} = V \tag{2.29}$$

with cell size Δx. Note, throughout this work subscript $i, j, k \in \mathbb{N}$ are indices for the three spatial directions, respectively. Also, they indicate that the respective quantity is discretized. Interface capturing schemes can be computed directly on the thus defined mesh. Sharp-interface schemes, however, require additional attention as each cell may contain part of the fluid interface S.

In the LS approach, the interface is linearized inside such cells, and cell-face apertures $A^{\xi}_{i,j\pm\frac{1}{2},k\pm\frac{1}{2}}(t)$, $A^{\xi}_{i\pm\frac{1}{2},j,k\pm\frac{1}{2}}(t)$, and $A^{\xi}_{i\pm\frac{1}{2},j\pm\frac{1}{2},k}(t)$ are defined, which describe the cell-face fraction

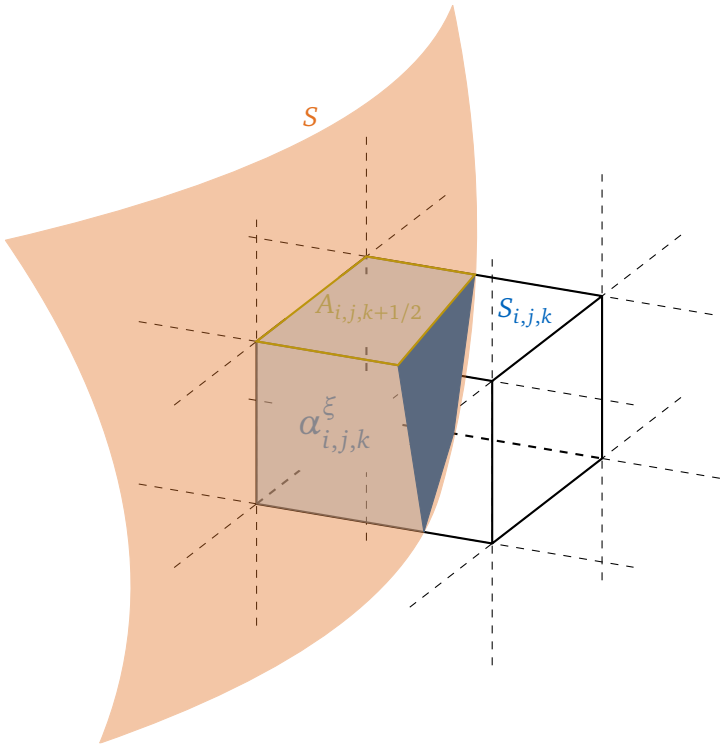

Figure 2.1: Schematic of a cut cell. The curved interface S (orange) is numerically approximated by the flat surface $S_{i,j,k}$ (blue). Exemplarily, the aperture $A_{i,j,k+1/2}$ is shown in green. The volume fraction $\alpha_{i,j,k}$ is shaded in light blue.

covered by fluid ξ. From them, the computation of the volume fraction $\alpha^{\xi}_{i,j,k}$ is straightforward. The volume fraction denotes the volumetric portion of $V_{i,j,k}$ which is covered by the respective fluid. Figure 2.1 visualizes the geometric properties in a cell cut by the interface. From this interface discretization two partially-overlapping sets of cells are defined as

$$\mathcal{V}^{\xi} = \left\{ V_{i,j,k} \middle| \, \alpha^{\xi}_{i,j,k} \neq 0 \right\}, \text{ and} \tag{2.30}$$

$$\mathcal{S} = \left\{ V_{i,j,k} \middle| \, \alpha^{1}_{i,j,k} \neq 0 \text{ and } \alpha^{2}_{i,j,k} \neq 0 \right\}. \tag{2.31}$$

Using volume-averaged conservative quantities indicated by the over bar

$$\bar{\mathbf{U}} = \frac{1}{\alpha^{\xi}_{i,j,k}} \int\limits_{\Delta^{\xi}_{i,j,k}} \mathbf{U} \, dV \tag{2.32}$$

together with eqs. (2.30) and (2.31), the discretized form of eq. (2.9) becomes

$$\frac{\partial \left(\alpha^{\xi} \bar{\mathbf{U}} \right)_{i,j,k}}{\partial t} = - \oint\limits_{\partial V^{\xi}_{i,j,k} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{\mathrm{c}} + \mathbf{F}^{\mu} + \mathbf{F}^{\mathrm{q}} \, d\partial V - \oint\limits_{\mathcal{S}_{i,j,k}} \mathbf{F}^{\mathrm{c}}_{S} + \mathbf{F}^{\mu}_{S} + \mathbf{F}^{\mathrm{q}}_{S} \, d\partial V + \int\limits_{V^{\xi}_{i,j,k}} \mathbf{f} \, dV. \tag{2.33}$$

Where the integrals on the right-hand side account for fluxes over the cell faces, the interaction between the fluids, and the volume forces acting on the fluids.

2.2.1 Flux Evaluations

Equation (2.33) is solved by introducing numerical flux functions for each of the fluxes and the volume force term. With these numerical flux functions the conservative balance equations for each term on the right-hand side of eq. (2.33) is defined. E. g., the convective flux integral becomes

$$
\begin{aligned}
-\oint_{\partial V^{\xi}_{i,j,k}\setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{\mathrm{c}}\, d\partial V = \Delta x^2 \Big(& A^{\xi}_{i-\frac{1}{2},j,k}{}^{1}\mathbf{F}^{\mathrm{c},1}_{i-\frac{1}{2},j,k} - A^{\xi}_{i+\frac{1}{2},j,k}\mathbf{F}^{\mathrm{c},1}_{i+\frac{1}{2},j,k} \\
& + A^{\xi}_{i,j-\frac{1}{2},k}\mathbf{F}^{\mathrm{c},2}_{i,j-\frac{1}{2},k} - A^{\xi}_{i,j+\frac{1}{2},k}\mathbf{F}^{\mathrm{c},2}_{i,j+\frac{1}{2},k} \\
& + A^{\xi}_{i,j,k-\frac{1}{2}}\mathbf{F}^{\mathrm{c},3}_{i,j,k-\frac{1}{2}} - A^{\xi}_{i,j,k+\frac{1}{2}}\mathbf{F}^{\mathrm{c},3}_{i,j,k+\frac{1}{2}} \Big).
\end{aligned} \tag{2.34}
$$

Here the number in the superscript indicates the component of the Flux vector $\mathbf{F} = \left[F^1, F^2, F^3\right]^T$ for each spatial dimension, respectively. The remaining integrals on the right-hand side are treated similarly.

Computation of $\mathbf{F}^{\mu}$ and $\mathbf{F}^{q}$ respectively is done by evaluating the gradients in eqs. (2.4) and (2.5) with discretization stencils of desired convergence order at the respective cell faces. Gravitational volume forces are trivially computed evaluating eq. (2.13) at cell centers.

The numerical cell-face flux $\mathbf{F}^{c}$ is determined by solving a Riemann problem between neighboring cells. While exact iterative solvers exists, using explicit approximate Riemann solver has shown to provide sufficient accuracy at a fraction of the computational cost. A variety of such approximate Riemann solvers is found in literature [97, 168, 186, 191, 222]. The solver of Roe [186] and the Harten-Lax-van Leer with restored contact surface (HLLC) [222] are the dominant representatives of two major groups of Riemann solvers commonly used in CFD.

The Roe flux reads

$$
F^{\mathrm{Roe}}_{i+\frac{1}{2},j,k} = \frac{1}{2}\left(F_{i+1,j,k} - F_{i,j,k}\right) - \frac{1}{2}\mathbf{R}_{i+\frac{1}{2},j,k}\left|\Lambda_{i+\frac{1}{2},j,k}\right|\mathbf{R}^{-1}_{i+\frac{1}{2},j,k}\left(\bar{\mathbf{U}}_{i+1,j,k} - \bar{\mathbf{U}}_{i,j,k}\right), \tag{2.35}
$$

with the right eigenvector $\mathbf{R}$ and the diagonal eigenvalue matrix Λ of the Jacobian $\partial\mathbf{F}/\partial\mathbf{U}$. If the solver is directly applied to cell-averaged states as described above the scheme is first-order accurate. Yet, higher convergence orders are achievable by using the flux-vector splitting form of eq. (2.35)

$$
F^{+}_{i+\frac{1}{2},j,k} = F_{i,j,k} + \mathbf{R}_{i+\frac{1}{2},j,k}\left|\Lambda_{i+\frac{1}{2},j,k}\right|\mathbf{R}^{-1}_{i+\frac{1}{2},j,k}\bar{\mathbf{U}}_{i,j,k}, \text{ and} \tag{2.36}
$$

$$
F^{-}_{i+\frac{1}{2},j,k} = F_{i+1,j,k} - \mathbf{R}_{i+\frac{1}{2},j,k}\left|\Lambda_{i+\frac{1}{2},j,k}\right|\mathbf{R}^{-1}_{i+\frac{1}{2},j,k}\bar{\mathbf{U}}_{i+1,j,k} \tag{2.37}
$$

and therein employing higher-order reconstructions of $F^{+}_{i+\frac{1}{2},j,k}$ and $F^{-}_{i+\frac{1}{2},j,k}$, denoted by a hat. The higher-order flux then reads

$$
F^{\mathrm{Roe}}_{i+\frac{1}{2},j,k} = \frac{1}{2}\left(\hat{F}^{+}_{i+\frac{1}{2},j,k} + \hat{F}^{-}_{i+\frac{1}{2},j,k}\right) \tag{2.38}
$$

For the HLLC solver [222], the flux is given by

$$
\mathbf{F}^{\mathrm{HLLC}} = \frac{1+\mathrm{sign}\left(\mathbb{S}^{*}\right)}{2}\left[\mathbf{F}_L + \mathbb{S}^{-}\left(\mathbf{U}^{*}_L - \bar{\mathbf{U}}_L\right)\right] + \frac{1+\mathrm{sign}\left(\mathbb{S}^{*}\right)}{2}\left[\mathbf{F}_R + \mathbb{S}^{+}\left(\mathbf{U}^{*}_R - \bar{\mathbf{U}}_R\right)\right], \tag{2.39}
$$

with

$$
\mathbb{S}^{-} = \min\left(\mathbb{S}_L, 0\right), \text{ and} \tag{2.40}
$$

$$
\mathbb{S}^{+} = \max\left(\mathbb{S}_R, 0\right) \tag{2.41}
$$

for the left and right states denoted by subscripts L and R, respectively. For the signal speeds estimates $\mathbb{S}_L, \mathbb{S}_R, \mathbb{S}^*$ various competing definitions exist [15, 44, 53, 86, 220]. The fluxes are given by $\mathbf{F}_{L/R} = F\left(\bar{\mathbf{U}}_{L/R}\right)$ with

$$\mathbf{U}^*_{L/R} = \frac{\mathbb{S}_{L/R} - \beta}{\mathbb{S}_{L/R} - \mathbb{S}^*} \begin{pmatrix} \rho_{L/R} \\ \rho_{L/R}\zeta_1 \\ \rho_{L/R}\zeta_2 \\ \rho_{L/R}\zeta_3 \\ E_{L/R} + (\mathbb{S}^* - \beta)\left(\rho_{L/R}\mathbb{S}^* + \frac{p_{L/R}}{\mathbb{S}_{L/R} - \beta}\right) \end{pmatrix}. \tag{2.42}$$

If the flux is computed in x^1-direction, $\beta = v^1$, $\zeta_1 = \mathbb{S}^*$, $\zeta_2 = v^2$, and $\zeta_3 = v^3$ for the respective components of the velocity vector $\mathbf{v}$. Along the x^1-direction and x^2-direction, respectively $\beta = v^2$ and $\beta = v^3$. Accordingly, $\zeta_1 = v^1$, $\zeta_2 = \mathbb{S}^*$, and $\zeta_3 = v^3$ or $\zeta_1 = v^1$, $\zeta_2 = v^2$, and $\zeta_3 = \mathbb{S}^*$. To obtain higher orders when using the HLLC solver, the input left and right states are determined from a high-order reconstruction.

For the reconstruction in both solver types, typically, essentially non-oscillatory (ENO) [96] type stencils are chosen as they minimize spurious oscillations. The most prominent such stencil is the (fifth-order) WENO stencil by Jiang and Shu [118]. It uses a nonlinear weighting of three second-order three-point sub-stencils $\Theta_p, p \in 1, 2, 3$. For each stencil an accompanying smoothness measure is computed. If the input function is sufficently smooth all stencils are optimally weighted and recover the nominal fifth-order convergence rate. Otherwise, the contribution of the sub-stencil containing the non-smooth region of the function is reduced accordingly. The reconstructed value along the first spatial dimension is obtained from

$$\mathbf{U}_{i+\frac{1}{2},j,k} = w_1\bar{\mathbf{U}}_{\Theta_1} + w_2\bar{\mathbf{U}}_{\Theta_2} + w_3\bar{\mathbf{U}}_{\Theta_3}, \tag{2.43}$$

where

$$\bar{\mathbf{U}}_{\Theta_1} = \frac{1}{3}\bar{\mathbf{U}}_{i-2,j,k} - \frac{7}{6}\bar{\mathbf{U}}_{i-1,j,k} + \frac{11}{6}\bar{\mathbf{U}}_{i,j,k}, \tag{2.44}$$

$$\bar{\mathbf{U}}_{\Theta_2} = -\frac{1}{6}\bar{\mathbf{U}}_{i-1,j,k} + \frac{5}{6}\bar{\mathbf{U}}_{i,j,k} + \frac{1}{3}\bar{\mathbf{U}}_{i+1,j,k}, \tag{2.45}$$

$$\bar{\mathbf{U}}_{\Theta_3} = \frac{1}{3}\bar{\mathbf{U}}_{i,j,k} + \frac{5}{6}\bar{\mathbf{U}}_{i+1,j,k} - \frac{1}{6}\bar{\mathbf{U}}_{i+2,j,k}, \tag{2.46}$$

with weights

$$w_p = \frac{a_p}{\sum_{q=0}^{3} a_q}, \tag{2.47}$$

$$a_p = \frac{d_p}{\left(\epsilon + b_p\right)^2}, \tag{2.48}$$

with coefficients $d_1 = 1/10$, $d_2 = 6/10$, and $d_3 = 3/10$. The value $\epsilon = 10^{-6}$ was introduced solely to prohibit division by zero, but has proven to affect the order [26]. Therefore, ϵ is set to machine precision in this work. The missing smoothness indicators read

$$b_1 = \frac{13}{12}\left(\bar{\mathbf{U}}_{i-2,j,k} - 2\bar{\mathbf{U}}_{i-1,j,k} + \bar{\mathbf{U}}_{i,j,k}\right)^2 + \frac{1}{4}\left(\bar{\mathbf{U}}_{i-2,j,k} - 4\bar{\mathbf{U}}_{i-1,j,k} + 3\bar{\mathbf{U}}_{i,jk}\right)^2, \tag{2.49}$$

$$b_2 = \frac{13}{12}\left(\bar{\mathbf{U}}_{i-1,j,k} - 2\bar{\mathbf{U}}_{i,j,k} + \bar{\mathbf{U}}_{i+1,j,k}\right)^2 + \frac{1}{4}\left(\bar{\mathbf{U}}_{i-1,j,k} - \bar{\mathbf{U}}_{i+1,j,k}\right)^2, \text{ and} \tag{2.50}$$

$$b_3 = \frac{13}{12}\left(\bar{\mathbf{U}}_{i,j,k} - 2\bar{\mathbf{U}}_{i+1,j,k} + \bar{\mathbf{U}}_{i+2,j,k}\right)^2 + \frac{1}{4}\left(\bar{\mathbf{U}}_{i,j,k} - \bar{\mathbf{U}}_{i+1,j,k} + \bar{\mathbf{U}}_{i+2,j,k}\right)^2. \tag{2.51}$$

The presented reconstruction is the one from the left. The one from the right is symmetric and obtained accordingly.

This initial scheme has not only been extended to higher [13] and even adaptive convergence orders [12], but has been extended and modified hundreds of times. E. g., the weights have been adjusted to reduce dissipation [72], the direction of the reconstruction has been adjusted creating a blend of upwind and central schemes [108], and in targeted ENO (TENO) sub-stencils containing discontinuities are ignored altogether [70].

The possibility to combine multiple Riemann solvers with multiple reconstruction stencils creates a huge variety of numerical schemes. These schemes provide different benefits such as reduced numerical dissipation or dispersion, but may be vulnerable to unphysical asymmetries or instabilities [64, 66]. Hence, it is beneficial to choose the best-fitting scheme for a particular problem.

2.2.2 Time Integration

The left-hand side of eq. (2.33) is discretized using a time-step size Δt to advance from time t^n to $t^{n+1} = t^n + \Delta t$ with an explicit total variation diminishing (TVD) [94] Runge-Kutta (RK) [134, 190] time integration scheme of second order

$$\bar{\mathbf{U}}^{*}_{i,j,k} = \bar{\mathbf{U}}^{n}_{i,j,k} + \Delta t \mathcal{L}\left(\bar{\mathbf{U}}^{n}_{i,j,k}\right) \tag{2.52}$$

$$\bar{\mathbf{U}}^{n+1}_{i,j,k} = \frac{1}{2}\left(\bar{\mathbf{U}}^{n}_{i,j,k} + \bar{\mathbf{U}}^{*}_{i,j,k}\right) + \frac{1}{2}\Delta t \mathcal{L}\left(\bar{\mathbf{U}}^{*}_{i,j,k}\right) \tag{2.53}$$

or third order

$$\bar{\mathbf{U}}^{*}_{i,j,k} = \bar{\mathbf{U}}^{n}_{i,j,k} + \Delta t \mathcal{L}\left(\bar{\mathbf{U}}^{n}_{i,j,k}\right) \tag{2.54}$$

$$\bar{\mathbf{U}}^{**}_{i,j,k} = \frac{3}{4}\bar{\mathbf{U}}^{n}_{i,j,k} + \frac{1}{4}\bar{\mathbf{U}}^{*}_{i,j,k} + \frac{1}{4}\Delta t \mathcal{L}\left(\bar{\mathbf{U}}^{*}_{i,j,k}\right) \tag{2.55}$$

$$\bar{\mathbf{U}}^{n+1}_{i,j,k} = \frac{1}{3}\bar{\mathbf{U}}^{n}_{i,j,k} + \frac{2}{3}\bar{\mathbf{U}}^{**}_{i,j,k} + \frac{2}{3}\Delta t \mathcal{L}\left(\bar{\mathbf{U}}^{**}_{i,j,k}\right) \tag{2.56}$$

with $\mathcal{L}\left(\bar{\mathbf{U}}_{i,j,k}\right)$ being the discretization operator of the right-hand side of eq. (2.33) as discussed above [82]. The stability criterion for the explicit time integration reads

$$\Delta t_{\mathrm{NSE}} = C \min\left(\Delta t_a, \Delta t_\mu, \Delta t_q, \Delta t_s, \Delta t_S\right), \tag{2.57}$$

with time step limitations according to acoustics [128], viscous shear [212, 213], thermal conductivity [106], volume forces [212, 213] and interface terms [29], respectively. If Δt is constant throughout the simulation, the Courant-Friedrichs-Lewy (CFL) constant has to be reduced from the theoretical stability limit $C < 1$ to account for a large safety factor.

Otherwise, Δt is recalculated every time step according to eq. (2.57) where

$$\Delta t_a = \max_{\Delta_{i,j,k} \in \mathcal{V}^1 \cup \Delta_{i,j,k} \in \mathcal{V}^2} \left(\frac{\Sigma \| |\mathbf{v}| + a \|}{\Delta x} \right)^{-1}, \tag{2.58}$$

$$\Delta t_\mu = \max_{\Delta_{i,j,k} \in \mathcal{V}^1 \cup \Delta_{i,j,k} \in \mathcal{V}^2} \left(\frac{14\mu}{3\rho \Delta x^2} \right)^{-1}, \tag{2.59}$$

$$\Delta t_s = \max_{\Delta_{i,j,k} \in \mathcal{V}^1 \cup \Delta_{i,j,k} \in \mathcal{V}^2} \left(\frac{\Sigma \| \|\mathbf{v}\| + a \| + \sqrt{\Sigma \| |\mathbf{v}| + a \| + 4|g|\Delta x}}{2\Delta x} \right)^{-1}, \tag{2.60}$$

$$\Delta t_q = \max_{\Delta_{i,j,k} \in \mathcal{V}^1 \cup \Delta_{i,j,k} \in \mathcal{V}^2} \left(\frac{14k}{3\rho c_p \Delta x^2} \right)^{-1}, \text{ and} \tag{2.61}$$

$$\Delta t_S = \max_{\Delta_{i,j,k} \in S} \left(\sqrt{\frac{8\pi\sigma}{\Delta x^3 \sum_i \rho^\xi}} \right)^{-1}, \tag{2.62}$$

with the cell-local speed of sound a.

2.2.3 Interface Discretization

Depending on the interface representation, cf. section 2.1.1, different numerical schemes need to be considered for the discretization and computation of interface position and the exchange terms across it. In diffusive interface and VOF schemes interface smearing needs to be prevented, e. g. by introducing anti-diffusion [205] or reverting to more complex geometric VOF formulations [129]. The interface position, its velocity and curvature are obtained from reconstruction algorithms [9]. Furthermore, special care has to be taken to prevent spurious oscillations at the interface [1], in particular when using high-order methods like WENO-type reconstruction [119, 121]. In the LS method, the interface position is trivially obtained from eq. (2.22). Solving eq. (2.21) via standard finite-difference schemes of desired order further allows to a straightforward computation of its curvature.

Similarly, the right-hand sides of eqs. (2.19), (2.23) and (2.24) can be discretized using finite-differences. However, the application of reconstruction stencils may be advantageous [117, 156, 166]. The left-hand side of eq. (2.19) is integrated with the same scheme as the fluid states, cf. section 2.2.2. For the integration of eqs. (2.23) and (2.24) in pseudo-time, however, a one-step Euler integration [56] is sufficent. In order to solve the Riemann problem eq. (2.20) a two-material Riemann solver needs to be applied [107, 144, 221].

Note, cells with small volume fractions $\alpha^\xi_{i,j,k} \ll 1$ require lowering the time-step size drastically or risk arising of instabilities. To prevent either of these a conservative mixing procedure is applied to clear the cell of the respective phase [109, 137].

2.3 Compression Algorithms

The numerical methods introduced above are computationally demanding. The required compute power is further increased as the studied physical effects such as compressible turbulence or shock-bubble/drop interactions are innately 3D [175, 179, 199, 218] and very high-resolutions are required to resolve enough physical scales. Hence, simulations with billions of degrees of freedom (DOFs) have to be conducted. Luckily, often the predominant physical processes only occupy a fraction of the computational domain [66, 120, 126, 224,

241]. This allows mimicking the effective separation of scales of dominating physical effects by investing more resolution in these areas. Thereby, reducing the computational load compared to a homogeneously discretized domain.

A commonly used technique in compressible CFD is the truncation-error based AMR [17]. It offers the possibility of a straightforward integration of the aforementioned numerical schemes, a regular mesh, and local scale support, i. e. the refinement criterion is local and if information is exchanged between the grids of different resolution only cells in the proximity are involved. Since its refinement is sensitive to the steepness of flow-field gradients, however, regions of interest without strong gradients, e. g. contact discontinuities or smooth rarefaction waves, may not refine properly [45, 46, 199]. A variety of other compression schemes exists such as adaptive (possibly wavelet-based) DGM [40, 243], multigrid [60], sparse girds [204] or wavelet collocation methods [231]. However, combining the previously introduced numerical schemes within these algorithms is challenging due to a lack of discrete conservation, heterogeneous meshes, or wide scale support.

In this regard, the MR scheme [95] is advantageous as it creates regular meshes, allows straightforward application of the numerical schemes presented in section 2.2, has local scale support, and adapts also on relevant but relatively smooth flow features, typically even with higher compression rates than AMR [45, 46, 199]. MR uses dyadic grids which naturally results in octree data structures in 3D. In the tree nodes may be *leaves* if they are not further refined with *children* or *parents* otherwise. Same-level neighboring nodes are called *brothers* or *cousins* depending on whether they share a parent. If a node does not have a cousin on a side, this side makes up a *resolution jump*. The original flux-compression [95] MR scheme is natively conservative, but requires more memory than a comparable homogeneous grid. Its *fully-adaptive* form [41, 189] also compresses memory, yet it requires a flux adjustment at resolution jumps [189].

The MR method is based on the wavelet-form of the solution function $\mathbf{U}(\mathbf{x},t)$

$$\mathbf{U}(\mathbf{x},t) = \sum_i \sum_j \sum_k c_{i,j,k}\theta_{i,j,k}(\mathbf{x},t) + \sum_m \sum_i \sum_j \sum_k d^{l_m}_{i,j,k}\psi^{l_m}_{i,j,k}(\mathbf{x},t), \tag{2.63}$$

where the subscript i,j, and k sample the function at grid point $\mathbf{x}^{l_m}_{i,j,k} = \left[2^{-l_m}i, 2^{-l_m}j, 2^{-l_m}k\right]^T, (i,j,k) \in \mathbb{Z}$ on m-th refinement level l_m, $m \in \mathbb{N}$. The first sum over the products of scaling coefficients c_k and compact scaling functions θ_k defines the general shape of the function. The *details* $d^{l_m}_{i,j,k}$ and the wavelet function $\psi^{l_m}_{i,j,k}$ define the local fluctuations on the respective level. Hence, the details indicate the smoothness of the solution. Compression is thus achieved by setting vanishing details to zero. By using *interpolation wavelets* the details can be computed without a full wavelet transformation by a simple difference between the solution function on different levels.

$$d^{l_m}_{i,j,k} = \sum_i \sum_j \sum_k d^{l_m}_{i,j,k}\psi^{l_m}_{i,j,k} = \bar{\mathbf{U}}^{l_{m+1}}_{i,j,k} - \bar{\mathbf{U}}^{l_m}_{i,j,k}, \tag{2.64}$$

The solution can be propagated to higher, i. e. finer, or lower levels using the consistently defined *averaging* and *prediction* operators [95, 103], illustrated in fig. 2.2. Mathematically, the averaging operator is defined as

$$\mathcal{A}\left(\bar{\mathbf{U}}^{l_{m+1}}\right) \to \bar{\mathbf{U}}^{l_m} : \bar{\mathbf{U}}^{l_m} = \frac{1}{N^3}\sum_p^N \sum_q^N \sum_r^N \bar{\mathbf{U}}^{l_{m+1}}_{p,q,r}, \tag{2.65}$$

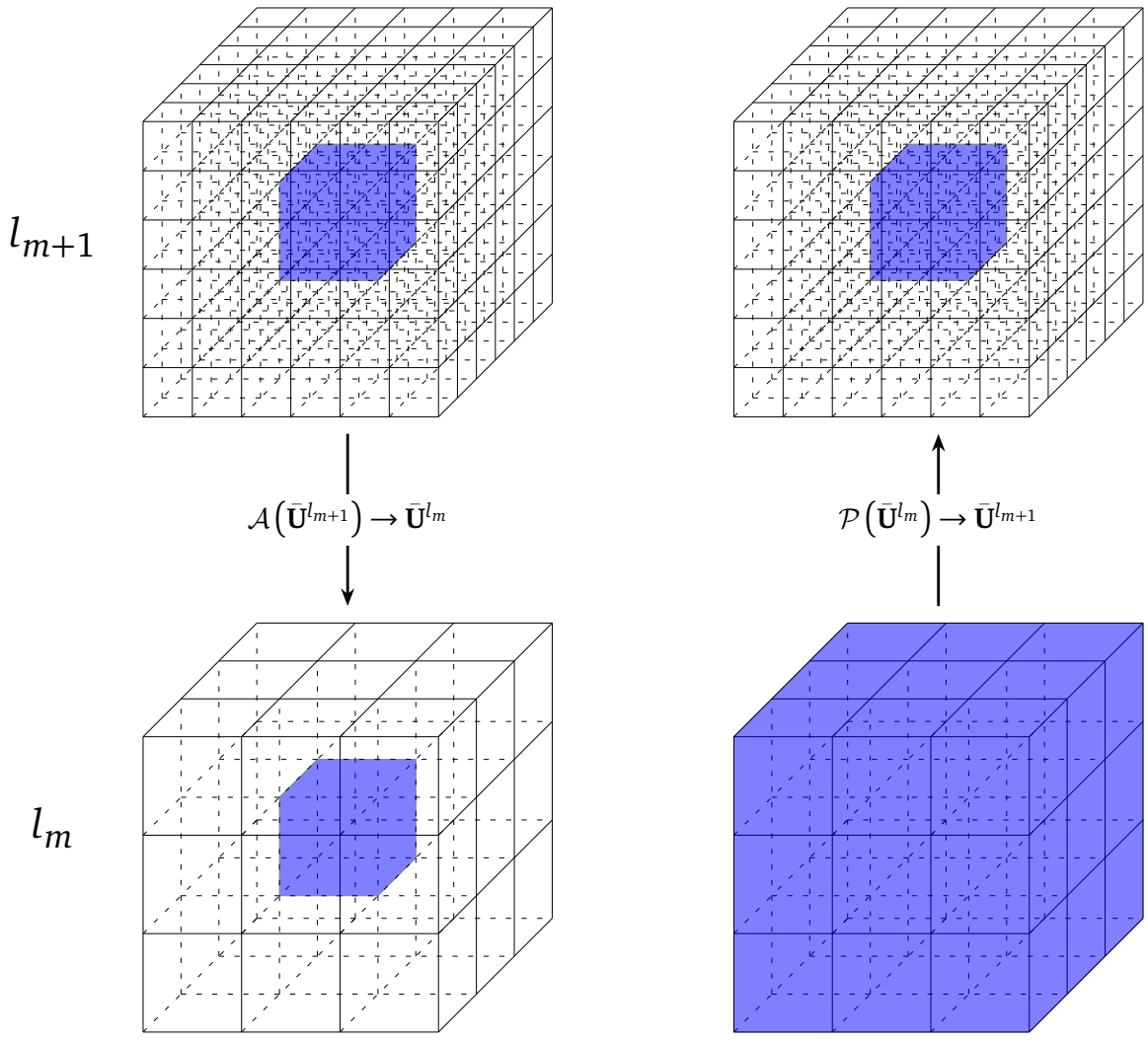

Figure 2.2: Averaging (left) and third-order prediction (right) operators in 3D. The cells used in the computation of the respective operator are colored. The coarse grid l_m is on the bottom with the once refined grid atop. Note, in the finer mesh eight cells are highlighted in both operators.

and the prediction operator as

$$\begin{aligned}\mathcal{P}\left(\bar{\mathbf{U}}^{l_m}\right) \rightarrow \bar{\mathbf{U}}^{l_{m+1}}:\\ \bar{\mathbf{U}}^{l_{m+1}}_{i,j,k} = \bar{\mathbf{U}}^{l_m}_{i,j,k} + (-1)^i Q^x_{i,j,k} + (-1)^j Q^y_{i,j,k} + (-1)^k Q^z_{i,j,k}\\ + (-1)^i(-1)^j Q^{xy}_{i,j,k} + (-1)^i(-1)^k Q^{xz}_{i,j,k} + (-1)^j(-1)^k Q^{yz}_{i,j,k}\\ + (-1)^i(-1)^j(-1)^k Q^{xyz}_{i,j,k},\end{aligned} \tag{2.66}$$

with the contributions of each separate dimension

$$Q^x_{i,j,k} = \sum_{p=1}^{s} \omega_p \left(\bar{\mathbf{U}}^{l_m}_{i+p,j,k} - \bar{\mathbf{U}}^{l_m}_{i-p,j,k}\right), \tag{2.67}$$

$$Q^y_{i,j,k} = \sum_{p=1}^{s} \omega_p \left(\bar{\mathbf{U}}^{l_m}_{i,j+p,k} - \bar{\mathbf{U}}^{l_m}_{i,j-p,k}\right), \tag{2.68}$$

$$Q^z_{i,j,k} = \sum_{p=1}^{s} \omega_p \left(\bar{\mathbf{U}}^{l_m}_{i,j,k+p} - \bar{\mathbf{U}}^{l_m}_{i,j,k-p}\right), \tag{2.69}$$

of two-dimensional cross interactions

$$Q^{xy}_{i,j,k} = \sum_{p=1}^{s} \omega_p \sum_{q=1}^{s} \omega_q \left(\bar{\mathbf{U}}^{l_m}_{i+p,j+q,k} - \bar{\mathbf{U}}^{l_m}_{i-p,j+q,k} - \bar{\mathbf{U}}^{l_m}_{i+p,j-q,k} + \bar{\mathbf{U}}^{l_m}_{i-p,j-q,k}\right), \tag{2.70}$$

$$Q^{xz}_{i,j,k} = \sum_{p=1}^{s} \omega_p \sum_{r=1}^{s} \omega_r \left(\bar{\mathbf{U}}^{l_m}_{i+p,j,k+r} - \bar{\mathbf{U}}^{l_m}_{i-p,j,k+r} - \bar{\mathbf{U}}^{l_m}_{i+p,j,k-r} + \bar{\mathbf{U}}^{l_m}_{i-p,j,k-r}\right), \tag{2.71}$$

$$Q^{yz}_{i,j,k} = \sum_{q=1}^{s} \omega_q \sum_{r=1}^{s} \omega_r \left(\bar{\mathbf{U}}^{l_m}_{i,j+q,k+r} - \bar{\mathbf{U}}^{l_m}_{i,j+q,k-r} - \bar{\mathbf{U}}^{l_m}_{i,j-q,k+r} + \bar{\mathbf{U}}^{l_m}_{i,j-q,k-r}\right), \tag{2.72}$$

and of 3D cross interactions

$$\begin{aligned}Q^{xyz}_{i,j,k} = \sum_{p=1}^{s} \omega_p \sum_{q=1}^{s} \omega_q \sum_{r=1}^{s} \omega_r \left(\bar{\mathbf{U}}^{l_m}_{i+p,j+q,k+r} - \bar{\mathbf{U}}^{l_m}_{i+p,j+q,k-r} - \bar{\mathbf{U}}^{l_m}_{i+p,j-q,k+r}\right.\\ \left. + \bar{\mathbf{U}}^{l_m}_{i+p,j-q,k-r} - \bar{\mathbf{U}}^{l_m}_{i-p,j+q,k+r} + \bar{\mathbf{U}}^{l_m}_{i-p,j+q,k-r} + u^{i-p,j-q,+r}_{l_m} - u^{i-p,j-q,k-r}_{l_m}\right).\end{aligned} \tag{2.73}$$

The resulting MR scheme is of order $\mathcal{O}(2s+1)$ and the coefficients ω_p read

$$\omega_1 = -\frac{1}{8} \qquad \text{for } s = 1, \text{ and} \tag{2.74}$$

$$\omega_1 = -\frac{22}{128},\ \omega_2 = \frac{3}{128} \qquad \text{for } s = 2. \tag{2.75}$$

The tensor structure of the Q-terms is illustrated in fig. 2.3.

The spatial compression allows for additional compression in time. Therein, coarser regions are advanced with a proportionally larger time step than in the finer regions. For the dyadic meshes of MR and AMR schemes the time step is simply scaled as

$$\Delta t_{l_m} = 2^{l_{\max} - l_m} \Delta t_{l_{\max}} \tag{2.76}$$

relative to the time step $\Delta t_{l_{\max}}$ on the finest level computed from eq. (2.57). In a straightforward implementation such local-time stepping (LTS) schemes synchronize the time-step sizes after the lowest level has been advanced once [51].

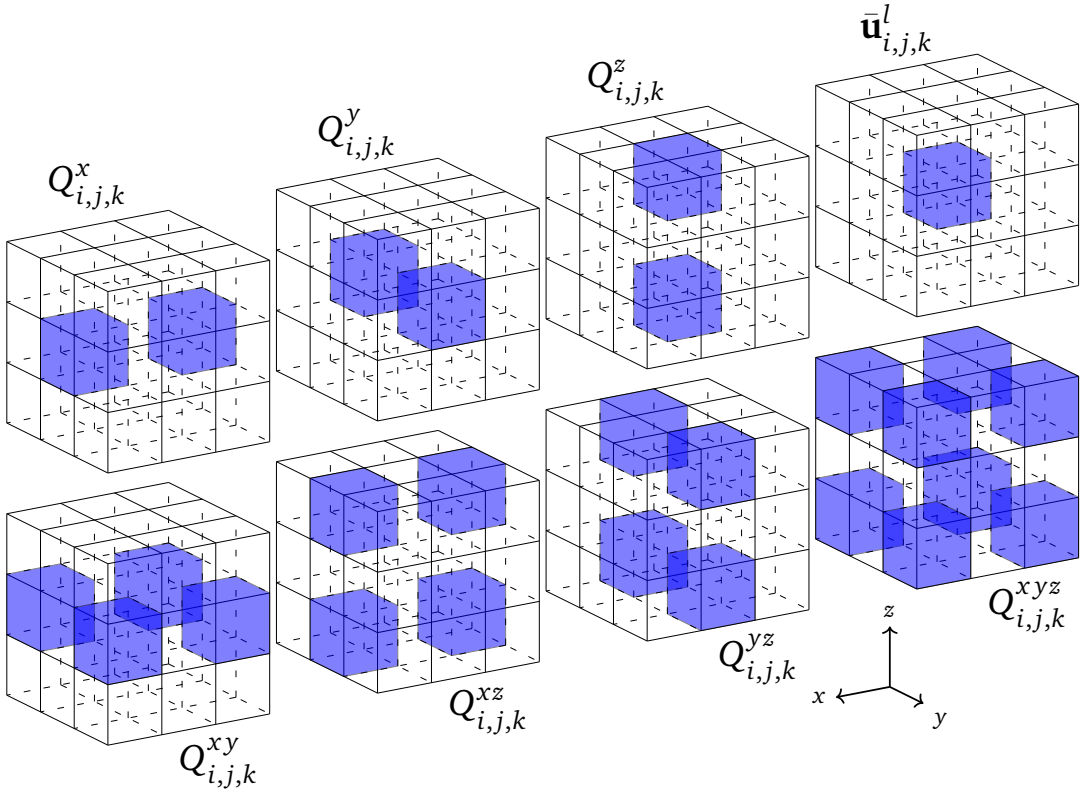

Figure 2.3: Q-terms for 3D a third-order accurate prediction. Cells used within the respective term are colored. The cubes represent the same $3 \times 3 \times 3$ surrounding of cell i, j, k within a larger mesh.

2.4 Build-up of Supercomputers

Despite application of the before mentioned compression algorithms, the numerical load of 3D compressible flows is still too high for average personal computers (PCs). Rather, HPC systems need to be employed. Writing efficient software for these systems, requires a thorough understanding of its hardware and the computational concepts employed in them. Until the late 1990s it was of little importance for the programmer to know these details. All programs benefited equally from the rapidly and continuously growing *clock frequency*, i. e. the speed at which the chip conducts one operation. Around the change of the century, however, the high clock frequency together with the shrinking size of the transistors lead to higher power dissipation in form of heat [27]. Hence, current and future (silicon-based) chips employ multiple levels of parallelism to increase the performance without increasing (electrical) power demands [28]. A wide variety of special-purpose architectures such as graphics processing units (GPUs) or tensor processing units (TPUs) has emerged with different emphasis on the respective levels of parallelism. Hence, the actual implementation must explicitly exploit each level of parallelism to yield good performance on the respective architecture [214].

Within an HPC system the obvious level of parallelism is the *distributed-memory* parallelism. Comparably to a manifold of PC connected via a local area network (LAN), a specialized interconnect links the compute nodes of a super computer together. The nodes typically hold two or four sockets in which the respective central processing unit (CPU) and its main memory is located. Even though the random-access memory (RAM) is shared between the CPUs of a node the access times depend on the physical location of data, i. e. to which socket it is associated. This non-uniform memory access (NUMA) design brings forth possible locality and contention problems. If data needs to be fetched from a remote location the application

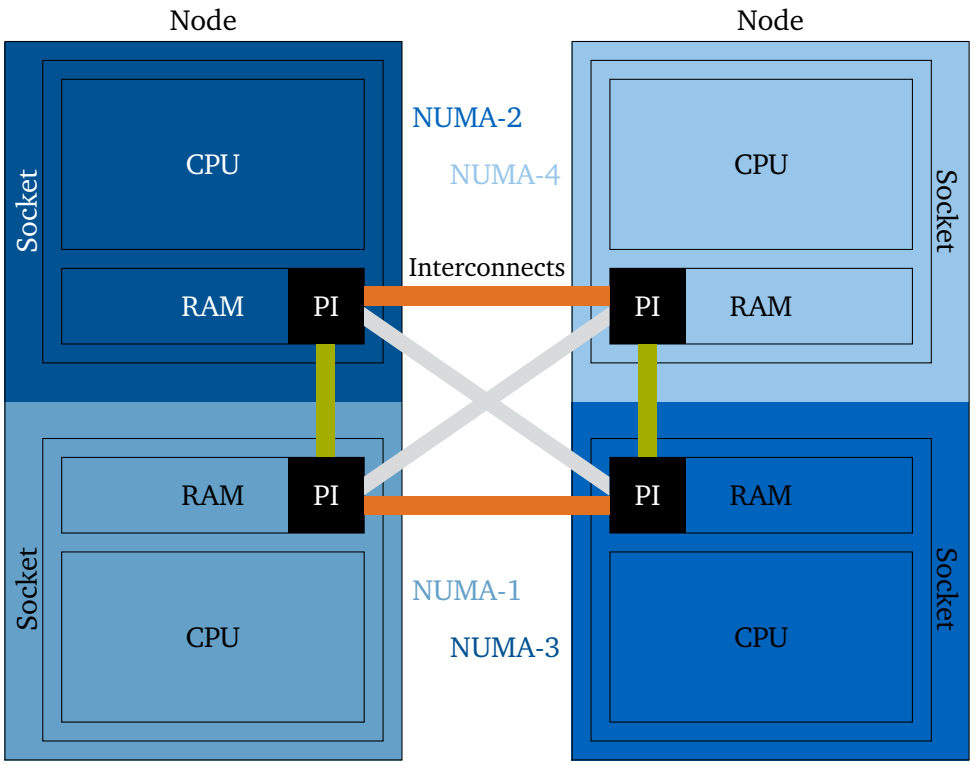

Figure 2.4: Schematic of the interconnect between and within compute node. The two nodes hold two sockets, each a separate NUMA domain. Data access to remote NUMA domains is routed through the processor interconnect (PI).

performance may degrade. Furthermore, the bandwidth of the interconnect may be depleted if many remote accesses to the same location happen simultaneously. Figure 2.4 shows the NUMA domains for an exemplary two-node machine. The main memory of the whole HPC machine is sum of the main memory available on all its compute nodes. No central intelligence keeps track of in which node's RAM a certain piece of information is stored. Hence, the application has to actively manage data transfer between the nodes.

A variety of different libraries for this communication between nodes exists, cf. [25, 123, 203], but the Message Passing Interface (MPI) [67] is the de-facto standard software running on distributed-memory machines. It is based on point-to-point communication. The MPI runtime starts multiple *ranks*. The ranks are pinned to cores according to user input; oversubscription of cores is possible. All ranks individually start the same program in parallel. Although MPI offers a wide variety of communication routines including collective and one-sided operations, data transfer between ranks is typically triggered via explicit `MPI_Send` and `MPI_Receive` calls from within the application. The asynchronous variants of these functions allow for intermediate work to be done while the messages are being send. Note, an MPI implementation may optimize the transfer based on the network topology and, e. g., replace network transfer with simple load instruction if the ranks are on cores of the same NUMA domain [43]. Such optimizations, however, cannot be actively exploited by developers as they are transparent to the application.

The next level of parallelism is the *shared memory* parallelism. All the cores within a CPU and typically even all cores within the same compute node have direct access to the same 'shared' RAM. Physically close cores may in addition share further parts of the memory hierarchy as depicted in fig. 2.5. In this example, the level three 'L3' cache is shared within a socket and the faster yet smaller L2-cache is only shared within a CPU. The smallest and

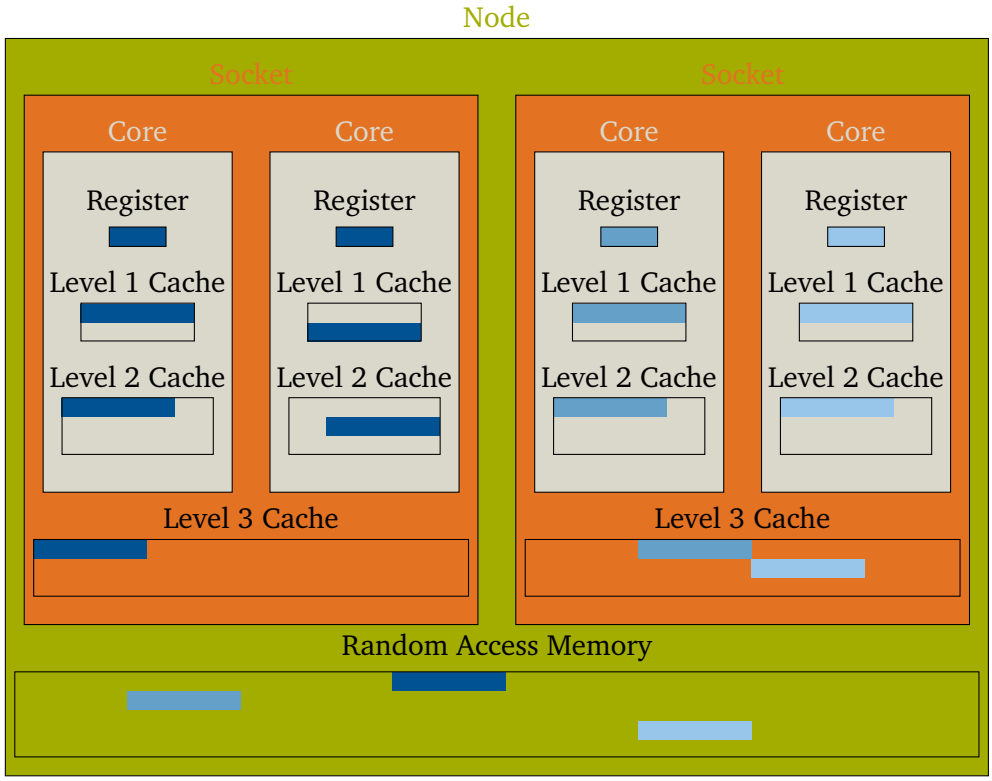

Figure 2.5: Memory hierarchy within a compute node of two sockets with two cores each and three cache levels with a shared level three cache. Note, although the cores in the left socket operate on the same data the data location within the low-level caches differ. The cache-line size is conveniently set to be twice the register size.

fastest L1-Cache is then exclusively used by the respective core. From thereon data is read into the registers on which the bit-wise operations specified by the program are performed.

When data is loaded into the caches, always a whole *cache line* is handled. Such a cache line comprises multiple bytes which are continuously located in a higher memory level. To keep the CPU fed an ideal program ensures that relevant data is grouped physically together in memory and the program flow is predictable. The load and stores are performed coherently, i. e. all loaded cache lines get invalidated if any core writes to this line. Nevertheless, cores can independently execute their computation and, hence, access the memory differently. This is indicated in the figure by the relative position of the colored loaded cache lines within the respective cache. This independence is typically exploited by running *threads*. These threads execute (part of) a program in parallel and may be spawned or joined arbitrarily. The shared-memory eliminates the need of explicit data transfer between threads. Nevertheless, access to the shared data must be coordinated to prevent race conditions and deadlocks. Numerous concepts and libraries for shared-memory parallelization exist, e. g. operating system (OS) level threads [110] are relatively slow to spin-up and typically cumbersome to program, but they provide full OS control to the thread creating application. In contrast, designated HPC threading libraries [123, 165, 183] provide less powerful threads which in turn are much faster created and retired, allowing their programming with simple instructions. It is also possible to use MPI within a shared-memory context as ranks can be pinned to individual cores. Furthermore, the MPI standard version 3 defines a separate shared-memory model [67] and certain implementations may provide additional shared-memory optimization [43].

Further parallelism is provided within the cores of a CPU. Modern cores use different types of registers for the different compute purposes. From only 16 bit-wide registers

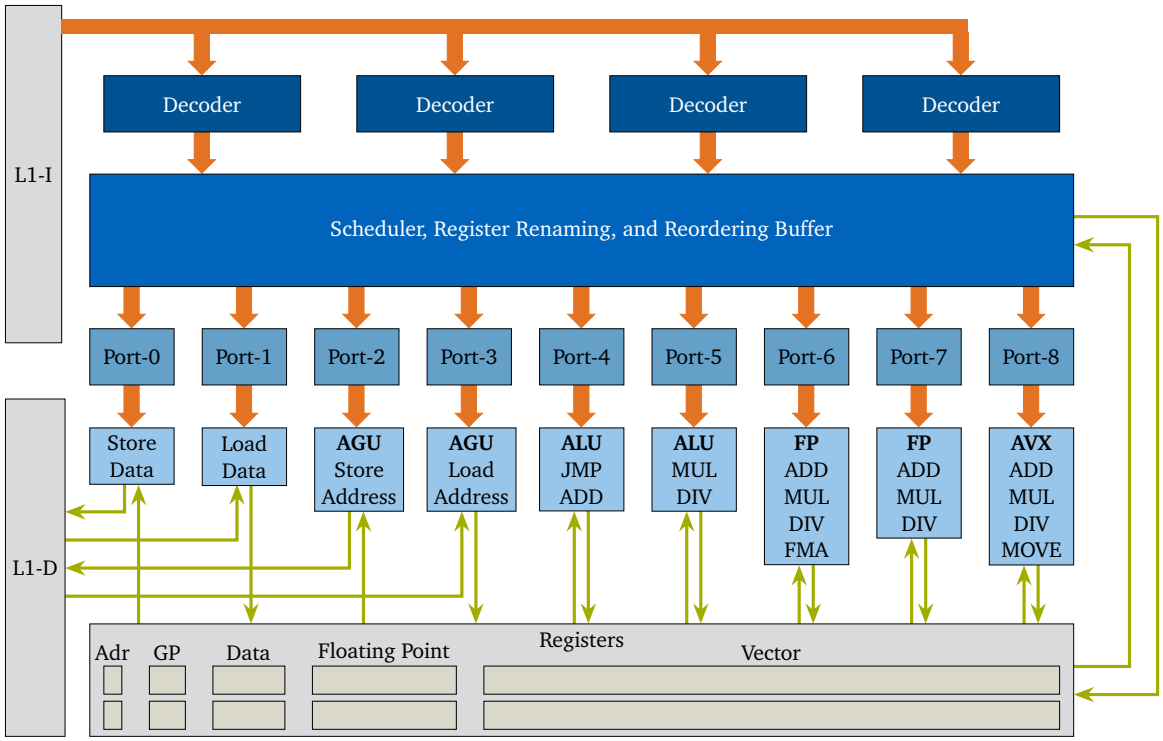

Figure 2.6: Concept of the internal structure of a core. The Level 1 Cache is split into an instruction cache (L1I) and a data cache (L1D). Instruction flow is indicated by thick orange arrows, while data flow is shown by thin green ones. The address and general purposes registers abbreviated Adr and GP, respectively. Instructions executed in the units are abbreviated according to assembly language.

for program instructions and arithmetic operations up-to 512 bit-wide vector registers for floating-point computations. The different registers are processed in different *units* of the processor. For example the integer registers are handled by an arithmetic logic unit (ALU) or address generation unit (AGU), whereas floating-point registers are handled in a floating-point unit (FPU). A schematic of the internals of a core is shown in fig. 2.6. Note, registers, decoders and processing units are duplicated. This *superscalar* architecture allows for multiple instructions being executed simultaneously. This first allows *simultaneous multithreading (SMT)*, i. e. running two different threads on the same core. While this might be beneficial in the context of server farms where diverse applications run concurrently, the scheduling overhead often outweighs the performance gained by the increased throughput for numerical computations [83, 89], and for CFD application in particular [14]. An alternative is to write a single application that fully exploit the superscalar capabilities of the core. This, however, is nearly impossible for sensible applications that where translated to machine code via a compiler [89]. Nevertheless, the abundance of units also enables *out-of-order execution*. Therein, upcoming instructions are saved in a separate buffer. If the instruction is executable with the current register states and the required unit is free in the current cycle, then the instruction is executed on the spot ahead of its ordinary execution time. Although speed-ups are mild, applications benefit automatically. Out-of-order scheduling is done by the hardware and modern compilers optimize for it.

The fine granularity of units, however, allows *pipelining* parallelism. In the pipeline, program operations are split into their instructions which are then executed by the designated unit. If an instruction is repeated multiple times, e. g. in a loop, the throughput of instructions per clock cycle is increased. Exemplary, fig. 2.7 shows an idealized pipeline for one operation consisting of just three instructions “fetch data”, “process data”, “write data”. It is assumed

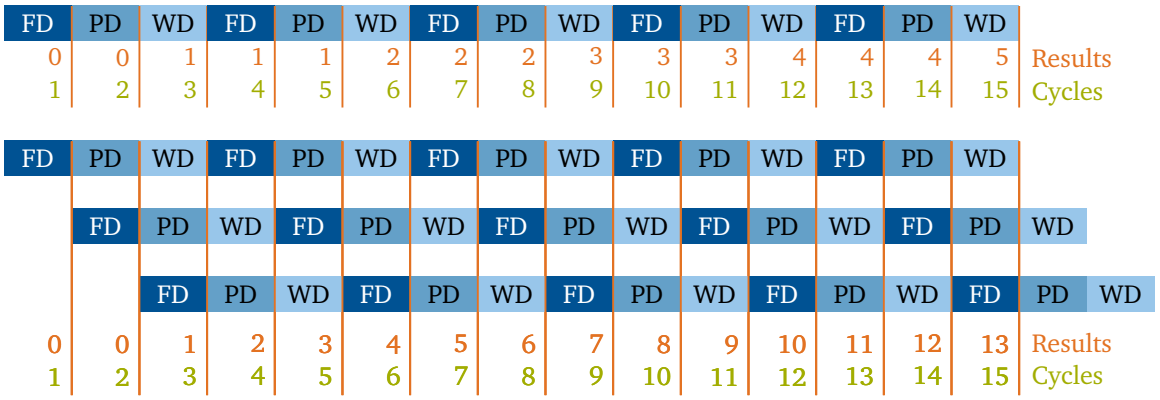

Figure 2.7: Idealized concept of pipelining: Sequential execution at the top and three-way pipeline in the bottom. Each of the instruction *Fetch Data (FD)*, *Process Data (PD)*, and *Write Data (WD)* takes one clock cycle. Once all three instruction have been executed the result is available to the remaining program. The vertical lines indicate the clock cycles. The numbers next to these lines show the number of results computed at the number of clock cycles spent, respectively.

that each instruction is handled by a corresponding unit and the execution of the instruction in these units takes exactly on clock cycle. After an upwind phase of two cycles, one whole operation is finished every cycle. Without pipelining, the completion of one operation, however, requires three cycles. In actual cores, a *reduced instruction set* is employed. This reduced set requires to split the operations into tiny instructions. Hence, operations are broken down into many such operations and modern pipelines contain more than 30 stages. This in turn, makes the up- and downwind phase relatively costly. The application's performance is thus increased by long branch-free loops. To keep the pipeline fed, the data processed within such loops needs to be limited. It further helps to have the data located continuously in memory, to benefit from the cache-line loading mechanism described above.

The final level of parallelism discussed here is SIMD parallelism, also called vectorization. By using wide registers with up to 512 bit the same floating-point operation can be executed on all elements in the register. The speedup depends on the register width and on the precision, e. g. for single-precision computations with 512 bit-wide registers a speedup of 16 over the sequential version is theoretically achievable. A comparison of vectorized and sequential computation of the summation of two vectors for four clock cycles is given fig. 2.8. Similarly to pipelining, vectorization only works on streams of data. Applications need to ensure branch-free loops with predictable memory access patterns in order to benefit from vectorization capabilities of the core. Furthermore, exceptions, data misalignment and pointer aliasing must be prohibited.

2.5 Code Quality

Naturally, it is in the interest of every programmer to write effective and efficient code. I. e. the compiled executable computes the correct answer to the stated problem at the cheapest cost. Note, cost is loosely defined here. Depending on the context it can be a measure of runtime, development time, memory usage, CPU power requirements, monetary cost, etc. Besides these obvious (software) design goals, the quality of the code base is an important metric. The way the source code is written and structured strongly affects its readability, understand-ability, (re-)usability, maintainability, flexibility, portability, testability, and adaptability. This code cleanliness becomes even more important for larger code bases since it

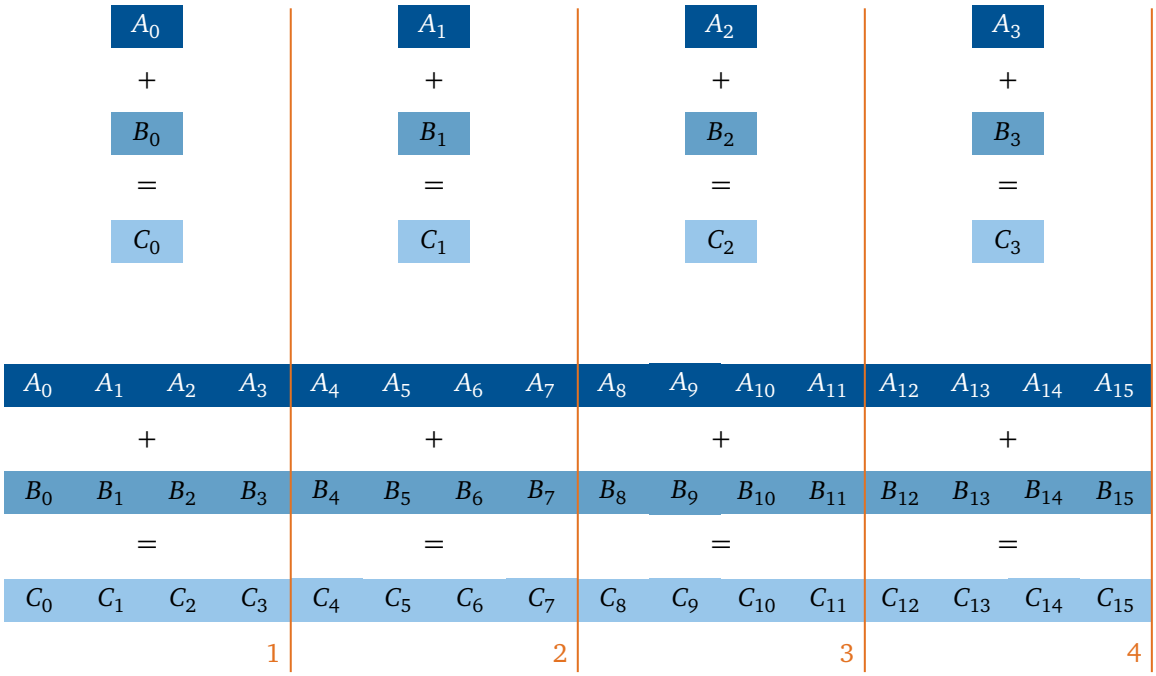

Figure 2.8: Comparison of sequential and four-way vectorized addition of two arrays A, and B. At the top sequential and in the bottom vectorized addition is performed, respectively. The clock cycles are indicated by the vertical lines. An addition is assumed to take one such cycle.

affects the productivity of more developers and users. Even though this software quality is often not incentivized directly, the increased development speed outweighs the added software-cleaning costs. Furthermore, clean code often automatically benefits from improvements in compilers or added libraries.

Although the quality of written code cannot be assessed by a (single) discrete measure, certain rules and guidelines became consensus as best-practice when developing software [84, 148, 149]. These include among others: selection of descriptive variable and function names [58, 69, 149], presence of an automated testing pyramid [136, 155], and adherence to language specific guidelines [81, 115]. For scientific codes in particular, it was found that solely providing automated testing increased the correctness, efficiency, reliability, integrity, accuracy, and robustness of the used software [22].

2.6 Challenges for Numerical Software & Parallel Efficiency

Naturally, the aim for scientific software is to produce the physically correct result as cheaply as possible. As discussed in section 2.2, however, the employed numerical schemes are computationally demanding. Hence, simulations are conducted on HPC systems. For apparent reasons, access to such systems is only granted if an application runs with sufficient (parallel) performance. A simple, yet powerful tool to assess an application's performance is measuring the amount of FLOP it performs and put it in relation to the processed amount of memory. The thus obtained value can be plotted into a *roofline model* [240] to assess its performance. As visualized in fig. 2.9, the roofline model indicates innate performance bounds of the used hardware, such as memory bandwidth limits of the caches in the memory hierarchy and the maximum number of computations an ideally fed unit can compute. The limiting factor is stated atop the respective bound in the figure. An ideal application will appear in the top right corner running at peak performance of the chip even if new data for computation has to be

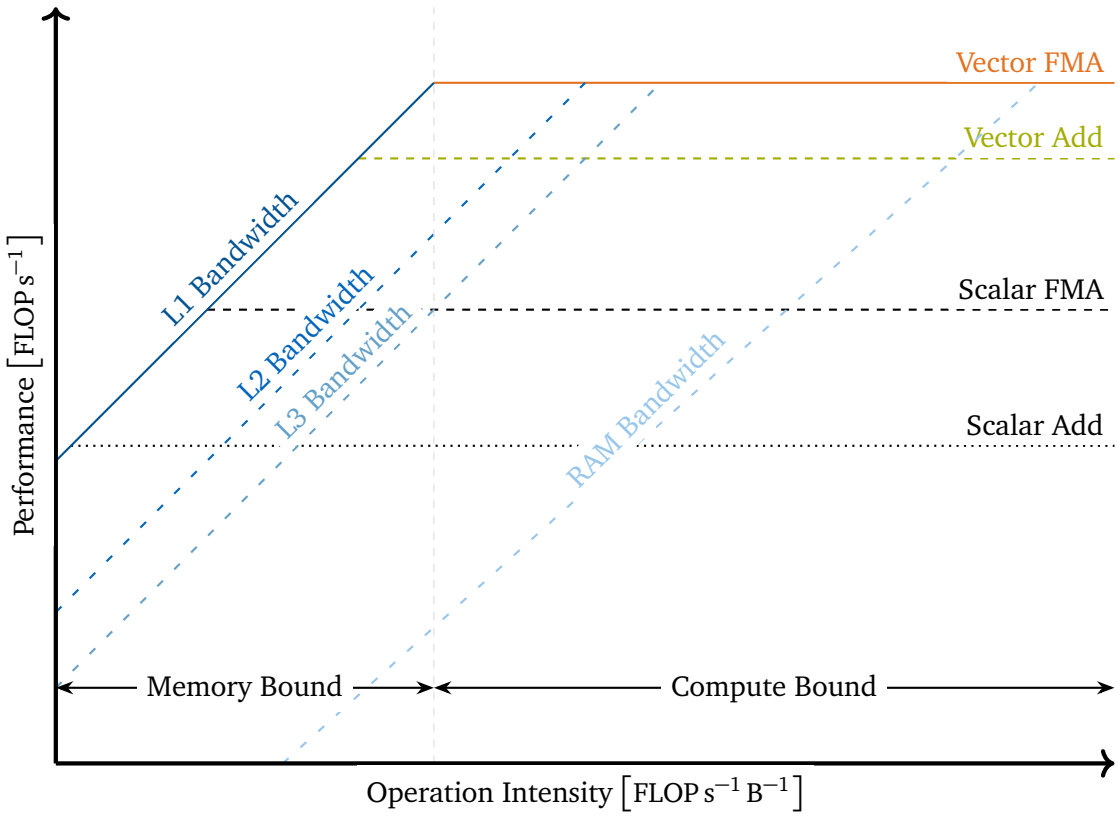

Figure 2.9: Outline of the roofline model in log-log scale. The limiting memory- or computation-type is given by the labeled lines. The bound-regions with respect to the peak performance are indicated by double-sided arrows.

retrieved from slower parts of the memory hierarchy. Such programs are said to be *compute bound* otherwise they are *memory bound*. Raising the memory bandwidth is a greater challenge than to increase the compute capabilities [89]. Therefore, compute-bound applications typically benefit most from newer hardware.

Naturally, the implemented algorithm and the employed numerical schemes have the largest impact on the application's performance. Nevertheless, modifications within the frame of the original algorithm, such as blocking strategies in matrix-matrix multiplications may provide a significant speedup. On a lower level, pure software improvements, such as loop reordering may help to move the application towards the top-right corner in the roofline model. However, extremely trimming the source code for highest performance may lower the code quality and contradict the guidelines stated in section 2.5. As an example, the MPI application programming interface (API) is defined on a low-level of abstraction, giving the application fine-grained control over the communication but hindering the usage of high-level programming paradigms.

The MR compression, the interface discretization, and numerical methods employed in this thesis allow for clean implementations straightforwardly. Nevertheless, incorporating them in a 3D framework with high parallel performance poses some challenges. So does e. g. the mapping of the 3D data to a flat memory complicates streamed data access in all spatial directions, the huge number of computations per value in the reconstruction stencils impede vectorization of the kernels, and nontrivial determination of neighbor values hinders parallelization of the MR algorithm.

2.7 Computational Frameworks for Complex Flows

A handful of CFD frameworks are generally suitable for the study of the complex flows phenomena stated in chapter 1. Theses include the commercial tool suites Ansys [7], Solid Works Flow [200] and Comsol [42], the semicommercial open-source openFOAM [85], and the research codebases Cubism(-Nova) [99, 239], MFC [33], UCNS3D [8] and ECOGEN solver [198]. While the (semi-)commercial frameworks have a wide area of application and provide additional plugin and modules, the research codes typically focus on certain problem types. So does ECOGEN focus on bubble dynamics, MFC on ultrasound and shock-induced bubble collapse, UCNS3D on turbulence effects for aviation, and Cubism on (clouds of) cavitation bubbles. A comparison of the implemented algorithms, the numerical models and the performance of each framework and the one developed within this thesis is given in chapter 5.

Chapter 3

Accomplishments

The main accomplishments of this work cover four categories: Improved distributed-memory parallelization of compression algorithms for compressible finite-volume method direct numerical simulation computational fluid dynamics simulations, improving the vectorization of the aforementioned schemes, enabling large-scale LS simulation by integration of the LS into the improved compression algorithms, and the modularity and cleanliness of the created software framework. The achievements comprising this thesis were published in three first author publications [103, 104, 106]. Note, alongside the manuscript the generated research data was published under permissive license allowing interested researchers to re-use the obtained data [105].

3.1 Improved Multiresolution Compression Algorithm

Here, only MR compression is considered for the reasons stated in section 2.3. A straightforward 3D MR implementation uses an octree to represent the computational mesh [41, 189]. Each node in the tree holds one cell or scale. Numerous such implementations exist [31, 48, 90, 207, 209]. When evaluating their capabilities, however, reveals their lack of support for high-order methods and limited parallel performance. This performance drawback typically stems from the difficulty of obtaining the value of the cell's neighbor states during (cell-face) flux computation. Such values have to be retrieved by a slow tree traversal. Worse, yet if the neighbor is missing on the given refinement level, the value has to be predicted from coarser levels first. Inevitably, this leads to non-regular memory access patterns, which should be avoided [31], cf. section 2.4. For broad high-order stencils this challenge is magnified as multiple neighboring cells are required.

Instead of following this approach, here, a predefined number of cells is packed into a block. Each octree node holds one such block instead of a single cell. This block-based approach is motivated by the remarkable performances reported for similar concepts in AMR and wavelet-collocation methods [61, 188]. This configuration also allows for streamlined and SIMD-optimized flux computations discussed in section 3.2. Equipping each block with its own set of *halo* cells also reduces communication overhead in distributed-memory environments.

The performance of the block-based MR algorithm is further improved by assigning each node an identifier (ID), which is based on a modified Morten order. This allows neighbor lookup by fast bit-shift operations, creation of tiled geometries and eases assignment of nodes to cores via space-filling curves (SFCs). In the implementation, this load balancing is further modified to account for the level of refinement when assigning nodes along the SFC. Although this leads to a more scattered distribution of the nodes, it is crucial to prevent idle cores if LTS is enabled. Details on the block-based MR algorithm, the modified Morton order and the adjusted SFC are published in [103], and summarized below.

Furthermore, an adaptive LTS (ALTS) scheme for temporal compression was developed that is integrated into the block-based MR algorithm. In contrast to previous LTS schemes the time-step size can be adjusted after every time step on the finest level. This is achieved by projecting and averaging the right-hand side of eq. (2.9) instead of the integrated conservative quantities. The higher frequency of time-step size adjustments allows using larger time steps while maintaining numerical stability. The achieved results are published in [128]. My contribution to the latter lies in guidance on integration into the MR algorithm and the implementation in C++ as well as conducting the profiling of the parallel execution of the code.

3.1.1 Summary of: "A parallel modular computing environment for three-dimensional multiresolution simulations of compressible flows"

The manuscript discusses the details of the block-based MR algorithm and its modular implementation in detail. Concepts to enable the desired level of parallelism include a modified Morton order indexing of nodes in the MR-tree and traversing of SFCs in a level-wise manner. A wide range of test cases are simulated to assess the parallel performance across different computational modules such as the type of SFC, the MR-thresholding error norm, the Riemann solver, the reconstruction and derivative stencils, the number of cells per block, and more. The achieved compression is thoroughly analyzed and strong and weak scalings on up to twenty-four thousand cores are performed.

The paper starts with a thorough review of compression algorithms applied to compressible multiphase flows. The differences between AMR and wavelet-based methods are given and the choice of MR is motivated. Publications based on single-scale MR algorithms are examined and their shortcomings with respect to large-scale parallel computations are discussed.

Next, the governing equations and their numerical discretization are presented in detail. Throughout the work, different combinations of the Roe [186], the Rusanov [191] and the HLLC [222] and first-, third-, fifth-, seventh- [118], central upwind sixth-order [108] or adaptive fifth-third-order [12] WENO as well as targeted ENO [70] reconstruction stencils are used.

In the following, details on the genuine MR algorithm and LTS are stated. Then the block-based variant of the MR algorithm with activated ALTS is discussed. The special load-balancing requirements implied by the ALTS scheme are addressed by level-wise modification of SFCs. A modified Morton order is proposed to assign unique IDs to nodes in the MR-octree, allowing neighbor lookup via bit-shift operations from a hash map. This modification allows for straightforward implementation of periodic boundary conditions and non-cubic domains while requiring less memory than previously published indexing schemes [48]. The underlying algorithm is stated in detail. Remarks on the initialization routine and the time-step size restriction are made. Implementation specific details and enforced programming patterns which enable the modularity of the solver are given. The grain size of the parallelization and the MPI-only approach are discussed.

Finally, the correctness of the acoustic and source term solvers are shown by comparison with analytic solutions for the Sod shock tube [206] and the growth-rates of the Rayleigh-Taylor instability (RTI) [35], respectively. A convergence analysis is conducted verifying the nominal convergence orders of the mentioned schemes. It is shown, that the compression of the presented block-based MR variant is comparable to those of 'traditional' single-cell versions, in particular for 3D cases. For numerous test cases, it is shown that the parallel performance is maintained if various of the modular building-blocks are exchanged. This is done by conducting weak and strong scalings for each case on the Leibniz Supercomputing

Centre (LRZ)'s SuperMUC-NG[1] supercomputer.

As lead author of this publication, I wrote the script and addressed the reviewer comments. I have designed, implemented and optimized the modular code framework and in particular the parallelization routines. I have conducted the validation, numerical and physical test cases. Furthermore, I set up and evaluated the scaling runs and have created the shown visualizations. My co-authors have coordinated, supervised, and secured funding for the project. They have also reviewed the manuscript and responses to the reviewers.

3.2 Addressing Auto-Vectorization

The previously introduced block-based MR algorithm allows exploiting the SIMD capabilities of the CPU, cf. section 2.4. It is known from profiling runs that the flux computation accounts for roughly 60 - 75% of the overall runtime. Hence, vectorizing the respective compute kernels yields the highest benefit. However, the numerical scheme to determine the flux does not natively support vectorization. Worse yet, depending on the choice of solvers the processing steps differ. E. g., in the HLLC [222] solver the reconstruction stencil is applied directly on the conservative values whereas in the Roe Riemann solver [186] the stencil operates on characteristic values. Since experimenting with and exchanging these types of compute kernels is the norm in a scientific code base manual vectorization of each routine is destined to fail. Rather than trying manual insertion of intrinsic into the code base, restructured compute routines were employed. While still giving enough flexibility to the developer, the new structure is (more) transparent to the compiler. This results in auto-vectorization of the compute kernels. The details on the revised code structure are presented in [104] and summarized below. Note, the reported conflict between numerical symmetry and vectorization performance have since been resolved. Tests on LRZ's SuperMUC, which also provides 512 bit wide registers, recover fully symmetric flow solutions.

3.2.1 Summary of: "Node-Level Optimization of a 3D Block-Based Multiresolution Compressible Flow Solver with Emphasis on Performance Portability"

This publication starts with listing challenges to SIMD vectorization in flux-based Riemann solvers with high-order reconstruction stencils. These challenges are then addressed by general and clean standard-compliant code improvements, rather than hard-coded intrinsic operations. These improvements result in compiler-generated vectorization and thus an increased performance across compute kernels.

The paper presents the governing NSE and the numerical discretization by the FVM, Riemann solvers and WENO-type reconstruction stencils. Throughout the work, different combinations of the Roe [186] and the HLLC [222] Riemann solver and the fifth- [118] and sixth-order central upwind [108] WENO stencils are used.

Then, the block-based MR algorithm is presented and its potential for efficient use of wide vector registers including its MPI-parallization is discussed. The modular C++ implementation is described and the need for multipurpose optimizations is stated. First performance-baselines are defined on two different Intel compute hardware with 256 bit and 512 bit wide vector registers, respectively. Baselines are defined with respect to the combinations of compute kernels with an additional variation in the number of cells per block.

Hindrances to the efficient use of the vector registers are listed next. They include unfavorable loop-nest structures and a lack of exception-free guarantees. The issues are subse-

[1] https://doku.lrz.de/display/PUBLIC/SuperMUC-NG

quently addressed. First, the loops are reorganized, as it was found that pre-computation and storage of intermediate values like the eigenvalues and -vectors is beneficial to the original computation on the fly. Also, array-index offsets stemming from the shift of total cells in a block to internal cells and from the mapping of cell-face fluxes to cell values now heavily use `constexpr`-features such that the compiler recognizes the constancy of array access patterns. Finally, potential exception throwing functions have been reformulated and marked `noexcept`. No non-standard intrinsic or even `#pragmas` have been employed.

The improved code version is then compared against the baseline on three 3D test cases: A Sod shock tube [206], an RTI [202] and the diffusion of a Gaussian pulse. Effective resolutions between the cases increase from one hundred thousand to one billion cells, respectively. It was found, that all cases benefit from the code changes. The hardware with the wider vector registers shows higher performance gains as do the configurations with more cells per block. As desired, all employed combinations of Riemann solver and reconstruction stencil benefit equally. Additional tuning of compiler flags shows to be only beneficial on the hardware with the wider vector registers. Yet, this compiler-flag optimization resulted in numerical symmetry breaking and reduces the program's portability.

My contributions to the work were the initial initiative to address vectorization hindrances by portable standard-compliant source code modifications. In close collaboration with the second author, I performed the profiles and benchmarks. The new code blocks are provided by me as well as the setup of the presented test cases. I was responsible for the creation of the visualizations and tables. I have predominantly written the final manuscripts and incorporated feedback from the reviewers. My co-authors have coordinated, supervised, and secured funding for the project. They have also reviewed the manuscript and responses to the reviewers.

3.3 Large Scale Sharp-interface Simulations

The simulation of multiphase flows requires additional mathematical and numerical modeling to represent the fluid interface(s), cf. section 2.1.1. The LS with its sharp-interface representation is a natural choice for this discretization, in particular for interface-driven flows and the investigation of interface instabilities. However, to the best of the author's knowledge, no simulation software exists that provides the LS algorithm within a parallelized framework that is capable to provide the needed resolution for 3D simulations.

This limitation is overcome by inserting the LS algorithm into the MR compressible CFD framework ALPACA [3]. Thereby, two key challenges for efficiently running the large scale LS simulations are addressed. First, in order to always fully refine the interface region, the narrowband [2, 39, 171] approach is picked-up on by defining interface tags representing the band structure alongside with prediction and averaging operators. This way the MR algorithm can react, i. e. refine or coarsen to approaching or vanishing interfaces, respectively. Without this tagging approach, the MR algorithm would coarsen on smooth $\phi(\mathbf{x}, t)$ and $\mathbf{U}(\mathbf{x}, t)$ as e. g. present in a pure advection case. Second, since the GFM requires information from both phases at both sides of the interface, the storage must be managed. Each phase is stored in a separate block within the MR node. The block structure stays unchanged to section 3.1. However, nodes close to the interface allocate a second block allowing the (unchanged) solver to process both blocks independently. The implemented interface interaction allows for a straightforward representation of complex solid geometries or free surfaces by the LS. For these purposes, simply the two-material interface-interaction Riemann solver needs to be modified, cf. section 2.2. The capabilities of the implementation have been demonstrated on representable large scale two-phase computations. Details of these

improvements are published in [106] and summarized below.

3.3.1 Summary of: "ALPACA - a level-set based sharp-interface multiresolution solver for conservation laws"

In this paper, the adaptation of the LS method into the block-based MR algorithm is presented. In particular, a tagging system is proposed to replicate the narrowband approach [2, 39, 171] across all levels of refinement. The modular concept of the single-phase implementation is extended by the LS building-blocks, cf. section 2.1.1. In addition, competing schemes for the discretization of interface interaction, the GFM extrapolation and the reinitalization are provided.

At first, an overview of applications for compressible flows, the variety of numerical schemes, and the competing interface and compression algorithms is given. The challenges to efficiently run these algorithms and schemes on HPC clusters are stated. Furthermore, comments on general code features concerning the build process, test coverage, and input/output (I/O) are made. The mathematical formulation to extend the single-phase NSE to multiple phases using the LS method is stated alongside its numerical discretization. Therein, the LS re-initialization procedure, the GFM extrapolation, and the two-material Riemann-problem based interface interaction terms are highlighted. Further remarks are made concerning time-step size restrictions and the handling of marginally filled cut cells.

Then, the modular block-based MR algorithm is reviewed before the newly proposed interface tagging system is introduced. Its purpose is three-fold. First, it gives a memory-saving representation of the interface structure across all refinement levels, allowing the MR to respectively refine or coarsen on approaching or vanishing interfaces. Second, it allows limiting allocation of multiple fluid blocks to the narrowband region on all levels. Finally, it simplifies the restriction of LS related computations to the narrowband region. The algorithmic extension compared to the single-phase block-based MR algorithm are given alongside parallelization and implementation details, which enable the modular setup of the solver.

Next, a wide variety of test cases is simulated. First, the convergence order of the employed schemes is verified together with the correctness of the surface tension and viscous forces' implementation. Subsequently, complex shock-bubble configurations with 'heavy' and 'light' bubbles are analyzed. Therein, complex material properties and high resolutions of up to four billion effective cells are used. In the following, the parallel performance of the multiphase implementation is evaluated on up to twenty-four thousand cores and shows high efficiencies across the different chosen compute kernels. The achieved compression over all conducted 3D simulations is determined. Four compression metrics have been proposed, revealing savings in terms of compute time and memory consumption. They are defined such that the MR compression and the enhanced narrowband-related compression are separately evaluable. High compression values are observed if the MR is allowed to provide high numbers of refinement levels.

For this publication, I have implemented the parallel framework and the interface tagging mechanism. I have conducted the convergence study, the scaling runs, and have defined and evaluated the compression rates. I have predominantly written sections 4 to 7, 8.1, 9 and 10. Furthermore, I have equally contributed to sections 1 and 11 as well as to the abstract and program summary. I have extensively reviewed the remaining sections. The visualizations in the respective sections were created by me. My first co-author has conducted the works presented in the remaining sections, which he has predominantly written and has equally contributed to sections 1 and 11 as well as to the abstract and program summary. He has extensively reviewed the sections written by me. Reviewer feedback was handled by both of us in a fair and equal manner. The remaining co-authors have coordinated, supervised, and

secured funding for the project. They have also reviewed the manuscript and responses to the reviewers.

3.4 Modular Framework

The presented software ALPACA [3] allows comparisons of competing numerical schemes for the simulation of compressible multiphase flows in high resolution and is available under open-source license. The clean code structure, the automated software testing, and the focus on high-level optimizations ease integration of new schemes and reduce development times. Therefore, data storage and data modification are strictly separated from data manipulation routines. Performance relevant settings such as e. g. the number of cells per block, the Riemann solver, the reconstruction stencil, etc. are determined at compile time to leverage maximum compiler support. Modern C++ 20 features are used. So do for example `if constexpr` clauses accompany the heavily used curiously recurring template pattern (CRTP) for adaptation of different numerical methods to the algorithmic API. Further settings e. g. material parameters, domain sizes, and boundary condition are choosable at runtime. Therein, the concept stays open for extension. We demonstrate the effectiveness of this approach in the before mentioned publications [103, 104, 106]. Therein, the scaling tests have always been conducted with different combinations of numerical schemes and settings and meaningful physical examples have been studied. Even exchanging schemes with different nominal convergence order yields similar parallel performance. The code setup even allows exchanging of the underlying system of equations. E. g. the five-equation NSE may be replaced by their four-equation isentropic counterpart. In this way, also the diffusive interface model is integrated by adding eq. (2.15) [3]. This modularity concept has already enabled a vast variety of research conducted by other authors [24, 63–66, 124–128, 241, 242].

Chapter 4

List of Peer-Reviewed Publications

4.1 Publications comprising this Thesis

- **Nils Hoppe, Stefan Adami, Nikolaus A. Adams**: A parallel modular computing environment for three-dimensional multiresolution simulations of compressible flows. Computer Methods in Applied Mechanics and Engineering, 391, 114486, 2022 [103].
- **Nils Hoppe, Josef M. Winter, Stefan Adami, Nikolaus A. Adams**: ALPACA — a level-set based sharp-interface multiresolution solver for conservation laws. Computer Physics Communications, 272, 108246, 2022 [106].
- **Nils Hoppe, Igor Pasichnyk, Momme Allalen, Stefan Adami, Nikolaus. A. Adams**: Node-Level optimization of a 3D Block-Based Multiresolution Compressible Flow Solver with Emphasis on Performance Portability, 2019 International Conference on High Performance Computing & Simulation (HPCS), pp. 732-740, 2019 [104].

4.2 Further Publications

- **Jakob W. J. Kaiser, Nils Hoppe, Stefan Adami, Nikolaus A. Adams**: An adaptive local time-stepping scheme for multiresolution simulations of hyperbolic conservation laws. Journal of Computational Physics: X, Volume 4, 100038, 2019 [128].
- **Benedikt Biller, Nils Hoppe, Stefan Adami, Nikolaus A. Adams**: Jetting mechanisms in bubble-pair interactions [20].

Chapter 5

Discussion and Concluding Remarks

Numerical investigation of compressible multiphase flows poses multiple challenges. Firstly, the large range of physically relevant scales requires high resolutions in particular for realistic 3D problem setups. Secondly, the representation of the interface and the interaction between the fluids adds additional complexity. Furthermore, sophisticated high-order methods are required to avoid unphysical effects while recovering most flow details. A wide variety of these numerical methods exist whose strengths and drawbacks are problem dependent.

The work presented in this thesis addresses these problems. It presents a block-based MR algorithm with ALTS for efficient computations on HPC systems. The sharp-interface LS method has been adjusted to leverage this block-based MR algorithm. The algorithm has been implemented in the open-source framework ALPACA written in C++ 20 which offers a modular structure to exchange the (high-order) numerical methods without loss of parallel performance. Even a seamless exchange of the underlying set of equations is possible, e. g. changing to the four-equation isentropic NSE or including additional advection equations for integration of diffusive interface models [3].

The presented MR algorithm follows the design of the single-scale MR algorithm [18, 19, 41, 95, 189]. In addition, however, it leverages the parallel capabilities of block-based approaches within the AMR [61] and wavelet-collocation [188] context. The scheme allows for straightforward integration of ALTS allowing for larger time-step sizes than in previously presented LTS [51] approaches.

For efficient domain decomposition with respect to the ALTS and MR requirements, a level-wise modification of well-established SFCs [10] is proposed. Therein, the position on a Hilbert [100] or Lebesque curve [139] can be computed directly from the introduced node IDs, which are generated from a modified Morton Order [157]. The ID mechanism allows neighbor lookup based on bit-shift operations. Given 64 bit per ID the same amount of refinement levels as in [48] is achievable. In contrast to many published MR schemes [48, 51, 189, 209], however, the presented ID scheme allows representation of non-cubic domains.

The LS sharp-interface method [47, 167, 213] is integrated into the block-based MR algorithm. Therefore, the efficient narrowband approach [2, 39, 171] is implemented based on an interface-tagging approach. On the finest resolution level, the original narrowband technique is used. In addition, the bands are represented via interface tags. By defining an average operator for interface tags, the evolution of the interface is propagated through the octree structure. This allows the tree to adjust for approaching or vanishing interfaces by refining or coarsening the appropriate leafs, respectively. In the LS approach the GFM method [59] is used to obtain conservative interface interactions [107, 109, 144, 221]. This method relies on information of all fluids in every cell close to the interface. Hence, memory for each fluid needs to be allocated. Here, the interface tagging mechanism is used again, allowing to invest this additional memory only close to the interface. In particular for 3D simulations, the overall memory-usage is drastically reduced over the implementation, as presented by Han et al. [90].

Highly resolved physical simulations on shock-bubble, shock-droplet, and bubble collapse

scenarions have been carried-out with the presented framework within this thesis [106]. The code has additionally been applied for further studies by other authors [64, 124, 126, 241]. The implemented parallelization strategy of the LS and MR algorithm allowed higher resolved simulations, larger time steps employing higher-order methods than presented in previous publications [52, 152]. The presented resolutions are comparable with large petascale simulations conducted on homogenous meshes [180, 238].

The simulation framework ALPACA developed within this thesis is offered under open-source license for re-use by the scientific community and to foster exchange within the field. It enables interested researchers to jump-start their own simulation work at a very high level in terms of computational efficiency on massively parallel architectures and accessible model complexity. The provided range of high-order numerical schemes for compressible multiphase flows is unmatched by commercial codes based on FVM [7, 200] and finite-element method (FEM) [42]. Similarly, the semicommercial open-source OpenFOAM solver provides only low-order discretization diffusive interface schemes [85]. However, its wide use in academia, results in frequent publication of additional libraries. E. g., a third-order WENO [74] and incompressible LS approaches [50, 145] have been implemented, but not yet reached the main releases of the framework.

Turning to research codes, a wide variety of parallelized compressible multiphase flow solvers exists. Solvers based on FEM or FVM, with different interface representations, with AMR or MR mesh compression, with low- or high-order methods can be found. Yet, no relevant parallel framework was found employing LTS. Reinarz et al. [182] provide a high-order discontinuous Galerkin framework with high parallel performance on $\mathcal{O}\left(10^4\right)$ cores [192]. It uses VOF for the interface representation and AMR for mesh compression and is available under open-source license. Similarly, the open-source Cubism(-Nova) project [88, 99, 239] also uses VOF for the interface representation and is based on a high-order FVM. Although, the code base provides AMR mesh compression the published computations [180, 211, 234, 238] ran on homogenous (stretched) meshes. It has won the Gordon Bell Price 2013 with its 11 PFLOP s^{-1} simulation on over one million compute cores [187]. Likewise, the MFC codebase [33] provides a high-order FVM implementation with a diffusive interface model on homogenous (stretched) meshes under open-source license. High parallel efficiency is recorded for scalings in $\mathcal{O}\left(10^3\right)$ cores. Another open-source framework running high-order FVM with diffusive interface model on homogenous meshes is UCNS3D [8, 177, 226]. The ECOGEN solver [198] is also built on a FVM but provides at most second-order methods. It does, however, include AMR mesh compression. Scalings are presented in $\mathcal{O}\left(10^3\right)$ and $\mathcal{O}\left(10^2\right)$ cores for calculations on homogenous and AMR meshes, respectively. Numerous computations using the sharp interface level-set method for computation of compressible multiphase flows exist: With AMR [162] or MR [90] mesh compression as well as on homogenous meshes [76, 140]. However, none of the underlying code bases is freely available and details on their parallel performance are incomplete or missing completely. A handful of codes using MR mesh compression report their parallel performance solving the compressible NSE. E. g., Soni et al. 's [207] implementation includes high-order methods and shows scalings in $\mathcal{O}\left(10^2\right)$ cores. Sroka et al. [209] and Descombes et al. [48] release their respective low-order method codes under open-source license and present scalings in $\mathcal{O}\left(10^2\right)$ cores.

The presented open-source C++20 codebase [3, 103, 104, 106] is hence unique as it provides a modular FVM framework for compressible multiphase simulations with high-order methods, the level-set method, MR mesh and ALTS temporal compression. Scalings in $\mathcal{O}\left(10^4\right)$ cores conducted with temporal and spatial adaptivity activated show comparable parallel performance across methods of varying nominal convergence order. The modular structure further allows substitution of the LS interface representation with a diffusive or VOF-type representations.

In conclusion, the continuing growth of compute power benefits the usage of DNS simulations for the study of compressible multiphase flows. Nevertheless, fully resolved 3D simulations will require compression algorithms and efficiently used HPC infrastructure for years to come. The parallelization strategy presented in this thesis and its implementation in ALPACA combines these demands with parallelization-optimized MR, ALTS and level-set algorithms. The capabilities of the presented strategy and implementation have been shown in a range of simulations of physically challenging problems with high (effective) resolution. Although successful, the current MPI-only parallelization may be improved upon by introducing a hybrid parallelization with an additional threading concept. This could reduce the domain-partitioning overhead for multi-node and pure shared-memory computations. The block-based structure of the presented algorithms allows adaptation of the approach by Wermelinger et al. [239], which may also help to exploit the heterogeneous hardware of newer supercomputers.

Bibliography

[1] Abgrall, R. “How to Prevent Pressure Oscillations in Multicomponent Flow Calculations: A Quasi Conservative Approach”. In: *Journal of Computational Physics* 125 (1996), pp. 150–160. DOI: 10.1006/jcph.1996.0085.

[2] Adalsteinsson, D. and Sethian, J. A. “A Fast Level Set Method for Propagating Interfaces”. In: *Journal of Computational Physics* 118.2 (1995), pp. 269–277. ISSN: 0021-9991. DOI: 10.1006/jcph.1995.1098.

[3] Adams, N. A., Adami, S., Bogdanov, V., Buhendwa, A., Bußmann, A., Fleischmann, N., Hoppe, N., Hosseini, N., Kaiser, J., Lunkov, A., Paula, T., Spaeth, F., Siguenza Torres, A., Wauligmann, P., Winter, J., and Gymnich, T. *ALPACA - Adaptive Level-set PArallel Code Alpaca*. en. Research Code. 2022. DOI: 10.14459/2022mp1647482.

[4] Allaire, G., Clerc, S., and Kokh, S. “A Five-Equation Model for the Simulation of Interfaces between Compressible Fluids”. In: *Journal of Computational Physics* 181.2 (2002), pp. 577–616. ISSN: 0021-9991. DOI: 10.1006/jcph.2002.7143.

[5] Amarala, S. and Wan, J. W. L. “Multigrid Methods for Systems of Hyperbolic Conservation Laws”. In: *Multiscale Modeling & Simulation* 11.2 (2013), pp. 586–614. DOI: 10.1137/110851316.

[6] Andrianov, N., Saurel, R., and Warnecke, G. “A simple method for compressible multiphase mixtures and interfaces”. In: *International Journal for Numerical Methods in Fluids* 41.2 (2003), pp. 109–131. DOI: 10.1002/fld.424.

[7] Ansys, I. *ANSYS fluent theory guide, release 2020 R1*. 2020.

[8] Antoniadis, A. F., Tsoutsanis, P., Rana, Z., Kokkinakis, I., and Drikakis, D. “Azure: An Advanced CFD Software Suite Based on High-Resolution and High-Order Methods”. In: *53rd AIAA Aerospace Sciences Meeting*. 2015. DOI: 10.2514/6.2015-0813.

[9] Aulisa, E., Manservisi, S., Scardovelli, R., and Zaleski, S. “Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry”. In: *Journal of Computational Physics* 225.2 (2007), pp. 2301–2319. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2007.03.015.

[10] Bader, M. *Space-Filling Curves*. Vol. 9. Texts in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-31045-4. DOI: 10.1007/978-3-642-31046-1.

[11] Ball, G., Howell, B., Leighton, T., and Schofield, M. “Shock-induced collapse of a cylindrical air cavity in water: a free-Lagrange simulation”. In: *Shock Waves* 10.4 (2000), pp. 265–276. DOI: 10.1007/s001930000060.

[12] Balsara, D. S., Garain, S., and Shu, C.-W. “An efficient class of WENO schemes with adaptive order”. In: *Journal of Computational Physics* 326 (Dec. 2016), pp. 780–804. ISSN: 00219991. DOI: 10.1016/j.jcp.2016.09.009.

[13] Balsara, D. S. and Shu, C.-W. “Monotonicity Preserving Weighted Essentially Non-oscillatory Schemes with Increasingly High Order of Accuracy”. In: *Journal of Computational Physics* 160.2 (2000), pp. 405–452. ISSN: 0021-9991. DOI: 10.1006/jcph.2000.6443.

[14] Banchelli, F., Garcia-Gasulla, M., Houzeaux, G., and Mantovani, F. “Benchmarking of State-of-the-Art HPC Clusters with a Production CFD Code”. In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC ’20. Geneva, Switzerland: Association for Computing Machinery, 2020. ISBN: 9781450379939. DOI: 10.1145/3394277.3401847.

[15] Batten, P., Clarke, N., Lambert, C., and Causon, D. M. “On the Choice of Wavespeeds for the HLLC Riemann Solver”. In: *SIAM Journal on Scientific Computing* 18.6 (1997), pp. 1553–1570. DOI: 10.1137/S1064827593260140.

[16] Bell, J. B., Colella, P., and Trangenstein, J. A. “Higher order Godunov methods for general systems of hyperbolic conservation laws”. In: *Journal of Computational Physics* 82.2 (1989), pp. 362–397. ISSN: 0021-9991. DOI: 10.1016/0021-9991(89)90054-5.

[17] Berger, M. and Colella, P. “Local adaptive mesh refinement for shock hydrodynamics”. In: *Journal of Computational Physics* 82.1 (1989), pp. 64–84. ISSN: 0021-9991. DOI: 10.1016/0021-9991(89)90035-1.

[18] Bihari, B. L. “Multiresolution schemes for conservation laws with viscosity”. In: *Journal of Computational Physics* 123.1 (Jan. 1996), pp. 207–225. ISSN: 00219991. DOI: 10.1006/jcph.1996.0017.

[19] Bihari, B. L. and Harten, A. “Multiresolution Schemes for the Numerical Solution of 2-D Conservation Laws I”. In: *SIAM Journal on Scientific Computing* 18.2 (1997), pp. 315–354. DOI: 10.1137/S1064827594278848.

[20] Biller, B., Hoppe, N., Adami, S., and Adams, N. A. “Jetting mechanisms in bubble-pair interactions”. In: *Physics of Fluids* 34.7 (2022), p. 076111. DOI: 10.1063/5.0097039.

[21] Birkhoff, G. and Zarantonello, E. “Jets, wakes and cavities”. In: (Jan. 1957).

[22] Bissi, W., Serra Seca Neto, A. G., and Emer, M. C. F. P. “The effects of test driven development on internal quality, external quality and productivity: A systematic review”. In: *Information and Software Technology* 74 (2016), pp. 45–54. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2016.02.004.

[23] Bode, M., Lehn, F. vom, Maeda, K., Colonius, T., and Pitsch, H. “Numerical investigation of the effect of dissolved non-condensable gases on hydraulic flip in cavitating nozzles”. In: *10th International Cavitation Symposium, Baltimore, USA*. 2018.

[24] Bogdanov, V., Schranner, F. S., Winter, J. M., Adami, S., and Adams, N. A. “A level-set-based sharp-interface method for moving contact lines”. In: *Journal of Computational Physics* 467 (2022), p. 111445. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111445.

[25] Bonachea, D. and Hargrove, P. H. “GASNet-EX: A High-Performance, Portable Communication Library for Exascale”. In: *Languages and Compilers for Parallel Computing*. Ed. by Hall, M. and Sundar, H. Cham: Springer International Publishing, 2019, pp. 138–158. ISBN: 978-3-030-34627-0. DOI: 10.1007/978-3-030-34627-0_11.

[26] Borges, R., Carmona, M., Costa, B., and Don, W. S. “An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws”. In: *Journal of Computational Physics* 227.6 (2008), pp. 3191–3211. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2007.11.038.

[27] Borkar, S. “Design challenges of technology scaling”. In: *IEEE Micro* 19.4 (1999), pp. 23–29. DOI: 10.1109/40.782564.

[28] Borkar, S. and Chien, A. A. “The future of microprocessors”. In: *Communications of the ACM* 54.5 (2011), pp. 67–77. DOI: {10.1145%2F1941487.1941507}.

[29] Brackbill, J., Kothe, D., and Zemach, C. “A continuum method for modeling surface tension”. In: *Journal of Computational Physics* 100.2 (1992), pp. 335–354. ISSN: 0021-9991. DOI: 10.1016/0021-9991(92)90240-Y.

[30] Brennen, C. E. “Cavitation in medicine”. In: *Interface Focus* 5.5 (2015), p. 20150022. DOI: 10.1098/rsfs.2015.0022.

[31] Brix, K., Melian, S., Müller, S., and Bachmann, M. “Adaptive Multiresolution Methods: Practical issues on Data Structures, Implementation and Parallelization”. In: *ESAIM: Proceedings* 34 (Dec. 2011), pp. 151–183. ISSN: 1270-900X. DOI: 10.1051/proc/201134003.

[32] Brujan, E.-A., Noda, T., Ishigami, A., Ogasawara, T., and Takahira, H. “Dynamics of laser-induced cavitation bubbles near two perpendicular rigid walls”. In: *Journal of Fluid Mechanics* 841 (2018), pp. 28–49. DOI: 10.1017/jfm.2018.82.

[33] Bryngelson, S. H., Schmidmayer, K., Coralic, V., Meng, J. C., Maeda, K., and Colonius, T. “MFC: An open-source high-order multi-component, multi-phase, and multi-scale compressible flow solver”. In: *Computer Physics Communications* 266 (2021), p. 107396. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2020.107396.

[34] Budich, B., Schmidt, S. J., and Adams, N. A. “Numerical simulation of cavitating ship propeller flow and assessment of erosion aggressiveness”. In: *MARINE VI: proceedings of the VI International Conference on Computational Methods in Marine Engineering*. CIMNE. 2015, pp. 709–721. ISBN: 978-84-943928-6-3.

[35] Chandrasekhar, S. *Hydrodynamic and hydromagnetic stability*. 1961. ISBN: 0-486-64071-X.

[36] Chauvin, A., Daniel, E., Chinnayya, A., Massoni, J., and Jourdan, G. “Shock waves in sprays: numerical study of secondary atomization and experimental comparison”. In: *Shock waves* 26.4 (2015), pp. 403–415. DOI: 10.1007/s00193-015-0593-0.

[37] Chauvin, A., Jourdan, G., Daniel, E., Houas, L., and Tosello, R. “Experimental investigation of the propagation of a planar shock wave through a two-phase gas-liquid medium”. In: *Physics of Fluids* 23.11 (2011), p. 113301. DOI: 10.1063/1.3657083.

[38] Chiapolino, A., Boivin, P., and Saurel, R. “A simple phase transition relaxation solver for liquid–vapor flows”. In: *International Journal for Numerical Methods in Fluids* 83.7 (2017), pp. 583–605. DOI: 10.1002/fld.4282.

[39] Chopp, D. L. “Computing Minimal Surfaces via Level Set Curvature Flow”. In: *Journal of Computational Physics* 106.1 (1993), pp. 77–91. ISSN: 0021-9991. DOI: 10.1006/jcph.1993.1092.

[40] Cockburn, B., Lin, S.-Y., and Shu, C.-W. “TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems”. In: *Journal of Computational Physics* 84.1 (1989), pp. 90–113. ISSN: 0021-9991. DOI: 10.1016/0021-9991(89)90183-6.

[41] Cohen, A., Kaber, S., Müller, S., and Postel, M. “Fully adaptive multiresolution finite volume schemes for conservation laws”. In: *Mathematics of Computation* 72.241 (2003), pp. 183–225. DOI: 10.1090/S0025-5718-01-01391-6.

[42] COMSOL, A. *CFD Module User’s Guide*. 2020.

[43] Corporation, I. *Intel MPI Library Developer Reference for Linux OS - Shared Memory Control*. URL: https://www.intel.com/content/www/us/en/develop/documentation/mpi-developer-reference-linux/top/environment-variable-reference/environment-variables-for-fabrics-control/shared-memory-control.html (visited on 05/26/2022).

[44] Davis, S. F. "Simplified Second-Order Godunov-Type Methods". In: *SIAM Journal on Scientific and Statistical Computing* 9.3 (1988), pp. 445–473. DOI: 10.1137/0909030.

[45] Deiterding, R., Domingues, M. O., Gomes, S. M., and Schneider, K. "Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible Euler equations". In: *SIAM Journal on Scientific Computing* 38.5 (2016), S173–S193. DOI: 10.1137/15M1026043.

[46] Deiterding, R., Domingues, M. O., and Schneider, K. "Multiresolution analysis as a criterion for effective dynamic mesh adaptation–A case study for Euler equations in the SAMR framework AMROC". In: *Computers & Fluids* (2020), p. 104583. DOI: 10.1016/j.compfluid.2020.104583.

[47] Dervieux, A. and Thomasset, F. "A finite element method for the simulation of Raleigh-Taylor instability". In: vol. 771. Nov. 1981, pp. 145–158. ISBN: 978-3-540-09734-1. DOI: 10.1007/BFb0086904.

[48] Descombes, S., Duarte, M., Dumont, T., Guillet, T., Louvet, V., and Massot, M. "Task-based adaptive multiresolution for time-space multi-scale reaction-diffusion systems on multi-core architectures". In: *The SMAI journal of computational mathematics* 3 (2017), pp. 29–51. DOI: 10.5802/smai-jcm.19.

[49] Devassy, B. M., Habchi, C., and Daniel, E. "Atomization modelling of liquid jets using a two-surface-density approach". In: *Atomization and Sprays* 25.1 (2015). DOI: 10.1615/AtomizSpr.2014011350.

[50] Dianat, M., Skarysz, M., and Garmory, A. "A Coupled Level Set and Volume of Fluid method for automotive exterior water management applications". In: *International Journal of Multiphase Flow* 91 (2017), pp. 19–38. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2017.01.008.

[51] Domingues, M. O., Gomes, S. M., Roussel, O., and Schneider, K. "An adaptive multiresolution scheme with local time stepping for evolutionary PDEs". In: *Journal of Computational Physics* 227.8 (2008), pp. 3758–3780. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2007.11.046.

[52] Dorschner, B., Biasiori-Poulanges, L., Schmidmayer, K., El-Rabii, H., and Colonius, T. "On the formation and recurrent shedding of ligaments in droplet aerobreakup". In: *Journal of Fluid Mechanics* 904 (2020), A20. DOI: 10.1017/jfm.2020.699.

[53] Einfeldt, B. "On Godunov-Type Methods for Gas Dynamics". In: *SIAM Journal on Numerical Analysis* 25.2 (1988), pp. 294–318. DOI: 10.1137/0725021.

[54] Engel, O. G. "Fragmentation of Waterdrops in the Zone". In: *Journal of Research of the National Bureau of Standards* 60.3 (1958), p. 245.

[55] Estrada, J. B., Barajas, C., Henann, D. L., Johnsen, E., and Franck, C. "High strain-rate soft material characterization via inertial cavitation". In: *Journal of the Mechanics and Physics of Solids* 112 (2018), pp. 291–317. ISSN: 0022-5096. DOI: 10.1016/j.jmps.2017.12.006.

[56] Euler, L. *Institutiones calculi integralis*. Vol. 4. Academia Imperialis Scientiarum, 1794.

[57] Favrie, N., Gavrilyuk, S., and Saurel, R. "Solid-fluid diffuse interface model in cases of extreme deformations". In: *Journal of Computational Physics* 228.16 (2009), pp. 6037–6077. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2009.05.015.

[58] Feathers, M. C. *Working effectively with legacy code*. Prentice Hall PTR, 2004, p. 434. ISBN: 9780131177055.

[59] Fedkiw, R. P., Aslam, T., Merriman, B., and Osher, S. "A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)". In: *Journal of Computational Physics* 152.2 (1999), pp. 457–492. ISSN: 0021-9991. DOI: 10.1006/jcph.1999.6236.

[60] Fedorenko, R. "The speed of convergence of one iterative process". In: *USSR Computational Mathematics and Mathematical Physics* 4.3 (1964), pp. 227–235. ISSN: 0041-5553. DOI: 10.1016/0041-5553(64)90253-8.

[61] Ferreira, C. R. and Bader, M. "Load Balancing and Patch-Based Parallel Adaptive Mesh Refinement for Tsunami Simulation on Heterogeneous Platforms Using Xeon Phi Coprocessors". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '17. Lugano, Switzerland: Association for Computing Machinery, 2017. ISBN: 9781450350624. DOI: 10.1145/3093172.3093237.

[62] Fisher, T. C. and Carpenter, M. H. "High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains". In: *Journal of Computational Physics* 252 (2013), pp. 518–557. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2013.06.014.

[63] Fleischmann, N., Adami, S., and Adams, N. "Numerical symmetry-preserving techniques for low-dissipation shock-capturing schemes". In: *Computers & Fluids* 189 (2019), pp. 94–107. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2019.04.004.

[64] Fleischmann, N., Adami, S., and Adams, N. A. "A shock-stable modification of the HLLC Riemann solver with reduced numerical dissipation". In: *Journal of Computational Physics* 423 (2020), p. 109762. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109762.

[65] Fleischmann, N., Adami, S., and Adams, N. A. "On an inconsistency of the arithmetic-average signal speed estimate for HLL-type Riemann solvers". In: *Journal of Computational Physics: X* 8 (2020), p. 100077. ISSN: 2590-0552. DOI: 10.1016/j.jcpx.2020.100077.

[66] Fleischmann, N., Adami, S., Hu, X. Y., and Adams, N. A. "A low dissipation method to cure the grid-aligned shock instability". In: *Journal of Computational Physics* 401 (2020), p. 109004. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.109004.

[67] Forum, M. P. I. *MPI: A Message-Passing Interface Standard Version 3.1*. June 2015. URL: https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf.

[68] Foundation, X. *The FLAC format*. 2022. URL: https://xiph.org/flac/format.html (visited on 05/01/2022).

[69] Fowler, M., Beck, K., Brant, J., Opdyke, W., and Roberts, D. *Refactoring: Improving the Design of Existing Code*. 1st ed. Berkeley, CA, USA: Addison-Wesley Professional, 1999. ISBN: 978-0-201-48567-7.

[70] Fu, L., Hu, X. Y., and Adams, N. A. "A family of high-order targeted ENO schemes for compressible-fluid simulations". en. In: *Journal of Computational Physics* 305 (Jan. 2016), pp. 333–359. ISSN: 10902716. DOI: 10.1016/j.jcp.2015.10.037.

[71] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design patterns : elements of reusable object-oriented software*. 35. print. Addison-Wesley, 1995, p. 395. ISBN: 0201633612.

[72] Gande, N. R. and Bhise, A. A. "Modified third and fifth order WENO schemes for inviscid compressible flows". In: *Numerical Algorithms* (2020), pp. 1–31. DOI: 10.1007/s11075-020-01039-9.

[73] Garnier, E., Adams, N., and Sagaut, P. *Large eddy simulation for compressible flows*. Springer Science & Business Media, 2009.

[74] Gärtner, J. W., Kronenburg, A., and Martin, T. "Efficient WENO library for OpenFOAM". In: *SoftwareX* 12 (2020), p. 100611. ISSN: 2352-7110. DOI: 10.1016/j.softx.2020.100611.

[75] Gavrilyuk, S., Favrie, N., and Saurel, R. "Modelling wave dynamics of compressible elastic materials". In: *Journal of Computational Physics* 227.5 (2008), pp. 2941–2969. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2007.11.030.

[76] Ghods, S. and Herrmann, M. "A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods". In: *Physica Scripta* T155 (July 2013), p. 014050. DOI: 10.1088/0031-8949/2013/t155/014050.

[77] Gibou, F., Fedkiw, R., and Osher, S. "A review of level-set methods and some recent applications". In: *Journal of Computational Physics* 353 (2018), pp. 82–109. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2017.10.006.

[78] Glimm, J., Grove, J. W., Li, X. L., and Tan, D. C. "Robust Computational Algorithms for Dynamic Interface Tracking in Three Dimensions". In: *SIAM Journal on Scientific Computing* 21.6 (2000), pp. 2240–2256. DOI: 10.1137/S1064827598340500.

[79] Glimm, J., Grove, J. W., Li, X. L., Shyue, K.-m., Zeng, Y., and Zhang, Q. "Three-Dimensional Front Tracking". In: *SIAM J. Sci. Comput.* 19.3 (May 1998), pp. 703–727. ISSN: 1064-8275. DOI: 10.1137/S1064827595293600.

[80] Godunov, S. K. and Bohachevsky, I. "Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics". In: *Matematičeskij sbornik* 47(89).3 (1959), pp. 271–306.

[81] Goodliffe, P. *Code Craft : the Practice of Writing Excellent Code.* No Starch Press, 2006, p. 625. ISBN: 9781593271190.

[82] Gottlieb, S., Shu, C.-W., and Tadmor, E. "Strong Stability-Preserving High-Order Time Discretization Methods". In: *SIAM Review* 43.1 (2001), pp. 89–112. DOI: 10.1137/S003614450036757X.

[83] Grant, R. E. and Afsahi, A. "Characterization of multithreaded scientific workloads on simultaneous multithreading intel processors". In: *Proceedings of the First Annual Workshop on Interaction between Operating System and Computer Architecture (IOSCA 2005)*. 2005.

[84] Green, R. "How to write unmaintainable code". In: *Java Developers' Journal* 2.6 (2001). URL: https://github.com/Droogans/unmaintainable-code.

[85] Greenshields, C. J. *OpenFOAM user guide*. 2021.

[86] Guermond, J.-L. and Popov, B. "Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations". In: *Journal of Computational Physics* 321 (2016), pp. 908–926. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.05.054.

[87] Haas, J.-F. and Sturtevant, B. "Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities". In: *Journal of Fluid Mechanics* 181 (1987), pp. 41–76. DOI: 10.1017/S0022112087002003.

[88] Hadjidoukas, P. E., Rossinelli, D., Hejazialhosseini, B., and Koumoutsakos, P. "From 11 to 14.4 PFLOPs: Performance Optimization for Finite Volume Flow Solver". In: *Proceedings of the 3rd International Conference on Exascale Applications and Software*. EASC '15. Edinburgh, UK: University of Edinburgh, 2015, pp. 7–12. ISBN: 978-0-9926615-1-9.

[89] Hager, G. and Wellein, G. *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press, 2011, p. 349. ISBN: 9781439811924. DOI: 10.1201/EBK1439811924.

[90] Han, L., Hu, X., and Adams, N. "Adaptive multi-resolution method for compressible multi-phase flows with sharp interface model and pyramid data structure". In: *Journal of Computational Physics* 262 (2014), pp. 131–152. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2013.12.061.

[91] Hank, S., Favrie, N., and Massoni, J. "Modeling hyperelasticity in non-equilibrium multiphase flows". In: *Journal of Computational Physics* 330 (2017), pp. 65–91. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.11.001.

[92] Hank, S., Gavrilyuk, S., Favrie, N., and Massoni, J. "Impact simulation by an Eulerian model for interaction of multiple elastic-plastic solids and fluids". In: *International Journal of Impact Engineering* 109 (2017), pp. 104–111. ISSN: 0734-743X. DOI: 10.1016/j.ijimpeng.2017.06.003.

[93] Harlow, F. H. and Amsden, A. A. *Fluid dynamics*. Tech. rep. NULL. Los Alamos National Lab. (LANL), 1971.

[94] Harten, A. "High resolution schemes for hyperbolic conservation laws". In: *Journal of Computational Physics* 49.3 (Mar. 1983), pp. 357–393. ISSN: 00219991. DOI: 10.1016/0021-9991(83)90136-5.

[95] Harten, A. "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws". In: *Communications on Pure and Applied Mathematics* 48.12 (1995), pp. 1305–1342. ISSN: 0010-3640. DOI: 10.1002/cpa.3160481201.

[96] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R. "Uniformly high order accurate essentially non-oscillatory schemes, III". In: *Journal of Computational Physics* 71.2 (1987), pp. 231–303. ISSN: 0021-9991. DOI: 10.1016/0021-9991(87)90031-3.

[97] Harten, A., Lax, P. D., and Leer, B. v. "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws". In: *SIAM Review* 25.1 (1983), pp. 35–61. DOI: 10.1137/1025002.

[98] Hartmann, R. and Houston, P. "Adaptive Discontinuous Galerkin Finite Element Methods for Nonlinear Hyperbolic Conservation Laws". In: *SIAM Journal on Scientific Computing* 24.3 (2003), pp. 979–1004. DOI: 10.1137/S106482750138908 4.

[99] Hejazialhosseini, B., Rossinelli, D., Conti, C., and Koumoutsakos, P. "High throughput software for direct numerical simulations of compressible two-phase flows". In: *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 2012, pp. 1–12. DOI: 10.1109/SC.2012.66.

[100] Hilbert, D. "Über die stetige Abbildung einer Linie auf ein Flächenstück". In: *Dritter Band: Analysis · Grundlagen der Mathematik · Physik Verschiedenes: Nebst Einer Lebensgeschichte*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1891, pp. 1–2. ISBN: 978-3-662-38452-7. DOI: 10.1007/978-3-662-38452-7_1.

[101] Hirt, C. and Nichols, B. "Volume of fluid (VOF) method for the dynamics of free boundaries". In: *Journal of Computational Physics* 39.1 (1981), pp. 201–225. ISSN: 0021-9991. DOI: 10.1016/0021-9991(81)90145-5.

[102] Hopfes, T., Petersen, J., Wang, Z., Giglmaier, M., and Adams, N. A. "Secondary Atomization of Liquid Metal Droplets at Moderate Weber Numbers". In: *International Journal of Multiphase Flow* 143 (2021), p. 103723. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2021.103723.

[103] Hoppe, N., Adami, S., and Adams, N. A. "A parallel modular computing environment for three-dimensional multiresolution simulations of compressible flows". In: *Computer Methods in Applied Mechanics and Engineering* 391 (2022), p. 114486. ISSN: 0045-7825. DOI: 10.1016/j.cma.2021.114486.

[104] Hoppe, N., Adami, S., Adams, N. A., Pasichnyk, I., and Allalen, M. "Node-Level optimization of a 3D Block-Based Multiresolution Compressible Flow Solver with Emphasis on Performance Portability". In: *2019 International Conference on High Performance Computing Simulation (HPCS)*. 2019, pp. 732–740. DOI: 10.1109/HPCS48598.2019.9188088.

[105] Hoppe, N., Pasichnyk, I., Allalen, M., Adami, S., and Adams, N. A. *Node-Level optimization of a 3D Block-Based Multiresolution Compressible Flow Solver with Emphasis on Performance Portability*. Dataset. 2020. DOI: 10.14459/2020mp1578279.

[106] Hoppe, N., Winter, J. M., Adami, S., and Adams, N. A. "ALPACA - a level-set based sharp-interface multiresolution solver for conservation laws". In: *Computer Physics Communications* 272 (2022), p. 108246. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2021.108246.

[107] Hu, X. Y., Adams, N. A., Johnsen, E., and Iaccarino, G. "Modeling full-Mach-range cavitating flow with sharp interface model". In: *Proceedings of the Summer Program*. Center for Turbulence Research, Stanford, California, USA, 2008, p. 183.

[108] Hu, X. Y., Wang, Q., and Adams, N. A. "An adaptive central-upwind weighted essentially non-oscillatory scheme". In: *Journal of Computational Physics* 229.23 (Nov. 2010), pp. 8952–8965. ISSN: 00219991. DOI: 10.1016/j.jcp.2010.08.019.

[109] Hu, X., Khoo, B., Adams, N., and Huang, F. "A conservative interface method for compressible flows". In: *Journal of Computational Physics* 219.2 (2006), pp. 553–578. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2006.04.001.

[110] IEEE, I. o. E. and Electronics Engineers Inc. Staff, C. *IEEE 1003.1c-1995 - Standard for Information Technology–Portable Operating System Interface (POSIX(TM)) - System Application Program Interface (API) Amendment 2: Threads Extension (C Language)*. Tech. rep. IEEE 1003.1c-1995. IEEE Standards Office, 1994.

[111] Igra, D. and Takayama, K. "Numerical simulation of shock wave interaction with a water column". In: *Shock Waves* 11.3 (2001), pp. 219–228. DOI: 10.1007/PL00004077.

[112] *Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines*. Standard. Geneva, CH: International Organization for Standardization, Feb. 1994.

[113] *Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio*. Standard. Geneva, CH: International Organization for Standardization, Apr. 1998.

[114] *Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification*. Standard. Geneva, CH: International Organization for Standardization, Mar. 2004.

[115] ISOCPP. *Cpp Core Guidelines (Github Repository)*. URL: https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines (visited on 05/06/2022).

[116] Itina, T. E. “On Nanoparticle Formation by Laser Ablation in Liquids”. In: *The Journal of Physical Chemistry C* 115.12 (Mar. 2011), pp. 5044–5048. ISSN: 1932-7447, 1932-7455. DOI: 10.1021/jp1090944.

[117] Jiang, G.-S. and Peng, D. “Weighted ENO Schemes for Hamilton–Jacobi Equations”. In: *SIAM Journal on Scientific Computing* 21.6 (2000), pp. 2126–2143. DOI: 10.1137/S106482759732455X.

[118] Jiang, G.-S. and Shu, C.-W. “Efficient Implementation of Weighted ENO Schemes”. In: *Journal of Computational Physics* 126.1 (1996), pp. 202–228. ISSN: 0021-9991. DOI: 10.1006/jcph.1996.0130.

[119] Johnsen, E. and Colonius, T. “Implementation of WENO schemes in compressible multicomponent flow problems”. In: *J. Comput. Phys.* 219.2 (Dec. 2006), pp. 715–732. ISSN: 00219991. DOI: 10.1016/j.jcp.2006.04.018.

[120] Johnsen, E. and Colonius, T. “Numerical simulations of non-spherical bubble collapse”. In: *Journal of Fluid Mechanics* 629 (2009), pp. 231–262. DOI: 10.1017/S0022112009006351.

[121] Johnsen, E. and Ham, F. “Preventing numerical errors generated by interface-capturing schemes in compressible multi-material flows”. In: *Journal of Computational Physics* 231.17 (2012), pp. 5705–5717. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2012.04.048.

[122] Joseph, D., Belanger, J., and Beavers, G. “Breakup of a liquid drop suddenly exposed to a high-speed airstream”. In: *International Journal of Multiphase Flow* 25.6 (1999), pp. 1263–1303. ISSN: 0301-9322. DOI: 10.1016/S0301-9322(99)00043-9.

[123] Kaiser, H., Diehl, P., Lemoine, A. S., Lelbach, B. A., Amini, P., Berge, A., Biddiscombe, J., Brandt, S. R., Gupta, N., Heller, T., et al. “Hpx - The C++ Standard Library for Parallelism and Concurrency”. In: *Journal of Open Source Software* 5.53 (2020), p. 2352. DOI: 10.21105/joss.02352.

[124] Kaiser, J., Adami, S., and Adams, N. “Three-dimensional direct numerical simulation of shock-induced bubble collapse near gelatin”. In: 2019.

[125] Kaiser, J., Adami, S., Akhatov, I., and Adams, N. “A semi-implicit conservative sharp-interface method for liquid-solid phase transition”. In: *International Journal of Heat and Mass Transfer* 155 (2020), p. 119800. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2020.119800.

[126] Kaiser, J., Appel, D., Fritz, F., Adami, S., and Adams, N. “A multiresolution local-timestepping scheme for particle-laden multiphase flow simulations using a level-set and point-particle approach”. In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113966. ISSN: 0045-7825. DOI: 10.1016/j.cma.2021.113966.

[127] Kaiser, J., Winter, J., Adami, S., and Adams, N. “Investigation of interface deformation dynamics during high-Weber number cylindrical droplet breakup”. In: *International Journal of Multiphase Flow* 132 (2020), p. 103409. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2020.103409.

[128] Kaiser, J. W., Hoppe, N., Adami, S., and Adams, N. A. “An adaptive local time-stepping scheme for multiresolution simulations of hyperbolic conservation laws”. In: *Journal of Computational Physics: X* 4 (2019), p. 100038. ISSN: 2590-0552. DOI: 10.1016/j.jcpx.2019.100038.

[129] Kannan, K., Kedelty, D., and Herrmann, M. “An in-cell reconstruction finite volume method for flows of compressible immiscible fluids”. In: *Journal of Computational Physics* 373 (2018), pp. 784–810. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.07.006.

[130] Kapila, A. K., Menikoff, R., Bdzil, J. B., Son, S. F., and Stewart, D. S. "Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations". In: *Physics of Fluids* 13.10 (2001), pp. 3002–3024. DOI: 10.1063/1.1398042.

[131] Khokhlova, T. D., Wang, Y.-N., Simon, J. C., Cunitz, B. W., Starr, F., Paun, M., Crum, L. a., Bailey, M. R., and Khokhlova, V. a. "Ultrasound-guided tissue fractionation by high intensity focused ultrasound in an in vivo porcine liver model." In: *Proceedings of the National Academy of Sciences of the United States of America* 111.22 (2014), pp. 8161–6. ISSN: 1091-6490. DOI: 10.1073/pnas.1318355111.

[132] Kiyama, A., Tagawa, Y., Ando, K., and Kameda, M. "Effects of a water hammer and cavitation on jet formation in a test tube". In: *Journal of Fluid Mechanics* 787 (2016), pp. 224–236. DOI: 10.1017/jfm.2015.690.

[133] Kreider, W., Crum, L. A., Bailey, M. R., and Sapozhnikov, O. A. "Observations of the collapses and rebounds of millimeter-sized lithotripsy bubbles". In: *The Journal of the Acoustical Society of America* 130.5 (2011), pp. 3531–3540. DOI: 10.1121/1.3626157.

[134] Kutta, W. "Beitrag zur näherungsweisen Integration von Differentialgleichungen". In: *Zeit. Math. Physik* 46 (1901), pp. 435–453.

[135] Laksari, K., Sadeghipour, K., and Darvish, K. "Mechanical response of brain tissue under blast loading". In: *Journal of the Mechanical Behavior of Biomedical Materials* 32 (2014), pp. 132–144. ISSN: 1751-6161. DOI: 10.1016/j.jmbbm.2013.12.021.

[136] Langr, J. *Modern C++ programming with test-driven development : code better, sleep better*. Pragmatic Programmers, 2014. ISBN: 9781937785482.

[137] Lauer, E., Hu, X., Hickel, S., and Adams, N. "Numerical modelling and investigation of symmetric and asymmetric cavitation bubble dynamics". In: *Computers & Fluids* 69 (2012), pp. 1–19. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2012.07.020.

[138] Le Martelot, S., Saurel, R., and Nkonga, B. "Towards the direct numerical simulation of nucleate boiling flows". In: *International Journal of Multiphase Flow* 66 (2014), pp. 62–78. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2014.06.010.

[139] Lebesgue, H. *Leçons sur l'intégration et la recherche des fonctions primitives*. Paris: Gauther-Villars, Imprimeur-Libraire, 1904. URL: http://ark.bnf.fr/ark:/12148/cb30759541r.

[140] Lee, J. and Son, G. "A sharp-interface level-set method for compressible bubble growth with phase change". In: *International Communications in Heat and Mass Transfer* 86 (2017), pp. 1–11. ISSN: 0735-1933. DOI: 10.1016/j.icheatmasstransfer.2017.05.016.

[141] LeMartelot, S., Nkonga, B., and Saurel, R. "Liquid and liquid-gas flows at all speeds". In: *Journal of Computational Physics* 255 (2013), pp. 53–82. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2013.08.001.

[142] LeVeque, R. J. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511791253.

[143] Lindau, O. and Lauterborn, W. "Cinematographic observation of the collapse and rebound of a laser-produced cavitation bubble near a wall". In: *Journal of Fluid Mechanics* 479 (2003), pp. 327–348. DOI: 10.1017/S0022112002003695.

[144] Luo, J., Hu, X., and Adams, N. "A conservative sharp interface method for incompressible multiphase flows". In: *Journal of Computational Physics* 284 (2015), pp. 547–565. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2014.12.044.

[145] Lyras, K. G., Hanson, B., Fairweather, M., and Heggs, P. J. “A coupled level set and volume of fluid method with a re-initialisation step suitable for unstructured meshes”. In: *Journal of Computational Physics* 407 (2020), p. 109224. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.109224.

[146] MacCormack, R. W. “The Effect of Viscosity in Hypervelocity Impact Cratering”. In: *AIAA Hypervelocity Impact Conference*. AIAA Paper 69. Cincinnati, Ohio, USA, 1969. DOI: 10.1142/9789812810793_0002.

[147] Maeda, K., Kreider, W., Maxwell, A., Cunitz, B., Colonius, T., and Bailey, M. “Modeling and experimental analysis of acoustic cavitation bubbles for Burst Wave Lithotripsy”. In: *Journal of Physics: Conference Series* 656 (Dec. 2015), p. 012027. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/656/1/012027.

[148] Martin, R. C. *The clean coder: a code of conduct for professional programmers*. 1. print. Prentice Hall, 2011, p. 210. ISBN: 9780137081073.

[149] Martin, R. C. *Clean Code: a handbook of agile software craftsmanship*. Vol. 53. Prentice Hall, 2013, pp. 1689–1699. ISBN: 9780789749802.

[150] Matula, T. J., Hilmo, P. R., Storey, B. D., and Szeri, A. J. “Radial response of individual bubbles subjected to shock wave lithotripsy pulses in vitro”. In: *Physics of Fluids* 14.3 (2002), pp. 913–921. DOI: 10.1063/1.1433970.

[151] Meng, J. and Colonius, T. “Numerical simulations of the early stages of high-speed droplet breakup”. In: *Shock Waves* 25.4 (2015), pp. 399–414. DOI: 10.1007/s00193-014-0546-z.

[152] Meng, J. C. and Colonius, T. “Numerical simulation of the aerobreakup of a water droplet”. In: *Journal of Fluid Mechanics* 835 (2018), pp. 1108–1135. DOI: 10.1017/jfm.2017.804.

[153] Messina, G. C., Wagener, P., Streubel, R., De Giacomo, A., Santagata, A., Compagnini, G., and Barcikowski, S. “Pulsed laser ablation of a continuously-fed wire in liquid flow for high-yield production of silver nanoparticles”. In: *Phys. Chem. Chem. Phys.* 15.9 (2013), pp. 3093–3098. ISSN: 1463-9076, 1463-9084. DOI: 10.1039/C2CP42626A.

[154] Meyer, M., Devesa, A., Hickel, S., Hu, X., and Adams, N. “A conservative immersed interface method for Large-Eddy Simulation of incompressible flows”. In: *Journal of Computational Physics* 229.18 (2010), pp. 6300–6317. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2010.04.040.

[155] Meyers, S. *Effective modern C++: 42 specific ways to improve your use of C++11 and C++14*. 1. ed. O’Reilly, 2015, p. 315. ISBN: 9781491903995.

[156] Min, C. “On reinitializing level set functions”. In: *Journal of Computational Physics* 229.8 (2010), pp. 2764–2772. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2009.12.032.

[157] Morton, G. M. *A computer oriented geodetic data base and a new technique in file sequencing*. Tech. rep. Ottawa, Canada: International Business Machines Company New York, 1966.

[158] Murman, S. M., Burgess, N. K., Diosady, L. T., and Garai, A. “A DGSEM Shock-capturing Scheme for Scale-resolving Simulations”. In: *23rd AIAA Computational Fluid Dynamics Conference*. DOI: 10.2514/6.2017-4106.

[159] Ndanou, S., Favrie, N., and Gavrilyuk, S. “Criterion of hyperbolicity in hyperelasticity in the case of the stored energy in separable form”. In: *Journal of Elasticity* 115.1 (2014), pp. 1–25. DOI: 10.1007/s10659-013-9440-7.

[160] Ndanou, S., Favrie, N., and Gavrilyuk, S. “Multi-solid and multi-fluid diffuse interface model: Applications to dynamic fracture and fragmentation”. In: *Journal of Computational Physics* 295 (2015), pp. 523–555. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2015.04.024.

[161] Noh, W. F. and Woodward, P. “SLIC (Simple Line Interface Calculation)”. In: *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2, 1976 Twente University, Enschede*. Ed. by Vooren, A. I. van de and Zandbergen, P. J. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, pp. 330–340. ISBN: 978-3-540-37548-7. DOI: 10.1007/3-540-08004-X_336.

[162] Nourgaliev, R. and Theofanous, T. “High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set”. In: *Journal of Computational Physics* 224.2 (2007), pp. 836–866. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2006.10.031.

[163] Ohl, C. D. and Ikink, R. “Shock-Wave-Induced Jetting of Micron-Size Bubbles”. In: *Phys. Rev. Lett.* 90 (21 May 2003), p. 214502. DOI: 10.1103/PhysRevLett.90.214502.

[164] Ohl, C.-D., Arora, M., Ikink, R., Jong, N. de, Versluis, M., Delius, M., and Lohse, D. “Sonoporation from Jetting Cavitation Bubbles”. In: *Biophys. J.* 91.11 (Dec. 2006), pp. 4285–4295. ISSN: 00063495. DOI: 10.1529/biophysj.105.075366.

[165] OpenMP, A. R. B. (, (editor), M. K., and (editor), B. R. S. *OpenMP Application Programming Interface Specification Version 5.0 Paperback*. indepent, 2018, p. 668. ISBN: 978-1795759885.

[166] Osher, S., Fedkiw, R., and Piechor, K. “Level set methods and dynamic implicit surfaces”. In: *Appl. Mech. Rev.* 57.3 (2004), B15–B15. DOI: 10.1007/b98879.

[167] Osher, S. and Sethian, J. A. “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations”. In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49. ISSN: 0021-9991. DOI: 10.1016/0021-9991(88)90002-2.

[168] Osher, S. and Solomon, F. “Upwind difference schemes for hyperbolic systems of conservation laws”. In: *Mathematics of computation* 38.158 (1982), pp. 339–374. DOI: 10.1090/S0025-5718-1982-0645656-0.

[169] Pelanti, M. and Shyue, K.-M. “A mixture-energy-consistent six-equation two-phase numerical model for fluids with interfaces, cavitation and evaporation waves”. In: *Journal of Computational Physics* 259 (2014), pp. 331–357. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2013.12.003.

[170] Pember, R. B., Bell, J. B., Colella, P., Curtchfield, W. Y., and Welcome, M. L. “An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions”. In: *Journal of Computational Physics* 120.2 (1995), pp. 278–304. ISSN: 0021-9991. DOI: 10.1006/jcph.1995.1165.

[171] Peng, D., Merriman, B., Osher, S., Zhao, H., and Kang, M. “A PDE-Based Fast Local Level Set Method”. In: *Journal of Computational Physics* 155.2 (1999), pp. 410–438. ISSN: 0021-9991. DOI: 10.1006/jcph.1999.6345.

[172] Petitpas, F., Massoni, J., Saurel, R., Lapebie, E., and Munier, L. “Diffuse interface model for high speed cavitating underwater systems”. In: *International Journal of Multiphase Flow* 35.8 (2009), pp. 747–759. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2009.03.011.

[173] Petitpas, F., Saurel, R., Ahn, B.-K., and Ko, S. “Modelling cavitating flow around underwater missiles”. In: *International Journal of Naval Architecture and Ocean Engineering* 3.4 (2011), pp. 263–273. ISSN: 2092-6782. DOI: 10.2478/IJNAOE-2013-0070.

[174] Pirozzoli, S. “Generalized conservative approximations of split convective derivative operators”. In: *Journal of Computational Physics* 229.19 (2010), pp. 7180–7190. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2010.06.006.

[175] Pirozzoli, S. “Numerical Methods for High-Speed Flows”. In: *Annual Review of Fluid Mechanics* 43.1 (2011), pp. 163–194. DOI: 10.1146/annurev-fluid-122109-160718.

[176] Pishchalnikov, Y. A., Behnke-Parks, W. M., Schmidmayer, K., Maeda, K., Colonius, T., Kenny, T. W., and Laser, D. J. “High-speed video microscopy and numerical modeling of bubble dynamics near a surface of urinary stone”. In: *The Journal of the Acoustical Society of America* 146.1 (2019), pp. 516–531. DOI: 10.1121/1.5116693.

[177] Ponweiser, T. and Tsoutsanis, P. *Optimising UCNS3D, a High-Order finite-Volume WENO Scheme Code for arbitrary unstructured Meshes*. Tech. rep. Partnership For Advanced Computing in Europe, 2016.

[178] Proud, W., Nguyen, T.-T., Bo, C., Butler, B., Boddy, R., Williams, A., Masouros, S., and Brown, K. “The high-strain rate loading of structural biological materials”. In: *Metallurgical and Materials Transactions A* 46.10 (2015), pp. 4559–4566. DOI: 10.1007/s11661-015-2975-4.

[179] Ranjan, D., Oakley, J., and Bonazza, R. “Shock-Bubble Interactions”. In: *Annual Review of Fluid Mechanics* 43.1 (2011), pp. 117–140. DOI: 10.1146/annurev-fluid-122109-160744.

[180] Rasthofer, U., Wermelinger, F., Karnakov, P., Šukys, J., and Koumoutsakos, P. “Computational study of the collapse of a cloud with 12 500 gas bubbles in a liquid”. In: *Phys. Rev. Fluids* 4 (6 June 2019), p. 063602. DOI: 10.1103/PhysRevFluids.4.063602.

[181] Rayleigh, L. “VIII. On the pressure developed in a liquid during the collapse of a spherical cavity”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 34.200 (1917), pp. 94–98. DOI: 10.1080/14786440808635681.

[182] Reinarz, A., Charrier, D. E., Bader, M., Bovard, L., Dumbser, M., Duru, K., Fambri, F., Gabriel, A.-A., Gallard, J.-M., Köppel, S., Krenz, L., Rannabauer, L., Rezzolla, L., Samfass, P., Tavelli, M., and Weinzierl, T. “ExaHyPE: An engine for parallel dynamically adaptive simulations of wave problems”. In: *Computer Physics Communications* 254 (2020), p. 107251. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2020.107251.

[183] Reinders, J. *Intel Threading Building Blocks*. First. USA: O’Reilly & Associates, Inc., 2007. ISBN: 9780596514808.

[184] Reitz, R. D. and Bracco, F. V. “Mechanism of atomization of a liquid jet”. In: *The Physics of Fluids* 25.10 (1982), pp. 1730–1742. DOI: 10.1063/1.863650.

[185] Rodio, M. and Abgrall, R. “An innovative phase transition modeling for reproducing cavitation through a five-equation model and theoretical generalization to six and seven-equation models”. In: *International Journal of Heat and Mass Transfer* 89 (2015), pp. 1386–1401. ISSN: 0017-9310. DOI: 10.1016/j.ijheatmasstransfer.2015.05.008.

[186] Roe, P. “Approximate Riemann solvers, parameter vectors, and difference schemes”. In: *Journal of Computational Physics* 43.2 (1981), pp. 357–372. ISSN: 0021-9991. DOI: 10.1016/0021-9991(81)90128-5.

[187] Rossinelli, D., Hejazialhosseini, B., Hadjidoukas, P., Bekas, C., Curioni, A., Bertsch, A., Futral, S., Schmidt, S. J., Adams, N. A., and Koumoutsakos, P. "11 PFLOP/s Simulations of Cloud Cavitation Collapse". In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. SC '13. Denver, Colorado: Association for Computing Machinery, 2013. ISBN: 9781450323789. DOI: 10.1145/2503210.2504565.

[188] Rossinelli, D., Hejazialhosseini, B., Spampinato, D. G., and Koumoutsakos, P. "Multicore/Multi-GPU Accelerated Simulations of Multiphase Compressible Flows Using Wavelet Adapted Grids". In: *SIAM Journal on Scientific Computing* 33.2 (2011), pp. 512–540. DOI: 10.1137/100795930.

[189] Roussel, O., Schneider, K., Tsigulin, A., and Bockhorn, H. "A conservative fully adaptive multiresolution algorithm for parabolic PDEs". en. In: *Journal of Computational Physics* 188.2 (July 2003), pp. 493–523. ISSN: 00219991. DOI: 10.1016/S0021-9991(03)00189-X.

[190] Runge, C. "Ueber die numerische Auflösung von Differentialgleichungen". In: *Mathematische Annalen* 46.2 (June 1895), pp. 167–178. ISSN: 00255831. DOI: 10.1007/BF01446807.

[191] Rusanov, V. V. "The calculation of the interaction of non-stationary shock waves with obstacles". In: *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 1.2 (1961), pp. 267–279.

[192] Samfass, P., Weinzierl, T., Charrier, D. E., and Bader, M. "Lightweight task offloading exploiting MPI wait times for parallel adaptive mesh refinement". In: *Concurrency and Computation: Practice and Experience* 32.24 (2020), e5916. DOI: 10.1002/cpe.5916.

[193] Sankin, G. N. and Zhong, P. "Interaction between shock wave and single inertial bubbles near an elastic boundary". In: *Phys. Rev. E* 74 (4 Oct. 2006), p. 046304. DOI: 10.1103/PhysRevE.74.046304.

[194] Saurel, R., Petitpas, F., and Abgrall, R. "Modelling phase transition in metastable liquids: application to cavitating and flashing flows". In: *Journal of Fluid Mechanics* 607 (2008), pp. 313–350. DOI: 10.1017/S0022112008002061.

[195] Saurel, R., Petitpas, F., and Berry, R. A. "Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures". In: *Journal of Computational Physics* 228.5 (2009), pp. 1678–1712. ISSN: 0021-9991. DOI: doi.org/10.1016/j.jcp.2008.11.002.

[196] Schmidmayer, K., Petitpas, F., and Daniel, E. "Adaptive Mesh Refinement algorithm based on dual trees for cells and faces for multiphase compressible flows". In: *Journal of Computational Physics* 388 (2019), pp. 252–278. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.03.011.

[197] Schmidmayer, K., Petitpas, F., Daniel, E., Favrie, N., and Gavrilyuk, S. "A model and numerical method for compressible flows with capillary effects". In: *Journal of Computational Physics* 334 (2017), pp. 468–496. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2017.01.001.

[198] Schmidmayer, K., Petitpas, F., Le Martelot, S., and Daniel, É. "ECOGEN: An open-source tool for multiphase, compressible, multiphysics flows". In: *Computer Physics Communications* 251 (2020), p. 107093. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2019.107093.

[199] Schneider, K. and Vasilyev, O. V. “Wavelet Methods in Computational Fluid Dynamics”. en. In: *Annual Review of Fluid Mechanics* 42.1 (Jan. 2010), pp. 473–503. ISSN: 0066-4189, 1545-4479. DOI: 10.1146/annurev-fluid-121108-145637.

[200] SE, D. S. *Technical Reference SolidWorks Flow Simulation 2021*. 2021.

[201] Sethian, J. A. and Smereka, P. “Level Set Methods for Fluid Interfaces”. In: *Annual Review of Fluid Mechanics* 35.1 (2003), pp. 341–372. DOI: 10.1146/annurev.fluid.35.101101.161105.

[202] Shi, J., Zhang, Y. T., and Shu, C. W. “Resolution of high order WENO schemes for complicated flow structures”. In: *Journal of Computational Physics* 186.2 (2003), pp. 690–696. ISSN: 0021-9991. DOI: 10.1016/S0021-9991(03)00094-9.

[203] Simmendinger, C., Rahn, M., and Gruenewald, D. “The GASPI API: A Failure Tolerant PGAS API for Asynchronous Dataflow on Heterogeneous Architectures”. In: *Sustained Simulation Performance 2014*. Ed. by Resch, M. M., Bez, W., Focht, E., Kobayashi, H., and Patel, N. Cham: Springer International Publishing, 2015, pp. 17–32. ISBN: 978-3-319-10626-7. DOI: 10.1007/978-3-319-10626-7_2.

[204] Smolyak, S. A. “Interpolation and quadrature formulas for the classes W_s^α and E_s^α”. In: *Doklady Akademii Nauk*. Vol. 131. 5. Russian Academy of Sciences. 1960, pp. 1028–1031.

[205] So, K., Hu, X., and Adams, N. “Anti-diffusion interface sharpening technique for two-phase compressible flow simulations”. In: *Journal of Computational Physics* 231.11 (2012), pp. 4304–4323. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2012.02.013.

[206] Sod, G. A. “A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws”. In: *Journal of Computational Physics* 27.1 (Apr. 1978), pp. 1–31. ISSN: 0021-9991. DOI: 10.1016/0021-9991(78)90023-2.

[207] Soni, V., Hadjadj, A., Roussel, O., and Moebs, G. “Parallel multi-core and multi-processor methods on point-value multiresolution algorithms for hyperbolic conservation laws”. In: *Journal of Parallel and Distributed Computing* 123 (2019), pp. 192–203. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2018.09.016.

[208] Speziale, C. G. “A combined large-eddy simulation and time-dependent RANS capability for high-speed compressible flows”. In: *Journal of Scientific Computing* 13.3 (1998), pp. 253–274.

[209] Sroka, M., Engels, T., Krah, P., Mutzel, S., Schneider, K., and Reiss, J. “An Open and Parallel Multiresolution Framework Using Block-Based Adaptive Grids”. In: Springer, Cham, 2019, pp. 305–319. DOI: 10.1007/978-3-319-98177-2_19.

[210] Stride, E. P. and Coussios, C. C. “Cavitation and contrast: The use of bubbles in ultrasound imaging and therapy”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 224.2 (2010), pp. 171–191. DOI: 10.1243/09544119JEIM622.

[211] Šukys, J., Rasthofer, U., Wermelinger, F., Hadjidoukas, P., and Koumoutsakos, P. “Multilevel Control Variates for Uncertainty Quantification in Simulations of Cloud Cavitation”. In: *SIAM Journal on Scientific Computing* 40.5 (2018), B1361–B1390. DOI: 10.1137/17M1129684.

[212] Sussman, M. and Puckett, E. G. “A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows”. In: *Journal of Computational Physics* 162.2 (2000), pp. 301–337. ISSN: 0021-9991. DOI: 10.1006/jcph.2000.6537.

[213] Sussman, M., Smereka, P., and Osher, S. “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”. In: *Journal of Computational Physics* 114.1 (1994), pp. 146–159. ISSN: 0021-9991. DOI: 10.1006/jcph.1994.1155.

[214] Sutter, H. “The free lunch is over: A fundamental turn toward concurrency in software”. In: *Dr. Dobb’s journal* 30.3 (2005), pp. 202–210.

[215] Tagawa, Y., Oudalov, N., Ghalbzouri, A. E., Sun, C., and Lohse, D. “Needle-free injection into skin and soft matter with highly focused microjets”. In: *Lab Chip* 13 (7 2013), pp. 1357–1363. DOI: 10.1039/C2LC41204G.

[216] Tagawa, Y., Oudalov, N., Visser, C. W., Peters, I. R., Meer, D. van der, Sun, C., Prosperetti, A., and Lohse, D. “Highly Focused Supersonic Microjets”. In: *Phys. Rev. X* 2 (3 July 2012), p. 031002. DOI: 10.1103/PhysRevX.2.031002.

[217] Takayama, K. and Igra, D. “Investigation of aerodynamic breakup of a cylindrical water droplet”. In: *Atomization and Sprays* 11.2 (2001). ISSN: 1044-5110. DOI: 10.1615/AtomizSpr.v11.i2.50.

[218] Theofanous, T. “Aerobreakup of Newtonian and Viscoelastic Liquids”. In: *Annual Review of Fluid Mechanics* 43.1 (2011), pp. 661–690. DOI: 10.1146/annurev-fluid-122109-160638.

[219] Titarev, V. A. and Toro, E. F. “Finite-volume WENO schemes for three-dimensional conservation laws”. In: *Journal of Computational Physics* 201.1 (Nov. 2004), pp. 238–260. ISSN: 00219991. DOI: 10.1016/j.jcp.2004.05.015.

[220] Toro, E., Müller, L., and Siviglia, A. “Bounds for Wave Speeds in the Riemann Problem: Direct Theoretical Estimates”. In: *Computers & Fluids* 209 (2020), p. 104640. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2020.104640.

[221] Toro, E. F. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. 3rd ed. Dordrecht ; New York: Springer, 2009, p. 724. ISBN: 978-3-540-49834-6.

[222] Toro, E. F., Spruce, M., and Speares, W. “Restoration of the contact surface in the HLL-Riemann solver”. In: *Shock waves* 4.1 (1994), pp. 25–34. DOI: 10.1007/BF01414629.

[223] Torregrosa, A., Serrano, J., Arnau, F., and Piqueras, P. “A fluid dynamic model for unsteady compressible flow in wall-flow diesel particulate filters”. In: *Energy* 36.1 (2011), pp. 671–684. ISSN: 0360-5442. DOI: 10.1016/j.energy.2010.09.047.

[224] Trummler, T., Schmidt, S. J., and Adams, N. A. “Effect of stand-off distance and spatial resolution on the pressure impact of near-wall vapor bubble collapses”. In: *International Journal of Multiphase Flow* 141 (2021), p. 103618. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2021.103618.

[225] Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., and Jan, Y.-J. “A Front-Tracking Method for the Computations of Multiphase Flow”. In: *Journal of Computational Physics* 169.2 (2001), pp. 708–759. ISSN: 0021-9991. DOI: 10.1006/jcph.2001.6726.

[226] Tsoutsanis, P., Adebayo, E. M., Merino, A. C., Arjona, A. P., and Skote, M. “CWENO Finite-Volume Interface Capturing Schemes for Multicomponent Flows Using Unstructured Meshes”. In: *Journal of Scientific Computing* 89:64 (Nov. 2021). DOI: 10.1007/s10915-021-01673-y.

[227] Turangan, C. K., Ball, G. J., Jamaluddin, A. R., and Leighton, T. G. “Numerical studies of cavitation erosion on an elastic–plastic material caused by shock-induced bubble collapse”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2205 (2017), p. 20170315. DOI: 10.1098/rspa.2017.0315.

[228] Turkle, S. *Simulation and its discontents*. MIT press, 2009.

[229] Unverdi, S. O. and Tryggvason, G. "A front-tracking method for viscous, incompressible, multi-fluid flows". In: *Journal of Computational Physics* 100.1 (1992), pp. 25–37. ISSN: 0021-9991. DOI: 10.1016/0021-9991(92)90307-K.

[230] van Leer, B. "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method". In: *Journal of Computational Physics* 32.1 (1979), pp. 101–136. ISSN: 0021-9991. DOI: 10.1016/0021-9991(79)90145-1.

[231] Vasilyev, O. V. and Bowman, C. "Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations". In: *Journal of Computational Physics* 165.2 (Dec. 2000), pp. 660–693. ISSN: 0021-9991. DOI: 10.1006/JCPH.2000.6638.

[232] Vasilyev, O. V. "Solving multi-dimensional evolution problems with localized structures using second generation wavelets". In: *International Journal of Computational Fluid Dynamics* 17.2 (Mar. 2003), pp. 151–168. ISSN: 10618562. DOI: 10.1080/10618562.1000011152.

[233] Veilleux, J.-C., Maeda, K., Colonius, T., and Shepherd, J. E. "Transient Cavitation in Pre-Filled Syringes During Autoinjector Actuation". In: *Proceedings of the 10th International Symposium on Cavitation (CAV2018)*. ASME Press, Dec. 2018. ISBN: 9780791861851. DOI: 10.1115/1.861851_ch203.

[234] Verma, S., Novati, G., and Koumoutsakos, P. "Efficient collective swimming by harnessing vortices through deep reinforcement learning". In: *Proceedings of the National Academy of Sciences* 115.23 (2018), pp. 5849–5854. DOI: 10.1073/pnas.1800923115.

[235] Villermaux, E. "Fragmentation". In: *Annual Review of Fluid Mechanics* 39.1 (2007), pp. 419–446. DOI: 10.1146/annurev.fluid.39.050905.110214.

[236] Weinzierl, T. *The Pillars of Science*. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-76194-3. DOI: 10.1007/978-3-030-76194-3.

[237] Welch, P. and Boyle, P. "New turbines to enable efficient geothermal power plants". In: *Geothermal Resources Council Transactions*. Vol. 33. Making Renewable Energy HOT! 2009, pp. 765–772. ISBN: 0-934412-94-4.

[238] Wermelinger, F., Rasthofer, U., Hadjidoukas, P., and Koumoutsakos, P. "Petascale simulations of compressible flows with interfaces". In: *Journal of Computational Science* 26 (2018), pp. 217–225. ISSN: 1877-7503. DOI: 10.1016/j.jocs.2018.01.008.

[239] Wermelinger, F., Hejazialhosseini, B., Hadjidoukas, P., Rossinelli, D., and Koumoutsakos, P. "An Efficient Compressible Multicomponent Flow Solver for Heterogeneous CPU/GPU Architectures". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '16. Lausanne, Switzerland: Association for Computing Machinery, 2016. ISBN: 9781450341264. DOI: 10.1145/2929908.2929914.

[240] Williams, S., Waterman, A., and Patterson, D. "Roofline: An Insightful Visual Performance Model for Multicore Architectures". In: *Commun. ACM* 52.4 (Apr. 2009), pp. 65–76. ISSN: 0001-0782. DOI: 10.1145/1498765.1498785.

[241] Winter, J., Kaiser, J., Adami, S., and Adams, N. "Numerical investigation of 3D drop-breakup mechanisms using a sharp interface level-set method". In: 2019.

[242] Winter, J., Kaiser, J., Adami, S., Akhatov, I., and Adams, N. "Stochastic multi-fidelity surrogate modeling of dendritic crystal growth". In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114799. ISSN: 0045-7825. DOI: 10.1016/j.cma.2022.114799.

[243] Y., M., V., P., and C., R. J. “Dynamic adaptivity using wavelets basis for the approximation of partial-differential equations”. In: *C. R. Acad. Sci. Ser. I Math* 312.405-10 (1991).

[244] Zeng, Q., Gonzalez-Avila, S. R., Voorde, S. T., and Ohl, C.-D. “Jetting of viscous droplets from cavitation-induced Rayleigh-Taylor instability”. In: *Journal of Fluid Mechanics* 846 (2018), pp. 916–943. DOI: 10.1017/jfm.2018.284.

[245] Zhu, X. and Zhang, Y.-T. “Fast Sparse Grid Simulations of Fifth Order WENO Scheme for High Dimensional Hyperbolic PDEs”. In: *J. Sci. Comput.* 87.2 (May 2021). ISSN: 0885-7474. DOI: 10.1007/s10915-021-01444-9.