

Technische Universität München
TUM School of Engineering and Design

Multi-Fidelity Approaches to Modeling and Simulation of Complex Flows

Josef Maximilian Winter

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. Katrin Wudy

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Nikolaus A. Adams
2. Prof. Dr.-Ing. Wolfgang Schröder
3. Prof. Phaedon-Stelios Koutsourelakis, Ph.D.

Die Dissertation wurde am 27.03.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 01.07.2024 angenommen.

© Josef Maximilian Winter, 2024
josef.winter@tum.de

All rights reserved. No part of this publication may be reproduced, modified, re-written or distributed in any form or by any means without prior written permission of the author.

Typesetting \LaTeX

Danksagung

Die vorliegende Arbeit ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Aerodynamik und Strömungsmechanik von 2017 bis 2024 entstanden. Ich möchte mich an dieser Stelle sehr herzlich bei all jenen bedanken, die mich auf diesem Weg begleitet und unterstützt haben und somit maßgeblich zum Erfolg dieser Arbeit beigetragen haben.

Zuallererst möchte ich mich bei meinem Doktorvater Prof. Dr.-Ing. Nikolaus A. Adams für seine hervorragende Betreuung und die vielen Freiheiten und Möglichkeiten bedanken, die er mir während meiner Zeit am Lehrstuhl geboten hat. Die von ihm geschaffene Atmosphäre am Lehrstuhl hat nicht nur meine fachliche Entwicklung gefördert, sondern mich auch persönlich sehr geprägt. Sein stetiges Ermuntern, Sachverhalte aus verschiedensten Blickwinkeln zu betrachten, hat es mir ermöglicht, eine Vielzahl an Themen kennenzulernen. Insbesondere möchte ich mich auch für die Gelegenheiten zur Teilnahme an zahlreichen Konferenzen bedanken, die nicht nur meine Fachkenntnisse erweitert haben, sondern oft auch mit schönen Auslandsaufenthalten verbunden waren.

Weiterhin möchte ich herzlich Prof. Wolfgang Schröder und Prof. Phaedon-Stelios Koutsourelakis dafür danken, die Begutachtung meiner Doktorarbeit übernommen zu haben. Ihre wertvollen Rückmeldungen und Anregungen haben maßgeblich zur Qualität dieser Arbeit beigetragen. Ein besonderer Dank gilt auch Prof. Katrin Wudy für die Übernahme des Vorsitzes.

Meinem Betreuer am AER, Herrn PD Dr.-Ing. habil. Stefan Adami, möchte ich meinen besonderen Dank aussprechen. Ohne deine Unterstützung während meiner Masterarbeit und die anschließende Möglichkeit, Teil der Nanoshock-Gruppe zu werden, wäre diese Arbeit nicht möglich gewesen. Vielen Dank für deine fachliche Anleitung und die zahlreichen persönlichen Gespräche.

Weiterhin möchte ich mich bei allen Kolleginnen und Kollegen am Lehrstuhl bedanken, mit denen ich bei verschiedensten Themen zusammenarbeiten durfte. Ein besonderer Dank gilt Jakob und Nils, mit denen ich einen Großteil meiner Zeit am Lehrstuhl verbracht habe. Ihr habt mich bereits bei der Masterarbeit maßgeblich unterstützt und dazu beigetragen, dass ich immer mit Freude an den Themen gearbeitet habe. Jakob, danke für die schöne Zeit auf Konferenzen und beim regelmäßigen Espresso trinken. Nils, vielen Dank für die Spaziergänge in den Pausen und das Organisieren von Aktivitäten außerhalb der Arbeit. Danke auch an Fabian, Alexander und Rima für die unterhaltsame, aber auch produktive Zeit im Büro, auf die ich mich jeden Tag gefreut habe. Danke Steffen, dass ich mich bei jeglichen fachlichen Fragen an dich wenden durfte und ich in Sachen Lehre so viel von dir lernen konnte. Ein weiterer Dank gilt auch Nico, Thomas, Deniz, Aaron, Christian, Ludger, Felix, Naeimeh, Vladimir, Alexandr und Yiqi. Die vielen Kaffeepausen, fachlichen Gespräche und der private Austausch haben meine Zeit am Lehrstuhl zu etwas Besonderem gemacht.

Ich möchte mich auch bei allen Studentinnen und Studenten bedanken, die ich während der vergangenen Jahre bei studentischen Arbeiten oder anderen Projekten betreuen durfte. Es freut mich ganz besonders, dass einige von euch inzwischen Kolleginnen und Kollegen geworden sind.

Einen großen Anteil am erfolgreichen Gelingen dieser Arbeit hat meine Familie. Vielen Dank Mama, dass du mich wirklich bei allem unterstützt und ermuntert hast. Meinen beiden Brüdern Benedikt und Leopold gilt ebenso ein großer Dank. Ihr habt mich stets auf eure eigene Art motiviert und ich konnte mich immer auf euch verlassen. Danke Oma, dass du all die Jahre für mich da warst. Ein großer Dank gilt auch meinem Vater, der mich stets unterstützt und meine Begeisterung für die Strömungsmechanik mit geweckt hat. Ein besonderer Dank gilt meiner Freundin Vicky. Danke, dass du meine übermäßige Begeisterung für Espresso erträgst und mich bis zur letzten Stunde vor Einreichung ermutigt hast. Die schöne Zeit mit dir zuhause, in den Bergen und auch überall sonst war der perfekte Ausgleich.

Abstract

This work develops computationally efficient techniques for modeling and simulating complex flows. It relies on the concept of adaptive numerical experimentation. Adaptive experimentation describes the machine-learning guided process of systematically exploring the parameter space of a specific problem while aiming to achieve defined objectives. It is a ubiquitous approach in science and engineering. Central to this study are two flow phenomena: shock-interface interaction and dendritic growth. Both are pivotal in numerous technical applications that range from therapeutic applications in modern medicine to the burgeoning field of additive manufacturing. Advanced simulation frameworks with state-of-the-art numerical methods are necessary to simulate these multi-scale problems accurately.

The first contribution of this work is the development of a modular simulation framework for compressible multi-phase flows utilizing a sharp-interface level-set method with conservative interface interaction. Its design efficiently harnesses spatial and temporal adaptation techniques and leverages high-performance computing architectures, enabling large-scale simulations of three-dimensional problems. Secondly, a novel multi-fidelity Gaussian Process model is proposed to construct stochastic surrogate models for dendritic growth problems. These models are integral to adaptive experimentation workflows. The proposed model ensures computational efficiency by combining numerous cheap, low-fidelity estimates with fewer computationally expensive, high-fidelity samples. The newly proposed simulation framework facilitates the generation of samples with varying fidelity. Thirdly, an efficient solution strategy for general inverse Stefan problems is proposed based on similar ideas. It integrates multi-fidelity Gaussian process models in a Bayesian optimization algorithm based on a cost-aware multi-fidelity knowledge gradient acquisition function. Exploiting the specific structure of the target function and reformulating it as a composite function significantly improves the stability of the optimization procedure. Solving the inverse problem of single dendritic growth, i. e. identifying process and material parameters that achieve prescribed growth properties of the crystal, exemplifies the approach's applicability.

Successfully implementing multi-fidelity adaptive numerical experimentation for complex dendritic growth problems could expedite the rapid and efficient design of new materials with targeted properties. The tools developed in this thesis allow for the efficient modeling and simulation of complex flow problems and contribute valuable methodologies to the broader scientific community.

Zusammenfassung

In der vorliegenden Arbeit werden rechnerisch effiziente Techniken zur Modellierung und Simulation komplexer Strömungen entwickelt. Sie stützt sich auf das Konzept adaptiver numerischer Experimente. Adaptive Experimente beschreiben den durch maschinelles Lernen gesteuerten Prozess der systematischen Erkundung des Parameterraums eines bestimmten Problems unter Berücksichtigung eines vorgegebenen Ziels. Es handelt sich um einen weit verbreiteten Ansatz in Wissenschaft und Technik. Zentral für diese Studie sind zwei Strömungsphänomene: Die Interaktion von Stoßwellen mit Phasengrenzflächen und dendritisches Kristallwachstum. Beide sind von wesentlicher Bedeutung für zahlreiche technische Anwendungen, die von therapeutischen Anwendungen in der modernen Medizin bis hin zum aufstrebenden Gebiet der additiven Fertigung reichen. Fortschrittliche Simulationsprogramme mit hochmodernen numerischen Methoden sind unabdingbar, um diese mehrskaligen Probleme mit der erforderlichen Genauigkeit zu simulieren.

Das erste Ziel dieser Arbeit ist die Entwicklung einer modularen Simulationsumgebung für kompressible Mehrphasenströmungen unter Verwendung einer Level-Set-Methode. Diese erlaubt es, die Phasengrenzfläche detailliert aufzulösen und die Interaktion von Fluiden konservativ zu modellieren. Die Nutzung räumlicher und zeitlicher Komprimierungsalgorithmen sowie Parallelisierungsstrategien für moderne Hochleistungsrechnerarchitekturen ermöglicht hoch aufgelöste Simulationen von dreidimensionalen Problemen. Zweitens wird ein neuartiges Multi-Fidelity Gaußsches Prozessmodell vorgestellt, um stochastische Ersatzmodelle für dendritische Wachstumsprozesse zu generieren. Diese Modelle sind integraler Bestandteil des adaptiven Experimentierens. Das vorgeschlagene Modell gewährleistet Recheneffizienz, indem es zahlreiche kostengünstige Abschätzungen mit wenigen rechenintensiven, hochgenauen Datenpunkten kombiniert. Die vorgestellte Simulationsumgebung erleichtert die Erzeugung von Datenpunkten mit variabler Genauigkeit. Drittens wird eine effiziente Lösungsstrategie für allgemeine inverse Stefan-Probleme vorgeschlagen welche Multi-Fidelity-Gaußsche Prozessmodelle in einen Bayesschen Optimierungsalgorithmus integriert. Der adaptive Algorithmus generiert iterativ neue Datenpunkte, um die Genauigkeit der Lösung des Optimierungsproblems zu verbessern. Die neuen Datenpunkten werden basierend auf einer Kosten-Nutzen-Abschätzung bestimmt. Die Ausnutzung der spezifischen Struktur der Zielgrößenfunktion und damit verbundene Reformulierung als zusammengesetzte Funktion verbessert die Stabilität des Optimierungsverfahrens signifikant. Die Anwendbarkeit des Ansatzes wird durch die Lösung des inversen Problems für dendritisches Kristallwachstum gezeigt. Es werden Prozess- und Materialparameter identifiziert, mit welchen vorgegebene Wachstumseigenschaften des Kristalls erreicht werden.

Die erfolgreiche Umsetzung von Multi-Fidelity Ansätzen des adaptiven Experimentierens für komplexe dendritische Wachstumsprobleme ermöglicht eine schnelle und effiziente Entwicklung neuartiger Materialien mit gezielten Materialeigenschaften. Zusammenfassend werden in dieser Arbeit effiziente Methodiken für die Modellierung und Simulation komplexer Strömungsprobleme entwickelt, welche für die breitere wissenschaftliche Gemeinschaft und Industrie von Nutzen sind.

Contents

Acronyms	xi
Nomenclature	xv
1 Introduction	1
1.1 Motivation	1
1.2 A modular simulation framework for compressible multi-phase flows	3
1.3 Numerical adaptive experimentation of single dendritic growth	4
2 Fundamentals	11
2.1 Mathematical and numerical model	11
2.1.1 Governing equations	11
2.1.2 Numerical model for the governing equations	12
2.1.3 Numerical model for the interface	15
2.1.4 Time integration	16
2.1.5 Spatial and temporal adaptivity	16
2.2 Stochastic surrogate modelling	18
2.2.1 Gaussian processes	18
2.2.2 Multi-fidelity modeling using Gaussian processes	19
2.3 Bayesian optimization	23
2.3.1 Single-fidelity acquisition strategy	23
2.3.2 Multi-fidelity acquisition strategy	24
3 Accomplishments	25
3.1 A modular high-performance multi-phase flow simulation framework	25
3.2 Modeling dendritic crystal growth using stochastic multi-fidelity surrogate models	28
3.3 Solving the inverse Stefan problem for dendritic crystal growth	30
4 List of Publications	33
4.1 Publications comprising this thesis	33
4.2 Further publications	33
4.2.1 Peer-reviewed journal publications	33
4.2.2 Conference proceedings	34
5 Discussion with Respect to the State of the Art and Conclusion	35
A Peer-Reviewed Journal Publications	39
A.1 ALPACA - a level-set based sharp-interface multiresolution solver for conservation laws	39
A.2 Stochastic multi-fidelity surrogate modeling of dendritic crystal growth	60
A.3 Multi-fidelity Bayesian optimization to solve the inverse Stefan problem	80
Bibliography	103

Acronyms

1D one dimensional

2D two dimensional

3D three dimensional

ALPACA Adaptive Levelset Parallel Code Alpaca

ALTS adaptive local time stepping

AMR adaptive mesh refinement

API application programming interface

BO Bayesian optimization

CDF cumulative distributions functions

CFD computational fluid dynamics

cfKG continuous-fidelity knowledge-gradient

CFL Courant-Friedrichs-Lewy

CPU central processing unit

CTS constant time stepping

DACE design and analysis of computer experiments

DoE design of experiments

EGO efficient global optimization

EI expected improvement

EOS equation of state

FVM finite volume method

GFM ghost-fluid method

GP Gaussian process

GP-UCB Gaussian process upper confidence bound

GPU graphics processing unit

HPC high-performance computing

KG knowledge-gradient

LBM Lattice Boltzmann method

LS level-set

LTS local time stepping

MF multi-fidelity

MFDGP multi-fidelity deep Gaussian process

mfKG multi-fidelity knowledge-gradient

ML machine learning

MPI Message Passing Interface

MR multiresolution

NSE Navier-Stokes equations

OOP object-oriented programming

p-s-p process-structure-property

PDE partial differential equation

QoIs quantities of interest

RK Runge-Kutta

RMI Richtmyer-Meshkov instability

SBI shock-bubble interaction

SF single-fidelity

SPH Smoothed Particle Hydrodynamics

SVI stochastic variational inference

VOF volume of fluid

WENO weighted essentially non-oscillatory

Nomenclature

Greek letters

α	Volume fraction
Δ	Finite-volume cell
γ	Surface tension coefficient
κ	Thermal conductivity
μ	Dynamic viscosity
ϕ	Level set
Π	Stress tensor
ρ	Density
σ	Standard deviation
τ	Pseudo time
ξ	Fluid identifier

Roman letters

Δt	Time step
Δx	Cell size
I	Identity matrix
n	Normal vector
Q	Interface quantity
S	Volume-force vector
U	Vector of conserved quantities
v	Velocity vector
\mathcal{D}	Dataset
\mathcal{N}	Normal distribution
\mathcal{S}	Interface
\mathcal{V}	Material volume

A	Aperture
c	Speed of sound
c_p	Specific heat capacity
CFL	CFL constant
E	Total specific energy
e	Specific internal energy
F	Flux
g	Gravitational acceleration
p	Pressure
q	Heat flux density
T	Temperature
T	Viscous stress tensor
t	Time
V	Volume

Super- and subscripts

\mathbf{S}	Volume forces
\mathcal{S}	Interface
μ	Viscous
c	Convection
i, j, k	Cell indices
l	Liquid phase
m	Property for the melt
q	Heat
s	Property for the solid

Mathematical symbols

\cdot	Dot product
\mathbb{E}	Expectation
∇	Gradient
\otimes	Tensor product
∂	Partial derivative

Chapter 1

Introduction

1.1 Motivation

This thesis overviews my research on the numerical experimentation of multi-phase flow problems involving interfacial instabilities. The focus is on three topics: building a performant numerical simulation framework for multi-phase flows, generating multi-fidelity stochastic surrogate models for dendritic crystal growth, and analyzing inverse Stefan problems using multi-fidelity Bayesian optimization techniques.

Multi-phase flows are paramount in everyday life and govern flow phenomena on all scales. They lead to the formation of droplets in the atmosphere or bubbles in the ocean [22], can be observed when preparing or enjoying meals [139], or govern critical biological and medical flows such as blood flow [22] to only name a few examples. Given this diversity and ubiquity, a characterization of multi-phase flow based on observations is difficult. Instead, a classification based on common phenomenological themes [22] is motivated. Disperse flows and separated flows are the resulting topologies. Disperse flows consist of finite particles, droplets, or bubbles interspersed within a connected, continuous medium. In contrast, separated flows involve distinct continuous phases of different fluids, separated by well-defined interfaces [22].

Multi-phase flows may be inherently unstable, i. e. they cannot sustain small perturbations to which any physical system is subjected to. Such flows depart progressively from their initial state when exposed to small disturbances and never revert to it. In contrast, stable multi-phase flows may dampen the small disturbances and revert to the initial state [27]. Unstable phase interfaces may again be found on all scales. Examples are regular wave formations of clouds in the atmosphere [61], visually fascinating effects when preparing drinks [139] or inside microfluidic devices [32].

A particularly important type of inherently unstable multi-phase flow is liquid-solid phase transition under non-equilibrium conditions. A prominent example thereof is the formation of dendritic microstructures. Phenomena in nature, such as ice crystals [29], and many manufacturing processes such as century-old welding, macroscopic castings, or modern additive manufacturing rely on the growth of dendritic microstructures [123]. The latter manufactures metallic bulk parts layer-by-layer based on complex computer-aided designs [118]. The single layers are microstructures that result from solidification processes that begin with the formation of dendrites. Additive manufacturing processes offer a wide variety of process parameters that influence the development of microstructures. The microstructure determines the mechanical properties of the solidified phase and, therefore, of the manufactured product. This process-structure-property (p-s-p) relationship is an important research topic investigated by many researchers [118]. Modeling the p-s-p relationship experimentally and numerically is necessary to understand the underlying mechanisms. However, identifying optimal operation conditions of additive manufacturing processes that yield targeted material properties ultimately requires solving the inverse problem. Solving the forward and inverse problem of the p-s-p relationship requires accurate models describing the unstable interface growth of dendritic microstructures.

Numerical simulations are a powerful tool for investigating interfacial instabilities. They allow insights into spatial and temporal features far out of reach when performing experiments and allow non-invasive measurements. The mathematical and numerical models have to be carefully chosen to get the desired

detailed insight into the phenomena of interest. Also, it has to be ensured that the model captures all relevant phenomena with sufficient accuracy. Various numerical methods are available to solve the governing equations of fluid flow. Those are e.g. the finite volume method (FVM), Smoothed Particle Hydrodynamics (SPH) or Lattice Boltzmann method (LBM), to list only a few. The methodology used for the investigations in this thesis is the FVM, as it is the most widespread approach for simulating the variety of flow phenomena considered in this work. Various methods to model phase interfaces are available. They can be categorized as interface-capturing schemes with smeared interfaces or interface-tracking schemes with a sharp interface representation. Prominent interface capturing schemes are volume of fluid (VOF) methods [90, 147] or the diffuse interface method [1, 5]. Prominent examples for the interface tracking schemes are the free-Lagrange [13, 204], front-tracking [69, 70, 202, 205], or level-set (LS) algorithm [39, 152, 197]. This work uses the LS approach since it naturally handles topology changes [66, 67], allows straightforward calculation of geometric quantities [183], and eases imposing interfacial physics [26]. An extensive overview of LS methods and a comparison to other interface modeling approaches can be found in [66, 183].

Applying high-fidelity numerical simulations to investigate interfacial instabilities has been successful in several fields and improved understanding of the underlying mechanisms [28, 68, 132, 180, 203, 220]. Interfacial instabilities occurring in real systems are often strongly influenced by the system state, which is characterized by a variety of parameters. Investigating the complete parameter space by simulating all possible and relevant parameter combinations quickly becomes time-consuming, computationally expensive or even infeasible. Thus, efficient computational techniques are necessary to explore the parameter space. Adaptive experimentation offers a powerful tool, therefore [11, 73]. It assumes that a system's response depends on independent input parameters and aims to find a set of input parameters that give an optimal system response. Based on an initially available set of (numerical) experiments, adaptive experimentation iteratively updates the beliefs about the system to decide which experiment to run next. Bayesian optimization (BO) [105, 124, 144] and bandit optimization [40, 119] are prominent examples of adaptive experimentation techniques. Whereas the former is specifically suitable for problems with continuous parameters, the latter is commonly used for problems with a finite set of choices. The parameter spaces describing the interfacial growth processes studied here are mostly continuous. Thus, BO techniques are used in this work. In addition, multi-fidelity modeling is used to improve the efficiency of the employed BO technique. Multi-fidelity modeling relies on the fact that a system's response may often be modeled with varying fidelity. The high-fidelity model describes the system with sufficient accuracy for the task at hand. Low-fidelity models represent the same system with less accuracy but are typically computationally cheaper [157]. Fusing data from various fidelities may reduce the overall cost of BO.

This work investigates the potential of adaptive experimentation techniques for the cost-efficient analysis of interface instabilities and includes three major contributions:

1. A modular high-performance computational fluid dynamics (CFD) simulation framework for compressible multi-phase flows. It serves as the simulator for the adaptive numerical experimentation of interface instabilities.
2. A novel strategy for generating stochastic multi-fidelity surrogate models based on multi-fidelity deep Gaussian process (MFDGPs). It is an approach for efficiently generating surrogate models leveraging resolution fidelities that are naturally given by the discretization scheme of the newly developed simulation framework.
3. An efficient multi-fidelity Bayesian optimization framework for solving the inverse problem for dendritic crystal growth. It relies on the developed simulation framework for solving the forward problem and again leverages resolution fidelities for multi-fidelity modeling.

In the following Section 1.2, an overview of the CFD simulation framework for compressible multi-phase flows developed in the scope of this thesis is given. Section 1.3 introduces dendritic growth

processes and the adaptive numerical experimentation framework used for their analysis. Chapter 2 presents the governing equations for fluid flow and their numerical discretization, multi-fidelity stochastic surrogate models, and the BO approaches used for adaptive experimentation. Chapter 3 highlights the contributions and novelties of the publications included in this work. For each publication, a short review of the current state in the corresponding field is given, pointing out the particular issues that are addressed. Chapter 4 lists all publications published during my time as a doctoral candidate. Chapter 5 concludes the thesis and gives an outlook on possible future work.

1.2 A modular simulation framework for compressible multi-phase flows

A wide range of scientific and engineering problems can be effectively modeled using conservation equations. Applications include, but are not limited to, traffic flow [131, 134, 171], the shallow water equations [177, 194, 200], electromagnetism [145], plasmaphysics [9] or gasdynamics [133]. The governing conservation equations for these problems often exhibit high nonlinearity and may include mechanisms describing the dynamics of discontinuities, such as shock waves or phase interfaces. Accurate, high-fidelity predictions of such multiscale problems necessitate significant temporal and spatial resolutions alongside numerical methods capable of capturing their complex dynamics.

For single-phase flows, the (cell-averaged) FVM Godunov's approach [71] is a popular method for discretizing the Navier-Stokes equations (NSE). It guarantees discrete conservation, reduces the smearing of flow states to a minimum, and allows suppressing oscillations at discontinuities. Also, it enables combining approximate Riemann solvers [53, 82, 172] with high-order low-dissipative weighted essentially non-oscillatory (WENO) type reconstruction stencils [102]. WENO are attractive since they allow constructing stencils of arbitrary order. However, they introduce high computational cost and in combination with selected Riemann solvers might show severe numerical instabilities [54, 55, 62]. To optimally select numerical schemes for a wide range of flow problems, a simulation framework must provide implementations for a variety of different schemes that can be interchanged seamlessly.

A suitable interface model must extend the framework for scenarios involving multiple phases. This work primarily focuses on separated multi-phase flows, where small interface structures might strongly influence the dynamics. The sharp-interface level-set method, introduced by Osher and Sethian [152], allows accurately locating the interface structure on the numerical grid and straightforward calculation of geometric surface parameters such as surface normals or curvature. It defines the interface as the zero-contour of a higher-dimensional function ϕ that denotes the signed distance to the interface. Thus, the absolute value of the level-set value at a certain position gives the normal distance to the interface, and its sign indicates the fluid. This inherently limits the level-set method to simulate two-phase flows. Extensions to an arbitrary number of fluids is a topic of ongoing research [154, 155]. The LS is advanced in time by solving an additional advection equation and subsequent reinitialization to re-establish the signed-distance property of the LS.

Modeling phase interfaces with the LS method requires imposing boundary conditions at fluid-fluid interfaces. This is achieved using the ghost-fluid method (GFM) proposed by Fedkiw et al. [47]. It introduces grid-aligned ghost values for each fluid on the opposite side of the interface. Imposing interfacial physics then reduces to appropriately populating the ghost cells. In the original GFM, ghost-cells are necessary for the single- and two-phase discretization. [96] propose explicit, conservative interface-exchange terms. These are based on the solution of a two-material Riemann problem at the interface and independent of the information in the ghost cells. Thus, ghost cells are solely used for single-phase discretization. The interface-exchange terms can account for inviscid- [96], viscous- [136], and heat-exchange [166] effects. Also, models to consider evaporation [129] or liquid-solid phase transition [108] exist. Combining the sharp-interface level-set method with conservative interface-exchange terms is compatible with the introduced high-order low-dissipative Godunov-type scheme for single-phase discretization. A simulation framework implementing the proposed method must consider several

performance aspects to enable large-scale simulations of complex multiscale and multi-phase flow phenomena.

Popular techniques to improve the efficiency of numerical solvers are spatial and temporal compression techniques. Two prevalent algorithms to achieve spatial compression, i. e. dynamic adaption of the computational-grid resolution to the underlying flow field, are gradient-based adaptive mesh refinement (AMR) [19] and wavelet-based multiresolution (MR) [81] algorithms [41, 78, 94, 148, 181, 189, 191, 210]. This thesis focuses on MR compression. Compared to AMR, it improves CPU and memory compression rates for the same accuracy level when discretizing hyperbolic conservation laws [38]. To fully leverage the potential of modern high-performance computing (HPC) systems, a block-based variant of the MR algorithm can be employed [79, 92, 93, 191]. The spatially varying grid resolution of block-based MR compression allows its combination with temporal compression. The time-step size Δt is often chosen to satisfy stability criteria, such as the Courant-Friedrichs-Lewy (CFL) condition, that directly depend on the local cell size Δx . This motivates local time stepping (LTS) [42] and adaptive local time stepping (ALTS) [111] approaches that advance coarser regions with larger time steps compared to finer regions. They thus outperform constant time stepping (CTS) schemes that advance all regions with the globally smallest time-step size [17, 84, 175]. Block-based MR schemes can be efficiently parallelized via the Message Passing Interface (MPI) using a domain-decomposition approach that treats MR block as the smallest inter-rank parallel unit [93]. Thereby, the efficient usage of modern HPC systems is possible and highly-resolved simulations, which give detailed insights into the problems relevant to this work, can be conducted. The modular open-source simulation framework Adaptive Levelset Parallel Code Alpaca (ALPACA) implements all the mentioned aspects. Section 3.1 highlights this thesis's contribution to the development of the framework.

1.3 Numerical adaptive experimentation of single dendritic growth using Gaussian Process surrogate models and Bayesian Optimization

The solidification process of a material in liquid state can either be stable or unstable. The stable process is characterized by a flat planar interface, while the unstable process exhibits microstructure patterns. These patterns emerge when small, local perturbations at the solid-liquid interface are enhanced and amplified over time. Unstable growth occurs when the liquid phase is supercooled, i. e. cooled below its solid-liquid equilibrium temperature, and thermodynamically metastable [101]. The dominating microstructure patterns are dendritic (tree-like) structures, featuring a main trunk and side branches that grow at different locations of the main trunk [4].

Modeling the liquid-solid phase transition that leads to microstructure formation poses a Stefan problem [193]. It describes a free boundary problem between phases of a material undergoing phase change. Decades of research have focused on understanding and mathematically modeling dendritic growth patterns [195]. After an initial transient phase, the growth of a single dendrite becomes steady. Crucial to this process is the relationship between a dendrite's operating point, specified by its tip radius and velocity, and local melt parameters like undercooling. For a two-dimensional parabolic dendrite, Ivantsov [100] first mathematically related the product of tip velocity and radius to melt undercooling. To uniquely determine the operating point, an additional independent criterion is required. Oldfield [149] introduced a stability criterion based on the balance of capillarity and diffusion, but this does not accurately predict dendritic branching behavior [99, 153]. Langer and Müller-Krumbhaar [128] also consider the interplay of capillarity and diffusion and derive a stability criterion applying a Mullins-Sekerka stability analysis, which, combined with Ivantsov's model, provides a unique solution for the operating point. Several authors [6, 14, 16, 18, 24, 115, 140, 158, 201] proposed to include capillary anisotropy to correctly capture the morphological development during crystal growth. This so-called microscopic solvability theory for the operating point of dendrites provides sufficiently accurate predictions at high undercooling [122], and can be used for validating numerical models. Validated numerical models may then provide detailed

insight into more complex directional dendritic growth patterns on the microscale, including multiple dendrites. Furthermore, these numerical models allow studying complex processes beyond the microscopic solvability theory, which e. g. involve varying material parameters.

Detailed and accurate investigation of dendritic microstructure formation often relies on physics-driven, simulation-based digital twins [118]. Popular, mesh-based approaches for interface-modeling in liquid-solid phase transition are the front-tracking [106, 173, 184], phase-field [113, 117], enthalpy [208, 209], VOF [33, 135], and the previously discussed LS [67, 152] method. The front-tracking approach explicitly represents the interface using lower-dimensional grids that evolve and deform with the solid-liquid interface. The phase-field, enthalpy, VOF and LS implicitly represent the interface and solve evolution equations for auxiliary variables defined on every grid cell [101]. Among the level-set approaches, Kaiser et al. [108] developed a semi-implicit sharp-interface formulation based on conservative interface fluxes into both phases. It is particularly appealing since it ensures low numerical dissipation by using high-order spatial and temporal discretization schemes, and is thus employed in this work.

The scheme proposed by Kaiser et al. [108] allows physics-driven investigation of dendritic microstructures - the finest scale of the p-s-p relationship. Data-driven methods offer a compelling alternative for modeling the p-s-p relationship, as discussed by Kouraytem et al. [118]. These methods are particularly useful for integrating diverse information sources, integrating data derived from physics-based simulations or experimental results [89]. A synergistic combination of both physics- and data-driven approaches may contribute to the “high-throughput highway to computational materials design” [34]. This combination can contribute to various major aspects in analyzing the p-s-p relationship. It facilitates a comprehensive understanding of the underlying mechanisms and prediction of outcomes (a bottom-up perspective). Additionally, it allows for optimizing process strategies and addressing inverse problems (a top-down perspective). This dissertation contributes to both of these paradigms, proposing efficient solution strategies for the forward and inverse problem of single dendritic growth.

Efficient and accurate models are crucial for linking the operational behavior of a dendrite to process parameters. These models tackle the forward problem of single dendritic growth, a crucial part of broader-scale modeling endeavors such as grain or part scale simulations [137]. Relying solely on numerical simulations to solve the forward problem necessitates significant computational resources and may not be suitable for scenarios where a simplified input-output relationship suffices. Understanding the dependence of steady-state growth parameters on relevant system properties and environmental conditions, while neglecting the transient evolution of the phase interface, may be mentioned as an example. In such cases, surrogate models offer a promising alternative.

The input-output relation investigated in this work considers directional dendritic solidification. An undercooled melt surrounds an initially spherical seed. Crystallization of the seed is initiated since the temperature of the surrounding melt is below the liquid-solid phase transition temperature. Due to anisotropic surface tension, the crystal exhibits preferred growth directions along the main coordinate axes, as sketched in Fig. 1.1. After an initially transient phase, the growth velocity of the dendritic tip, denoted by u_Γ , converges against a steady-state solution. The steady-state tip velocity depends on material and local melt parameters. Those are the normalized undercooling τ , capillary anisotropy ε , and the thermal conductivity ratio κ_m/κ_s between melt and solid. The normalized undercooling can be determined using

$$\tau = \frac{(T - T_m)c}{L}, \quad (1.1)$$

where T is the initial temperature in the melt, T_m the melting temperature, c the specific heat capacity, and L the latent heat. Thus, the functional dependence for the tip velocity that is modeled using surrogate models reads

$$u_\Gamma = f(\tau, \varepsilon, \kappa_m/\kappa_s). \quad (1.2)$$

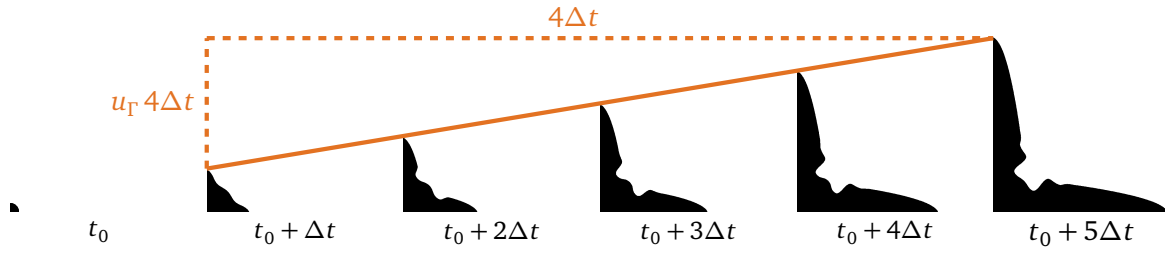


Figure 1.1: Temporal evolution of the growth of an initially spherical seed for equidistant snapshots in time. The initial time and time interval between the snapshots are denoted by t_0 and Δt , respectively. The visualization of the single snapshots leverages the symmetry of the crystal and only depicts one-fourth of the dendrite. Starting from the second image, a constant growth velocity u_T of the tip can be observed.

Modern data-driven techniques based on statistical learning allow the generation of elaborated surrogate models that accurately capture complex engineering systems [25, 58, 83, 169]. A prominent representative for regression surrogate models are Gaussian process (GP) surrogates, which are already successfully applied to model or control micro- and macro-scale additive-manufacturing processes [120, 168, 179, 223] of, e.g., composites [95] or ceramics [141, 219]. The cost for data generation often dominates when generating GP surrogate models solely relying on data obtained from highly accurate numerical simulations. This may be overcome by multi-fidelity modeling. Therein, not just the high-fidelity model - which provides a sufficiently accurate system description for the intended application - but also less accurate, computationally more efficient or cheaper lower-fidelity models are utilized. Supplementing high-fidelity data with lower-fidelity approximations may improve model accuracy. Typical examples of low-fidelity models are models with reduced dimensionality, previously developed experimental relationships, or coarse-grid approximations [10, 157]. Multi-fidelity and multi-level methods are closely connected. Both aim to increase overall modeling quality by augmenting the high-fidelity model with lower-fidelity information. This concept is widely spread for the efficient construction of response surfaces [10, 116, 142, 161], uncertainty quantification [15, 43, 46, 52, 143] or solutions for inverse problems [45]. Multi-level methods typically define a hierarchy of low-fidelity models by varying a single parameter, such as the grid resolution of a numerical simulator. In contrast, multi-fidelity methods allow for deriving fidelity hierarchies based on more general models and various information sources such as experimental data, regression models directly derived from high-fidelity data or linearized models [157]. Peherstorfer et al. [157] provide an extensive review of multi-fidelity modeling. This work focuses on multi-fidelity GP regression to cost-efficiently construct surrogate models, which is particularly attractive due to the inherent capability to quantify predictive uncertainty.

In their pioneering work, Kennedy and O'Hagan [114] propose the first multi-fidelity GP model. It combines a GP model with a linear autoregressive scheme. Their approach was successfully applied for problems where the relationship between the low and high-fidelity model is linear [31, 58, 160, 162]. This, however, is the major shortcoming, as for many real-world applications strong non-linear relationships between fidelity levels occur [161]. Thus, several authors extend the work of Kennedy and O'Hagan [114] and develop models that capture nonlinear and space-dependent relationships between fidelity levels [35, 161, 167]. Among them, Raissi and Karniadakis [167] combine the linear autoregressive scheme of [114] with deep neural networks. Their model is especially suitable when the relationship between fidelity levels contains discontinuities. Inspired by the structure of deep GPs [36, 37], the functional composition of GPs, Perdikaris et al. [161] and Cutajar et al. [35] develop an approach integrating GP regression and nonlinear autoregressive schemes. This allows capturing nonlinear and space-dependent relationships between fidelity levels. This work focuses on the MFDGP model of Cutajar et al. [35] as it better resists overfitting. Another factor that might harm the accuracy of standard GP regression models is the non-stationarity of input-output relations. These are relations where the covariance is not invariant to translation in input space. As a remedy, Snoek et al. [188] motivate learning parameters of a family of

cumulative distributions functions (CDF) to transform the input space. Their approach is referred to as input warping and has successfully been applied in many practical applications such as for the estimation of loss reserves [126] or model-tuning frameworks [163].

In addition to addressing the direct Stefan problem related to dendritic growth, the rapid and efficient development of new materials with desired properties necessitates tackling the inverse problem [63, 185]. In the context of dendritic growth, this involves determining local melt parameters, such as undercooling, as well as material parameters that lead to specific growth characteristics of the dendrite. While exact analytical solutions for the inverse Stefan problem exist, they are limited to certain one-dimensional scenarios [186]. In other cases, approximate solution strategies are required.

Diverse methodologies exist to address the inverse Stefan problem. Several authors [7, 103, 104, 146] consider a one-dimensional inverse Stefan problem and determine heat fluxes to obtain a prescribed phase-interface. Other approaches consider determining material parameters, such as thermal coefficients, in the context of solidification problems [23, 192]. Zabarar et al. [221, 222] identify a heat flux at the domain boundary given interface velocities and the heat flux passing through the interface. They reduce the inverse Stefan problem to a minimization problem. Voller [207] uses enthalpy and sensitivity coefficients to find the heat flux at a boundary with a prescribed position of the interface. Grzymkowski and Słota [74, 75] employ optimization in the context of an Adomian decomposition method to solve the one-phase inverse Stefan problem. The aforementioned literature mainly considers one-dimensional and one-phase problems. Studies considering multiple phases, two-dimensional problems, or their combination are scarce or have practical limitations due to the reliance on specific mathematical models. Future studies addressed these gaps. Słota [187] focuses on a two-phase problem and uses a genetic optimization algorithm to select the convective heat-transfer coefficient of the boundary for a given interface position. Following the idea of employing optimization approaches for solving the inverse problem, Hetmaniok et al. [88] propose using Ant Colony Optimization to determine the heat transfer coefficient in two dimensional (2D) binary alloy solidification. They highlight the computational complexity of their approach and emphasize the importance of computationally efficient algorithms to investigate inverse problems for complex solidification processes.

When formulating inverse Stefan problems as an optimization task, BO is an attractive solution strategy. BO is an iterative optimization procedure that relies on two main components: A stochastic surrogate model that mimics the behavior of the target function based on available samples, and an acquisition function that iteratively updates prior beliefs about the real system and guides the optimization by proposing the location for the next samples. Figure 1.2 exemplarily sketches the BO workflow. The idea of BO originates from the work of Kushner [124]. He considers a one dimensional (1D) problem and employs Wiener processes as surrogate model. The optimization process is guided by maximizing the probability of improvement. Perttunen [164] and Elder [44] extend this work to higher-dimensional problems. Another important work is by Moćkus [144], who proposes the widely used expected improvement (EI) acquisition function. The efficient global optimization (EGO) algorithm proposed by Jones et al. [105] motivates applying BO to optimization problems where the amount of function evaluations is severely limited by time or cost. This makes it especially attractive for engineering problems. They employ a design and analysis of computer experiments (DACE) stochastic process model. DACE is based on Kriging, which in some variants is equivalent to GPs regression [206], the most prominent type of surrogate modeling in BO. To further improve the efficiency of BO, combining GP surrogate models and acquisition functions capable of multi-fidelity (MF) modeling is an attractive approach. MF acquisition functions must balance between sample fidelity and accuracy when proposing the next sample. Developing multi-fidelity BO frameworks is an active field of research. Huang et al. [98], Lam et al. [127] and Picheny et al. [165] propose an augmented EI capable of MF modeling. Kandasamy et al. [112] propose a multi-fidelity acquisition function based on the Gaussian process upper confidence bound (GP-UCB) [190]. Wu et al. [217, 218] proposed the multi-fidelity knowledge-gradient (mfKG) acquisition function which considers the cost of observing samples with a certain fidelity. The choice of the next sample's location and fidelity is based on the trade-off between the estimated improvement in solution quality and

the sampling cost at a certain fidelity level. This formulation allows both batch and sequential optimization. Compared to other state-of-the-art algorithms, the mfKG achieves better performance and is thus considered in this work. The contributions of this thesis to analyzing the forward and inverse Stefan problem are highlighted in Sections 3.2 and 3.3.

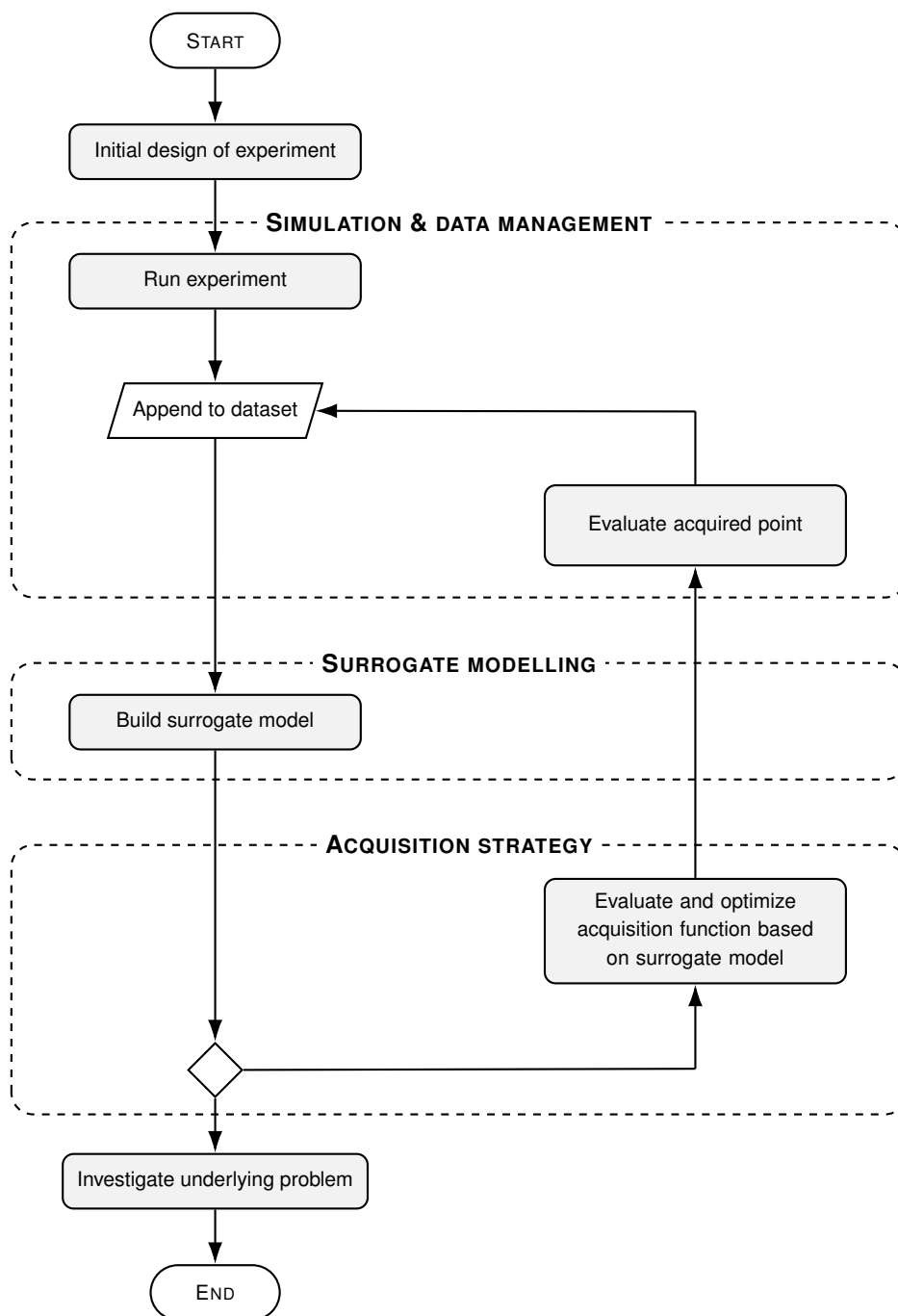


Figure 1.2: Flow graph of a BO algorithm. First, an initial design of experiments (DoE) is generated. Then, the target function is evaluated for the points generated by the initial DoE. The points and corresponding targets are collected in a dataset. Based on it, a stochastic surrogate model mimicking the behavior of the target function is constructed. Using the surrogate model, investigations on the real system, such as e.g. optimization, can be realized. In case an optimum for the target function is sought, an acquisition function is constructed. It expresses the most promising setting for the next sample to be generated. The next sample is generated at the location that maximizes the acquisition function. The newly acquired point is then appended to the dataset and the procedure repeats. By that, the model quality and consequently the estimated solution of the optimization problem are improved.

Chapter 2

Fundamentals

2.1 Mathematical and numerical model

2.1.1 Governing equations

The compressible NSE are given by the system of conservation equations

$$\frac{\partial \rho}{\partial t} = -\operatorname{div} \rho \mathbf{u} \quad (2.1)$$

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

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

considering the conservation of mass, momentum, and energy, respectively. Therein, ρ denotes the density, \mathbf{u} the velocity vector, \mathbf{g} the gravitational acceleration vector, and $E = e + \mathbf{u}^2/2$ the specific total energy. The specific total energy consists of specific internal and kinetic energy. Heat fluxes read

$$\mathbf{q} = -\kappa \nabla T \quad (2.4)$$

with the thermal conductivity κ and temperature T . The stress tensor for Newtonian fluids reads

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

where p denotes the pressure, μ the dynamic viscosity, \mathbf{I} the identity matrix, and \mathbf{T} the viscous stress tensor.

In integral formulation, Eqs. (2.1) to (2.3) read

$$\int_{\mathcal{V}} \frac{\partial \rho}{\partial t} dV = - \oint_{\partial \mathcal{V}} (\rho \mathbf{u}) \cdot \mathbf{n} dA \quad (2.6)$$

$$\int_{\mathcal{V}} \frac{\partial \rho \mathbf{u}}{\partial t} dV = - \oint_{\partial \mathcal{V}} (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{I} - \mathbf{T}) \cdot \mathbf{n} dA + \int_{\mathcal{V}} \rho \mathbf{g} dV \quad (2.7)$$

$$\int_{\mathcal{V}} \frac{\partial \rho e}{\partial t} dV = - \oint_{\partial \mathcal{V}} (\rho e \mathbf{u} + p\mathbf{I} - \mathbf{T} \mathbf{u} + \mathbf{q}) \cdot \mathbf{n} dA + \int_{\mathcal{V}} \rho \mathbf{g} \cdot \mathbf{u} dV, \quad (2.8)$$

where \mathcal{V} denotes a material volume with boundary $\partial \mathcal{V}$. Reformulations yield the flux-based formulation

$$\int_{\mathcal{V}} \frac{\partial \mathbf{U}}{\partial t} dV = - \oint_{\partial \mathcal{V}} (\mathbf{F}^c + \mathbf{F}^\mu + \mathbf{F}^q) dA + \int_{\mathcal{V}} \mathbf{S} dV \quad (2.9)$$

for the state vector $\mathbf{U} = [\rho \quad \rho \mathbf{u} \quad \rho e]^T$, with convective, viscous and heat-flux densities

$$\mathbf{F}^c = \begin{bmatrix} (\rho \mathbf{u}) \cdot \mathbf{n} \\ (\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n} \\ (\rho e \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n} \end{bmatrix}, \mathbf{F}^\mu = \begin{bmatrix} 0 \cdot \mathbf{n} \\ -\mathbf{T} \cdot \mathbf{n} \\ -\mathbf{T} \mathbf{u} \cdot \mathbf{n} \end{bmatrix}, \text{ and } \mathbf{F}^q = \begin{bmatrix} 0 \\ 0 \\ \mathbf{q} \cdot \mathbf{n} \end{bmatrix}, \quad (2.10)$$

respectively. The volume-force vector is given by

$$\mathbf{S} = \begin{bmatrix} 0 \cdot \mathbf{n} \\ (\rho \mathbf{g}) \cdot \mathbf{n} \\ (\rho \mathbf{g} \cdot \mathbf{u}) \cdot \mathbf{n} \end{bmatrix}. \quad (2.11)$$

For two-phase flows, the material volume \mathcal{V} consists of two distinct and immiscible fluid phases ξ_a , where $a \in [1, 2]$, that occupy non-overlapping subdomains \mathcal{V}^{ξ_a} , $a \in [1, 2] \mid \mathcal{V} = \mathcal{V}^{\xi_1} \cup \mathcal{V}^{\xi_2}$. They are separated by the internal surface \mathcal{S} that represents the phase interface. The interface-normal vector pointing towards \mathcal{V}^{ξ_a} is denoted as $\mathbf{n}_S^{\xi_a}$. When omitting the superscript the interface-normal vector pointing towards ξ_1 is considered, i. e. $\mathbf{n}_S = \mathbf{n}_S^{\xi_1}$.

In multi-phase flow, each phase satisfies the NSE separately. This allows to reformulate the flux-based formulation given by Eq. (2.9) to

$$\begin{aligned} \int_{\mathcal{V}^{\xi_1} \cup \mathcal{V}^{\xi_2}} \frac{\partial \mathbf{U}}{\partial t} dV = & - \oint_{(\partial \mathcal{V}^{\xi_1} \cup \partial \mathcal{V}^{\xi_2}) \setminus \mathcal{S}} (\mathbf{F}^c + \mathbf{F}^\mu + \mathbf{F}^q) dA \\ & - \oint_{\mathcal{S}} (\mathbf{F}_S^c + \mathbf{F}_S^\mu + \mathbf{F}_S^q) dA \\ & + \int_{\mathcal{V}^{\xi_1} \cup \mathcal{V}^{\xi_2}} \mathbf{S} dV, \end{aligned} \quad (2.12)$$

with the convective, viscous and heat transfer interface flux densities

$$\mathbf{F}_S^c = \begin{bmatrix} (\rho \mathbf{u}) \cdot \mathbf{n}_S \\ (\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n}_S \\ (\rho e \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n}_S \end{bmatrix}, \mathbf{F}_S^\mu = \begin{bmatrix} 0 \cdot \mathbf{n}_S \\ -\mathbf{T} \cdot \mathbf{n}_S \\ -\mathbf{T} \mathbf{u} \cdot \mathbf{n}_S \end{bmatrix}, \text{ and } \mathbf{F}_S^q = \begin{bmatrix} 0 \\ 0 \\ \mathbf{q} \cdot \mathbf{n}_S \end{bmatrix}, \quad (2.13)$$

respectively.

Thermodynamic closure of the NSE is obtained by an equation of state (EOS)

$$p = f(\rho, e) \quad (2.14)$$

relating pressure, density and energy.

2.1.2 Numerical model for the governing equations

To numerically solve the balance equations, the material volume \mathcal{V} is partitioned into a disjunct set of cuboid finite volumes

$$\Delta_{i,j,k} \leq \mathcal{V} \mid \Sigma \Delta_{i,j,k} = \mathcal{V}; \quad i, j, k \in \mathbb{N} \quad (2.15)$$

with cell size Δx and cell volume $\Delta V = \Delta x^3$. A finite volume cell may contain parts of an interface \mathcal{S} , as shown in Fig. 2.1. Therein, the cell in the center, $\Delta_{i,j,k}$, is cut by an interface. The linearized interface

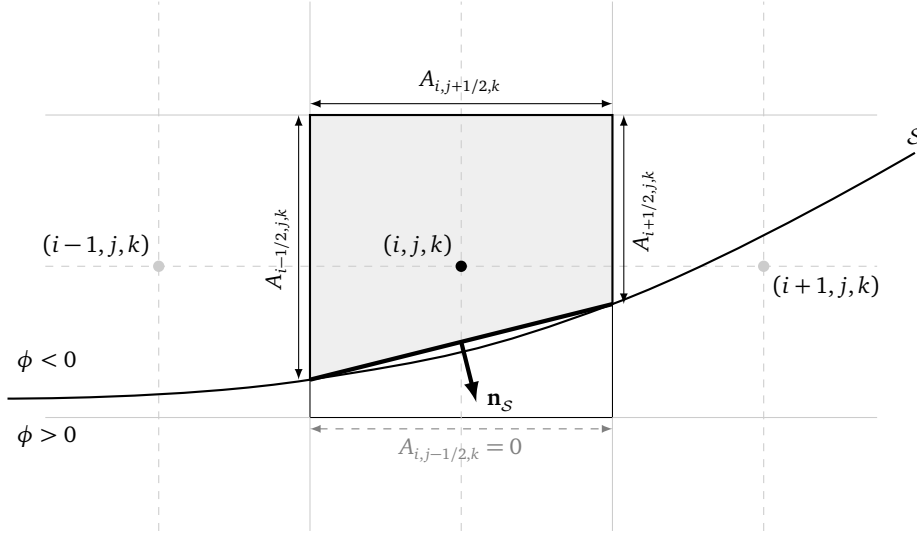


Figure 2.1: Sketch of the discretization of the part of a material volume cut by an interface \mathcal{S} . The finite-volume cell $\Delta_{i,j,k}$ and its characterizing geometrical measures are shown. Note, the figure considers a 2D slice in the i, j -plane for simplicity.

inside such *cut cells* is denoted by $\mathcal{S}_{i,j,k}$. The volumetric portion $\alpha_{i,j,k}^{\xi_a}$ of $\Delta_{i,j,k}$ is covered by fluid ξ_a . It is also referred to as volume fraction and is indicated by the gray region in the figure. Analogously, the portion of a cell face covered by ξ_a is denoted by apertures $A_{i\pm\frac{1}{2},j,k}^{\xi_a}$, $A_{i,j\pm\frac{1}{2},k}^{\xi_a}$, and $A_{i,j,k\pm\frac{1}{2}}^{\xi_a}$. The discrete set of cells comprising all cells which contain parts of a single phase is denoted by

$$\tilde{\mathcal{V}}^{\xi_a} = \left\{ \Delta_{i,j,k} \mid \alpha_{i,j,k}^{\xi_a} \neq 0 \right\}. \quad (2.16)$$

The discrete set of cells comprising all cut cells is defined as

$$\tilde{\mathcal{S}} = \left\{ \Delta_{i,j,k} \mid \alpha_{i,j,k}^{\xi_1} \neq 0 \text{ and } \alpha_{i,j,k}^{\xi_2} \neq 0 \right\}. \quad (2.17)$$

Integrating the NSE in flux formulation given in Eq. (2.12) over $\Delta_{i,j,k}$ yields

$$\int_{\Delta_{i,j,k}^{\xi_a}} \frac{\partial \mathbf{U}}{\partial t} dV = - \oint_{\partial \Delta_{i,j,k}^{\xi_a} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{c+\mu+q} dA - \oint_{\mathcal{S}_{i,j,k}} \mathbf{F}_S^{c+\mu+q} dA + \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{S} dV \quad (2.18)$$

for phase ξ_a with $\Delta_{i,j,k}^{\xi_a} = \mathcal{V}^{\xi_a} \cap \Delta_{i,j,k}$. Therein, the short notations $\mathbf{F}^{c+\mu+q} = \mathbf{F}^c + \mathbf{F}^\mu + \mathbf{F}^q$ and $\mathbf{F}_S^{c+\mu+q} = \mathbf{F}_S^c + \mathbf{F}_S^\mu + \mathbf{F}_S^q$ summarizing convective, viscous and heat-flux contributions is introduced for brevity. Equation (2.18) can be rewritten as

$$\frac{\partial (\alpha^{\xi_a} \bar{\mathbf{U}})_{i,j,k}}{\partial t} = - \oint_{\partial \Delta_{i,j,k}^{\xi_a} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{c+\mu+q} dA - \oint_{\mathcal{S}_{i,j,k}} \mathbf{F}_S^{c+\mu+q} dA + \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{S} dV \quad (2.19)$$

with the volume-averaged conservative quantities

$$\bar{\mathbf{U}} = \frac{1}{\alpha_{i,j,k}^{\xi_a}} \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{U} dV. \quad (2.20)$$

In Eq. (2.19), the first term on the right accounts for cell-face fluxes, the second term for the interaction between fluids at the phase interface, and the last term for volume forces.

The cell face fluxes can be further simplified to

$$\begin{aligned}
 - \oint_{\partial \Delta_{i,j,k}^{\xi_a} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{c+\mu+q} dA = & \left[(A^{\xi_a} \widetilde{\mathbf{F}}_1^{c+\mu+q})_{i-\frac{1}{2},j,k} - (A^{\xi_a} \widetilde{\mathbf{F}}_1^{c+\mu+q})_{i+\frac{1}{2},j,k} + \right. \\
 & (A^{\xi_a} \widetilde{\mathbf{F}}_2^{c+\mu+q})_{i,j-\frac{1}{2},k} - (A^{\xi_a} \widetilde{\mathbf{F}}_2^{c+\mu+q})_{i,j+\frac{1}{2},k} + \\
 & \left. (A^{\xi_a} \widetilde{\mathbf{F}}_3^{c+\mu+q})_{i,j,k-\frac{1}{2}} - (A^{\xi_a} \widetilde{\mathbf{F}}_3^{c+\mu+q})_{i,j,k+\frac{1}{2}} \right] \Delta x^2, \tag{2.21}
 \end{aligned}$$

assuming constant, discretized flux densities at cell faces, which are indicated by the superscript $\widetilde{(\cdot)}$. The subscripts 1, 2, 3 denote the Cartesian component of the three-dimensional flux vector. The convective fluxes are evaluated by solving a Riemann problem at cell faces and by using low-dissipative high-order shock-capturing reconstruction schemes. Determining viscous fluxes requires evaluating the viscous stress tensor at cell faces. Therefore, cell-centered velocity gradients are calculated and then reconstructed at cell faces together with respective velocities. Heat fluxes are discretized in the same way.

The interface-interaction fluxes read

$$- \oint_{\mathcal{S}_{i,j,k}} \mathbf{F}_S^{c+\mu+q} dA = (\widetilde{\mathbf{F}}_S^{c+\mu+q} \Delta \mathcal{S})_{i,j,k}, \tag{2.22}$$

assuming constant, discretized flux densities at the linearized interface inside a cell. Therein, the flux densities are calculated using the interface normal vector $\mathbf{n}_{i,j,k}^S$ and the interface segment

$$\begin{aligned}
 \Delta \mathcal{S}_{i,j,k} = & \left[(A_{i-\frac{1}{2},j,k} - A_{i+\frac{1}{2},j,k})^2 \right. \\
 & + (A_{i,j-\frac{1}{2},k} - A_{i,j+\frac{1}{2},k})^2 \\
 & \left. + (A_{i,j,k-\frac{1}{2}} - A_{i,j,k+\frac{1}{2}})^2 \right]^{\frac{1}{2}} \Delta x^2. \tag{2.23}
 \end{aligned}$$

The contribution from the convective flux densities are calculated based on the solution of the two-material Riemann problem satisfying the stress balance in interface normal direction

$$-\mathbf{n}_S \cdot \boldsymbol{\Pi}(\mathbf{U}_S^{\xi_1}) \cdot \mathbf{n}_S + \mathbf{n}_S \cdot \boldsymbol{\Pi}(\mathbf{U}_S^{\xi_2}) \cdot \mathbf{n}_S = -\gamma \mathbf{n}_S (\nabla^T \cdot \mathbf{n}_S) \cdot \mathbf{n}_S, \tag{2.24}$$

where $\mathbf{U}_S^{\xi_a}$, $i = 1, 2$ denote the interface states of the respective fluid. The above stress balance considers capillary forces, which depend on the surface-tension coefficient γ between the two fluids. The stress balance can be formulated as a two-material Riemann problem. Its solution yields the absolute values for interface velocity $(u_S)_{i,j,k}$ and interface pressures $(p_S^{\xi_1})_{i,j,k}$ and $(p_S^{\xi_2})_{i,j,k}$. These are required to evaluate the flux densities \mathbf{F}_S . Viscous interface fluxes are calculated by using the same differentiation stencil as for the viscous cell-face fluxes following [136].

The volumetric source term on the right of Eq. (2.18) reads

$$\int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{s} dV = (\alpha^{\xi_a} \widetilde{\mathbf{S}})_{i,j,k} \Delta x^3, \tag{2.25}$$

where $\widetilde{\mathbf{S}}$ denotes the constant, discretized volume-force vector inside the cell. More details and a thorough validation can be found in [93].

2.1.3 Numerical model for the interface

The LS approach captures the interface by a scalar LS function $\phi(\mathbf{x}, t)$ that fulfills the signed-distance property $|\nabla\phi| = 1$ [2, 150]. Thus, the zero LS contour

$$\mathcal{S}(\mathbf{x}, t) = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\} \quad (2.26)$$

always gives the location of the interface. The sign of the LS function allows distinguishing the two phases, i.e. $\phi > 0$ in \mathcal{V}^{ξ_1} and $\phi < 0$ in \mathcal{V}^{ξ_2} .

The previously described discretization of the balance equations in Section 2.1.2 requires information about the location and other geometric interface characteristics. The interface normal \mathbf{n}_S follows from the normalized gradient of the level set

$$\mathbf{n}_S(\mathbf{x}) = \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|}. \quad (2.27)$$

Linearly interpolating the LS function inside $\Delta_{i,j,k}$ allows analytically calculating cell-face apertures $A_{i\pm\frac{1}{2},j,k}^{\xi_a}$, $A_{i,j\pm\frac{1}{2},k}^{\xi_a}$, and $A_{i,j,k\pm\frac{1}{2}}^{\xi_a}$. The volume fractions $\alpha_{i,j,k}^{\xi_a}$ can be determined based on the cell-face apertures following [129].

The level set is evolved in time by solving the partial differential equation (PDE)

$$\frac{\partial\phi(\mathbf{x}, t)}{\partial t} = -u_S(\mathbf{x}, t)\mathbf{n}_S \cdot \nabla\phi(\mathbf{x}, t) = -u_S(\mathbf{x}, t)|\nabla\phi(\mathbf{x}, t)| = \mathcal{D}_{LSA,2}(\phi(\mathbf{x}, t)), \quad (2.28)$$

where the advection velocity u_S follows from Eq. (2.24). The advection Eq. (2.28) is strictly only valid at the interface. However, its numerical discretization requires solving it in a small region around the interface. This might lead to an irregular LS field that violates the signed-distance property. The signed-distance property can be recovered by solving the reinitialization equation

$$\frac{\partial\phi}{\partial\tau} = \text{sgn}(\phi_0)(1 - |\nabla\phi|) \quad (2.29)$$

iteratively in pseudo-time τ to steady state, according to [198]. In Eq. (2.29), $\text{sgn}(\cdot)$ denotes the sign function and ϕ_0 the level set prior to reinitialization.

Stencils used to calculate the right-hand side of the NSE or the LS advection equation might cross the interface. Thus, appropriate boundary conditions have to be imposed at the phase interface for several quantities. Therefore, the original GFM proposed by Fedkiw et al. [47] is employed. It introduces so-called *ghost cells*. For the fluid with $\phi > 0$, all cells in the vicinity of the interface with $\phi < 0$ are ghost cells. Cells with $\phi > 0$ are real-fluid cells. The definition for the fluid with $\phi < 0$ is vice versa. For the sharp interface LS method employed in this thesis, Hu et al. [96] propose filling ghost cells by extrapolating real-fluid conservative states \mathbf{U} in interface-normal direction. Therefore, the one-way extrapolation equation

$$\frac{\partial\mathbf{U}(\mathbf{x})}{\partial\tau} = \mathbf{n}_S^{\xi_i}(\mathbf{x}) \cdot \nabla\mathbf{U}(\mathbf{x}) \quad (2.30)$$

is solved iteratively in pseudo-time τ to steady-state. Details about the discretization of the right-hand side in Eq. (2.30) can be found in [151].

A two-way extrapolation is necessary to extrapolate quantities known only at the interface \mathbf{Q} to the near-interface region. This is done by solving

$$\frac{\partial\mathbf{Q}(\mathbf{x})}{\partial\tau} = \text{sgn}(\phi)\mathbf{n}_S^{\xi_i}(\mathbf{x}) \cdot \nabla\mathbf{Q}(\mathbf{x}) \quad (2.31)$$

iteratively in pseudo time τ to steady state. Details about the discretization of the right-hand side in Eq. (2.31) can be found in [151].

2.1.4 Time integration

The NSE and LS advection equations are evolved in time using explicit Runge-Kutta (RK) integration [125, 176]. To avoid spurious oscillations and ensure stability, strong stability-preserving integration schemes [80] together with a time-step size restriction

$$\Delta t_{\text{NSE}} = CFL \cdot \min(\Delta t_c, \Delta t_\mu, \Delta t_q, \Delta t_S, \Delta t_S) \quad (2.32)$$

are employed. Therein,

$$\begin{aligned} \Delta t_c &= \min_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}_1 \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}_2} \left(\frac{\Delta x}{\Sigma |||v|| + c} \right), \\ \Delta t_\mu &= \min_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}_1 \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}_2} \left(\frac{3\rho \Delta x^2}{14\mu} \right), \\ \Delta t_S &= \min_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}_1 \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}_2} \left(\frac{2\Delta x}{\Sigma |||v|| + c + \sqrt{\Sigma |||u|| + c} + 4||g||\Delta x} \right), \\ \Delta t_q &= \min_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}_1 \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}_2} \left(\frac{3\rho c_p \Delta x^2}{14\kappa} \right), \text{ and} \\ \Delta t_S &= \min_{\Delta_{i,j,k} \in \tilde{\mathcal{S}}} \left(\sqrt{\frac{\Delta x^3}{8\pi\gamma} \sum_i \rho^{\xi_a}} \right) \end{aligned} \quad (2.33)$$

denote the stability criteria for convective fluxes [111], viscous fluxes [196, 198], heat fluxes, volume forces [196, 198] and interface terms [21], respectively. The speed of sound c follows from the EOS and c_p denotes the specific heat capacity at constant pressure.

For full cells with $\alpha_{i,j,k}^{\xi_a} = 1$, stability is ensured by setting the CFL constant according to the restriction given by the employed RK integration scheme. To avoid spurious oscillations in cut cells with $\alpha_{i,j,k}^{\xi_a} \ll 1$, a conservative mixing procedure proposed in [94] is employed.

2.1.5 Spatial and temporal adaptivity

A naive implementation of the LS method in combination with the GFM leads to a high memory footprint. Also, compared to a single-phase implementation without LS, it introduces a high amount of floating point operations. Spatial and temporal adaption techniques help to reduce the computational effort. The narrow-band technique [2, 159] based on ideas of [30] improves the overall efficiency of the LS method. It limits LS related operations such as advection, reinitialization or extrapolation to a small region around the interface. To identify the cells for which the respective operations are performed, a tagging system is used. It distinguishes the following cell types:

- **Cut cells** are cells that contain the interface. For those cells, the conservative interface-interaction fluxes are calculated.
- **Cut-cell neighbors** are cells directly adjacent to cut cells.
- **Extension-band cells** are cells which are not any of the above. In addition, their distance to the nearest cut-cell neighbor cell is less than or equal to $2\Delta x$, $2\sqrt{2}\Delta x$, or $2\sqrt{3}\Delta x$ for one-, two-, or three-dimensional problems, respectively. For all cells enclosed by the extension band, the extrapolation equations for the GFM are solved.
- **Reinitialization-band cells** are cells directly adjacent to extension-band cells that are none of the above. For all cells enclosed by the reinitialization band, LS reinitialization is performed.

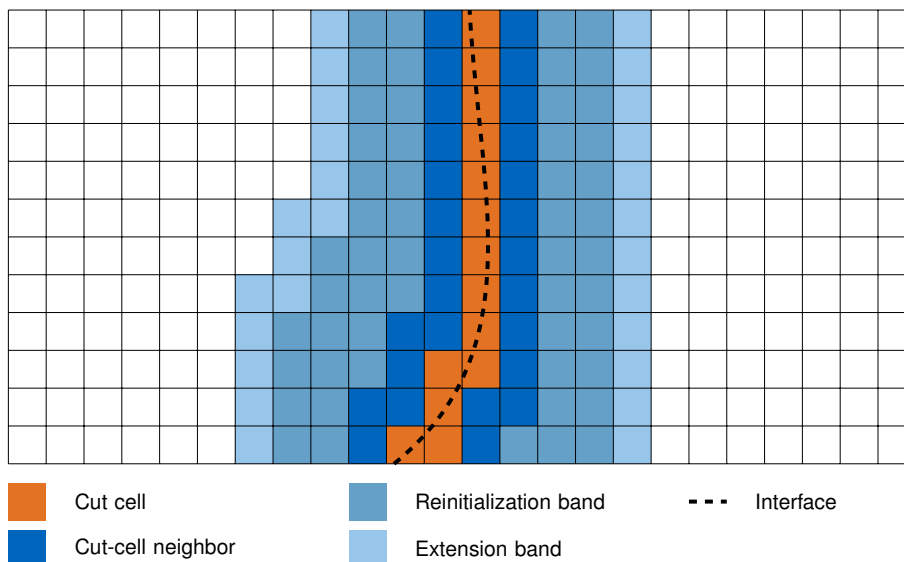


Figure 2.2: Interface tag structure for a 2D grid with curved interface. The interface line is shown in orange. Cells containing a piece of the interface are consequently marked as cut cells. The band structure of cut-cell neighbors, extension-band cells, and reinitialization-band cells is clearly visible.

- Remaining cells are referred to as **bulk cells**.

As an example, the interface-tag field for a simple 2D domain with a curved interface is shown in Fig. 2.2.

Spatial adaptivity can be realized using MR compression techniques. Compared to other spatial adaptation methods, such as AMR, they show improved central processing unit (CPU) usage and memory compression rates [38]. MR compression is based on a dyadic refinement strategy and therefore motivates its implementation using an octree to represent a three dimensional (3D) mesh. Often, nodes in the octree correspond to a single cell. However, following the concept of block-based spatial adaption techniques presented in [51, 174], this work uses a block-based MR algorithm presented by Hoppe et al. [93]. Instead of considering single cells as elements of the octree, blocks containing a predefined number of cells are considered. Each block holds its own set of halo cells. Compared to non-block-based MR approaches, this reduces the communication overhead in distributed-memory environments. The block-based MR algorithm can be seamlessly coupled with the previously introduced tagging system. Besides increasing the computational efficiency, the tagging system also reduces the memory footprint of the LS method [94]. For a 1D setup, Fig. 2.3 exemplarily shows the block-based MR octree including the mesh and halo cells. It refines around the location of the interface. The location of the interface can be traced using the introduced tagging system. Interface tags are defined and set on each resolution level.

Temporal adaptivity is realized using the ALTS scheme presented by Kaiser et al. [111]. It achieves temporal adaptivity by integrating coarser levels with larger time-step sizes than finer levels.

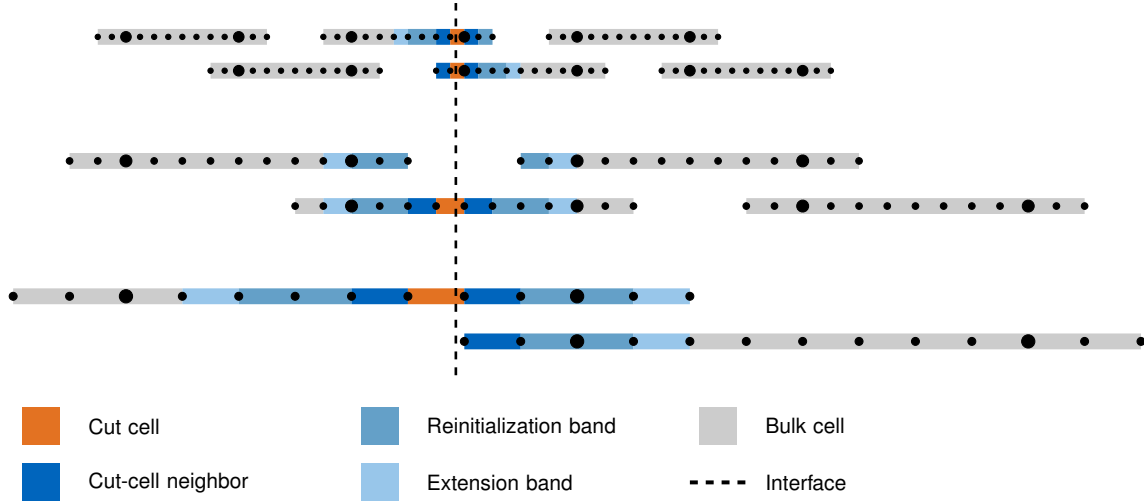


Figure 2.3: Interface tag representation for a three-level block-based MR tree in one dimension. Black dots represent cell faces, with thicker dots indicating the cell faces separating inner cells from halo cells. Each node contains two halo cells on both sides. To indicate the overlap of halo cells, blocks of the same level are offset on top or below each other.

2.2 Stochastic surrogate modelling

2.2.1 Gaussian processes

Building surrogate models for outputs generated by a numerical solver states a regression problem. The target is to learn the mapping $f(\mathbf{x})$ given a set of n input vectors $X = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ and observations $\mathbf{y} = [y_1, \dots, y_n]$ with $y_i \in \mathbb{R}$. Assuming that observations may be corrupted with zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, they are related to the unknown function by a homoscedastic noise model

$$y_i = f(\mathbf{x}_i) + \epsilon_i. \quad (2.34)$$

The available observation can be summarized in the training dataset

$$\mathcal{D} = \{X, \mathbf{y}\}. \quad (2.35)$$

In the scope of this work, the unknown function $f(\mathbf{x})$ is assigned a GP prior. A "Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution" [169, p. 13]. A multivariate Gaussian distribution is fully specified by its mean vector and covariance matrix. For a GP, the entries of the mean vector and covariance matrix are generated by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad (2.36)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \quad (2.37)$$

where \mathbb{E} denotes the expectation. A zero mean function is often assumed for a GP prior, thus $m(x) = 0$. The symmetric positive covariance function, also denoted as kernel function, quantifies the pairwise covariance between two input points x and x' . The kernel function can encode prior knowledge, such as monotonicity or periodicity, on the function into the GP model.

For regression problems, the primary goal is to compute the output of the function $\mathbf{f}_* = f(X_*)$ for n_* (unseen) inputs $X_* = [\mathbf{x}_{1,*}, \dots, \mathbf{x}_{n_*,*}]$. Therefore, the conditional posterior distribution of the

unknown function given observations $p(\mathbf{f}_*|\mathcal{D}, \mathbf{x}_*)$ has to be determined. According to the GP prior, the joint distribution of the observations in the training dataset and the predicted function values is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_\epsilon^2 \mathbf{I} & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right), \quad (2.38)$$

where \mathcal{N} indicates a normal distribution. Its mean and covariance are given by the first and second argument, respectively. In Eq. (2.38), $K(X, X)$ is the $n \times n$ matrix of the covariances evaluated at all pairs of training points, $K(X, X_*)$ a $n \times n_*$ matrix of the covariances evaluated at all pairs of training and test points, $K(X_*, X)$ a $n_* \times n$ covariance matrix of test and training points, and $K(X_*, X_*)$ a $n_* \times n_*$ covariance matrix between all pairs of test points. Making predictions then follows from conditioning the joint Gaussian prior distribution on the observations. This yields the predictive Gaussian distribution

$$\begin{aligned} \mathbf{f}_* | X_*, X, \mathbf{y} \sim \mathcal{N}(K(X_*, X) [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y}, \\ K(X_*, X_*) - K(X_*, X) [K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*)). \end{aligned} \quad (2.39)$$

For better readability, the notations $K = K(X, X)$ and $K_* = K(X, X_*)$ are introduced. If only one test point \mathbf{x}_* is considered, i.e. $n_* = 1$, the vector containing the covariances between the test point and the training points will be abbreviated by $\mathbf{k}(\mathbf{x}_*) = \mathbf{k}_*$. Following this notation, the mean and covariance of the predictive distribution for a single test point can be written as

$$\bar{f}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \quad (2.40)$$

and

$$\text{Cov} f_* = k(x_*, x_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*, \quad (2.41)$$

respectively. An example of GP regression is visualized in Fig. 2.4.

The kernel function is typically parameterized using a set of hyperparameters θ that are learned from the data. Therefore, the log marginal likelihood $\log p(\mathbf{y}|X, \theta)$, which is the probability of the observed data in the training set given their inputs and the hyperparameters of the kernel function, is maximized. It can be determined by the marginalization over the function values \mathbf{f} and is given by

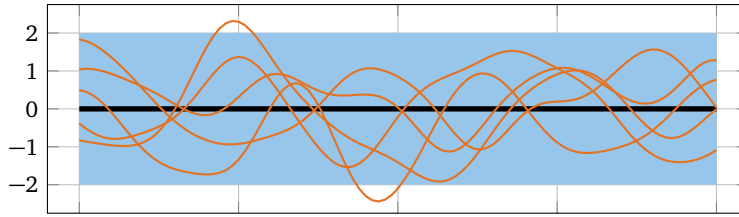
$$p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|\mathbf{f}, X) p(\mathbf{f}|X, \theta) d\mathbf{f}. \quad (2.42)$$

For a GP under the above-mentioned assumptions, $\mathbf{f}|X \sim \mathcal{N}(\mathbf{0}, K)$ and $\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 I)$ hold [169]. This yields the log marginal likelihood

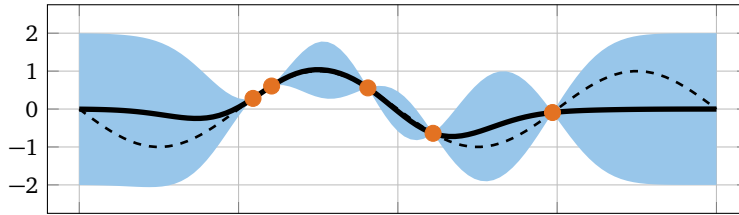
$$\log p(\mathbf{y}|X) = -\frac{1}{2} \mathbf{y}^T (K + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{n}{2} \log 2\pi. \quad (2.43)$$

2.2.2 Multi-fidelity modeling using Gaussian processes

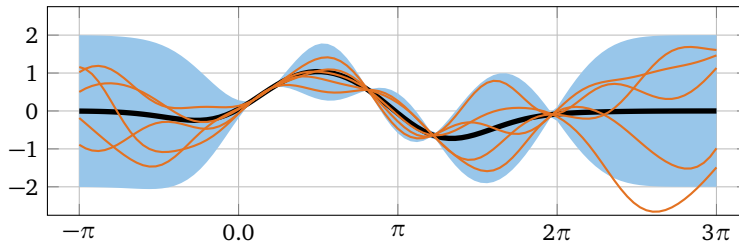
The described GP regression framework can be systematically extended for MF modeling. Multi-fidelity regression fuses limited true observations (*high-fidelity*) with many cheap approximate observations (*low-fidelity*) [114, 157]. GPs are particularly attractive as their stochastic nature allows incorporating prior beliefs about the relation between fidelity levels and quantifying predictive uncertainty. In the scope of this work, two MF modeling approaches based on GPs are considered. First, MF modeling is achieved by an appropriate kernel function. Second, an autoregressive scheme is adapted for GP regression.



(a) Five different function realizations sampled from a GP prior with zero mean and a squared exponential kernel function are shown in orange. The prior mean is shown by the black line. The 95% confidence interval, which corresponds to twice the standard deviation, is shown in blue.



(b) Based on five training points, which are shown by the orange dots, the mean of the posterior predictive distribution is indicated by the black line. The 95% confidence interval is shown in blue. The true sine function is shown by the black dotted line.



(c) The posterior predictive mean and 95% confidence interval are highlighted as before. The orange lines show five different function realizations of the GP fitted to the data.

Figure 2.4: GP regression for the 1D regression problem of the sine function.

Kernel functions with continuous fidelity parameter

In case the unknown function $f(\mathbf{x})$ can be evaluated with different fidelity, a continuous fidelity parameter s can be introduced. The unknown function then depends on the inputs and the fidelity parameter $f(\mathbf{x}, s)$. A GP prior with an appropriately shaped kernel function that captures both dependencies can be introduced [218] to obtain a MF GP regression model.

Autoregressive schemes

Kennedy & O'Hagan [114] and Le Gratiet & Garnier [130] introduce multi-fidelity GPs based on the autoregressive scheme of [114]. It is based on a hierarchy of discrete fidelity levels and assumes a linear relation

$$f_l(\mathbf{x}) = \beta_l f_{l-1}(\mathbf{x}) + \delta_l(\mathbf{x}) \quad (2.44)$$

between them. Therein, f_l denotes the high-fidelity function and f_{l-1} is the function at the preceding lower fidelity. Both functions are modeled using a GP prior. The sum on the right side of the equation highlights that the high-fidelity function at level l contains two contributions. The first contribution relates

fidelity levels by linearly scaling samples drawn from the GP at the preceding lower fidelity $l - 1$ with the constant factor β_l . The second contribution $\delta_l(\mathbf{x})$ models an input-space dependent bias between fidelities and also gets assigned a GP prior. Since fidelity levels are scaled using a constant in this approach, it misses accuracy when the relationship between fidelities is nonlinear and input-space dependent [161]. As a remedy, Perdikaris et al. [161] and Cutajar et al. [35] generalize the autoregressive multi-fidelity scheme of Eq. (2.44) to

$$f_l(\mathbf{x}) = \beta_l(f_{l-1}(\mathbf{x}), \mathbf{x}) + \delta_l(\mathbf{x}). \quad (2.45)$$

It has a similar structure to the linear autoregressive scheme. But, instead of introducing a linear scaling between fidelity levels, a space-dependent nonlinear transformation β_l , that is modeled using a GP prior, reflects the influence of the preceding lower fidelity level $l - 1$. The second term in the right-hand side of Eq. (2.45) again considers a space-dependent bias $\delta_l(\mathbf{x})$. Perdikaris et al. [161] and Cutajar et al. [35] suggest leveraging the additive structure and independence assumption between the GPs β_l and δ_l , and rewrite the autoregressive scheme of Eq. (2.45) as a composition of GPs

$$f_l(\mathbf{x}) = g_l(f_{l-1}^*(\mathbf{x}), \mathbf{x}). \quad (2.46)$$

Therein, the contributions from the space-dependent nonlinear transformation between fidelity levels and the space-dependent bias are summarized in a single GP g_l . As a consequence, it takes \mathbf{x} and samples from the posterior of the GP modeling the preceding fidelity $f_{l-1}^*(\mathbf{x})$ as inputs. Cutajar et al. [35] express the compositional structure of Eq. (2.46) with deep GPs [37, 178] and derive MFDGPs that accurately capture nonlinear and space-dependent relations between fidelity levels. In the scope of this work, MFDGPs in settings with up to three fidelity levels are employed. Therefore, a MF dataset

$$\mathcal{D}_{MF} = \{(\mathbf{X}^1, \mathbf{y}^1), (\mathbf{X}^2, \mathbf{y}^2), (\mathbf{X}^3, \mathbf{y}^3)\} \quad (2.47)$$

comprising data from three fidelity levels is used. In Eq. (2.47), \mathbf{X}^l and \mathbf{y}^l denote inputs and corresponding outputs at fidelity level $l \in \{1, 2, 3\}$. Following the notation of the autoregressive schemes, the fidelity increases with the superscript l . Based on a MF training dataset, MFDGP models as sketched in the lower part of Fig. 2.5 can be built. Since the GP in the first layer refers to the lowest fidelity level, it only depends on data from the original input space. In contrast, the GPs in the second and third layer combine data from the input space and corresponding function evaluations from the GPs modeling the first and second fidelity level, respectively. The mean and standard deviation of the posterior distribution of the MFDGP at fidelity level l are denoted as \bar{y}_l^* and $\sigma[y_l^*]$, respectively, see Fig. 2.5. The GPs at each level are modeled using standard GPs as presented in Section 2.2.1 with a zero mean function. The covariance function is defined according to Cutajar et al. [35]. The covariance between two inputs \mathbf{x}^i and \mathbf{x}^j of the GP prior at each level is expressed by

$$k_l = k_l^\beta(\mathbf{x}^i, \mathbf{x}^j | \theta_l^\beta) \left[\sigma_l^2 f_{l-1}^*(\mathbf{x}^i)^T f_{l-1}^*(\mathbf{x}^j) + k_l^{f-1}(f_{l-1}^*(\mathbf{x}^i), f_{l-1}^*(\mathbf{x}^j) | \theta_l^{f-1}) \right] + k_l^\delta(\mathbf{x}^i, \mathbf{x}^j | \theta_l^\delta). \quad (2.48)$$

Inspired by the structure of the autoregressive schemes, it consists of two contributions. The first scales the covariance between outputs obtained from the preceding fidelity level with an input-space-dependent scaling factor that considers the covariance between points from the input space k_l^β . The covariance between outputs from the preceding fidelity level consists of the sum of k_l^{f-1} and the linear covariance $\sigma_l^2 f_{l-1}^*(\mathbf{x}^i)^T f_{l-1}^*(\mathbf{x}^j)$ with variance hyperparameter σ_l^2 . The second contribution of the right-hand side of Eq. (2.48), k_l^δ , captures the bias at the l -th fidelity level. Except for the linear covariance term, all remaining covariances are assigned Matérn covariance functions [138] with smoothness measure

5/2 and automatic relevance determination. For the first layer, the covariance function from Eq. (2.48) reduces to

$$k_1 = k_1^\delta(\mathbf{x}^i, \mathbf{x}^j | \theta_1^\delta). \quad (2.49)$$

The MFDGP model is trained using the two-step approach proposed by Cutajar et al. [35]. In turn, this is based on stochastic variational inference (SVI) techniques [86, 91]. The integrals to calculate the posterior distribution of the MFDGP model are generally intractable [36]. Thus, sampling-based approaches are required for their evaluation.

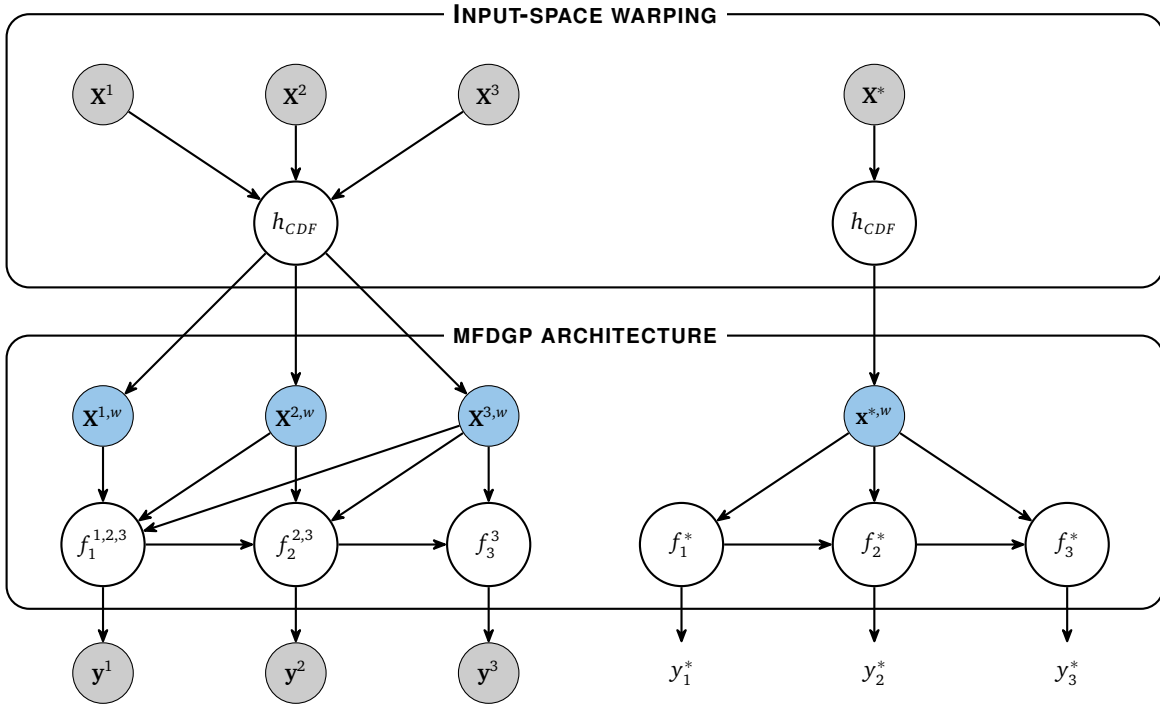


Figure 2.5: Left: Combination of input warping and a MFDGP model architecture with three fidelity levels. Gray nodes indicate inputs and outputs from a multi-fidelity training dataset with three fidelity levels. Blue nodes indicate warped inputs. White nodes show latent variables and denote samples drawn from a GP. The fidelity level associated with quantities is indicated by superscripts. Subscripts indicate the layer of the MFDGP. Thus, f_1^2 denotes the evaluation of the MFDGP at the first layer for the inputs of the second fidelity level. Right: Predictions using the same architecture with three fidelity levels for an (unseen) input \mathbf{x}^* . The input for the GP on each level consists of the original input \mathbf{x}^* and the evaluation of the model up to the previous fidelity level. The output of the model for fidelity l is the posterior predictive distribution at the respective level, which is denoted by y_l^* . Picture adapted from [213].

The predictive accuracy of MFDGPs suffers when using stationary kernel functions, such as the previously mentioned kernel functions from the Matérn family, to model non-stationary functions with spatially varying length scales. To overcome this issue, Snoek et al. [188] warp the input space using learned transformations based on CDFs. Integrating this idea with the autoregressive scheme of Eq. (2.46) yields

$$f_t(\mathbf{x}) = g_t(f_{t-1}^*(h_{CDF}(\mathbf{x})), h_{CDF}(\mathbf{x})), \quad (2.50)$$

where $h_{CDF}(\mathbf{x})$ is a bijective warping function. The MFDGP then gets a transformed representation of the original inputs $h_{CDF}(\mathbf{x})$ as input, cf. Fig. 2.5. In the scope of this work, the class of Kumaraswamy CDF [57, 121] is used as input warping function. The hyperparameters of the Kumaraswamy CDF and the MFDGP model are jointly optimized. The implementation of the model architecture combining MFDGPs and input-space warping is based on GPyTorch [65] and BoTorch [12].

2.3 Bayesian optimization

BO is an iterative optimization algorithm that typically aims to optimize quantities of interest (QoIs) that depend on a set of design parameters $\mathbf{x} := \{x_1, x_2, \dots, x_d\}$ on the design space $\mathcal{X} \in \mathbb{R}^d$. Throughout this work, evaluating the QoIs requires running computationally expensive numerical simulations, limiting the number of feasible evaluations to a few hundred. The target of BO is to optimize a target function $f(\mathbf{x})$, representing the QoIs, that depends on the simulator outputs. The goal is to find \mathbf{x}_{opt} such that

$$\mathbf{x}_{\text{opt}} = \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad (2.51)$$

A dependency between simulator observations \mathbf{y} and the target function can be expressed as

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) + \epsilon \quad (2.52)$$

with zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. When designing an algorithm used to solve the described optimization problem, several challenges have to be considered. Besides the fact that evaluating the target function is computationally expensive, its evaluation typically does not include any information about gradients with respect to the design parameters. Also, information about the structure of the target function, such as concavity or linearity, is missing. As a solution, BO optimizes a stochastic surrogate model, the first crucial component of a BO algorithm, that mimics the behavior of the target function, instead of relying on the real model. The second crucial component of a BO algorithm is an acquisition function $\varphi_t : \mathcal{X} \rightarrow \mathbb{R}$. It is a policy that drives the iterative search for the optimal point in the design space. A typical BO algorithm follows the following procedure:

1. Based on an initial DoE, an initial dataset \mathcal{D}_{BO} is generated.
2. A stochastic surrogate model is fitted to \mathcal{D}_{BO} . In the scope of this work, GP surrogate models are employed.
3. The solution of the optimization problem specified in Eq. (2.51) is estimated by either optimizing the surrogate model using standard optimization algorithms or by assuming the currently available observation with the best target-function value as optimum.
4. Based on the estimated solution of the optimization problem and the posterior distribution of the stochastic surrogate model, an acquisition function is constructed. Acquisition functions are defined in a way, that high values correspond to a high interest in placing additional data points at the respective locations. High interest can be because a certain point is close to the point that solves the optimization problem, has high predictive uncertainty, or a combination of both. The point maximizing the acquisition function is considered as next sample. Optimization of the acquisition function can be realized using standard optimization approaches.
5. Evaluate the sample proposed by the acquisition function and extend the dataset with it.
6. Iterate starting from the second step, until a convergence criterion is met.

In order to further increase the efficiency of BO, MF modeling can be used. Therefore, both the stochastic surrogate model and the acquisition function must be suitable for MF modeling. GP surrogate models suitable for single-fidelity (SF) and MF settings have already been presented in Section 2.2.

2.3.1 Single-fidelity acquisition strategy

The EI maximizes the expected difference between the currently observed best value and the objective at the next query point. It is especially attractive since its analytic form including the gradient is

available [105]. Look-ahead acquisition functions are another noteworthy approach. They explicitly take into account the effect of observations on the surrogate model in subsequent optimization steps. Knowledge-gradient (KG) [60, 76], predictive entropy search [87], and heuristic approaches [72] are popular representatives.

Assuming a maximization problem, the KG quantifies the expected increase in the solution from obtaining an additional set of data \mathcal{D}_{KG} . Compared to simpler acquisition functions such as EI, it often shows improved performance [182]. Nevertheless, naive implementations of KG are computationally expensive and hard to implement. Balandat et al. [12] propose a performant, simple, and flexible variant of the KG leveraging modern paradigms of computation. Further improvements to the standard KG include considering correlated beliefs [59] and enabling parallel batch-optimization [216]. The parallel KG is referred to as q-KG.

If in a certain state during the optimization n observations of the simulator are contained in the dataset \mathcal{D}_{BO} , $\max_{x \in \mathcal{X}} \bar{f}_n^*(x)$ denotes the maximum of the posterior mean, and thus the estimated solution of the optimization problem after observing n samples. The parallel q-KG policy then aims to suggest the next batch of q samples, which in combination with corresponding observations are then added to \mathcal{D}_{BO} . Subsequently, the next iteration of the BO starts. The extended dataset is denoted by $\mathcal{D}_{\text{BO+KG}}$. The maximum of the posterior mean of the extended dataset is $\max_{x \in \mathcal{X}} \bar{f}_{n+q}^*(x)$. The expected increase in the maximum posterior mean after gathering a batch of q samples is the difference $\max_{x \in \mathcal{X}} \bar{f}_{n+q}^*(x) - \max_{x \in \mathcal{X}} \bar{f}_n^*(x)$. Its probability distribution can be computed for the batch of q samples $\mathbf{x}_{1:q} := \{z_1, z_2, \dots, z_q\}$ by defining the q-KG factor

$$\text{q-KG}(\mathbf{x}_{1:q}, \mathcal{X}) = \mathbb{E}_n \left[\max_{x \in \mathcal{X}} \bar{f}_{n+q}^*(x) \mid \mathbf{y}(\mathbf{x}_{1:q}) \right] - \max_{x \in \mathcal{X}} \bar{f}_n^*(x), \quad (2.53)$$

where $\mathbb{E}_n[\cdot] := \mathbb{E}[\cdot \mid \mathbf{x}^{(1:n)}, \mathbf{y}^{(1:n)}]$ is the expectation conditioned on the posterior distribution after n observations. The next sample batch $\mathbf{x}_{1:q}$ is given by the points maximizing the q-KG factor.

2.3.2 Multi-fidelity acquisition strategy

In the MF setting, the target function depends on the design space and a continuous fidelity parameter $s \in [0, 1]$, where increasing values of s indicate increasing fidelity. This defines the optimization task to find the optimum of $f(\mathbf{x}, s = 1)$ for the highest fidelity level with $s = 1$. The cost for sampling the target function depends on the fidelity parameter s . This has to be taken into account by the acquisition policy. Also, the acquisition policy must not only suggest the location where to sample next, but also the corresponding fidelity s . Therefore, the continuous-fidelity knowledge-gradient (cfKG) function [217] is used. The cfKG [217] acquisition function extends the q-KG policy introduced in Eq. (2.53) to multi-fidelity problems described by a continuous fidelity parameter s . The solution of the BO after having n observations in the dataset is estimated by the highest-fidelity point that maximizes the posterior mean $x_{\text{HF}}^* = \operatorname{argmax}_{x \in \mathcal{X}} \bar{f}_n^*(x, 1)$. To further guide the BO, the q-cfKG suggests a batch of q input locations and fidelity parameters $z_{1:q} = ((x_1, s_1), \dots, (x_q, s_q))$ for the next iteration. Analogous to the SF setting, the expected increase in the maximum posterior gathering q additional samples can be expressed by $\max_{x \in \mathcal{X}} \bar{f}_{n+1}^*(x, 1) - \max_{x \in \mathcal{X}} \bar{f}_n^*(x, 1)$. Based on this measure for the expected increase in solution quality, and by considering the cost-function $\text{cost}_n(x, s)$ that estimates the cost of querying a sample at point x with fidelity s , the q-cfKG describes the benefit-cost ratio

$$\text{q-cfKG}(z_{1:q}) = \frac{\mathbb{E}_{n+q} \left[\max_{x' \in \mathcal{X}} \bar{f}_{n+q}^*(x', 1_m) \mid z_{n+1:n+q} = z_{1:q} \right] - \max_{x' \in \mathcal{X}} \bar{f}_n^*(x', 1_m)}{\max_{1 \leq i \leq q} \text{cost}_n^*(z_{n+i})}. \quad (2.54)$$

The batch of q points that maximizes the q-KG factor is the next to be sampled.

Chapter 3

Accomplishments

This chapter offers a comprehensive summary of the three primary, first-author, peer-reviewed publications [94, 212, 213] comprising this thesis, and highlights their accomplishments. The publications collectively address the three principal elements of the adaptive numerical experimentation workflow sketched in Fig. 1.2: The first component is the simulation framework, which delivers a high-fidelity numerical simulator for complex flow problems. The second are stochastic surrogate models that accurately mimic the physical system. Lastly, the acquisition function is discussed, which plays a crucial role in steering the experimentation workflow towards effectively and precisely solving the target problem.

3.1 A modular high-performance multi-phase flow simulation framework

The capability of accurately simulating demanding multi-phase flow phenomena requires state-of-the-art numerical schemes. As presented in Section 2.1, the sharp-interface LS method with conservative interface interaction is a particularly attractive choice for compressible two-phase flows. Especially when overall dynamics are heavily influenced or even dominated by the interface. The accurate representation of the interface by the LS is beneficial. However, the integration of the proposed LS algorithm into a simulation framework is demanding due to the additional computational complexity.

This challenge is addressed by seamlessly integrating the LS algorithm into the compressible CFD simulation framework ALPACA [3]. ALPACA provides spatial adaptivity by a sophisticated multiresolution compression algorithm [93] and temporal adaptivity by ALTS [111]. The algorithmic backbone of the LS implementation inside ALPACA are interface tags motivated by the narrowband approach [2, 30, 159]. They represent the interface and its surrounding band. Only the thereby tagged region is influenced by the discretization of the LS algorithm. Additionally, these interface tags enable the MR algorithm to accurately locate the interface, guaranteeing that it is always fully refined and the MR can react adequately to the interface movement, deformation, and other topological changes.

A second important aspect is the extendability and flexibility of the simulation framework. The reasons, therefore, are manifold. The development of numerical schemes requires comparison with related schemes. Ideally, this is done based on the same implementation and by easily switching schemes. In addition, when integrating a simulation environment in an adaptive numerical experimentation workflow, a user-friendly application programming interface (API) eases investigations. These goals are achieved by a modular implementation using modern object-oriented programming (OOP) paradigms that ease extendibility and the development of new methods.

Summary of: **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*, Volume 272, 108246, 2022. [94]

This paper describes the integration of the LS algorithm into the MR scheme [94] implemented in the compressible CFD simulation framework ALPACA [3]. Modular building blocks for the multi-phase discretization of the governing equations presented in Section 2.1.2 and the discretization of the interface using the LS method presented in Section 2.1.3 extend the pre-existing single-phase implementation. Alternative discretization schemes of e. g. the interface extrapolation for the GFM or the LS reinitialization are presented where available.

The introduction of the publication highlights various applications for compressible multi-phase flows, introduces different numerical schemes for multi-phase flow simulation, motivates the usage of the LS method, gives an overview of spatial adaption techniques, and motivates the implementation of the LS algorithm in combination with a block-based MR scheme. Furthermore, relevant implementation aspects are stated.

Then, the compressible NSE for multiple distinct and immiscible phases and their numerical discretization are presented. The LS method is employed to sharply represent the interface. Fluxes across the interface are considered by conservative interface exchange terms that are evaluated by solving a two-material Riemann problem at the interface. The evolution of the interface with the flow field is obtained by solving an additional advection equation for the LS and a subsequent reinitialization procedure to ensure the validity of the LS. Boundary conditions at the interface are imposed using the GFM method. Various numerical discretization schemes for these building blocks of the LS algorithm are presented.

Next, the efficient integration of the LS method into the MR algorithm proposed by [93] is described. The narrowband approach limits the computations relevant for the interface discretization to a small band surrounding the interface. The location of the narrowband is dynamically tracked by interface tags, which can also be used to sense approaching and vanishing interfaces. This fact is leveraged by the MR algorithm to adjust the block structure to the dynamic evolution of the interface.

The remainder of the publication investigates the implementation's capability by simulating various challenging test cases. First, the convergence of the overall algorithm is investigated using an academic test case. Second, multi-phase flow configurations are simulated and compared to numerical and experimental reference data from literature. Using a 2D Richtmyer-Meshkov instability (RMI), we investigate the evolution of characteristic interface locations and compare them to numerical results from literature. Thereafter, 2D and 3D shock-bubble interactions (SBIs) simulations are presented in convergent and divergent configurations. A convergent configuration refers to the case when the acoustic impedance of the bubble is greater than for the surrounding liquid. The divergent configuration denotes the opposite. The results show excellent agreement with reference data from literature. The simulated cases highlight the capability of simulating resolutions of up to four billion effective cells and complex material models such as the Sutherland law to express temperature-dependent viscosities. Third, the parallelization for different combinations of the numerical building blocks is evaluated by scaling runs on up to twenty-four thousand cores. Last, compute and memory compression of the coupled LS and narrowband approach are assessed for the 3D cases presented in the work. Therefore, different metrics have been proposed that allow measuring savings in terms of compute time and memory consumption. They are defined to distinguish savings due to the MR compression and the interface-tag-based narrowband approach. For test cases with many MR refinement levels, high compression metrics >90% are observed.

For this publication, I have implemented the LS algorithm. I have conducted the simulations presented in sections 8.1 to 8.6 and predominantly written sections 2, 3, and 8.5 to 8.6. 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.

3.2 Modeling dendritic crystal growth using stochastic multi-fidelity surrogate models

The stochastic surrogate model is a crucial component of the adaptive experimentation workflow highlighted in Fig. 1.2. It mimics the behavior of the investigated physical system and thus heavily influences the quality of the evaluated target. A commonly used class of surrogate models are GPs, which are inherently able to quantify predictive uncertainty. To reduce the required high-fidelity data, the potential of MF modeling based on resolution fidelities for building stochastic surrogate models for complex multi-phase flow phenomena is explored. The MF model is based on MFDGPs [35] in combination with input warping [188]. The implementation of the MFDGPs is realized using the open-source Gaussian process library GPyTorch developed by Gardner et al. [65]. Their implementation provides a performant framework for scalable, flexible, and modular GP models and is based on the open-source high-performance deep learning library PyTorch [156].

The dependency of the steady-state tip velocity of a single crystal on process- and material parameters states the physical problem for which the surrogate models is constructed. The simulations therefore are conducted using the semi-implicit conservative sharp-interface method for liquid-solid phase transition developed by Kaiser et al. [108]. Their implementation extends the LS algorithm presented in the preceding Section 3.1.

Summary of: **Josef M. Winter, Jakob W. J. Kaiser, Stefan Adami, Nikolaus A. Adams:** *Stochastic multi-fidelity surrogate modeling of dendritic crystal growth*. Computer Methods in Applied Mechanics and Engineering, Volume 393, 114799, 2022. [213]

This work employs an enhanced multi-fidelity Gaussian process framework to generate stochastic response surfaces for the tip velocity of a single crystal. It combines many rough estimates of the tip velocity (low-fidelity model) with only a few high-fidelity measurements. Data for all fidelity levels is created using the simulation framework ALPACA, introduced in Section 3.1, applying the model developed by Kaiser et al. [108] to simulate two-dimensional crystal growth. The fidelity hierarchy bases on coarse-grid approximations of high-fidelity numerical simulations.

Dendritic microstructures and their defects significantly impact the mechanical properties of final products, particularly those manufactured through additive manufacturing processes. Modeling the formation process of these final products, encompassing all scales from single dendritic growth to evolving microstructures and ultimately the part scale, poses a multiscale challenge. The introduction of this publication emphasizes the importance of simplified surrogate models, particularly those linking the operational state of a crystal with process and material parameters. They allow for bridging the various hierarchy levels of this multiscale problem. The attractiveness of MFDGPs for this purpose is highlighted, as they enable the construction of sophisticated surrogates capable of accurately capturing the response of complex engineering systems. Leveraging the potential of MF modeling, MFDGPs help reduce the cost associated with dataset generation. The error-based refinement strategy of the MR algorithm, implemented in ALPACA, facilitates the definition of a hierarchy of resolution fidelities using the same simulator.

The following section of the publication presents the physical and mathematical model to describe the operating point of a single crystal, which is characterized by its steady-state tip velocity and radius. The parameters that influence the growth process are presented. Those are normalized undercooling, capillary anisotropy and the thermal conductivity ratio between solid and melt, cf. Section 1.3. The relation between the tip velocity and these parameters is identified as the system for which a surrogate model is constructed. The description of the numerical model solving the crystal-growth process allows for deriving a hierarchy of three fidelity levels based on coarse grid approximations of the high-fidelity solution. A convergence analysis and comparison with data from literature ensures the validity of the

high-fidelity model.

The MF models considered in the publication are based on autoregressive schemes presented in Section 2.2.2. Fidelity hierarchies for physical systems often show nonlinear and space-dependent relationships between fidelity levels. MF models based on the linear autoregressive scheme of Kennedy & O'Hagan [114] feature deficiencies for such problems as highlighted by Perdikaris et al. [161]. Thus, the publication combines MFDGPs [35], which base on a generalized autoregressive scheme, cf. Eq. (2.45), with input warping [188] to regularize the model-parameter space.

The dataset generated for the publication relies on three fidelity levels. The thorough analysis of the datasets unveils that the underlying problem of single dendritic growth features a nonlinear relationship between the selected fidelity levels. It thus justifies the choice of the selected MF modeling strategy. Also, sensitivities of the tip velocity on model parameters can be identified. Based on the presented dataset, surrogate models with varying compositions of the training dataset are generated.

The remainder of the publication analyses the created surrogate models in detail. First, it is shown that input warping improves the predictive accuracy of MF models. When increasing the number of data points used for fitting the model, the predictive error of the trained models converges. Second, the composition of the dataset is varied and its influence on the predictive accuracy of models combining MFDGPs and input warping is investigated. The analysis highlights that three-fidelity modeling does not outperform two-fidelity modeling. Furthermore, two-fidelity modeling drastically outperforms single-fidelity modeling. When increasing the number of high-fidelity samples used for training, the predictive error converges. The same trends hold when increasing the number of low-fidelity samples. Analyzing the computational cost highlights that the drastically improved accuracy of two-fidelity models comes with a negligible cost overhead compared to single-fidelity modeling.

As lead author of this publication, I predominantly wrote the script including the shown visualizations and addressed the reviewer comments. I have implemented the MF model, combining MFDGPs and input warping, inside the modular open-source GP library GPyTorch. Also, I selected the simulation setups and implemented the dataset generation routines. I defined the hierarchy of fidelities, conducted the dataset generation and its analysis, trained the MF models, and evaluated their properties. 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 Solving the inverse Stefan problem for dendritic crystal growth using Bayesian optimization techniques

This publication covers the complete iterative workflow for adaptive numerical experimentation and solves the inverse Stefan problem for single dendritic growth. Again, the relation between the steady-state tip velocity and process- and material parameters is considered. Solving the inverse problem aims to identify the process- and material parameters that yield a targeted, prescribed tip velocity. The algorithm used for the adaptive experimentation workflow is MF BO. It is highlighted that the proposed method consists of several building blocks that can be exchanged flexibly to adjust the algorithm for various scenarios.

MF BO consists of three crucial components. Those are the simulation framework that models the forward problem, the stochastic surrogate model capturing the input-output relation of the forward problem, and a policy that guides the algorithm towards the solution. As simulation framework ALPACA is used. The posterior predictive distribution of the surrogate model is required to evaluate the acquisition function. Thus, the surrogate model and acquisition function must be compatible with each other. In the context of this work, MF GPs that include the MF aspect in the kernel function, cf. Section 2.2.2, in combination with the q-cfKG acquisition strategy, cf. Section 2.3.2, are employed.

The implementation of the GP model is again based on GPyTorch. The acquisition function is implemented based on BoTorch, developed by Balandat et al. [12], a modern programming framework for Bayesian optimization. Recent advances in the field of machine learning (ML) lead to the development of open-source libraries for efficient tensor calculations and the training of deep-learning models. These libraries are based on the concept of automatic differentiation and allow leveraging the full performance of graphics processing unit (GPU) hardware. PyTorch is a prominent representative. BoTorch leverages the powerful framework of PyTorch and provides a modular implementation of various acquisition functions. Furthermore, it interfaces with GPyTorch as backbone for the GP models.

Summary of: **Josef M. Winter, Rim Abaidi, Jakob W. J. Kaiser, Stefan Adami, Nikolaus A. Adams:** *Multi-fidelity Bayesian optimization to solve the inverse Stefan problem*. Computer Methods in Applied Mechanics and Engineering, Volume 410, 115946, 2023.

This paper addresses the solution of the inverse Stefan problem for single dendritic growth utilizing MF BO. In this approach, multiple low-fidelity estimates of a solidification problem are amalgamated with a limited number of high-fidelity measurements to construct a comprehensive MF GP. The inverse problem is tackled by employing the GP model within a BO framework based on a MF KG acquisition function. The target function for the optimization process quantifies the deviation between observed and target values and is reformulated as a composite function. This reformulation significantly enhances the stability of the optimization procedure.

The motivation for this work, as discussed in the preceding Section 3.2, arises from the influence of dendritic growth patterns on material properties, particularly in modern additive manufacturing processes. Solutions to the inverse problem of dendritic growth can contribute to designing materials with desired properties. Following the motivation, an extensive literature review is provided, highlighting that while several authors have employed optimization-based approaches for solving the inverse Stefan problem, they often face challenges due to specific mathematical formulations of the target function or computational complexity. The publication identifies BO as a promising approach since it offers cost-efficient optimization of black-box functions and efficient integration with MF modeling. The q-cfKG acquisition function, optimizing a benefit-cost ratio, is employed in this publication.

The subsequent section introduces the methodology employed in this study. The SF and MF optimization approaches are detailed and a composite structure for the target function is proposed. The composite structure allows separating the surrogate model for observed data from the target function.

The target function may be non-differentiable at certain points in the domain since it quantifies the absolute deviation between observed and target values. Traditional kernel functions of GP models may be inadequate for such relations. Consequently, constructing the surrogate model for simulator observations and considering the loss only during optimization and acquisition function evaluation circumvents this issue. This approach brings additional advantages, including the reusability of observed data points and improved physical interpretability of the surrogate model.

The optimization problem's solution is approximated by optimizing the target function using the posterior mean of the GP surrogate model. To assess the accuracy of the estimated solution, it is necessary to evaluate the numerical solver at the currently suggested optimum and subsequently compare it with the recommended solution. The proposed methodology allows augmenting the dataset used to train the GP by incorporating the data point generated during the evaluation process. This augmentation introduces additional feedback into the iterative optimization process, promoting exploitation beyond the batches suggested by the acquisition function that tend to be exploratory. This procedure is referred to as recommendation fitting.

The performance of the proposed adaptive experimentation framework is evaluated in two settings. First, the inverse problem for the growth of a planar solidification front is considered in a single-fidelity setting. The results show that the modified acquisition strategy, including recommendation fitting, shows better convergence results compared to solely relying on the KG acquisition function. Second, the inverse problem for the growth of a single dendrite, already investigated in the preceding Section 3.2, is considered. There, it is found that MF modeling improves convergence properties and cost-efficiency compared to a SF setting, especially when access to high-fidelity data is limited and low- and high-fidelity models are well correlated in the region of the input space where the solution is located.

For this publication, I have implemented the MF BO. I have conducted the simulations and analysis for the dendritic growth problem. I have predominantly written section 7. Furthermore, I have equally contributed to sections 1 to 5 and 8 as well as to the abstract. I have extensively reviewed the remaining sections. The visualizations in the respective sections were created by me. My first co-author has conducted the work presented in the remaining sections, which she has predominantly written and has equally contributed to sections 1 to 5 and 8 as well as to the abstract. She has extensively reviewed the sections I wrote. The remaining co-authors have coordinated, supervised, and secured funding for the project. They have also reviewed the manuscript.

Chapter 4

List of Publications

4.1 Publications comprising this thesis

- **Josef M. Winter, Rim Abaidi, Jakob W. J. Kaiser, Stefan Adami, Nikolaus A. Adams:** *Multi-fidelity Bayesian optimization to solve the inverse Stefan problem.* Computer Methods in Applied Mechanics and Engineering, Volume 410, 115946, 2023. [212]
- **Josef M. Winter, Jakob W. J. Kaiser, Stefan Adami, Nikolaus A. Adams:** *Stochastic multi-fidelity surrogate modeling of dendritic crystal growth.* Computer Methods in Applied Mechanics and Engineering, Volume 393, 114799, 2022. [213]
- **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, Volume 272, 108246, 2022. [94]

4.2 Further publications

4.2.1 Peer-reviewed journal publications

- **Jakob W. J. Kaiser, Josef M. Winter, Stefan Adami, Nikolaus A. Adams:** *Investigation of interface deformation dynamics during high-Weber number cylindrical droplet breakup.* International Journal of Multiphase Flow, Volume 132, 103409, 2020. [110]
- **Yiqi Feng, Felix S. Schraner, Josef M. Winter, Nikolaus A. Adams:** *A multi-objective Bayesian optimization environment for systematic design of numerical schemes for compressible flow.* Journal of Computational Physics, Volume 468, 111477, 2022. [49]
- **Yiqi Feng, Felix S. Schraner, Josef M. Winter, Nikolaus A. Adams:** *A deep reinforcement learning framework for dynamic optimization of numerical schemes for compressible flow simulations.* Journal of Computational Physics, Volume 493, 112436, 2023. [50]
- **Vladimir Bogdanov, Felix S. Schraner, Josef M. Winter, Stefan Adami, Nikolaus A. Adams:** *A level-set-based sharp-interface method for moving contact lines.* Journal of Computational Physics, Volume 467, 111445, 2022. [20]
- **Nico Fleischmann, Josef M. Winter, Stefan Adami, Nikolaus A. Adams:** *High-order modeling of interface interactions using level sets.* GAMM-Mitteilungen, Volume 45, e202200012, 2022. [56]

4.2.2 Conference proceedings

- **Josef M. Winter, Felix S. Schraner, Nikolaus A. Adams:** *Iterative Bayesian Optimization of an Implicit LES Method for Underresolved Simulations of Incompressible Flows.* 10th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 10, 2017. [[215](#)]
- **Josef M. Winter, Jakob W. J. Kaiser, Stefan Adami, Nikolaus A. Adams:** *Numerical investigation of 3D drop-breakup mechanisms using a sharp interface level-set method.* 11th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 11, 2019. [[214](#)]
- **Yiqi Feng, Felix S. Schraner, Josef M. Winter, Nikolaus A. Adams:** *Bayesian Optimization on Fifth-Order Targeted ENO Scheme for Compressible Flows.* 15th World Congress on Computational Mechanics (WCCM-XV) and 8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII), 2023. [[48](#)]

Discussion with Respect to the State of the Art and Conclusion

Numerically investigating multi-phase flow phenomena remains challenging, even today. Reasons for this include the multi-scale nature of such problems involving multiple spatial and temporal scales, configurations that might be inherently unstable, and the presence of phase interfaces that potentially dominate investigated flow configurations. As a consequence, the simulation frameworks used to investigate these problems must provide sophisticated numerical models that accurately and stably capture relevant flow details and possibly existing phase interfaces. Also, its implementation must be highly performant, and leverage modern computing hardware to enable highly-resolved simulations. Realistic flow problems typically depend on a set of parameters. Their influence on relevant output quantities has to be investigated. Adaptive experimentation provides a powerful tool for this purpose. Based on an initially available set of observations, it iteratively updates prior beliefs about the system to decide which configuration should additionally be investigated. In this work, a simulation framework for multi-phase flow based on the level-set method is developed and integrated into an adaptive experimentation workflow based on multi-fidelity Bayesian optimization. It covers contributions to the three main components of such an algorithm which are listed in the following.

First, a sharp-interface model is integrated into the compressible multiresolution [93] simulation framework ALPACA [3] with local time stepping [111]. A level-set function [39, 152, 198] gives a sharp representation of phase interfaces and allows solving the governing equations for each phase separately. Conservative interface interaction terms consider the exchange of mass, momentum, and energy over the interface [96, 97, 136, 199]. Boundary conditions at the interface are imposed using the GFM method [47]. The efficiency of the level-set algorithm is ensured using an interface-tagging system inspired by the narrowband approach [2, 30, 159]. Interface tags dynamically track the near-interface region. Based on them, the expensive level-set-related calculations are limited to the near-interface region. Also, the memory usage of the algorithm can be reduced. The GFM requires information about all fluids in cells close to the interface. The interface tags ensure that memory for both fluids is only allocated close to the interface. Away from it, it is sufficient to allocate memory for the respective present fluid. The developed simulation framework is published under an open-source license for reuse by the scientific community. Various other parallel open-source research codes for compressible multi-phase flows exist [8, 26, 64, 85, 170, 181, 210]. However, they mostly use diffuse-interface methods and do not provide sharp interface level-set interface models for compressible multi-phase flows. Also, they do not employ LTS. Sharp interface level-set simulations of compressible multi-phase flows using spatial adaption techniques are presented in [78, 148], however, their code bases are not freely available. The developed sharp interface level-set model for compressible multi-phase flows is hence unique and has already been employed in several studies [107, 109, 214]. Also, the careful algorithmic design and implementation did not put limitations on the extendability of the framework but fostered the extension of existing numerical schemes and the development of new numerical schemes [53].

Second, a novel stochastic-surrogate modeling approach combining input warping techniques [188] and multi-fidelity deep Gaussian processes [35] is developed. It is applied to model the tip velocity of a single dendrite. The dataset is generated using the semi-implicit conservative sharp interface method for

non-equilibrium liquid-solid phase transition of pure melts developed by Kaiser et al. [108] that is readily available in the simulation framework ALPACA. The dataset contains a hierarchy of resolution fidelities that leverages the error-based refinement strategy of the MR algorithm. The surrogate models trained using the novel approach are significantly more accurate compared to single-fidelity models, especially when a limited amount of high-fidelity data is available. Input warping further improves the accuracy of the models. The resulting models accurately capture the tip velocity's dependence on process- and material parameters. The stochastic nature of the model allows for quantifying the uncertainty associated with the predictions. The employed multi-fidelity modeling strategy relies on Gaussian Process models that allow capturing nonlinear and space-dependent relationships between fidelity levels. This fact is crucial for the dendritic growth problem, since the generated dataset evinces this property. Babaei et al. [10] investigate mixed convection flows using multi-fidelity Gaussian processes. Their problem also shows nonlinear and space-dependent relationships between fidelity levels. The authors find, that in regions where low-fidelity are not informative, high-fidelity samples are important to achieve accurate results. Perdikaris et al. [161] assume that this is caused by the limited expressivity of the linear autoregressive schemes used by [10]. To verify their assumption, Perdikaris et al. [161] use multi-fidelity Gaussian processes based on the generalized autoregressive scheme. They find that the ability to capture a space-dependent relationships between fidelity levels yields consistently better results. Wiens et al. [211] apply the same multi-fidelity Gaussian process model to model chemical energy surfaces. A drawback of the model employed in [161] and [211] is the fact, that it requires nested datasets. This implies that at locations where high-fidelity data is available, low-fidelity data must also be available. This restriction does not hold for the proposed method based on MFDGP. Also, to the best of the author's knowledge, MFDGP in combination with input warping techniques have not been used to analyze flow problems before.

Third, a novel multi-fidelity Bayesian optimization framework combining composite BO, multi-fidelity modeling using the cost-aware knowledge gradient acquisition function, and recommendation fitting is proposed. It is used to solve the inverse Stefan problem for single-dendritic growth. Process and material parameters that yield a targeted tip velocity are identified. Leveraging multi-fidelity modeling and recommendation fitting strongly improves the convergence properties of the algorithm and significantly speeds up the process of obtaining the solution. Formulating the target function as a composite function improves the interpretability of the surrogate model and eases re-using data samples and pre-trained surrogate models. Solving the inverse Stefan problem is an important topic investigated by several authors [7, 23, 74, 75, 103, 104, 146, 192, 207]. However, many are limited to one-dimensional settings, one-phase problems, or rely on specific mathematical formulations of the problem. These studies are extended to more complex cases involving two-dimensional setups and multiple phases by Słota [187] and Hetmaniok et al. [88]. Both studies rely on optimization-based approaches. Hetmaniok et al. [88] emphasize the computational complexity when employing optimization-based approaches to realistic problems and highlight the importance of efficient algorithms. The proposed method tackles this challenge for the first time using the presented approach based on multi-fidelity Bayesian optimization, and observes a significant reduction of computational complexity due to the multi-fidelity modeling. Multi-fidelity Bayesian optimization for flow problems has also been proposed by Perdikaris et al. [160] and Han et al. [77]. However, their models rely on the linear autoregressive scheme and require nested datasets. The approach proposed here is more flexible and handles arbitrary datasets. Flexibility also exists in the fidelity hierarchy. The proposed algorithm makes no assumptions on the type of fidelity levels and thus can be easily extended to also include data from other information sources, such as experiments. Applying the algorithm in the rapid and efficient design of new materials with targeted properties is a perspective for future work.


In conclusion, this thesis has introduced an adaptive numerical experimentation framework designed to efficiently and flexibly model and simulate complex flow phenomena. Novel approaches for the foundational building blocks of the framework are presented. The level-set algorithm implemented in the compressible multi-phase flow simulation framework ALPACA allows efficient and highly detailed simu-


lations of complex flow phenomena. It has proven to be a powerful simulator for the data-driven investigation of flow phenomena. ML architectures combining MFDGPs with input warping allow to efficiently generate surrogate models for simulation data. Additionally, multi-fidelity modeling has been used to develop a highly adaptable and efficient Bayesian optimization framework. The versatility of the proposed framework makes it an attractive candidate for several applications in research and industry.

Appendix A

Peer-Reviewed Journal Publications

A.1 ALPACA - a level-set based sharp-interface multiresolution solver for conservation laws

Sign in/Register?🔍



ALPACA - a level-set based sharp-interface multiresolution solver for conservation laws

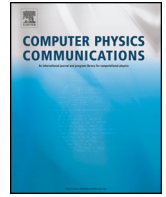
Author: Nils Hoppe, Josef M. Winter, Stefan Adami, Nikolaus A. Adams
Publication: Computer Physics Communications
Publisher: Elsevier
Date: March 2022

© 2021 Elsevier B.V. All rights reserved.

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: <https://www.elsevier.com/about/our-business/policies/copyright#Author-rights>

BACK CLOSE WINDOW



ALPACA - a level-set based sharp-interface multiresolution solver for conservation laws ^{☆, ☆☆}



Nils Hoppe ^{*,1}, Josef M. Winter ^{*,1}, Stefan Adami, Nikolaus A. Adams

Chair of Aerodynamics and Fluid Mechanics, Technical University of Munich, Boltzmannstr. 15, 85748 Garching bei München, Germany

ARTICLE INFO

Article history:

Received 12 March 2021
Received in revised form 19 November 2021
Accepted 23 November 2021
Available online 1 December 2021

Keywords:

Computational fluid dynamics
Compressible Navier-Stokes equations
Sharp interface
Temporal and spatial adaptivity
HPC
C++20

ABSTRACT

ALPACA is a simulation environment for simulating hyperbolic and (incompletely) parabolic conservation laws with multiple distinct and immiscible phases. As prominent example, consider the compressible Navier-Stokes equations (NSE). Solutions to these equations give insight and understanding of many important engineering applications. Numerical simulations of nonlinear parabolic systems of equations are very challenging for their complex nonlinear dynamics including the propagations of discontinuities such as shocks and phase interfaces. Accurate predictions require high temporal and spatial resolutions for such multi-scale problems. We utilize low dissipation high-resolution methods to capture the dynamics inside the separate phases. Their interaction is modeled by a sharp-interface level-set method with conservative interface-interaction. This allows to accurately locate the interface position and to easily prescribe arbitrary coupling conditions. We tackle the resulting immense computational loads by using a block-based multiresolution (MR) algorithm and adaptive local time stepping. The level-set treatment is integrated into the MR algorithm with little overhead by employing a smart tagging system and adaptive storage of the fluid data in the MR nodes. We embed these methods in a C++20 object-oriented modular framework using state-of-the-art programming paradigms. Furthermore, our implementation is capable to exploit the multiple levels of parallelism in modern high-performance computing (HPC) systems efficiently. We demonstrate the capabilities of our framework by simulating a variety of compressible multi-phase flow problems. Problem-sizes are of $\mathcal{O}(10^{10})$ effective degree of freedom (DOFs). By the use of MR, we typically achieve memory and compute compressions of $> 90\%$. We demonstrate near-optimal parallel performance for scaling runs using $\mathcal{O}(10^4)$ cores, regardless of the employed numerical models.

Program summary

Program Title: ALPACA - Adaptive Level-set Parallel Code Alpaca

CPC Library link to program files: <https://doi.org/10.17632/5zr3sg83ct.1>

Developer's repository link: <https://gitlab.lrz.de/nanoshock/ALPACA>

Licensing provisions: GPLv3

Programming language: C++20

Supplementary material: Code: Copy of the git repository, Videos: Air-helium shock bubble interaction (front and back view), Air-R22 shock bubble interaction, Three bubble shock interface interaction.

Nature of problem: Numerical simulation of conservation laws such as the compressible Navier-Stokes equation with several interacting gaseous and liquid phases remains challenging even today. These flows often involve singularities such as shocks and interfaces as well as instabilities driven by their interaction. The inherent highly nonlinear dynamics of those systems leads to a broad range of temporal and spatial scales that have to be resolved. There exists a variety of mutually exclusive numerical models that are able to tackle these challenges. Those, however, are computationally expensive and require computational power that is offered only by large-scale state-of-the-art distributed-memory machines.

Solution method: We have developed a modular simulation environment for conservation laws allowing exchange of numerical methods without loss of parallel performance. Computational efficiency is enhanced by employing multi-resolution schemes with adaptive local-time stepping. Our block-based

[☆] The review of this paper was arranged by Prof. N.S. Scott.

^{☆☆} This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding authors.

E-mail addresses: nils.hoppe@tum.de (N. Hoppe), josef.winter@tum.de (J.M. Winter).

¹ Both authors have contributed equally.

implementation of these schemes is parallelized using the Message Passing Interface and exploits vectorization capabilities of the compute hardware. To simulate distinct and immiscible phases inside the computational domain, we utilize a sharp-interface level-set method. The level-set method allows to accurately locate the interface position and to easily prescribe arbitrary coupling conditions. The narrow-band approach reduces the computational load of the level-set method. We extend this method by a smart tagging system that exploits the block-based nature of the algorithm and further reduces the computational load. The simulation framework is written in modern C++20 and provides a Python interface for integration in Machine Learning and Uncertainty Quantification toolchains. We use parallel HDF5 in combination with XDMF to output field quantities. CMake is used as build system. We have completely annotated the source code using doxygen-style comments, allowing automated documentation generation in different formats. The source code is equipped with CI/CD-automated unit tests and an exhaustive integration test suite.

Additional comments including restrictions and unusual features: ALPACA relies on open-source third-party libraries for input and testing. These are packaged as git submodules and automatically integrated. ALPACA is tested on Linux and macOS.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Conservation equations are used to model a wide variety of scientific and engineering problems, ranging from traffic flow [1–3], the shallow water equations [4–6], electromagnetism [7] to gasdynamics [8]. For our research, the compressible Navier-Stokes equations (NSE) are of particular interest to describe fluid mechanical problems such as the generation of nanoparticles [9,10], cavitation phenomena [11] or biomedical procedures like lithotripsy [12,13]. The multi-phase nature of such phenomena poses additional challenges to numerical models, as the interface has to be properly modeled. We provide the simulation environment “Adaptive Levelset Parallel Code Alpaca (ALPACA)” to solve such kind of problems. During implementation, we considered the following aspects, which we will discuss in detail below: Single-phase discretization of the compressible NSE, modeling of the interface, efficient compression algorithms on modern high-performance computing (HPC) systems, and modular implementation of different numerical methods.

The highly nonlinear dynamics of compressible flows inside a single phase require sophisticated and thus compute intensive numerical schemes. Examples include Riemann solvers in combination with high-order low dissipation weighted essentially non-oscillatory (WENO) reconstruction stencils or alike. Many such competing schemes are available [14–19] and have to be chosen according to the precise problem at hand. Hence, support for a wide variety of these schemes is beneficial to researchers and should be available in modern software frameworks.

When multiple phases govern a physical problem, the numerical description typically relies on either interface tracking or interface capturing schemes. The latter describe interfaces as discontinuities in material properties by using advected volume fractions [20]. For conservative schemes, they guarantee discrete conservation. However, interfaces might be smeared by numerical diffusion [21–23]. In contrast, interface tracking schemes allow a sharp representation of interfaces. The most prominent representatives are free-Lagrange [24,25], front tracking [26,27] and level-set methods [28]. In our framework, we use the level-set approach since it is able to naturally handle changes in topology [29], eases calculating geometric quantities [30], and allows imposing interfacial physics straightforwardly [20]. For a more extensive overview about level-set methods and comparison to other interface modeling approaches we refer to [29,30].

As described above, solving the three-dimensional NSE with several interacting phases requires resolving a large range of physically relevant scales [31–33]. Hence, resolutions easily reach $\mathcal{O}(10^9)$. To handle such resolutions, the usage of modern distributed-memory HPC systems is essential together with time and/or

space adaptive compression schemes. Unfortunately, some compression techniques hinder the necessary multiple-levels of parallelization needed to efficiently use the hardware of HPC systems [34,35]. Here, we focus on multiresolution (MR) compression [36] in fully adaptive formulation [37,38]. Other possible techniques are the adaptive mesh refinement (AMR) [39], Wavelet-Galerkin [40] methods, or (second-generation) wavelet-collocation methods [41]. Compared to AMR, a larger compression is achieved via MR and the error can be better controlled [42]. Compared to Wavelet-Galerkin methods, the treatment of nonlinearities is straightforward [31]. In contrast to Wavelet-Collocation, the (data) locality is stronger [43]. To obtain an efficient HPC implementation, the classical adaptive MR scheme of Harten needs to be improved. The original one-dimensional hyperbolic formulation needs to be generalized to higher dimensions [44] and to parabolic problems [45]. Additionally, memory compression needs to be added [37,38] and temporal compression needs to be included via (adaptive) local time stepping (LTS) [46,47]. We further translate the concept of blocking methods from wavelet-collocation [48] and AMR [49] approaches to the MR algorithm. This block-based MR algorithm provides a HPC friendly compression. Also, this allows compiler generated single instruction multiple data (SIMD) parallelization [50]. We combine this approach with narrow-band techniques [51,52] based on ideas of [53] to reduce computational overhead of the level-set method. We deduce a tagging system which extends the narrow-band technique into the MR and thus reduces the consumed memory considerably.

We have seen that a competitive simulation environment should offer a collection of state-of-the-art discretizations schemes for both single- and multiphase problems with efficient compute and memory compression. At the same time, efficient use of modern HPC architectures has to be ensured. Therefore, we developed our software in a modular fashion. The parallelization of the overall algorithm is decoupled from the numerical discretization schemes. This way the parallelization overhead is separated from the compute kernel implementation and the (parallel) compute performance is comparable across different modules. This modular setup together with the usage of modern object-oriented programming (OOP) paradigms eases extension and development of new methods.

The remainder of this paper is structured as follows. In Section 2 we give the governing equations. In Section 3 their numerical discretization and the numerical interface model are given. We describe our block-based MR in Section 5. Section 6 highlights its extension via our newly developed tagging system. We discuss implementation details in Section 7. Multi-phase validation cases and challenging problems are shown in Section 8. In Section 9 we state the resulting (parallel) performance with scaling runs on up

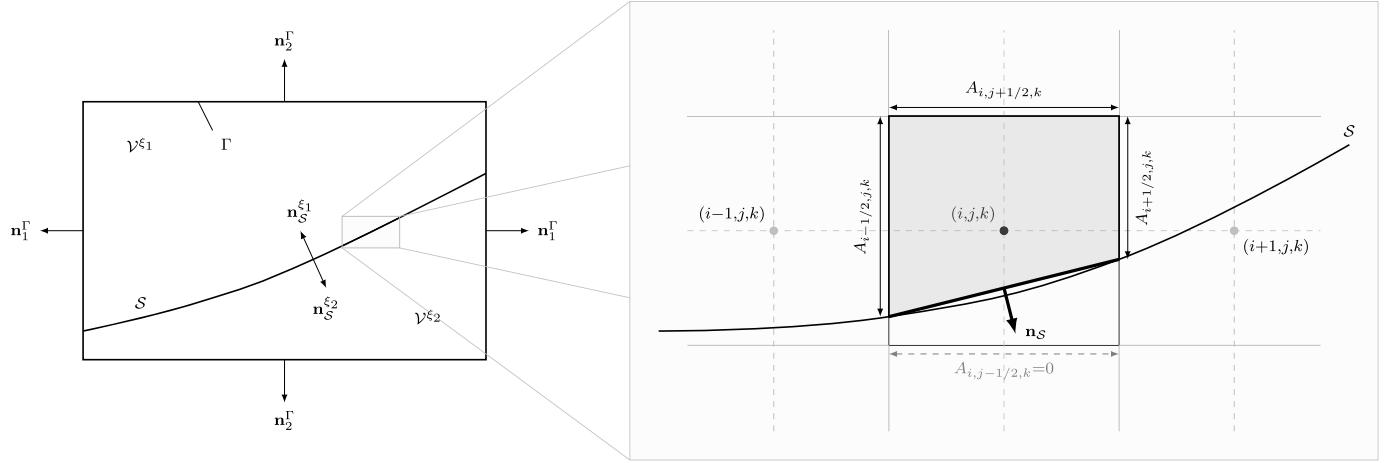


Fig. 1. Sketch of a material volume cut by an interface (left). The region framed by the dashed line is highlighted on the right and shows a finite volume cell $\Delta_{i,j,k}$. Note, the figure shows a two-dimensional slice in the i, j -plane.

to $\mathcal{O}(10^4)$ compute cores. Also, the achieved compression across the conducted test cases is shown. We conclude with a short summary and an outlook to future work.

2. Governing equations

The general form of a conservation law reads

$$\frac{\partial \psi}{\partial t} = -\operatorname{div} f(\psi) + s(\psi) \quad (1)$$

where ψ is the balanced quantity, $f(\psi)$ denotes a flux function and $s(\psi)$ a source term. We introduce \mathcal{D} as short notation for an arbitrary right-hand side.

2.1. Navier-Stokes equations

We focus on the compressible NSE given by the system of equations

$$\frac{\partial \rho}{\partial t} = -\operatorname{div} \rho \mathbf{u} \quad (2)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\operatorname{div} (\rho \mathbf{u} \otimes \mathbf{u} - \Pi) + \rho \mathbf{g} \quad (3)$$

$$\frac{\partial \rho E}{\partial t} = -\operatorname{div} (\rho E \mathbf{u} - \Pi \mathbf{u} - \mathbf{q}) + \rho \mathbf{g} \cdot \mathbf{u}. \quad (4)$$

Therein, ρ denotes the density, \mathbf{u} the velocity vector, \mathbf{g} the gravitational acceleration vector and $E = e + \mathbf{u}^2/2$ the specific total energy consisting of specific internal and kinetic energy. Heat fluxes are considered by

$$\mathbf{q} = \kappa \nabla T \quad (5)$$

with the thermal conductivity κ and temperature T . The stress tensor is given by

$$\Pi = -p\mathbf{I} + \mathbf{T} = -p\mathbf{I} + \mu \left(\nabla \otimes \mathbf{u} + (\nabla \otimes \mathbf{u})^T - \frac{2}{3} \operatorname{div}(\mathbf{u})\mathbf{I} \right) \quad (6)$$

with pressure p , dynamic viscosity μ , identity matrix \mathbf{I} and viscous stress tensor \mathbf{T} .

In integral formulation, Eqs. (2) to (4) read

$$\int_{\mathcal{V}} \frac{\partial \rho}{\partial t} dV = - \oint_{\partial \mathcal{V}} (\rho \mathbf{u}) \cdot \mathbf{n} dA \quad (7)$$

$$\begin{aligned} \int_{\mathcal{V}} \frac{\partial \rho \mathbf{u}}{\partial t} dV &= - \oint_{\partial \mathcal{V}} (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{I} - \mathbf{T}) \cdot \mathbf{n} dA + \int_{\mathcal{V}} \rho \mathbf{g} dV \quad (8) \\ \int_{\mathcal{V}} \frac{\partial \rho e}{\partial t} dV &= - \oint_{\partial \mathcal{V}} (\rho e \mathbf{u} + p\mathbf{I} - \mathbf{T} \mathbf{u} - \mathbf{q}) \cdot \mathbf{n} dA + \int_{\mathcal{V}} \rho \mathbf{g} \cdot \mathbf{u} dV, \end{aligned} \quad (9)$$

where \mathcal{V} denotes a material volume. Rearranging and summarizing terms in this set of equations yields the flux-based formulation

$$\int_{\mathcal{V}} \frac{\partial \mathbf{U}}{\partial t} dV = - \oint_{\partial \mathcal{V}} (\mathbf{F}^c + \mathbf{F}^\mu + \mathbf{F}^q) dA + \int_{\mathcal{V}} \mathbf{S} dV \quad (10)$$

for the state vector $\mathbf{U} = [\rho \quad \rho \mathbf{u} \quad \rho e]^T$ with convective, viscous and heat-flux densities

$$\mathbf{F}^c = \begin{bmatrix} (\rho \mathbf{u}) \cdot \mathbf{n} \\ (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{I}) \cdot \mathbf{n} \\ (\rho e \mathbf{u} + p\mathbf{I}) \cdot \mathbf{n} \end{bmatrix}, \quad \mathbf{F}^\mu = \begin{bmatrix} 0 \cdot \mathbf{n} \\ -\mathbf{T} \cdot \mathbf{n} \\ -\mathbf{T} \mathbf{u} \cdot \mathbf{n} \end{bmatrix}, \quad \text{and} \quad (11)$$

$$\mathbf{F}^q = \begin{bmatrix} 0 \\ 0 \\ \mathbf{q} \cdot \mathbf{n} \end{bmatrix}, \quad (11)$$

respectively. The volume-force vector reads

$$\mathbf{S} = \begin{bmatrix} 0 \cdot \mathbf{n} \\ (\rho \mathbf{g}) \cdot \mathbf{n} \\ (\rho \mathbf{g} \cdot \mathbf{u}) \cdot \mathbf{n} \end{bmatrix}. \quad (12)$$

For multi-phase flows, we consider a material volume consisting of two distinct and immiscible fluid phases ξ_a , $i \in [1, 2]$. They occupy non-overlapping subdomains \mathcal{V}^{ξ_a} , $i \in [1, 2] \mid \mathcal{V} = \mathcal{V}^{\xi_1} \cup \mathcal{V}^{\xi_2}$, and are separated by an internal surface S representing the phase interface. We sketch such a material volume in Fig. 1. Therein, $\mathbf{n}_S^{\xi_a}$ denotes the interface-normal vector pointing towards \mathcal{V}^{ξ_a} . If the superscript is omitted, $\mathbf{n}_S = \mathbf{n}_S^{\xi_1}$.

The NSE hold for each phase separately. This yields for Eq. (10)

$$\begin{aligned} \int_{\mathcal{V}^{\xi_1} \cup \mathcal{V}^{\xi_2}} \frac{\partial \mathbf{U}}{\partial t} dV &= - \oint_{(\partial \mathcal{V}^{\xi_1} \cup \partial \mathcal{V}^{\xi_2}) \setminus S} (\mathbf{F}^c + \mathbf{F}^\mu + \mathbf{F}^q) dA \\ &\quad - \oint_S (\mathbf{F}_S^c + \mathbf{F}_S^\mu + \mathbf{F}_S^q) dA \quad (13) \\ &\quad + \int_{\mathcal{V}^{\xi_1} \cup \mathcal{V}^{\xi_2}} \mathbf{S} dV \end{aligned}$$

Table 1
Quantities used for nondimensionalization.

Quantity	Nomenclature	Nondimensionalization
Length	x	$x^* = \frac{x}{l_{ref}}$
Density	ρ	$\rho^* = \frac{\rho}{\rho_{ref}}$
Velocity	u	$u^* = \frac{u}{u_{ref}}$
Temperature	T	$T^* = \frac{T}{T_{ref}}$
Time	t	$t^* = \frac{t u_{ref}}{l_{ref}}$
Pressure	p	$p^* = \frac{p}{\rho_{ref} u_{ref}^2}$
Energy	E	$E^* = \frac{E}{\rho_{ref} u_{ref}^2}$
Gravity	g	$g^* = \frac{g l_{ref}}{u_{ref}^2}$
Viscosity	μ	$\mu^* = \frac{\mu}{\rho_{ref} u_{ref} l_{ref}}$
Surface tension coefficient	σ	$\sigma^* = \frac{\sigma}{\rho_{ref} u_{ref}^2 l_{ref}}$
Thermal conductivity	κ	$\kappa^* = \frac{\kappa T_{ref}}{\rho_{ref} u_{ref}^3 l_{ref}}$
Specific gas constant	R_{spec}	$R_{spec}^* = \frac{R_{spec} T_{ref}}{u_{ref}^2}$

with the convective, viscous and heat transfer interface flux densities

$$\mathbf{F}_S^c = \begin{bmatrix} (\rho \mathbf{u}) \cdot \mathbf{n}_S \\ (\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n}_S \\ (\rho e \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n}_S \end{bmatrix}, \quad \mathbf{F}_S^\mu = \begin{bmatrix} \mathbf{0} \cdot \mathbf{n}_S \\ -\mathbf{T} \cdot \mathbf{n}_S \\ -\mathbf{T} \mathbf{u} \cdot \mathbf{n}_S \end{bmatrix}, \quad \text{and} \quad \mathbf{F}_S^q = \begin{bmatrix} 0 \\ 0 \\ \mathbf{q} \cdot \mathbf{n}_S \end{bmatrix}, \quad (14)$$

respectively.

Thermodynamic closure of the NSE is obtained by an equation of state (EOS) of the form

$$p = f(\rho, e) \quad (15)$$

that relates pressure, density and energy. In our framework we have implemented numerous EOSes as highlighted in Table 2.

We nondimensionalize the equations described above using reference states for the length l_{ref} , density ρ_{ref} , velocity u_{ref} and temperature T_{ref} , see Table 1.

3. Numerical modelling

The numerical modeling of the governing equations and the interface is described below.

3.1. Numerical model for the governing equations

We solve the balance equations numerically by partitioning \mathcal{V} into a disjunct set of cuboid finite volumes

$$\Delta_{i,j,k} \leq \mathcal{V} \mid \Sigma \Delta_{i,j,k} = \mathcal{V}; \quad i, j, k \in \mathbb{N} \quad (16)$$

with cell size Δx and cell volume $\Delta V = \Delta x^3$. A finite volume cell may contain an interface \mathcal{S} as exemplarily shown in Fig. 1. The cell $\Delta_{i,j,k}$ in the center of the figure is cut by an interface. We call such cells *cut cells*. We denote the linearized interface inside the cell $\mathcal{S}_{i,j,k}$. The volumetric portion of $\Delta_{i,j,k}$ which is covered by fluid ξ_a is denoted as $\alpha_{i,j,k}^{\xi_a}$ and indicated by the gray region in the figure. We define apertures $A_{i \pm \frac{1}{2}, j, k}^{\xi_a}(t)$, $A_{i, j \pm \frac{1}{2}, k}^{\xi_a}(t)$, and $A_{i, j, k \pm \frac{1}{2}}^{\xi_a}(t)$ to geometrically describe the cell-face fraction covered by ξ_a . With this we build the discrete sets of cells

$$\tilde{\mathcal{V}}^{\xi_a} = \left\{ \Delta_{i,j,k} \mid \alpha_{i,j,k}^{\xi_a} \neq 0 \right\} \quad (17)$$

$$\tilde{\mathcal{S}} = \left\{ \Delta_{i,j,k} \mid \alpha_{i,j,k}^{\xi_1} \neq 0 \text{ and } \alpha_{i,j,k}^{\xi_2} \neq 0 \right\} \quad (18)$$

comprising all cells containing parts of \mathcal{V}^{ξ_a} or \mathcal{S} respectively. Integrating Eq. (13) over $\Delta_{i,j,k}$ yields

$$\int_{\Delta_{i,j,k}^{\xi_a}} \frac{\partial \mathbf{U}}{\partial t} dV = - \oint_{\partial \Delta_{i,j,k}^{\xi_a} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{c+\mu+q} dA - \oint_{\mathcal{S}_{i,j,k}} \mathbf{F}_S^{c+\mu+q} dA + \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{S} dV \quad (19)$$

for phase ξ_a with $\Delta_{i,j,k}^{\xi_a} = \mathcal{V}^{\xi_a} \cap \Delta_{i,j,k}$. By introducing volume-averaged conservative quantities as

$$\bar{\mathbf{U}} = \frac{1}{\alpha_{i,j,k}^{\xi_a}} \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{U} dV \quad (20)$$

we can rewrite (19) as

$$\frac{\partial (\alpha_{i,j,k}^{\xi_a} \bar{\mathbf{U}})_{i,j,k}}{\partial t} = - \oint_{\partial \Delta_{i,j,k}^{\xi_a} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{c+\mu+q} dA - \oint_{\mathcal{S}_{i,j,k}} \mathbf{F}_S^{c+\mu+q} dA + \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{S} dV. \quad (21)$$

In the above, the first term on the right accounts for fluxes over the cell faces, the second term for interaction between the fluids, and the last term for volume forces acting on the fluids. We introduce the short notation $\mathbf{F}^{c+\mu+q} = \mathbf{F}^c + \mathbf{F}^\mu + \mathbf{F}^q$ and $\mathbf{F}_S^{c+\mu+q} = \mathbf{F}_S^c + \mathbf{F}_S^\mu + \mathbf{F}_S^q$.

Assuming constant, discretized flux densities at cell faces, indicated by the superscript $(\tilde{\cdot})$, the cell face fluxes read

$$\begin{aligned} \tilde{\mathcal{D}}_{\text{FF}} &= - \oint_{\partial \Delta_{i,j,k}^{\xi_a} \setminus \mathcal{S}_{i,j,k}} \mathbf{F}^{c+\mu+q} dA \\ &= \left[\left(A_{i \pm \frac{1}{2}, j, k}^{\xi_a} \tilde{\mathbf{F}}_1^{c+\mu+q} \right)_{i \pm \frac{1}{2}, j, k} - \left(A_{i \pm \frac{1}{2}, j, k}^{\xi_a} \tilde{\mathbf{F}}_1^{c+\mu+q} \right)_{i \pm \frac{1}{2}, j, k} + \right. \\ &\quad \left. \left(A_{i, j \pm \frac{1}{2}, k}^{\xi_a} \tilde{\mathbf{F}}_2^{c+\mu+q} \right)_{i, j \pm \frac{1}{2}, k} - \left(A_{i, j \pm \frac{1}{2}, k}^{\xi_a} \tilde{\mathbf{F}}_2^{c+\mu+q} \right)_{i, j \pm \frac{1}{2}, k} + \right. \\ &\quad \left. \left(A_{i, j, k \pm \frac{1}{2}}^{\xi_a} \tilde{\mathbf{F}}_3^{c+\mu+q} \right)_{i, j, k \pm \frac{1}{2}} - \left(A_{i, j, k \pm \frac{1}{2}}^{\xi_a} \tilde{\mathbf{F}}_3^{c+\mu+q} \right)_{i, j, k \pm \frac{1}{2}} \right] \Delta x^2. \end{aligned} \quad (22)$$

Here, the subscripts 1, 2, 3 denote the Cartesian component of the flux vector. The convective fluxes are obtained by solving a Riemann problem at the respective cell faces and by using low-dissipative high-order shock-capturing reconstruction schemes. Viscous fluxes are obtained by evaluating the viscous stress tensor at cell faces. Therefore, we calculate velocity gradients at cell centers. These gradients together with respective velocities are reconstructed at cell faces. Heat fluxes are computed in the same manner.

At interfaces, the interaction fluxes read

$$\tilde{\mathcal{D}}_{\text{IF}} = - \oint_{\mathcal{S}_{i,j,k}} \mathbf{F}_S^{c+\mu+q} dA = \left(\tilde{\mathbf{F}}_S^{c+\mu+q} \Delta \mathcal{S} \right)_{i,j,k}, \quad (23)$$

where the flux densities are calculated using the interface normal vector $\mathbf{n}_{i,j,k}^{\mathcal{S}}$ and the interface segment

$$\begin{aligned} \Delta \mathcal{S}_{i,j,k} &= \left[\left(A_{i \pm \frac{1}{2}, j, k} - A_{i \pm \frac{1}{2}, j, k} \right)^2 \right. \\ &\quad \left. + \left(A_{i, j \pm \frac{1}{2}, k} - A_{i, j \pm \frac{1}{2}, k} \right)^2 \right. \\ &\quad \left. + \left(A_{i, j, k \pm \frac{1}{2}} - A_{i, j, k \pm \frac{1}{2}} \right)^2 \right]^{\frac{1}{2}} \Delta x^2. \end{aligned} \quad (24)$$

Table 2
Overview of building-blocks options available in the compute framework.

Building-Block	Name	Description
Level-set Advection	Projection-based	Discretization of $\mathcal{D}_{LSA,1}(\phi(\mathbf{x}, t))$ in Eq. (31)
	WENO	Discretization of $\mathcal{D}_{LSA,2}(\phi(\mathbf{x}, t))$ in Eq. (31)
Level-set Reinitialization	Min	According to [67]
	WENO	Iterative procedure of [60] using HJ WENO schemes [68]
Ghost-fluid Extrapolation	Iterative	According to [60]
	Iterative Upwind	According to [58]
Interface Riemann Solver	Linearized	According to [69]. Extension to surface tension effects according to [54]
	Iterative Harten-Lax-van Leer-contact (HLLC)	According to [15]
	Leer-	According to [70]
Riemann Solver	Roe	According to [14]
	Local-Lax Friedrichs	According to [16, sec 2.2.2]
	Global-Lax Friedrichs	According to [16, sec 2.2.2]
	HLLC	According to [71]. Different signal speeds are available, see below.
	HLLC-LM	According to [17]. Different signal speeds are available, see below.
Signal Speeds	HLL	According to [15, ch. 10.3]. Different signal speeds are available, see below.
	Toro	According to [71]
	Einfieldt	According to [72]
Time Integration	Davis	According to [73]
	Arithmetic	According to [74]
	RK-2	According to [75]
Reconstruction Stencils	RK-3	According to [75]
	WENO-AO-53	According to [76]
	WENO-CU-6	According to [77]
	WENO-3	According to [78]
	WENO-5-HM	According to [79]
	WENO-5-Z	According to [80]
	WENO-5	According to [18]
	WENO-7	According to [81]
WENO-9	According to [81]	
Derivative Stencils	TENO-5	According to [19]
	2nd-order central difference	
	4th-order central difference	
Equation of State	HOU5	According to [82]
	Stiffened Gas	According to [83]
	Noble-Able	According to [84]
	Stiffened Gas Complete	According to [85]
	Tait	According to [86]

We evaluate the convective flux densities based on the solution of the two-material Riemann problem satisfying the stress balance in interface normal direction

$$-\mathbf{n}_S \cdot \Pi(\mathbf{U}_S^{\xi_1}) \cdot \mathbf{n}_S + \mathbf{n}_S \cdot \Pi(\mathbf{U}_S^{\xi_2}) \cdot \mathbf{n}_S = -\sigma \mathbf{n}_S (\nabla^T \cdot \mathbf{n}_S) \cdot \mathbf{n}_S. \quad (25)$$

Therein, $\mathbf{U}_S^{\xi_a}$, $i = 1, 2$ are the fluid states at the interface of the respective fluid. Note, the above stress balance takes into account capillary forces, which are evaluated using the surface-tension coefficient σ between the two fluids. By solving this two-material Riemann problem, we obtain the absolute values for interface velocity $(u_S)_{i,j,k}$ and interface pressures $(p_S^{\xi_1})_{i,j,k}$ and $(p_S^{\xi_2})_{i,j,k}$. Different two-material Riemann solvers exist. Table 2 list the

solvers implemented in our framework. Velocity gradients at the interface are calculated using the same differentiation stencil as for the viscous cell-face fluxes. Details on the calculation of viscous interface fluxes are presented in [54].

The volumetric source term on the right of Eq. (19) reads as

$$\tilde{\mathcal{D}}_{VF} = \int_{\Delta_{i,j,k}^{\xi_a}} \mathbf{S} dV = (\alpha^{\xi_a} \tilde{\mathbf{S}})_{i,j,k} \Delta x^3. \quad (26)$$

More details and a thorough validation of the volume forces implemented in ALPACA can be found in [55].

With the previously defined terms of Eqs. (22), (23), and (26), we rewrite Eq. (21) as

$$\frac{\partial (\alpha^{\xi_a} \tilde{\mathbf{U}})_{i,j,k}}{\partial t} = \tilde{\mathcal{D}}_{\text{FF}} + \tilde{\mathcal{D}}_{\text{IF}} + \tilde{\mathcal{D}}_{\text{VF}} = \tilde{\mathcal{D}}_{\text{NSE}}. \quad (27)$$

3.2. Numerical model for the interface

We implicitly capture the interface by a scalar level-set function $\phi(\mathbf{x}, t)$ fulfilling the signed-distance property $|\nabla\phi| = 1$ [51,56]. The location of the interface is always given by the zero level-set contour

$$\mathcal{S}(\mathbf{x}, t) = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}. \quad (28)$$

Two-phases are distinct from each other by the sign of the level-set, i.e. $\phi > 0$ in \mathcal{V}^{ξ_1} and $\phi < 0$ in \mathcal{V}^{ξ_2} .

The discretization of the balance equations presented in Section 3.1 requires the geometric information of the interface, including the interface normal, cell-face apertures and volume fractions. We determine the interface normal $\mathbf{n}_{\mathcal{S}}$ by the normalized gradient of the level set

$$\mathbf{n}_{\mathcal{S}}(\mathbf{x}) = \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|}. \quad (29)$$

Assuming a linear approximation of the level set inside $\Delta_{i,j,k}$, cell-face apertures $A_{i,j,k}^{\xi_a}$ are calculated analytically. Given the cell-face apertures, we calculate volume fractions $\alpha_{i,j,k}^{\xi_a}$ following [57].

In addition to the NSE, we solve the partial differential equation (PDE)

$$\frac{\partial\phi(\mathbf{x}, t)}{\partial t} = \mathcal{D}_{\text{LSA}}(\phi(\mathbf{x}, t)) \quad (30)$$

to evolve the level set in time. Two mathematically equivalent formulations of the right-hand side can be found in literature [51,58]:

$$\begin{aligned} \mathcal{D}_{\text{LSA},1}(\phi(\mathbf{x}, t)) &= -u_{\mathcal{S}}(\mathbf{x}, t) \mathbf{n}_{\mathcal{S}} \cdot \nabla\phi(\mathbf{x}, t) \\ &= -u_{\mathcal{S}}(\mathbf{x}, t) |\nabla\phi(\mathbf{x}, t)| = \mathcal{D}_{\text{LSA},2}(\phi(\mathbf{x}, t)). \end{aligned} \quad (31)$$

The advection equation is strictly only valid where $\phi(\mathbf{x}, t) = 0.0$. However, solving the equation requires evaluation in a small region around the interface.

Solving Eq. (30) for $\phi \neq 0$ might lead to an irregular level-set field violating the signed-distance property. We recover the signed-distance property by solving the reinitialization equation

$$\frac{\partial\phi}{\partial\tau} = \mathcal{D}_{\text{Reinit}} \quad \text{with} \quad (32)$$

$$\mathcal{D}_{\text{Reinit}} = \text{sgn}(\phi_0) (1 - |\nabla\phi|) \quad (33)$$

iteratively in pseudo-time τ to steady state, according to [59]. We denote the operator to solve an equation to steady state as $\mathcal{I}SSS$. Implementation details can be found in Algorithm 4. In Eq. (33), $\text{sgn}(\cdot)$ denotes the sign function and ϕ_0 denotes the level set prior to reinitialization. Temporal evolution of Eq. (32) in pseudo time is done using the single step Euler scheme

$$\phi^{m+1} = \mathcal{E}_{\text{Euler}}(\phi^m, \tilde{\mathcal{D}}_{\text{Reinit}}(\phi, \phi^n), \Delta\tau). \quad (34)$$

The subscript $(\cdot)^m$ indicates the iteration step in pseudo time. The pseudo time step $\Delta\tau$ is kept constant. The temporal evolution operator \mathcal{E} denotes an explicit time-integration scheme. The subscript indicates the specific scheme. Section 3.3 below gives more information about time integration.

Stencils used to calculate $\bar{\mathbf{F}}_{\alpha,i,j,k}$, $\bar{\mathbf{X}}_{i,j,k}^{\xi_a}$ and $\mathbf{u}_{\gamma,i,j,k}$ might cross the interface. However, these stencils work on each fluid separately and thus require information of the considered fluid on both sides of the interface. We call the cells on the opposing side of the interface *ghost cells*. Each fluid has its own ghost cells, which overlap

with the real cells of the other fluid. We fill ghost cells by extrapolating real-fluid conservatives \mathbf{U} normal to the interface. Therefore, we solve the one-way extrapolation equation

$$\frac{\partial\mathbf{U}(\mathbf{x})}{\partial\tau} = \mathcal{D}_{\text{Ext},1W} \quad \text{with} \quad (35)$$

$$\mathcal{D}_{\text{Ext},1W} = \mathbf{n}_{\mathcal{S}}^{\xi_i}(\mathbf{x}) \cdot \nabla\mathbf{U}(\mathbf{x}) \quad (36)$$

iteratively in pseudo-time τ to steady-state. Details about the discretization $\tilde{\mathcal{D}}_{\text{Ext},1W}$ of the right-hand side can be found in [60]. We use a single step Euler scheme

$$\mathbf{U}^{m+1} = \mathcal{E}_{\text{Euler}}(\mathbf{U}^m, \tilde{\mathcal{D}}_{\text{Ext},1W}(\mathbf{U}), \Delta\tau) \quad (37)$$

to evolve Eq. (35) with constant $\Delta\tau$.

A two-way extrapolation is necessary to extrapolate quantities known only at the interface \mathcal{Q} to the narrow band. The respective equation reads

$$\frac{\partial\mathbf{Q}(\mathbf{x})}{\partial\tau} = \mathcal{D}_{\text{Ext},2W} \quad \text{with} \quad (38)$$

$$\mathcal{D}_{\text{Ext},2W} = \text{sgn}(\phi) \mathbf{n}_{\mathcal{S}}^{\xi_i}(\mathbf{x}) \cdot \nabla\mathbf{Q}(\mathbf{x}), \quad (39)$$

and is solved iteratively in pseudo time τ to steady state. Again, we use a single step Euler scheme

$$\mathbf{Q}^{m+1} = \mathcal{E}_{\text{Euler}}(\mathbf{Q}^m, \tilde{\mathcal{D}}_{\text{Ext},2W}(\mathbf{Q}), \Delta\tau) \quad (40)$$

to evolve Eq. (38) with constant $\Delta\tau$. Details about the discretization of $\tilde{\mathcal{D}}_{\text{Ext},2W}$ can be found in [60].

3.3. Time integration

We advance initial-boundary value problems of the form $\frac{\partial\mathbf{u}}{\partial t} = \tilde{\mathcal{D}}(\mathbf{u})$ in time using explicit Runge-Kutta (RK) integration [61,62]. $\tilde{\mathcal{D}}(\mathbf{u})$ denotes the discretization of the right-hand side. To avoid spurious oscillations, we only apply strong stability preserving [63] integration schemes. We denote the operator, which evolves the equation by one time step according to a RK scheme, by \mathcal{E} . For the NSE, this gives the solution at the next time step as

$$\mathbf{U}^{n+1} = \mathcal{E}_{\text{RK}}(\mathbf{U}^n, \tilde{\mathcal{D}}(\mathbf{U}), \Delta t). \quad (41)$$

Analogously, the level set is advected in time using the same evolution operator

$$\phi^{n+1} = \mathcal{E}_{\text{RK}}(\phi^n, \tilde{\mathcal{D}}(\phi), \Delta t). \quad (42)$$

For stability, we have to restrict the time-step size by

$$\Delta t_{\text{NSE}} = \text{CFL} \cdot \min(\Delta t_c, \Delta t_\mu, \Delta t_q, \Delta t_s, \Delta t_S), \quad (43)$$

where

$$\begin{aligned} \Delta t_c &= \max_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_1} \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_2}} \left(\frac{\sum \|v\| + c\|}{\Delta x} \right)^{-1}, \\ \Delta t_\mu &= \max_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_1} \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_2}} \left(\frac{14\mu}{3\rho\Delta x^2} \right)^{-1}, \\ \Delta t_s &= \max_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_1} \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_2}} \left(\frac{\sum \|v\| + c\| + \sqrt{\sum \|u\| + c\| + 4\|g\|\Delta x}}{2\Delta x} \right)^{-1}, \\ \Delta t_q &= \max_{\Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_1} \cup \Delta_{i,j,k} \in \tilde{\mathcal{V}}^{\xi_2}} \left(\frac{14\kappa}{3\rho c_p \Delta x^2} \right)^{-1}, \text{ and} \\ \Delta t_S &= \max_{\Delta_{i,j,k} \in \tilde{\mathcal{S}}} \left(\sqrt{\frac{8\pi\sigma}{\Delta x^3 \sum_i \rho^{\xi_a}}} \right)^{-1} \end{aligned} \quad (44)$$

denote the stability criteria for convective fluxes [47], viscous fluxes [59,64], heat fluxes, volume forces [59,64] and interface terms [65], respectively. Here, the speed of sound c is given by the EOS and c_p denotes the specific heat capacity at constant pressure.

For full cells with $\alpha_{i,j,k}^{sa} = 1$, stability is ensured by setting the Courant-Friedrichs-Lewy (CFL) constant smaller 1. To avoid spurious oscillations in small cells with $\alpha_{i,j,k}^{sa} \ll 1$, a conservative mixing procedure following [57,58,66] is employed, cf. Appendix A.

4. Modularity table

For each of the building blocks of the numerical model presented above, a variety of alternative discretization schemes exist. We list the alternatives available in our framework in Table 2. The table consists of three parts. The first part lists the larger building blocks that form the numerical model presented above. The second part gives stencils used within the building blocks of part one. The third part states available EOSes to model the fluids. The building blocks are independent of each other and give maximum flexibility to the user.

5. Block-based multiresolution

We employ MR compression to decrease the numerical load and thus increase the feasible resolution of our simulations. The MR representation of a (solution) function $u(x, t)$ is expressed as

$$u(x, t) = \sum_k c_k \theta_k(x, t) + \sum_m \sum_k d_k^{lm} \psi_k^{lm}(x, t) \quad (45)$$

on overlaying dyadic grids with levels $l_m, m \in [0, \dots, \infty]$. Therein, c_k and θ_k are scaling coefficients and scaling functions, respectively. Their product sum defines the general shape of the function. Respectively, d_k^{lm} and ψ_k^{lm} denote the *details* and *wavelet* functions. They build up the second term on the right which gives an indicator of the smoothness on the m -th refinement level l_m . The subscript k samples the respective functions at grid points $x_k^m = 2^{-l_m} k \in \mathbb{Z}$. We achieve compression by restricting m .

Data on level l_m is brought to a finer level l_{m+1} via the prediction operator $\mathcal{P}(u_{l_m}) \rightarrow u_{l_{m+1}}$. Vice versa, the conservative averaging operator $\mathcal{A}(u_{l_{m+1}}) \rightarrow u_{l_m}$ brings information from a finer to a coarser level. Both operators are consistent, i.e. $\mathcal{A}(\mathcal{P}(u_{l_m})) = u_{l_m}$. By using interpolation wavelets [36,41], the detail function

$$d_k^{lm}(x) = \sum_k d_k^{lm} \psi_k^{lm}(x) = u_{l_{m+1}}(x) - u_{l_m}(x) \quad (46)$$

gives a direct measure for the smoothness of the solution $u(x, t)$. Small details indicate smooth regions. In these regions, we compress the solution by setting details below a level-dependent threshold to zero. This MR thresholding naturally defines the respective grid resolution in the flow. Vanishing and non-zero details indicate regions to be coarsened, or refined, respectively. In addition, memory compression is achieved by not storing zero-details [37,38]. This results in binary-, quad- or octree data-structures in one, two or three dimensions, respectively. We refer to a node in this tree as *parent* if it has *children* and as *leaf* otherwise. Harten applied this approach successfully to one-dimensional (compressible) fluid-flow simulations [36]. It was extended to viscous flows [45] and multiple dimensions [44].

Additionally, further compression is achieved by the use of (adaptive) LTS schemes [46,47]. This reduction is achieved by integrating coarser levels with larger time-step sizes than finer levels. We call a time step on the finest level a *micro* time step and on the coarsest a *macro* time step, respectively. The number of micro

time steps in a macro time step is $r = 2^{l_{max}-1}$. For the adaptive LTS, the time-step size (43) is adjusted after every micro time step.

We use the previously introduced fully-adaptive MR and adaptive LTS scheme to achieve spatial and temporal adaptivity, respectively. However, we do not coarsen or refine single cells. Instead, we define blocks which are treated uniformly. Blocks on all levels consist of a predefined number of cells, e.g. 32^3 . Although the compression might be lower than for cell-based MR approaches, this is of little relevance for three-dimensional cases [55]. Yet, the block-based approach allows to leverage the multiple levels of parallelism in modern HPC systems [50,55]. For distributed-memory parallelization a block is a domain-decomposition element and the smallest parallel unit. We use halo cells to handle data dependencies between adjacent blocks [47,50]. Halo cells are filled with data from neighboring blocks using the halo-update operator \mathcal{H} . Implementation details are presented in Algorithm 2. At domain borders, halo cells are used to impose boundary conditions. In addition, the halo concept ensures the graded-tree property [37] even for arbitrary resolution jumps between neighboring blocks.

6. Interface tagging system

We have developed a tagging system for the block-based MR algorithm that is inspired by the narrow-band approach. The aim is to limit the computational and memory overhead of the interface model to the narrow-band region only. Regions away from the interface are not affected and still benefit from the clean MR mechanism. This is achieved by introducing tags that link the narrow-band information into the MR topology. We augment the block-based single-phase MR scheme by introducing two node types. First, we have single-phase MR nodes that contain one fluid block as described in [55]. Second, we have multi-phase nodes that contain two fluid blocks. Multi-phase nodes which are leaves additionally hold a level-set block. We also store our interface tags (ITs) in blocks. All node types hold such an IT block. The different block types within a node have the same amount of cells and the same geometric location and orientation. The introduced node types allow to define the following rules, which ensure the objectives stated above.

1. A tree node may hold one or two blocks. Each block contains the data of one and only one phase.
2. A multi-phase node which is a leaf holds an additional level-set block, which contains the level-set and interface data. Interface data are e.g. interface velocity and pressures.
3. Parents of multi-phase nodes are multi-phase nodes themselves.
4. Only single-phase nodes participate in the MR based coarsening and refinement.
5. Single-phase leaves are refined if the interface approaches unless they reside on the maximum level. Thus, they become a parent node.
6. Maximum-level single-phase leaves allocate a second fluid and a level-set block if the interface approaches.
7. Vice versa, maximum level multi-phase leaves deallocate the second fluid and level-set block if the interface retreats.
8. Single-phase parent nodes allocate a second fluid block if the interface approaches.
9. Conversely, multi-phase parent nodes deallocate the second fluid if the interface retreats.

Complying with the first four rules is trivial. The challenging part lies in the remaining rules, which require notification of approaching or retreating interfaces.

For approaching interfaces we have to ensure that nodes are successively refined, such that l_{max} is reached before the inter-

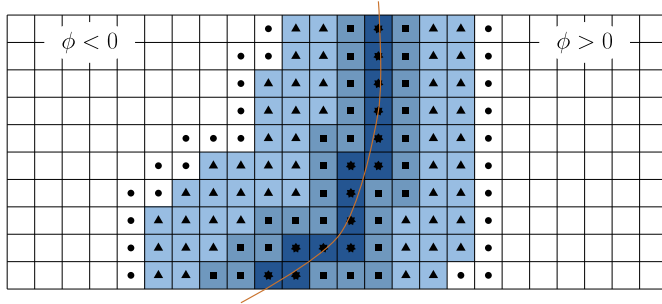


Fig. 2. Interface tag structure for a two-dimensional grid. The interface line is shown in orange. Cells containing a piece of the interface are consequently marked as cut cells (★). The band structure of $\widetilde{\text{CNB}}$ (■), $\widetilde{\text{EB}}$ (▲) and $\widetilde{\text{RB}}$ (●) is clearly visible. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

face reaches the respective node. We achieve this by the previously mentioned ITs. The ITs are a cell-wise quantity. They are denoted as M_{l_m} and represent the interface and its surrounding. We use four different interface tags. First, the tag value 0 (★) indicates a cut cell. Second, the tag value ± 3 (■) indicates the *cut-cell neighbor band* ($\widetilde{\text{CNB}}$), defined as

$$\widetilde{\text{CNB}} = \{ \Delta_{i,j,k} | \exists \Delta_{i+l,j+m,k+n} \in \widetilde{\mathcal{S}}, \text{ with } l, m, n \in \{-1, \dots, +1\} \}. \quad (47)$$

Third, the tag value ± 7 (▲) indicates the *extension band* ($\widetilde{\text{EB}}$), defined as

$$\widetilde{\text{EB}} = \{ \Delta_{i,j,k} | \exists \Delta_{i+l,j+m,k+n} \in \widetilde{\mathcal{S}}, \text{ with } l, m, n \in \{-3, \dots, +3\} \}. \quad (48)$$

In this region, the extrapolation equations cf. Section 3.2 are solved on the finest level. Fourth, the tag value ± 8 (●) indicates the *reinitialization band* ($\widetilde{\text{RB}}$), defined as

$$\widetilde{\text{RB}} = \{ \Delta_{i,j,k} | \exists \Delta_{i+l,j+m,k+n} \in \widetilde{\mathcal{S}}, \text{ with } l, m, n \in \{-4, \dots, +4\} \}. \quad (49)$$

In this region, the level-set reinitialization equation cf. Section 3.2 is solved. The sign of the IT corresponds to the level-set function and indicates the fluid. The used IT values allow efficient bit shift operations. Other values could be chosen. Nevertheless, the value 0 is particularly helpful for the IT averaging described below. As an example, we show the interface tag field for a simple two-dimensional domain in Fig. 2.

With an interface field as shown in Fig. 2, we comply with the remaining rules and minimize the computational load. On the finest level, the interface tags define the region for the narrow-band approach. On the other levels, the level set can be neglected completely. The interface tags suffice to represent the interface and its narrow band in a coarse manner. This saves memory as interface tags are one byte integers compared to the eight byte level-set double precision values. We deduce ITs on coarser levels from the ITs on the finest level using an averaging operator $\mathcal{A}(M_{l_m}) \rightarrow M_{l_{m-1}}$ analogous to the MR averaging.

In the interface tag averaging, we multiply the interface tag values of the fine cells. Thus, the position of cut cells is clearly identified by zero values. The other coarse interface tag values are in an inconsistent state afterwards. We overwrite them arithmetically based on the distance to the cut cells, cf. Eqs. (47) to (49). We call this the tagging operator $\mathcal{T}(M_{l_m}) \rightarrow M_{l_m}$. With correct ITs

Table 3

Operators involved in the algorithm. For details refer to the respective sections.

Operator symbol	Operator description	Details
\mathcal{D}	Right-hand side	Section 2
\mathcal{E}	Temporal evolution	Section 3.3
\mathcal{A}	Averaging	Section 5
\mathcal{P}	Prediction	Section 5
\mathcal{T}	Tagging	Section 6
\mathcal{H}	Halo update	Section 5
$\mathcal{I}SSS$	Iterative steady-state solver	Section 3.2

Table 4

Buffers involved in the algorithm.

Symbol	Symbol description
\mathbf{U}	Conservatives
\mathbf{P}	Prime States
M	Interface tags
ϕ	Level set
\mathbf{Q}	Interface quantities

Table 5

Sets of nodes involved in the algorithm.

N_{all} :	All nodes (on all levels)
N_L :	All nodes on levels $l_m \in \{l_{max}, \dots, l_{max} - ((r+1) \cdot r)\}$, with the current micro time step $r \in 0, \dots, 2^{l_{max}}$
N_0 :	All nodes on level zero
L :	All leaves (on all levels)
LL :	All Leaves holding a level-set block

on all levels, adjusting the MR octree to approaching and retreating interfaces is straightforward. Note, this concept allows to adapt the MR representation implicitly as with the details-thresholding for approaching flow discontinuities. Fig. 3 shows exemplarily the MR discretization of a simple one-dimensional domain. For the given time instance, all nodes on the coarsest level are multi-phase nodes. The interface then moves to the position indicated by the dotted line. Based on the new interface location the ITs are set according to the averaging and tagging procedure described above. This shifts all ITs on all levels one cell to the right. Hence, the left most block does not hold an IT value anymore, not even in its halo, and can thus deallocate the second fluid. After that, this node and its child nodes will participate in the MR thresholding.

7. Implementation

The implementation of the presented building blocks in a holistic HPC simulation environment is not trivial. Careful attention has to be paid to the algorithmic details without compromising the computational performance. We summarize the operators used in the final algorithm in Table 3.

As stated earlier, the operators work on different quantities throughout the algorithm. We keep each such quantity in a respective buffer. For each node in the MR octree, these buffers are allocated as blocks. We summarize the used buffers in Table 4. Also, operators are applied to different sets of nodes as we have a space- and time-adaptive algorithm. We summarize these sets in Table 5.

We give the final algorithm to advance the simulation in time in Algorithm 1. Detailed descriptions of several subroutines in this algorithm are shown in Algorithms 2 to 4. Note, that Algorithm 1 requires a correctly initialized MR tree. This includes the level-set field and deduced interface tags.

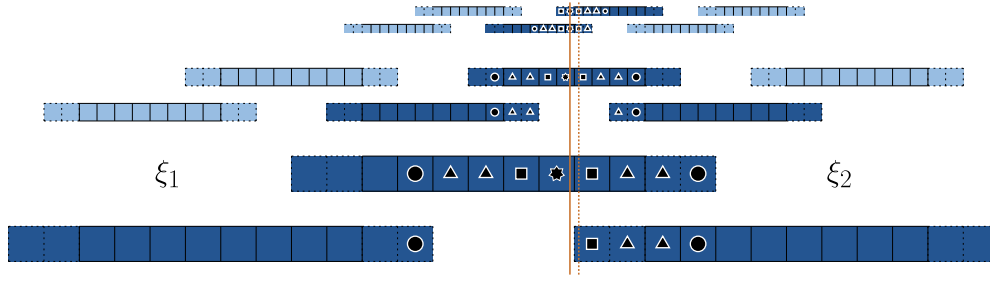


Fig. 3. Interface tag representation for a three-level block-based MR tree in one dimension. For each node the halo cells are represented by dashed lines. To indicate the overlap of halo cells, blocks of the same level are offset on top or below each other. The interface for the current and future time is shown as solid, respectively dashed, line in orange. The interface tag markers denote the current interface position. The dark blue color indicates a multi-phase node, the brighter one shows a single-phase node. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

```

while  $t < t_{end}$  do
  for Micro time steps  $r$  do
     $\Delta t_m \leftarrow$  satisfying Eq. (43), according to [47]
    for RK stages do
      if  $r = 0$  then  $\tilde{D}_{NSE}(\mathbf{U})|_{N_{all}}$ 
      else  $\tilde{D}_{NSE}(\mathbf{U})|_{N_L}$ 
       $\tilde{D}_{LSA}(\phi)|_{LL}$ 
       $\mathcal{A}(\tilde{D}_{NSE}(\mathbf{U}_m)|_{N_{all} \setminus N_0} \rightarrow \tilde{D}_{NSE}(\mathbf{U}_{m-1}))$ 
       $\mathcal{H}(\tilde{D}_{NSE}(\mathbf{U})|_{N_{all}}, true)$  // cf. Algorithm 2
       $\mathcal{H}(\tilde{D}_{LSA}(\phi)|_{LL}, true)$ 
       $\mathcal{E}(\mathbf{U}, \tilde{D}, \Delta t_m)$  // cf. Algorithm 3
       $\mathcal{E}(\phi, \tilde{D}(\phi), \Delta t)$ ;  $\mathcal{A}(\mathbf{U}_m)|_{N_L \setminus \min(N_L)} \rightarrow \mathbf{U}_{m-1}$ 
      if last stage then Resolution jump flux correction [47]
       $\mathcal{H}(\tilde{\mathbf{U}})$  and  $\mathcal{H}(\phi)$ 
      if last stage then
        | Remesh and re-distribute load // cf. Section 7.1
      end
      if  $\alpha_{i,j,k}^{Ea} \ll 1$  then Mix cell // cf. Section 3.3
       $ISSS(\phi, \tilde{D}_{reinit})$  // cf. Section 3.2
      and Algorithm 4
       $\mathcal{T}(M)$  // cf. Section 6
       $ISSS(\mathbf{P}, \tilde{D}_{ext,1W})$  // Extrapolate prime states
      cf. Section 3.2 and Algorithm 4
       $ISSS(\mathbf{Q}, \tilde{D}_{ext,2W})$  // Extrapolate interface
      quantities cf. Section 3.2 and Algorithm 4
       $\mathcal{A}(M_m)|_{N_L \setminus N_0} \rightarrow M_{m-1}$ 
    end
  end
   $t += \Delta t_m$ 
end

```

Algorithm 1: Compute Loop executed in a simulation.

```

Function  $\mathcal{H}$  (Buffer  $b_m$ , Bool include_jumps = false)
  for All sides  $s$  of  $b_m$  do
    if side is a resolution jump and include_jumps then
      | Fill halo cells via prediction from coarser level  $l_m$ 
      |  $b_{l_m,s} = \mathcal{P}(b_{l_m,s})$ 
    end
    else
      | Fill halo cells  $b_{l_m,s}$  from neighbor on same level  $l_m$ 
    end
  end
end

```

Algorithm 2: Halo update \mathcal{H} implementation.

7.1. Distributed implementation

We parallelized the above algorithm using Message Passing Interface (MPI). This is also visible by the term “re-distribute load” in Algorithm 1. The parallelization is also implied in all routines exchanging data between nodes of the MR tree.

Our parallelization builds on the separation of heavy-computation data from lightweight topology data. We tag each node with a unique id. This allows to store the heavy data in a rank-local hash

```

Function  $\mathcal{E}$  (Buffer  $\mathbf{U}$ ,  $\Delta t$ )
  for All sides  $s$  of  $\mathbf{U}$  do
    if  $s$  is a resolution jump then
      // Integrate its halo cells:
       $s^{n+1} = \mathcal{E}_{RK}(s^n, \tilde{D}(\mathbf{U}), \Delta t_m)$  // cf. Section 3.3
      and Eq. (41)
    end
  end
  // Integrate the internal cells:
   $\mathbf{U}^{n+1} = \mathcal{E}_{RK}(\mathbf{U}^n, \tilde{D}(\mathbf{U}), \Delta t_m)$ 
end

```

Algorithm 3: Time integration \mathcal{E} implementation according to [47].

```

Function  $ISSS(b, \tilde{D}(b))$ 
  Iteration counter  $i = 0$ 
   $m =$  maximum number of iterations while  $i < m$  do
    // Check convergence
    if  $\tilde{D}(b) < \epsilon$  and  $i > 0$  then
      | break
    end
    Calculate  $\tilde{D}(b)$  for internal cells
    Update  $b$  using  $\mathcal{E}_{Euler}(b, \tilde{D}(b), \Delta \tau)$ 
     $\mathcal{H}(b)|_{LL}$  // set correct values for the entire
    field
     $i = i + 1$ 
  end
end

```

Algorithm 4: Iterative implementation of $ISSS$ to reach steady-state.

map. The id also identifies communication partners for cross-node communication. We distribute the computational load by assigning nodes to ranks using a level-wise space filling curve (SFC) [55]. Parent nodes are assigned to the same rank as the majority of their children reside on. We balance the load every micro time step if the topology changed.

7.2. Modularity

We paid special attention to usability, extendability and flexibility of our implementation without sacrificing parallel performance. Therefore, we chose an object-oriented (OO) approach written in C++20. We strictly separate data containers and data manipulators. This separation eases variation of the building blocks given in Table 2. Building blocks are selected at compile-time using the curiously recurring template pattern (CRTP). This enables full compile- and link-time optimization. Hence, the overall performance is comparable between schemes [50] and fair comparisons between competing schemes can be drawn. In addition to compile-time modularity, our framework offers various runtime options, e.g. topology settings, boundary conditions or EOSes. We implement those us-

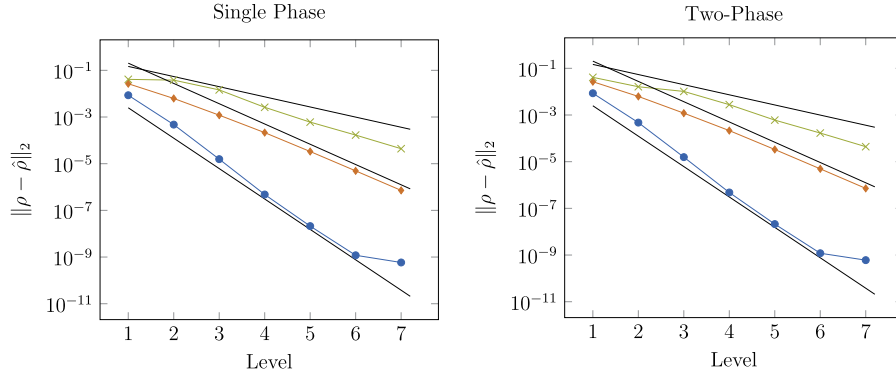
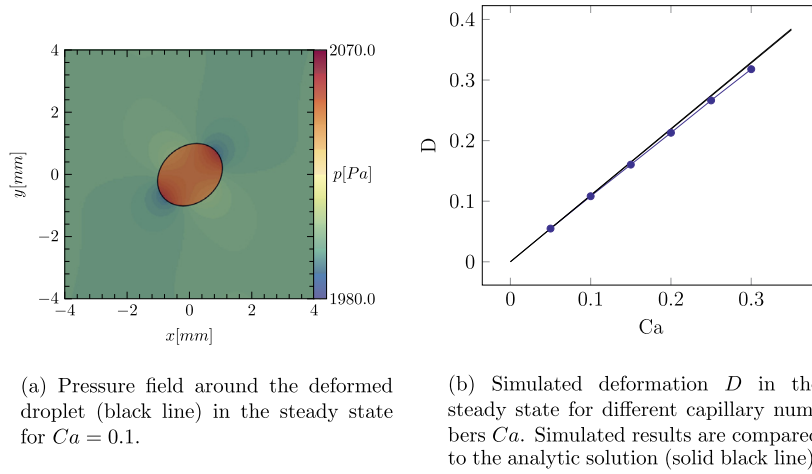


Fig. 4. Error convergence over maximum MR refinement level for the advecting Gaussian density bells. For both, single-phase (left) and two-phase (right), three different combinations of reconstruction stencil and RK time integration were tested: WENO-5 and RK-3 (—●—) WENO-3 and RK-2 (—◆—), as well as WENO-1 and RK-2 (—×—).



(a) Pressure field around the deformed droplet (black line) in the steady state for $Ca = 0.1$.

(b) Simulated deformation D in the steady state for different capillary numbers Ca . Simulated results are compared to the analytic solution (solid black line).

Fig. 5. The simulated domain is shown on the left. The deformation D for different simulated capillary numbers Ca is shown on the right.

ing runtime polymorphism and well-known programming patterns like the factory method and the proxy.

8. Numerical test cases

We show the capabilities of our solver by numerous challenging test cases. We present a convergence study as well as validation for the implemented viscous and capillary terms. Afterwards, physically interesting shock-interface interactions are presented. Unless stated otherwise, we use the following configuration of building blocks cf. Table 2: projection-based WENO level-set advection using the third-order Hamilton-Jacobi WENO (HJ-WENO) stencil, WENO reinitialization using the third-order HJ-WENO stencil, iterative extrapolation, linearized interface Riemann solver, WENO-5 reconstruction stencil, Roe Riemann solver, RK-3 time integration. We compute velocity and temperature gradients for viscous and heat fluxes using fourth-order central differences and reconstruct them to cell faces using a fourth-order central scheme.

8.1. Convergence study

We advect two gaussian density bells

$$\rho = 1 + 5 \exp\left(-200(x - 0.5)^2\right) + 5.0 \exp\left((x - 1.5)^2\right) \quad (50)$$

in a periodic domain of size two with constant velocity $u = 1$ and pressure $p = 1$. The simulation is run until $t_{end} = 0.5$. We ran this case as single-phase simulation and with an additional interface located between the density bells. For both setups, three different configurations of the numerical schemes were tested. The

first configuration is the standard configuration. For the second and third configuration we varied the orders of the reconstruction stencil and of the time integration. Convergence is demonstrated for increasing maximum MR refinement level. The coarsest resolution $l_{max} = 1$ implies a homogenous mesh. The finest resolution is $l_{max} = 7$ and accounts for 4096 effective cells. The obtained l_2 -error norms are given in Fig. 4.

The convergence rates show the expected trends. The scheme of lowest-order dominates the overall convergence order. Only the first order spatial reconstruction exceeds its nominal convergence order slightly. For very small errors the convergence levels off due to the dominance of the perturbation error [38].

8.2. Shear-drop deformation

We used an immersed circular liquid drop in a sheared ambient liquid to validate the interplay of viscous and capillary forces. In this case, viscous forces shearing the drop compete with the surface-minimizing capillary forces. For stable parameters, the drop reaches a terminal ellipsoidal shape. Exemplarily, Fig. 5a shows the converged state of such a drop. The final state is determined by the viscosity ratio μ_b/μ_d and the capillary number

$$Ca = \frac{\mu_b R_0 \dot{s}}{\sigma}. \quad (51)$$

Therein, R_0 denotes the initial radius of the droplet, \dot{s} the shear rate, σ the surface tension coefficient and μ_d and μ_b the viscosities of the drop and bulk fluid, respectively. For small deforma-

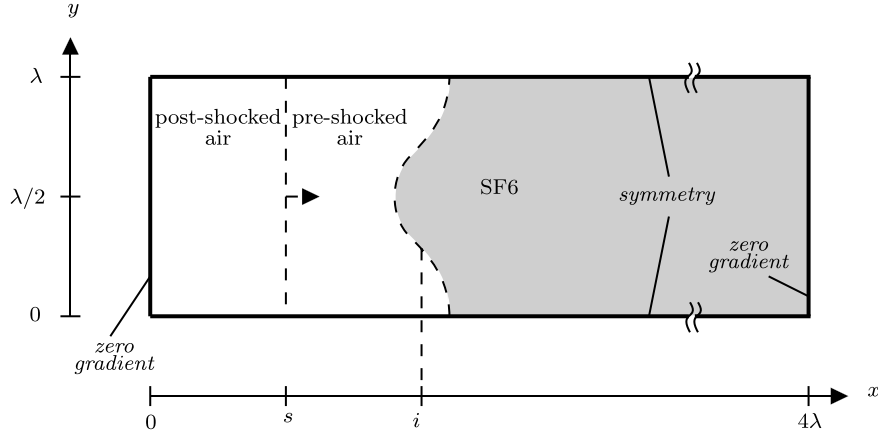


Fig. 6. Schematic of the computational domain of a RMI. The wavelength of the single-mode RMI is $\lambda = 1$. The interface is initially located at $i = 2.9$ and disturbed with an amplitude $a_0 = 0.1$. The shock is located at $s = 0.8$. Symmetry and zero-gradient boundary conditions are applied as indicated in the figure.

Table 6
Initial fluid states and EOS parameters for the shock-interface interaction cases. For the upper three cases, the stiffened-gas EOS is used. The remaining cases use the stiffened-gas complete EOS.

		ρ	p	u	v	γ	Π	R	e_t	E_∞
SF6 RMI	SF6	5.04	1.0	0.0	0.0	1.093	0.0	-	-	-
	pre-shock air	1.0	1.0	0.0	0.0	1.4	0.0	-	-	-
	post-shocked air	1.411	1.628	0.39	0.0	1.4	0.0	-	-	-
Helium SBI 2D	Helium	0.166	101325.0	0.0	0.0	1.66	0.0	-	-	-
	pre-shocked air	1.2041	101325.0	0.0	0.0	1.4	0.0	-	-	-
	post-shocked air	1.6573	159056.0	114.5	0.0	1.4	0.0	-	-	-
Helium SBI 3D	Helium	0.166	101325.0	0.0	0.0	1.66	0.0	-	-	-
	pre-shocked air	1.2041	101325.0	0.0	0.0	1.4	0.0	-	-	-
	post-shocked air	1.7201	167819.5	128.7	0.0	1.4	0.0	-	-	-
R22 SBI 3D	R22	3.117	1.0	0.0	0.0	1.178	0.0	0.3208	0.0	0.0
	pre-shocked air	1.0	1.0	0.0	0.0	1.4	0.0	1.0	0.0	0.0
	post-shocked air	1.376	1.5698	0.3947	0.0	1.4	0.0	1.0	0.0	0.0
Three bubble	Air bubble	1.0204	101325.0	0.0	0.0	1.178	0.0	287.058	0.0	0.0
	pre-shocked water	993.89	101325.0	0.0	0.0	4.4	6.0e8	1700.0	-1.8e6	7.456e6
	post-shocked water	1222.9	1.0e9	434.02	0.0	4.4	6.0e8	1700.0	-1.8e6	7.456e6

tions, the following relation between the drop the deformation D and Ca holds [87]:

$$D = \frac{B_1 - B_2}{B_1 + B_2} = Ca \frac{19\mu_b/\mu_d + 16}{16\mu_b/\mu_d + 16}, \quad (52)$$

where B_1 and B_2 denote the major- and minor-axis of the ellipsoidal drop. We use a square-shaped domain of length $L = 8$. Shearing is imposed by constant velocities $U = \pm 4m/s$ at the top and bottom boundaries, respectively. This yields $\dot{\gamma} = 1$. Periodicity is imposed at the remaining boundaries. Initial conditions and material parameters using Tait's EOS for both fluids are

$$p = 2 \cdot 10^3, \quad \rho = 1, \quad \rho_0 = 1, \quad A = 2 \cdot 10^3, \quad B = 2 \cdot 10^4, \quad \gamma = 7.15, \quad \sigma = 50, \quad \mu = 5. \quad (53)$$

The initial condition for the level set is

$$\phi = R_0 - \sqrt{(x - 4)^2 + (y - 4)^2}. \quad (54)$$

We set $R_0 = 1$ and simulate $Ca = 0.05, 0.1, \dots, 0.30$. The capillary number is adjusted by varying σ . The results agree well with the analytic solution as presented in Fig. 5b.

8.3. Richtmyer-Meshkov instability

We conduct a two-dimensional Euler simulation of the Richtmyer-Meshkov Instability (RMI) [88]. Our computational setup fol-

lows [89,90]. We sketch the computational domain in Fig. 6 and give the initial conditions in Table 6.

Fig. 7 shows the evolution of the RMI over time. Fig. 7a gives the initial state. Then, the incident shock wave moves from left to right. When it hits the interface, a reflected shock wave moves to the left and a transmitted shock moves to the right, cf. Fig. 7b. Baroclinic vorticity generation triggers the interface disturbance, cf. Fig. 7c. Over time, the instabilities at the interface grow further and the characteristic mushroom like shape develops, cf. Fig. 7d. The overall flow features are in good agreement with [89].

We show the temporal evolution of characteristic interface points in Fig. 8. The mixing layer width shows a kink at approximately $t = 0.15$, where the initial shock passes the right-most point of the interface. Up to this point the mixing layer decreases. Thereafter, it starts to grow. We compare our results for three different resolutions $\lambda/\Delta x = 128, 256, 512$, where λ denotes the wavelength of the single-mode RMI, with [89] and find good agreement.

8.4. Air helium shock-bubble interaction

We conduct two-dimensional Euler simulations of a helium column interacting with a shock wave in air. This test case is frequently used to assess the performance of numerical methods [86,89,91–93]. Our setup follows the experiments of [94] and is sketched in Fig. 9. We give the problem dimensions, initial condi-

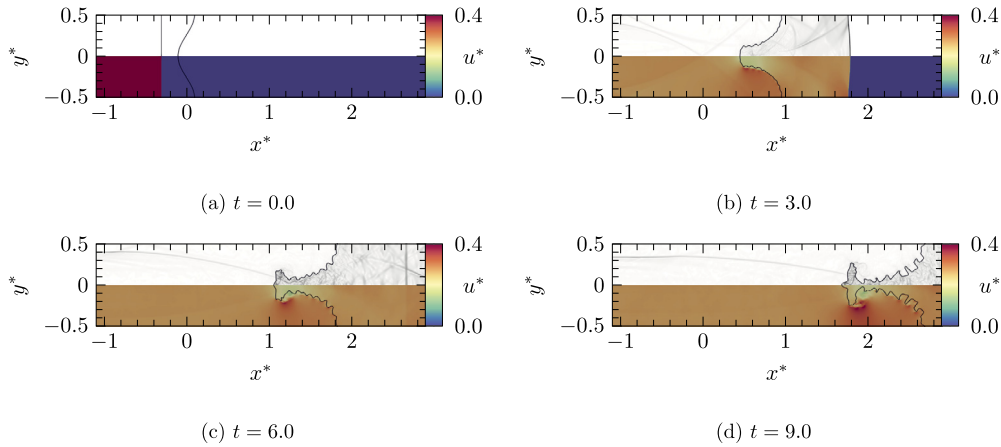


Fig. 7. The flow field in the simulated domain for different time instances. The upper half of each image shows numerical schlieren images, and the lower half the velocity in x-direction. The interface is shown as a black line.

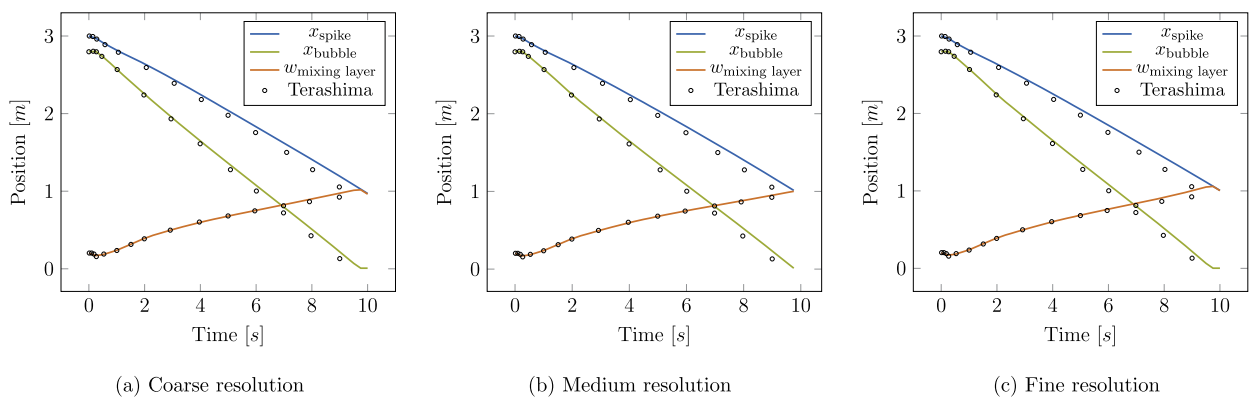


Fig. 8. Evolution of characteristic interface points. The position of the spike x_{spike} (left-most point of the interface), the position of the bubble d_{bubble} (right-most point of the interface), and the mixing layer width $w_{\text{mixing layer}} = x_{\text{spike}} - x_{\text{bubble}}$ are shown. From left to right, we show the coarse, medium and fine results, respectively. Results of Terashima, Tryggvason [89] are included for comparison. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

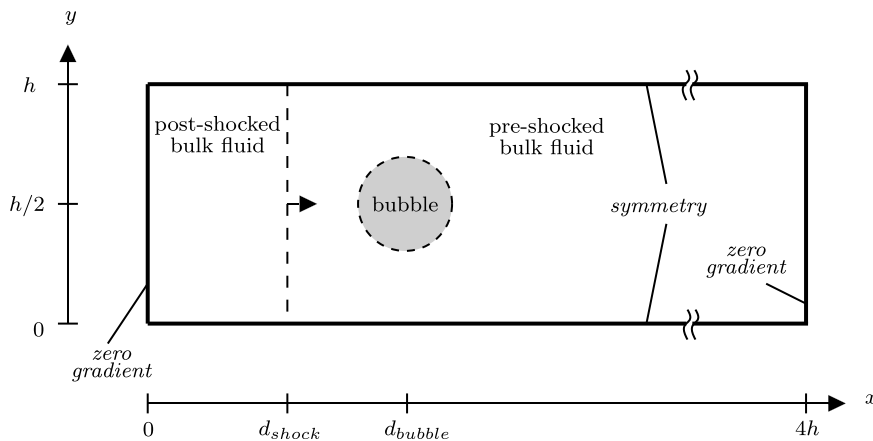


Fig. 9. Schematic two-dimensional projection of the computational domain for the shock-bubble interactions.

tions and material parameters in Tables 6 and 7, respectively. The Mach number of the initial shock wave is $Ma_s = 1.22$.

Fig. 10 shows the flow field for two different time instances. When the initial shock wave hits the helium bubble, a reflected shock moves to the left, and a transmitted shock inside the bubble is generated. Due to the higher speed of sound, the transmitted shock inside the bubble moves faster than the initial shock in air, cf. the flow field in Fig. 10a. Initiated by the shock passage, the bubble deforms strongly by the induced jet, cf. Fig. 10b. We compare the flow features in our results to 2D numerical simula-

tions [89] and experiments [94] with a helium column. Also, we compare quantitatively the temporal evolution of characteristic interface points to results of [92] in Fig. 11. Both, qualitative and quantitative assessments show good agreement.

We also conduct three-dimensional simulations of a spherical setup as shown in experiments with a helium bubble by [94]. A helium sphere interacts with a $Ma_s = 1.25$ shock wave in air. In Table 6 we give initial conditions and material parameters. Problem dimensions and a sketch of the computational domain are given in Table 7 and Fig. 9, respectively. Note, Fig. 9 only shows the two

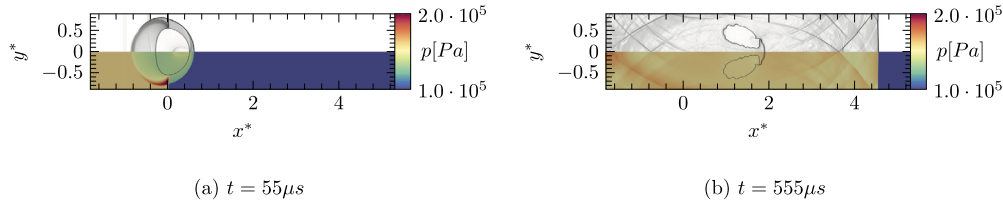


Fig. 10. The flow field in the simulated domain for different time instances. The upper half of each image shows numerical schlieren images, and the lower half the pressure field. The interface is shown as a black line.

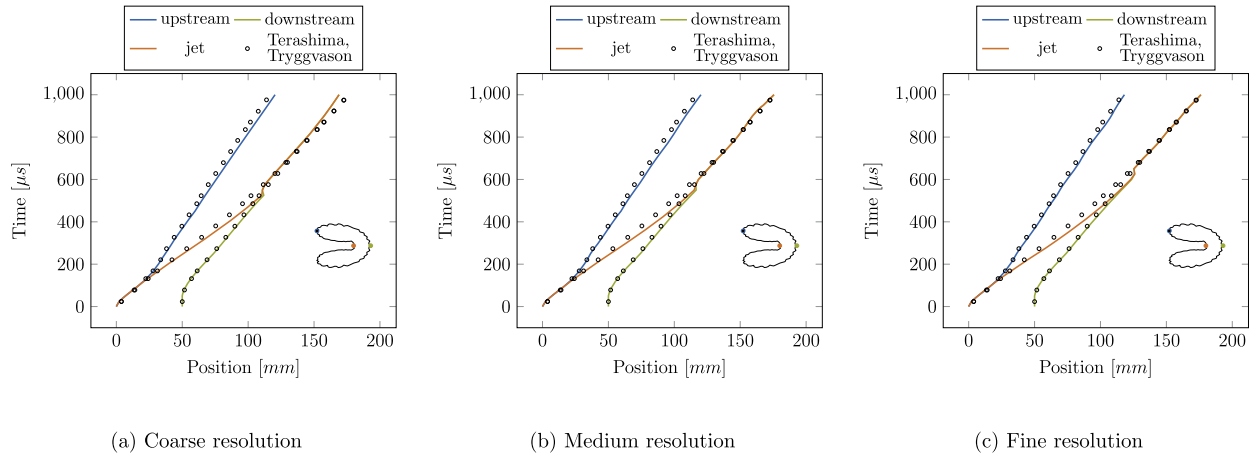


Fig. 11. Evolution of characteristic points of the interface. The position of the upstream point (left-most point of the interface), the position of the downstream point (right-most point of the interface), and the position of the jet (left-most point of the interface in the symmetry plane) are shown. Exemplarily, the characteristic points are sketched in the lower right of the picture in blue, green and orange, respectively. From left to right, we show coarse (512×128 effective cells), medium (1024×256 effective cells) and fine (2048×512 effective cells) results, respectively. Reference data is taken from Terashima, Tryggvason [89]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 7
Dimensions of the shock-bubble interaction cases.

	$l_{channel}$	$h_{channel}$	d_{shock}	d_{bubble}	$D_{0,bubble}$
Helium 2D	356mm	89mm	60mm	90mm	50mm
Helium 3D	356mm	89mm	60mm	90mm	45mm
R22	0.4	0.2	0.1	0.15	0.025

spatial dimensions x and y . In the z -direction, we impose the same setup regarding size, boundary conditions, and initial conditions as in the y -direction. The effective resolution is $4096 \times 1024 \times 1024$ cells. Fig. 12 shows a comparison between the experimental images of [94] and our numerical results. We find excellent agreement for the drop deformation and flow evolution.

8.5. Air R22 shock-bubble interaction

In contrast to the previous examples, here we study the shock interaction with a denser bubble than the surrounding fluid. For this purpose, the Air/R22 shock-bubble interaction is simulated based on the setup of [95]. We conduct two- and three-dimensional simulations of this case.

In Table 6 we give initial conditions and material parameters. Problem dimensions and a sketch of the computational domain are given in Table 7 and Fig. 9, respectively. Note, Fig. 9 only shows the two spatial dimensions x and y . In the three-dimensional case, we impose for the z -direction the same setup regarding size, boundary conditions, and initial conditions as in the y -direction. For this setup, we neglect body and capillary forces. The viscosities for air and R22 depend on the temperature. We employ the Sutherland law [96] to express this relation. Following [95], we use $(T_0, \mu_0, S, Pr)_{air} = (275, 1.725e^{-5}, 110.9, 0.7)$ and

$(T_0, \mu_0, S, Pr)_{R22} = (273, 1.15e^{-5}, 330.0, 0.65)$, respectively, with the dimensionless Prandtl number $Pr = c_p \mu / \kappa$.

For the two-dimensional setup, we show the flow field at different time instances in Fig. 13. Due to the shock-focusing in the bubble, cf. Fig. 13a, a small cusp develops on the downstream centerline of the bubble, cf. Figs. 13b and 13c. At the same time baroclinic vorticity generation triggers small interface disturbances that eventually grow and shear off from the equatorial plane, cf. Fig. 13d. The quantitative comparison of the most upstream interface point with data from Kannan et al. [95] in Fig. 14 shows excellent agreement.

Three-dimensional results with a resolution of $2048 \times 1024 \times 1024$ effective cells are shown in Fig. 15. The flow features are similar to the two-dimensional setup presented above. However, stronger shock-focusing leads to a much prolonged filament at the downstream side in three dimensions, cf. Figs. 15b to 15d.

8.6. Water air shock-bubble interaction

Finally, we consider a shock-interface interaction problem with a high density ratio. Here, we focus on a shock-bubble configuration, where the ambient bulk fluid is denser. The opposite configuration has already been studied using ALPACA, e.g. [97]. We consider a complex interaction with three staggered bubbles, cf. Fig. 16. The initial conditions are provided in Table 6. This test case is simulated in 2D and 3D as done in [98]. Note, we choose periodic boundary conditions in the third direction analogous to the y -direction.

The flow field for different time instances is shown in Fig. 17. After shock passage, the lower bubbles are impinged and skew towards the center bubble, cf. Fig. 17a. Later, the remaining parts of the large bubbles disintegrate, cf. Fig. 17b. The small bubble then also collapses due to interaction with the initial and reflected

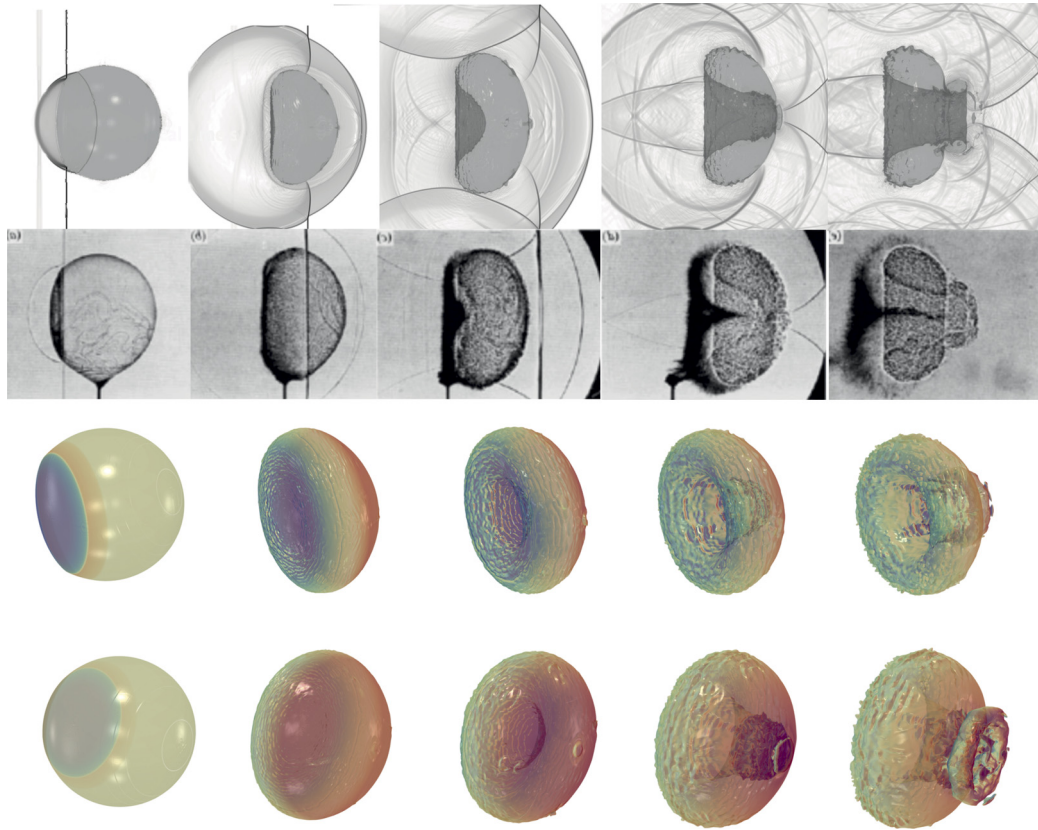


Fig. 12. Comparison of computational and experimental results. We show numerical schlieren and the bubble contour for our simulation in the first row. The second row shows experimental images for comparison. Images from Haas, J. F., & Sturtevant, B. (1987), Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, *Journal of Fluid Mechanics*, 181, 41-76, reproduced with permission. A back and front view of the bubble contour colored by the interface velocity is shown in the bottom two rows. Blue indicates a high interface velocity opposite to \mathbf{n}_S . Red indicates a high interface velocity in direction of \mathbf{n}_S . The shown snapshots are taken at $t = 20, 82, 145, 223, 350 \mu\text{s}$ after the initial shock hits the bubble. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

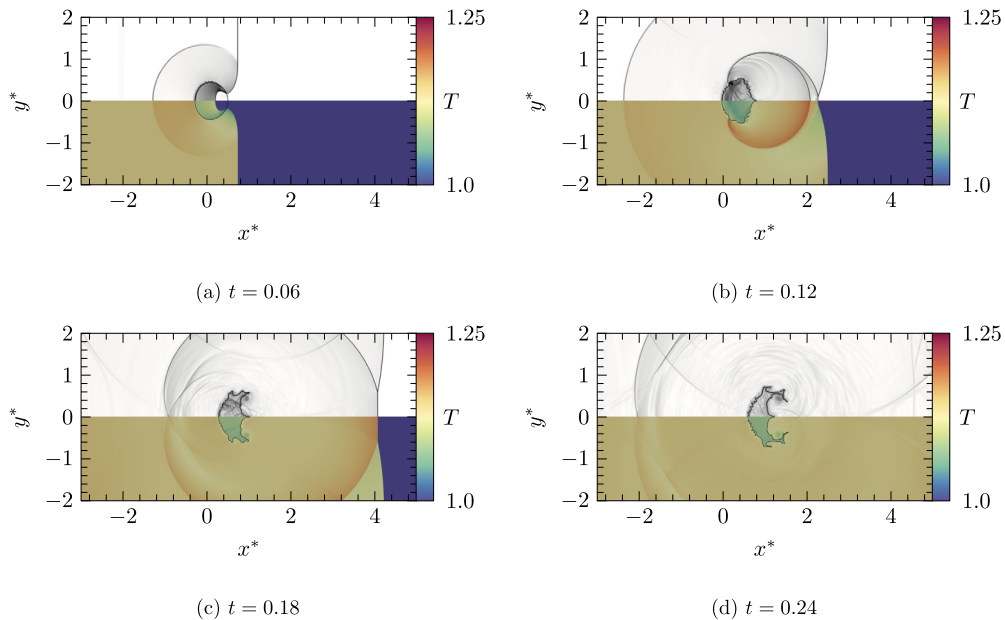


Fig. 13. The flow field in the simulated domain for different time instances. The upper half of each image shows numerical schlieren. The lower half depicts the temperature. The interface is shown as a black line.

shock waves, cf. Figs. 17c and 17d. We compare the evolution of the maximum pressure inside the domain with [98] in Fig. 18 and find excellent agreement.

Three-dimensional results with an effective resolution of $1792 \times 1024 \times 1024$ cells are shown in Fig. 19. The flow features are similar to the two-dimensional setup.

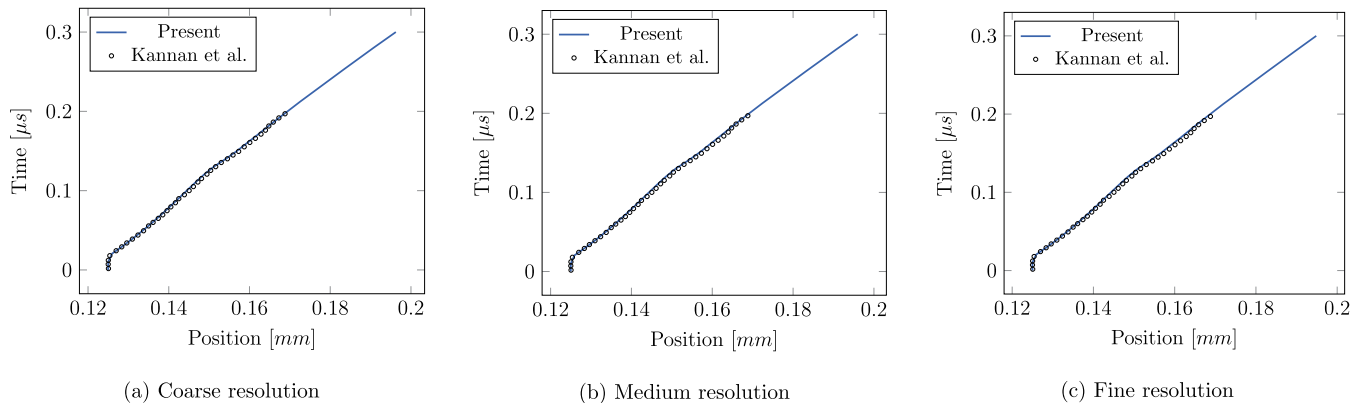


Fig. 14. Evolution of the upstream point (left-most point of the interface). From left to right, we show coarse (1024×512 effective cells), medium (2048×1024 effective cells) and fine (4096×2048 effective cells) results, respectively. For comparison, results of Kannan et al. [95] are shown.

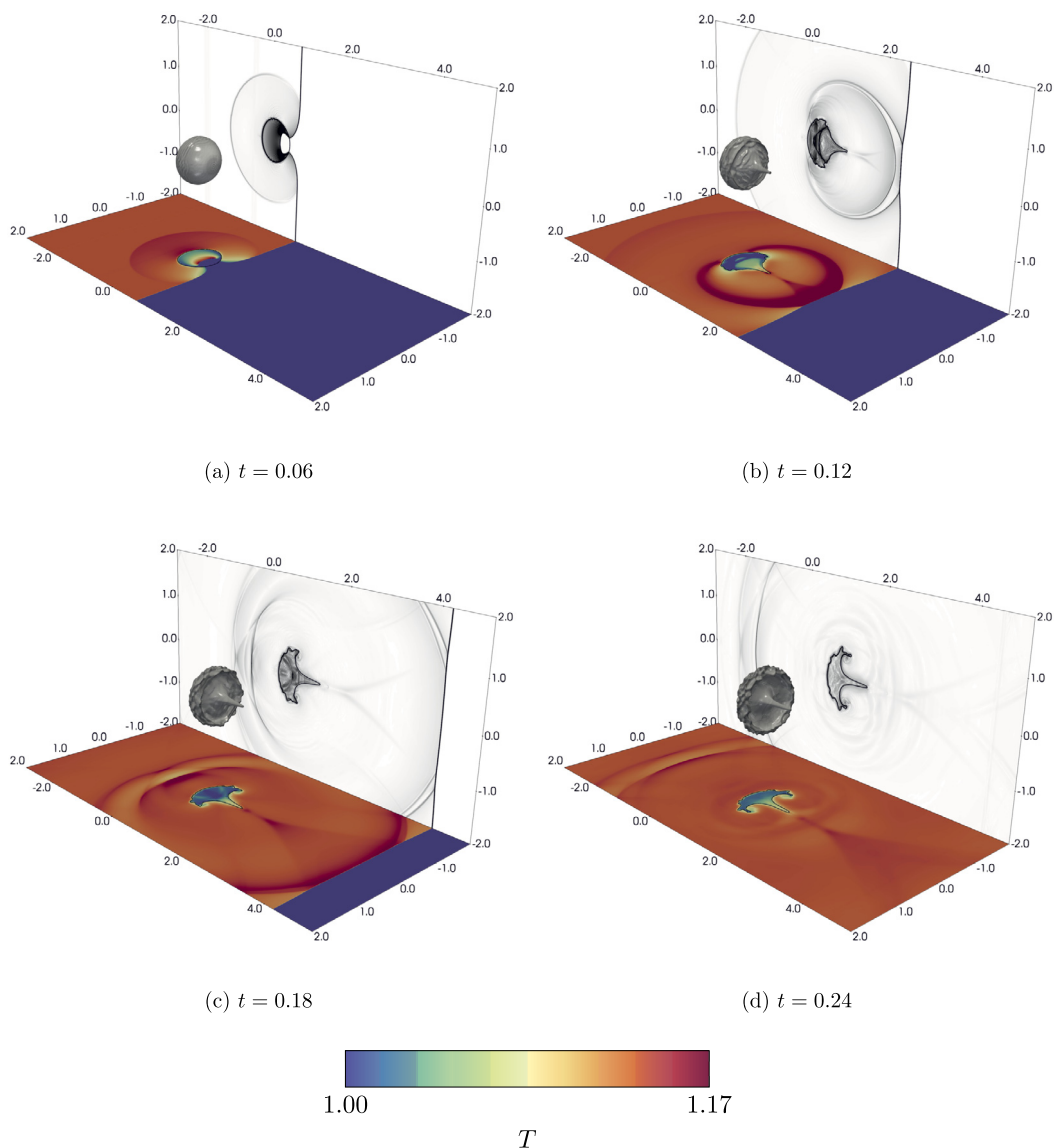


Fig. 15. The flow field in the simulated domain and the interface contour for different time instances. Numerical schlieren in the x - y center plane are projected to the upper right. Similarly, the temperature in the x - z center plane is shown in the bottom. The corresponding legend is given at the bottom. The contour of the R22 bubble is shown in gray. Light gray indicates a low interface velocity. Dark gray indicates a high interface velocity.

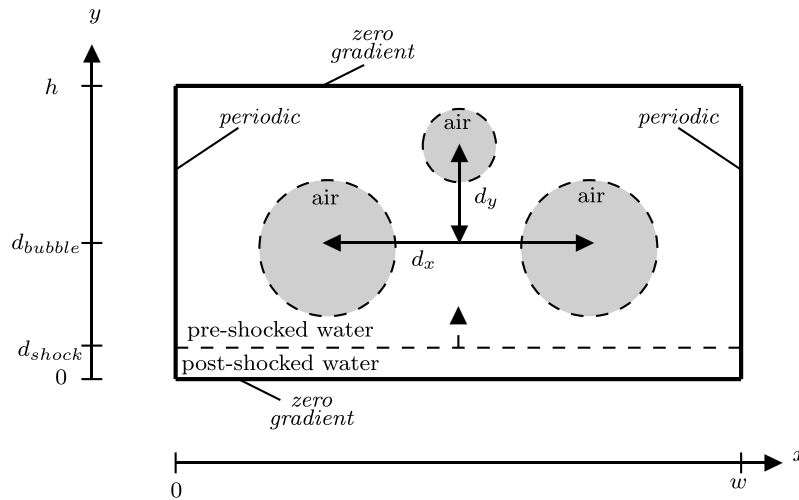


Fig. 16. Two-dimensional projection of the three-bubble setup. The width and height of the computational domain are $w = 0.0035$ and $h = 0.002$, respectively. The initial position of the shock and bigger bubbles are $d_{shock} = 0.00015$ and $d_{bubble} = 0.00075$, respectively. The initial distances between the bubbles are $d_x = 0.0015$ and $d_y = 0.0005$, respectively.

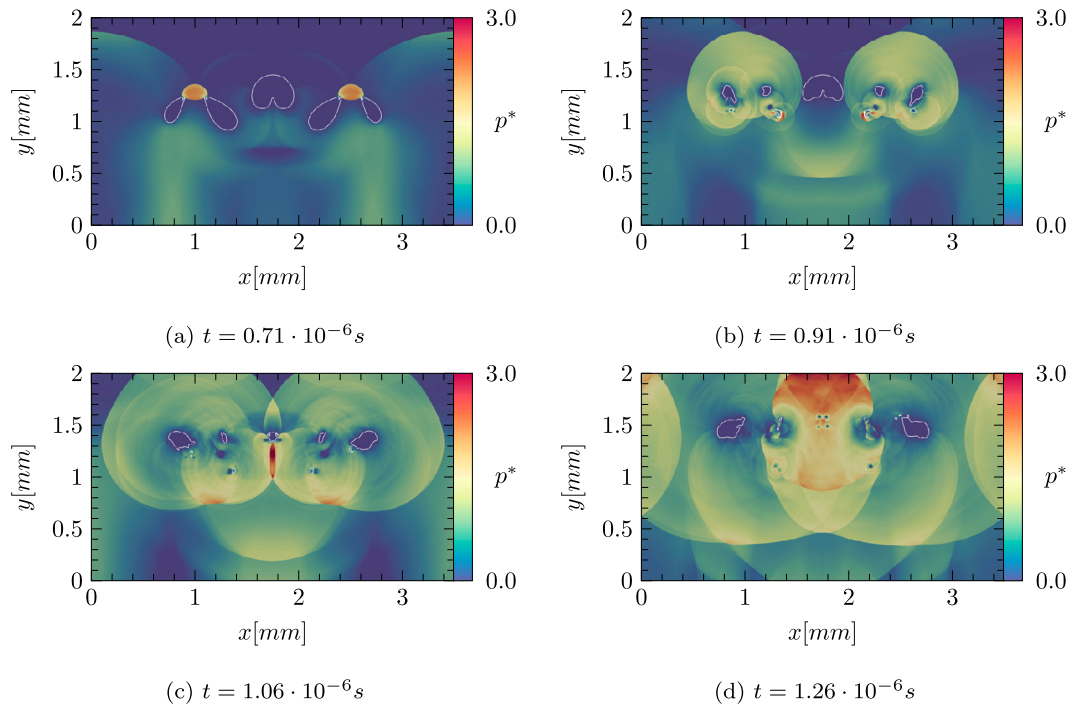


Fig. 17. The pressure field in the simulated domain for different time instances. The interface is shown as a white line.

9. Performance

We show the parallel performance of our implementation by means of profilings, and weak- and strong-scaling studies. We use two most differently configured executables, in terms of the modular algorithmic building blocks, cf. Table 2. Thereby, we show that performance is maintained across different variants. We refer to these two executables as ‘standard’ and ‘shuffle’. The standard configuration coincides with the default configuration of the previous Section 8. For the shuffle configuration, we employ HOUC level-set advection using the HOUC-5 derivative stencil, Min reinitialization, iterative upwind extrapolation, HLLC interface Riemann solver, WENO-7 reconstruction stencil, HLLC Riemann solver, and RK-2 time integration. Viscous and heat terms are calculated the same as in the standard configuration.

We conducted all measurements on ‘SuperMUC-NG’ at the Leibniz-Rechenzentrum. It consists of Intel Xeon Platinum 8174 Codename “Skylake” compute nodes. Each node consists of 48 cores running at a base frequency of 2.7 GHz. Nodes are connected via a 100 Gbits⁻¹ OmniPath network.² To counteract measurement uncertainties, we conduct each simulation three times and report the average. The variance across all runs was < 2%.

First, we determine the overhead of the multi-phase treatment using Intel’s “VTune Amplifier 2019 Update 6”. Therefore, we set up a domain consisting of a single block with an interface in the middle. We run it on a single core. For the standard and the shuffle configuration, the treatment of the level set accounts for roughly

² <https://doku.lrz.de/display/PUBLIC/SuperMUC-NG>.

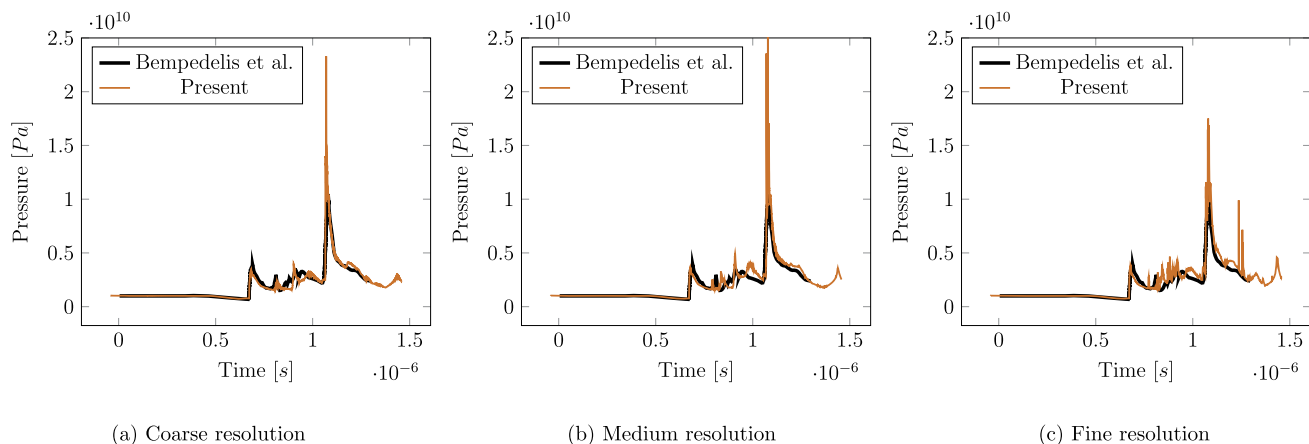


Fig. 18. Evolution of the maximum pressure inside the domain. From left to right, we show coarse (896×512 effective cells), medium (1792×1024 effective cells) and fine (3584×2048 effective cells) results, respectively. For comparison, we show results of [98] in black.

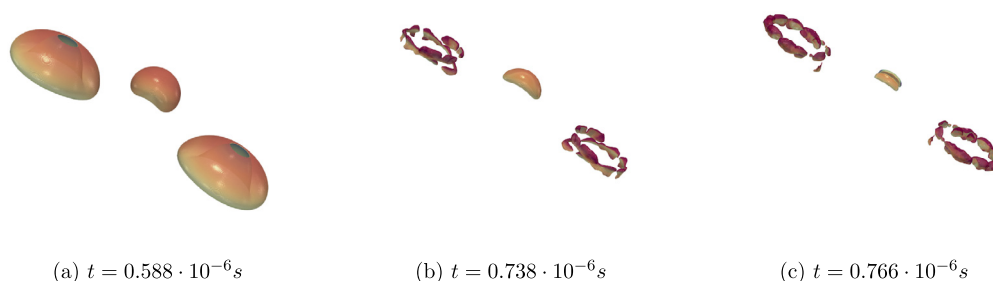


Fig. 19. Three-dimensional bubble contours colored by interface velocity. Blue indicates a low interface velocity. Red indicates a high interface velocity. The shock travels from the lower left to the upper right. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

26% of the runtime. We also simulate a more realistic test case with a single resolution jump from $l_{max} = 3$ to the coarsest level. It is run with 24 MPI ranks. The standard configuration still spends about 25% in subroutines related to the level-set treatment. In the shuffle configuration this fraction reduces to 20%. In this case, 8.5% and 9.3% of the runtime are spent in MR related routines for standard and shuffle configuration, respectively. For all cases, however, the major part of the runtime ($> 52 - 66\%$) is spent in the flux computation. Our multi-block setup allows using our well-optimized SIMD-capable modular implementation for single-phase flux computations [50].

9.1. Weak scaling

We use a Sod shock-tube problem [99] for our weak-scaling analysis. The density, pressure and velocity states at $t = 0.2$ are given in Fig. 20a. The initial discontinuity in the middle of the domain is also the initial position of the interface. We model the phases as ideal gas with $\gamma = 1.4$. We use two blocks on l_0 and set $l_{max} = l_4$ for the single-node setup. This yields an effective resolution of $512 \times 512 \times 1024$ cells. The quasi one-dimensionality of the problem allows to precisely increase the numerical load. Regardless the adaptive mesh size, the load grows proportional to the number of compute nodes. We set up the problem along the z-axis and increase the domain size in the “superfluous” dimensions as needed. The increased workload for 512 compute nodes results in a problem size of roughly 685 billion degree of freedom (DOFs). In this case, despite the MR compression, 100 billion DOFs are actually computed cf. Section 10.

In the scaling runs, we reduce the amount of time steps. We advance to $t_{end} = 0.013$ with a constant CFL = 0.95. In Fig. 20b, the weak-scaling efficiency η_w for both configurations is given. In both configurations, we observe a reasonable scaling behavior.

The scaling efficiency stays above 75% for all workloads. The shuffle configuration shows better efficiency for smaller node counts. However, it is surpassed by the standard configuration for high node counts. We credit this behavior to the differences in the proportion of compute load to communication. This difference stems mostly from the different amount of RK stages.

9.2. Strong scaling

For the strong scaling runs, we extend the two-dimensional RMI setup from Section 8.3 to three dimensions. We start the simulation at the point where the initial shock hits the interface. From there, we simulate twelve macro time steps. We vary the resolution by increasing $l_{max} \in [l_0, l_5]$. The effective problem size thus varies from roughly 2.6 million to 85 billion DOFs. Again, we use the two distinct ALPACA configurations: standard and shuffle. Each resolution is run with different core numbers. We record the time spent per computed cell in Fig. 21.

The scaling behavior for both configurations is very similar. The shuffle configuration requires roughly 2/3 of the time-per-cell of the standard configuration. This matches our expectation as a two-, respectively three-stage RK scheme is used. For every l_{max} the curve levels off for higher core counts. This is due to starvation of the additional added cores. However, for an increased l_{max} the proper scaling continues. Independent of the problem size, the runtime per cell and core stays practically the same.

10. Achieved compression

Here, we investigate the compute and memory compression of our coupled MR and narrow-band approach. Compression is achieved by the single-phase block-based MR scheme, cf. Section 5.

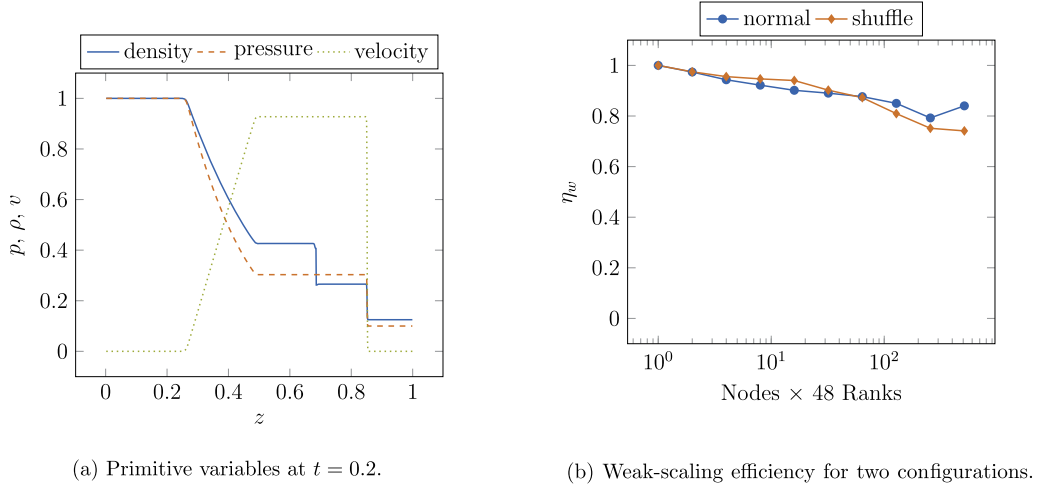


Fig. 20. Two-phase Sod shock tube: Final states and weak scaling efficiency.

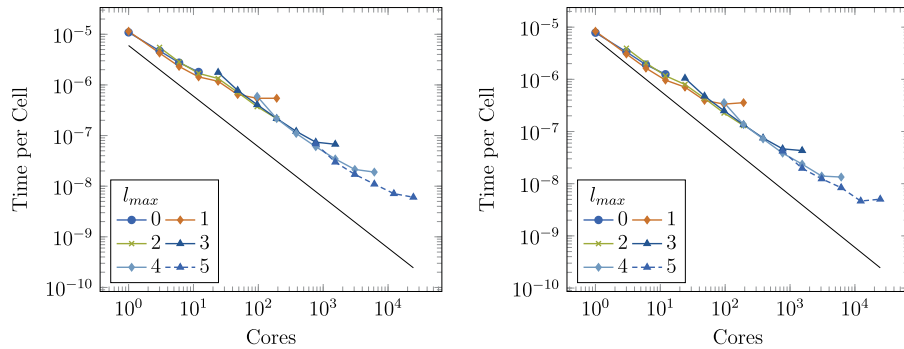


Fig. 21. Strong scaling study for the standard (left) and the shuffle (right) configuration. The black line indicates theoretical ideal scaling.

Furthermore, we restrict the additional memory and compute demand of the multi-phase treatment. This restriction is incorporated horizontally and vertically in the MR topology, cf. Section 6.

To quantify the savings, we define four normalized (higher is better) compression metrics:

1. Compression of total cells C_{mr} : The fraction of all actually stored cells to the effective cells on a homogenous mesh for the same resolution. C_{mr} hence gives the overhead associated with storing ancestor nodes.
2. Compression of flux computations C_{flux} : It is the fraction of leaf cells to needed cells on a homogenous mesh. This represents the saving of costly numerical-flux function evaluations.
3. Vertical multi-phase compression C_v : It is the fraction of multi-phase cells to a homogenous mesh equivalent. It represents the saved memory by vertically restricting the multi-phase treatment.
4. Level-set compression C_{ls} : It is the fraction of cells holding a level-set value to a homogenous level-set representation. It represents the saved memory by horizontally restricting the multi-phase treatment.

Fig. 22 shows the obtained compression rates for the three-dimensional cases presented in this work. We abbreviate those as ‘Sod’, ‘Helium’, ‘Three bubble’ and ‘R22’ according to Sections 8.4 to 8.6 and 9.1, respectively.

We observe very high compression rates in all four metrics. The higher l_{max} , the higher the compression. The physical phenomena taking place in the simulated domain are also observable in the compression plots. The saw-tooth pattern in the Sod case stems from the traversing interface and shock front. For the Three-bubble

case, C_v and C_{ls} increase as the bubbles disintegrate. Inversely, C_{mr} and C_{flux} decrease due to the increasing amount of shock fronts in the domain. The compression for the other cases remains constantly high. No severe trends are observable. Overall, the high compression rates C_{mr} and C_{flux} show that the exclusion of the interface region from the MR thresholding does not affect compression considerably. At the same time, the even higher compressions $C_v > 90\%$ and $C_{ls} > 92\%$ indicate the advantages of the tagging system.

11. Conclusion and outlook

We presented a simulation environment that allows solving multi-phase hyperbolic and (incompletely) parabolic conservation laws. They are solved using high-order shock-capturing finite-volume schemes and the sharp-interface level-set method. For the first time, we have coupled a block-based MR approach with narrow-band like compression techniques. This is achieved by introducing a light-weight tagging system. This algorithm reduces compute and memory overhead of the multi-phase scheme to a minimum. The modern OO implementation allows exchange of numerical building blocks at compile time without loss of parallel performance. Scaling results were shown on a state-of-the-art HPC system. Various challenging test cases were simulated and compared to relevant literature. Therein, we reach resolutions of $\mathcal{O}(10^{10})$ effective DOFs with compression rates $> 90\%$.

For the future, we plan to develop a hybrid parallelization utilizing shared-memory parallelism and to integrate further numerical interface models.

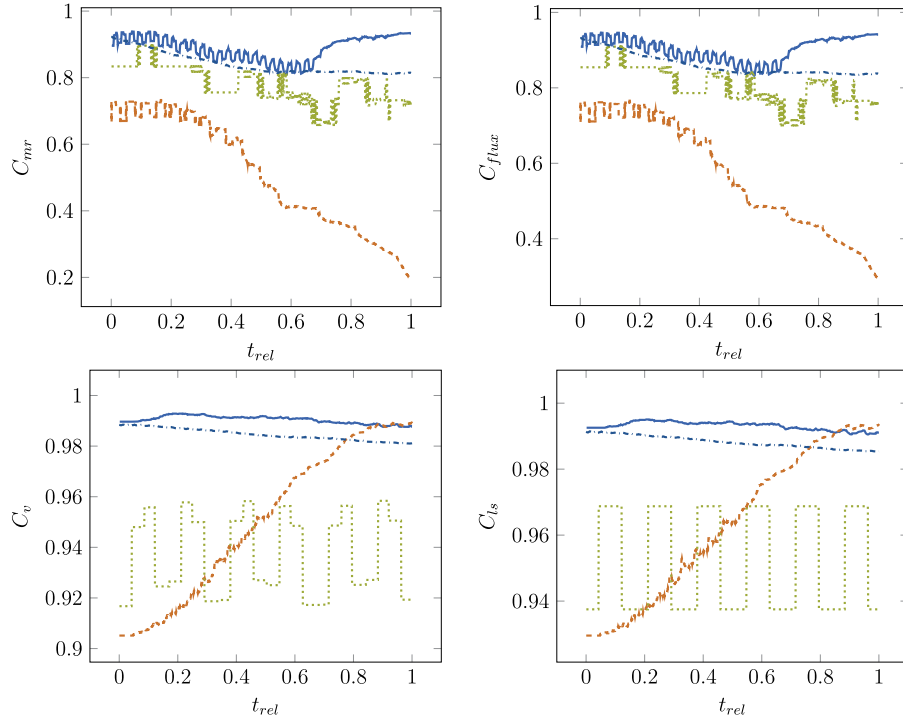


Fig. 22. Compression of the three-dimensional cases over relative time. In all plots \cdots refers to the Helium case, \cdots to the Sod case, $-\cdot- to the Three bubble case, and $—$ to the R22 case. Mind the different scales for the y-axes.$

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 667483).

The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project by providing computing time on the GCS Supercomputer SuperMUC-NG at Leibniz-Rechenzentrum.

Appendix A. Small-cell mixing

The time-step restriction according to the CFL criterion, cf. Section 3.3, only holds for full cells. When integrating small cells, i.e. cells cut by the interface, we employ a small-cell mixing approach following [57,58,66]. This avoids instabilities and spurious oscillations. We define mixing weights $\beta_{i+l,j+m,k+n}$, $(l, m, n) \in \{-1, 0, +1\}$ for any small cell $\Delta_{i,j,k}$ requiring mixing. Mixing weights are defined for each neighboring cell. The mixing weights are normalized such that $\sum_{l,m,n} \beta_{i+l,j+m,k+n} = 1.0$. Cells with $\beta_{i+l,j+m,k+n} \neq 0.0$ are referred to as target cells Δ_{trg} . The flux vector between a small cell and a target cell reads

$$\mathbf{M}_{trg} = \frac{\beta_{trg}}{\alpha_{i,j,k}^{\xi_a} * \beta_{trg} + \alpha_{trg}^{\xi_a}} \cdot \left[(\alpha^{\xi_a} \bar{\mathbf{U}})_{trg} \alpha_{i,j,k}^{\xi_a} - (\alpha^{\xi_a} \bar{\mathbf{U}})_{i,j,k} \alpha_{trg}^{\xi_a} \right]. \quad (\text{A.1})$$

It allows calculating mixed conservative states $(\alpha^{\xi_a} \bar{\mathbf{U}})^m$. This is done using the conservative update

$$(\alpha^{\xi_a} \bar{\mathbf{U}})_{i,j,k}^m = (\alpha^{\xi_a} \bar{\mathbf{U}})_{i,j,k}^m + \mathbf{M}_{trg}, \quad (\text{A.2})$$

$$(\alpha^{\xi_a} \bar{\mathbf{U}})_{trg}^m = (\alpha^{\xi_a} \bar{\mathbf{U}})_{trg}^m - \mathbf{M}_{trg}. \quad (\text{A.3})$$

Two cell types require mixing:

1. Cells with $\alpha_{i,j,k}^{\xi_a} \leq \alpha_{mix}$ after integration.
2. Cells where $\alpha_{i,j,k}^{\xi_a} = 0.0$ after integration but $\alpha_{i,j,k}^{\xi_a} \neq 0.0$ before.

For the first type of cells we use $\beta_{i+l,j+m,k+n} = A_{i+\frac{l}{2},j+\frac{m}{2},k+\frac{n}{2}}^{\xi_a}$. For the second type of cells, mixing weights are computed according to [58].

Appendix B. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cpc.2021.108246>.

References

- [1] R.J. LeVeque, Numerical Methods for Conservation Laws, 2nd ed., Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, Boston, 1992.
- [2] M.J. Lighthill, G.B. Whitham, Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci. 229 (1955) 317–345.
- [3] P.I. Richards, Oper. Res. 4 (1956) 42–51.
- [4] A.J.-C. de Saint-Venant, et al., C. R. Acad. Sci. Paris 73 (1871) 5.
- [5] E.F. Toro, Shock-Capturing Methods for Free-Surface Shallow Flows, John Wiley, 2001.
- [6] J. Stoker, R.B. Lindsay, PhT 11 (1958) 28.
- [7] C.-D. Munz, P. Ommes, R. Schneider, Comput. Phys. Commun. 130 (2000) 83–117.
- [8] H.W. Liepmann, A. Roshko, Elements of Gasdynamics, John Wiley & Sons, Inc., New York, 1957.
- [9] G.C. Messina, P. Wagener, R. Streubel, A. De Giacomo, A. Santagata, G. Compagnini, S. Barcikowski, Phys. Chem. Chem. Phys. 15 (2013) 3093–3098, <https://doi.org/10.1039/C2CP42626A>.
- [10] T.E. Itina, J. Phys. Chem. C 115 (2011) 5044–5048, <https://doi.org/10.1021/jp1090944>.

- [11] B. Budich, S.J. Schmidt, N.A. Adams, in: *MARINE VI: Proceedings of the VI International Conference on Computational Methods in Marine Engineering, CIMNE, 2015*, pp. 709–721.
- [12] C. Chaussy, E. Schmiedt, *Urol. Radiol.* 6 (1984) 80–87.
- [13] K. Maeda, W. Kreider, A. Maxwell, B. Cunitz, T. Colonius, M. Bailey, *J. Phys. Conf. Ser.* 656 (2015) 012027, <https://doi.org/10.1088/1742-6596/656/1/012027>.
- [14] P.L. Roe, *J. Comput. Phys.* 43 (1981) 357–372.
- [15] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, 3rd ed., Springer, Dordrecht, New York, 2009.
- [16] N. Fleischmann, S. Adami, X.Y. Hu, N.A. Adams, *J. Comput. Phys.* 401 (2020) 109004, <https://doi.org/10.1016/j.jcp.2019.109004>.
- [17] N. Fleischmann, S. Adami, N.A. Adams, *J. Comput. Phys.* 423 (2020) 109762.
- [18] G.S. Jiang, C.W. Shu, *J. Comput. Phys.* 126 (1996) 202–228.
- [19] L. Fu, X.Y. Hu, N.A. Adams, *J. Comput. Phys.* 305 (2016) 333–359, <https://doi.org/10.1016/j.jcp.2015.10.037>.
- [20] S.H. Bryngelson, K. Schmidmayer, V. Coralic, J.C. Meng, K. Maeda, T. Colonius, *Comput. Phys. Commun.* (2020) 107396, <https://doi.org/10.1016/j.cpc.2020.107396>.
- [21] E. Johnsen, F. Ham, *J. Comput. Phys.* 231 (2012) 5705–5717.
- [22] G. Perigaud, R. Saurel, *J. Comput. Phys.* 209 (2005) 139–178.
- [23] K.-M. Shyue, *J. Comput. Phys.* 156 (1999) 43–88.
- [24] G. Ball, B. Howell, T. Leighton, M. Schofield, *Shock Waves* 10 (2000) 265–276.
- [25] C. Turangan, G. Ball, A. Jamaluddin, T. Leighton, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 473 (2017) 20170315.
- [26] J. Glimm, J.W. Grove, X.L. Li, K.-M. Shyue, Y. Zeng, Q. Zhang, *SIAM J. Sci. Comput.* 19 (1998) 703–727, <https://doi.org/10.1137/S1064827595293600>.
- [27] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahy, W. Tauber, J. Han, S. Nas, Y.-J. Jan, *J. Comput. Phys.* 169 (2001) 708–759.
- [28] S. Osher, J.A. Sethian, *J. Comput. Phys.* 79 (1988) 12–49, [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2).
- [29] F. Gibou, R. Fedkiw, S. Osher, *J. Comput. Phys.* 353 (2018) 82–109.
- [30] J.A. Sethian, P. Smereka, *Annu. Rev. Fluid Mech.* 35 (2003) 341–372.
- [31] K. Schneider, O.V. Vasilyev, *Annu. Rev. Fluid Mech.* 42 (2010) 473–503, <https://doi.org/10.1146/annurev-fluid-121108-145637>.
- [32] D. Rossinelli, B. Hejazialhosseini, P. Hadjidoukas, C. Bekas, A. Curioni, A. Bertsch, S. Futral, S.J. Schmidt, N.A. Adams, P. Koumoutsakos, in: *SC13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE, 2013*, pp. 1–13.
- [33] S. Pirozzoli, *Annu. Rev. Fluid Mech.* 43 (2011) 163–194, <https://doi.org/10.1146/annurev-fluid-122109-160718>.
- [34] H. Sutter, Dr. Dobb's J. (2005) 1–9, <https://doi.org/10.1002/minf.201100042>.
- [35] G. Hager, G. Wellein, *Introduction to high performance computing for scientists and engineers*, <https://doi.org/10.1201/EBK1439811924>, 2011.
- [36] A. Harten, *Commun. Pure Appl. Math.* 48 (1995) 1305–1342, <https://doi.org/10.1002/cpa.3160481201>.
- [37] A. Cohen, S. Kaber, S. Müller, M. Postel, *Math. Comput.* 72 (2001) 183–225.
- [38] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, *J. Comput. Phys.* 188 (2003) 493–523, [https://doi.org/10.1016/S0021-9991\(03\)00189-X](https://doi.org/10.1016/S0021-9991(03)00189-X).
- [39] M.J. Berger, P. Colella, *J. Comput. Phys.* 82 (1989) 64–84.
- [40] S. Qian, J. Weiss, *J. Comput. Phys.* 106 (1993) 155–175, <https://doi.org/10.1006/jcph.1993.1100>.
- [41] O.V. Vasilyev, C. Bowman, *J. Comput. Phys.* 165 (2000) 660–693, <https://doi.org/10.1006/JCPH.2000.6638>.
- [42] R. Deiterding, M.O. Domingues, K. Schneider, *Comput. Fluids* 205 (2020) 104583, <https://doi.org/10.1016/j.compfluid.2020.104583>.
- [43] A. Nejadmalayeri, A. Vezolainen, E. Brown-Dymkoski, O.V. Vasilyev, *J. Comput. Phys.* 298 (2015) 237–253, <https://doi.org/10.1016/j.jcp.2015.05.028>.
- [44] B.L. Bihari, A. Harten, *SIAM J. Sci. Comput.* 18 (1997) 315–354.
- [45] B.L. Bihari, *J. Comput. Phys.* 123 (1996) 207–225, <https://doi.org/10.1006/jcph.1996.0017>.
- [46] M.O. Domingues, S.M. Gomes, O. Roussel, K. Schneider, *J. Comput. Phys.* 227 (2008) 3758–3780, <https://doi.org/10.1016/j.jcp.2007.11.046>.
- [47] J.W. Kaiser, N. Hoppe, S. Adami, N.A. Adams, *J. Comput. Phys.* X 4 (2019) 100038.
- [48] B. Hejazialhosseini, D. Rossinelli, M. Bergdorf, P. Koumoutsakos, *J. Comput. Phys.* 229 (2010) 8364–8383, <https://doi.org/10.1016/j.jcp.2010.07.021>.
- [49] W. Eckhardt, T. Weinzierl, in: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, in: LNCS, vol. 6067, Springer, Berlin, Heidelberg, 2010, pp. 567–575.
- [50] N. Hoppe, I. Pasichnyk, M. Allalen, S. Adami, N.A. Adams, in: *2019 International Conference on High Performance Computing Simulation (HPCS), 2019*, pp. 732–740.
- [51] D. Adalsteinsson, J.A. Sethian, *J. Comput. Phys.* 118 (1995) 269–277.
- [52] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, *J. Comput. Phys.* 155 (1999) 410–438.
- [53] D. Chopp, *J. Comput. Phys.* 106 (1993) 77–91, <https://doi.org/10.1006/jcph.1993.1092>, cited by 357.
- [54] J. Luo, X. Hu, N.A. Adams, *J. Comput. Phys.* 284 (2015) 547–565.
- [55] N. Hoppe, S. Adami, N.A. Adams, A modular massively parallel computing environment for three-dimensional multiresolution simulations of compressible flows, arXiv:2012.04385, 2020.
- [56] S. Osher, R.P. Fedkiw, *J. Comput. Phys.* 169 (2001) 463–502.
- [57] E. Lauer, X.Y. Hu, S. Hicckel, N.A. Adams, *Comput. Fluids* 69 (2012) 1–19.
- [58] X.Y. Hu, B. Khoo, N.A. Adams, F. Huang, *J. Comput. Phys.* 219 (2006) 553–578.
- [59] M. Sussman, P. Smereka, S. Osher, *J. Comput. Phys.* 114 (1994) 146–159.
- [60] S. Osher, R.P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 200, Springer, New York, 2005.
- [61] C. Runge, *Math. Ann.* 46 (1895) 167–178, <https://doi.org/10.1007/BF01446807>.
- [62] W. Kutta, *Z. Math. Phys.* 46 (1901) 435–453.
- [63] A. Harten, *J. Comput. Phys.* 49 (1983) 357–393, [https://doi.org/10.1016/0021-9991\(83\)90136-5](https://doi.org/10.1016/0021-9991(83)90136-5).
- [64] M. Sussman, E.G. Puckett, *J. Comput. Phys.* 162 (2000) 301–337.
- [65] J.U. Brackbill, D.B. Kothe, C. Zemach, *J. Comput. Phys.* 100 (1992) 335–354.
- [66] F.S. Schraner, *Weakly Compressible Models for Complex Flows*, Ph.D. thesis, Technische Universität München, 2016.
- [67] C. Min, *J. Comput. Phys.* 229 (2010) 2764–2772.
- [68] G.-S. Jiang, D. Peng, *SIAM J. Sci. Comput.* 21 (2000) 2126–2143.
- [69] R. Saurel, S. Gavriluk, F. Renaud, *J. Fluid Mech.* 495 (2003) 283–321.
- [70] X. Hu, N. Adams, E. Johnsen, G. Iaccarino, in: *Proceedings of the Summer Program, 2008*, p. 183.
- [71] E.F. Toro, M. Spruce, W. Speares, *Shock Waves* 4 (1994) 25–34.
- [72] B. Einfeldt, *SIAM J. Numer. Anal.* 25 (1988) 294–318.
- [73] S. Davis, *SIAM J. Sci. Stat. Comput.* 9 (1988) 445–473.
- [74] V. Coralic, T. Colonius, *J. Comput. Phys.* 274 (2014) 95–121.
- [75] S. Gottlieb, C.-W. Shu, *Math. Comput.* 67 (1998) 73–85.
- [76] D.S. Balsara, S. Garain, C.-W. Shu, *J. Comput. Phys.* 326 (2016) 780–804, <https://doi.org/10.1016/j.jcp.2016.09.009>.
- [77] X.Y. Hu, Q. Wang, N.A. Adams, *J. Comput. Phys.* 229 (2010) 8952–8965, <https://doi.org/10.1016/j.jcp.2010.08.019>.
- [78] C.W. Shu, in: *High-Order Methods for Computational Physics*, Springer, 1999, pp. 439–582.
- [79] F. Zeng, Y. Shen, S. Liu, L. Liu, *Int. J. Numer. Methods Fluids* 91 (2019) 159–182.
- [80] R. Borges, M. Carmona, B. Costa, W.S. Don, *J. Comput. Phys.* 227 (2008) 3191–3211.
- [81] D.S. Balsara, C.-W. Shu, *J. Comput. Phys.* 160 (2000) 405–452.
- [82] R.R. Nourgaliev, T.G. Theofanous, *J. Comput. Phys.* 224 (2007) 836–866, <https://doi.org/10.1016/j.jcp.2006.10.031>.
- [83] F.H. Harlow, A.A. Amsden, *Fluid Dynamics. A LASL Monograph*, Technical Report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), 1971.
- [84] O. Le Métayer, R. Saurel, *Phys. Fluids* 28 (2016) 046102.
- [85] R. Menikoff, B. Plohr, *Rev. Mod. Phys.* 61 (1989) 75130.
- [86] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, et al., *J. Comput. Phys.* 152 (1999) 457–492.
- [87] G.I. Taylor, *Proc. R. Soc. Lond. Ser. A, Contain. Pap. Math. Phys. Character* 146 (1934) 501–523.
- [88] M. Brouillette, *Annu. Rev. Fluid Mech.* 34 (2002) 445–468.
- [89] H. Terashima, G. Tryggvason, *J. Comput. Phys.* 228 (2009) 4012–4037.
- [90] R.R. Nourgaliev, T.-N. Dinh, T.G. Theofanous, *J. Comput. Phys.* 213 (2006) 500–529.
- [91] J. Picone, J. Boris, *J. Fluid Mech.* 189 (1988) 23–51.
- [92] J.J. Quirk, S. Karni, *J. Fluid Mech.* 318 (1996) 129–163, <https://doi.org/10.1017/S0022112096007069>.
- [93] C.-H. Chang, M.-S. Liou, *J. Comput. Phys.* 225 (2007) 840–873.
- [94] J.-F. Haas, B. Sturtevant, *J. Fluid Mech.* 181 (1987) 41–76.
- [95] K. Kannan, D. Kedely, M. Herrmann, *J. Comput. Phys.* 373 (2018) 784–810.
- [96] W. Sutherland, *Philos. Mag.* 36 (1893) 507–531, <https://doi.org/10.1080/14786449308620508>.
- [97] J. Kaiser, J. Winter, S. Adami, N. Adams, *Int. J. Multiph. Flow* 132 (2020) 103409, <https://doi.org/10.1016/j.ijmultiphaseflow.2020.103409>.
- [98] N. Bempedelis, Y. Ventikos, *J. Fluid Mech.* 900 (2020).
- [99] G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, [https://doi.org/10.1016/0021-9991\(78\)90023-2](https://doi.org/10.1016/0021-9991(78)90023-2), 1978.

A.2 Stochastic multi-fidelity surrogate modeling of dendritic crystal growth



RightsLink

[Sign in/Register](#)

Stochastic multi-fidelity surrogate modeling of dendritic crystal growth

Author: J.M. Winter, J.W.J. Kaiser, S. Adami, I.S. Akhatov, N.A. Adams

Publication: Computer Methods in Applied Mechanics and Engineering

Publisher: Elsevier

Date: 1 April 2022

© 2022 Elsevier B.V. All rights reserved.

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: <https://www.elsevier.com/about/our-business/policies/copyright#Author-rights>

[BACK](#)[CLOSE WINDOW](#)



ELSEVIER



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 393 (2022) 114799

Computer methods
in applied
mechanics and
engineering

www.elsevier.com/locate/cma

Stochastic multi-fidelity surrogate modeling of dendritic crystal growth

J.M. Winter^{a,b,*}, J.W.J. Kaiser^a, S. Adami^{a,b}, I.S. Akhatov^c, N.A. Adams^{a,b}

^a Technical University of Munich, Department of Mechanical Engineering, Chair of Aerodynamics and Fluid Mechanics, Boltzmannstraße 15, 85748 Garching, Germany

^b Technical University of Munich, Munich Institute of Integrated Materials, Energy and Process Engineering (MEP), Lichtenbergstraße 4a, 85748 Garching, Germany

^c Skolkovo Institute of Science and Technology, Skolkovo Innovation Center, Bolshoy Boulevard 30, bld. 1, Moscow 121205, Russia

Received 8 September 2021; received in revised form 15 December 2021; accepted 20 February 2022

Available online xxxx

Abstract

In this work, we propose a novel framework coupling state-of-the-art multi-fidelity Gaussian Process modeling techniques with input-space warping for a cost-efficient construction of a stochastic surrogate model. During model generation, we achieve high computational efficiency by combining a large number of cheap estimates (low-fidelity model) with only a few, computationally expensive, high-fidelity measurements. We base the fidelity hierarchy on coarse-grid approximations of high-fidelity numerical simulations and show its successful application within the proposed framework. Utilizing coarse-grid approximations for multi-fidelity modeling is attractive for many practical applications, since it often allows for multi-fidelity data generation with a single simulator. As benchmark, we apply this framework to generate a surrogate model for crystal growth velocities in directional dendritic solidification. The derivation of a relation between this tip velocity and process parameters, such as undercooling, thermal diffusivity, capillarity, and capillary anisotropy, has been in the focus of research for decades due to its important role on microstructure evolution during solidification. It defines the thermo-mechanical properties of the solidified part and influences its behavior in subsequent manufacturing steps.

As data generator, we use the open-source simulation framework ALPACA, applying a conservative sharp-interface level-set model. We assess the accuracy of the multi-fidelity tip velocity model by using cross-validation techniques. Compared to single-fidelity models purely based on high-fidelity data, our approach improves prediction accuracy significantly but only requires a little cost overhead for data generation. The stochastic nature of the resulting surrogate model allows for quantifying the uncertainty associated with predictions. This motivates the application of the model in Bayesian-optimization algorithms for inverse problems. Also, it may serve as input for microstructure simulations which rely on accurate relations between local solidification velocities and process parameters such as undercooling to predict grain-scale crystalline structures and which need material-dependent model calibration.

© 2022 Elsevier B.V. All rights reserved.

Keywords: Gaussian processes; Multi-fidelity model; Stochastic surrogate modeling; Input warping; Dendritic growth; Multiresolution

* Corresponding author.

E-mail addresses: josef.winter@tum.de (J.M. Winter), jakob.kaiser@tum.de (J.W.J. Kaiser), stefan.adami@tum.de (S. Adami), i.akhatov@skoltech.ru (I.S. Akhatov), nikolaus.adams@tum.de (N.A. Adams).

<https://doi.org/10.1016/j.cma.2022.114799>

0045-7825/© 2022 Elsevier B.V. All rights reserved.

1. Introduction

The formation of dendritic microstructures and their defects plays a pivotal role in manufacturing, ranging from century-old techniques such as steel casting to modern additive manufacturing. Since microstructure and defects determine the mechanical properties of the solidified phase and, therefore, the final product, research has focused on both understanding and, in the long term, controlling the microstructure evolution. For a detailed overview, we refer to recent reviews by Kurz et al. [1,2]. Experimental investigations are aggravated by the extreme process conditions in the melt and the multi-scale nature of the problem, which is dominated by the macroscopic diffusion of temperature and solute concentration, macroscopic melt-pool convection, and microscopic phase interface characteristics. Therefore, hierarchical analytic and numerical models are essential to deal with this complexity and allow for systematic investigations [3]. In such a model hierarchy, simplified surrogate models with narrow focus play an important role as connection between different hierarchy levels.

Of particular interest is the relation between the operating point of the dendrite, i.e. the tip radius and velocity, and material and local melt parameters such as the undercooling [4]. A first mathematical model was provided in the seminal work of Ivantsov [5] for a two-dimensional parabolic dendrite, relating the product of tip radius and velocity to the melt undercooling. Yet, this corresponds to an infinite number of possible dendrites. Oldfield [6] solved this uniqueness problem by introducing a stability criterion based on the balance of capillarity and diffusion. A similar stability criterion was derived by Langer and Müller-Krumbhaar [7] applying a Mullins–Sekerka stability analysis, also considering the interplay of capillarity and diffusion. Several authors [8–16] proposed to include capillary anisotropy. This so-called microscopic solvability theory provides sufficiently accurate predictions for the operating point of the dendrite at high undercoolings [1], and allows for validation of numerical models. Such validated models may then provide insight into more complex directional growth configurations on the microscale, including multiple dendrites or varying material parameters which are not included in microscopic solvability theory.

For use in simulations on the grain scale or part scale, however, simplified surrogate models are needed [17]. Since these models imply many expensive simulations to adapt their free parameters to different material configurations, an efficient approach is required for model calibration that reduces the computational cost without loss of accuracy. Modern data-driven approaches based on statistical learning provide techniques to construct sophisticated surrogates that can accurately capture the response of complex engineering systems, see [18–21]. A popular example is GP regression surrogate models. Due to the increasing relevance of data-driven approaches in computational science and engineering, they are being applied in modeling or control of additive-manufacturing processes on the micro- and macro-scale [22–25] of, e. g. , composites [26] or ceramics [27,28]. Often, multi-fidelity modeling techniques allow to reduce the computational cost necessary to obtain datasets suitable for generating accurate surrogate models. Multi-fidelity models rely on the fact, that a high-fidelity numerical model with high accuracy but high computational cost can be supplemented by information from low-fidelity approximations. Low-fidelity models are less accurate but require considerably less computational effort [29]. Typical examples of low-fidelity models are coarse-grid approximations, previously developed experimental correlations, or models with reduced dimensionality [30,31]. Multi-fidelity methods are closely linked to multi-level methods. Both aim at increasing information augmenting the high-fidelity model by including information from models with lower fidelity. This is a widespread concept for efficient construction of response surfaces [31–34], uncertainty quantification [35–39] or solving inverse problems [40]. Multi-level methods typically vary a single parameter, such as the grid resolution of a high-fidelity numerical simulator, to derive a hierarchy of low-fidelity models [30]. Multi-fidelity methods allow more general low-fidelity models such as linearized models or regression models directly derived from high-fidelity data. A detailed overview about the variety of low-fidelity models can be found in [30]. Here, we focus on multi-fidelity GP regression for the cost-efficient construction of surrogate models.

The first multi-fidelity GP model was presented in the pioneering work of [41], combining GPs with a linear autoregressive scheme. This approach was successfully applied in scenarios where the correlation between the low- and high-fidelity model is linear [19,42–44]. However, many practical systems of interest exhibit linear correlations only for certain parameter ranges, while showing strong non-linear behavior otherwise [32]. Thus, several authors extended this work to nonlinear and space-dependent cross-correlations between fidelity levels [32,45,46]. Among them, [45] combined the linear autoregressive scheme of [41] with deep neural networks. The resulting model is especially suitable when fidelity levels show cross correlations with discontinuities. Motivated by the structure of deep GPs [47,48], the functional composition of GPs, [32,46] formulated an approach combining nonlinear autoregressive schemes and GP regression. This allows capturing nonlinear and space-dependent cross correlation

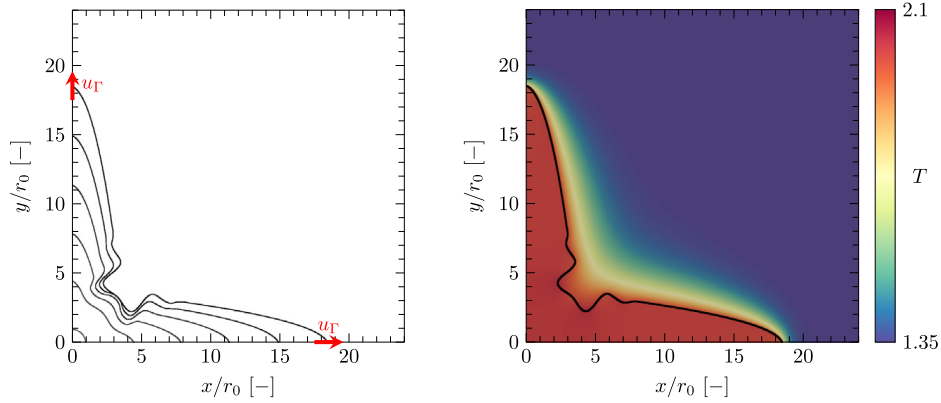


Fig. 1. Left: Solid–liquid interface for the growth of an initially spherical seed with radius r_0 under anisotropic surface tension. Time increases as the interface extends in x and y . The tip-velocity vector with magnitude u_T is highlighted in red for the last snapshot. Right: Temperature field and solid–liquid interface (black line) for the last snapshot.

between fidelity levels. We employ the MFDGP model of [46] as it reduces the risk of overfitting. The quality of standard GP regression models for non-stationary functions, i.e. functions where the covariance is not invariant to translation in input space, can be improved using input warping [49]. It has successfully been applied in many practical applications such as model-tuning frameworks [50] or to estimate loss reserves [51].

In this work, we propose a novel information-fusion framework coupling MFDGPs with input warping to cost-efficiently generate stochastic surrogate models. The input warping allows for incorporating prior knowledge about the relation between parameters and outcomes, and regularizes the input space. The transformed inputs are fed to the MFDGP model which captures nonlinear cross correlations between fidelity levels. The implementation of the model is realized with GPyTorch [52] and BoTorch [53]. We base the hierarchy of fidelity levels on coarse-grid approximations of the high-fidelity model. We employ a wavelet-based multiresolution scheme as data generator for all fidelities. This generator is implemented in the open-source simulation framework ALPACA [54–56]. Its error-based refinement strategy gives access to data on different fidelity levels through the same simulator, thus drastically simplifying the data generation process compared to using different data sources. We demonstrate the capabilities of the framework by generating a surrogate model for crystal growth velocities in directional dendritic solidification.

The structure of the paper is as follows. In Section 2 we describe the mathematical and physical model for crystal growth. In Section 3 we present the multi-fidelity model hierarchy, the MFDGP model, and the generated dataset. In Sections 4 and 5 we validate the multi-fidelity model and present the final stochastic surrogate model, respectively. Conclusions are given in Section 6.

2. Problem statement

The investigated setup is sketched in Fig. 1. An initially spherical solid seed is surrounded by an undercooled melt, i.e. the local temperature in the melt is below the liquid–solid phase transition temperature. This temperature difference results in the growth of the crystal. As latent heat is released during the phase change process, a temperature boundary layer forms in both the solid and the melt. The crystal yields preferred growth directions along the main coordinate axes due to anisotropic surface tension. After an initial transient growth period, the growth velocity at the dendritic tip converges against a steady-state solution. We choose a domain $(x, y) \in [0, 2400] \times [0, 2400]$, with the origin being at the center of the seed with initial radius $r_0 = 50$. The capillary anisotropy is modeled by the standard formulation

$$d(\theta) = d_0(1 - 15\varepsilon\cos(4\theta)), \tag{1}$$

where d_0 is the equilibrium capillary length, ε the capillary anisotropy factor, and θ the angle between the interface normal and the x -axis. This setup has been widely used with some variation for validation of numerical methods for crystal growth [57–62]. Microscopic solvability theory provides a reference solution for the steady-state tip velocity

and radius as a function of d_0 , ε , the thermal diffusivity α , and the normalized undercooling

$$\mathcal{T} = \frac{(T - T_m)c}{L}, \tag{2}$$

where T denotes the temperature in the melt, T_m the melting temperature, c the specific heat capacity, and L the latent heat. A stability criterion can be derived with

$$\sigma^* = \frac{2\alpha d_0}{u_\Gamma r_\Gamma^2} \cong \sigma_0 \varepsilon^{1.75}, \tag{3}$$

where σ_0 is a constant on the order of unity that requires calibration from numerical simulations or experiments [2,4]. The steady-state tip velocity u_Γ and the steady-state tip radius r_Γ follow from this stability criterion and the solution of Ivantsov [5] for parabolic dendrites in 2D

$$\mathcal{T} = \sqrt{\pi \text{Pe}} e^{\text{Pe}} \left(1 - \text{erf} \left(\sqrt{\text{Pe}} \right) \right), \tag{4}$$

where

$$\text{Pe} = \frac{u_\Gamma r_\Gamma}{2\alpha} \tag{5}$$

is the Peclet number. This approach provides reasonable accuracy [4], yet calibration from experimental or numerical data remains necessary, and material parameters have to be constant across the interface.

In our investigations, we choose $d_0 = 1$, $L = 1$, $c_m = c_s = c = 1$, and $\alpha_m \leq \alpha_s = 1$, allowing for a jump in the thermal conductivity across the phase interface. Therefore, the steady-state tip velocity can be written as function of undercooling, capillary anisotropy, and thermal-conductivity ratio following

$$u_\Gamma = f(\mathcal{T}, \varepsilon, \alpha_m/\alpha_s). \tag{6}$$

The objective is to develop a surrogate model for f in three-dimensional parameter space

$$\mathbf{x} = (\mathcal{T}, \varepsilon, \alpha_m/\alpha_s) \in \mathbb{R}^3 \tag{7}$$

with $0.5 \leq \mathcal{T} \leq 0.9$, $0.02 \leq \varepsilon \leq 0.05$, and $0.5 \leq \alpha_m/\alpha_s \leq 1$. We employ the numerical model for liquid–solid phase transition as presented in Kaiser et al. [63] to generate data for fitting the surrogate model. This simulation model is introduced in the following section.

3. Multi-fidelity framework

3.1. Hierarchy of models

The liquid–solid phase transition problem is governed by the continuity equation and the energy-diffusion equation with interface exchange terms and can be written as

$$\frac{\partial \mathbf{U}}{\partial t} = \nabla \cdot \mathbf{F}_k + \mathbf{X}. \tag{8}$$

Here, $\mathbf{U} = (\rho, \rho c T)^T$ is the state vector of mass and energy, $\mathbf{F}_k = (0, \rho c \alpha \nabla T)^T$ models heat fluxes, $\mathbf{X} = (X_m, X_E)^T$ denotes the interface exchange terms and ρ is the density. The interface exchange terms \mathbf{X} appear only at the sharp interface between melt and crystal. This phase interface is modeled by a level-set function ϕ which is evolved in time by solving the level-set advection equation

$$\frac{\partial \phi}{\partial t} = \mathbf{u}_\Gamma \cdot \nabla \phi = 0 \tag{9}$$

numerically. The derivation and validation of the interface exchange terms for the two-material Stefan problem, including the level-set advection velocity \mathbf{u}_Γ , are given in [63].

We apply a second-order central scheme to reconstruct thermal fluxes on a Cartesian grid. The solution is evolved in time by a third-order strongly-stable Runge–Kutta scheme [64,65]. Stability is enforced by computing the timestep size from the CFL stability criterion

$$\Delta t = \text{CFL} \cdot \min \left(\frac{\Delta x}{|u_\Gamma|}, \frac{3}{14} \frac{(\Delta x)^2}{\alpha} \right). \tag{10}$$

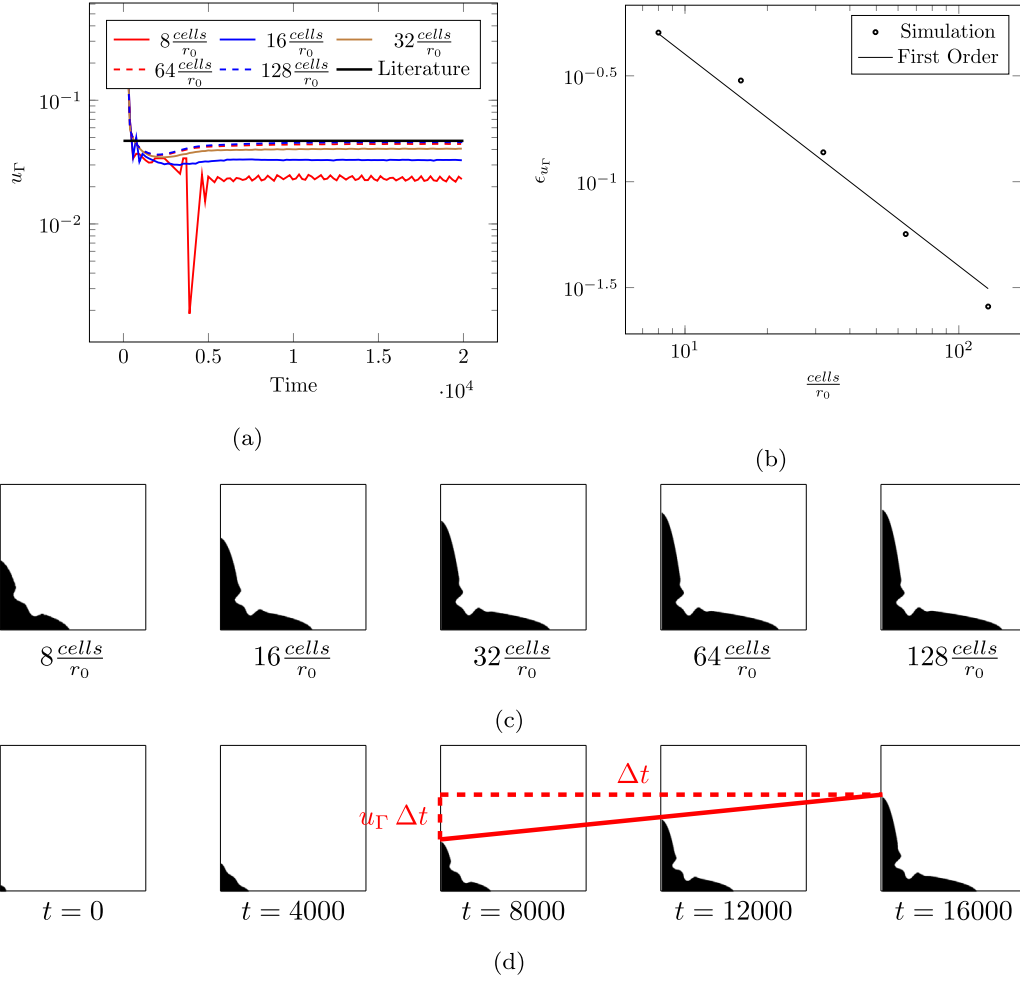


Fig. 2. (a): Velocity convergence for a single four-fold symmetric crystal for successive grid refinement. The simulation was conducted with $\mathcal{T} = 0.65$, $\epsilon = 0.05$, and $\frac{\alpha_m}{\alpha_s} = 1.0$. (b): Accuracy order analysis of the converged tip velocity. (c): Shape of the crystal at $t = 20000$ for the simulated resolutions. (d): Temporal evolution of the crystal for a resolution of $64 \frac{cells}{r_0}$.

Spatial and temporal efficiency is enforced through the wavelet based multiresolution (MR) scheme of Harten [66] with adaptive local timestepping for full temporal adaptivity [54]. For a detailed introduction to the applied computational framework, we refer to Hoppe et al. [55].

The level-dependent spatial resolution with level-dependent timestep sizes results in significant accuracy improvements for the predicted tip velocity using increasingly finer meshes. This is shown for a single crystal in Fig. 2. This setup applies $\Delta T = 0.65$, $\epsilon = 0.05$, and $\alpha_m = 1.0$, which is a typical test configuration with known steady-state tip velocity $u_{\Gamma,ref} = 0.047$ [57–60,67]. Independently of the chosen resolution, the initial condition with temperature jump at the interface results in an overprediction of the interface velocity. However, with evolving time, the velocity decreases and reaches a steady state. Fig. 2(b) shows the relative error $\epsilon_{u_\Gamma} = 1 - |\frac{u_\Gamma}{u_{\Gamma,ref}}|$ of the simulated steady-state tip velocity for different mesh resolutions. It converges with approximately first order for successively finer grids. For the underlying setup, the tips of the dendrite grow along the axes of the structured mesh. For a different problem setup, tips that would not grow along the grid axes may be affected by mesh anisotropy. The consistency of the numerical model for such scenarios is investigated in the Appendix.

To allow for a detailed analysis and validation of the resulting surrogate models, we aim for a dataset including many samples. Also, we aim to have accurate, i.e. highly resolved, high-fidelity measurements as basis for the surrogate model. Taking into account the available computational resources, we balance the accuracy of the high-fidelity model with the objective to generate many high-fidelity samples, and define the low-, intermediate- and high-fidelity models:

- **Low fidelity:** Simulations are performed with a spatial resolution of 8 cells per initial seed radius. This results in very short computing times, yet significant errors in the predicted steady-state velocity.
- **Intermediate fidelity:** Simulations are performed with a spatial resolution of 32 cells per initial seed radius. This corresponds to two additional refinement levels in the MR tree compared to the low-fidelity setup.
- **High fidelity:** Simulations are performed with a spatial resolution of 64 cells per initial seed radius, corresponding to three additional refinement levels in the MR tree compared to the low-fidelity setup.

3.2. Deep GPs for multi-fidelity modeling

Building the surrogate model for the tip velocity is a supervised learning problem. We aim to learn the mapping $f(\mathbf{x})$ between a set of N input vectors $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, where \mathbf{x}_i is the parameter vector described in Eq. (7) and observations $\mathbf{y} = [y_1, \dots, y_N]$, with $y_i = u_{\Gamma,i} \in \mathbb{R}$. We model observations with a homoscedastic noise model

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad (11)$$

assuming that they may be corrupted with zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The unknown function $f(\mathbf{x})$ is modeled using a GP prior $f(\mathbf{X}) \sim \mathcal{GP}(\mathbf{m}, \mathbf{K})$, i.e. any finite subset of input points is considered as a multivariate Gaussian with mean vector $\mathbf{m} \in \mathbb{R}^N$ and covariance matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$. The elements of the mean vector and covariance matrix are generated using the mean $\mathbf{m}_i = m(\mathbf{x}_i | \boldsymbol{\theta})$ and covariance function $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j | \boldsymbol{\theta})$, respectively, which typically are parametrized by hyperparameters $\boldsymbol{\theta}$. The mean and kernel function allow incorporating prior knowledge about the real model, such as monotonicity or periodicity. The posterior distribution of the GP describes a distribution over candidate functions characterized by a posterior mean and covariance, thus allowing predictions at unobserved data points. Consequently, the output $f^*(\mathbf{x}^*)$ at an unobserved input \mathbf{x}^* is normally distributed with mean μ^* and variance σ^{*2} , where the superscript $*$ indicates posterior values.

Multi-fidelity regression enables fusing limited true observations (*high-fidelity*) with many cheap approximate observations (*low-fidelity*) [30,41]. GPs are well suited for multi-fidelity regression since their stochastic nature allows modeling prior beliefs about the relation between fidelities and gives predictions accompanied by uncertainty estimates. [41,68] introduce multi-fidelity GPs based on the autoregressive scheme of [41] which assumes a linear relation between fidelities, i.e.

$$f_t(\mathbf{x}) = \beta_t f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}), \quad (12)$$

where f_t is the high-fidelity function and f_{t-1} the function at the preceding lower fidelity. Both are modeled using a GP prior. The constant β_t scales the contribution of samples drawn from the GP at the preceding lower fidelity $t-1$, and a GP prior is assigned to $\delta_t(\mathbf{x})$ which models the bias between fidelities. Such models deliver poor results when the cross-correlation between fidelities is nonlinear and input-space dependent [32]. [32,46] generalize the autoregressive multi-fidelity scheme of Eq. (12) to

$$f_t(\mathbf{x}) = \beta_t(f_{t-1}(\mathbf{x}), \mathbf{x}) + \delta_t(\mathbf{x}), \quad (13)$$

where the correlation between fidelities is not constant but expressed by the unknown function β_t . We assign a GP prior to β_t . Following [32,46], the additive structure and independence assumption between the GPs β_t and δ_t allow to rewrite the autoregressive scheme of Eq. (13) as a composition of GPs

$$f_t(\mathbf{x}) = g_t(f_{t-1}^*(\mathbf{x}), \mathbf{x}). \quad (14)$$

Therein, g_t is a GP which takes \mathbf{x} and a sample from the posterior of the GP modeling the preceding fidelity $f_{t-1}^*(\mathbf{x})$ as inputs. Based on deep GPs [47,69] and the compositional structure of Eq. (14) and [46] develop MFDGPs that accurately capture a nonlinear and space dependent relation between fidelities. We employ MFDGPs in settings with up to three fidelity levels. We use a dataset

$$\mathcal{D} = \{(\mathbf{X}^1, \mathbf{y}^1), (\mathbf{X}^2, \mathbf{y}^2), (\mathbf{X}^3, \mathbf{y}^3)\}, \quad (15)$$

where \mathbf{X}^t and \mathbf{y}^t denote inputs and corresponding outputs at fidelity level t . The fidelity increases with the superscript t . Given the training set we build the MFDGP model as sketched in the lower part of Fig. 3. The GP in the first layer only receives inputs from the input space. The GP in the second layer combines data from the input space as

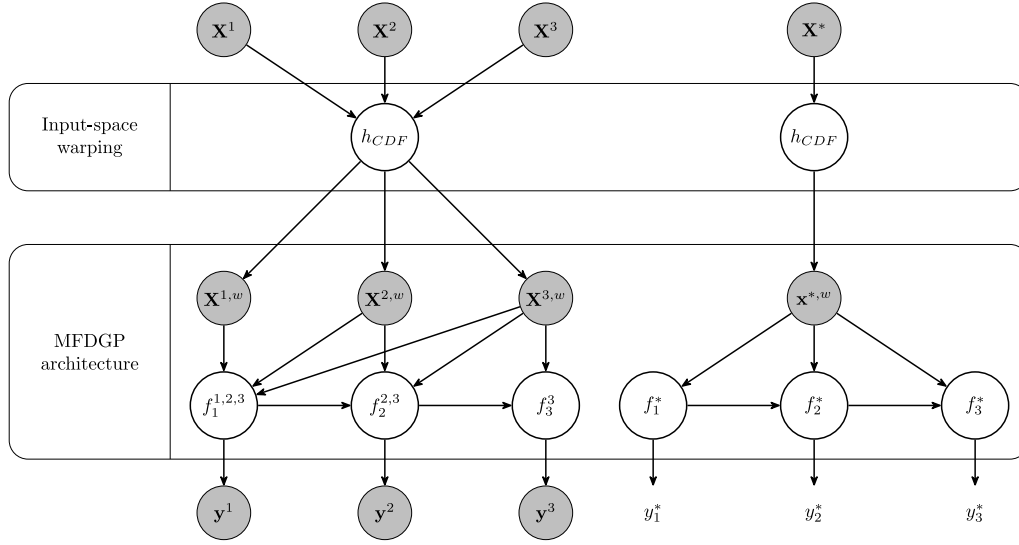


Fig. 3. Left: Combined input warping and MFDGP architecture with three fidelity levels. Gray nodes show inputs and outputs from the training set as labeled. White nodes show latent variables and denote samples drawn from a GP. Superscripts indicate the associated fidelity level. Subscripts indicate the layer. Thus, the evaluation of MFDGP at layer 1 for the inputs observed with fidelity 2 is denoted as f_1^2 . Right: Predictions using the combination of input warping and a MFDGP model with three fidelity levels. Each level gets the original input \mathbf{x}^* as well as evaluations up to the previous level as input. The output y_i^* denotes the prediction for fidelity t . Source: Picture adapted from [46].

well as the corresponding function evaluation returned from the previous layer. We denote the mean and standard deviation of the posterior distribution of the MFDGP at fidelity level t as $\bar{y}^{t,*}$ and $\sigma [y^{t,*}]$, respectively, see Fig. 3. We assume a zero mean function as prior for the GP at each level. Following [46], we define the covariance between two inputs \mathbf{x}^i and \mathbf{x}^j of the GP prior at each level as

$$k_l = k_l^\beta (\mathbf{x}^i, \mathbf{x}^j | \theta_l^\beta) \left[\sigma_l^2 f_{l-1}^*(\mathbf{x}^i)^T f_{l-1}^*(\mathbf{x}^j) + k_l^{f-1} (f_{l-1}^*(\mathbf{x}^i), f_{l-1}^*(\mathbf{x}^j) | \theta_l^{f-1}) \right] + k_l^\delta (\mathbf{x}^i, \mathbf{x}^j | \theta_l^\delta). \tag{16}$$

Therein, k_l^β is a input-space dependent scaling factor, k_l^{f-1} denotes the covariance between outputs obtained from the previous fidelity level, and k_l^δ captures the bias at the l th fidelity level. For each covariance function we assign a Matérn covariance function [70] with smoothness measure $\nu = 5/2$ and automatic relevance determination. The remaining term, $\sigma_l^2 f_{l-1}^*(\mathbf{x}^i)^T f_{l-1}^*(\mathbf{x}^j)$, considers a linear covariance between outputs obtained from the previous fidelity level, where σ_l is the variance hyperparameter. For the first layer, the covariance function reduces to

$$k_1 = k_1^\delta (\mathbf{x}^i, \mathbf{x}^j | \theta_1^\delta) \tag{17}$$

since no outputs from previous layers exist. To train the MFDGP model, we follow the two-step approach presented in [46] based on stochastic variational inference (SVI) techniques [71,72]. We assume a normal prior distribution for the noise parameter of the homoscedastic noise model $\sigma_\epsilon^2 \sim \mathcal{N}(0, \sigma_{\epsilon,prior}^2)$, with $\sigma_{\epsilon,prior} = 10^{-2}$. Since the integrals to calculate the posterior distribution are generally intractable [48], we use a sampling-based approach to obtain the posterior mean and covariance.

When using stationary kernel functions in a MFDGP, difficulties to model non-stationary functions with spatially varying length scales may arise. [49] propose to warp the input space using learned transformations based on a cumulative distributions functions (CDF). Following this idea, the autoregressive scheme of Eq. (14) becomes

$$f_t(\mathbf{x}) = g_t (f_{t-1}^*(h_{CDF}(\mathbf{x})), h_{CDF}(\mathbf{x})), \tag{18}$$

where $h_{CDF}(\mathbf{x})$ is a bijective warping function. The MFDGP then operates on the transformed inputs $h_{CDF}(\mathbf{x})$ as sketched in Fig. 3. [49] use the family of Beta cumulative distribution functions as warping function. We follow

the implementation of the BoTorch library [53] and use the class of Kumaraswamy CDF [73,74] as input warping function. The hyperparameters of the Kumaraswamy CDF are jointly optimized with the hyperparameters of the MFDGP. Our implementation of the MFDGP including the input-space warping is realized using GPyTorch [52] and BoTorch [53].

3.3. Data generation

We base the creation and validation of our stochastic surrogate model of the tip velocity on a design of experiment with $N_{LF} = 1000$ low-fidelity, $N_{IF} = 400$ intermediate-fidelity, and $N_{HF} = 400$ high-fidelity samples. To analyze the correlation between the different fidelities, we use 150 of the high-, intermediate-, and low-fidelity data for the same parameter sets. They are obtained from a quasi-Monte Carlo (qMC) sampler based on Sobol' sequences [75]. The remaining samples for each fidelity level are also generated using a qMC sampler based on Sobol' sequences [75].

With the low-fidelity model as reference, we increase the effective resolution for the intermediate- and high-fidelity models by adding levels to the MR tree. We employ a dyadic refinement strategy, thus halving the cell size with each additional refinement level. For the two-dimensional domain of this case, this results in up to four-times larger meshes with each additional resolution level. Assuming that the diffusive time scale determines the smallest permissive timestep size, cf. Eq. (10), halving the smallest cell size results in four-times smaller timestep sizes. Thus, without spatial adaptivity, adding one level to the MR tree increases the simulation cost by a factor of 16. Compared to the low-fidelity model, we add $\Delta l_{IF} = 2$ levels to the MR tree for the intermediate-fidelity model. Thus, without spatial adaptivity, the intermediate-fidelity model is $16^{\Delta l_{IF}} = 256$ times more expensive than the low-fidelity model. For the high-fidelity model, we add $\Delta l_{HF} = 3$ levels to the MR tree compared to the low-fidelity model. Thus, without spatial adaptivity, the high-fidelity model is $16^{\Delta l_{HF}} = 4096$ times more expensive than the low-fidelity model.

However, spatial and temporal adaptivity of the numerical model result in a reduced computing-time increase. Measurements of the computing time for the 150 samples available at each fidelity level indicate that the intermediate-fidelity simulations are by a factor of 27 more expensive than the low-fidelity simulations, and the high-fidelity simulations by a factor of 224.

In the left column of Fig. 4, we use the 150 samples available at each fidelity level to depict the correlation between the fidelity levels. It shows, that the correlation is highly nonlinear and that the low-fidelity model underestimates the true tip velocity. [31] investigates a physical system with similar properties. A low-fidelity model is employed, giving fairly accurate estimates of the high-fidelity model for a certain region of the input space. Outside of this region, the low-fidelity estimates are significantly more inaccurate. Multi-fidelity GPs are applied, based on the linear model presented in Eq. (12). It is observed that prediction quality decreases in regions where the low-fidelity model is inaccurate. [32] investigates the same system employing GPs based on the nonlinear autoregressive scheme presented in Eq. (14) and reports improved predictive accuracy compared to [31]. This motivates using MFDGPs, which also are based on the nonlinear autoregressive scheme, to investigate the tip-velocity data.

In Fig. 4(a), the darker color of points with high tip velocity indicates that the tip velocity strongly increases with \mathcal{T} . This is confirmed by Fig. 4(b), where the high-fidelity tip velocity u_{Γ}^{HF} is shown for varying \mathcal{T} , and an exponential-like dependence of u_{Γ} on \mathcal{T} is implied. When analyzing the colors in Figs. 4(c) and 4(e), no clear trend for the dependence of the tip velocity on ε and α_m/α_s can be deduced for the chosen parameter space. This is confirmed by Figs. 4(d) and 4(f), where u_{Γ}^{HF} is shown for varying ε and $\frac{\alpha_m}{\alpha_s}$, respectively. A comparison between the plots suggests that u_{Γ} is most sensitive to \mathcal{T} , while ε and α_m/α_s influence u_{Γ} only moderately. The same can be concluded when analyzing the correlation between the low- and intermediate-fidelity level.

Fig. 5 shows the shape of the dendrite during steady growth for different parameters for the three different fidelity levels. From top to bottom, \mathcal{T} , and thus u_{Γ} , increases. The tip is more pronounced for higher \mathcal{T} and depicts more instabilities. For the highest undercooling (bottom row), the dendrite develops secondary branches for the intermediate- and high-fidelity simulations, cf. middle and right picture in the bottom row. This is suppressed for the lowest fidelity due to the low resolution, cf. left picture in the bottom row. For all parameter sets, the tip is more pronounced for increasing fidelity. As already visible in the correlation plots, the low-, intermediate- and high-fidelity model exhibit stronger differences for high \mathcal{T} , and thus high u_{Γ} . This is caused by the fact, that increasing tip velocities require resolving successively smaller tip radii.

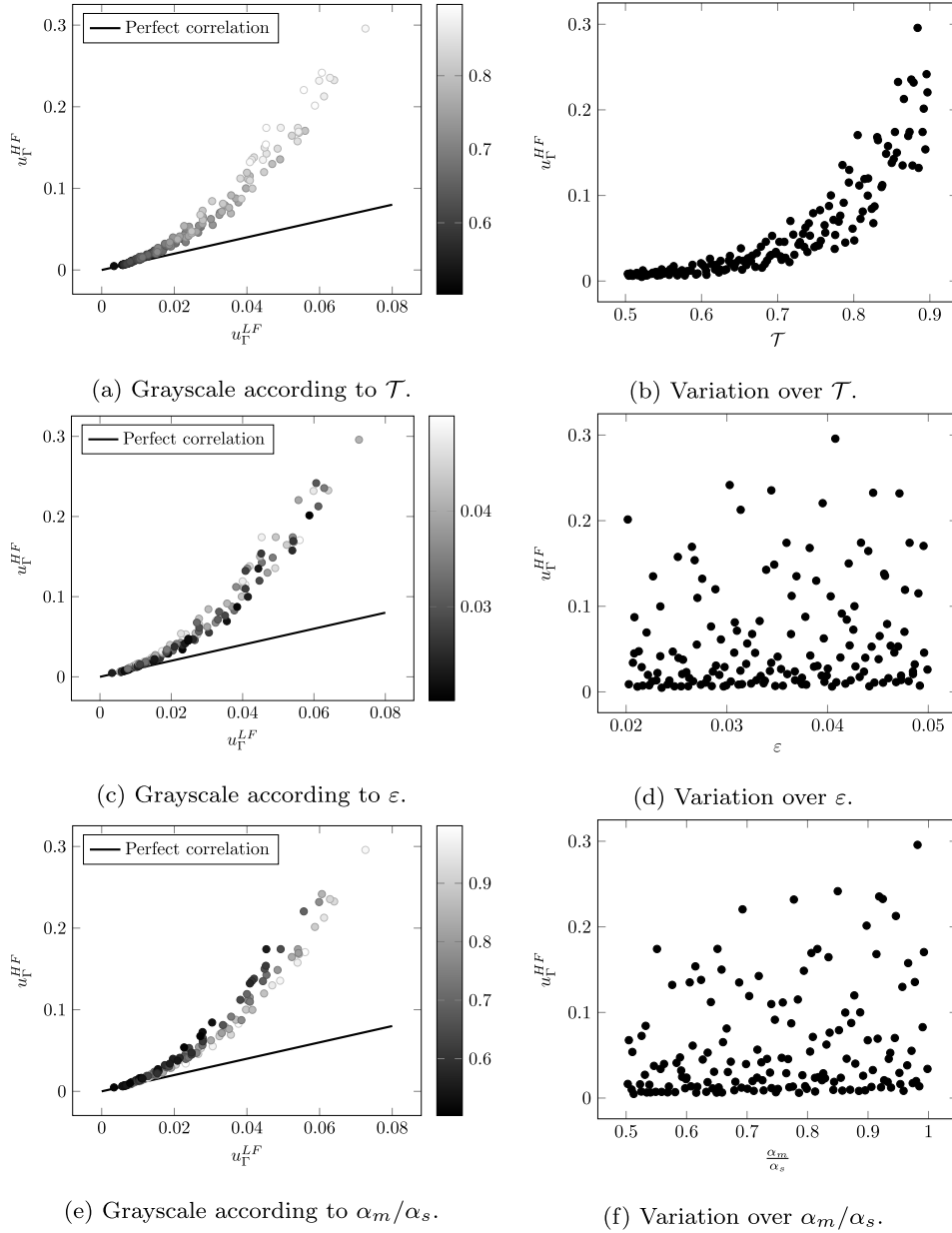


Fig. 4. Left column: Correlation between u_Γ obtained from the low-fidelity and high-fidelity model, which are denoted as u_Γ^{LF} and u_Γ^{HF} , respectively. From top to bottom, the grayscale of the points is according to \mathcal{T} , ε , and α_m/α_s . Right column: u_Γ^{HF} plotted over a single parameter. From top to bottom, \mathcal{T} , ε , and α_m/α_s are varied.

4. Model validation

We assess the quality of our MFDGP model by cross validation. We split the low-, intermediate- and high-fidelity data sets into a training and validation set. The training sets at the fidelity levels consist of $N_{LF,t}$, $N_{IF,t}$ and $N_{HF,t}$ samples that are randomly chosen from the overall data. Their combination is used to train the MFDGP model. The validation sets at each fidelity level consist of the remaining $N_{LF,v} = N_{LF} - N_{LF,t}$, $N_{IF,v} = N_{IF} - N_{IF,t}$ and $N_{HF,v} = N_{HF} - N_{HF,t}$ samples, and are used to cross validate the model. Following [31], we define the L_2 error of a trained model as

$$\epsilon_{cv} = \frac{1}{N_{HF,v}} \sum_{i=1}^{N_{HF,v}} \left(\frac{u_{\Gamma,i}^{HF} - u_{\Gamma,i}^{GP}}{u_{\Gamma,i}^{HF}} \right)^2, \quad (19)$$

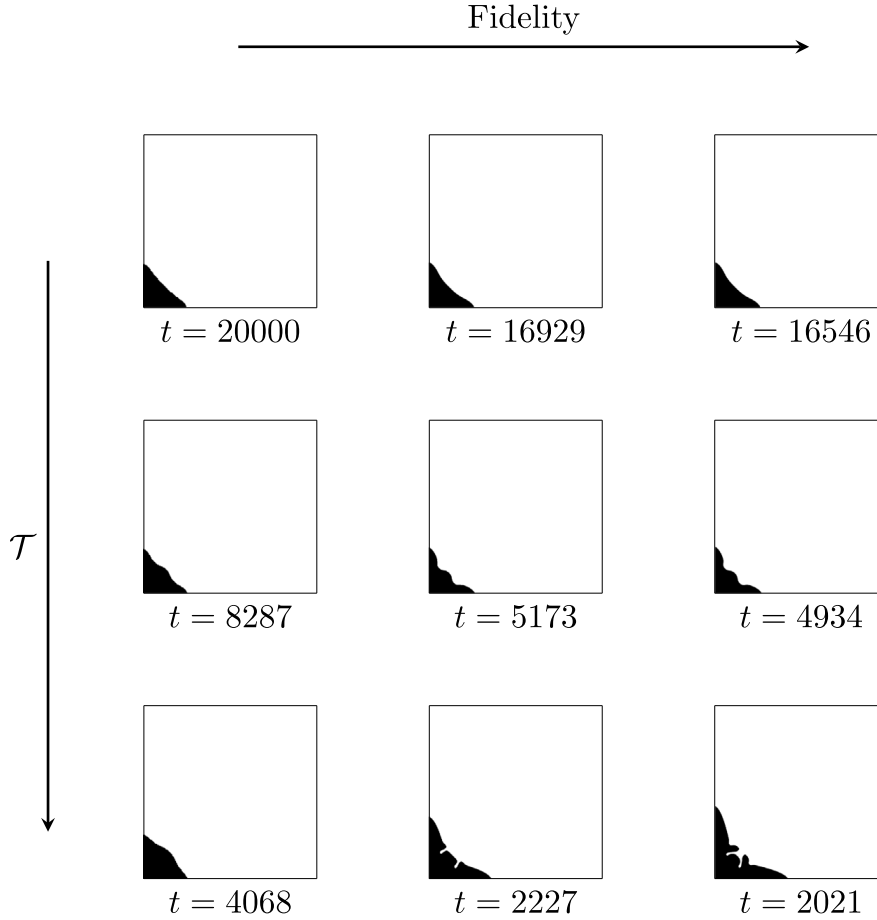


Fig. 5. Shape of the dendrite when a steady-state tip velocity has developed. The low-, middle-, and high-fidelity results are shown in the left, middle, and right column respectively. The first, second and third rows show results for $\mathbf{x} = [0.523, 0.0481, 0.921]$, $\mathbf{x} = [0.721, 0.0422, 0.804]$, and $\mathbf{x} = [0.897, 0.0395, 0.693]$ respectively. The respective time is indicated for each plot.

where $u_{\Gamma,i}^{HF}$ denotes the i th high-fidelity measurement of u_{Γ} from the validation set, and $u_{\Gamma,i}^{GP}$ denotes the corresponding high-fidelity prediction from the MFDGP. Compared to [31], we normalize each summand with the true tip velocity from the validation set. For comparison, we also train single-fidelity GP regression models solely based on the high-fidelity points in the training set. In that case, $u_{\Gamma,i}^{GP}$ denotes the prediction of the single-fidelity GP. To account for the randomness in choosing a training set with $N_{LF,t} < N_{LF}$, $N_{IF,t} < N_{IF}$ and $N_{HF,t} < N_{HF}$ points, we create an ensemble of 20 different random realizations of the training set and train the corresponding GP models. We calculate the L_2 error for each model following Eq. (19) and denote their mean and standard deviation as $\overline{\epsilon_{cv}}$ and $\sigma_{\epsilon_{cv}}$, respectively. Note that $\sigma_{\epsilon_{cv}}$ measures the sensitivity of the trained model to the random selection of the training points.

In Fig. 6, we vary $N_{HF,t}$ and show the cross-validation error $\overline{\epsilon_{cv}}$ for a single-fidelity GP (Fig. 6(a)) and a two-fidelity MFDGP with $N_{LF,t} = 250$ (Fig. 6(b)). We train both models either with or without input warping. The dataset in Section 3.3 implies, that the tip velocity’s dependence on \mathcal{T} is nonlinear and potentially non-stationary. Thus, we apply input warping for \mathcal{T} . The quality of the single-fidelity models with and without input warping is approximately similar, see Fig. 6(a). The noticeable higher error for the model with input warping for $N_{HF,t} = 5$ is due to overfitting caused by the additional hyperparameters of the Kumaraswamy CDF. The quality of the two-fidelity model improves when input warping is applied, see Fig. 6(b). Thus, we focus on models with input warping in the following.

In Figs. 7(a) and 7(c) we vary $N_{HF,t}$ and show $\overline{\epsilon_{cv}}$ and $\sigma_{\epsilon_{cv}}$ for a single-fidelity GP, and two-fidelity MFDGPs with $N_{LF,t} = 10, 20$ and 100 . For all models, $\overline{\epsilon_{cv}}$ decreases with increasing $N_{HF,t}$. For $N_{LF,t} = 10$ and 20 , the

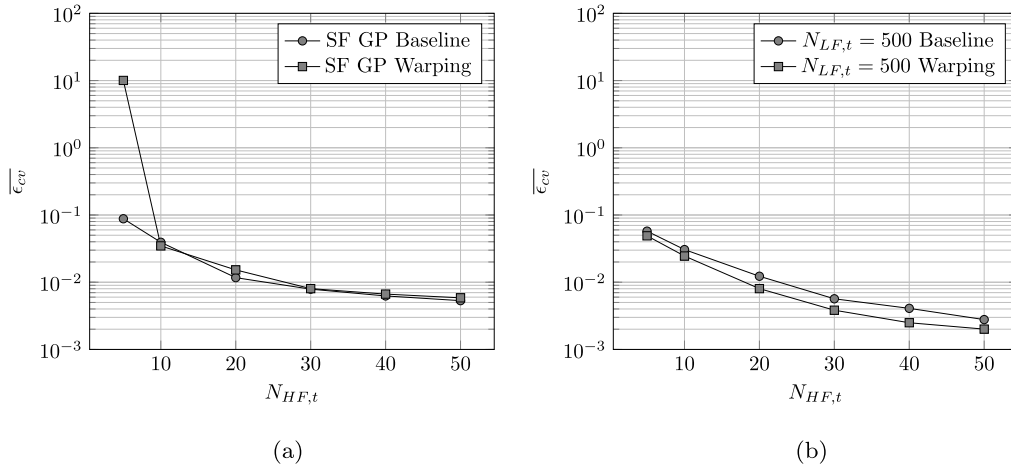


Fig. 6. (a): Mean cross-validation error $\overline{\epsilon}_{cv}$ for single-fidelity GP regression with and without input-space warping, which are denoted with “Baseline” and “Warping”, respectively. (b): Mean cross-validation error $\overline{\epsilon}_{cv}$ for two-fidelity MFDGP regression with and without input-space warping for $N_{LF,t} = 500$.

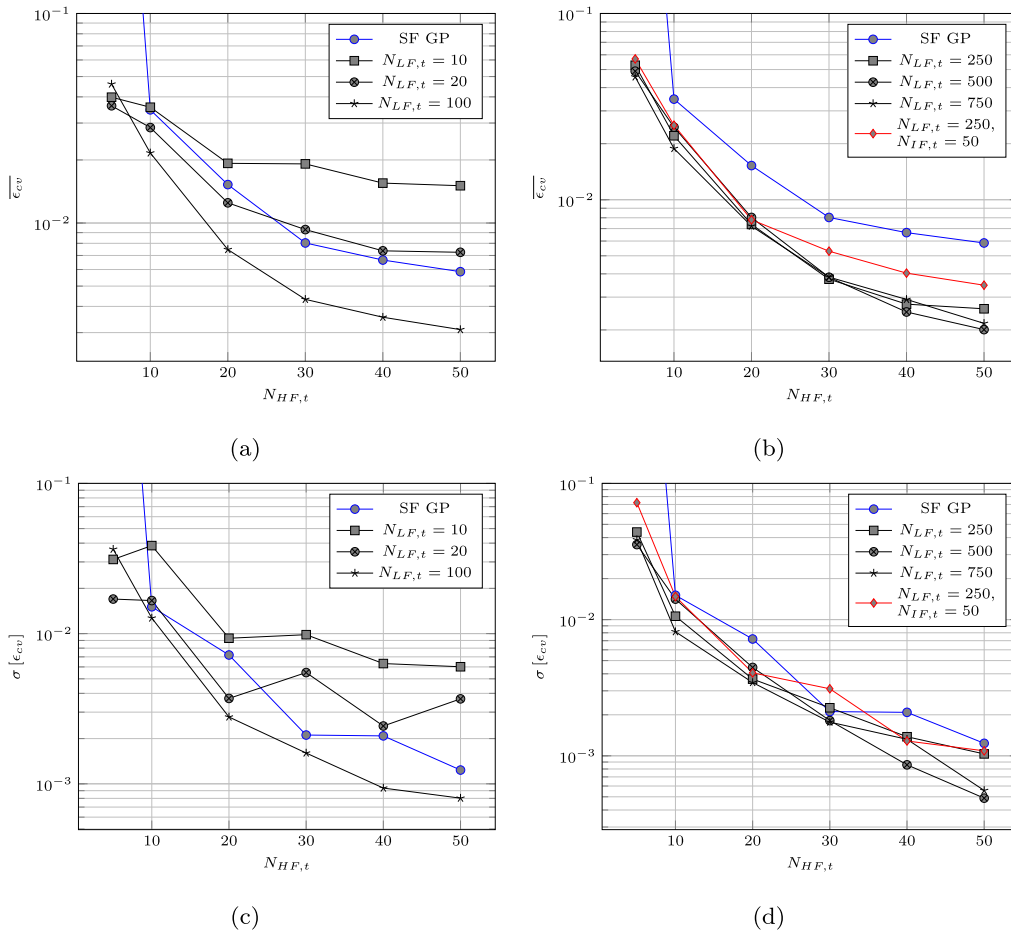


Fig. 7. (a): Mean cross-validation error $\overline{\epsilon}_{cv}$ of two-fidelity MFDGPs and a standard single-fidelity GP for varying $N_{HF,t}$. The two-fidelity MFDGPs are trained with $N_{LF,t} = 10, 20$ and 100 low-fidelity points in the training set respectively. (b): Mean cross-validation error $\overline{\epsilon}_{cv}$ of two- and three-fidelity MFDGPs and a standard single-fidelity GP for varying $N_{HF,t}$. The two-fidelity MFDGPs are trained with $N_{LF,t} = 250, 500$ and 750 low-fidelity points in the training set respectively. The three-fidelity MFDGP is trained with $N_{LF,t} = 250$ and $N_{IF,t} = 50$. (c) and (d): Standard deviation $\sigma_{\overline{\epsilon}_{cv}}$ of the cross-validation error for the models in (a) and (b), respectively.

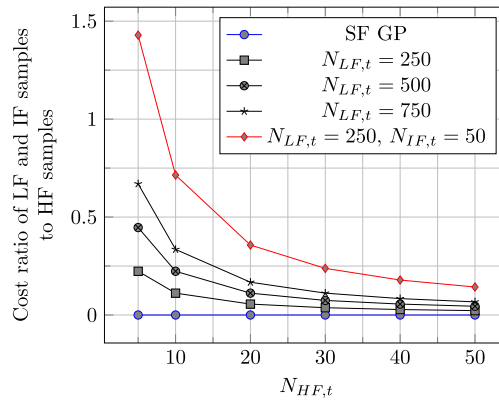


Fig. 8. Approximate computational cost ratio of the low- and intermediate-fidelity samples to the high-fidelity samples of the used training dataset for the models in Fig. 7(b).

multi-fidelity models show similar or better $\overline{\epsilon_{cv}}$ than the single-fidelity model where $N_{HF,t} \leq N_{LF,t}$. When further increasing $N_{HF,t}$, the single-fidelity model outperforms the multi-fidelity models, as the number of low-fidelity samples used to train the low-fidelity part of the MFDGP model is insufficient to build an accurate base for the high-fidelity part of the MFDGP model. For $N_{LF,t} = 100$, multi-fidelity modeling outperforms single-fidelity modeling for the investigated range of $N_{HF,t}$. The sensitivity of the models on the selection of the training points exhibits the same trends as $\overline{\epsilon_{cv}}$, see Fig. 7(c).

In Figs. 7(b) and 7(d) we investigate the quality of multi-fidelity models when further increasing the amount of low-fidelity training data. Therefore, we vary $N_{HF,t}$ and show $\overline{\epsilon_{cv}}$ and $\sigma_{\epsilon_{cv}}$ for a single-fidelity GP, two-fidelity MFDGPs with $N_{LF,t} = 250, 500$ and 750 , and a three-fidelity MFDGP with $N_{LF,t} = 250$ and $N_{IF,t} = 50$. The mean of the cross-validation error and its standard deviation decrease rapidly up to $N_{HF,t} = 30$. For $N_{HF,t} > 30$, a constant rate of improvement of $\overline{\epsilon_{cv}}$ adjusts, where the overall error magnitude of the multi-fidelity model is significantly lower than for the single-fidelity model. The sensitivity of the models on the selection of the training points is also better for the multi-fidelity case, see Fig. 7(d). Figs. 7(b) and 7(d) show, that the quality of the two-fidelity model is approximately similar for $N_{LF,t} = 250, 500$ and 750 for low $N_{HF,t}$. For high $N_{HF,t}$, the models with more $N_{LF,t}$ tend to outperform the models with less low-fidelity points. Comparing the results for $N_{LF,t} = 500$ and 750 , no significant improvement can be observed. This suggests that the parameter space is sampled sufficiently dense for $N_{LF,t} = 500$ to capture the information content of the low-fidelity model. Considering intermediate-fidelity data does not improve model quality significantly. For high $N_{HF,t}$, the two-fidelity models perform even better than the three-fidelity model. This may be due to the fact that with $N_{IF,t} = 50$, a similar amount of intermediate- and high-fidelity data is present for high $N_{HF,t}$.

Fig. 8 highlights, that the improved quality of the two-fidelity models incurs a negligible computational overhead to generate the training points compared to the single-fidelity model. For models with $N_{LF,t} = 500$ and $N_{HF,t} > 30$, the overhead is less than 10%, yet the obtained models provide significantly more accurate results. For example, the additional cost for two-fidelity modeling with $N_{LF,t} = 500$ at $N_{HF,t} = 30$ is 7.5%. Investing the computational overhead in HF samples would deliver less than three additional HF samples. Adding three HF samples to the SF model reduces $\overline{\epsilon_{cv}}$ by 0.01, whereas the two-fidelity modeling reduces $\overline{\epsilon_{cv}}$ by 0.04. The computational overhead for the three-fidelity model is higher than for the two-fidelity models, but the model quality does not improve. Thus we take the two-fidelity model with $N_{LF,t} = 500$ as reference in the following.

Fig. 9 visualizes the quality of the generated multi-fidelity models by means of correlation plots. We show the correlation between high-fidelity measurements of u_T in the validation set and the predictions from the MFDGP model. MFDGP models with $N_{LF,t} = 500$ and $N_{HF,t} = 5, 10, 30$ and 50 , respectively, are chosen. Figs. 9(a) and 9(b) show strong deviations between the measured and predicted tip velocities. Predictions are more accurate for low than for high tip velocities. High tip velocities are underpredicted by the MFDGP model. For increasing $N_{HF,t}$,

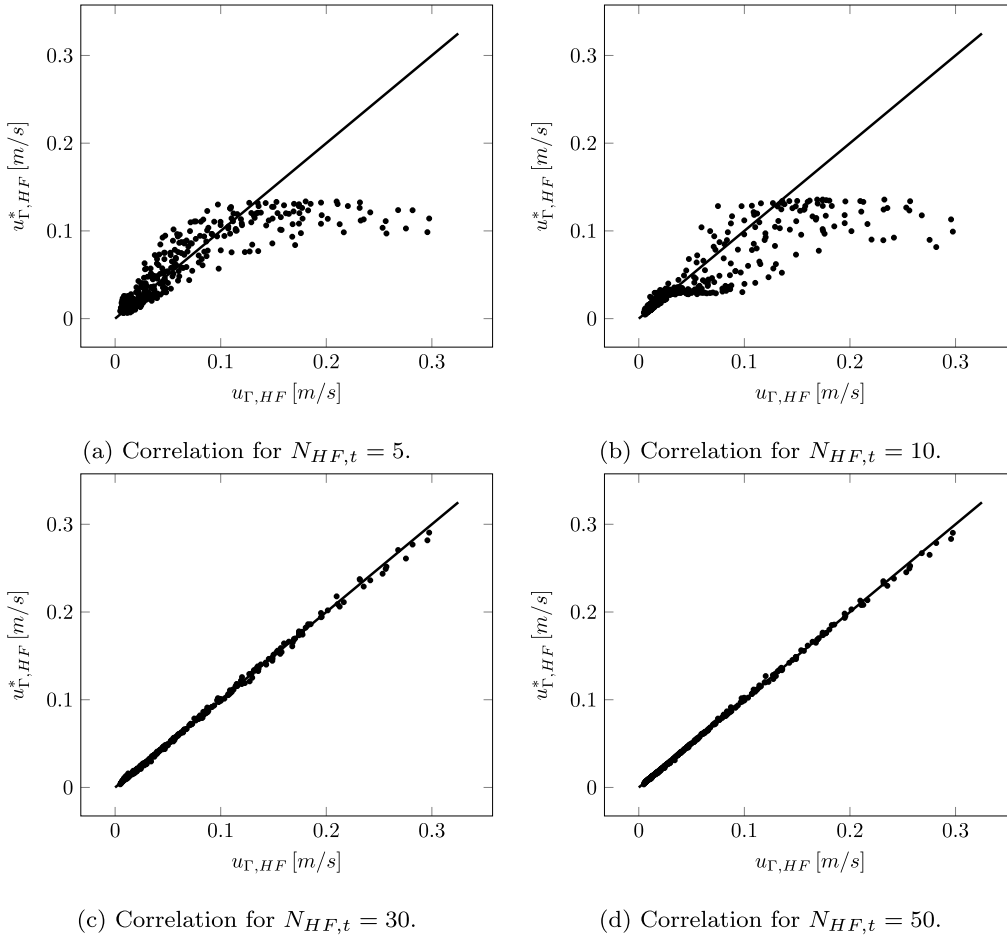


Fig. 9. Correlation between u_{Γ} obtained from high-fidelity simulations $u_{\Gamma, HF}$ and predicted from the MFDGP model $u_{\Gamma, HF}^*$. The MFDGP model was trained with $N_{LF} = 500$. We show plots for $N_{HF,t} = 5, 10, 30$ and 50 from top left to bottom right.

see Figs. 9(c) and 9(d), the model clearly gets more accurate, the points accumulate near the diagonal $y = x$, which indicates perfect correlation, and the variance of the estimate decreases.

5. Multi-fidelity tip-velocity model

We describe the resulting stochastic response surface of the tip velocity by using a two-fidelity MFDGP model with $N_{LF} = 500$ and $N_{HF} = 50$. The cross-validation error of this model is shown in Fig. 9(d). Fig. 10 shows the low- and high-fidelity prediction of u_{Γ} for varying \mathcal{T} . The remaining parameters are fixed as $\frac{\alpha_m}{\alpha_s} = 1.0$ and $\varepsilon = 0.05$, respectively. To demonstrate model accuracy, we show additionally generated low- and high-fidelity points for $\mathcal{T} = 0.55, 0.65, 0.75$ and 0.85 , which were not used for training. They agree very well with the posterior mean of the MFDGP. The low-fidelity estimate of u_{Γ} grows slightly with increasing \mathcal{T} , whereas the high-fidelity measurement strongly grows with increasing \mathcal{T} , indicating that the low-fidelity model is more accurate for low u_{Γ} and underpredicts the nearly exponential growth of u_{Γ} for increasing \mathcal{T} . As seen in Section 4, adding low-fidelity samples further improves model quality.

Figs. 11(a)–11(c) depict the characteristics of the tip-velocity response surface. The plots indicate that u_{Γ} grows with increasing \mathcal{T} , $\frac{\alpha_m}{\alpha_s}$, and ε . Comparing the figures confirms that u_{Γ} depends on \mathcal{T} exponentially, whereas the dependence of u_{Γ} on $\frac{\alpha_m}{\alpha_s}$ and ε is almost linear, see Figs. 11(b) and 11(c).

We show the combined influence of two process parameters on u_{Γ} in Fig. 12. The bilinear influence of $\frac{\alpha_m}{\alpha_s}$ and ε on u_{Γ} , as already indicated by preceding, is visible in Fig. 12(a). When fixing $\frac{\alpha_m}{\alpha_s}$ or ε , see Figs. 12(b) and 12(c), the influence of \mathcal{T} dominates and the exponential increase is clearly visible.

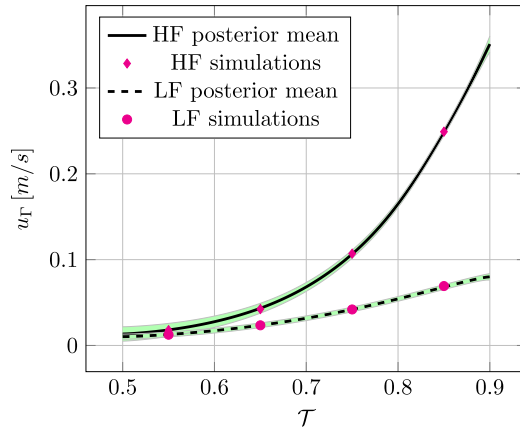
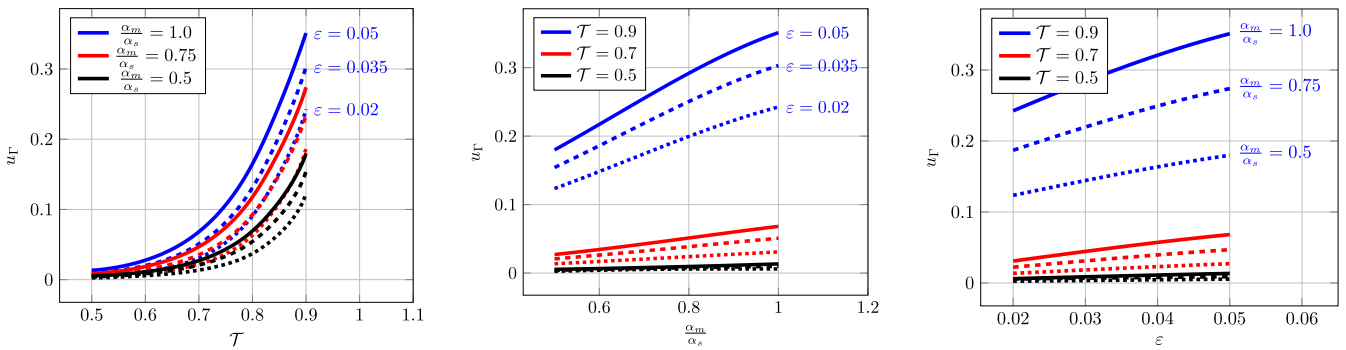


Fig. 10. High- and low-fidelity prediction of the MFDGP model of u_Γ for varying \mathcal{T} and fixed $\frac{\alpha_m}{\alpha_s} = 1.0$ and $\varepsilon = 0.05$. For comparison, low- and high-fidelity measurements are plotted for several \mathcal{T} . The predictive mean of the low- and high-fidelity prediction, $\overline{u}_{\Gamma,LF}^*$ and $\overline{u}_{\Gamma,HF}^*$, are shown by the dashed and solid line, respectively. Their respective uncertainty band of $\pm 2\sigma [u_{\Gamma,LF}^*]$ and $\pm 2\sigma [u_{\Gamma,HF}^*]$ is indicated by the green region.



(a) Posterior mean of the high-fidelity prediction of u_Γ for varying \mathcal{T} and different $\frac{\alpha_m}{\alpha_s}$. The dotted, dashed and solid lines indicate fixed $\varepsilon = 0.02$, $\varepsilon = 0.035$, and $\varepsilon = 0.05$, respectively.

(b) Posterior mean of the high-fidelity prediction of u_Γ for varying $\frac{\alpha_m}{\alpha_s}$ and different \mathcal{T} . The dotted, dashed and solid lines indicate fixed $\varepsilon = 0.02$, $\varepsilon = 0.035$, and $\varepsilon = 0.05$, respectively.

(c) Posterior mean of the high-fidelity prediction of u_Γ for varying ε and different \mathcal{T} . The dotted, dashed and solid lines indicate fixed $\frac{\alpha_m}{\alpha_s} = 0.5$, $\frac{\alpha_m}{\alpha_s} = 0.75$, and $\frac{\alpha_m}{\alpha_s} = 1.0$, respectively.

Fig. 11.

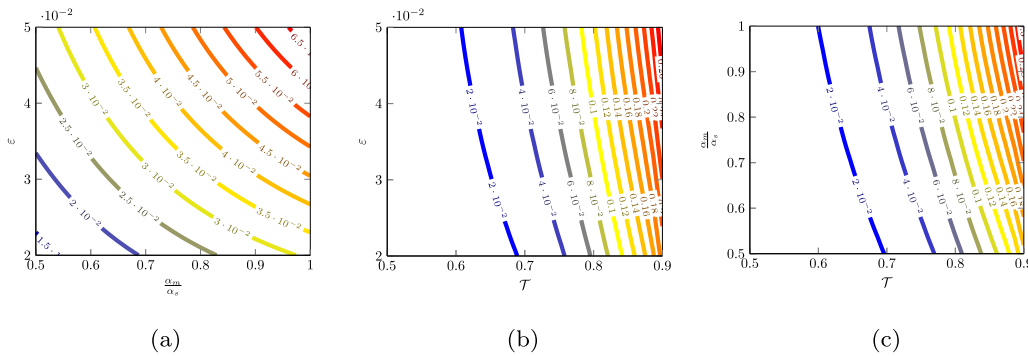


Fig. 12. (a): Contour of the posterior mean of the high-fidelity prediction of u_Γ for varying $\frac{\alpha_m}{\alpha_s}$ and ε and fixed $\mathcal{T} = 0.7$. (b): Contour of the posterior mean of the high-fidelity prediction of u_Γ for varying \mathcal{T} and ε and fixed $\frac{\alpha_m}{\alpha_s} = 0.75$. (c): Contour of the posterior mean of the high-fidelity prediction of u_Γ for varying \mathcal{T} and $\frac{\alpha_m}{\alpha_s}$ and fixed $\varepsilon = 0.035$.

6. Summary and conclusions

We have developed a stochastic surrogate model for the tip velocity of a single dendrite. The model is based on a novel machine-learning model combining input warping techniques and multi-fidelity deep Gaussian processes. We show, that the hierarchy of resolution fidelities allows for cost-efficiently generating surrogate models that are significantly more accurate compared to single-fidelity models, especially when access to high-fidelity data is limited. The resulting model accurately captures the system of interest. Its stochastic nature allows for quantifying the uncertainty associated with the predictions, and motivates its usage in Bayesian-optimization algorithms for inverse problems. Also, it may serve as input for microstructure simulations which rely on accurate relations between local solidification velocities and process parameters to predict grain-scale crystalline structures and which need material-dependent model calibration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first, third, and fifth authors acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 667483). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer SuperMUC-NG at Leibniz Supercomputing Centre (www.lrz.de).

Appendix. Note on grid effects of the numerical model

The two-dimensional dendritic growth investigated in this work features four tips growing along the main axes. The numerical model used to solve the problem is based on a structured mesh that is oriented along the main axes. When simulating similar scenarios with tips not growing along the main axes, grid effects of the underlying numerical scheme may affect the observed steady-state tip velocities.

To investigate the consistency of the numerical model for such scenarios we simulate the case presented in Section 3.1 with capillary anisotropy formulation

$$d^*(\theta) = d_0(1 - 15\epsilon\cos(4\theta + \pi/4)). \quad (\text{A.1})$$

This is equivalent to rotating the crystal anisotropy with regard to the grid coordinate system by 45° . Again, we apply $\Delta T = 0.65$, $\epsilon = 0.05$, and $\alpha_m = 1.0$. The level-dependent spatial resolution with level-dependent timestep sizes results in significant accuracy improvements for the predicted tip velocity using increasingly finer meshes. This is shown for a single crystal in Fig. A.13. Independently of the chosen resolution, the initial condition with temperature jump at the interface results in an overprediction of the growth velocity. Once the temperature boundary layer is fully developed, the tip velocity reaches a steady state. Fig. A.13(b) shows the relative error $\epsilon_{u_\Gamma} = 1 - |\frac{u_\Gamma}{u_{\Gamma,ref}}|$ of the simulated steady-state tip velocity for different resolutions. Effects of grid anisotropy result in larger errors than for the original case. This indicates that with sufficiently fine grid resolutions, errors of the same order of magnitude can be reached as for the original case.

To further investigate the consistency of the numerical method, we simulate a six-fold symmetric crystal and apply $\Delta T = 0.65$, $\epsilon = 0.05$, and $\alpha_m = 1.0$. We modify the capillary anisotropy model of the four-fold symmetric crystal presented in Eq. (1), and use

$$d^{**}(\theta) = d_0(1 - 15\epsilon\cos(6\theta)). \quad (\text{A.2})$$

We choose a domain $(x, y) \in [0, 3200] \times [0, 3200]$, and locate the seed with initial radius $r_0 = 50$ at the center of the square. Fig. A.14 shows the shape of the crystal for different resolutions. For resolutions below $128 \frac{\text{cells}}{r_0}$, the tips growing along the main axes spread further from the center of the crystal than the remaining tips. For a resolution of $128 \frac{\text{cells}}{r_0}$, all six tips spread equally from the center of the crystal indicating diminishing grid effects.

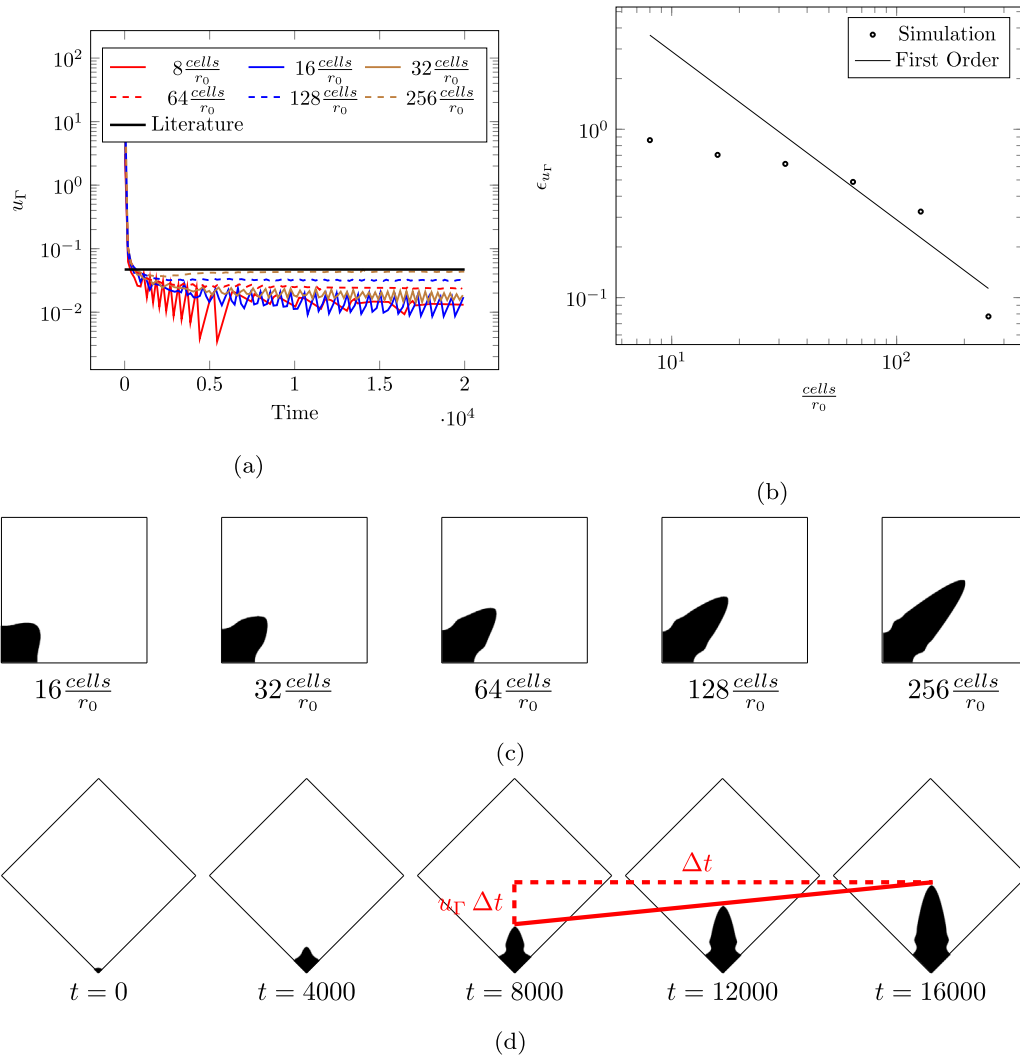


Fig. A.13. (a): Velocity convergence for a 45° rotated single four-fold symmetric crystal for successive grid refinement. The simulation is conducted with $\mathcal{T} = 0.65$, $\varepsilon = 0.05$, and $\frac{\alpha_m}{\alpha_s} = 1.0$. (b): Accuracy order analysis of the converged tip velocity. (c): Shape of the crystal at $t = 14000$ for the simulated resolutions. (d): Temporal evolution of the crystal for a resolution of $256 \frac{cells}{r_0}$.

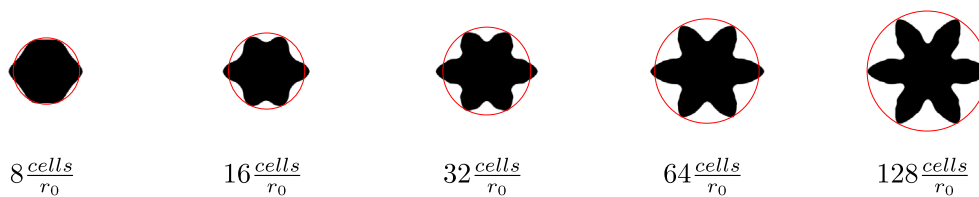


Fig. A.14. Velocity convergence for a six-fold symmetric crystal for successive grid refinement. The simulation was conducted with $\mathcal{T} = 0.65$, $\varepsilon = 0.05$, and $\frac{\alpha_m}{\alpha_s} = 1.0$. The shape of the crystal at $t = 20000$ is shown in black for different resolutions. A circle envelope of the four tips not growing along the main axes is shown in red for each crystal.

References

- [1] W. Kurz, D.J. Fisher, R. Trivedi, Progress in modelling solidification microstructures in metals and alloys: dendrites and cells from 1700 to 2000, *Int. Mater. Rev.* 64 (6) (2019) 311–354.
- [2] W. Kurz, M. Rappaz, R. Trivedi, Progress in modelling solidification microstructures in metals and alloys. Part II: dendrites from 2001 to 2018, *Int. Mater. Rev.* 66 (1) (2021) 30–76, <http://dx.doi.org/10.1080/09506608.2020.1757894>.

- [3] B.G. Thomas, Review on modeling and simulation of continuous casting, *Steel Res. Int.* 1 (89) (2018) 1700312, <http://dx.doi.org/10.1002/srin.201700312>.
- [4] J. Strickland, B. Nenchev, H. Dong, On directional dendritic growth and primary spacing—a review, *Crystals* 10 (7) (2020) 1–30, <http://dx.doi.org/10.3390/cryst10070627>.
- [5] G. Ivantsov, *Dokl. Akad. Nauk SSSR* (1947) 558–567.
- [6] W. Oldfield, Computer model studies of dendritic growth, *Mater. Sci. Eng.* 11 (1973) 211–218.
- [7] J.S. Langer, J. Müller-Krumbhaar, Stability effects in dendritic crystal growth, *J. Cryst. Growth* 42 (C) (1977) 11–14, [http://dx.doi.org/10.1016/0022-0248\(77\)90171-3](http://dx.doi.org/10.1016/0022-0248(77)90171-3).
- [8] R. Trivedi, P. Mazumder, S.N. Tewari, The effect of convection on disorder in primary cellular and dendritic arrays, *Metall. Mater. Trans. A* 33 (12) (2002) 3763–3775.
- [9] M.B. Amar, Y. Pomeau, Theory of dendritic growth in a weakly undercooled melt, *Europhys. Lett.* 2 (4) (1986) 307.
- [10] D. Bensimon, P. Pelcé, B.I. Shraiman, Dynamics of curved fronts and pattern selection, *J. Physique* 48 (12) (1987) 2081–2087.
- [11] E. Ben-Jacob, N. Goldenfeld, J. Langer, G. Schön, Dynamics of interfacial pattern formation, *Phys. Rev. Lett.* 51 (21) (1983) 1930.
- [12] R.C. Brower, D.A. Kessler, J. Koplik, H. Levine, Geometrical approach to moving-interface dynamics, *Phys. Rev. Lett.* 51 (13) (1983) 1111.
- [13] D.I. Meiron, Selection of steady states in the two-dimensional symmetric model of dendritic growth, *Phys. Rev. A* 33 (4) (1986) 2704.
- [14] A. Barbieri, D.C. Hong, J. Langer, Velocity selection in the symmetric model of dendritic crystal growth, *Phys. Rev. A* 35 (4) (1987) 1802.
- [15] P. Pelce, Y. Pomeau, Dendrites in the small undercooling limit, in: *Dynamics of Curved Fronts*, Elsevier, 1988, pp. 327–340.
- [16] D.A. Kessler, H. Levine, Stability of dendritic crystals, *Phys. Rev. Lett.* 57 (24) (1986) 3069.
- [17] M. Markl, C. Körner, Multiscale modeling of powder bed-based additive manufacturing, *Annu. Rev. Mater. Res.* 46 (May) (2016) 93–123, <http://dx.doi.org/10.1146/annurev-matsci-070115-032158>.
- [18] C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge Massachusetts, USA, 2006.
- [19] A. Forrester, A. Sobester, A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, 2008.
- [20] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, vol. 2, Springer, New York, NY, 2009, <http://dx.doi.org/10.1007/978-0-387-84858-7>.
- [21] S.H. Bryngelson, A. Charalampopoulos, T.P. Sapsis, T. Colonius, A Gaussian moment method and its augmentation via LSTM recurrent neural networks for the statistics of cavitating bubble populations, *Int. J. Multiph. Flow.* 127 (2020) 103262.
- [22] H. Zhang, J.P. Choi, S.K. Moon, T.H. Ngo, A knowledge transfer framework to support rapid process modeling in aerosol jet printing, *Adv. Eng. Inf.* 48 (2021) 101264.
- [23] B. Rankouhi, S. Jahani, F. Pfefferkorn, D. Thoma, Compositional grading of a 316l-cu multi-material part using machine learning for the determination of selective laser melting process parameters, *Addit. Manuf.* 38 (2021) <http://dx.doi.org/10.1016/j.addma.2021.101836>.
- [24] H.A. Kumar, S. Kumaraguru, C. Paul, K. Bindra, Faster temperature prediction in the powder bed fusion process through the development of a surrogate model, *Opt. Laser Technol.* 141 (2021) 107122.
- [25] R. Saunders, C. Butler, J. Michopoulos, D. Lagoudas, A. Elwany, A. Bagchi, Mechanical behavior predictions of additively manufactured microstructures using functional Gaussian process surrogates, *Npj Comput. Mater.* 7 (1) (2021) 1–11.
- [26] C. Hu, W. Hau, W. Chen, Q.-H. Qin, The fabrication of long carbon fiber reinforced polylactic acid composites via fused deposition modelling: Experimental analysis and machine learning, *J. Compos. Mater.* 55 (11) (2021) 1459–1472, <http://dx.doi.org/10.1177/0021998320972172>.
- [27] E. Mendoza Jimenez, D. Ding, L. Su, A. Joshi, A. Singh, B. Reeja-Jayan, J. Beuth, Parametric analysis to quantify process input influence on the printed densities of binder jetted alumina ceramics, *Addit. Manuf.* 30 (2019) <http://dx.doi.org/10.1016/j.addma.2019.100864>.
- [28] Y. Yabansu, V. Rehn, J. Hötzer, B. Nestler, S. Kalidindi, Application of Gaussian process autoregressive models for capturing the time evolution of microstructure statistics from phase-field simulations for sintering of polycrystalline ceramics, *Modelling Simulation Mater. Sci. Eng.* 27 (8) (2019) <http://dx.doi.org/10.1088/1361-651X/ab413e>.
- [29] T. Simpson, V. Toropov, V. Balabanov, F. Viana, Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come-or not, in: *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2008, p. 5802.
- [30] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, *Siam Rev.* 60 (3) (2018) 550–591.
- [31] H. Babae, P. Perdikaris, C. Chrysostomidis, G. Karniadakis, Multi-fidelity modelling of mixed convection based on experimental correlations and numerical simulations, *J. Fluid Mech.* 809 (2016) 895.
- [32] P. Perdikaris, M. Raissi, A. Damianou, N.D. Lawrence, G.E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, *Proc. R. Soc. A* 473 (2198) (2017) 20160751.
- [33] L. Meng, J. Zhang, Process design of laser powder bed fusion of stainless steel using a Gaussian process-based machine learning model, *JOM* 72 (1) (2020) 420–428.
- [34] G. Kissas, Y. Yang, E. Hwuang, W.R. Witschey, J.A. Detre, P. Perdikaris, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks, *Comput. Methods Appl. Mech. Engrg.* 358 (2020) 112623.
- [35] C.M. Fleeter, G. Geraci, D.E. Schiavazzi, A.M. Kahn, A.L. Marsden, Multilevel and multifidelity uncertainty quantification for cardiovascular hemodynamics, *Comput. Methods Appl. Mech. Engrg.* 365 (2020) 113030, <http://dx.doi.org/10.1016/j.cma.2020.113030>.
- [36] X. Meng, Z. Wang, D. Fan, M. Triantafyllou, G.E. Karniadakis, A fast multi-fidelity method with uncertainty quantification for complex data correlations: Application to vortex-induced vibrations of marine risers, 2020, [arXiv:2012.13481](https://arxiv.org/abs/2012.13481).

- [37] A. Doostan, G. Geraci, G. Iaccarino, A bi-fidelity approach for uncertainty quantification of heat transfer in a rectangular ribbed channel, in: *Turbo Expo: Power for Land, Sea, and Air*, vol. 49712, American Society of Mechanical Engineers, 2016, V02CT45A031.
- [38] I.-G. Farcaş, P.C. Sârbu, H.-J. Bungartz, T. Neckel, B. Uekermann, Multilevel adaptive stochastic collocation with dimensionality reduction, in: J. Garcke, D. Pflüger, C.G. Webster, G. Zhang (Eds.), *Sparse Grids and Applications - Miami 2016*, Springer International Publishing, Cham, 2018, pp. 43–68.
- [39] A. Beck, J. Dürrwächter, T. Kuhn, F. Meyer, C.-D. Munz, C. Rohde, Hp-multilevel Monte Carlo methods for uncertainty quantification of compressible Navier–Stokes equations, *SIAM J. Sci. Comput.* 42 (4) (2020) B1067–B1091.
- [40] I.-G. Farcaş, J. Latz, E. Ullmann, T. Neckel, H.-J. Bungartz, Multilevel adaptive sparse leja approximations for Bayesian inverse problems, *SIAM J. Sci. Comput.* 42 (1) (2020) A424–A451.
- [41] M.C. Kennedy, A. O’Hagan, Predicting the output from a complex computer code when fast approximations are available, *Biometrika* 87 (1) (2000) 1–13.
- [42] P. Perdikaris, G.E. Karniadakis, Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond, *J. R. Soc. Interface* 13 (118) (2016) 20151107.
- [43] P. Perdikaris, D. Venturi, G.E. Karniadakis, Multifidelity information fusion algorithms for high-dimensional systems and massive data sets, *SIAM J. Sci. Comput.* 38 (4) (2016) B521–B538.
- [44] F.S. Costabal, P. Perdikaris, E. Kuhl, D.E. Hurtado, Multi-fidelity classification using Gaussian processes: accelerating the prediction of large-scale computational models, *Comput. Methods Appl. Mech. Engrg.* 357 (2019) 112602.
- [45] M. Raissi, G. Karniadakis, Deep multi-fidelity Gaussian processes, 2016, arXiv preprint [arXiv:1604.07484](https://arxiv.org/abs/1604.07484).
- [46] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, J. González, Deep Gaussian processes for multi-fidelity modeling, 2019, arXiv:1903.07320.
- [47] A. Damianou, N.D. Lawrence, Deep gaussian processes, in: *Artificial Intelligence and Statistics*, PMLR, 2013, pp. 207–215.
- [48] A. Damianou, Deep Gaussian processes and variational propagation of uncertainty (Ph.D. thesis), University of Sheffield, 2015.
- [49] J. Snoek, K. Swersky, R. Zemel, R. Adams, Input warping for bayesian optimization of non-stationary functions, in: *International Conference on Machine Learning*, PMLR, 2014, pp. 1674–1682.
- [50] V. Perrone, H. Shen, A. Zolic, I. Shcherbatyi, A. Ahmed, T. Bansal, M. Donini, F. Winkelmolen, R. Jenatton, J.B. Faddoul, et al., Amazon SageMaker automatic model tuning: Scalable gradient-free optimization, in: *KDD21*, 2021.
- [51] N. Lally, B. Hartman, Estimating loss reserves using hierarchical Bayesian Gaussian process regression with input warping, *Insurance Math. Econom.* 82 (2018) 124–140.
- [52] J.R. Gardner, G. Pleiss, D. Bindel, K.Q. Weinberger, A.G. Wilson, GPyTorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration, in: *Advances in Neural Information Processing Systems*, 2018.
- [53] M. Balandat, B. Karrer, D.R. Jiang, S. Daulton, B. Letham, A.G. Wilson, E. Bakshy, BoTorch: A framework for efficient Monte-Carlo Bayesian optimization, in: *Advances in Neural Information Processing Systems* 33, 2020, URL [http://arxiv.org/abs/1910.06403](https://arxiv.org/abs/1910.06403).
- [54] J.W.J. Kaiser, N. Hoppe, S. Adami, N.A. Adams, An adaptive local time-stepping scheme for multiresolution simulations of hyperbolic conservation laws, *J. Comput. Phys.* X 4 (2019) 100038.
- [55] N. Hoppe, S. Adami, N.A. Adams, A modular massively parallel computing environment for three-dimensional multiresolution simulations of compressible flows, 2020, [arXiv:2012.04385](https://arxiv.org/abs/2012.04385).
- [56] N. Hoppe, J.M. Winter, S. Adami, N.A. Adams, ALPACA - A level-set based sharp-interface multiresolution solver for conservation laws, *Comput. Phys. Comm.* (2021) 108246, [http://dx.doi.org/10.1016/j.cpc.2021.108246](https://doi.org/10.1016/j.cpc.2021.108246).
- [57] A. Karma, W.-J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, *Phys. Rev. E* 57 (4) (1998) 4323.
- [58] Y.-T. Kim, N. Goldenfeld, J. Dantzig, Computation of dendritic microstructures using a level set method, *Phys. Rev. E* 62 (2) (2000) 2471.
- [59] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, *Phys. Rev. Lett.* 80 (1998) 3308–3311, [http://dx.doi.org/10.1103/PhysRevLett.80.3308](https://doi.org/10.1103/PhysRevLett.80.3308).
- [60] L. Tan, N. Zabaras, A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods, *J. Comput. Phys.* 211 (2006) 36–63, [http://dx.doi.org/10.1016/j.jcp.2005.05.013](https://doi.org/10.1016/j.jcp.2005.05.013).
- [61] H.S. Udaykumar, L. Mao, R. Mittal, A finite-volume sharp interface scheme for dendritic growth simulations: Comparison with microscopic solvability theory, *Numer. Heat Transfer B* 42 (5) (2002) 389–409, [http://dx.doi.org/10.1080/10407790190054003](https://doi.org/10.1080/10407790190054003).
- [62] V. Ramanuj, R. Sankaran, B. Radhakrishnan, A sharp interface model for deterministic simulation of dendrite growth, *Comput. Mater. Sci.* 169 (2019) 109097, [http://dx.doi.org/10.1016/j.commatsci.2019.109097](https://doi.org/10.1016/j.commatsci.2019.109097).
- [63] J. Kaiser, S. Adami, I. Akhatov, N. Adams, A semi-implicit conservative sharp-interface method for liquid-solid phase transition, *Int. J. Heat Mass Transfer* 155 (2020) 119800.
- [64] S. Gottlieb, C.-W. Shu, Total variation diminishing runge-kutta schemes, *Math. Comp.* 67 (221) (1998) 73–85.
- [65] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* 135 (1983) 260–278, [http://dx.doi.org/10.1016/0021-9991\(83\)90136-5](https://doi.org/10.1016/0021-9991(83)90136-5).
- [66] A. Harten, Adaptive multiresolution schemes for shock computations, *J. Comput. Phys.* 115 (1994) 319–338.
- [67] A. Sinhababu, A. Bhattacharya, S. Ayyalasomayajula, An efficient pseudo-spectral based phase field method for dendritic solidification, *Comput. Mater. Sci.* 186 (October 2020) (2021) 109967, [http://dx.doi.org/10.1016/j.commatsci.2020.109967](https://doi.org/10.1016/j.commatsci.2020.109967).
- [68] L. Le Gratiet, J. Garnier, Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, *Int. J. Uncertain. Quantif.* 4 (5) (2014).
- [69] H. Salimbeni, M. Deisenroth, Doubly stochastic variational inference for deep gaussian processes, in: *Advances in Neural Information Processing Systems*, 2017.

- [70] B. Matérn, *Spatial Variation*, vol. 36, Springer Science & Business Media, 2013.
- [71] J. Hensman, N. Fusi, N.D. Lawrence, Gaussian processes for big data, in: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, AUAI Press, Arlington, Virginia, USA, 2013, pp. 282–290.
- [72] M.D. Hoffman, D.M. Blei, C. Wang, J. Paisley, Stochastic variational inference, *J. Mach. Learn. Res.* 14 (5) (2013).
- [73] P. Kumaraswamy, A generalized probability density function for double-bounded random processes, *J. Hydrol.* 46 (1–2) (1980) 79–88.
- [74] S. Fletcher, K. Ponnambalam, Estimation of reservoir yield and storage distribution using moments analysis, *J. Hydrol.* 182 (1–4) (1996) 259–275.
- [75] I.M. Sobol, *A Primer for the Monte Carlo Method*, CRC Press, 1994.

A.3 Multi-fidelity Bayesian optimization to solve the inverse Stefan problem



RightsLink

[Sign in/Register](#)





Multi-fidelity Bayesian optimization to solve the inverse Stefan problem
Author: J.M. Winter,R. Abaidi,J.W.J. Kaiser,S. Adami,N.A. Adams
Publication: Computer Methods in Applied Mechanics and Engineering
Publisher: Elsevier
Date: 15 May 2023
© 2023 Elsevier B.V. All rights reserved.

Journal Author Rights

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: <https://www.elsevier.com/about/our-business/policies/copyright#Author-rights>

[BACK](#) [CLOSE WINDOW](#)



ELSEVIER



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 410 (2023) 115946

Computer methods
in applied
mechanics and
engineering

www.elsevier.com/locate/cma

Multi-fidelity Bayesian optimization to solve the inverse Stefan problem

J.M. Winter^{a,*}, R. Abaidi^{a,*}, J.W.J. Kaiser^a, S. Adami^{a,b}, N.A. Adams^{a,b}

^a *Technical University of Munich, School of Engineering & Design, Department Engineering Physics and Computation, Chair of Aerodynamics and Fluid Mechanics, Boltzmannstraße 15, 85748 Garching, Germany*

^b *Technical University of Munich, Munich Institute of Integrated Materials, Energy and Process Engineering (MEP), Lichtenbergstr. 4a, 85748 Garching, Germany*

Received 22 December 2022; accepted 6 February 2023

Available online xxxx

Abstract

In this work, we propose an efficient solution of the inverse Stefan problem by multi-fidelity Bayesian optimization. We construct a multi-fidelity Gaussian process surrogate model by combining many low-fidelity estimates of a solidification problem with only a few high-fidelity measurements. To solve the inverse problem, we employ the Gaussian process model in a Bayesian optimization approach based on a multi-fidelity knowledge gradient acquisition function. To account for the specific structure of the target function, we reformulate it as a composite function and thus significantly improve the stability of the optimization procedure. Target values can be switched easily, and previously obtained samples and surrogate models can be reused. The proposed method iteratively improves the recommended solution of the inverse problem. Explicitly adding recommended points of previous iterations to the solution procedure enhances the convergence properties of the algorithm.

We demonstrate the applicability of the algorithm by solving the inverse problem for a planar solidification front in a single-fidelity setting. Process parameters are identified for targeted crystal-growth velocities during directional dendritic solidification. The relation between these velocities and process parameters, such as undercooling, thermal diffusivity, capillarity, and capillary anisotropy, defines the thermo-mechanical properties of the solidified part in metal-based additive manufacturing. Material design is based on the corresponding inverse problem. Cost-efficiency of solving the inverse problem is improved by introducing a fidelity hierarchy based on coarse-grid approximations of high-fidelity numerical simulations. The open-source simulation framework ALPACA for multiphase flows allows to generate data at all fidelities. We demonstrate the superior convergence properties of the presented multi-fidelity approach by comparison with an approach solely based on high-fidelity measurements of the tip velocity.

© 2023 Elsevier B.V. All rights reserved.

Keywords: Bayesian optimization; Multi-fidelity modeling; Inverse problem; Dendritic growth; Multiresolution

* Corresponding authors.

E-mail addresses: josef.winter@tum.de (J.M. Winter), rim.abaidi@tum.de (R. Abaidi), jakob.kaiser@tum.de (J.W.J. Kaiser), stefan.adami@tum.de (S. Adami), nikolaus.adams@tum.de (N.A. Adams).

¹ Both authors contributed equally.

1. Introduction

Solidification processes play a pivotal role in most manufacturing applications, from established manufacturing techniques such as steel casting to modern additive manufacturing. Metal microstructure evolution during solidification directly determines the mechanical properties of the solid phase and, therefore, the properties of the manufactured parts. Kurz et al. [1,2] provide detailed reviews on recent research.

The mathematical description of the solidification processes leading to microstructure formation poses a Stefan problem [3]. A Stefan problem in general describes a free boundary problem between phases of a material undergoing phase change. Examples are the solidification of metals, freezing of water, and crystal growth. Solving the Stefan problem is often understood as solving the direct problem. That is, given initial conditions, boundary conditions, and the material properties, the temporal evolution of the phase boundary, i.e. its position and velocity in normal direction is solved. To solve the Stefan problem numerically, different methods can be used. They differ mostly in the methodology used to describe the phase interface. Prominent examples are phase-field [4–6], level-set [7], or volume of fluid [8–10] methods.

Besides solving the direct Stefan problem, the rapid and efficient design of new materials with targeted properties requires solving the inverse problem [11,12]. Taking dendritic growth processes as example, this means identifying the material and local melt parameters, such as the undercooling, yielding a targeted growth of the dendrite. Çadırlı et al. [13] establish a clear correlation between the growth velocity and mechanical properties, such as microhardness and ultimate tensile strength for aluminum 7075 alloys. They consider the inverse Stefan problem to select candidates and appropriate process conditions that would yield certain properties. Jochum [14,15] applies a non-linear approximation to the one-dimensional one-phase inverse Stefan problem. The problem is reformulated as choosing a heat flux that minimizes the norm of the difference between a given position of the interface and the resulting position for the heat flux. Tarzia et al. [16] found the unknown parameters of a solidifying body such as specific heat, density and latent heat in the context of a one-dimensional one-phase solidification setting. A similar endeavour in the case of a two-phase problem is undertaken in [17]. Ang et al. [18] formulate a one-dimensional one-phase inverse Stefan problem that determines the heat flux at the interface, which has a specified position. The problem is reformulated as a first-kind Volterra integral equation. Murio [19] suggested a method of mollification to numerically solve ill-posed problems. The method was applied to a one-dimensional one-phase inverse Stefan problem, where the temperature and heat flux were reconstructed at the domain boundary with a prescribed interface position. In [20], Zabarás et al. reformulated a two-dimensional inverse Stefan problem into a minimization problem and resorted to dynamic programming methods for its solution. In [21], an inverse problem, where both conduction and convection are considered in the liquid phase, is split into two solvable problems: a direct convection problem in the liquid, and an inverse problem in the solid. Voller [22] used enthalpy and sensitivity coefficients for the task of finding the heat flux at a boundary with a prescribed position of the interface. The Adomian decomposition method combined with optimization was also used in [23,24] to solve the one-phase inverse Stefan problem.

State-of-the-art literature highlights the importance and complexity of solving the inverse Stefan problem. Often, the methodology to obtain the solution relies on specific mathematical formulations. Also, most of the approaches only present results for one-phase and one-dimensional inverse Stefan problems. To circumvent these drawbacks, we employ a data-driven optimization-based methodology to solve the inverse Stefan problem for one- and two-dimensional settings. Therein, the target function which is optimized allows to infer the solution of the inverse problem. This way, we circumvent the need for a mathematical reformulation of the problem. Instead we use the methods already available to solve some types of direct Stefan problems to infer the reverse process. Słota [25] uses a similar approach based on a genetic algorithm to select the convective heat-transfer coefficient of the boundary with a given interface position. Our approach is based on Bayesian optimization (BO) and Gaussian Process (GP) surrogate models due to their flexibility and feasibility of cost-efficient solution strategies. Statistical learning provides techniques to construct sophisticated surrogate models that can accurately capture the response of complex engineering systems, see [26–29]. GP regression is a prominent example of statistical learning and successfully used in modeling of additive-manufacturing processes [30–33] of, e.g., composites [34] or ceramics [35,36]. BO is particularly suitable for optimizing expensive black-box functions, when the gradient information is not available [37,38]. It constructs a surrogate for the objective function with uncertainty quantification by a Bayesian machine learning method, namely GP regression. An acquisition function is evaluated based on the surrogate to choose where to sample next [39]. Surrogate models such as feed-forward neural networks (NN) provide alternatives

to GPs. However, they are more effective when used with large-size training data. On the other hand, GP regression performs favorably with small- to medium-size training sets [40].

We employ a composite structure of the target function, as proposed by Astudillo and Frazier [41]. This improves stability of the optimization procedure, and allows reusing surrogate models, trained for previous target values, for problems with new, unseen target values. Also, we employ multi-fidelity modeling to efficiently solve the optimization problem by combining a large number of rough estimates of the target function (low-fidelity model) with only a few high-fidelity measurements.

Another important aspect of GPs is transfer learning, which incorporates the knowledge from an auxiliary task to improve the learning and the inference of the targeted initial task. This is especially useful when the generation of training-data is expensive [42]. Multi-task GPs are one of the common approaches of transfer learning, where the GP is trained with different tasks in parallel. It uses the inductive bias learned from the auxiliary tasks to infer the target. A subset of multi-task approaches is multi-fidelity modeling, which uses the correlations between high- and low-fidelity training data to make more accurate predictions of the high-fidelity function [40]. It is a widespread concept for efficient construction of response surfaces [43–46], uncertainty quantification [47–51] or solving inverse problems [52]. Low-fidelity data are for example coarse-grid approximations, previously developed experimental correlations, or models with reduced dimensionality [44,53]. High-fidelity data are by definition more accurate and thus more expensive to obtain. These can be fine-grid simulations or data gathered from experiments. The aim of such multi-fidelity models is to use as little high-fidelity data as possible, while maximizing prediction accuracy. Within a BO framework, the balance between sample fidelity and accuracy is subject of current research. Several multi-fidelity BO frameworks have been developed in the recent years. Huang et al. [54], Lam et al. [55] and Picheny et al. [56] proposed an augmented expected improvement (EI), that handles multi-fidelity modeling. Kandasamy et al. [57] developed a multi-fidelity acquisition function based on the GP upper confidence bound (GP-UBC) [58]. Wu et al. [59] came up with a trace-aware knowledge-gradient (taKG) acquisition function, where a function that models the cost of observing points based on their fidelity is introduced. The selection of the candidate and its fidelity is based on the trade-off between the extent to which sampled low-fidelity data improves the quality of the solution, and the cost for generating the sample data. Moreover, the formulation allows both batch and sequential optimization. The optimization of synthetic functions shows a faster convergence and better accuracy of the taKG over other state-of-the-art multi-fidelity acquisition functions, while minimizing the costs. The taKG acquisition function is extended to continuous fidelity-levels and referred to as continuous-fidelity knowledge-gradient (cfKG).

In this work, we propose a general multi-fidelity framework for solving inverse Stefan problems based on BO. The approach combines stochastic multi-fidelity surrogate modeling based on GPs with a multi-fidelity knowledge-gradient acquisition function. To account for the nested structure of the target function, we use composite BO to improve the convergence properties of the iterative solution procedure. Additionally, we suggest recommendation fitting to further improve the convergence properties. The implementation of the optimization algorithm is based on GPyTorch [60] and BoTorch [61]. We show its applicability by solving the inverse problem for a moving ice front and a directional dendritic solidification problem. For the latter, we use the open-source multiphase simulation framework ALPACA [62] as data generator, and base a hierarchy of fidelity levels on coarse-grid approximations of the high-fidelity model.

The structure of the paper is as follows. In Section 2 we describe the BO algorithm used to solve the inverse Stefan problems. In Section 3 we describe the multi-fidelity Bayesian optimization (MFBO) algorithm. In Section 4 we introduce several measures used to monitor the performance and accuracy of the solution procedure. In Section 5, we introduce the general formulation of an inverse Stefan problem. In Sections 6 and 7 we apply the algorithm to solve two different Stefan problems. Conclusions are given in Section 8.

2. Bayesian optimization

We consider the set of design parameters $\mathbf{x} := \{x_1, x_2, \dots, x_d\}$ on the feasible space $\mathcal{X} \in \mathbb{R}^d$. The quantities of interest (QoIs) are acquired by running the simulator, which is in this work a numerical solver. The target of BO is to maximize a function $\mathbf{y}(\mathbf{x})$ that is given by the simulator outputs. The goal is to find \mathbf{x}_{opt} such that

$$\mathbf{x}_{\text{opt}} = \arg \max_{\mathbf{x} \in \mathcal{X}} \mathbf{y}(\mathbf{x}) \quad (1)$$

A dependency between the simulator observations \mathbf{y} and the parameters \mathbf{x} is usually expressed as

$$\mathbf{y}(\mathbf{x}) = f(\mathbf{x}) + \boldsymbol{\epsilon} \quad (2)$$

where $\boldsymbol{\epsilon}$ represents the observation noise and is assumed to be sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$.

Typically, f is expensive to evaluate, limiting the number of affordable evaluations to a few hundred. Information about the structure of the function such as concavity or linearity are missing. Moreover, evaluating f does not include any information about first- or second-order derivatives. As a solution, we construct a surrogate $\tilde{f}(\mathbf{x})$ that uses the simulator as a blackbox. A training set $\mathcal{D}_n = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^n$ is used to construct the surrogate, where \mathbf{X} are samples of \mathbf{x} and $\mathbf{Y} = \mathbf{y}(\mathbf{X})$ are their corresponding simulator outputs.

BO contains two components: a statistical model that acts as a surrogate for the objective function and an acquisition function that drives the search for the optimal candidate. The interplay of these two components creates the BO framework. In this work, we use a GP as a surrogate model for the simulator. A GP is a ‘‘collection of random variables, any finite number of which have a joint Gaussian distribution’’ [26]. For the optimization task, the GP is used as a prior for the function $f(\mathbf{x})$. It is specified by its mean function and its covariance function, which are defined as

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))] \end{aligned} \quad (3)$$

In this case, the GP can be written as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (4)$$

The posterior $f | \mathcal{D}_n$ conditioned on the training set \mathcal{D}_n is also a GP with a mean function and covariance function defined as

$$\begin{aligned} \mu_n(\mathbf{x}) &= k^\top (K + \eta^2 I_n)^{-1} \mathbf{y}, \\ \kappa_n(\mathbf{x}, \mathbf{x}') &= \kappa(\mathbf{x}, \mathbf{x}') - k^\top (K + \eta^2 I_n)^{-1} k' \end{aligned} \quad (5)$$

where $k, k' \in \mathbb{R}^n$ are such that $k_i = \kappa(\mathbf{x}, \mathbf{x}_i)$, $k'_i = \kappa(\mathbf{x}', \mathbf{x}_i)$. I_n is the $n \times n$ identity matrix and $K \in \mathbb{R}^{n \times n}$ is given by $K_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. The posterior is used to compute the acquisition function $\varphi_t : \mathcal{X} \rightarrow \mathbb{R}$. The point maximizing the acquisition function, $\mathbf{x}_t \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \varphi_t(\mathbf{x})$, is the next point to evaluate. The acquisition function encodes the utility of observing the point \mathbf{x} based on the posterior. Generally, the acquisition function is defined in such a way that a high value corresponds to a high value of the objective function, either due to a high prediction or a high uncertainty, or a combination of both. One of the most used acquisition functions is EI, which might tend to exploit the space rather than to explore it. This may lead to exploiting regions close to local optima and thus missing the global optimum.

Acquisition strategy

For the optimization, a novel acquisition strategy, based on the KG acquisition function [63,64] and an exploitation of the current optimum of the posterior, is adopted in this work. In the following, we describe the strategy for the single-fidelity setting. The extension to multi-fidelity is carried out in Section 3.

Gupta and Miescke [63] introduced the KG acquisition function as simple look-ahead policy that provided tractability. Frazier et al. [64] analyzed it and referred to it as KG policy. It was extended to include correlated beliefs in [65] and to enable parallel batch-optimization in [66]. The parallel KG is referred to as q-KG.

The KG policy chooses the next sampling candidate, such that it maximizes the expected incremental value of an observation. Unlike the EI, it does not assume that the optimum must be a previously observed point, and the reward is received only after the final observation.

If we have collected n observations of the simulator and we were to terminate the optimization loop, $\max_{\mathbf{x} \in \mathcal{X}} \mu_n(\mathbf{x})$ would be the maximum of the posterior mean. If we were to allow one more batch of samples of the size q , the maximum of the posterior mean would be $\max_{\mathbf{x} \in \mathcal{X}} \mu_{n+q}(\mathbf{x})$. The expected incremental value that results from the additional batch is the difference $\max_{\mathbf{x} \in \mathcal{X}} \mu_{n+q}(\mathbf{x}) - \max_{\mathbf{x} \in \mathcal{X}} \mu_n(\mathbf{x})$. The probability distribution

of the difference can be computed for the batch of sampling decisions $\mathbf{z}_{1:q} := \{z_1, z_2, \dots, z_q\}$ by defining the q-KG factor

$$\text{q-KG}(\mathbf{z}_{1:q}, \mathcal{X}) = \mathbb{E}_n \left[\max_{x \in \mathcal{X}} \mu_{n+q}(x) \mid \mathbf{y}(\mathbf{z}_{1:q}) \right] - \max \mu_n(x) \quad (6)$$

where $\mathbb{E}_n[\cdot] := \mathbb{E}[\cdot \mid \mathbf{x}^{(1:n)}, \mathbf{y}^{(1:n)}]$ is the expectation with respect to the posterior distribution after n observations. The batch $\mathbf{z}_{1:q}$ of points that maximize the q-KG factor is the next to be sampled.

The idea behind our modified KG policy originates in the analogy between the optimization process and a dynamical system. Since the reward of the q-KG policy in [66] is only received after the final observation, the system does not receive a direct feedback on its performance throughout the whole optimization loop. Observing the candidates that maximize the q-KG factor gives implicitly a feedback to the system, but it is not a direct indicator of the optimization performance. It is rather a feedback on the direction of the search. This leads to an exploratory behavior of the search space. On the other hand, other acquisition functions such as EI receive a direct feedback at every iteration, as the next observed point is also the reward. This behavior favors exploitation of the current best optimum.

The idea is to add an extra step to the optimization that would give a feedback to the system on its performance, by observing the reward at some intermediate stages of the optimization and not only in the end. This step is referred to as recommendation, as the system recommends a point \mathbf{x}_{rec} , that is believed to be the optimum based on the posterior mean of the surrogate.

The following algorithm describes the steps of the optimization with the modified q-KG acquisition policy. Steps 6 and 7 are the addition to the original algorithm in [66].

Algorithm 1: Bayesian Optimization with modified q-KG policy

Require: number of initial samples I , budget on feasible observations N and recommendation interval s_{rec} .

1. Initial stage: Construct a quasi-random batch of I points in \mathbb{X} , \mathbf{x}_i for $i = 1, \dots, I$.
2. Observe the simulator at the samples \mathbf{x}_i . Let \mathcal{D} represent the complete collection of observations.
3. Set $s=0$.

while $s < N$ **do**

4. Get $(z_1^*, z_2^*, \dots, z_q^*) = \text{argmax}_{z_{1:q} \subset \mathbb{X}} \text{q-KG}(z^{(1:q)}, \mathbb{X})$

5. $\mathcal{X}_{opt} = \{z_1^*, z_2^*, \dots, z_q^*\}$ and $s = s + q$

if $s \pmod{s_{rec}} = 0$ **then**

6. $\mathbf{x}_{rec} = \text{argmax}_{x \in \mathbb{X}} \mu_{I+s}(x)$.

7. $\mathcal{X}_{opt} = \mathcal{X}_{opt} \cup \{\mathbf{x}_{rec}\}$ and $s = s + 1$

end

8. Sample the points in \mathcal{X}_{opt} , retrain the hyperparameters of the GP by MLE, and update the posterior distribution of f .

end

return $x^* = \text{argmax}_{x \in \mathbb{X}} \mu_{I+s}(x)$.

Composite function

In this work, we solve inverse problems using BO optimization. Let \mathbf{y}_{target} be the targeted simulator output, for which the input parameters are sought. The optimization task is then

$$\min_{\mathbf{x} \in \mathcal{X}} \|f(\mathbf{x}) - \mathbf{y}_{target}\|, \quad (7)$$

where $\|\cdot\|$ is often the L_1 or L_2 norm. If we define the function $g(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}_{target}\|$, then the optimization task can be reformulated as

$$\min_{\mathbf{x} \in \mathcal{X}} g(f(\mathbf{x})). \quad (8)$$

Astudillo and Frazier [41] introduced a novel BO approach that uses the composite structure of the objective function and results in a more efficient optimization. This approach consists in building a GP model on f and then using

the EI under the implied statistical model on $g \circ f$ to direct the search for new points to sample. The acquisition function is referred to as EI-CF. The intuition behind this approach is that the observations $f(x)$ might supply relevant information to the optimization that are not provided by the observations $g(f(x))$.

In this work, we make use of the composite structure of the objective function and combine the approach with our acquisition policy described above. There are three main advantages to this approach compared to a standard BO:

- The norm function g may be non-differentiable at some points of the domain, thus non-smooth, which makes it inadequate for the usual kernel functions for the GP model.
- Re-usability of observations from previous experiments with different target values.
- The GP model, fitted directly with the simulator outputs, might give some insights about the underlying function of the simulator.

3. Multi-fidelity Bayesian optimization

Standard BO methods are often studied in a single-fidelity setting, where there is one single expensive-to-evaluate objective function. However, cheaper approximations of the function are usually available and can be used to discard regions of the space that might not be promising [67]. We refer to these approximations as “information sources (IS)”. Examples of a multi-fidelity source of information are numerical simulations generated by using different grid refinement levels.

We denote the objective function and its lower-fidelity approximations by $g(x, s)$. The expensive objective function is defined as $f(x) = g(x, 1_m)$ and $s \in [0, 1]^m$ are the m fidelity-parameters. The final goal is to maximize the high-fidelity function f . Given that, in a multi-fidelity setting, the cost of evaluating the function depends on the fidelity level, the acquisition policy has to take into account both the value of information provided by a design point and the cost of querying it. In this work, cfKG function [68] is adopted as acquisition policy. Analogous to the single-fidelity implementation, we modify the strategy by observing intermediately promising high-fidelity design points based on the posterior distribution.

We now briefly describe the statistical model that integrates the different IS and then discuss the acquisition strategy.

Multi-fidelity surrogate model

The statistical model is a GP that has a kernel function appropriate for multi-fidelity IS. The kernel structure is originally adopted in the work of Wu et al. [59] for hyperparameter optimization.

We assume $g \sim \mathcal{GP}(\mathbf{0}, \kappa)$, and a query at $z = (x, s)$ gives the observations $y = g(x, s) + \epsilon$. κ is the prior kernel function. For two levels of fidelity s_1 and s_2 , the kernel function is defined as

$$\kappa(z, z') = K(x, x') \times K_1(s_1, s'_1) \times K_2(s_2, s'_2) \quad (9)$$

where $K(\cdot, \cdot)$ is a square-exponential kernel. The idea of the kernel K_1 follows from the work of Swersky et al. [69]. It is defined as

$$K_1(s_1, s'_1) = w + \frac{\beta^\alpha}{s_1 + s'_1 + \beta^\alpha} \quad (10)$$

where w, α and β are strictly positive hyperparameters. The kernel K_2 is defined as

$$K_2(s_2, s'_2) = c + (1 - s_2)^{1+\delta} + (1 - s'_2)^{1+\delta} \quad (11)$$

where c and δ are also strictly positive hyperparameters.

Multi-fidelity acquisition function

The cfKG [68] acquisition function extends the q-KG policy [66] to multi-fidelity problems where the fidelity levels are allowed to be continuous. Similarly to q-KG, the strategy is to maximize the information gained by one additional observation.

The solution after sampling n points would be the highest fidelity point x_{HF}^* maximizing the posterior mean, i.e. $x_{\text{HF}}^* = \operatorname{argmax}_{x \in \mathcal{X}} \mu_n(x, 1_m)$.

If we were to sample one additional point x_{n+1} with fidelity s_{n+1} and the simulator outcome y_{n+1} , the gained information value is the difference $\max_{x \in \mathcal{X}} \mu_{n+1}(x, 1_m) - \max_{x \in \mathcal{X}} \mu_n(x, 1_m)$. To obtain the cfKG acquisition function, the expectation of the difference over the random simulator outcome y_{n+1} is normalized with the cost of obtaining the additional sample. This yields

$$\text{cfKG}(x, s) = \left(\mathbb{E}_n \left[\max_{x' \in \mathcal{X}} \mu_{n+1}(x', 1_m) \mid x_{n+1} = x, s_{n+1} = s \right] - \max_{x' \in \mathcal{X}} \mu_n(x', 1_m) \right) \frac{1}{\text{cost}_n(x, s)} \quad (12)$$

where $\text{cost}_n(x, s)$ is the estimation of the cost of querying the point x at fidelity s . This cost depends also on the samples observed until iteration n . The next sample according to the cfKG policy is the pair of design point and fidelity that jointly maximize the cfKG acquisition function. The cfKG generalizes to q-cfKG to include batch optimization, which then can sample a set of pairs (design point and fidelity) in one iteration. If we define this set of be $z_{1:q} = ((x_1, s_1), \dots, (x_q, s_q))$, where $q \geq 1$ is the size of the set, then the q-cfKG acquisition function is formulated as

$$\text{q-cfKG}(z_{1:q}) = \left(\mathbb{E}_{n+q} \left[\max_{x' \in \mathcal{X}} \mu_{n+q}(x', 1_m) \mid z_{n+1:n+q} = z_{1:q} \right] - \max_{x' \in \mathcal{X}} \mu_n(x', 1_m) \right) \frac{1}{\max_{1 \leq i \leq q} \text{cost}_n(z_{n+i})} \quad (13)$$

Similarly to the single-fidelity setting of our framework, we add the possibility of querying the high-fidelity solution at intermediate stages of the optimization.

4. Definition of terms

In this section, we introduce and define key terms and metrics related to the BO method that are used frequently in this work:

- Recommendation point: after each iteration i , the surrogate model is updated with a new observation. The recommendation point $x_{\text{rec},i}$ is the point that maximizes the posterior mean of the surrogate. This point is the solution of the optimization if iteration i were to be the last.
- True recommendation error: if $\mathbf{y}_{\text{target}}$ is the targeted simulator output, the true recommendation error at iteration i is defined as

$$\epsilon_{\text{true},i} = \frac{|f_{\text{objective}}(x_{\text{rec},i}) - \mathbf{y}_{\text{target}}|}{\mathbf{y}_{\text{target}}} \quad (14)$$

where $x_{\text{rec},i}$ is the recommendation point and $f_{\text{objective}}(x_{\text{rec},i})$ is its objective function evaluation. Since each optimization setup is run multiple times with different random initialization seeds, the error is averaged over the optimization runs. The mean and the standard deviation of the true error are respectively $\overline{\epsilon_{\text{true},i}}$ and $\sigma[\epsilon_{\text{true},i}]$.

- Prediction error: it is defined as the absolute difference between the prediction of the surrogate and the true objective function at a point. In this work, the prediction error is estimated at the recommendation points and the predictions correspond to the value of the posterior mean at those points. The prediction error can be written as

$$\epsilon_{\text{prediction},i} = |f_{\text{objective}}(x_{\text{rec},i}) - f_{\text{surrogate}}(x_{\text{rec},i})|. \quad (15)$$

Here again the error is averaged over multiple optimization runs and the mean is denoted by $\overline{\epsilon}_{\text{prediction},i}$

- Average cumulative prediction error: it is defined as

$$\bar{\epsilon}_{\text{prediction, cumulative}} = \frac{1}{N} \sum_{i=1}^N \bar{\epsilon}_{\text{prediction},i} \quad (16)$$

where N is the number of iterations in a single optimization run.

5. General formulation of the two-phase Stefan problem

Solving the Stefan problem requires finding the temperature and the interface between two phases of a pure substance. The dynamics of these two variables are governed by the heat diffusion from internal or external heat sources. We start by giving a general formulation of the two-phase Stefan problem to then explain the cases of inverse Stefan problems investigated in this work.

We consider a domain D of a pure substance which contains both a liquid and a solid phase. We refer to the solid region as Ω_s and the liquid region as Ω_l . The boundary is denoted by Γ . Heat conduction is present in both phases, yielding the following governing equations:

$$\rho_s c_s \frac{\partial T}{\partial t} = \nabla \cdot (k_s \nabla T), \mathbf{x} \in \Omega \quad (17a)$$

$$\rho_l c_l \frac{\partial T}{\partial t} = \nabla \cdot (k_l \nabla T), \mathbf{x} \in \Omega^c \quad (17b)$$

where c_l and c_s are the heat capacities, ρ_l and ρ_s the densities, k_l and k_s the thermal conductivities and, T_l and T_s the temperatures, respectively in the liquid and solid phases. In order to fully describe the problem, the interface conditions need to be considered in addition to the heat Eqs. (17). We assume a continuous temperature field across the boundary. The interface temperature T_Γ is treated as a Dirichlet boundary condition and follows from the Gibbs-Thomson relation as

$$T_\Gamma = T_m - \frac{\gamma T_m}{\rho_s L} \kappa - \epsilon_u \|\mathbf{u}_\Gamma\|, \quad (18)$$

where T_m is the melting temperature, γ the surface tension coefficient, L the latent heat, κ the boundary curvature, \mathbf{u}_Γ the normal velocity at the interface, and ϵ_u the kinetic coefficient. Additionally, the Stefan condition

$$\begin{aligned} \dot{m} [L + (T_\Gamma - T_m) (c_l - c_s)] &= \rho \|\mathbf{u}_\Gamma\| [L + (T_\Gamma - T_m) (c_l - c_s)] \\ &= (k_s \nabla T_s - k_l \nabla T_l) \cdot \mathbf{n}_\Gamma \end{aligned} \quad (19)$$

describes how the heat fluxes and the latent heat produced by phase change are balanced at the interface. In Eq. (19), \dot{m} is the rate of the liquid mass that is becoming solid and \mathbf{n}_Γ is the normal vector at the interface pointing from the solid to the liquid.

6. Case 1: Growth of a planar solidification front

6.1. Problem formulation

The first case is the classical two-phase Stefan problem, which depicts the growth of a planar solidification front. We simulate the case of an ice front in undercooled water, i.e. the local temperature in the water is below the liquid–solid phase transition temperature, as described in [70]. Solidification is initiated at $t = 0$ from the position $x_\Gamma = 0$ and the front propagates in the liquid domain $x_\Gamma(t) > 0$. The initial and boundary conditions are

$$T_l|_{x,t=0} = T_{l,0} \quad (20a)$$

$$T_l|_{x \rightarrow \infty, t} = T_{l,\infty} = T_{l,0} < T_m \quad (20b)$$

$$T_s|_{x,t=0} = T_{s,0} < T_m, \quad (20c)$$

Table 1

Material parameters for the one-dimensional Stefan problem: water at 243.15 K and ice at 263.15 K.

	ρ [kg/m ³]	c [J/(kgK)]	k [W/(mK)]	L [kJ/kg]	T_m [K]
Water	1000	4864.0	0.4829	333.6	273.1
Ice	1000	2030.0	2.319		

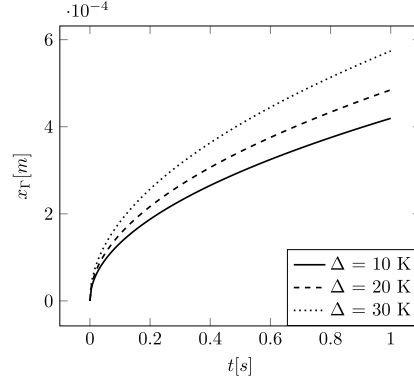


Fig. 1. Temporal evolution of the interface position for different values of the undercooling.

where T_m is the melting temperature assigned to the flat interface. The analytical solution for this system is given by Carslaw and Jaeger [71] and gives the temperature profile as

$$\begin{aligned}
 0 < x < x_I(t), t > 0 : T(x, t) &= T_{s,0} + (T_m - T_{s,0}) \frac{\operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_s t}}\right)}{\operatorname{erf}(\beta)} \\
 x > x_I(t), t > 0 : T(x, t) &= T_{l,0} + (T_m - T_{l,0}) \frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{\alpha_l t}}\right)}{\operatorname{erfc}(v\beta)},
 \end{aligned} \tag{21}$$

with β given by the transcendental equation

$$\beta\sqrt{\pi} = \frac{St_s}{\exp(\beta^2)\operatorname{erf}(\beta)} + \frac{St_l}{v\exp(v^2\beta^2)\operatorname{erfc}(v\beta)}. \tag{22}$$

Here, St_l and St_s are the Stefan numbers in the liquid and solid, respectively and defined by

$$St_l = c_l(T_m - T_l)/L \tag{23a}$$

$$St_s = c_s(T_m - T_s)/L. \tag{23b}$$

v is the square-root of the ratio of the thermal diffusivities of the phases $v = \sqrt{\alpha_s/\alpha_l}$. The temporal evolution of the interface location is given by

$$x_I(t) = 2\beta\sqrt{\alpha_s t}. \tag{24}$$

We reformulate the growth of a solidification front into an inverse problem. The task consists in inferring the initial undercooling $\Delta = T_m - T_{l,0}$ that propagates the interface up until a given position after a given time. Since the melting temperature is fixed, the initial liquid temperature is the output quantity of the inverse problem. We use the material settings found in the original problem proposed by Rauschenberger et al. [70]. The material parameters are described in Table 1. The initial solid temperature $T_{s,0}$ is also kept constant at 263.15 K. For simplicity, the dependency of some of the material parameters on the temperature is not considered in this work.

In order to formulate the desired objective function for the inverse problem, we compute the interface position given by the analytical solution of the growth of the ice front as a function of the undercooling and the time, as can be seen in Fig. 1, where the temporal evolution of the interface position is shown for different values of undercooling.

Table 2

Optimization setups for the inverse 1D Stefan problem.

Optimization setup	Interface position [mm]	Duration [s]
1	0.09	0.04
2	0.3	0.4
3	0.4	0.5

We refer to the solver as $f_{analytical}$ and to the target interface position at the time t_{target} as x_{target} . The inverse problem is then formulated as:

$$\min_{\Delta \in A} f_{objective} = \min_{\Delta \in A} |f_{analytical}(\Delta, t_{target}) - x_{target}| \quad (25)$$

where A is the range of undercooling. We then seek to minimize the error between the computed interface position and the target interface position. Δ^* is the true solution to this problem if $f_{objective}(\Delta^*) = 0$. We use the composite formulation of the objective function to train the GP with the solver outputs.

6.2. Solution to the inverse problem

Table 2 shows the three different optimization setups investigated in the case of the inverse one-dimensional Stefan problem. Three combinations of target interface position and time duration are specified and the goal of the optimization is to find their corresponding undercooling. The initial training-sets of size 2, 5 and 10 are generated quasi-randomly. Each setup, characterized by its target values and the size of the initial set, is run 20 times with different seeds and the results are averaged.

We run a single-fidelity BO and investigate the effect of the acquisition strategy described in Section 2. To this end, we compare the results of a SKG acquisition strategy with the results of a combination of KG and fitting the recommended point to the GP every 1 and 5 iterations of the optimization. For each experiment, a budget of 100 evaluations of the objective function is allowed, not including the initial training points. After each iteration i , the recommendation point $x_{rec,i}$ is evaluated to measure the performance of the optimization algorithm. For the standard KG acquisition strategy, this point is never fitted to the GP. For the modified acquisition strategy, depending on the chosen fitting-interval, this point and its evaluation are added to the training set for the GP-regression as a new sample.

The mean of the true recommendation error $\overline{\epsilon_{true}}$ is shown in Fig. 2. Fitting the recommendation, regardless of the interval, clearly reduces the error across all setups. The standard KG acquisition policy results not only in the highest errors, but also in the slowest decrease of the error during the optimization, which is characterized by the plateauing of the error.

The recommendation-fitting interval has also a big impact on the decrease of the error and the convergence of the optimization. The smaller the interval, the smaller the error is for the same number of evaluations needed. This means that the algorithm converges faster towards the true solution. The results show that the best acquisition strategy is to observe the point suggested by the KG function, followed by observing the recommendation and fitting it to the surrogate model, as this strategy results in the lowest error across all setups and in a steady diminishing of the error. The error is at least two orders of magnitude smaller than the error produced by the SKG.

The size of the initial training-set is an important design choice as can be seen in Fig. 2. A larger size of the initial training-set has the effect of decreasing the error. The choice of the size has to be made carefully, as adding a few extra points to the initial set might provide more benefits than allocating more points later during the optimization. An example is Fig. 2(b) where the setups with initial size 1 and 10 are compared. For each recommendation-fitting interval, the error reached after 10 evaluations for the setup with the larger initial set is not recovered for the one with the smaller initial set even after 100 evaluations.

In Fig. 3, the average cumulative prediction error $\bar{\epsilon}_{prediction, cumulative}$ is shown for the different setups. The predictive accuracy of the surrogate model increases with the size of the initial set. Moreover, the modified KG acquisition policy yields a better prediction accuracy in the near-solution region, than the SKG, and that for both fitting-intervals. This confirms the intuition that recommendation-fitting acts as exploitation of the current solution. As the surrogate model observes many samples, usually clustered in the same region, its predictions become more accurate in that region.

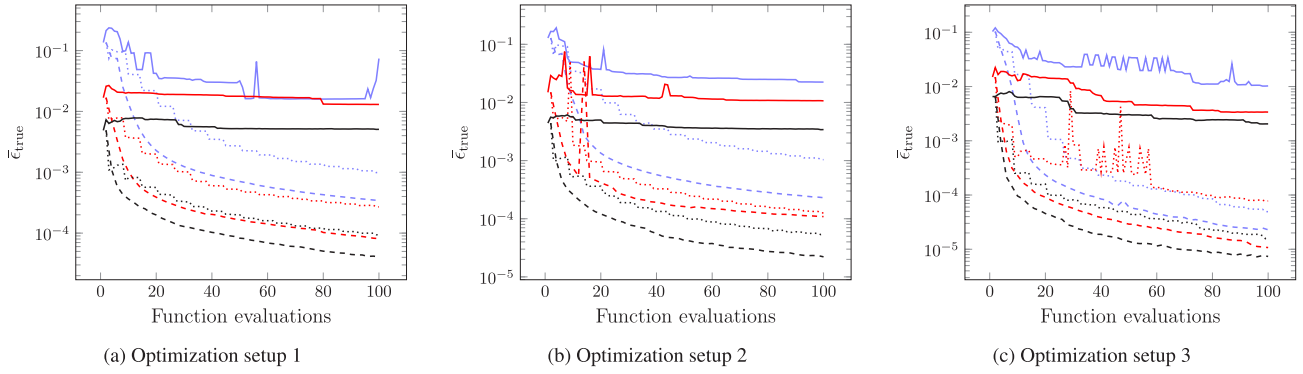


Fig. 2. Mean of true error of recommendations for all optimization setups. The full line corresponds to the SKG acquisition function, dotted line corresponds to the modified KG function with interval 5 and the dashed line corresponds to the modified KG function with interval 1. The blue color corresponds to the initial set of size 2, the red one to the size 5 and the black one to the size of 10. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

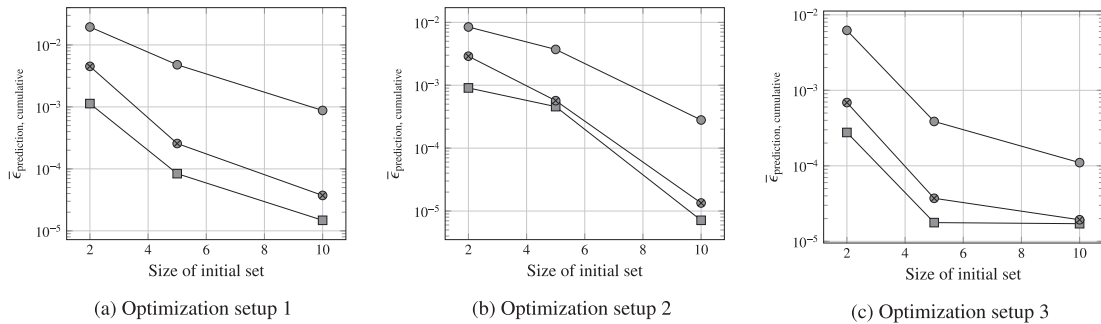


Fig. 3. Average of cumulative prediction errors of recommendations over the optimization iterations, for different sizes of the initial set. The prediction error is evaluated at each iteration and each optimization is run for 100 iterations. For each setup, we show results for a standard KG acquisition function (\bullet), a modified KG function with interval 1 (\blacksquare), and a modified KG function with interval 4 (\blacklozenge).

We have used different setups to solve the one-dimensional inverse Stefan problem for various target-values. The experiments show that the modified KG acquisition policy yields not only smaller errors and thus samples closer to the true solution, but also more accurate predictions. For a budget of 100 function evaluations during optimization, the combination of a larger initial set and a smaller interval for recommendation-fitting solves the inverse problem with an error on the order of 10^{-5} .

7. Case 2: Growth of a crystal with four-fold symmetry

7.1. Problem formulation

The second problem we investigate is directional dendritic solidification. We consider an initially spherical seed, as sketched in Fig. 4, which is surrounded by an undercooled melt. The temperature of the surrounding melt is below the liquid–solid phase transition temperature and leads to crystallization of the seed. Along the main axes, the crystal exhibits growth in preferred directions due to anisotropic surface tension, cf. Fig. 4. The temporal evolution of the crystal is shown in Fig. 4(c). The early phase of the crystallization process is highly transient. Thereafter, the velocity at the dendritic tip converges against a steady-state solution. Many authors [72–77] consider this setup with slight variations for the validation of numerical methods for crystal growth. [78] investigate this setup and develop a stochastic surrogate model for the relation between the tip velocity and material and local melt parameters. They highlight that multi-fidelity modeling based on a resolution-fidelity hierarchy allows cost-efficient generation of a surrogate model while improving model accuracy compared to single-fidelity modeling. This motivates investigating whether multi-fidelity modeling also yields superior performance when solving the inverse Stefan problem. We infer process and local melt parameters that lead to a certain predefined tip velocity. Following [78], we denote

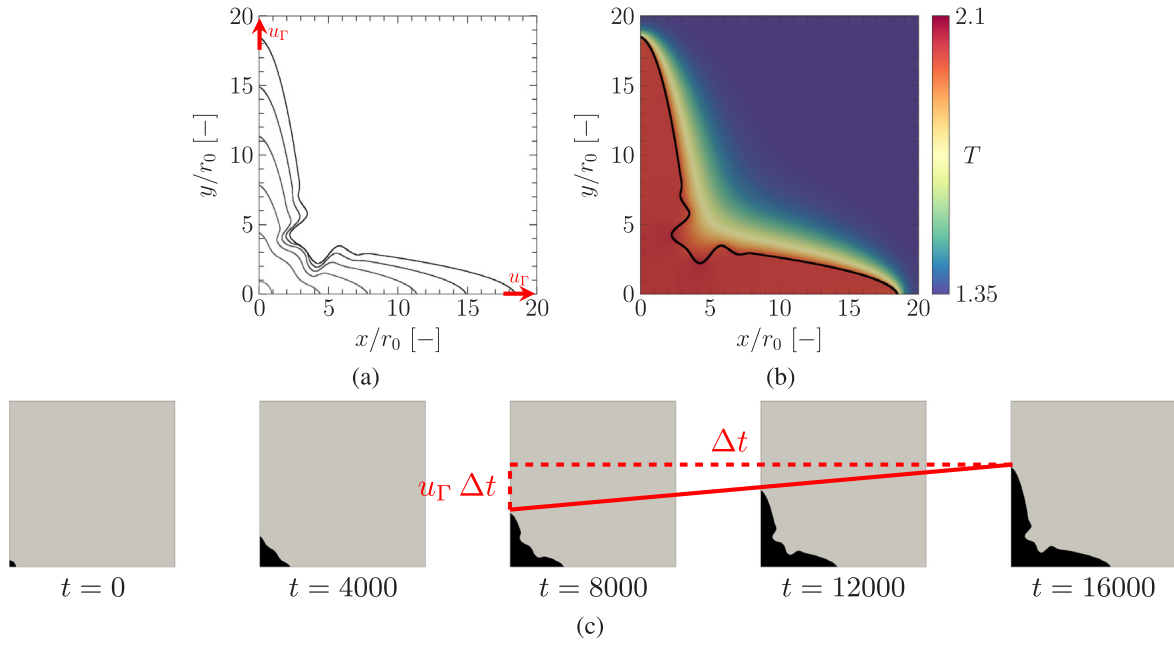


Fig. 4. (a): Solid–liquid interface for the growth of an initially spherical seed with radius r_0 under anisotropic surface tension. Time increases as the interface extends in x and y . The tip-velocity vector with magnitude u_Γ is highlighted in red for the last snapshot. (b): Temperature field and solid–liquid interface (black line) for the last snapshot. (c): Temporal evolution of the crystal for a resolution of $64 \frac{cells}{r_0}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the steady-state tip velocity as u_Γ . It is a function of the normalized undercooling \mathcal{T} , capillary anisotropy ε , and thermal-conductivity ratio α_m/α_s between melt and solid. The normalized undercooling reads

$$\mathcal{T} = \frac{(T - T_m) c}{L}, \quad (26)$$

where T denotes the temperature in the melt, T_m the melting temperature, c the specific heat capacity, and L the latent heat. It corresponds to the dimensionless Stefan number St . Thus, the functional dependence for the tip velocity is

$$u_\Gamma = f(St, \varepsilon, \alpha_m/\alpha_s). \quad (27)$$

We denote the vector of independent parameters as

$$\mathbf{x} = (St, \varepsilon, \alpha_m/\alpha_s) \in \mathbb{R}^3, \quad (28)$$

where $0.5 \leq St \leq 0.9$, $0.02 \leq \varepsilon \leq 0.05$, and $0.5 \leq \alpha_m/\alpha_s \leq 1$ are the bounds of the parameter space. This allows formulating the inverse problem as optimization task

$$\min_{\mathbf{x}_{dendrite}} f_{\text{objective, dendrite}} = \min_{\Delta} |u_\Gamma(\mathbf{x}_{dendrite}) - u_{\Gamma, \text{target}}|, \quad (29)$$

where $u_{\Gamma, \text{target}}$ is the predefined, targeted tip velocity. To determine the tip velocity for arbitrary operating points, we employ the numerical model for liquid–solid phase transition as presented in Kaiser et al. [79]. It is implemented in the open-source simulation framework ALPACA [62,80]. This model captures the interface using a sharp-interface level-set function and considers the phase change between the solid and the melt using conservative interface-interaction terms. The model offers full spatio-temporal adaptivity through a multiresolution approach with adaptive local timestepping [81], which is based on the multiresolution approach of Harten [82] with level-dependent cell sizes. This comprises a natural multi-fidelity property, and allows constituting multi-fidelity data sets by varying the maximum refinement level as shown in [78]. A detailed description and convergence analysis of the investigated setup can be found in [78].

To study the behavior of the multi-fidelity BO approach applied to the inverse problem for the dendritic growth in detail, we need a huge number of evaluations of the target function. At the same time, we aim to have an accurate,

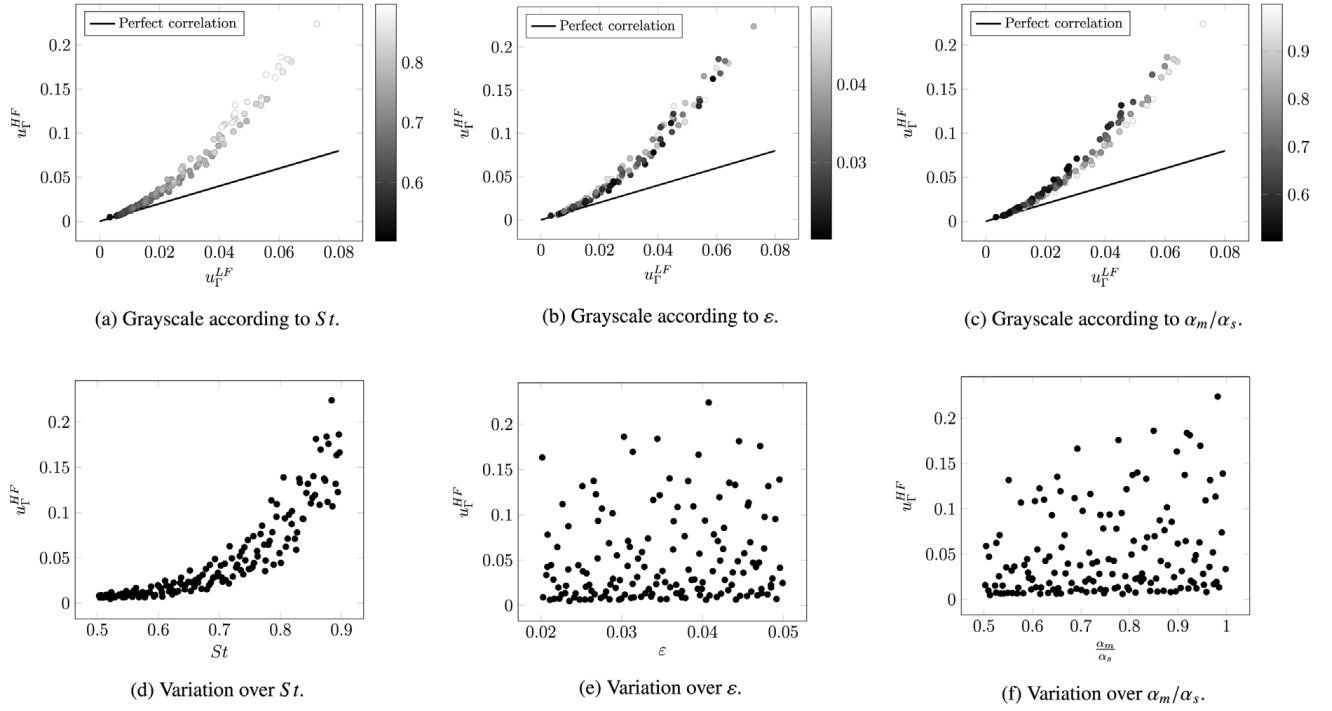


Fig. 5. Top row: Correlation between u_T obtained from the low-fidelity and high-fidelity model. They are denoted as u_T^{LF} and u_T^{HF} , respectively. From left to right, the grayscale of the points is according to St , ε , and α_m/α_s . Bottom row: u_T^{HF} plotted over St , ε , and α_m/α_s , respectively.

i. e. highly resolved, high-fidelity model for the analysis. Thus, we base the definition of the fidelity hierarchy on a preexisting data set, that was used in [78]. Taking further into account the available computational resources, we define the low- and high-fidelity models:

- **Low fidelity:** Following [78], simulations are performed with a spatial resolution of 8 cells per initial seed radius. This results in very short computing times, yet with significant errors in the predicted steady-state velocity.
- **High fidelity:** Simulations are performed with a spatial resolution of 32 cells per initial seed radius. This corresponds to two additional refinement levels in the multiresolution (MR) tree compared to the low-fidelity setup.

The preexisting data set consists of $N_{LF} = 1000$ low-, and $N_{HF} = 400$ high-fidelity samples. Based on 150 samples available for both fidelities, we visualize their correlation in Fig. 5. The correlation between low- and high-fidelity model is nonlinear. For low tip velocities, the low-fidelity model gives accurate estimates of the true, high-fidelity tip velocity. For increasing tip velocities, the correlation is worse, and the low-fidelity model strongly underestimates the high-fidelity model. Fig. 5(d) shows that the tip velocity exponentially grows with the Stefan number. For the capillary anisotropy and the thermal conductivity ratio, cf. Figs. 5(e) and 5(f), no clear trends can be deduced. Based on the 150 samples available at each fidelity, [78] measure that the compute time for a high-fidelity sample is on average by a factor of 27 more expensive than a low-fidelity sample. When adding one additional multiresolution level to the present high-fidelity model, a sample is on average by a factor of 224 more expensive than a low-fidelity sample. Based on this information, we deduce a quadratic cost model, that is necessary for the multi-fidelity BO strategy. We linearly map the fidelity-level resolutions onto a continuous fidelity parameter $0.0 \leq s \leq 1.0$, and define that $s = 0.0$ corresponds to the low-fidelity model, $s = \frac{2}{3}$ corresponds to the high-fidelity model, and $s = 1.0$ corresponds to one additional refinement level compared to the high-fidelity model. Based on the available cost estimates, this allows fitting the quadratic cost function

$$c^*(s) = 552.0s^2 - 329.0s + 1.0. \tag{30}$$

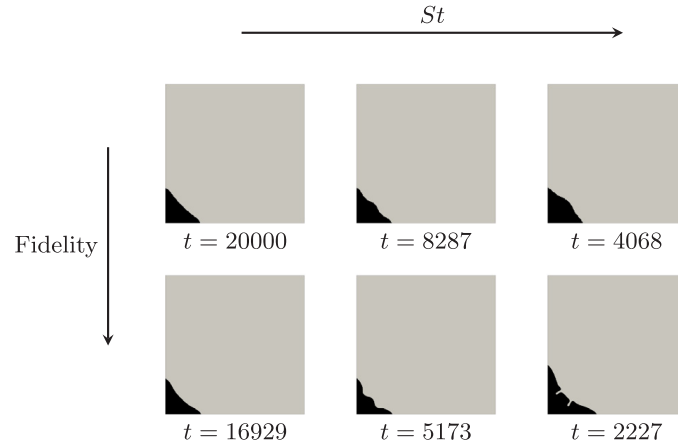


Fig. 6. Shape of the dendrite when a steady-state tip velocity has developed. The low-, and high-fidelity results are shown in the top and bottom row, respectively. The left, middle, and right column show results for $\mathbf{x} = [0.523, 0.0481, 0.921]$, $\mathbf{x} = [0.721, 0.0422, 0.804]$, and $\mathbf{x} = [0.897, 0.0395, 0.693]$ respectively. The respective time is indicated for each plot.

In this work, we scale the cost function to

$$c(s) = c^* \left(\frac{2}{3}s \right). \tag{31}$$

Then, $s = 0.0$ corresponds to the low-fidelity model, and $s = 1.0$ corresponds to the high-fidelity model. To ensure that the acquisition function only proposes points corresponding to the low-, and high-fidelity model, we limit s to the discrete set $s \in 0.0, 1.0$.

Fig. 6 shows the dendrite during steady growth for different parameters for both fidelity levels. From left to right, St , and thus u_T , increases. For higher St , the tip is more pronounced and depicts more instabilities. For the highest Stefan number (right column), the dendrite develops secondary branches for the high-fidelity model. Due to the low resolution, the low-fidelity model suppressed these secondary instabilities. Increasing tip velocities, which occur for higher St , require resolving smaller tip radii. For high tip velocities, the low-fidelity model is not able to accurately capture the required tip radii. Thus, in this region, the low-fidelity model strongly deviates from the high-fidelity model and underestimated the high-fidelity tip velocity, as already seen in Fig. 5.

7.2. Solution of the inverse problem

We apply the MFKG algorithm in combination with the composite BO in order to solve the inverse problem defined in Eq. (29). We run the BO algorithm for 12 iterations and fit the recommendation point after every 4th iteration. For each iteration we perform a recommendation, and evaluate the true target function at the predicted solution point.

The computational cost to evaluate the sample proposed by the acquisition function dominates the cost of a single iteration of the optimization algorithm. Due to the multi-fidelity setting, the cost to evaluate a single sample varies strongly, depending whether a low- or high-fidelity sample has to be generated. Thus, besides using the iteration index, the evolution of the optimization algorithm can also be tracked by the cumulative cost of the samples iteratively proposed by the acquisition strategy. We denote this cumulative cost by $c_{\text{cumulative}}$.

To account for the randomness in choosing the initial training set, we create an ensemble of 10 different random realizations of each optimization run, and average their results. Thus, we investigate the convergence of the algorithm using the mean and standard deviation of the true recommendation error over all available runs. Also, we can use the mean value of the cumulative cost $\overline{c_{\text{cumulative}}}$ to track the evolution of the algorithm. Note, $\sigma[\epsilon_{\text{true}}]$ measures the sensitivity of the MFKG algorithm to the random selection of training points. We explore the behavior of the algorithm by varying several parameters. We solve the inverse problem for several target tip velocities $u_{T,\text{target}}$, and consider different initial training sets for each. $N_{LF,i}$ and $N_{HF,i}$ denote the number of low- and high-fidelity samples in the initial training set, respectively. Also, for comparison, we conduct single-fidelity runs which solely base on

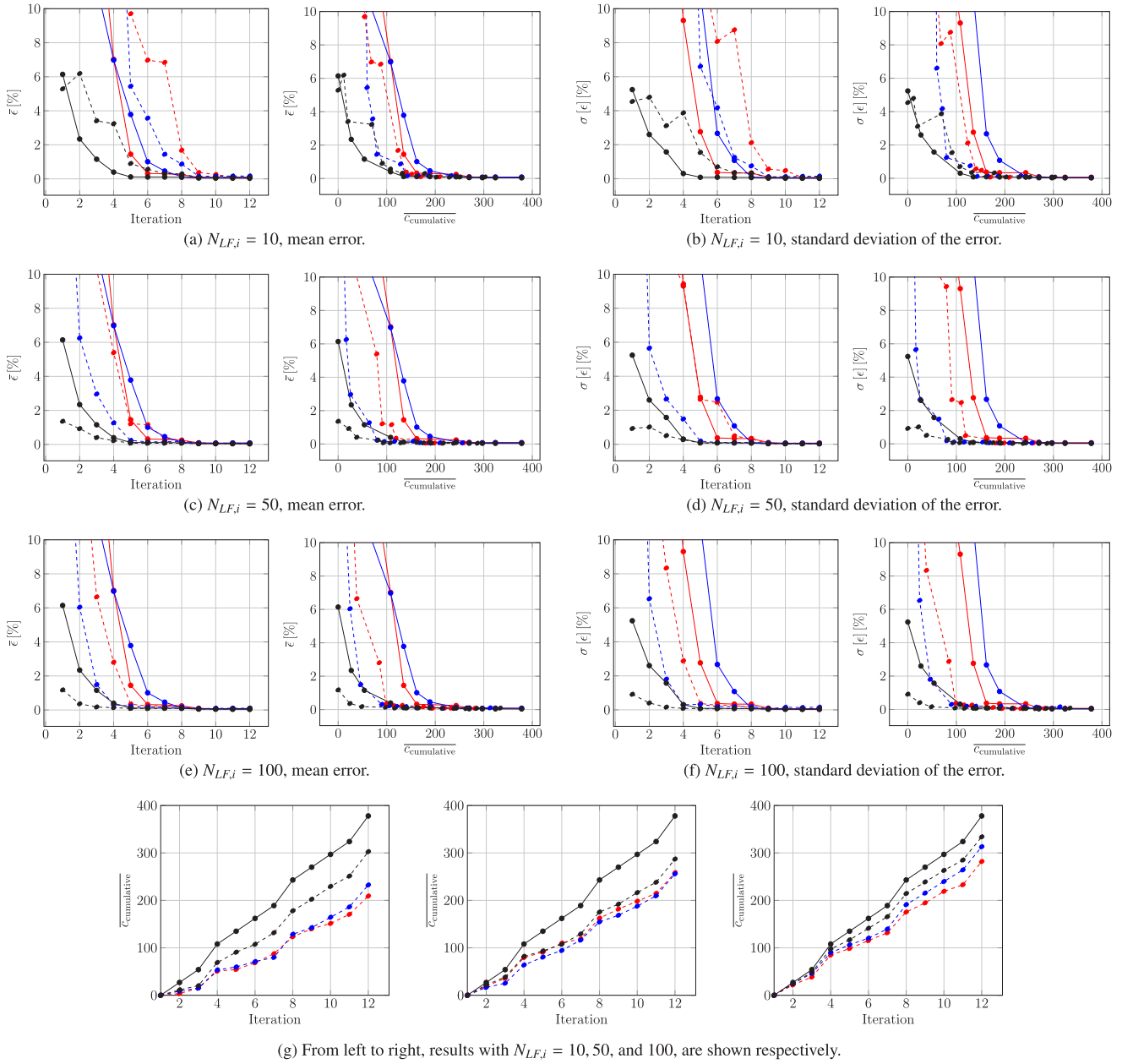


Fig. 7. Convergence of the MFBO algorithm for varying initial training sets and $u_{\Gamma,target} = 0.05$. (a), (c), and (e): We show the mean error $\bar{\epsilon}$ over the number of iterations and the cumulative cost in the left and right part, respectively. (b), (d), and (f): We show the standard deviation of the error $\sigma[\epsilon]$ over the number of iterations and the cumulative cost in the left and right part, respectively. (g): Comparison between the cumulative sample generating cost of single- and multi-fidelity modeling.

In all plots, results with $N_{HF,i} = 2, 5$ and 10 high-fidelity samples in the initial training set are indicated by the red (—), blue (—), and black (—) color, respectively. Single-fidelity results are shown by solid lines (—). Multi-fidelity results are shown by dashed lines (- -). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

high-fidelity data. Therefore, the initial training set only bases on high-fidelity data, and we limit s to the discrete set $s \in 1.0$ to ensure that the acquisition function only proposes high-fidelity points.

In Fig. 7 we solve the inverse problem for $u_{\Gamma,target} = 0.05 \frac{m}{s}$ and show $\overline{\epsilon_{true}}$, $\sigma[\epsilon_{true}]$, and $\overline{c_{cumulative}}$. As already shown in Fig. 5, for this targeted tip velocity a good correlation between the low- and high-fidelity model exists. We investigate the convergence behavior of the algorithm considering $N_{LF,i} = 10, 50$, and 100 low-fidelity samples in the initial training set. We show the evolution of $\overline{\epsilon_{true}}$ in Figs. 7(a), 7(c) and 7(e), respectively, and the evolution of $\sigma[\epsilon_{true}]$ in Figs. 7(b), 7(d) and 7(f), respectively. The mean recommendation error converges for all investigated

initial training sets. Also, the standard deviation of the recommendation error decreases for all configurations. This highlights, that during the early iterations of the optimization algorithm, the solution quality strongly depends on the chosen initial training set. For the later stages of the optimization, the solution quality is independent of the initial training set. For $N_{LF,i} = 10$, cf. Fig. 7(a), the single-fidelity setting converges faster than the high-fidelity setting when monitoring $\overline{\epsilon_{true}}$ over the number of iterations. When considering the evolution of $\overline{\epsilon_{true}}$ over the cumulative cost, the multi-fidelity setting outperforms the single-fidelity setting. For this setup, a similar amount of low- and high-fidelity data is present in the initial training set. Due to the low amount of data present in the initial training set, the GP surrogate model exhibits high predictive uncertainties even for low-fidelity predictions. Thus, the acquisition function often suggests generating low-fidelity samples, which is confirmed when comparing the average cumulative cost for single- and multi-fidelity runs, cf. Fig. 7(g) (left). This leads to the differences when comparing the evolution of single- and multi-fidelity settings either over the number of iterations, or the cumulative cost. Since the low- and high-fidelity model show a good correlation for the investigated target tip velocity, the generated low-fidelity samples contribute significantly to the quality of the GP surrogate, and thus to $\overline{\epsilon_{true}}$. For $N_{LF,i} = 50$, and 100, cf. Figs. 7(c) and 7(e), multi-fidelity modeling outperforms single-fidelity modeling for all considered initial training sets. Also, no severe differences can be observed between monitoring $\overline{\epsilon_{true}}$ over the number of iterations or the cumulative cost since the parameter space is sufficiently dense sampled for both $N_{LF,i}$, and during optimization less focus is put on generating low-fidelity samples. For $\sigma[\epsilon_{true}]$, cf. Figs. 7(d) and 7(f), the same trends can be observed.

In Fig. 8 we solve the inverse problem for $u_{\Gamma,target} = 0.1 \frac{m}{s}$. Fig. 5 shows, that for the targeted tip velocity the correlation between the low- and high-fidelity model is worse compared to the previous case. We investigate the convergence behavior of the algorithm considering $N_{LF,i} = 10, 50$, and 100 low-fidelity samples in the initial training set. We show the evolution of $\overline{\epsilon_{true}}$ in Figs. 8(a), 8(c) and 8(e), respectively, and the evolution of $\sigma[\epsilon_{true}]$ in Figs. 8(b), 8(d) and 8(f), respectively. As for the previously investigated target tip velocity, the mean recommendation error converges for all investigated configurations. Again, we observe that the sensitivity of the result on the initial training set decreases during later stages of the iterative optimization process, when considering the evolution of $\sigma[\epsilon_{true}]$. Due to the bad correlation between low- and high-fidelity model for $u_{\Gamma,target} = 0.1 \frac{m}{s}$, multi-fidelity modeling does not outperform single-fidelity modeling here. Nevertheless, we observe convergence for both settings and achieve relevant recommendation errors $\overline{\epsilon_{true}} < 1\%$ for the same cumulative cost. When adding more low-fidelity samples to the initial training set, the differences between multi- and single-fidelity modeling decrease when monitoring the evolution over the cumulative cost, cf. Fig. 8(g).

The convergence study of the inverse dendritic growth problem suggests several trends that require consideration when defining a multi-fidelity BO setting. Multi-fidelity modeling is especially useful and drastically improves the convergence of the algorithm, when low- and high-fidelity model correlate well for the targeted value. Also, for successful multi-fidelity modeling it is necessary that more low- than high-fidelity data are present. These findings agree with [78]. In case the low- and high-fidelity model do not correlate well for the target value, multi-fidelity modeling does not improve the optimization process compared to single-fidelity modeling. Nevertheless, the multi-fidelity setting does not harm the convergence and achieves relevant recommendation errors $< 1\%$ with a similar computational cost as the single-fidelity setup.

In the Appendix, we solve the inverse problem for $u_{\Gamma,target} = 0.02 \frac{m}{s}$ and show similar plots as for the previous targeted tip velocities. For this tip velocity, the low- and high-fidelity model correlate well, cf. Fig. 5, even better than for $u_{\Gamma,target} = 0.05 \frac{m}{s}$. The trend observed is confirmed, and the multi-fidelity modeling strongly outperforms the single-fidelity modeling.

We validate the recommended points of the BO algorithm in Fig. 9. The targeted tip velocity is $u_{\Gamma,target} = 0.05$, and the number of low-fidelity points in the initial data set $N_{LF,i} = 100$. We show the recommendation points generated for all optimization runs for varying number of high-fidelity points in the initial training set $N_{HF,i}$, where the recommendation points are colored by their recommendation error. For $N_{HF,i} = 2$, cf. Fig. 9(a), the recommendation points cover the whole range of recommendations error. In contrast, for $N_{HF,i} = 10$, cf. Fig. 9(b), they focus only cover small recommendation errors. This observation matches with convergence results shown in Fig. 7(e). For $N_{HF,i} = 2$, and 10, the recommendation points focus around certain paths for small recommendation errors (red color). These paths accumulate around $0.7 < St < 0.8$. For ϵ and α_m/α_s , the paths are more widely spread in the parameter space. This indicates that the solution of the inverse problem for this $u_{\Gamma,target}$ is not unique. Also, it shows that $u_{\Gamma,target}$ is most sensitive to St . These trends agree with the observations already shown in [78].

In Fig. 10 we show the same plots but fix $N_{LF,i} = 100$ and $N_{HF,i} = 10$ while varying the targeted tip velocity $u_{\Gamma,target}$. Again, the recommendation points focus around certain paths. Due to the large number of low-fidelity

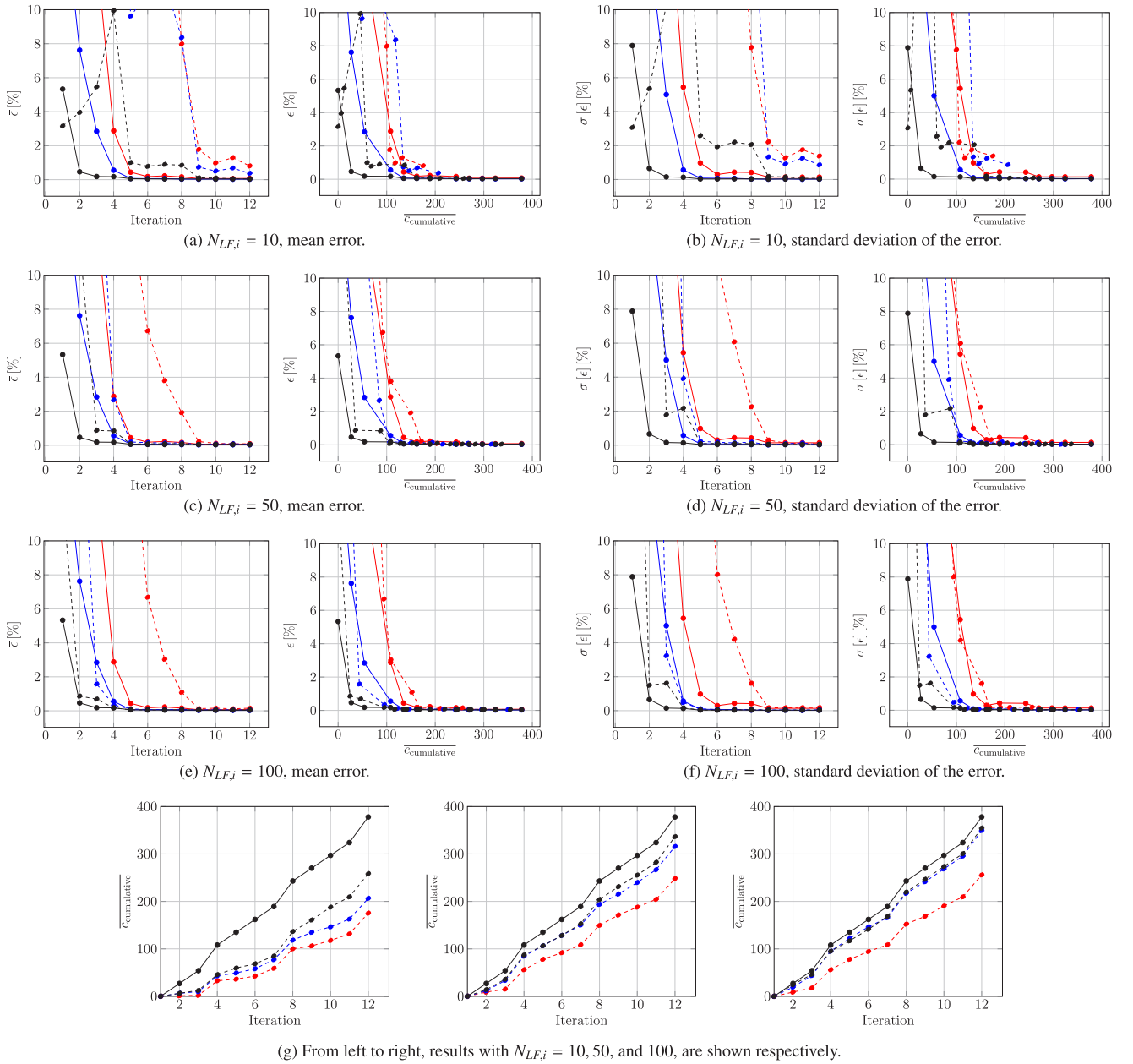


Fig. 8. Convergence of the MFBO algorithm for varying initial training sets and $u_{\Gamma, target} = 0.1$. (a), (c), and (e): We show the mean error $\bar{\epsilon}$ over the number of iterations and the cumulative cost in the left and right part, respectively. (b), (d), and (f): We show the standard deviation of the error $\sigma[\epsilon]$ over the number of iterations and the cumulative cost in the left and right part, respectively. (g): Comparison between the cumulative sample generating cost of single- and multi-fidelity modeling.

In all plots, results with $N_{HF,i} = 2, 5$ and 10 high-fidelity samples in the initial training set are indicated by the red (—), blue (—), and black (—) color, respectively. Single-fidelity results are shown by solid lines (—). Multi-fidelity results are shown by dashed lines (- -). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

samples in the initial training set $N_{LF,i} = 100$, all paths exhibit small recommendation errors. The solution is again most sensitive to St . When increasing $u_{\Gamma, target}$, a higher Stefan number is necessary to obtain the targeted tip velocity. For ϵ and α_m/α_s , the recommendation points are again widely spread over the parameter space.

8. Summary and conclusions

We have developed a novel multi-fidelity Bayesian optimization framework combining composite BO, multi-fidelity modeling using the cost-aware knowledge gradient acquisition function, and a recommendation fitting

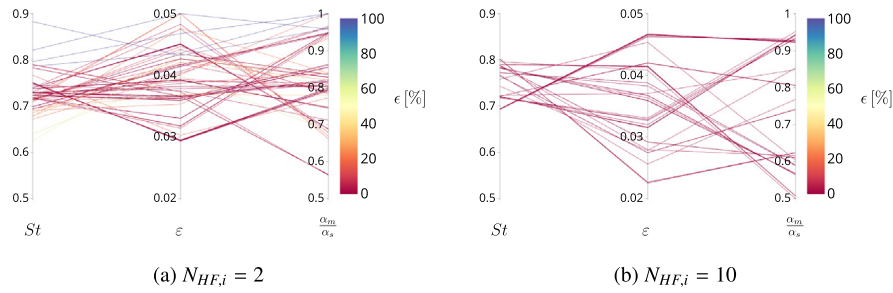


Fig. 9. Parallel coordinates plot for the recommendation points for $u_{\Gamma,target} = 0.05 \frac{m}{s}$, $N_{LF,i} = 100$ and varying $N_{HF,i}$.

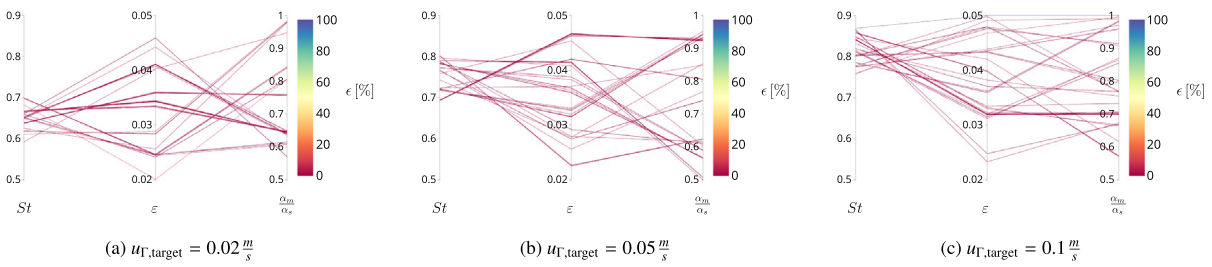


Fig. 10. Parallel coordinates plot for the recommendation points for $N_{LF,i} = 100$, $N_{HF,i} = 10$ and varying $u_{\Gamma,target}$.

approach to solve inverse Stefan problems. We show the convergence properties of the algorithm by considering the inverse problem for a planar solidification front and highlight the benefits of the recommendation-fitting procedure. To solve the more complex inverse problem for directional dendritic solidification we introduce a hierarchy of resolution fidelities. This strongly improves convergence properties and cost-efficiency compared to a single-fidelity setting, especially when access to high-fidelity data is limited. The proposed algorithm is attractive for many applications due to its cost-efficient usage of multi-fidelity data generators and the possibility to reuse existing data samples and pre-trained surrogate models. Since no assumptions on the type of fidelity levels is made, the algorithm can be easily extended to also include data from other sources, like experiments. Perspectives for future work are to apply the algorithm in the rapid and efficient design of new materials with targeted properties.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The first, fourth, and fifth author acknowledge funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 667483). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer SuperMUC-NG at Leibniz Supercomputing Centre (www.lrz.de).

Appendix. Additional results to analyze the bo convergence

Fig. A.11 shows additional convergence plots for the inverse dendritic growth problem at $u_{\Gamma,target} = 0.02 \frac{m}{s}$. Note, for further description or the setup of the case and the graphs we refer to Section 7.

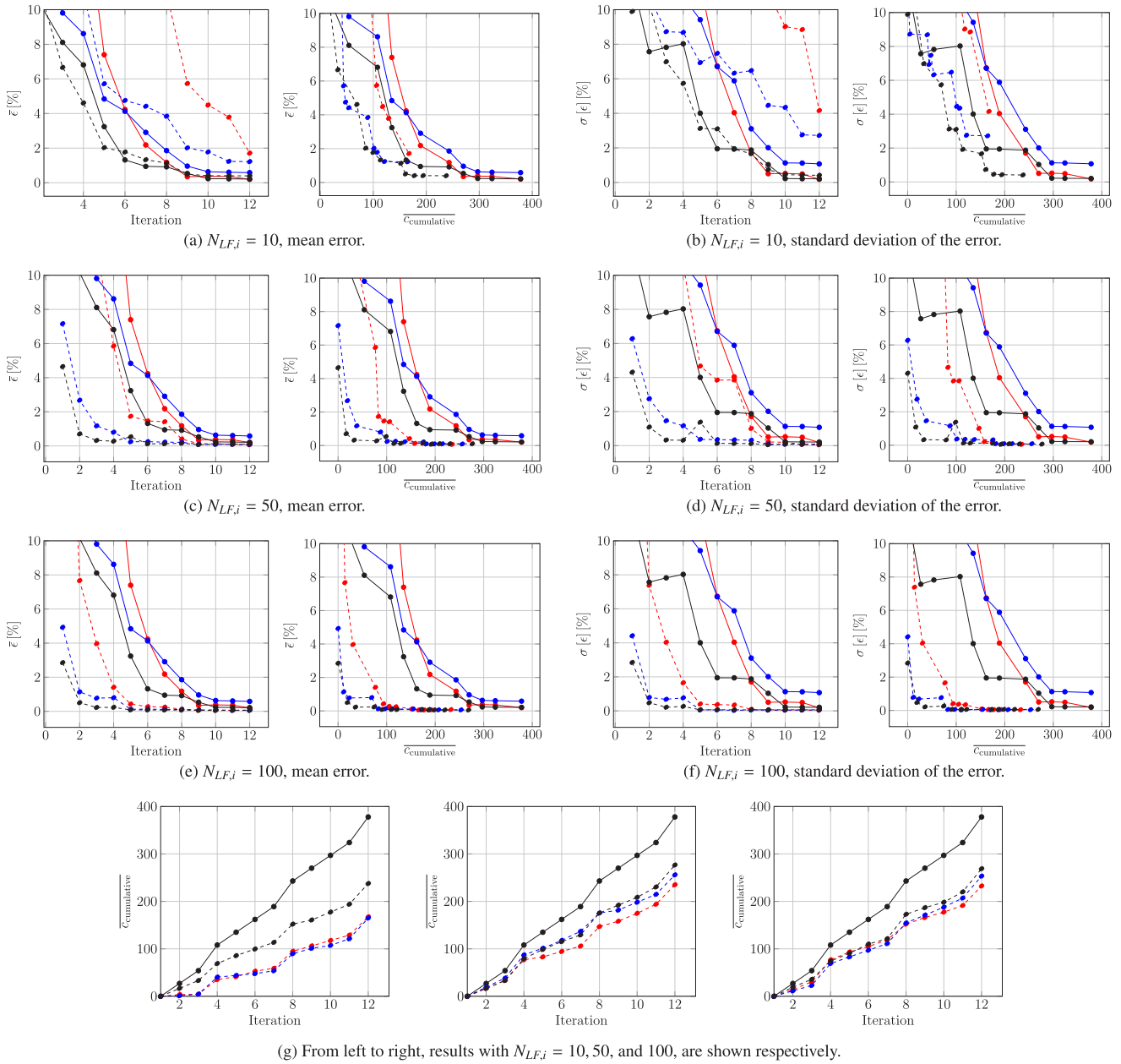


Fig. A.11. Convergence of the MFBO algorithm for varying initial training sets and $u_{\Gamma,target} = 0.02$. (a), (c), and (e): We show the mean error $\bar{\epsilon}$ over the number of iterations and the cumulative cost in the left and right part, respectively. (b), (d), and (f): We show the standard deviation of the error $\sigma[\epsilon]$ over the number of iterations and the cumulative cost in the left and right part, respectively. (g): Comparison between the cumulative sample generating cost of single- and multi-fidelity modeling.

In all plots, results with $N_{HF,i} = 2, 5$ and 10 high-fidelity samples in the initial training set are indicated by the red (—), blue (—), and black (—) color, respectively. Single-fidelity results are shown by solid lines (—). Multi-fidelity results are shown by dashed lines (- -). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

- [1] W. Kurz, D.J. Fisher, R. Trivedi, Progress in modelling solidification microstructures in metals and alloys: dendrites and cells from 1700 to 2000, *Int. Mater. Rev.* 64 (6) (2019) 311–354.
- [2] W. Kurz, M. Rappaz, R. Trivedi, Progress in modelling solidification microstructures in metals and alloys. Part II: dendrites from 2001 to 2018, *Int. Mater. Rev.* 66 (1) (2021) 30–76, <http://dx.doi.org/10.1080/09506608.2020.1757894>.
- [3] J. Stefan, Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere, *Annalen Der Physik* 278 (2) (1891) 269–286.

- [4] G. Caginal, Surface tension and supercooling in solidification theory, in: *Applications of Field Theory To Statistical Mechanics*, Springer, 1985, pp. 216–226.
- [5] A.A. Wheeler, N.A. Ahmad, W.J. Boettinger, R.J. Braun, G.B. Mcfadden, B.T. Murray, Recent developments in phase-field models of solidification, *Adv. Space Res.* 16 (7) (1995) 163–172, [http://dx.doi.org/10.1016/0273-1177\(95\)00153-6](http://dx.doi.org/10.1016/0273-1177(95)00153-6).
- [6] R. Kobayashi, Modeling and numerical simulations of dendritic crystal growth, *Physica D* 63 (3–4) (1993) 410–423, [http://dx.doi.org/10.1016/0167-2789\(93\)90120-p](http://dx.doi.org/10.1016/0167-2789(93)90120-p).
- [7] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving stefan problems, *J. Comput. Phys.* 135 (1) (1997) 8–29, <http://dx.doi.org/10.1006/jcph.1997.5721>.
- [8] E. Attar, C. Körner, Lattice Boltzmann model for thermal free surface flows with liquid–solid phase transition, *Int. J. Heat Fluid Flow* 32 (1) (2011) 156–163.
- [9] V.E. Küng, R. Scherr, M. Markl, C. Körner, Multi-material model for the simulation of powder bed fusion additive manufacturing, *Comput. Mater. Sci.* 194 (2021) 110415.
- [10] Z. Pan, J.A. Weibel, S.V. Garimella, A saturated-interface-volume phase change model for simulating flow boiling, *Int. J. Heat Mass Transfer* 93 (2016) 945–956.
- [11] V. Fung, J. Zhang, G. Hu, P. Ganesh, B.G. Sumpter, Inverse design of two-dimensional materials with invertible neural networks, 2021, [arXiv:2106.03013](https://arxiv.org/abs/2106.03013).
- [12] S.B. Sinnott, Material design and discovery with computational materials science, 1970, URL <https://avs.scitation.org/doi/10.1116/1.4813689>.
- [13] E. Çadırlı, E. Nergiz, H. Kaya, U. Büyük, M. Şahin, M. Gündüz, Effect of growth velocity on microstructure and mechanical properties of directionally solidified 7075 alloy, *Int. J. Cast Metals Res.* 33 (1) (2020) 11–23, <http://dx.doi.org/10.1080/13640461.2020.1738131>.
- [14] P. Jochum, The inverse Stefan problem as a problem of nonlinear approximation theory, *J. Approx. Theory* 30 (2) (1980) 81–98.
- [15] P. Jochum, The numerical solution of the inverse Stefan problem, *Numer. Math.* 34 (4) (1980) 411–429.
- [16] A.C. Briozzo, M.F. Natale, D.A. Tarzia, Determination of unknown thermal coefficients for Storm’s-type materials through a phase-change process, *Int. J. Non-Linear Mech.* 34 (2) (1999) 329–340.
- [17] M.B. Stampella, D.A. Tarzia, Determination of one or two unknown thermal coefficients of a semi-infinite material through a two-phase Stefan problem, *Internat. J. Engrg. Sci.* 27 (11) (1989) 1407–1419.
- [18] D. Ang, A.P.N. Dinh, D. Thanh, Regularization of an inverse Stefan problem, *Differential Integral Equations* 9 (2) (1996) 371–380.
- [19] D.A. Murio, *The Mollification Method and the Numerical Solution of Ill-Posed Problems*, John Wiley & Sons, 2011.
- [20] N. Zabararas, K. Yuan, Dynamic programming approach to the inverse Stefan design problem, *Numer. Heat Transfer* 26 (1) (1994) 97–104.
- [21] N. Zabararas, T.H. Nguyen, Control of the freezing interface morphology in solidification processes in the presence of natural convection, *Internat. J. Numer. Methods Engrg.* 38 (9) (1995) 1555–1578.
- [22] V.R. Voller, Enthalpy method for inverse Stefan problems, *Numer. Heat Transfer B* 21 (1) (1992) 41–55.
- [23] R. Grzymkowski, D. Słota, An application of the Adomian decomposition method for inverse Stefan problem with Neumann’s boundary condition, in: *International Conference on Computational Science*, Springer, 2005, pp. 895–898.
- [24] R. Grzymkowski, D. Słota, One-phase inverse Stefan problem solved by Adomian decomposition method, *Comput. Math. Appl.* 51 (1) (2006) 33–40.
- [25] D. Słota, Solving the inverse Stefan design problem using genetic algorithms, *Inverse Problems Sci. Eng.* 16 (7) (2008) 829–846.
- [26] C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge Massachusetts, USA, 2006.
- [27] A. Forrester, A. Sobester, A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, 2008.
- [28] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, vol. 2, Springer, New York, NY, 2009, <http://dx.doi.org/10.1007/978-0-387-84858-7>.
- [29] S.H. Bryngelson, A. Charalampopoulos, T.P. Sapsis, T. Colonius, A Gaussian moment method and its augmentation via LSTM recurrent neural networks for the statistics of cavitating bubble populations, *Int. J. Multiph. Flow.* 127 (2020) 103262.
- [30] H. Zhang, J.P. Choi, S.K. Moon, T.H. Ngo, A knowledge transfer framework to support rapid process modeling in aerosol jet printing, *Adv. Eng. Inform.* 48 (2021) 101264.
- [31] B. Rankouhi, S. Jahani, F. Pfefferkorn, D. Thoma, Compositional grading of a 316L-Cu multi-material part using machine learning for the determination of selective laser melting process parameters, *Addit. Manuf.* 38 (2021) <http://dx.doi.org/10.1016/j.addma.2021.101836>.
- [32] H.A. Kumar, S. Kumaraguru, C. Paul, K. Bindra, Faster temperature prediction in the powder bed fusion process through the development of a surrogate model, *Opt. Laser Technol.* 141 (2021) 107122.
- [33] R. Saunders, C. Butler, J. Michopoulos, D. Lagoudas, A. Elwany, A. Bagchi, Mechanical behavior predictions of additively manufactured microstructures using functional Gaussian process surrogates, *Npj Comput. Mater.* 7 (1) (2021) 1–11.
- [34] C. Hu, W. Hau, W. Chen, Q.-H. Qin, The fabrication of long carbon fiber reinforced polylactic acid composites via fused deposition modelling: Experimental analysis and machine learning, *J. Compos. Mater.* 55 (11) (2021) 1459–1472, <http://dx.doi.org/10.1177/0021998320972172>.
- [35] E. Mendoza Jimenez, D. Ding, L. Su, A. Joshi, A. Singh, B. Reeja-Jayan, J. Beuth, Parametric analysis to quantify process input influence on the printed densities of binder jetted alumina ceramics, *Addit. Manuf.* 30 (2019) <http://dx.doi.org/10.1016/j.addma.2019.100864>.
- [36] Y. Yabansu, V. Rehn, J. Hötzer, B. Nestler, S. Kalidindi, Application of Gaussian process autoregressive models for capturing the time evolution of microstructure statistics from phase-field simulations for sintering of polycrystalline ceramics, *Modelling Simul. Mater. Sci. Eng.* 27 (8) (2019) <http://dx.doi.org/10.1088/1361-651X/ab413e>.
- [37] J. Snoek, K. Swersky, R. Zemel, R. Adams, Input warping for bayesian optimization of non-stationary functions, in: *International Conference on Machine Learning*, PMLR, 2014, pp. 1674–1682.

- [38] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization, *Proc. IEEE* 104 (1) (2015) 148–175.
- [39] P.I. Frazier, Bayesian optimization, *Recent Adv. Optim. Model. Contemporary Problems* (2018) 255–278, <http://dx.doi.org/10.1287/educ.2018.0188>.
- [40] A.E. Wiens, A.V. Copan, H.F. Schaefer, Multi-fidelity Gaussian process modeling for chemical energy surfaces, *Chem. Phys. Lett. X* 3 (2019) <http://dx.doi.org/10.1016/j.cpletx.2019.100022>.
- [41] R. Astudillo, P. Frazier, Bayesian optimization of composite functions, in: *International Conference on Machine Learning*, PMLR, 2019, pp. 354–363.
- [42] S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.* 22 (10) (2010) 1345–1359, <http://dx.doi.org/10.1109/tkde.2009.191>.
- [43] P. Perdikaris, M. Raissi, A. Damianou, N.D. Lawrence, G.E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, *Proc. R. Soc. A* 473 (2198) (2017) 20160751.
- [44] H. Babae, P. Perdikaris, C. Chrysostomidis, G. Karniadakis, Multi-fidelity modelling of mixed convection based on experimental correlations and numerical simulations, *J. Fluid Mech.* 809 (2016) 895.
- [45] L. Meng, J. Zhang, Process design of laser powder bed fusion of stainless steel using a Gaussian process-based machine learning model, *JOM* 72 (1) (2020) 420–428.
- [46] G. Kissas, Y. Yang, E. Hwuang, W.R. Witschey, J.A. Detre, P. Perdikaris, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks, *Comput. Methods Appl. Mech. Engrg.* 358 (2020) 112623.
- [47] C.M. Fleeter, G. Geraci, D.E. Schiavazzi, A.M. Kahn, A.L. Marsden, Multilevel and multifidelity uncertainty quantification for cardiovascular hemodynamics, *Comput. Methods Appl. Mech. Engrg.* 365 (2020) 113030, <http://dx.doi.org/10.1016/j.cma.2020.113030>.
- [48] X. Meng, Z. Wang, D. Fan, M. Triantafyllou, G.E. Karniadakis, A fast multi-fidelity method with uncertainty quantification for complex data correlations: Application to vortex-induced vibrations of marine risers, 2020, [arXiv:2012.13481](https://arxiv.org/abs/2012.13481).
- [49] A. Doostan, G. Geraci, G. Iaccarino, A bi-fidelity approach for uncertainty quantification of heat transfer in a rectangular ribbed channel, in: *Turbo Expo: Power for Land, Sea, and Air*, vol. 49712, American Society of Mechanical Engineers, 2016, V02CT45A031.
- [50] I.-G. Farcaş, P.C. Sârbu, H.-J. Bungartz, T. Neckel, B. Uekermann, Multilevel adaptive stochastic collocation with dimensionality reduction, in: J. Garcke, D. Pflüger, C.G. Webster, G. Zhang (Eds.), *Sparse Grids and Applications - Miami 2016*, Springer International Publishing, Cham, 2018, pp. 43–68.
- [51] A. Beck, J. Dürrwächter, T. Kuhn, F. Meyer, C.-D. Munz, C. Rohde, Hp-multilevel Monte Carlo methods for uncertainty quantification of compressible Navier–Stokes equations, *SIAM J. Sci. Comput.* 42 (4) (2020) B1067–B1091.
- [52] I.-G. Farcaş, J. Latz, E. Ullmann, T. Neckel, H.-J. Bungartz, Multilevel adaptive sparse Leja approximations for Bayesian inverse problems, *SIAM J. Sci. Comput.* 42 (1) (2020) A424–A451.
- [53] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, *Siam Rev.* 60 (3) (2018) 550–591.
- [54] D. Huang, T.T. Allen, W.I. Notz, R.A. Miller, Sequential kriging optimization using multiple-fidelity evaluations, *Struct. Multidiscip. Optim.* 32 (5) (2006) 369–382, <http://dx.doi.org/10.1007/s00158-005-0587-0>.
- [55] R. Lam, D.L. Allaire, K.E. Willcox, Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources, in: *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, <http://dx.doi.org/10.2514/6.2015-0143>.
- [56] V. Picheny, D. Ginsbourger, Y. Richet, G. Caplin, Quantile-based optimization of noisy computer experiments with tunable precision, *Technometrics* 55 (1) (2012) 2–13, <http://dx.doi.org/10.1080/00401706.2012.707580>.
- [57] K. Kandasamy, G. Dasarathy, J. Oliva, J. Schneider, B. Póczos, Multi-fidelity Gaussian process bandit optimisation, *J. Artificial Intelligence Res.* 66 (2019) 151–196, <http://dx.doi.org/10.1613/jair.1.11288>.
- [58] N. Srinivas, A. Krause, S.M. Kakade, M.W. Seeger, Information-theoretic regret bounds for Gaussian process optimization in the bandit setting, *IEEE Trans. Inform. Theory* 58 (5) (2012) 3250–3265, <http://dx.doi.org/10.1109/tit.2011.2182033>.
- [59] J. Wu, S. Toscano-Palmerin, P.I. Frazier, A.G. Wilson, Practical multi-fidelity Bayesian optimization for hyperparameter tuning, 2019, [arXiv:1903.04703](https://arxiv.org/abs/1903.04703).
- [60] J.R. Gardner, G. Pleiss, D. Bindel, K.Q. Weinberger, A.G. Wilson, GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration, in: *Advances in Neural Information Processing Systems*, 2018.
- [61] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A.G. Wilson, E. Bakshy, BoTorch: a framework for efficient Monte-Carlo Bayesian optimization, *Adv. Neural Inf. Process. Syst.* 33 (2020) 21524–21538.
- [62] N. Hoppe, J.M. Winter, S. Adami, N.A. Adams, ALPACA - A level-set based sharp-interface multiresolution solver for conservation laws, *Comput. Phys. Comm.* (2021) 108246, <http://dx.doi.org/10.1016/j.cpc.2021.108246>.
- [63] S.S. Gupta, K.J. Miescke, Bayesian look ahead one-stage sampling allocations for selection of the best population, *J. Statist. Plann. Inference* 54 (2) (1996) 229–244.
- [64] P.I. Frazier, W.B. Powell, S. Dayanik, A knowledge-gradient policy for sequential information collection, *SIAM J. Control Optim.* 47 (5) (2008) 2410–2439.
- [65] P. Frazier, W. Powell, S. Dayanik, The knowledge-gradient policy for correlated normal beliefs, *INFORMS J. Comput.* 21 (4) (2009) 599–613.
- [66] J. Wu, P. Frazier, The parallel knowledge gradient method for batch Bayesian optimization, *Adv. Neural Inf. Process. Syst.* 29 (2016) 3126–3134.
- [67] K. Kandasamy, K.R. Vysyaraju, W. Neiswanger, B. Paria, C.R. Collins, J. Schneider, B. Póczos, E.P. Xing, Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with dragonfly, *J. Mach. Learn. Res.* 21 (81) (2020) 1–27.

- [68] J. Wu, P.I. Frazier, Continuous-fidelity Bayesian optimization with knowledge gradient, in: Workshop on Bayesian Optimization, Neural Information Processing Systems (NIPS), 2017.
- [69] K. Swersky, J. Snoek, R.P. Adams, Freeze-thaw Bayesian optimization, 2014, arXiv preprint [arXiv:1406.3896](https://arxiv.org/abs/1406.3896).
- [70] P. Rauschenberger, A. Criscione, K. Eisenschmidt, D. Kintea, S. Jakirlić, Ž. Tuković, I. Roisman, B. Weigand, C. Tropea, Comparative assessment of volume-of-fluid and level-set methods by relevance to dendritic ice growth in supercooled water, *Comput. & Fluids* 79 (2013) 44–52.
- [71] H.S. Carslaw, J.C. Jaeger, *Conduction of Heat in Solids*, Technical Report, Clarendon Press, 1959.
- [72] A. Karma, W.-J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, *Phys. Rev. E* 57 (4) (1998) 4323.
- [73] Y.-T. Kim, N. Goldenfeld, J. Dantzig, Computation of dendritic microstructures using a level set method, *Phys. Rev. E* 62 (2) (2000) 2471.
- [74] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, *Phys. Rev. Lett.* 80 (1998) 3308–3311, <http://dx.doi.org/10.1103/PhysRevLett.80.3308>.
- [75] L. Tan, N. Zabaras, A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods, *J. Comput. Phys.* 211 (2006) 36–63, <http://dx.doi.org/10.1016/j.jcp.2005.05.013>.
- [76] H.S. Udaykumar, L. Mao, R. Mittal, A finite-volume sharp interface scheme for dendritic growth simulations: Comparison with microscopic solvability theory, *Numer. Heat Transfer B* 42 (5) (2002) 389–409, <http://dx.doi.org/10.1080/10407790190054003>.
- [77] V. Ramanuj, R. Sankaran, B. Radhakrishnan, A sharp interface model for deterministic simulation of dendrite growth, *Comput. Mater. Sci.* 169 (2019) 109097, <http://dx.doi.org/10.1016/j.commatsci.2019.109097>.
- [78] J. Winter, J. Kaiser, S. Adami, I. Akhatov, N. Adams, Stochastic multi-fidelity surrogate modeling of dendritic crystal growth, *Comput. Methods Appl. Mech. Engrg.* 393 (2022) 114799, <http://dx.doi.org/10.1016/j.cma.2022.114799>.
- [79] J. Kaiser, S. Adami, I. Akhatov, N. Adams, A semi-implicit conservative sharp-interface method for liquid-solid phase transition, *Int. J. Heat Mass Transfer* 155 (2020) 119800.
- [80] N. Hoppe, S. Adami, N.A. Adams, A parallel modular computing environment for three-dimensional multiresolution simulations of compressible flows, *Comput. Methods Appl. Mech. Engrg.* 391 (2022) 114486, <http://dx.doi.org/10.1016/j.cma.2021.114486>.
- [81] J.W.J. Kaiser, N. Hoppe, S. Adami, N.A. Adams, An adaptive local time-stepping scheme for multiresolution simulations of hyperbolic conservation laws, *J. Comput. Phys.* X 4 (2019) 100038.
- [82] A. Harten, Adaptive multiresolution schemes for shock computations, *J. Comput. Phys.* 115 (1994) 319–338.

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.
- [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] Alexandrov, D. V. and Galenko, P. K. “A review on the theory of stable dendritic growth”. In: *Philosophical Transactions of the Royal Society A* 379.2205 (2021), p. 20200325.
- [5] 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.
- [6] Amar, M. B. and Pomeau, Y. “Theory of dendritic growth in a weakly undercooled melt”. In: *EPL (Europhysics Letters)* 2.4 (1986), p. 307.
- [7] Ang, D., Dinh, A. P. N., and Thanh, D. “Regularization of an inverse Stefan problem”. In: *Differential and Integral Equations* 9.2 (1996), pp. 371–380.
- [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] Appleton, J. P. and Bray, K. N. C. “The conservation equations for a non-equilibrium plasma”. In: *Journal of Fluid Mechanics* 20.4 (1964), pp. 659–672. DOI: 10.1017/S0022112064001458.
- [10] Babaei, H., Perdikaris, P., Chrysostomidis, C., and Karniadakis, G. “Multi-fidelity modelling of mixed convection based on experimental correlations and numerical simulations”. In: *Journal of Fluid Mechanics* 809 (2016), p. 895.
- [11] Bakshy, E., Dworkin, L., Karrer, B., Kashin, K., Letham, B., Murthy, A., and Singh, S. “AE: A domain-agnostic platform for adaptive experimentation”. In: *Conference on neural information processing systems*. 2018, pp. 1–8.
- [12] Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. “BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization”. In: *Advances in Neural Information Processing Systems* 33. 2020. URL: <http://arxiv.org/abs/1910.06403>.
- [13] 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.
- [14] Barbieri, A., Hong, D. C., and Langer, J. “Velocity selection in the symmetric model of dendritic crystal growth”. In: *Physical Review A* 35.4 (1987), p. 1802.

- [15] Beck, A., Dürrwächter, J., Kuhn, T., Meyer, F., Munz, C.-D., and Rohde, C. “hp-Multilevel Monte Carlo Methods for Uncertainty Quantification of Compressible Navier–Stokes Equations”. In: *SIAM Journal on Scientific Computing* 42.4 (2020), B1067–B1091.
- [16] Ben-Jacob, E., Goldenfeld, N., Langer, J., and Schön, G. “Dynamics of interfacial pattern formation”. In: *Physical Review Letters* 51.21 (1983), p. 1930.
- [17] Bendahmane, M., Bürger, R., Ruiz-Baier, R., and Schneider, K. “Adaptive multiresolution schemes with local time stepping for two-dimensional degenerate reaction–diffusion systems”. In: *Applied numerical mathematics* 59.7 (2009), pp. 1668–1692.
- [18] Bensimon, D., Pelcé, P., and Shraiman, B. I. “Dynamics of curved fronts and pattern selection”. In: *Journal de Physique* 48.12 (1987), pp. 2081–2087.
- [19] Berger, M. J. and Colella, P. “Local Adaptive Mesh Refinement for Shock Hydrodynamics”. In: *J. Comput. Phys.* 82 (1989), pp. 64–84.
- [20] Bogdanov, V., Schraner, 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: <https://doi.org/10.1016/j.jcp.2022.111445>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999122005071>.
- [21] Brackbill, J. U., Kothe, D. B., and Zemach, C. “A continuum method for modeling surface tension”. In: *Journal of computational physics* 100.2 (1992), pp. 335–354.
- [22] Brennen, C. E. *Fundamentals of Multiphase Flow*. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511807169.
- [23] Briozzo, A. C., Natale, M. F., and Tarzia, D. A. “Determination of unknown thermal coefficients for Storm’s-type materials through a phase-change process”. In: *International journal of non-linear mechanics* 34.2 (1999), pp. 329–340.
- [24] Brower, R. C., Kessler, D. A., Koplik, J., and Levine, H. “Geometrical approach to moving-interface dynamics”. In: *Physical review letters* 51.13 (1983), p. 1111.
- [25] Bryngelson, S. H., Charalampopoulos, A., Sapsis, T. P., and Colonius, T. “A Gaussian moment method and its augmentation via LSTM recurrent neural networks for the statistics of cavitating bubble populations”. In: *International Journal of Multiphase Flow* 127 (2020), p. 103262.
- [26] 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.
- [27] Chandrasekhar, S. *Hydrodynamic and hydromagnetic stability*. 1961. ISBN: 0-486-64071-X.
- [28] Chang, C.-H., Deng, X., and Theofanous, T. G. “Direct numerical simulation of interfacial instabilities: a consistent, conservative, all-speed, sharp-interface method”. In: *Journal of Computational Physics* 242 (2013), pp. 946–990.
- [29] Cheng, K. and Seki, N. *Freezing and melting heat transfer in engineering: selected topics on ice-water systems and welding and casting processes*. CRC Press, 1991.
- [30] Chopp, D. “Computing minimal surfaces via level set curvature flow”. In: *Journal of Computational Physics* 106.1 (1993). cited By 357, pp. 77–91. DOI: 10.1006/jcph.1993.1092.
- [31] Costabal, F. S., Perdikaris, P., Kuhl, E., and Hurtado, D. E. “Multi-fidelity classification using Gaussian processes: accelerating the prediction of large-scale computational models”. In: *Computer Methods in Applied Mechanics and Engineering* 357 (2019), p. 112602.
- [32] Cubaud, T. and Mason, T. G. “Interacting viscous instabilities in microfluidic systems”. In: *Soft Matter* 8 (41 2012), pp. 10573–10582. DOI: 10.1039/C2SM25902H. URL: <http://dx.doi.org/10.1039/C2SM25902H>.

- [33] Cummins, S. J., Francois, M. M., and Kothe, D. B. "Estimating curvature from volume fractions". In: *Computers & structures* 83.6-7 (2005), pp. 425–434.
- [34] Curtarolo, S., Hart, G. L., Nardelli, M. B., Mingo, N., Sanvito, S., and Levy, O. "The high-throughput highway to computational materials design". In: *Nature materials* 12.3 (2013), pp. 191–201.
- [35] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., and González, J. *Deep Gaussian Processes for Multi-fidelity Modeling*. 2019. arXiv: 1903.07320 [stat.ML].
- [36] Damianou, A. "Deep Gaussian processes and variational propagation of uncertainty". PhD thesis. University of Sheffield, 2015.
- [37] Damianou, A. and Lawrence, N. D. "Deep gaussian processes". In: *Artificial intelligence and statistics*. PMLR. 2013, pp. 207–215.
- [38] 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.
- [39] 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.
- [40] Desautels, T., Krause, A., and Burdick, J. W. "Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization". In: *Journal of Machine Learning Research* 15 (2014), pp. 3873–3923.
- [41] 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.
- [42] Domingues, M. O., Gomes, S. M., Roussel, O., and Schneider, K. "An adaptive multiresolution scheme with local time stepping for evolutionary PDEs". en. In: *Journal of Computational Physics* 227.8 (Apr. 2008), pp. 3758–3780. ISSN: 00219991. DOI: 10.1016/j.jcp.2007.11.046.
- [43] Doostan, A., Geraci, G., and Iaccarino, G. "A bi-fidelity approach for uncertainty quantification of heat transfer in a rectangular ribbed channel". In: *Turbo Expo: Power for Land, Sea, and Air*. Vol. 49712. American Society of Mechanical Engineers. 2016, V02CT45A031.
- [44] Elder, J. F. "Global R/sup d/optimization when probes are expensive: the GROPE algorithm". In: *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 1992, pp. 577–582.
- [45] Farcas, I.-G., Latz, J., Ullmann, E., Neckel, T., and Bungartz, H.-J. "Multilevel adaptive sparse Leja approximations for Bayesian inverse problems". In: *SIAM Journal on Scientific Computing* 42.1 (2020), A424–A451.
- [46] Farcaş, I.-G., Sârbu, P. C., Bungartz, H.-J., Neckel, T., and Uekermann, B. "Multilevel Adaptive Stochastic Collocation with Dimensionality Reduction". In: *Sparse Grids and Applications - Miami 2016*. Ed. by Garcke, J., Pflüger, D., Webster, C. G., and Zhang, G. Cham: Springer International Publishing, 2018, pp. 43–68. ISBN: 978-3-319-75426-0.
- [47] Fedkiw, R. P., Aslam, T., Merriman, B., Osher, S., et al. "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.
- [48] Feng, Y., Schraner, F., Winter, J., and Adams, N. "Bayesian Optimization on Fifth-Order Targeted ENO Scheme for Compressible Flows". en. In: *15th World Congress on Computational Mechanics (WCCM-XV) and 8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII)*. CIMNE, 2022. DOI: 10.23967/wccm-apcom.2022.024.

- [49] Feng, Y., Schraner, F. S., Winter, J., and Adams, N. A. "A multi-objective Bayesian optimization environment for systematic design of numerical schemes for compressible flow". en. In: *Journal of Computational Physics* 468 (2022), p. 111477. DOI: 10.1016/j.jcp.2022.111477.
- [50] Feng, Y., Schraner, F. S., Winter, J., and Adams, N. A. "A deep reinforcement learning framework for dynamic optimization of numerical schemes for compressible flow simulations". en. In: *Journal of Computational Physics* 493 (2023), p. 112436. DOI: 10.1016/j.jcp.2023.112436.
- [51] 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*. 2017, pp. 1–12.
- [52] Fleeter, C. M., Geraci, G., Schiavazzi, D. E., Kahn, A. M., and Marsden, A. L. "Multilevel and multifidelity uncertainty quantification for cardiovascular hemodynamics". In: *Computer Methods in Applied Mechanics and Engineering* 365 (2020), p. 113030. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2020.113030>.
- [53] 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.
- [54] 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.
- [55] Fleischmann, N., Adami, S., Hu, X. Y., and Adams, N. A. "A low dissipation method to cure the grid-aligned shock instability". en. In: *Journal of Computational Physics* 401 (2020), p. 109004. DOI: 10.1016/j.jcp.2019.109004.
- [56] Fleischmann, N., Winter, J. M., Adami, S., and Adams, N. A. "High-order modeling of interface interactions using level sets". In: *GAMM-Mitteilungen* 45.2 (2022), e202200012. DOI: <https://doi.org/10.1002/gamm.202200012>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/gamm.202200012>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.202200012>.
- [57] Fletcher, S. and Ponnambalam, K. "Estimation of reservoir yield and storage distribution using moments analysis". In: *Journal of hydrology* 182.1-4 (1996), pp. 259–275.
- [58] Forrester, A., Sobester, A., and Keane, A. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [59] Frazier, P., Powell, W., and Dayanik, S. "The knowledge-gradient policy for correlated normal beliefs". In: *INFORMS journal on Computing* 21.4 (2009), pp. 599–613.
- [60] Frazier, P. I., Powell, W. B., and Dayanik, S. "A knowledge-gradient policy for sequential information collection". In: *SIAM Journal on Control and Optimization* 47.5 (2008), pp. 2410–2439.
- [61] Fritts, D. C. "Shear excitation of atmospheric gravity waves". In: *Journal of Atmospheric Sciences* 39.9 (1982), pp. 1936–1952.
- [62] Fu, L., Hu, X. Y., and Adams, N. A. "A family of high-order targeted ENO schemes for compressible-fluid simulations". In: *Journal of Computational Physics* 305 (2016), pp. 333–359. ISSN: 10902716. DOI: 10.1016/j.jcp.2015.10.037.
- [63] Fung, V., Zhang, J., Hu, G., Ganesh, P., and Sumpter, B. G. *Inverse design of two-dimensional materials with invertible neural networks*. 2021. arXiv: 2106.03013 [cond-mat.mtrl-sci].
- [64] Fuster, D. and Popinet, S. "An all-Mach method for the simulation of bubble dynamics problems in the presence of surface tension". In: *Journal of Computational Physics* 374 (2018), pp. 752–768. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.07.055>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999118305187>.

- [65] Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. "GPYtorch: Black-box Matrix-Matrix Gaussian Process Inference with GPU Acceleration". In: *Advances in Neural Information Processing Systems*. 2018.
- [66] 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.
- [67] Gibou, F., Fedkiw, R., Caflich, R., and Osher, S. "A level set approach for the numerical simulation of dendritic growth". In: *Journal of Scientific Computing* 19 (2003), pp. 183–199.
- [68] Glimm, J., Li, X. L., Menikoff, R., Sharp, D., and Zhang, Q. "A numerical study of bubble interactions in Rayleigh–Taylor instability for compressible fluids". In: *Physics of Fluids A: Fluid Dynamics* 2.11 (1990), pp. 2046–2054.
- [69] 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.
- [70] 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.
- [71] 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.
- [72] González, J., Osborne, M., and Lawrence, N. "GLASSES: Relieving the myopia of Bayesian optimisation". In: *Artificial Intelligence and Statistics*. PMLR. 2016, pp. 790–799.
- [73] Greenhill, S., Rana, S., Gupta, S., Vellanki, P., and Venkatesh, S. "Bayesian optimization for adaptive experimental design: A review". In: *IEEE access* 8 (2020), pp. 13937–13948.
- [74] Grzymkowski, R. and Słota, D. "One-phase inverse Stefan problem solved by Adomian decomposition method". In: *Computers & Mathematics with Applications* 51.1 (2006), pp. 33–40.
- [75] Grzymkowski, R. and Słota, D. "An application of the Adomian decomposition method for inverse Stefan problem with Neumann's boundary condition". In: *International Conference on Computational Science*. Springer. 2005, pp. 895–898.
- [76] Gupta, S. S. and Miescke, K. J. "Bayesian look ahead one-stage sampling allocations for selection of the best population". In: *Journal of statistical planning and inference* 54.2 (1996), pp. 229–244.
- [77] Han, B.-Z., Huang, W.-X., and Xu, C.-X. "Multi-fidelity Bayesian optimization for spatially distributed control of flow over a circular cylinder". In: *Physics of Fluids* 35.11 (2023).
- [78] 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.
- [79] Han, L., Indinger, T., Hu, X., and Adams, N. A. "Wavelet-based adaptive multi-resolution solver on heterogeneous parallel architecture for computational fluid dynamics". In: *Computer Science-Research and Development* 26 (2011), pp. 197–203.
- [80] 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.
- [81] 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.

- [82] 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.
- [83] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. Vol. 2. Springer, New York, NY, 2009. DOI: doi.org/10.1007/978-0-387-84858-7.
- [84] Hejazialhosseini, B., Rossinelli, D., Bergdorf, M., and Koumoutsakos, P. "High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions". en. In: *Journal of Computational Physics* 229.22 (Nov. 2010), pp. 8364–8383. ISSN: 00219991. DOI: 10.1016/j.jcp.2010.07.021.
- [85] 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.
- [86] Hensman, J., Fusi, N., and Lawrence, N. D. "Gaussian Processes for Big Data". In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*. Bellevue, WA: AUAI Press, 2013, pp. 282–290.
- [87] Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. "Predictive entropy search for efficient global optimization of black-box functions". In: *Advances in neural information processing systems* 27 (2014).
- [88] Hetmaniok, E., Hristov, J., Slota, D., and Zielonka, A. "Identification of the heat transfer coefficient in the two-dimensional model of binary alloy solidification". In: *Heat and Mass Transfer* 53 (2017), pp. 1657–1666.
- [89] Hill, J., Mulholland, G., Persson, K., Seshadri, R., Wolverson, C., and Meredig, B. "Materials science with large-scale data and informatics: Unlocking new opportunities". In: *Mrs Bulletin* 41.5 (2016), pp. 399–409.
- [90] 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.
- [91] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. "Stochastic variational inference." In: *Journal of Machine Learning Research* 14.5 (2013).
- [92] 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". In: *2019 International Conference on High Performance Computing Simulation (HPCS)*. 2019, pp. 732–740. DOI: 10.1109/HPCS48598.2019.9188088.
- [93] 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.
- [94] 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: <https://doi.org/10.1016/j.cpc.2021.108246>. URL: <https://www.sciencedirect.com/science/article/pii/S0010465521003581>.
- [95] Hu, C., Hau, W., Chen, W., and Qin, Q.-H. "The fabrication of long carbon fiber reinforced polylactic acid composites via fused deposition modelling: Experimental analysis and machine learning". In: *Journal of Composite Materials* 55.11 (2021), pp. 1459–1472. DOI: 10.1177/0021998320972172.
- [96] Hu, X. Y., Khoo, B., Adams, N. A., and Huang, F. "A conservative interface method for compressible flows". In: *Journal of Computational Physics* 219.2 (2006), pp. 553–578.

- [97] Hu, X., Adams, N., Johnsen, E., and Iaccarino, G. "Modeling full-Mach-range cavitating flow with sharp interface model". In: *Proceedings of the Summer Program*. 2008, p. 183.
- [98] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. "Sequential kriging optimization using multiple-fidelity evaluations". In: *Structural and Multidisciplinary Optimization* 32.5 (2006), pp. 369–382. DOI: 10.1007/s00158-005-0587-0.
- [99] Huang, S.-C. and Glicksman, M. "Overview 12: Fundamentals of dendritic solidification—I. Steady-state tip growth". In: *Acta Metallurgica* 29.5 (1981), pp. 701–715. ISSN: 0001-6160. DOI: [https://doi.org/10.1016/0001-6160\(81\)90115-2](https://doi.org/10.1016/0001-6160(81)90115-2).
- [100] Ivantsov, G. In: *Doklady Akademii Nauk SSSR* (1947), pp. 558–567.
- [101] Jaafar, M. A., Rousse, D. R., Gibout, S., and Bédécarrats, J.-P. "A review of dendritic growth during solidification: Mathematical modeling and numerical simulations". In: *Renewable and Sustainable Energy Reviews* 74 (2017), pp. 1064–1079.
- [102] Jiang, G. S. and Shu, C. W. "Efficient implementation of weighted ENO schemes". In: *Journal of Computational Physics* 126 (1996), pp. 202–228.
- [103] Jochum, P. "The inverse Stefan problem as a problem of nonlinear approximation theory". In: *Journal of Approximation Theory* 30.2 (1980), pp. 81–98.
- [104] Jochum, P. "The numerical solution of the inverse Stefan problem". In: *Numerische Mathematik* 34.4 (1980), pp. 411–429.
- [105] Jones, D. R., Schonlau, M., and Welch, W. J. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13 (1998), pp. 455–492.
- [106] Juric, D. and Tryggvason, G. "A front-tracking method for dendritic solidification". In: *Journal of computational physics* 123.1 (1996), pp. 127–148.
- [107] Kaiser, J., Adami, S., and Adams, N. "Three-dimensional direct numerical simulation of shock-induced bubble collapse near gelatin". In: 2019.
- [108] 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.
- [109] 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.
- [110] 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.
- [111] 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.
- [112] Kandasamy, K., Dasarathy, G., Oliva, J., Schneider, J., and Póczos, B. "Multi-fidelity Gaussian Process Bandit Optimisation". In: *Journal of Artificial Intelligence Research* 66 (2019), pp. 151–196. DOI: 10.1613/jair.1.11288.
- [113] Karma, A. and Rappel, W.-J. "Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics". In: *Physical review E* 53.4 (1996), R3017.
- [114] Kennedy, M. C. and O'Hagan, A. "Predicting the output from a complex computer code when fast approximations are available". In: *Biometrika* 87.1 (2000), pp. 1–13.

- [115] Kessler, D. A. and Levine, H. "Stability of dendritic crystals". In: *Physical review letters* 57.24 (1986), p. 3069.
- [116] Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., and Perdikaris, P. "Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 358 (2020), p. 112623.
- [117] Kobayashi, R. "Modeling and numerical simulations of dendritic crystal growth". In: *Physica D: Nonlinear Phenomena* 63.3-4 (1993), pp. 410–423.
- [118] Kouraytem, N., Li, X., Tan, W., Kappes, B., and Spear, A. D. "Modeling process–structure–property relationships in metal additive manufacturing: a review on physics-driven versus data-driven approaches". In: *Journal of Physics: Materials* 4.3 (Apr. 2021), p. 032002. DOI: 10.1088/2515-7639/abca7b. URL: <https://dx.doi.org/10.1088/2515-7639/abca7b>.
- [119] Krause, A. and Ong, C. "Contextual gaussian process bandit optimization". In: *Advances in neural information processing systems* 24 (2011).
- [120] Kumar, H. A., Kumaraguru, S., Paul, C., and Bindra, K. "Faster temperature prediction in the powder bed fusion process through the development of a surrogate model". In: *Optics & Laser Technology* 141 (2021), p. 107122.
- [121] Kumaraswamy, P. "A generalized probability density function for double-bounded random processes". In: *Journal of hydrology* 46.1-2 (1980), pp. 79–88.
- [122] Kurz, W., Fisher, D. J., and Trivedi, R. "Progress in modelling solidification microstructures in metals and alloys: dendrites and cells from 1700 to 2000". In: *International Materials Reviews* 64.6 (2019), pp. 311–354.
- [123] Kurz, W., Rappaz, M., and Trivedi, R. "Progress in modelling solidification microstructures in metals and alloys. Part II: dendrites from 2001 to 2018". In: *International Materials Reviews* 66.1 (2021), pp. 30–76.
- [124] Kushner, H. J. "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise". In: *Journal of Basic Engineering* 86 (1964), pp. 97–106. DOI: <https://doi.org/10.1115/1.3653121>. URL: <https://api.semanticscholar.org/CorpusID:62599010>.
- [125] Kutta, W. "Beitrag zur naherungsweise Integration von Differentialgleichungen". In: *Zeit. Math. Physik* 46 (1901), pp. 435–453.
- [126] Lally, N. and Hartman, B. "Estimating loss reserves using hierarchical Bayesian Gaussian process regression with input warping". In: *Insurance: Mathematics and Economics* 82 (2018), pp. 124–140.
- [127] Lam, R., Allaire, D. L., and Willcox, K. E. "Multifidelity Optimization using Statistical Surrogate Modeling for Non-Hierarchical Information Sources". In: *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2015). DOI: 10.2514/6.2015-0143.
- [128] Langer, J. S. and Müller-Krumbhaar, J. "Stability effects in dendritic crystal growth". In: *Journal of Crystal Growth* 42.C (1977), pp. 11–14. ISSN: 00220248. DOI: 10.1016/0022-0248(77)90171-3.
- [129] Lauer, E., Hu, X., Hickel, S., and Adams, N. A. "Numerical investigation of collapsing cavity arrays". In: *Physics of Fluids* 24.5 (2012).
- [130] Le Gratiet, L. and Garnier, J. "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity". In: *International Journal for Uncertainty Quantification* 4.5 (2014).
- [131] LeVeque, R. J. *Numerical methods for conservation laws*. 2nd ed. Lectures in mathematics ETH Zürich. Basel; Boston: Birkhäuser Verlag, 1992, p. 214. ISBN: 978-3-7643-2723-1 978-0-8176-2723-2.

- [132] Liang, Y. and Luo, X. "Hydrodynamic instabilities of two successive slow/fast interfaces induced by a weak shock". In: *Journal of Fluid Mechanics* 955 (2023), A40.
- [133] Liepmann, H. W. and Roshko, A. *Elements of Gasdynamics*. John Wiley & Sons, Inc., New York, 1957.
- [134] Lighthill, M. J. and Whitham, G. B. "On kinematic waves II. A theory of traffic flow on long crowded roads". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229.1178 (1955), pp. 317–345.
- [135] López, J., Gómez, P., and Hernández, J. "A volume of fluid approach for crystal growth simulation". In: *Journal of Computational Physics* 229.19 (2010), pp. 6663–6672.
- [136] Luo, J., Hu, X., and Adams, N. A. "A conservative sharp interface method for incompressible multiphase flows". In: *Journal of Computational Physics* 284 (2015), pp. 547–565.
- [137] Markl, M. and Körner, C. "Multiscale Modeling of Powder Bed-Based Additive Manufacturing". In: *Annual Review of Materials Research* 46.May (2016), pp. 93–123. ISSN: 15317331. DOI: 10.1146/annurev-matsci-070115-032158.
- [138] Matérn, B. *Spatial variation*. Vol. 36. Springer Science & Business Media, 2013.
- [139] Mathijssen, A. J. T. M., Lisicki, M., Prakash, V. N., and Mossige, E. J. L. "Culinary fluid mechanics and other currents in food science". In: *Rev. Mod. Phys.* 95 (2 June 2023), p. 025004. DOI: 10.1103/RevModPhys.95.025004. URL: <https://link.aps.org/doi/10.1103/RevModPhys.95.025004>.
- [140] Meiron, D. I. "Selection of steady states in the two-dimensional symmetric model of dendritic growth". In: *Physical Review A* 33.4 (1986), p. 2704.
- [141] Mendoza Jimenez, E., Ding, D., Su, L., Joshi, A., Singh, A., Reeja-Jayan, B., and Beuth, J. "Parametric analysis to quantify process input influence on the printed densities of binder jetted alumina ceramics". In: *Additive Manufacturing* 30 (2019). DOI: 10.1016/j.addma.2019.100864.
- [142] Meng, L. and Zhang, J. "Process Design of Laser Powder Bed Fusion of Stainless Steel Using a Gaussian Process-Based Machine Learning Model". In: *JOM* 72.1 (2020), pp. 420–428.
- [143] Meng, X., Wang, Z., Fan, D., Triantafyllou, M., and Karniadakis, G. E. *A fast multi-fidelity method with uncertainty quantification for complex data correlations: Application to vortex-induced vibrations of marine risers*. 2020. arXiv: 2012.13481 [physics.flu-dyn].
- [144] Močkus, J. "On Bayesian methods for seeking the extremum". In: *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*. Springer. 1975, pp. 400–404.
- [145] Munz, C.-D., Ommes, P., and Schneider, R. "A three-dimensional finite-volume solver for the Maxwell equations with divergence cleaning on unstructured meshes". In: *Computer Physics Communications* 130.1-2 (2000), pp. 83–117.
- [146] Murio, D. A. *The mollification method and the numerical solution of ill-posed problems*. John Wiley & Sons, 2011.
- [147] 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.
- [148] Nourgaliev, R. R. and Theofanous, T. G. "High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set". In: *Journal of Computational Physics* 224.2 (June 2007), pp. 836–866. ISSN: 00219991. DOI: 10.1016/j.jcp.2006.10.031.
- [149] Oldfield, W. "Computer model studies of dendritic growth". In: *Material Science and Engineering* 11 (4 1973), pp. 211–218.

- [150] Osher, S. and Fedkiw, R. P. "Level set methods: an overview and some recent results". In: *Journal of Computational Physics* 169.2 (2001), pp. 463–502.
- [151] Osher, S. and Fedkiw, R. P. *Level set methods and dynamic implicit surfaces*. Vol. 200. Springer New York, 2005.
- [152] 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.
- [153] "Overview 12: Fundamentals of dendritic solidification—II development of sidebranch structure". In: *Acta Metallurgica* 29.5 (1981), pp. 717–734. ISSN: 0001-6160. DOI: [https://doi.org/10.1016/0001-6160\(81\)90116-4](https://doi.org/10.1016/0001-6160(81)90116-4).
- [154] Pan, S., Han, L., Hu, X., and Adams, N. A. "A conservative interface-interaction method for compressible multi-material flows". In: *Journal of Computational Physics* 371 (2018), pp. 870–895. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.02.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999118300834>.
- [155] Pan, S., Lyu, X., Hu, X. Y., and Adams, N. A. "High-order time-marching reinitialization for regional level-set functions". In: *Journal of Computational Physics* 354 (2018), pp. 311–319. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2017.10.054>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999117308215>.
- [156] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [157] Peherstorfer, B., Willcox, K., and Gunzburger, M. "Survey of multifidelity methods in uncertainty propagation, inference, and optimization". In: *Siam Review* 60.3 (2018), pp. 550–591.
- [158] Pelce, P. and Pomeau, Y. "Dendrites in the small undercooling limit". In: *Dynamics of Curved Fronts*. Elsevier, 1988, pp. 327–340.
- [159] 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.
- [160] Perdikaris, P. and Karniadakis, G. E. "Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond". In: *Journal of The Royal Society Interface* 13.118 (2016), p. 20151107.
- [161] Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2198 (2017), p. 20160751.
- [162] Perdikaris, P., Venturi, D., and Karniadakis, G. E. "Multifidelity information fusion algorithms for high-dimensional systems and massive data sets". In: *SIAM Journal on Scientific Computing* 38.4 (2016), B521–B538.
- [163] Perrone, V., Shen, H., Zolic, A., Shcherbatyi, I., Ahmed, A., Bansal, T., Donini, M., Winkelmoen, F., Jenatton, R., Faddoul, J. B., et al. "Amazon SageMaker Automatic Model Tuning: Scalable Gradient-Free Optimization". In: *KDD21*. 2021.
- [164] Perttunen, C. D. and Stuckman, B. E. "The rank transformation applied to a multivariate method of global optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics* 20.5 (1990), pp. 1216–1220.

- [165] Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. “Quantile-Based Optimization of Noisy Computer Experiments With Tunable Precision”. In: *Technometrics* 55.1 (2012), pp. 2–13. DOI: 10.1080/00401706.2012.707580.
- [166] Pieper, J. L., Schranner, F. S., Hermange, G., and Adams, N. A. “Sharp-Interface Simulations of Heat-Driven Flows with Marangoni Effect”. In: *10th International Symposium on Turbulence and Shear Flow Phenomena (TSFP10), Chicago, USA, July, 2017*. 2017.
- [167] Raissi, M. and Karniadakis, G. *Deep Multi-fidelity Gaussian Processes*. 2016. arXiv: 1604.07484 [cs.LG].
- [168] Rankouhi, B., Jahani, S., Pfefferkorn, F., and Thoma, D. “Compositional grading of a 316L-Cu multi-material part using machine learning for the determination of selective laser melting process parameters”. In: *Additive Manufacturing* 38 (2021). DOI: 10.1016/j.addma.2021.101836.
- [169] Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. Cambridge Massachusetts, USA: The MIT Press, 2006.
- [170] 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.
- [171] Richards, P. I. “Shock waves on the highway”. In: *Operations research* 4.1 (1956), pp. 42–51.
- [172] Roe, P. L. “Approximate Riemann solvers, parameter vectors, and difference schemes”. In: *Journal of computational physics* 43.2 (1981), pp. 357–372.
- [173] Roosen, A. R. and Taylor, J. E. “Modeling crystal growth in a diffusion field using fully faceted interfaces”. In: *Journal of computational physics* 114.1 (1994), pp. 113–128.
- [174] 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.
- [175] Roussel, O. and Schneider, K. “Adaptive multiresolution computations applied to detonations”. In: *Zeitschrift für Physikalische Chemie* 229.6 (2015), pp. 931–953.
- [176] Runge, C. “Ueber die numerische Auflösung von Differentialgleichungen”. In: *Mathematische Annalen* 46.2 (June 1895), pp. 167–178. ISSN: 00255831. DOI: 10.1007/BF01446807.
- [177] Saint-Venant, A. J.-C. de et al. “Théorie du mouvement non-permanent des eaux, avec application aux crues des rivières et à l’introduction des marées dans leur lit”. In: *CR Acad. Sci. Paris* 73.147-154 (1871), p. 5.
- [178] Salimbeni, H. and Deisenroth, M. “Doubly stochastic variational inference for deep gaussian processes”. In: *Advances in Neural Information Processing Systems*. 2017.
- [179] Saunders, R., Butler, C., Michopoulos, J., Lagoudas, D., Elwany, A., and Bagchi, A. “Mechanical behavior predictions of additively manufactured microstructures using functional Gaussian process surrogates”. In: *npj Computational Materials* 7.1 (2021), pp. 1–11.
- [180] Scardovelli, R. and Zaleski, S. “Direct numerical simulation of free-surface and interfacial flow”. In: *Annual review of fluid mechanics* 31.1 (1999), pp. 567–603.
- [181] 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.
- [182] Scott, W., Frazier, P., and Powell, W. “The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression”. In: *SIAM Journal on Optimization* 21.3 (2011), pp. 996–1026.

- [183] Sethian, J. A. and Smereka, P. "Level set methods for fluid interfaces". In: *Annual review of fluid mechanics* 35.1 (2003), pp. 341–372.
- [184] Shyy, W., Udaykumar, H. S., and Liang, S.-J. "An interface tracking method applied to morphological evolution during phase change". In: *International journal of heat and mass transfer* 36.7 (1993), pp. 1833–1844.
- [185] Sinnott, S. B. *Material design and discovery with computational materials science*. Jan. 1970. URL: <https://avs.scitation.org/doi/10.1116/1.4813689>.
- [186] Słota, D. "Direct and inverse one-phase Stefan problem solved by the variational iteration method". In: *Computers & Mathematics with Applications* 54.7-8 (2007), pp. 1139–1146.
- [187] Słota, D. "Solving the inverse Stefan design problem using genetic algorithms". In: *Inverse Problems in Science and Engineering* 16.7 (2008), pp. 829–846.
- [188] Snoek, J., Swersky, K., Zemel, R., and Adams, R. "Input warping for bayesian optimization of non-stationary functions". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1674–1682.
- [189] Soni, V., Hadjadj, A., Roussel, O., and Moebis, 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.
- [190] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting". In: *IEEE Transactions on Information Theory* 58.5 (2012), pp. 3250–3265. DOI: 10.1109/tit.2011.2182033.
- [191] 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.
- [192] Stampella, M. B. and Tarzia, D. A. "Determination of one or two unknown thermal coefficients of a semi-infinite material through a two-phase Stefan problem". In: *International journal of engineering science* 27.11 (1989), pp. 1407–1419.
- [193] Stefan, J. "Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere". In: *Annalen der Physik* 278.2 (1891), pp. 269–286.
- [194] Stoker, J. and Lindsay, R. B. "Water waves". In: *PhT* 11.8 (1958), p. 28.
- [195] Strickland, J., Nenchev, B., and Dong, H. "On directional dendritic growth and primary spacing—a review". In: *Crystals* 10.7 (2020), pp. 1–30. ISSN: 20734352. DOI: 10.3390/cryst10070627.
- [196] 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.
- [197] 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.
- [198] 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.
- [199] Toro, E. F. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. 3rd ed. Dordrecht ; New York: Springer, 2009, p. 724. ISBN: 978-3-540-25202-3 978-3-540-49834-6.
- [200] Toro, E. F. *Shock-capturing methods for free-surface shallow flows*. John Wiley, 2001.

- [201] Trivedi, R., Mazumder, P., and Tewari, S. N. "The effect of convection on disorder in primary cellular and dendritic arrays". In: *Metallurgical and Materials Transactions A* 33.12 (2002), pp. 3763–3775.
- [202] 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.
- [203] Tryggvason, G. "Numerical simulations of the Rayleigh-Taylor instability". In: *Journal of Computational Physics* 75.2 (1988), pp. 253–282.
- [204] 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.
- [205] 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.
- [206] Van Beers, W. C. and Kleijnen, J. P. "Kriging for interpolation in random simulation". In: *Journal of the Operational Research Society* 54 (2003), pp. 255–262.
- [207] Voller, V. R. "Enthalpy method for inverse Stefan problems". In: *Numerical Heat Transfer, Part B Fundamentals* 21.1 (1992), pp. 41–55.
- [208] Voller, V. R. "An enthalpy based scheme for simulating dendritic growth". In: *Modeling of Casting, Welding and Advanced Solidification Processes-XI*. 2006, pp. 465–472.
- [209] Voller, V. R. "An enthalpy method for modeling dendritic growth in a binary alloy". In: *International Journal of Heat and Mass Transfer* 51.3-4 (2008), pp. 823–834.
- [210] 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.
- [211] Wiens, A. E., Copan, A. V., and Schaefer, H. F. "Multi-fidelity Gaussian process modeling for chemical energy surfaces". In: *Chemical Physics Letters* 737 (2019), p. 100022.
- [212] Winter, J., Abaidi, R., Kaiser, J., Adami, S., and Adams, N. "Multi-fidelity Bayesian optimization to solve the inverse Stefan problem". In: *Computer Methods in Applied Mechanics and Engineering* 410 (2023), p. 115946. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2023.115946>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782523000695>.
- [213] 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: <https://doi.org/10.1016/j.cma.2022.114799>.
- [214] Winter, J., Kaiser, J., Adami, S., and Adams, N. "Numerical investigation of 3D drop-breakup mechanisms using a sharp interface level-set method". In: *11th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 11*. 2019.
- [215] Winter, J., Schraner, F. S., and Adams, N. "Iterative Bayesian Optimization of an Implicit LES Method for Underresolved Simulations of Incompressible Flows". In: *10th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 10*. 2017.
- [216] Wu, J. and Frazier, P. "The parallel knowledge gradient method for batch Bayesian optimization". In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 3126–3134.

- [217] Wu, J. and Frazier, P. I. "Continuous-fidelity Bayesian optimization with knowledge gradient". In: (2018).
- [218] Wu, J., Toscano-Palmerin, S., Frazier, P. I., and Wilson, A. G. "Practical multi-fidelity bayesian optimization for hyperparameter tuning". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 788–798.
- [219] Yabansu, Y., Rehn, V., Hötzer, J., Nestler, B., and Kalidindi, S. "Application of Gaussian process autoregressive models for capturing the time evolution of microstructure statistics from phase-field simulations for sintering of polycrystalline ceramics". In: *Modelling and Simulation in Materials Science and Engineering* 27.8 (2019). DOI: 10.1088/1361-651X/ab413e.
- [220] Youngs, D. L. "Numerical simulation of turbulent mixing by Rayleigh-Taylor instability". In: *Physica D: Nonlinear Phenomena* 12.1-3 (1984), pp. 32–44.
- [221] Zabaras, N. "Inverse finite element techniques for the analysis of solidification processes". In: *International Journal for Numerical Methods in Engineering* 29.7 (1990), pp. 1569–1587.
- [222] Zabaras, N., Ruan, Y., and Richmond, O. "Design of two-dimensional Stefan processes with desired freezing front motions". In: *Numerical Heat Transfer, Part B Fundamentals* 21.3 (1992), pp. 307–325.
- [223] Zhang, H., Choi, J. P., Moon, S. K., and Ngo, T. H. "A knowledge transfer framework to support rapid process modeling in aerosol jet printing". In: *Advanced Engineering Informatics* 48 (2021), p. 101264.