

## Efficient simulation of multiphysics problems with application to metal-additive manufacturing

Patrick Zimbrod

### Angaben zur Veröffentlichung / Publication details:

Zimbrod, Patrick. 2025. "Efficient simulation of multiphysics problems with application to metal-additive manufacturing." Augsburg: Universität Augsburg.



UNIVERSITÄT AUGSBURG  
Fakultät für angewandte Informatik

# **Efficient Simulation of Multiphysics Problems with Application to Metal-Additive Manufacturing**

**Dissertation  
zur Erlangung des akademischen Grades  
Doktoringenieur (Dr.-Ing.)**

an der Fakultät für angewandte Informatik der Universität Augsburg

---

vorgelegt von: Patrick Zimbrod, M.Sc.

geboren am: 18.10.1994 in Augsburg

eingereicht am: 10.07.2024

Anfertigung am Lehrstuhl: Produktionsinformatik  
Fakultät für angewandte Informatik

1. Gutachter: Prof. Dr.-Ing. Johannes Schilp

2. Gutachter: Prof. Dr.-Ing. Lars Mikelsons

Tag der mündlichen Prüfung: 27.02.2025



## Declaration

The results discussed in this dissertation have been published in the following articles:

1. Patrick Zimbrod and Johannes Schilp. Modelling of microstructures during in-situ alloying in additive manufacturing for efficient material qualification processes. *Simulation in Produktion und Logistik 2021: 19. ASIM-Fachtagung Simulation in Produktion und Logistik, Erlangen, 15. – 17. September 2021*, pages 177–188, 2021. URL <https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/docId/87973>
2. Patrick Zimbrod, Magdalena Schreter, and Johannes Schilp. Efficient Simulation of Complex Capillary Effects in Advanced Manufacturing Processes using the Finite Volume Method. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6, November 2022. doi: 10.1109/ICECCME55909.2022.9988504
3. Patrick Zimbrod, Michael Fleck, and Johannes Schilp. An Application-Driven Method for Assembling Numerical Schemes for the Solution of Complex Multiphysics Problems. *Applied System Innovation*, 7(3):35, June 2024. ISSN 2571-5577. doi: 10.3390/asi7030035. URL <https://www.mdpi.com/2571-5577/7/3/35>

The following reports and theses that were supervised by me also contributed to the contents of this dissertation:

1. Brixle, Nicolas. Modellierung von Transportvorgängen für die pulverbettbasierte Additive Fertigung mittels Maschinellen Lernens. Master's thesis, University of Augsburg, Faculty of Applied Computer Science, June 2022
2. Mayer, Dennis. Automatisierte Transformation von partiellen Differentialgleichungen in die schwache Form für die Finite-Elemente-Methode. Master's thesis, University of Augsburg, Faculty of Applied Computer Science, October 2023
3. Thörner, Katharina. Analyse von Simulationsmethoden auf Mesoebene für das selektive Laser Schmelzen. Seminar paper, University of Augsburg, Faculty of Applied Computer Science, July 2023
4. Wiesner, Leonard. Multiphysics-Simulationen in der Prototypenentwicklung: Fortschritte beim Digital Prototyping und dem Digitalen Zwilling. Seminar paper, University of Augsburg, Faculty of Applied Computer Science, July 2023
5. Nothelfer, Tim. Recherche zu bestehenden Umsetzungen von numerischen Berechnungsverfahren für Multiphysik Probleme. Seminar paper, University of Augsburg, Faculty of Applied Computer Science, July 2023

Patrick Zimbrod, M.Sc.

10.07.2024

## **Abstract**

The accurate representation and prediction of physical phenomena through numerical computer codes remains a vast and intricate interdisciplinary topic of research. Especially within the last decades, there has been a considerable push toward high-performance numerical schemes to solve partial differential equations (PDEs) from the applied mathematics and numerics community. The resulting landscape of choices regarding numerical schemes for a given system of PDEs can thus easily appear daunting for an application expert who is familiar with the relevant physics, but not necessarily with the numerics. These high-performance schemes in particular pose a substantial hurdle for domain scientists regarding their theory and implementation. In this thesis, a unifying scheme for grid-based approximation methods is proposed to address this issue. Some well-defined restrictions are introduced to systematically guide an application expert through the process of classifying a given multiphysics problem, identifying suitable numerical schemes and implementing them. By defining a fixed set of input parameters, amongst them for example the governing equations and the hardware configuration, the process can be executed in a systematic and reproducible manner. This method not only helps to identify and assemble suitable schemes but enables the unique combination of multiple methods. This process and its effectiveness are exemplarily demonstrated using different approaches. As a practically relevant and complex multiphysics problem, the powder bed scale process dynamics during Laser Powder Bed Fusion is investigated. After a thorough investigation of current simulation approaches, it is shown how this work contributes to enhancing the current state of research by proposing a tailored discretization to this problem. Overall, it is systematically shown how one can exploit some given properties of a PDE problem to attain an efficient compound discretization.



# Table of contents

- List of figures** **xi**
  
- List of tables** **xv**
  
- List of Symbols** **xvii**
  
- 1 Introduction** **1**
  - 1.1 Simulation and Virtual Prototypes . . . . . 1
  - 1.2 Simulation of Physical Processes . . . . . 3
  - 1.3 Motivation . . . . . 6
  - 1.4 Contributions . . . . . 6
  
- 2 Theory of Multiphysics Problems** **9**
  - 2.1 Modeling Strategies and Partial Differential Equations . . . . . 9
  - 2.2 Common Grid Based Approximations . . . . . 12
    - 2.2.1 Restrictions . . . . . 13
    - 2.2.2 Finite Difference Method . . . . . 14
    - 2.2.3 Finite Volume Method . . . . . 16
    - 2.2.4 Finite Element Method . . . . . 19
    - 2.2.5 Discontinuous Galerkin Method . . . . . 24
  - 2.3 Discretisations of Multiphysics Problems . . . . . 27
  - 2.4 Similarities between Approximation Schemes . . . . . 27
  
- 3 Simulation of Laser Powder Bed Fusion** **31**
  - 3.1 Process Description . . . . . 31
  - 3.2 Multiphysics Problems of Interest . . . . . 32

3.2.1	Macroscopic Scale . . . . .	33
3.2.2	Microscopic Scale . . . . .	33
3.2.3	Mesosopic Scale . . . . .	35
3.3	Research Gap . . . . .	39
<b>4</b>	<b>Scientific Methodology</b>	<b>41</b>
4.1	Classification and Problem Statement . . . . .	41
4.2	Formal Analysis . . . . .	42
4.3	Deduction of a Decision Scheme . . . . .	43
4.4	Application and Validation . . . . .	43
<b>5</b>	<b>Formal Characterization of Discretisation Methods</b>	<b>45</b>
5.1	Discontinuous Galerkin Method . . . . .	45
5.2	Continuous Galerkin Finite Element Method . . . . .	47
5.3	Finite Difference Method . . . . .	49
5.4	Finite Volume Method . . . . .	53
5.5	Findings . . . . .	55
<b>6</b>	<b>Deduction of the Unified Modeling Methodology</b>	<b>59</b>
6.1	Preliminary Assumptions . . . . .	59
6.2	PDE Classification . . . . .	61
6.3	Domain Geometry . . . . .	64
6.4	PDE Linearity . . . . .	64
6.5	Computing Environment . . . . .	66
6.6	Problem Scale . . . . .	67
6.7	Resulting Method . . . . .	69
6.8	Weak Form Derivation . . . . .	73
6.9	Findings . . . . .	78
<b>7</b>	<b>Computational Examples</b>	<b>81</b>
7.1	Allen Cahn Equation . . . . .	81
7.2	Two-phase Advection . . . . .	94

---

<b>8</b>	<b>Application to Melt Flow Evolution in Metal AM</b>	<b>105</b>
8.1	Physics and Governing Equations . . . . .	106
8.2	Computational Domain . . . . .	108
8.3	Computing Resources . . . . .	109
8.4	Problem Scale . . . . .	109
8.5	Classification . . . . .	110
8.6	Weak Formulation . . . . .	112
8.7	Findings . . . . .	116
<b>9</b>	<b>Conclusions and Future Work</b>	<b>119</b>
	<b>References</b>	<b>123</b>
	<b>Appendix A Literature Research for Mesoscopic PBF-LB/M Simulation</b>	<b>161</b>
	<b>Appendix B Implementation Details of the Presented Models</b>	<b>169</b>
B.1	Advection Equation . . . . .	169
B.1.1	Derivation of the Weak Form . . . . .	169
B.1.2	Simulation Parameters . . . . .	170



# List of figures

1.1	A coarse overview of the virtual prototyping process . . . . .	2
1.2	The multiple length scales involved in modeling the mechanical response of additively manufactured metallic parts. . . . .	3
1.3	Flow simulation of a turbulent, isentropic vortex advection problem using two different solvers. . . . .	5
1.4	Unstable (a) and stable (b) solution of a mixed diffusion problem using the FEM . . . . .	6
2.1	A typical simulation pipeline . . . . .	9
2.2	Approximation of a two-dimensional derivative of the field $u$ by a 4-point Finite Difference stencil. . . . .	15
2.3	Approximation of a Finite Difference interpolation to a one-dimensional function . . . . .	16
2.4	Approximation of a conserved quantity using the FVM . . . . .	18
2.5	Approximation of a Finite Volume interpolation to a one-dimensional function . . . . .	19
2.6	Approximation of a Finite Element interpolation to a one-dimensional function . . . . .	21
2.7	Linear shape functions in the FEM on a two-dimensional, triangular mesh	23
2.8	Approximation of a DG interpolation to a one-dimensional function .	25
2.9	Linear shape functions in the DGM on a two-dimensional, triangular mesh. . . . .	26
3.1	Schematic overview of the PBF-LB/M process. . . . .	32
3.2	Phase field simulation of the solidification process of a CoCrFeMnNi High Entropy Alloy during PBF-LB/M after different time steps. . . .	34

3.3	Finite Volume model of the melting and solidification of 316L stainless steel during PBF-LB/M. . . . .	36
3.4	Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M. . . . .	37
3.5	Quantitative summary of previous works that use the Finite Volume Method to resolve melt pool dynamics in PBF-LB/M. . . . .	38
5.1	Coupling of global DoFs in the CGM versus DGM . . . . .	49
5.2	Comparison of the nodal nature of the FDM versus the cell-wise assembly used in the CG FEM for an identical, cartesian triangulation with 9 nodes. . . . .	49
5.3	Comparison of FVM versus DGM on an identical quadrilateral triangulation. . . . .	54
6.1	Classification of PDEs up to second order by qualitative nature and types	63
6.2	Coupling of global DoFs in the CGM versus DGM . . . . .	69
6.3	Graphic summary of the proposed process for choosing appropriate numerical schemes. . . . .	72
6.4	Generation of a weak form expression for a differential operator $L(u)$ .	77
7.1	Evolution of the phase front at $t = 100s$ with respect to the initial condition. . . . .	89
7.2	Comparison of Finite Difference and Finite Element solutions to the analytical solution given by Equation 7.6 . . . . .	90
7.3	Solution of the two-dimensional Allen Cahn equation using the FDM and FEM. . . . .	92
7.4	Evolution of radius over time of the vanishing grain problem. Both Finite Difference and Finite Element solutions show a considerable, accumulating error toward the analytical solution. . . . .	93
7.5	Solution of the two-dimensional advection equation using the Finite Volume and DGM. . . . .	100
7.6	Comparison of shared DoFs (colored dots) between cells for a fourth-order DGM with an FV method with an equal amount of total DoFs (100). . . . .	102

---

7.7	Comparison of run times for the DGM and FVM advection benchmark case . . . . .	102
8.1	Slice of the computational domain for the Laser Powder Bed Fusion test case . . . . .	109



# List of tables

2.1	Different modeling approaches and some typical applications ( <a href="#">Bungartz et al., 2013</a> ). . . . .	10
5.1	Comparison of numerical schemes regarding their restrictions . . . . .	56
6.1	Tableau of differential operator weak forms, depending on the dimensionality of the target function $u$ . . . . .	78
7.1	Parameters for the Allen Cahn model . . . . .	82
7.2	Run times of the Finite Element and Finite Difference model of the 1D Allen Cahn equation. . . . .	91
7.3	Run times of the Finite Element and Finite Difference model of the 2D Allen Cahn equation. . . . .	93
7.4	Hardware configurations for the advection equation model problem . . . . .	96
8.1	Resulting schemes of the proposed decision method for the Laser Powder Bed Fusion example problem . . . . .	116
A.1	Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M. . . . .	161
A.2	Quantitative summary of previous works that use the FVM to resolve melt pool dynamics in PBF-LB/M. . . . .	166



# List of Symbols

## Roman Symbols

$a_{ij}$	Matrix of coefficients of second order derivatives
$a_{ij}$	Matrix of coefficients of second order derivatives
$a(\cdot, \cdot)$	Bilinear form
$a(\cdot, \cdot)$	Bilinear form
$c_p$	Heat capacity
$F(\cdot; \cdot)$	Semilinear form
$F(\cdot; \cdot)$	Semilinear form
$G(u)$	Functional of $u$
$h$	Grid spacing
$\underline{\underline{I}}$	Identity tensor
$\underline{\underline{K}}$	Global stiffness matrix
$\underline{k}$	Gravitational force vector
$K$	Closed $n$ -dimensional geometric set
$\underline{\underline{K}}^{(e)}$	Element stiffness matrix
$l(\cdot)$	Linear form
$l(\cdot)$	Linear form

---

$L_v$	Latent heat of vaporisation
$\underline{\underline{M}}$	Global mass matrix
$\underline{\underline{M}}$	Global mass matrix
$\underline{n}$	Unit normal vector of a mesh facet
$\underline{n}_{-ab}$	Unit interface normal vector between phases $a$ and $b$
$P$	Space of real-valued functions defined over $K$
$p$	Pressure
$P_L$	Laser power
$p_{\text{recoil}}$	Recoil pressure
$R$	Universal gas constant
$R_L$	Laser spot radius
$T$	Temperature
$\underline{\underline{T}}$	Capillary stress tensor
$u$	Solution function of a PDE
$\underline{u}$	Flow velocity
$V$	A function space
$v$	Test function
$v_u$	Test function corresponding to function $u$

### Greek Symbols

$\alpha$	Passively advected, scalar function
$\tilde{\alpha}$	Wavelength dependent laser absorptivity of a material
$\partial\Omega$	The boundary of $\Omega$

---

$\partial\Omega$	The boundary of $\Omega$
$\epsilon$	Radiation emissivity
$\eta$	Dynamic viscosity
$\Gamma$	A subdomain of the boundary $\partial\Omega$
$\Gamma$	A subdomain of the boundary $\partial\Omega$
$\kappa$	Thermal conductivity
$\lambda$	Eigenvalue
$\lambda$	Eigenvalue
$\Omega$	A domain of interest
$\Omega$	A domain of interest
$\phi$	Phase field variable
$\phi$	Phase field variable
$\phi_i$	$i$ -th Finite Element shape function
$\psi$	Test function
$\rho$	Density

### Subscripts

h	Discrete variant of a function
---	--------------------------------

### Other Symbols

$D^k u$	$k$ -th derivative of the function $u$
$d\tilde{s}$	General surface integration measure
$dS$	Integration measure over exterior facets
$ds$	Integration measure over interior facets

---

$dV$	Volume integration measure
$ \cdot _{H^1}$	$H^1$ norm
$H^1(\Omega)$	Function space equipped with the $H^1$ norm over the domain $\Omega$
$L(u)$	A differential operator in the function $u$
$L(u)$	A differential operator in the function $u$
$ \cdot _{L^2}$	$L^2$ norm
$L^2(\Omega)$	Function space equipped with the $L^2$ norm over the domain $\Omega$
$\nabla$	Gradient operator
$\nabla$	Gradient operator
$\nabla \cdot$	Divergence operator
$\nabla \cdot$	Divergence operator
$\nabla \times$	Rotation operator
$\nabla \times$	Rotation operator
$\Delta$	Laplacian operator
$\Delta$	Laplacian operator
$\langle \cdot, \cdot \rangle$	Inner product of two vectors
$\otimes$	Outer product: $C_{ij} = a_i \otimes b_j$
$\otimes$	Outer product: $C_{ij} = a_i \otimes b_j$
$\mathbb{P}^k$	Degree $k$ Lagrange element on a triangle/tetrahedron
$\partial_t$	Time derivative
$\partial_t$	Time derivative
$\mathbb{Q}^k$	Degree $k$ Lagrange element on an interval/quadrilateral/cube

**Acronyms / Abbreviations**

ALE	Arbitrary Lagrangian Eulerian
AM	Additive Manufacturing
BC	Boundary Condition
BC	Boundary Condition
BC	Boundary Condition
IC	Initial Condition
CFD	Computational Fluid Dynamics
CG(M)	Continuous Galerkin (Method)
DEM	Discrete Element Method
DG(M)	Discontinuous Galerkin (Method)
DoF	Degree of Freedom
FD(M)	Finite Difference (Method)
FE(M)	Finite Element (Method)
FV(M)	Finite Volume (Method)
HDG	Hybridizable Discontinuous Galerkin (Method)
LBM	Lattice Boltzmann Method
ODE	Ordinary Differential Equation
PBF-LB/M	Laser Powder Bed Fusion of Metals
PDE	Partial Differential Equation
SPH	Smoothed Particle Hydrodynamics
SUPG	Streamline Upwind Petrov Galerkin

UFL	Unified Form Language
UFL	Unified Form Language
VoF	Volume of Fluid

# Chapter 1

## Introduction

### 1.1 Simulation and Virtual Prototypes

The design and validation of products in an engineering context has undergone substantial changes in the last two decades towards more digitally dependent processes. Especially simulation of products and their behavior under certain conditions has made a notable impact on product development. One of the most prevalent use cases for simulation within this domain is for the design of virtual prototypes. A virtual prototype, or digital mock-up, is a computer simulation of a physical product that can be presented, analyzed, and tested from concerned product life-cycle aspects such as design and engineering, manufacturing, service, and recycling as if on a real physical model. The construction and testing of a virtual prototype is called virtual prototyping ([Wang, 2002](#)).

Simulation of physical processes plays a crucial role in virtual prototyping. It allows engineers to model and understand complex physical phenomena that would otherwise not be accessible, e.g. due to lack of proper measuring equipment. From a more practical perspective, it allows for experimental studies with low effort. Tuning a given set of parameters in a model enables experts to understand and qualify processes with minimal empirical effort. Thus, simulation may contribute considerably towards saving cost, material and time. The tight integration of simulation into the virtual prototyping process is shown in [Figure 1.1](#). For instance, simulation-based virtual prototypes have been employed to accelerate the development of battery electric car components and are furthermore extensively used within the aerospace industry ([Moatamedi et al.](#),

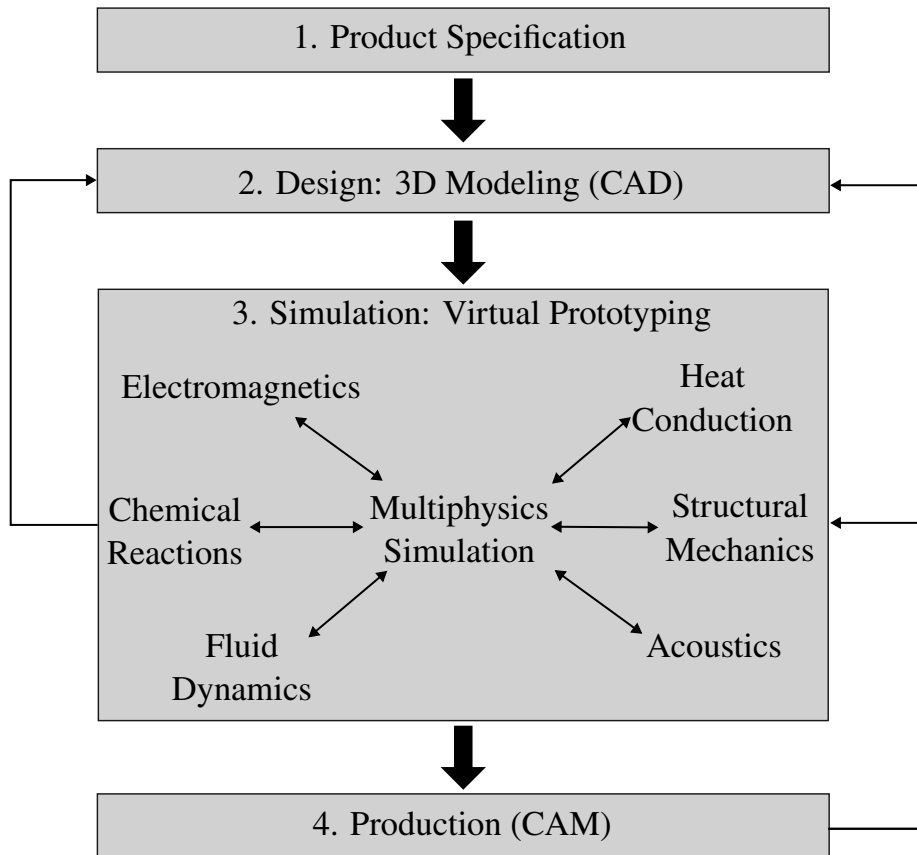


Fig. 1.1: A coarse overview of the virtual prototyping process. Simulation is crucially involved in refining and reworking the virtual product through multiple feedback loops. Following [Hirsch \(2007\)](#)

[2021](#)). Overall, several studies have found a clear upward trend in the worldwide market revolving around virtual prototyping. In a market study from 2017, the international market this year was quantified to be worth around \$210.40 million. Until 2025, an annual growth of 19.4% is expected in this scenario ([Grand View Research, 2017](#)). Another study quantifies the worldwide market to be \$1,058.63 million by 2026 ([Knowledge Sourcing Intelligence, 2021](#)). The market for simulation software in particular was found to have a turnover of \$11.08 billion and is expected to grow up to \$39.20 billion by the year 2028 ([Emergen Research, 2022](#)).

## 1.2 Simulation of Physical Processes

Simulation in itself, however, is an overall intricate, interdisciplinary topic that has been studied by the science, technology, engineering and mathematics (STEM) community for multiple decades. To this day, the majority of physical processes are described using Partial Differential Equations (PDEs) have to be approximated numerically to enable simulation (Morton and Mayers, 2005; Pinchover and Rubinstein, 2005). This process, oftentimes also referred to as discretizing a problem, is ambiguous as there does not yet exist a universal mathematical method capable of performing equally well for all modeling tasks. As such, generating numerical simulations for real physical problems remains a challenging topic.

To make an appropriate choice of modeling strategy, one first needs to decide on a set of relevant physics. This task in itself might already pose a considerable challenge, as shown in Figure 1.2 for the case of metal additive manufacturing (AM). For each level of hierarchy, the relevant quantity of interest is given along the accompanying physics and the characteristic length scale. As the process poses harsh, unsteady thermodynamic conditions, varying material constants, irregular microstructure and melt pool topography have to be considered (Hariharan et al., 2022, 2023). If one were to predict the mechanical properties of a metallic microstructure, it is necessary to consider a large amount of physics in the process. These physics may often span multiple

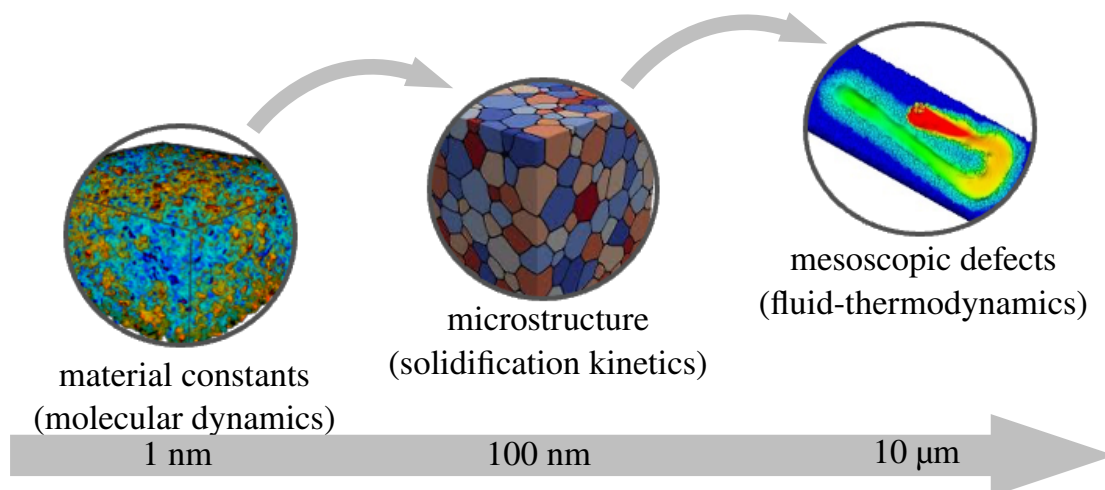


Fig. 1.2: The multiple length scales involved in modeling the mechanical response of additively manufactured metallic parts.

length scales, thus posing a so-called multiscale problem (Weinan, 2011; Weinan and Lu, 2011). Such processes are naturally even more challenging to solve than single-field problems. Not only does a modeling strategy have to fit each model involved, but, in many cases, there is also some coupling that has to be performed between models.

Furthermore, recent developments in numerical methods in the academic community have noticeably impacted the capabilities of such models, both regarding accuracy and performance. Especially within FEMs, there exists an increasing gap regarding overall performance between academic and commercially available software. The recent utilization of massively parallel hardware, such as Graphics Processing Units (GPUs) has contributed to widening this gap (Kirby and Mavriplis, 2020; Vargas et al., 2021). To emphasize that this difference in performance not only is of theoretical use but rather also has considerable practical impact, some studies have been published that also address the capabilities of methods using resource efficiency metrics. For instance, Vermeire and colleagues developed a custom fluid dynamics solver based on a flux reconstruction approach and benchmarked it against the popular Computational Fluid Dynamics software Star-CCM+ that is commercially available. The evaluation of a common problem encountered in external aerodynamics is shown in Figure 1.3. Lines drawn in orange denote the custom solver PyFR, black lines indicate solutions generated by the commercial solver Star-CCM+. For the same target model error, resource utilization differs by roughly a factor of 300. The PyFR model was run on an NVIDIA Tesla GPU whilst Star-CCM+ could only be executed on an Intel XEON-based CPU architecture. The authors use a custom resource metric that multiplies the sum of hardware acquisition and operating cost with model execution time. Using this as scale, one can observe that there exists a large gap between the two simulation environments that can mainly be attributed to the GPU acceleration of the custom, academic flow solver. For the same approximation error, the difference in resource utilization is around two orders of magnitude, highlighting the potential for savings in development time and cost in a product development environment.

However, given a particular choice of numerical method, there usually still is a non-negligible amount of setting up and tuning of model parameters involved. This process naturally becomes more intricate with more high-performance methods, as parameters have to be aligned with the target model and computing architecture. As such, developing and operating high-performance methods often remains a topic for experts in

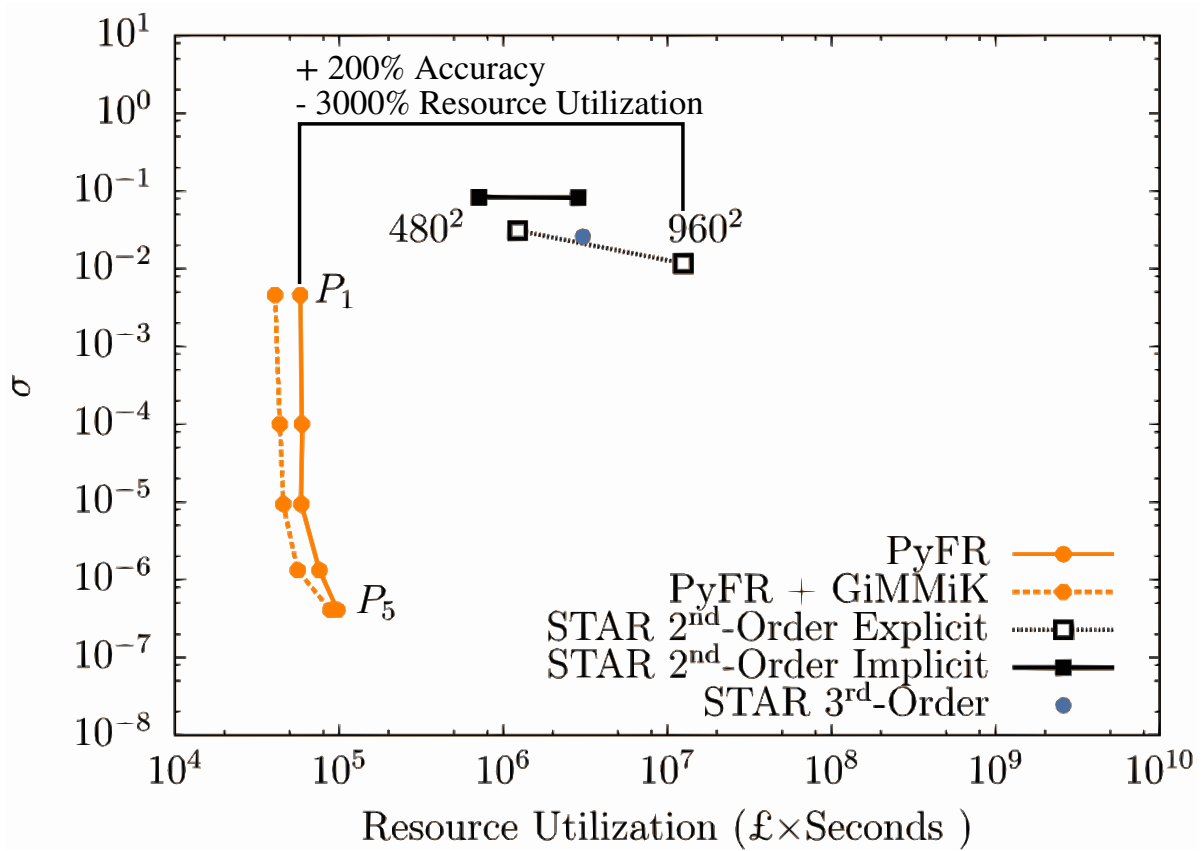


Fig. 1.3: Flow simulation of a turbulent, isentropic vortex advection problem using two different solvers. For the same error  $\sigma$ , resource utilization differs by two orders of magnitude. This figure is adapted from [Vermeire et al. \(2017\)](#).

numerical computing ([Kirby and Mavriplis, 2020](#); [Vargas et al., 2021](#)). Unfortunately, a bad parametrization may not only affect model performance but even stability to a point where no solution might be obtained. Figure 1.4 shows an example of how varying a single parameter of a model may completely dictate the stability and thus usability of an approximation. The two approximations were generated using the open-source software FEniCS with two different types of Finite Elements. Here, model accuracy is destroyed by an improper choice of finite element function space, i.e. the types of functions used to generate a discrete approximation. Such phenomena are common in so-called saddle point problems, which in turn may be found in various branches of physics ([Logg et al., 2012c](#)).

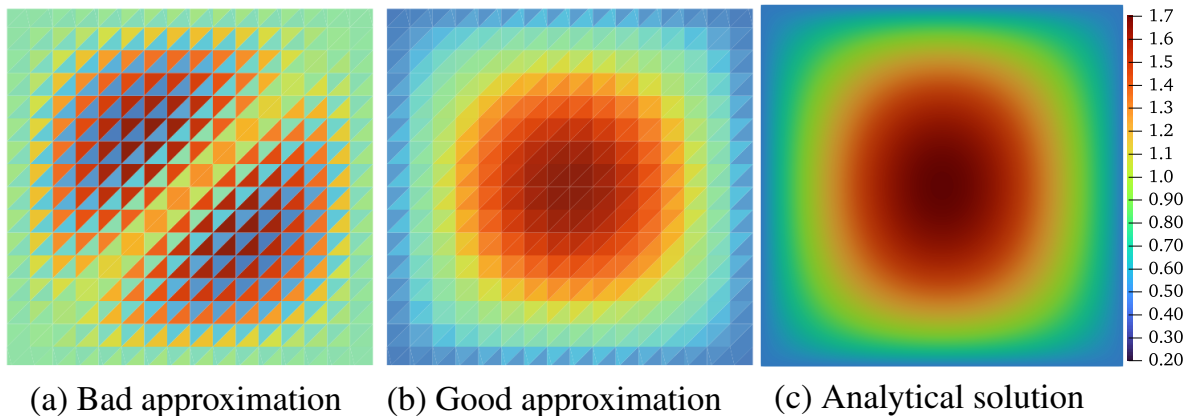


Fig. 1.4: Unstable (a) and stable (b) approximations of a mixed diffusion problem using the FEM. The typically smooth analytical solution (c) is well represented using solution (b) whereas (a) produces invalid results governed by unphysical oscillations. This figure is adapted from [Logg et al. \(2012c\)](#).

### 1.3 Motivation

In summary, simulation is of crucial importance for product development and is expected to further grow in the next years. The accurate modeling of complex physical processes offers large future potential regarding more efficient simulation due to recent developments in the research community. The developed, highly efficient methods, however, seem to lack adoption in practice. This is expected to be due to the high and increasing complexity of implementation as well as the intricate knowledge necessary for optimal parametrization and tuning. From a practical point of view, the question remains if and to what extent modern numerical methods may be applied to practically relevant simulation problems by engineers and researchers outside the respective developer community - henceforth denoted as "application experts".

### 1.4 Contributions

This work contributes to the wider field of application-driven simulation of physical processes that are governed by PDEs. It aims to improve the field of research in the following three key areas:

1. A common taxonomy for the most widespread, yet high-performance, grid-based approximation methods for PDEs is established, given some well-defined restric-

tions. By doing so, this work aims to unify the view on grid-based simulation techniques from an application point of view and create a better understanding of the individual properties that each discretization method exhibits. This work explicitly does not claim universal equivalence of these methods but rather outlines useful cases where equivalence can be shown.

2. A method is established that, given a particular multiphysics problem and hardware to execute on, recommends a set of numerical methods that fall in the above-mentioned taxonomy. The resulting mixed spatial discretization is stable, fits the qualitative nature of the quantities that are solved and suits the computational hardware in terms of performance requirements. Especially stability has been shown in Figure 1.4 to be an issue that is nontrivial to solve in more application-oriented domains. Furthermore, the method always yields, for the multiphysics problems of interest, a result that follows a strict, easy-to-follow process. For more intricate discretization schemes based on the FEM, it is also shown how to derive the necessary mathematical weak form that is required to implement such methods, which oftentimes also poses a considerable hurdle in practice.
3. This work discusses the state of the art for the simulation of mesoscopic melting and solidification within Laser Powder Bed Fusion. This model problem is taken as an illustrative example for a wider class of multiphysics problems. These are characterized by high complexity in terms of the number of PDEs in the system, varying physics and degree of coupling between equations. It is shown that the motivation for this work holds in particular for this specific model problem as well. That is, there exists a large number of proposed methods to solve this problem with no universal solution standing out in terms of accuracy, performance and ease of implementation. This gap is addressed using the established method, i.e., a mixed spatial discretization of the system is proposed that respects the characteristics of the problem. It is furthermore shown how the intricate mathematical formulation required to set up a corresponding computational model can be systematically attained from the system of PDEs. Lastly, by comparing the resulting configuration with the literature, it is shown that it falls well within current trends in this field regarding high-performance schemes whilst retaining the easier applicability of simpler discretizations.



# Chapter 2

## Theory of Multiphysics Problems

### 2.1 Modeling Strategies and Partial Differential Equations

The overarching goal of investigating Multiphysics Problems is the accurate representation and prediction of the behavior of physical processes.

This process may be separated into discrete parts which are shown in Figure 2.1 and consecutively executed. As a first step, a simplified and formal description of the real phenomenon has to be generated. This description that forms the basis for further calculations is denoted as the model. As such, an accurate model that captures all relevant aspects is considered preliminary for obtaining valid results from a subsequent simulation. Within this work, this will also be taken as a requirement and is thus not part of the presented research. The focus will instead be on the following two steps, that is, computation and implementation. The former denotes the adequate preparation of a model, such that it may be processed by a computer. This necessarily involves creating a discretization. During implementation, the previously worked-out simula-

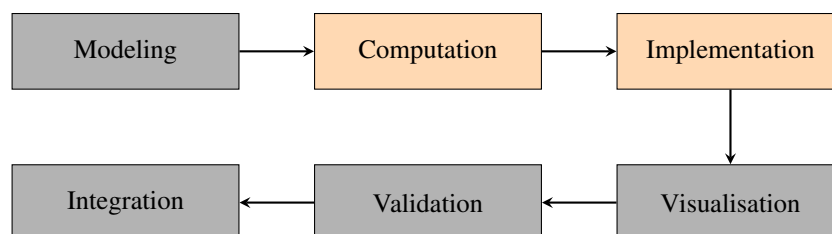


Fig. 2.1: A typical simulation pipeline. The core aspects of simulation that this work is concerned with are highlighted in orange. Adapted from [Bungartz et al. \(2013\)](#).

tion is typically tailored and rewritten towards computational performance. Such steps are oftentimes necessary from a practical perspective since the computation time may quickly become a bottleneck in terms of overall development resources (Bungartz et al., 2013).

In addition, not all possible modeling strategies are considered in this work. To model a given process, there exist several approaches that each focus on different aspects that are listed in Table 2.1. The problems of interest for this work are typically in the form of physical processes and their spatial and temporal evolution. Thus, the focus will lie on models that are described by PDEs. A more formal definition has been established by Evans:

**Definition 2.1.1** (Partial Differential Equation (Evans, 2010)). An expression of the form

$$F(D^k u(x), D^{k-1} u(x), \dots, Du(x), u(x), x) = 0 \quad (x \in U) \quad (2.1)$$

Table 2.1: Different modeling approaches and some typical applications (Bungartz et al., 2013).

Type	Exemplary	
	Formulation	Application
Algebraic equations	$E = mc^2$	Physical laws
Ordinary Differential Equations	$\frac{d}{dt}y(t) = y(t)$	Temporal evolution of systems, e.g. growth
Partial Differential Equations	$\Delta u = f$	Deflection of a membrane under load
Automata and State Diagrams		Queueing, text recognition
Graphs		Processes and work flows
Probability distributions		Stochastic processes and correlations
Rule based systems and fuzzy logic		Control system tasks
Language concepts	UML	Complex software systems
Algebraic structures		Quantum mechanical groups

is called partial differential equation (PDE) of  $k$ th order, where

$$F : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \dots \times \mathbb{R}^n \times \mathbb{R} \times U \rightarrow \mathbb{R} \quad (2.2)$$

is given and one seeks to find the function

$$u : U \rightarrow \mathbb{R} \quad (2.3)$$

The solution variable  $u$  might appear as a scalar-, vector- or even tensor-valued quantity. As such,  $u$  can also represent an entire system of variables described by a system of equations. For instance, the flow of a viscous fluid through a pipe is always characterized by its velocity  $\underline{u}$  and pressure field  $p$ . In this case, the solution function  $\underline{u}$  can be defined as:

$$\underline{u}(x, y, z, t) = \begin{bmatrix} v_x(x, y, z, t) \\ v_y(x, y, z, t) \\ v_z(x, y, z, t) \\ p(x, y, z, t) \end{bmatrix}, \quad (2.4)$$

with the underline notation  $\underline{\cdot}$  denoting a vector-valued quantity.

As a prototypical PDE throughout this work, the nonhomogeneous heat equation

$$\partial_t u - \Delta u = f \quad (2.5)$$

is considered, where  $\partial_t$  denotes the temporal derivative,  $\Delta$  the laplacian operator and  $f$  is an arbitrary forcing function. In addition to Equation 2.5, one must supply a set of boundary conditions (BCs) as well as an initial condition (IC) to fully determine a solvable PDE problem. This equation is of high importance throughout physics and describes the evolution in time of some quantity  $u$  such as heat, chemical concentration and others. It is also known as the diffusion equation (Evans, 2010). In this case,  $f$  may act as a source term, e.g. providing either a heat source or sink within the domain. This equation always produces smooth and continuous solutions due to its diffusive character, even if the initial and boundary values that define the problem are nonsmooth (Evans, 2010).

It is commonly known that only a small subset of PDEs that have been discovered to date possess an obtainable analytic solution (Pinchover and Rubinstein, 2005).

One must instead for most problems rely on numerical approximations that involve the abovementioned discretization of the established model. However, obtaining a valid, stable and accurate discretization in general is ambiguous and nontrivial. Therefore, to form a baseline for this work, the most common techniques to discretize a PDE-based model are given in the following.

## 2.2 Common Grid Based Approximations

In this section, the most prevalent methods to solve PDEs based on a computational grid will be outlined alongside their most important properties. The focus of this outline is set on a problem-oriented and practical perspective to introduce the characteristic properties of each method.

At this point, it should be noted that the presented methods are only described in terms of computing spatially discretized systems. Some PDEs however also involve temporal derivatives, such as the heat or the wave equation. A very common method to deal with those terms separately is to only discretize the system of equations in space. This leaves the analyst with a large system of Ordinary Differential Equations (ODEs) which can then be solved using standard techniques for such problems. This procedure is called Method of Lines and will be employed throughout this work (Liskovets, 1965; Jones et al., 1972; Schiesser, 2012). In the case of the heat equation (Equation 2.5), one would first discretize the laplacian  $\Delta u$  and the right-hand side  $f$  at  $n$  discrete points using some spatial discretization technique. Afterward, one is left with  $n$  ODEs which can, for example, be integrated in time using the implicit Euler method

$$\frac{u_{n,i+1} - u_{n,i}}{\Delta t} - L[u_{n,i+1}] = f_{n,i+1}, \quad (2.6)$$

where  $n$  is the discrete spatial point of evaluation,  $i$  and  $i + 1$  are the previous and current time steps, respectively, and  $L$  is the discretized spatial differential operator (Pinchover and Rubinstein, 2005).

Alternatively to the Method of Lines, there exist other approaches to compute time derivatives, such as the space-time FEM as originally proposed by Argyris and Scharpf (1969). However, as such methods are tightly coupled to specific spatial discretization techniques, those do not apply universally and hence those will be omitted from the

following discussion.

### 2.2.1 Restrictions

Within the scope of this work, only methods that employ computational grids will be considered to solve a given set of physically sound PDEs in a valid, reliable and fully reproducible manner. In particular, this excludes mesh-free methods as well as non-deterministic methods, such as Machine Learning.

Mesh-free approximations tend to have good performance but lack physical substance in the way that they require artificial terms to stabilize a solution. While this typically produces systems that are fast and easy to evaluate, it also introduces an additional calibration of the model in practice that must be accounted for (Lind et al., 2020). Consequently, one encounters an additional stage in product development, more specifically in modeling which requires careful attention, expertise and resources and is hence undesirable.

The prominent advantage of mesh-free methods lies within the locality of the differential operator approximations. An instructive example of this is the Smoothed Particle Hydrodynamics (SPH) method. Here, integral kernels are used that only incorporate values from the immediate neighbors of an individual particle (Colagrossi and Landrini, 2003). This creates a strongly localized operator that leads to a high degree of parallelism due to few messages being passed between processors. That in turn makes the method very suitable for the use of highly parallel hardware architectures, such as GPUs (Lind et al., 2020).

Hence, careful tuning of pseudo-physical parameters is required to obtain a viable solution. They are therefore not suitable for solving arbitrary physical problems in a unified fashion. For example, when solving the mesoscopic flow field during Laser Powder Bed Fusion (c.f. section 3.2.3) using the SPH method, one must introduce artificial and arbitrarily set parameters, such as an artificial viscosity (Colagrossi and Landrini, 2003) or artificial speed of sound (Hu and Adams, 2006) to stabilize the model.

Machine Learning approaches, on the other hand, do not exhibit deterministic convergence behavior (Shin et al., 2020). In this context, some approaches exist that leverage neural network architectures as a means to achieve a purely synthetic solution to a PDE, i.e. without the need for training data (Raissi et al., 2019). However, to this date,

the convergence of set order in general cannot be guaranteed for all classes of PDE problems. Furthermore, obtaining solutions to optimization problems via stochastic algorithms always poses the question of reliability. This nondeterministic behavior however is generally unfavorable in the context of engineering applications, where solution strategies need to exhibit strong and uniform convergence. Otherwise, physical laws are not necessarily obeyed and e.g. structural integrity cannot be guaranteed.

It is therefore given by the principle of exclusion that this work will be focused on more classical numerical methods that operate on computational grids.

### 2.2.2 Finite Difference Method

Developed initially in the early twentieth century, the Finite Difference Method (FDM) can be regarded as one of the oldest methods to generate numerical solutions to PDEs as its original appearance dates back to the early 1910s ([Richardson and Glazebrook, 1911](#)).

It operates in a point-wise manner on a computational grid that should be spanned by an equispaced, cartesian coordinate system.

The guiding principle of the FDM lies in approximating differential terms by computing the slope of the desired field quantity at a given point with respect to its neighbors. In one dimension, this process closely resembles the definition of the derivative with the exception that the limit in the expression is simply never taken:

$$\frac{d}{dx}f(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x+h) - f(x)}{h} \quad (2.7)$$

It immediately follows that omitting the limit creates a finite-dimensional approximation of the derivative. One may alternatively view this type of approximation as a Taylor series expansion where higher-order terms are simply omitted. This idea is easily extended to functions of multiple spatial variables as will be shown.

By taking into account all immediate neighbors of a given point, a stencil is created for each location being evaluated. The order of the method can be adjusted by widening or narrowing the stencil such that more or less points in the vicinity are taken into account. [Figure 2.2](#) gives a visual impression of how such stencils are constructed. The resulting partial derivative expressions can consequently be used to approximate more complicated differential operators, such as the gradient, divergence or curl ([Smith](#)

et al., 1985). The formulation of the derivatives given in Figure 2.2 is a particularly popular choice as it is second-order accurate, i.e. the error of approximation decreases with  $\mathcal{O}(h^2)$ . Its construction is straightforward and is given by combining Equation 2.7 evaluated at the previous and the next vertex respectively and taking the sum of both terms.

If the differencing scheme is chosen carefully depending on the type of PDE, the FDM produces efficient, stable and accurate solutions. Its implementation can also be regarded as relatively straightforward. This is indicated by its simple construction of derivative expressions.

However, treatment of BCs can, for more complicated expressions, prove to be non-trivial (Smith et al., 1985). Another shortcoming of the method is the comparatively harsh restrictions on the problem geometry. If the domain cannot be approximated properly by a hypercube but instead has a more complicated, possibly curved shape, the method loses much of its computational efficiency. There are however some works that extend the method to also work with a wider scope of domains (Fornberg, 1988).

Concerning the approximation of functions, the FDM can be seen as a technique to generate linear interpolations to more complex solutions. This property is shown in

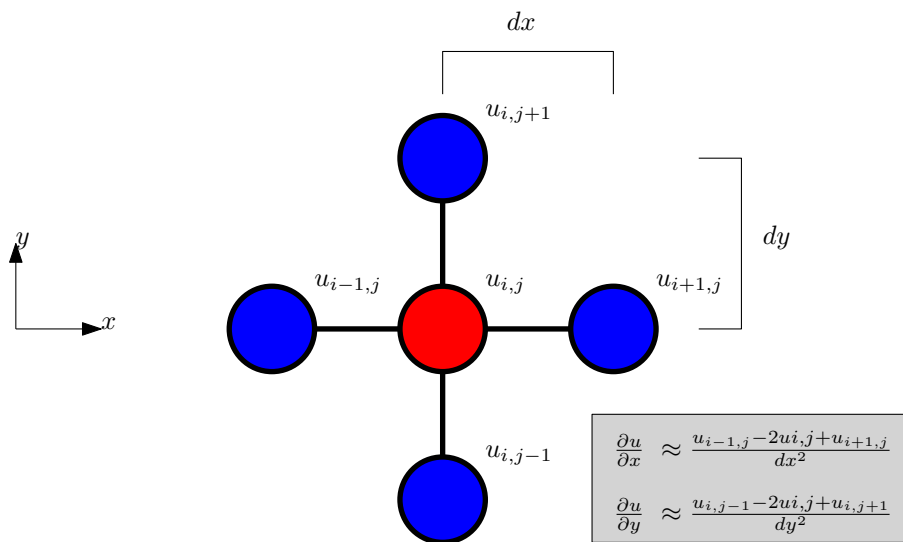


Fig. 2.2: Approximation of a two-dimensional derivative of the field  $u$  by a 4-point Finite Difference stencil. The neighboring points (blue) are arranged around the central vertex (red) in two cartesian directions and are equispaced in each direction. One can then construct the two partial derivatives using the central differencing scheme, where the formulae are given in the grey box. This scheme is second-order accurate.

Figure 2.3. At the discretized grid points, the solution approximation is of high accuracy whereas elsewhere considerably larger errors of the absolute value can be observed. The two functions shown in Figure 2.3 can, for example, be regarded as continuous (blue) and discrete (orange) solutions to the heat equation (2.5) at some given time.

### 2.2.3 Finite Volume Method

In contrast to the FDM, which can be regarded as a universal approximation scheme, the Finite Volume Method (FVM) was initially conceived with a rather specific purpose in mind, namely modeling conservation laws (Samarskii, 1965; Eymard et al., 2000). As such, this method excels in representing the temporal evolution of conserved quantities and is hence often used in Computational Fluid Dynamics.

To illustrate the guiding principle of the FVM, a simple scalar conservation law is

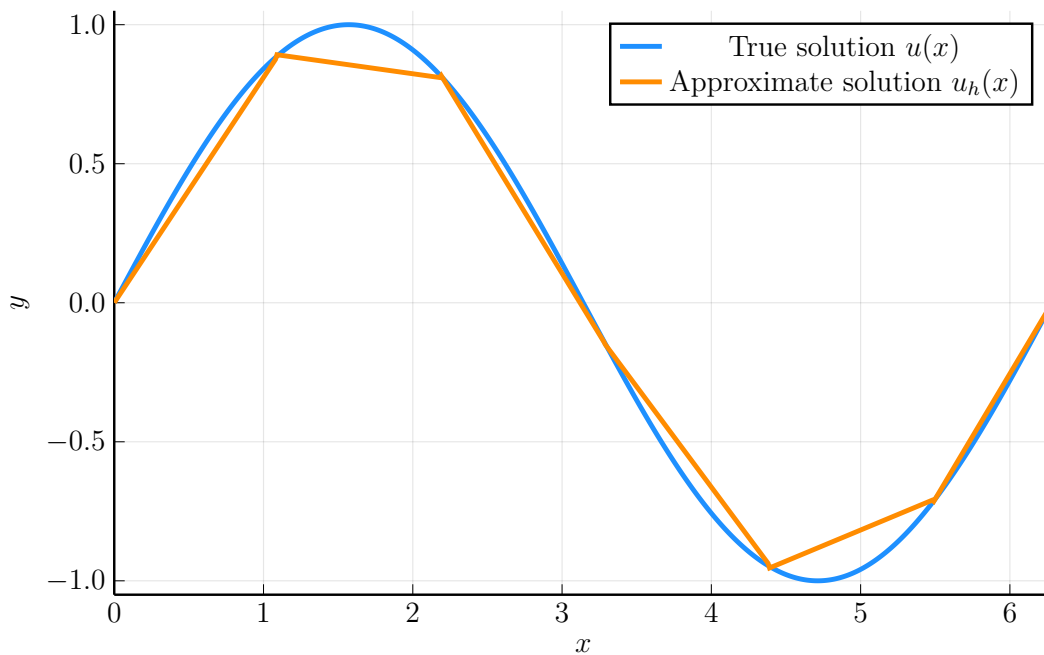


Fig. 2.3: Approximation of a first-order Finite Difference interpolation (orange line) to a one-dimensional function  $u(x) = \sin(x)$  (blue line) in an interval from  $[0; 2\pi]$  using 7 grid points. At these,  $u(x) = u_h(x)$  holds up to machine precision. At all other intermediate points, the solution is interpolated linearly.

considered (Versteeg and Malalasekera, 2007)

$$\partial_t u + \underline{v} \cdot \nabla u = 0, \quad (2.8)$$

or equivalently in integral form

$$\iiint_V \partial_t u \, dV + \iiint_V \underline{v} \cdot \nabla u \, dV = 0. \quad (2.9)$$

This simple first-order PDE describes the conservation of a passive scalar quantity  $u$  in a domain  $V$ , where  $\underline{v}$  denotes the convective velocity. Using Gauss' theorem and applying it to the second term in Equation 2.9 yields

$$\iiint_V \partial_t u \, dV + \iiint_V \underline{v} \cdot \nabla u \, dV = \iiint_V \partial_t u \, dV + \oint_{\partial V} \underline{v}(u \cdot \underline{n}) \, dS = 0. \quad (2.10)$$

As a result, the volume integral containing the gradient operator is now replaced by a simple hull integral over the domain boundary  $\partial V$ . Hence, the problem is reduced to finding the face values or fluxes  $\underline{v}(u \cdot \underline{n})$  of the conserved quantity without having to compute any derivatives. This procedure is illustrated for a two-dimensional domain in Figure 2.4. Cells with shared faces are marked blue, and neighboring cells that do not share common faces are drawn in grey. The volume integral over the (arbitrarily shaped) volume  $V_i$  of the  $i$ -th cell is replaced by taking the sum of the face values  $u_{i,j}$  that belong to the cell. The red-marked faces  $S_{i,j}$  are shared with the blue cells and hence contribute to the integrals of the adjacent cells as well. The face values  $u_{i,j}$  can in turn be interpolated from the cell values of the previous time step using multiple methods. Choosing an appropriate scheme for this step is crucial to generate a stable and accurate solution.

In its most basic form where the fluxes are calculated using simple interpolation of adjacent cells, the FVM is first-order accurate and unconditionally stable, but also generates very diffusive solutions. In terms of a conserved quantity, this means that there tends to be an artificial or numerical loss of the quantity within the domain. This is often regarded as unwanted and unphysical behavior. Therefore, extensive efforts have been put into developing stable, high order and conservative schemes (Eymard et al., 2000; van Leer, 1974).

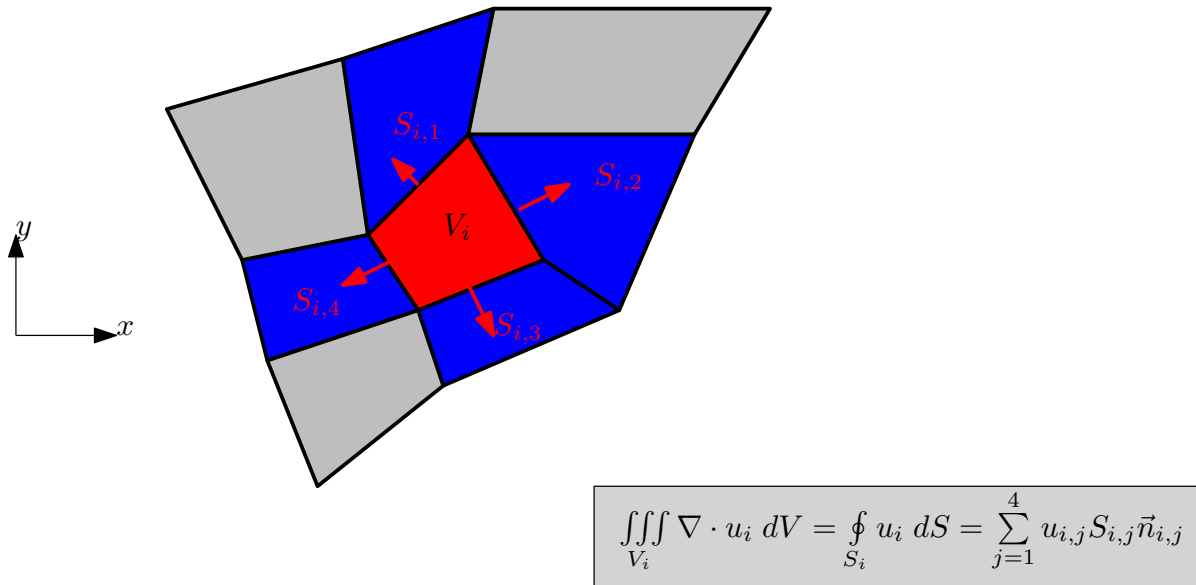


Fig. 2.4: Approximation of a two-dimensional divergence of the conserved quantity  $u$  within the cell  $i$  (red) using the FVM.

In the above discussion, it has been shown how to discretize the gradient operator. A very similar treatment applies to the discretization of other differential operators. Higher order derivatives are often discretized using a combination of the previously shown FVM and FDM (Eymard et al., 2000).

One key difference to the FDM that has already become apparent is that the FVM operates in a cell-wise manner instead of relying on point-wise evaluations. Additionally, due to its guiding principle, the method does not rely on any kind of regularity within the domain. The FVM can even handle cells with an arbitrary amount of faces, since once the fluxes are computed, the cell value reconstruction amounts to a simple sum of directed quantities.

From the perspective of function approximation rather than solving PDEs, the properties of the FVM are depicted by Figure 2.5. Regarding the error of absolute values, this method visibly introduces comparably large errors. Its main purpose instead is finding an approximation that matches the integral of the solution, which often relates to the conserved quantity, for a given cell as closely as possible. For clarity, vertical dashed lines have been added to better disambiguate the individual cells. The discontinuous nature of this method typically leads to rather large approximation errors regarding the solution itself for individual points. If Figure 2.5 is again considered to be a solution

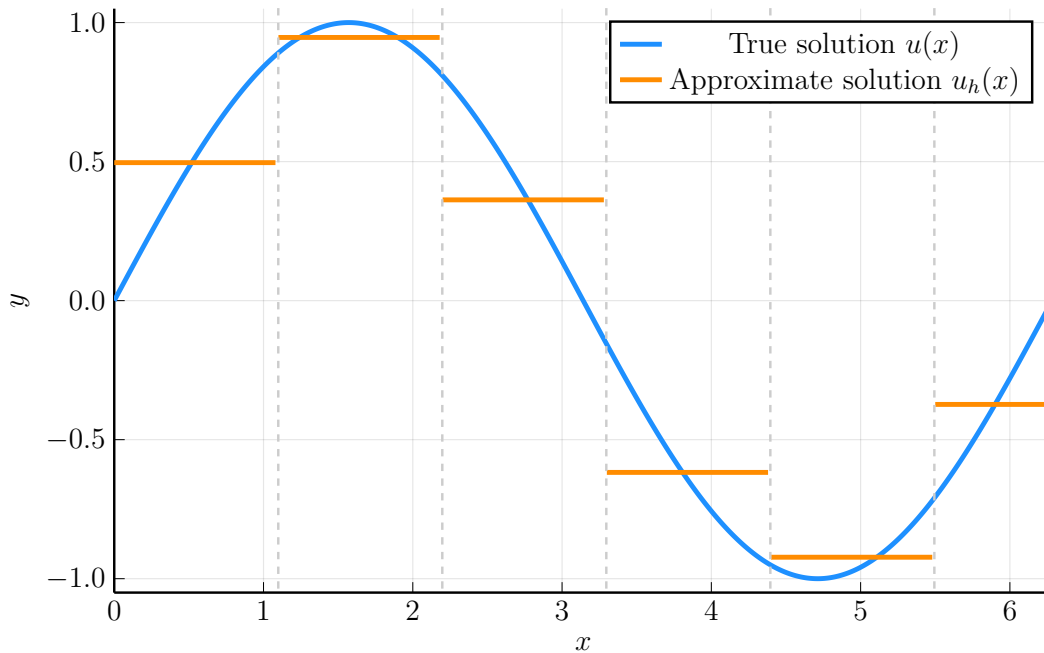


Fig. 2.5: Approximation of a Finite Volume interpolation (orange line) to a one-dimensional function  $u(x) = \sin(x)$  (blue line) in an interval from  $[0; 2\pi]$  using 6 cells.

of the heat equation (2.5), it is obvious that the inherently smooth character of the solution is poorly represented. However, if instead the quantity of interest is conserved, i.e. unphysical sources or sinks must not appear in the numerical solution, the FVM generates more accurate approximations in that sense. In the example of Figure 2.5, each cell value is generated such that the integral value of the approximation matches that of the true function as closely as possible. It also becomes apparent that the FVM is by design suited for approximating functions that may contain discontinuities since these appear naturally in the formulation of this method.

### 2.2.4 Finite Element Method

The primal formulations of the Finite Element Method (FEM) originate back to applications in solid mechanics and are ultimately influenced by several achievements from applied and pure mathematics over the last three centuries. A concise overview of the most influential works on that matter is given by [Zienkiewicz et al. \(2010\)](#). Similar to the history of the FVM, its purpose originally was driven by one specific application. However, over time, this method has evolved to be considerably more versatile as it and

its derivatives can be found in many different applications. Thus, the latter variation of this method will be outlined here, as it on the one hand may be more complex to set up and understand but on the other hand, gives vastly more room for possible applications.

The basic idea of the FEM is conceptually different from the previously presented schemes. Given a certain norm, i.e., a quantity measuring the magnitude of a function, the method is designed to minimize the error of the approximation measured in this norm. For the classic FEM, using an approximation that yields continuous solutions, this is oftentimes the globally defined  $L^2$  Norm

$$|u|_{L^2}^2 = \int_{\Omega} u^2 \, dx \quad (2.11)$$

or the  $H^1$  Norm

$$|u|_{H^1}^2 = \int_{\Omega} u^2 + (\nabla u)^2 \, dx = |u|_{L^2}^2 + |\nabla u|_{L^2}^2. \quad (2.12)$$

Hence, the basic idea does not involve any special manipulations of the differential operator expressions such that problematic terms can be avoided, as demonstrated for the FVM.

The solution approximation  $u_h$  to a field quantity  $u$  is generated by interpolating it into a discrete function space  $V_h$ , which is a subspace of the original function space  $V$ . This in turn is done by choosing an appropriate function basis  $\phi$  that spans a vector space. The evaluations of the basis functions are associated with the vertices of the computational grid. In combination with the norms given by Equation 2.11 and 2.12, these function spaces are formulated as sets of functions where a given norm is finite, e.g. the space  $H^1(\Omega)$  that contains functions that are defined over a domain  $\Omega$  and have well defined first derivatives (Larson and Bengzon, 2013).

FEMs that use such vertex-associated function bases are also called Nodal FEMs (Ern and Guermond, 2004). Figure 2.6 illustrates such an interpolation and its error for a one-dimensional function. Increasing the number of interpolation points, i.e. increasing the number of vertices will decrease the approximation error of this method. However, the FEM is not restricted to using linear interpolation as Figure 2.6 suggests. Like the FDM, discretizations of arbitrary order can be generated by increasing the order of the basis functions  $\phi$ .

The actual approximation of the PDE is done by evaluating the weak form of the equivalent problem on a discrete level. Conceptually, the weak form of a PDE can be thought of as seeking a function, belonging to a certain function space, that minimizes a certain functional. The exact form of this functional is then given by the PDE. The origins of this method stem from variational calculus, where e.g. one may seek a function that best describes the deformation of a membrane which in turn is governed by the minimal surface equation  $G(u) = \frac{1}{2} \int_{\Omega} |\nabla u|^2 dx dy$  (Pinchover and Rubinstein, 2005). Due to this minimization principle, the FEM does not yield an exact solution, but rather a member of the prescribed function space that comes closest to the exact solution.

Setting up the weak formulation of a problem involves forming the inner product with a so-called test function  $v$ , i.e. multiplying and integrating over the entire domain of interest  $\Omega$ . The resulting integral equation must then hold for all possible test functions belonging to the function space. Within the field of variational calculus, the test function is also oftentimes referred to as the infinitesimal variation  $\delta u$ . Alternatively,

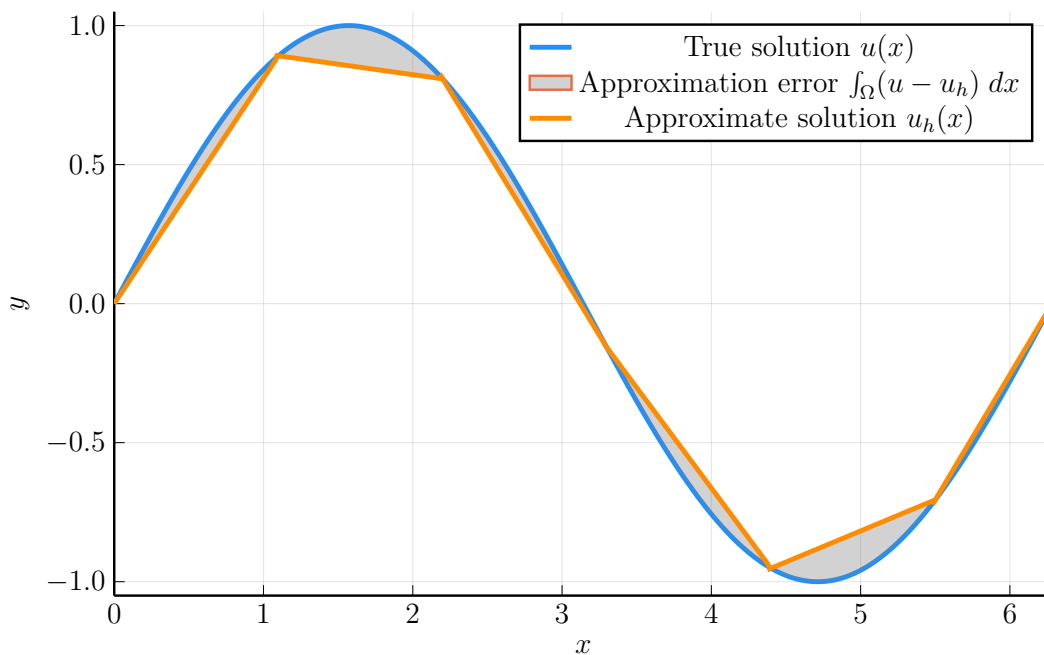


Fig. 2.6: Approximation of a Finite Element interpolation (orange line) to a one-dimensional function  $u(x) = \sin(x)$  (blue line) in an interval from  $[0; 2\pi]$  using 7 interpolation points. The method tries to minimize the error of approximation which is related to the area under the two curves (area with grey fill).

one oftentimes considers the first variation of a functional  $\delta G(u)(\psi)$  with respect to the test function  $\psi$  (Pinchover and Rubinstein, 2005). This work uses  $v$  to denote test functions for weak formulations. If the PDE solution  $u$  and the test function  $v$  belong to the same function space, the resulting scheme is called a Bubnov Galerkin FEM. Otherwise, i.e. if the function spaces are distinct from each other, the scheme is called a Petrov Galerkin FEM (Zienkiewicz et al., 2010).

In practice, seeking a PDE solution based on its weak form in this way coincides well with the restrictions that numerical approximations pose. Since numerical approximations also in most cases do not coincide with an exact analytical solution, solving a PDE using variational calculus naturally lends itself well to this problem.

These weak form functionals provide expressions that integrate the unknown function over the problem domain  $\Omega$ . Within the Finite Element context, the domain is further split up into single elements, as the name suggests. These elements in turn are defined by the vertices of the computational mesh. Integration of the functionals is then carried out on the element level, where each element gives a contribution to the interpolation function of its vertices. As one vertex is owned by multiple elements, these contributions must be accumulated into a global matrix. This process in the Finite Element context is also called assembly. Figure 2.7 illustrates this for a set of adjacent elements (a) as well as one single element (b).

As the previous discussion already suggests, the FEM poses a less trivial connection between the physical grid and the numerical approximation of a PDE. This warrants a more concise definition of what is referred to as a Finite Element. The most modern and arguably universal definition is given by Ciarlet:

**Definition 2.2.1** (Finite Element (Ciarlet, 2002)). A Finite Element in  $\mathbb{R}^n$  is a triple  $(K, P, \Sigma)$ , where:

- (i)  $K$  is a closed subset of  $\mathbb{R}^n$  with a non empty interior and a Lipschitz-continuous boundary,
- (ii)  $P$  is a space of real-valued functions defined over the set  $K$ ,
- (iii)  $\Sigma$  is a finite set of linearly independent linear forms  $\phi_i, 1 \leq i \leq N$ , defined over the space  $P$ .

In other words, a Finite Element contains a geometric shape of sufficient regularity as well as a set of functions that form a mathematical space to reconstruct the solution

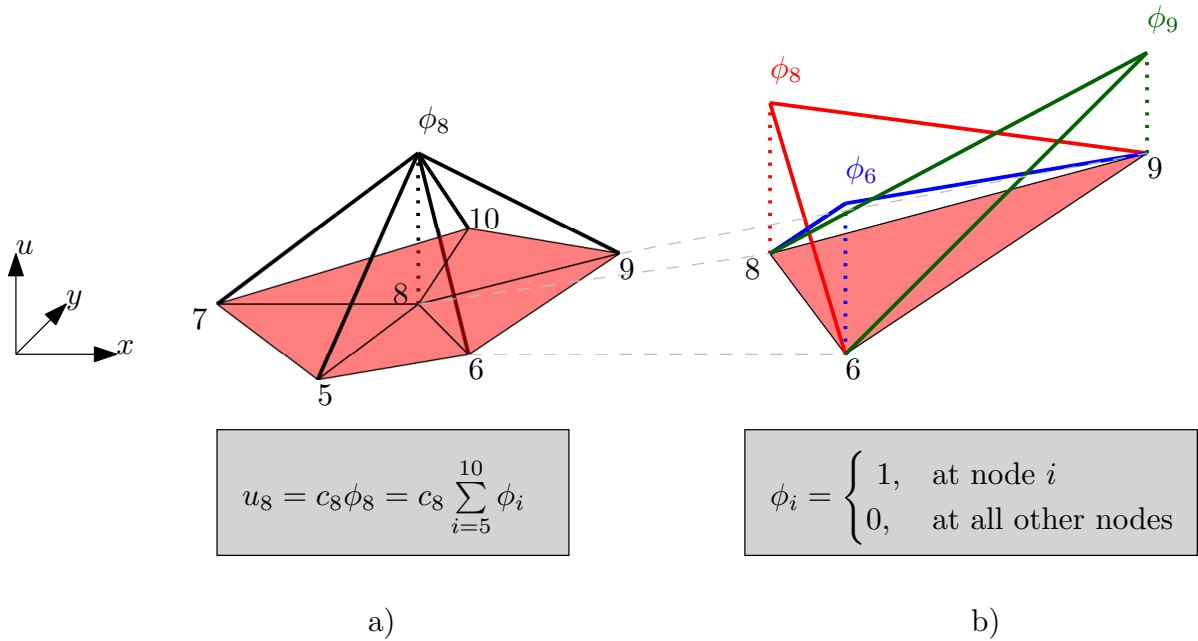


Fig. 2.7: Linear shape functions  $\phi$  in the FEM on a two-dimensional, triangular mesh. a) Composition of the shape function  $\phi_8$  associated with the 8th node of the mesh: Each element that contains node 8 as one of its vertices has a contribution to the global shape function. The function value  $u_8$  at this node is the product of the unknown scaling coefficient  $c_8$  and the global shape function, which in turn is the sum of all local contributions.

b) Illustration of the fundamental property of shape functions. At each element, the shape functions are only nonzero at the nodes that are associated with them. In this example,  $\phi_8$  is one at node 8 and drops to zero at adjacent nodes 6 and 9.

function  $u$ . As the basis vectors fully span the function space, arbitrary functions belonging to  $P$  can be reconstructed from the basis through linear combinations. Hence, the unknowns of a Finite Element discretization are simply scalar values that prescribe the appropriate contribution of each shape function to the global solution. The fundamental property of the function spaces shown in Figure 2.7 (b) is what enables the FEM to gain its efficiency. When evaluating weak form integrals of e.g. type  $\int_{\Omega} \phi_i \phi_j dx$ , only those contributions do not vanish in the entire domain  $\Omega$  that come from shape functions within one element. Otherwise, at least one of the terms will be zero, hence yielding an integral value of zero. This leads to typically very sparse global matrices that can be solved using appropriate linear algebra routines (Larson and Bengzon, 2013).

In principle, the FEM can operate on arbitrarily shaped domains and hence does

not share the same restrictions that the FDM imposes. However, in contrast to the FVM, there is a distinct set of primitive shapes, e.g. triangles and quadrilaterals in two dimensions, where shape functions on corresponding elements are defined ([Arnold and Logg, 2014](#)). In reality, this selection of primitives handily suffices to approximate domains of interest, yet it does pose some restrictions on the computational mesh.

Another part of the efficiency of this method is due to the weak form integrals not being performed on the computational mesh itself, sometimes called the physical domain. Instead, integration is only carried out once per Finite Element type on a well-defined reference element. Then, the integral value is mapped to the physical domain through a geometric transformation which oftentimes is a simple linear map. This way, costly numerical integration through Gauss quadrature can be kept to a minimum and instead be replaced by less complex geometric transformations ([Ern and Guermond, 2004](#)).

### 2.2.5 Discontinuous Galerkin Method

The Discontinuous Galerkin Method (DGM) is a numerical scheme that was originally proposed in 1973 to solve challenging hyperbolic transport equations in nuclear physics with high order ([Reed and Hill, 1973](#)).

Because the DGM is strictly speaking also a kind of FEM, the Classical FEM described previously in section 2.2.4 is often alternatively called Continuous Galerkin Method (CGM). This nomenclature shall be adopted here to disambiguate the distinct approaches more clearly.

The key difference between the CGM and the DGM is the sense in which the approximation error is minimized. It has been stated in the previous section that the CGM aims to minimize some global error norms. The DGM, however, minimizes the error norm on the element level which in some cases yields better approximations to certain problems. Such a strategy necessitates that shape function values do not need to be continuous across elements. In other words, the DG variational problem does not prescribe a global function space and thus a global function that solves the given problem. Instead, the function space is given on the element level ([Hesthaven and Warburton, 2008](#)). Otherwise, the continuity requirement would impose an additional restriction on the shape function values of adjacent elements hence constraining the minimization of the error norm. This effect can be observed by once again looking at Figure 2.6.

For some intervals, i.e. one-dimensional elements, the error integral is small in comparison, e.g. at the first and last element. However, in regions of larger changes of the true solution  $u(x)$ , the error is visibly larger. Figure 2.8 shows an approximation of the same function  $u(x)$  using the DGM. In contrast to the CGM, the aim is to minimize the approximation error in an element-wise fashion. One can observe the discontinuities in the solution at element boundaries. For clarity, vertical dashed lines have been added to better disambiguate the elements. It quickly becomes apparent that this variant of the FEM behaves differently regarding its approximation properties. First, one can already observe from visual comparison that the DGM can generate an approximation with a smaller error norm than the CGM for this configuration. However, the obvious drawback is that there are some points located at strongly discontinuous element boundaries, which makes the function value at this point ill-defined. Hence, this method produces solutions that are only continuous within an element, but not at boundaries. This behavior can be observed more clearly for dimensions higher than one in Figure 2.9. Another consequence of the more relaxed requirements on the solution compared to the CGM is that the treatment of differential operators also requires special attention.

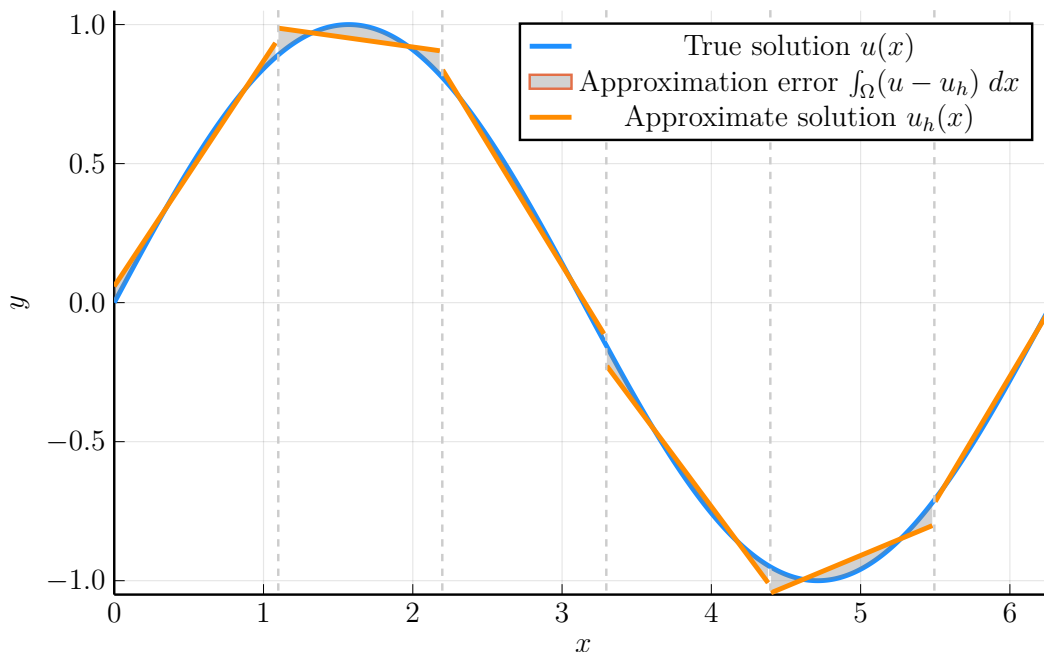


Fig. 2.8: DG approximation (orange line) of the same one-dimensional function  $u(x) = \sin(x)$  (blue line) as in Figure 2.6 using 7 nodes.

As Figure 2.8 suggests, the derivative of the solution at element boundaries is not well defined. Hence, it would not be possible to compute the necessary terms needed for assembly. This problem is typically circumvented by manipulating the weak form of the problem using partial integration (Larson and Bengzon, 2013). As a result, derivatives appear on the test function instead and in other parts sometimes vanish. Through this technique, however, additional boundary terms are introduced into the weak form. Normally, within the CGM formulation, hull integrals over element boundaries sum up to zero due to the continuity across elements. Thus the contributions from adjacent elements would cancel out each other exactly, as the integrals at common faces involve normals of opposite directions. In the case of the DGM, there are contributions of different magnitudes and hence those elemental boundary terms need to be evaluated, as shown by the different shape function values in Figure 2.9.

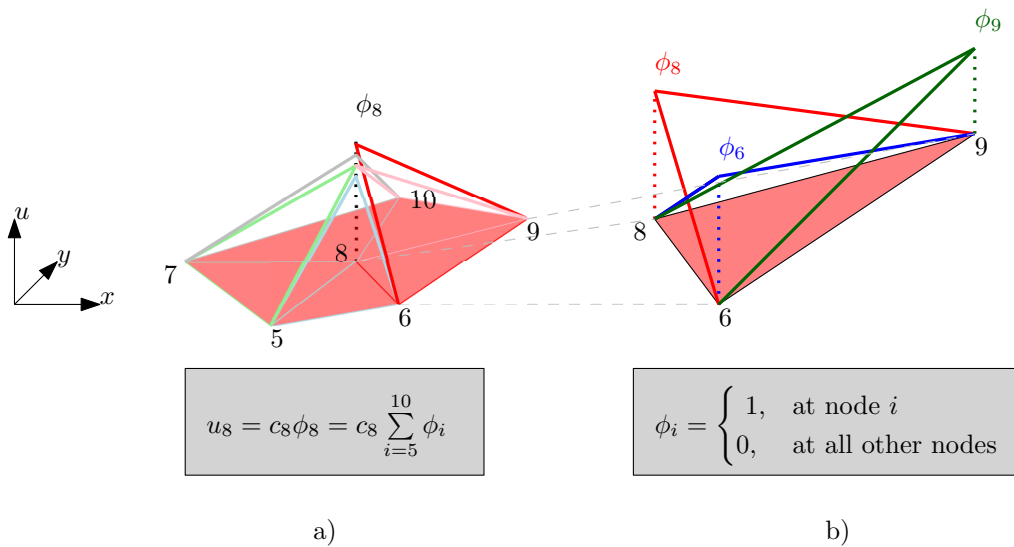


Fig. 2.9: Linear shape functions  $\phi$  in the DGM on a two-dimensional, triangular mesh.

a) In contrast to Figure 2.7, the local shape functions that contribute to the global, nodal function  $\phi_8$  need not be continuous at this point. As a consequence,  $\phi_8$  is not uniquely defined there. For better visual clarity, shape function contributions per element are marked in distinct colors.

b) Within one Finite Element, the same structure holds for both CGMs and DGMs: discontinuities only appear between element boundaries.

## 2.3 Discretisations of Multiphysics Problems

So far, the presented methods have been demonstrated based on comparatively simple PDEs that describe the evolution of one single field. Multiphysics problems on the other hand are comprised of multiple primary variables, where each in turn is governed by a PDE, yielding a system of PDEs. As each equation prescribes some kind of physical behavior that may differ considerably from the others, choosing an appropriate way to discretize and solve such a system is not trivial. As has been shown in previous sections, each numerical scheme possesses individual properties that suit different circumstances. These reflect the original problems that these schemes were intended to address.

If multiple governing equations are used to describe a more complex process, one has to deal with a multiphysics problem. In this work, the following formal definition is used:

**Definition 2.3.1** (Multiphysics Problem ([Keyes et al., 2013](#))). A multiphysics system consists of more than one component governed by its principle(s) for evolution or equilibrium, typically conservation or constitutive laws.

Employing one singular numerical method to solve a possibly very complex multiphysics problem may thus yield suboptimal results. This can be formulated as a consequence of a so-called no free lunch theorem, which states that a computational advantage for some type of problem is inherently offset by drawbacks when solving others. As a consequence, for an efficient solution, approximation methods need to be tailored to the operators that describe a given field ([Wolpert and Macready, 1997](#)).

## 2.4 Similarities between Approximation Schemes

With the numerical schemes presented in sections [2.2.2](#) to [2.2.5](#) in place, the question naturally arises whether these methods are entirely distinct from each other, or if there are certain similarities they share beyond them all being grid-based. In the literature, there are rather few works that are concerned with spanning the connection between different grid-based approximation schemes.

Some authors rigorously showed the equivalence of the FVM to either Mixed FEMs ([Baranger et al., 1996](#)) or Petrov Galerkin FEMs ([Ye, 2001](#); [Idelsohn and Oñate, 1994](#)).

With regards to the FDM and FEM, Thomée (1984) has shown early on that the FEM can be understood as a somewhat equivalent, yet generalized variant of the FDM on arbitrary grids. Some general differences between these schemes were outlined by Key and Krieg (1973). In the work of Shu (2003), some analogies have been brought up between FVM and FDM in WENO formulation. A theoretical and numerical comparison between higher-order FVMs and DGMs was conducted by Zhou et al. (2001). Additionally, Dumbser et al. (2008) constructed a unifying framework to accommodate high-order FVMs and DGMs. In the context of elliptic PDEs, Lin et al. (2015) present a theoretical and empirical comparison between the comparably new Weak Galerkin Method, DGM and Mixed FEM. A comparative study between DGM and the Streamline Upwind Petrov Galerkin method for flow problems can be found in Yurun (1997). These works in summary draw point-wise comparisons between some grid-based approximation schemes. Despite being quite useful for disseminating individual advantages and disadvantages for a given application, one may still lack an understanding of the general properties. Furthermore, Bui-Thanh has presented an encompassing analysis and application of the Hybridizable DGM (HDG) to solve a wide variety of PDE-governed problems. It was therefore shown that this numerical scheme is general and powerful enough to form a unified baseline (Bui-Thanh, 2015). In addition, due to the generality of this method, there have been works that attempt to benchmark DGMs to more conventional and widely adopted CGMs (Yakovlev et al., 2016; Kronbichler and Wall, 2018; Kirby et al., 2012). Some authors proposed combinations of numerical schemes that operate optimally to solve hyperbolic (Gaburro, 2021) or parabolic (Yang et al., 2017) systems of PDEs.

An encompassing summary and comparison of the presented numerical schemes to address arbitrarily characterized systems of PDEs does not exist to date.

Having a common baseline would, however, be beneficial to provide flexibility in solving multiphysics problems. As has been shown in the previous sections, the presented approximation methods possess rather distinct properties that, motivated by their origins, lend themselves to solving different kinds of problems. The prospect of being able to combine these methods in a unified framework can be regarded as an important contribution to the simulation of multiphysics problems. Thus, the question arises whether a common formulation can be found and, if so, which restrictions need to be posed. Addressing this question from a theoretical standpoint forms the foundation of

this work.



# Chapter 3

## Simulation of Laser Powder Bed Fusion

In this chapter, a brief description of the Laser Powder Bed Fusion process is given. Due to the highly complex material behavior during melting and solidification, a multi-scale problem typically arises, which will be outlined in brief. Furthermore, a literature review of the powder bed scale problem is provided and for this particular application of multiphysics problem, a research gap is identified.

### 3.1 Process Description

Powder Bed Fusion using a Laser Beam of Metals (PBF-LB/M) is a process of additive manufacturing, where a powder bed consisting of a metallic material is selectively fused using thermal energy. Additive manufacturing denotes the process of joining materials to make parts from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing and formative manufacturing methodologies ([ISO/TC 261, 2021](#)). This process of selectively melting material layer by layer is schematically illustrated in [Figure 3.1](#). The molten material thus encounters several phase changes, including melting, solidification as well as vaporization for a small fraction of the mass due to local overheating. Thus, a new microstructure is formed during the process that is influenced by the thermodynamic conditions present. During the build, one may encounter imperfections and impurities that affect the final mechanical properties of a manufactured part negatively. Such defects include delamination, i.e. large voids between layers, pores, lack of fusion between adjacent melt tracks and keyhole pores that appear due to excessive energy input and subsequent vaporization of material. However, even if a part

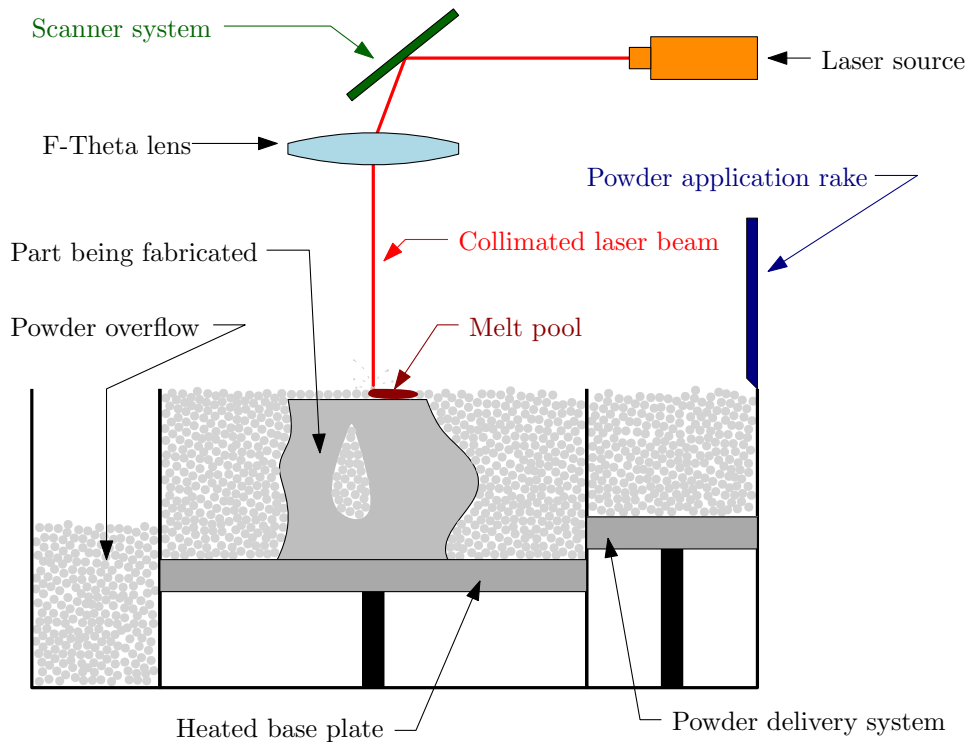


Fig. 3.1: Schematic overview of the PBF-LB/M process. Fresh powder is applied by moving a rake from a reservoir over the part, leaving a thin layer of unmelted powder. Through the laser beam, heat is applied to the powder, producing a melt pool and effectively welding the newly applied material to the already fabricated geometry. The scanner system directs the laser beam across a specified path until the entire layer has been exposed. The base plate is then lowered and a fresh layer of powder is applied (King et al., 2015b).

appears to be defect-free from a topographical perspective, mechanical properties may still be subpar due to an unfavorable microstructure. This problem may however be addressed by including heat treatment as a post-processing step (DebRoy et al., 2018).

### 3.2 Multiphysics Problems of Interest

The physics occurring during powder bed fusion is typically split into three separate length scales that are detailed in the following. The taxonomy coincides mostly with the quantities of interest, i.e. part scale defects, melt pool topography and microstructure.

### 3.2.1 Macroscopic Scale

Within the coarsest scale, typically the entire part to be manufactured is modeled, i.e. the viewpoint roughly coincides with the scale of Figure 3.1. In terms of the physics involved, mainly the thermomechanical response of the part is considered. Most simulation models approximate the topography of a single layer via a uniform grid in the build direction and use approximate material properties to account for the powder bed density. Then, a heat source approximating the laser beam travels along a given path within the layer. The thermally induced eigenstresses and eigenstrains are then calculated using the simulated thermal history in a postprocessing step. As such, the macroscopic model consists of a thermal model that is coupled with an elastoplastic material model. The typical length scale of such models therefore spans the region of 1 mm to 100 mm. The temporal length scale spans the real build time of a part and is within the order of 10 h to 100 h (Bayat et al., 2021).

As the thermal history is highly dependent on the geometry that is to be built, simulations need to be conducted separately for each layer of each part to accurately predict part deformation and stress state. Thus, commercial interest in applications is naturally high, resulting in widespread use so far.

The computational problem has been solved using Element-free Galerkin (Chen and Duan, 2020), Finite Difference (Ren and Wang, 2023), Finite Element (Krol et al., 2009) and Finite Volume methods (Wang et al., 2020) as well as meshfree approaches based on graph theory (Yavari et al., 2019) and machine learning (Yang et al., 2020). The latter uses a simple Artificial Neural Network that takes some process parameters as input and predicts the resulting melt pool width of a scan pattern over several layers. This can in turn be used to reconstruct defects on a part-scale level (Yang et al., 2020).

### 3.2.2 Microscopic Scale

An even finer discretization of the powder bed is given by the microscopic scale which aims to accurately predict the microstructure of a part. Here, one takes the thermal fields obtained by a larger, mesoscopic model as input to model the fluid-solid transition. The morphology of solidified grains is typically the primary quantity of interest, as this is mostly indicative of the mechanical performance of a part. As the process itself poses harsh spatial and temporal gradients in the vicinity of the melt pool, the microstructure

can also be expected to vary locally. Other input quantities to such a model depend largely on the physical models used for solidification. In the case of the phase field method, which is a popular choice, one requires several thermochemical properties of the material used, such as lattice diffusion coefficients, latent heat, crystal anisotropy, etc (Zimbrod and Schilp, 2021). The typical length scale of this process is around 100 nm to 10  $\mu\text{m}$ , the time scale is of order 1  $\mu\text{s}$  to 100  $\mu\text{s}$  (Yang et al., 2021). Other methods that have been employed to solve the computational problem include cellular automata (Acharya et al., 2017), crystal plasticity (Liu et al., 2021), kinetic monte carlo (Rodgers et al., 2017) and molecular dynamics (Zhang et al., 2018a).

Such models are also typically used for material qualification, to ensure that the process produces a usable microstructure. Figure 3.2 shows how such a model may give insight into how a microstructure forms under certain process conditions and how prolonged heating may contribute to grain impingement and coalescence.

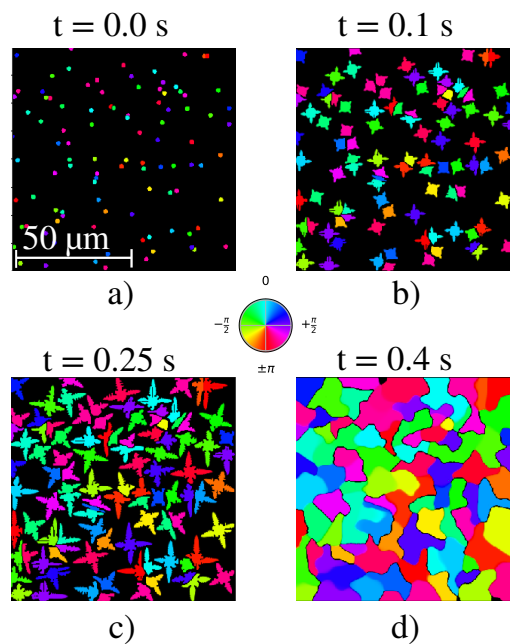


Fig. 3.2: Phase field simulation of the solidification process of a CoCrFeMnNi High Entropy Alloy during PBF-LB/M after different time steps. Colors denote different grain orientations. One can observe the originally small, circular grains (a) evolving with a dendritic growth pattern (b and c) as well as the subsequent grain impingement that forms the final microstructure (d). The model was simulated using a Finite Volume Method implemented in FiPy

### 3.2.3 Mesoscopic Scale

On the mesoscopic powder bed scale, typically one singular layer is considered and only one or very few melt tracks are modeled. Models on this scale are of especially high importance within the early qualification stages of AM processes, e.g. for novel alloys. In comparison to the other two scales presented, the mathematical problem that describes the physics here forms the most complicated and intricate one due to the number of PDEs that need to be solved as well as the coupling between equations. It is thus chosen in this work as a model problem to illustrate the amount of considerations needed to solve such a problem.

The quantities of interest for this problem mainly are the exact topography of solidified material, including the presence of defects, and the thermal history. For the former, one needs to accurately resolve the flow of fluid, molten material that is in turn largely governed by capillary effects (Zimbrod et al., 2022). Input quantities for this model are mainly the material-specific process parameters, i.e. laser power, laser scanning speed, etc. Furthermore, an exact discretization of the powder bed is needed that in turn depends on the particle size distribution of raw powder, which can largely vary. Lastly, one needs to ensure that accurate thermophysical material properties are used. As the temperature range that the metallic material encounters is typically large, these properties need to be variable with temperature to ensure accurate results. The typical length scale of this model is in the order of  $1\ \mu\text{m}$  to  $10\ \mu\text{m}$ . For a single melt track, a temporal scale of 1 ms can be assumed. An example of such a mesoscopic model that illustrates the length scales is depicted in Figure 3.3. White arrows denote the local flow field. In addition, the computational grid is marked with blue lines, showing the discretized powder bed. The temperature and flow field are calculated using a Finite Volume Method implemented in OpenFOAM. The powder packing as a function of particle size distribution has been calculated using the Discrete Element Method implemented in Yade. Overall, the purpose of this multiphysics model is to check if a set of material and process parameters produces a defect-free and dense structure. The relevance of this model is thus mainly given to material and process qualification. As this problem is investigated further in this work, the review of previous works is discussed in more detail.

To make an informed judgment on the prevalence and thus usefulness of each scheme

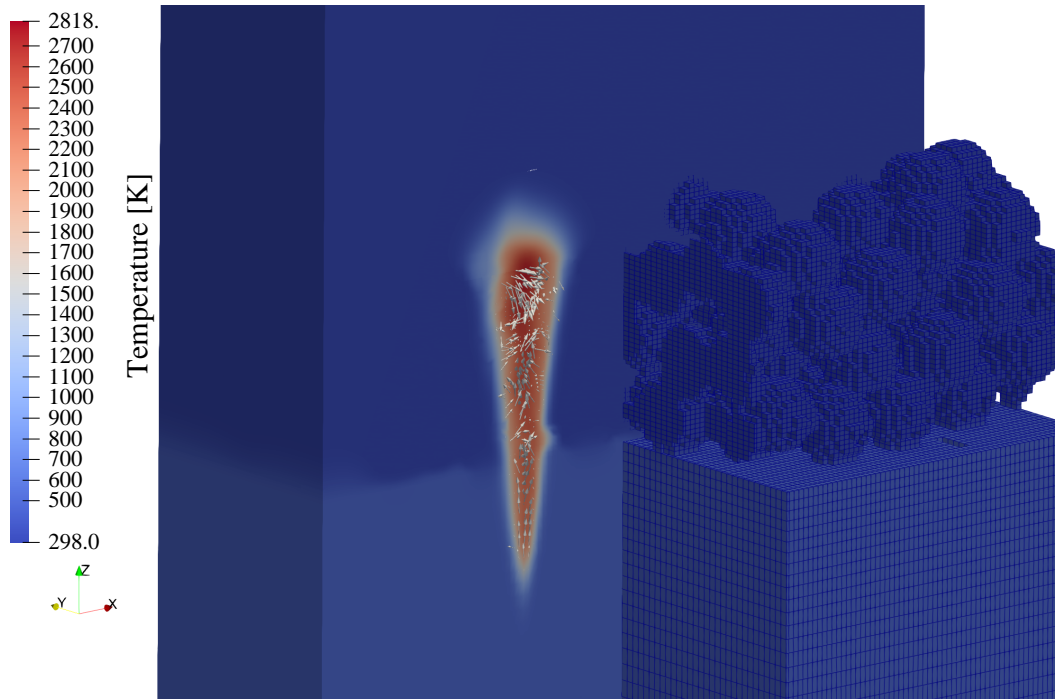


Fig. 3.3: Finite Volume model of the melting and solidification of 316L stainless steel during PBF-LB/M.

in practice, it is useful to evaluate the amount of works that implement or use a particular method to solve the problem. Thus, an extensive literature review has been conducted that should incorporate the most important works over mostly the past fifteen years. The results are illustrated in Figure 3.4. Some works employ or compare multiple methods and are thus counted and multiplied. In total, 177 publications were considered that have been published mainly within the last 15 years. Daggers<sup>†</sup> denote mesh-based (eulerian) approaches, asterisks<sup>\*</sup> signify mesh-free (lagrangian) methods. An exception is given by the Arbitrary Lagrangian-Eulerian method, which implements both material and field viewpoints. A quantitative summary including the list of references for each scheme is given in appendix Table A.1. As the physics on the mesoscopic scale is intricate and complex, involving multiple phases, the comparably large amount of schemes investigated over the years falls within expectation. This evaluation enables some key observations.

First, one notices that the numerical schemes used are conceptually very distinct from each other. That is, there are mesh-based and mesh-free methods as well as Lagrangian and Eulerian frameworks. This variety of methods leads to the conclusion that

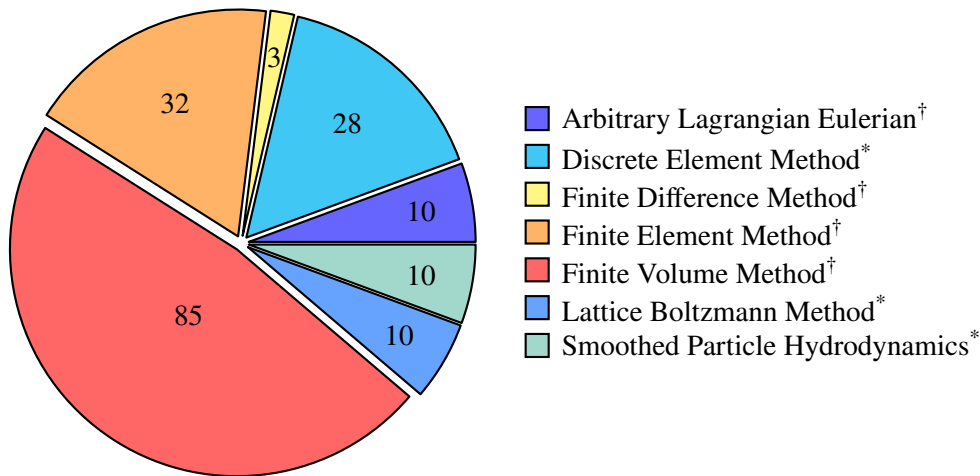


Fig. 3.4: Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M.

the modeling aspect of PBF-LB/M on this scale is not trivial, as the research community does not seem to be in universal agreement on which models to use.

Second, there does not appear to be a clear choice of method that is rendered superior to all others in a quantitative sense. However, the Finite Volume method seems to have received notably more research interest.

Consequently, it is worthwhile to investigate the latter observation further and to answer the question of why the FVM is the most prevalent scheme for modeling PBF-LB/M on the mesoscale. Thus, another study was conducted that separates the 75 works given in Table A.1 regarding the software that was employed within the publications. For the remaining ten publications, no information on the software used was available. A graphic summary is shown in Figure 3.5. Commercial software is annotated with an asterisk<sup>\*</sup>, academic and open source software with a dagger<sup>†</sup>. The full list of references for each software is given in appendix Table A.2. It becomes apparent that commercial software is in sum almost equally widespread as academic and open-source software. Thus, even within the FVM being the most widespread scheme, there is no apparent state of the art that is prevalent within the academic community. However, if compared with the amount of industrial-grade software that is available for the other methods given in Table A.1, one finds that the Finite Volume Method appears to be the most industrially mature and widespread scheme. The most used software in this case, OpenFOAM, albeit an open source code that requires substantial programming

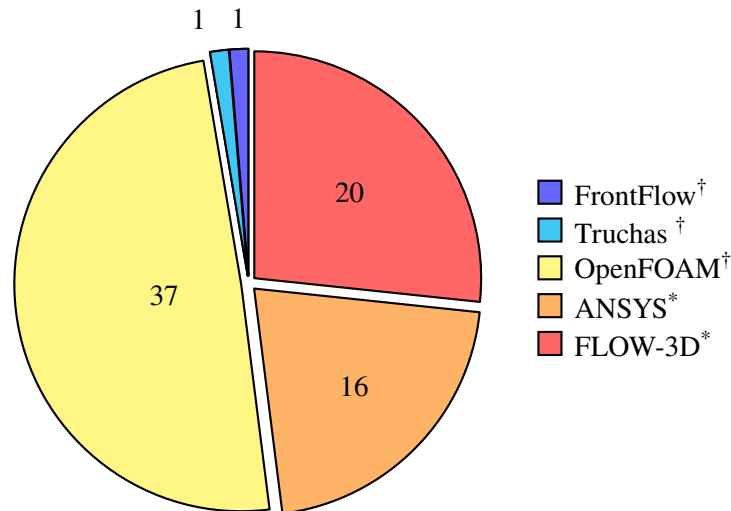


Fig. 3.5: Quantitative summary of previous works that use the Finite Volume Method to resolve melt pool dynamics in PBF-LB/M.

capabilities, has shown to be capable of fulfilling the needs of academic and industrial research alike (Jasak, 2011; Shademan et al., 2013). This in combination with its ease of implementation is consequently hypothesized as the reason why FVM is the most widely used method.

However, regarding the performance aspects, FVM-based models are far from the most performant in comparison with the rest. For example, Ninpetch and colleagues report run times for their FV model based on FLOW-3D between 8 and 24 hours for reasonable mesh resolutions on a 28-core machine (Ninpetch et al., 2021). In contrast, other authors report implementations of the mesoscale model based on adaptive SPH that can calculate the melting of a single track in the order of minutes on workstation grade hardware (Fürstenau et al., 2021; Lüthi et al., 2023).

Introducing model accuracy into the discussion, FVM as well as SPH do not show optimal behavior. The relatively superior performance of SPH comes at the cost of introducing artificial quantities that need to be tuned to stabilize the simulation which considerably hinders reproducibility and accuracy if those parameters are ill-defined. The FVM models on the other hand are severely bound by the order of accuracy that they can achieve due to inherent restrictions of the model. There are ways to circumvent this, e.g. by introducing very wide reconstruction stencils for the conserved quantities. This, however, increases the computational cost considerably. Furthermore, the

multiphase models introduced in this formulation increase model complexity and arithmetic intensity of the solution process (Zimbrod et al., 2022). Regarding this aspect, some other methods perform notably better, such as the Arbitrary Lagrangian-Eulerian (ALE) method. Due to the combination of material and field viewpoints in the model, the fluid behavior can be captured much more naturally without the need for artificial stabilization. Though this method enables exceptionally accurate prediction of physics, the method itself is vastly more complex to set up, thus increasing implementation effort (Khairallah et al., 2020).

### 3.3 Research Gap

With the PBF-LB/M mesoscale model as a prototype for a complex multiphysics problem, one may thus conclude that the question of how to resolve such problems in an accurate, performant and reproducible manner remains largely unsolved. Compared to the macro- and microscopic scales, the landscape of methods to solve this coupled multiphysics problem is rather large, which forms another reason why it is chosen as a model problem in this work. It appears that model accuracy, performance, as well as ease of implementation, form a magic triangle in this regard. The question of to what extent this problem can be addressed thus serves as a blueprint for this work.



# Chapter 4

## Scientific Methodology

This chapter will serve as an outline of the forthcoming content that contains the central statements and results of this work.

Beforehand, a thorough investigation of the scientific approach is needed. In the following, a concrete formulation of this problem is introduced, how it is formalized using an overarching research question and more granular hypotheses and which definitions and restrictions need to be posed.

### 4.1 Classification and Problem Statement

It has been outlined that there currently is a lack of reliable methods to guide an expert with domain knowledge by selecting a numerical method tailored to a specific application. Chapters 2 and 3 show that this gap in the current field of research is both of theoretical nature, but also of high practical relevance for complex, real-world multiphysics problems such as the mesoscopic Laser Powder Bed Fusion problem. The duality of this scientific approach - inductive and deductive through formal analysis and subsequent derivation - will be considered in the structure of this work. The overarching research question that shall thus be answered is formulated as follows.

**Research Question.** How can an accurate, stable and performant numerical method for a given multiphysics problem be derived in a systematic, reproducible and viable manner?

To conclusively answer this question, the following parts of this work are separated into three major parts that consecutively build upon each other. Each section will be

concerned with addressing a distinct research hypothesis that makes up parts of the research endeavor.

## 4.2 Formal Analysis

The first part of this work is dedicated to whether or not it is possible to even identify a numerical method that particularly well suits a given multiphysics problem. Thus, it is initially required to form a common baseline for numerical schemes, as outlined in section 2.4. The proposition that such a baseline exists is formulated in the first research hypothesis.

**Hypothesis 1** (Uniqueness and Comparability). In general, there exists a combination of grid-based approximation methods that approximates the solution of a given multiphysics problem in a stable and computationally efficient way.

In this context, a method is considered computationally efficient if, from an algorithmic standpoint, there are no steps that can be trivially skipped to solve a given problem. A practical example of an inefficient algorithm would be a geometric transformation on a computational mesh that only applies constant scaling. Then, the transformation reduces to multiplying with a constant scalar which can be done in a single instruction cycle, instead of performing many matrix multiplications. In addition, the solution produced by the method should reflect the qualitative behavior of the true solution.

For a combination of grid-based approximation methods to be able to exist, they must be comparable as well as interoperable with each other. To verify that this is the case is an important part of this hypothesis.

Thus, it is necessary to unify the solution space of numerical schemes under a common denominator. Deriving such a common scheme will be an integral part of the following section. As this procedure is inductive by design, formal analysis will have to be carried out, as empirical methods in scientific theory are not able to conduct inductive reasoning by definition (Popper, 1989; Lakatos, 1976). For such formal analysis, some mathematical work will be needed to construct a sound framework. In this case, large amounts of prior relevant work can be found in the literature. Some prior point-wise comparisons that have been outlined have been discussed in section 2.4. The main

task is thus to systematically arrange previous results and to restrict the mathematical landscape where necessary for statements to be valid.

### 4.3 Deduction of a Decision Scheme

In the subsequent section, the common mathematical scheme will be utilized to deduce a specific combination of methods for a given set of parameters in the form of a Multiphysics Problem.

The difficulty of deriving such a methodology mainly consists of distinguishing between and deciding on a set of numerical schemes. Beforehand though, it is crucial to first ask whether it is at all possible, given the restrictions discussed, to make such a decision unambiguously. If such a decision cannot be made, the solution space of methods must be narrowed down appropriately. Additionally, deciding on an efficient scheme should be a reproducible process. As reproducibility is strongly dependent on prior knowledge and thus the targeted user, this needs to be considered as well. Thus, the so-called application expert is introduced. This term denominates an end user who is familiar with the desired problem to be solved, along with the relevant physical phenomena and modeling aspects, but not with the necessary mathematics and numerics. The resulting formulation for the second research hypothesis can then be summarised as follows:

**Hypothesis 2** (Decidability and Feasibility). Given a specific multiphysics problem, one can identify the most suitable numerical method in a systematic and for an application expert reproducible way, such that the process of finding this method can be strictly formalized.

### 4.4 Application and Validation

With some sort of common baseline as well as an unambiguous decision metric in place, the derived method can then be applied to real-world problems. Thus, in the third part of this work, the usefulness of the results to the mentioned application experts should be outlined. Therefore, the findings of the previous chapters will be demonstrated on a real multiphysics problem. In this case, the previously discussed mesoscale problem during

PBF-LB/M will be considered as a representative of a complex and coupled problem with high practical relevance. The question of choosing an appropriate scheme for this problem, as outlined in chapter 3, continues to be an open field of research and is thus well suited to demonstrate how such a decision method can contribute to a current problem.

By deductively evaluating the stated methodology, the usefulness, especially in comparison with works from the literature that have previously investigated the model test case will be demonstrated. The final hypothesis for this work is thus formulated as:

**Hypothesis 3** (Applicability and Usefulness). The previously developed methodology is fully applicable to the mesoscopic PBF-LB/M model problem and consequently delivers a numerical method that approximates the solution of the problem efficiently.

Upon comparison between the derived scheme and those from the literature, the focus will lie on accuracy, performance and reproducibility, as these aspects have already been discussed in the context of other schemes in section 3.2.3. The performance aspect is especially important for the usefulness of the scheme, as it dictates computational efficiency on current hardware and thus the viability of such simulations in practice.

# Chapter 5

## Formal Characterization of Discretisation Methods

The general procedure of this chapter is as follows: First, the DGM being the most general scheme considered here is introduced. Then, for each additional scheme considered, the necessary simplifications to recover that numerical method from the DGM are worked out. In the remaining sections of this article, underline notation ( $\underline{\cdot}$ ,  $\underline{\cdot}$ ) is used to indicate vectors and matrices and roman indices ( $i, j$ ) to denote elements of lists or arrays on the computational level.

### 5.1 Discontinuous Galerkin Method

In spirit, the DGM can be held as a synthesis of Finite Element and Finite Volume schemes, and poses a generalized variant of both, as will be shown.

To derive such a scheme, the strong form of a given PDE is considered. The most straightforward example in this case would be a first-order linear advection equation with a homogeneous von Neumann boundary condition:

$$\partial_t \alpha + \underline{u} \cdot \nabla \alpha = 0, \quad \frac{\partial \alpha}{\partial n} = 0 \quad \forall x \in \partial\Omega \quad (5.1)$$

where  $\partial_t$  signifies the temporal derivative,  $x$  is the set of spatial coordinates,  $n$  is the outwards-pointing domain boundary normal and  $\partial\Omega$  denotes the boundary of the computational domain  $\Omega$ . As outlined in section 2.2.4, evaluating a PDE using some kind of FEM requires establishing the corresponding weak formulation first. In the case of

Equation 5.1, it can be constructed by multiplying with a test function  $v$ , integrating over the entirety of the domain  $\Omega$ , and applying partial integration to the second term on the left-hand side that contains the nabla operator. This results in the following formulation (Hesthaven and Warburton, 2008): Find  $\alpha \in V$  such that

$$\int_{\Omega} v \partial_t \alpha \, dx + \int_{\partial\Omega} v \alpha (\underline{u} \cdot \underline{n}) \, ds - \int_{\Omega} \nabla v \cdot \underline{u} \alpha \, dx = 0 \quad \forall v \in W, \quad (5.2)$$

where it is necessary to make an appropriate choice for the solution space  $V$  and the test space  $W$  which may but do not need to differ from each other.

Due to partial integration, there now is an additional term that has to be integrated over the domain boundary  $\partial\Omega$ , where  $(\underline{u} \cdot \underline{n})$  denotes the velocity component normal to the boundary.

To make such a problem solvable by a computer, one must additionally choose the discretization of the solution space  $V$ , denoted  $V_h$ . A particularly popular choice of space is the set of Lagrange polynomials which form an orthogonal function basis. In addition, the physical space must be discretized in the form of a triangulation. The DGM then consists of assembling the finite-dimensional, linear system on the element level. This enables high locality of the solution process, which leads to efficient computation on parallel architectures as less data transfer is required.

One resulting key feature of the DGM is that the elements now do not overlap anymore in terms of their degrees of freedom. Thus, the global problem is broken up into individual problems. This in general leads to large systems that are however sparse and in the case of the mass matrix, i.e. the form  $\int_{\Omega} u v \, dx$ , even block diagonal. The remaining term, often denoted the numerical flux, is the only term within the physical domain that ensures coupling across elements. Through evaluation of this surface integral, adjacent degrees of freedom are weakly coupled and thus global conservation of quantities can be assured.

As the polynomial space of DGMs only belongs to the  $L^2$  space of functions but not  $H^1$ , the basis functions are discontinuous and thus the derivative at boundaries is not well defined. Solving PDEs involving second derivatives is thus not possible as is. As a consequence, there have been many successful extensions of this method to circumvent that problem: The most prevalent schemes in this category are the Symmetric Interior Penalty (Epshteyn and Rivière, 2007), Hybridizable (Warburton and Karniadakis,

1999) and Local DGM (Cockburn and Shu, 1998). These methods, despite having different approaches, have been extensively studied and compared to each other (Arnold et al., 2002). As it turns out, all methods work well and have individual advantages and disadvantages. For this work, the Hybridizable DG (HDG) scheme is chosen as a general framework. It should be noted here that the proposed method would however also work with the other schemes given above (Arnold et al., 2002).

Within the abovementioned methods, one introduces an additional term in the weak form that serves as a penalty for discontinuous solutions. An alternative approach that is also pursued within the HDG scheme is the algebraic manipulation of the PDE system by splitting. One recursively introduces new dependent variables for quantities that appear in higher-order derivatives such that each quantity is differentiated at most once. This can be illustrated using the Poisson equation

$$\Delta u = 0. \quad (5.3)$$

The corresponding, well-known weak form is: Find  $u \in V$ , such that for all  $v \in V$

$$\int_{\partial\Omega} v(\nabla u \cdot \underline{n}) \, ds - \int_{\Omega} \nabla v \nabla u \, dx = 0. \quad (5.4)$$

By introducing the auxiliary variable  $\sigma = \nabla u$ , Equation 5.4 is extended to the following system:

$$\int_{\partial\Omega} v(\sigma \cdot \underline{n}) \, ds - \int_{\Omega} \nabla v \underline{\sigma} \, dx = 0 \quad (5.5)$$

$$\underline{\sigma} = \nabla u \quad (5.6)$$

Employing such an approach enables splitting PDE systems of arbitrary order resulting in larger systems of first-order PDEs (Hesthaven and Warburton, 2008).

## 5.2 Continuous Galerkin Finite Element Method

The most straightforward step to conduct is to derive the CGM from the DGM, although the former formally being the original formulation used to solve problems in structural

mechanics (Liu et al., 2022).

In this case, all degrees of freedom (DoFs) in the domain are global, in contrast to being local to each cell. However, each basis function associated with a given degree of freedom has compact support and is thus only non-zero within the direct vicinity, that is, at cells that the degree of freedom is associated with. The resulting linear system hence remains sparse but has considerably fewer DoFs than an equivalent discretization produced by a DGM.

One may obtain a CGM starting from the DGM by strongly coupling the degrees of freedom at cell interfaces. In other words, the previously discontinuous approximation must be made continuous. In terms of the weak form of a given problem, the numerical flux that has been introduced by partial integration has to vanish. This step is exactly taken in deriving weak forms for the CGM. The equivalent weak form of the advection equation given by Equation 5.2 is then: Find  $u \in V$  such that

$$\int_{\Omega} v \cdot \partial_t \alpha \, dx - \int_{\Omega} \nabla v \cdot \underline{u} \alpha \, dx = 0 \quad \forall v \in V \quad (5.7)$$

By coupling coinciding DoFs, one may equivalently introduce shared DoFs between cells. This results, compared to a DGM discretization, in a smaller global system that is in turn more coupled, yielding more non-zero entries per row and column in the system matrices. The condensation of such a system by coupling DoFs is illustrated in Figure 5.1. In the latter case, DoFs are entirely local to the cell and thus receive no contribution from neighboring cells. Weak coupling is only introduced by the additional numerical flux. Coupled DoFs are drawn in identical colors. From the numbering of DoFs in both figures, it becomes apparent that the amount of additional allocations grows drastically when the dimensionality of the problem is increased. As the numerical flux is zero by definition for a CGM, it may also be omitted from the computation. Thus, the CGM is noticeably less arithmetically intensive in this regard. However, this computational saving is offset by the strong coupling of DoFs, resulting in a more dense linear system and possibly a more complex assembly process in terms of memory management.

As equivalence can be shown here based on the weak form and thus early on in the model assembly process, the choice of Finite Element is unaffected. This in consequence also applies to the chosen type of triangulation or the order of approximation.

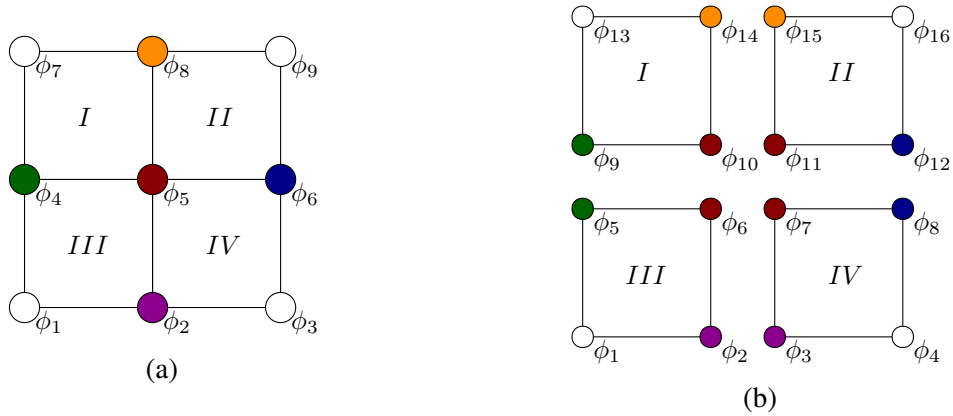


Fig. 5.1: Coupling of global DoFs in the CGM (a) versus DGM (b), both of first order.

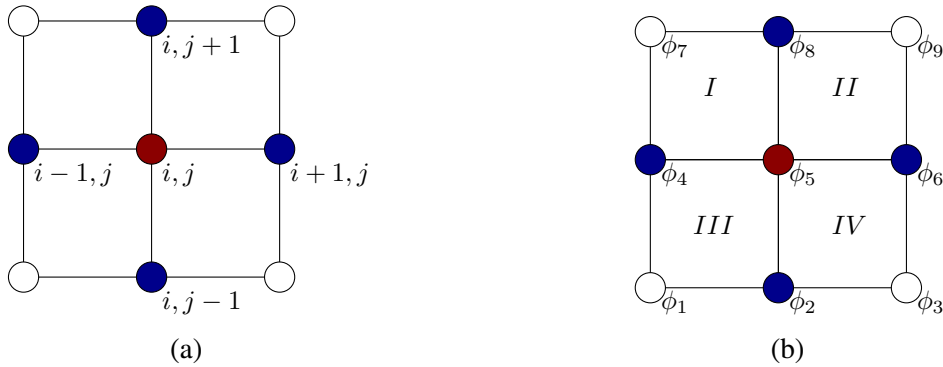


Fig. 5.2: Comparison of the nodal nature of the FDM (a) versus the cell-wise assembly used in the CG FEM (b) for an identical, cartesian triangulation with 9 nodes.

### 5.3 Finite Difference Method

At first glance, the FDM might appear to be conceptually different from the FEMs given above. Instead of treating the discretized problem in an element-wise manner, the FDM operates on discrete points directly and lacks a notion of cells in the domain. Yet, both methods still may yield identical results in discretization. A comparison of both approaches is shown in Figure 5.2. Both methods are formulated as first-order approximations. Nodes colored in dark red signify the points where the PDE is evaluated. Contributions to this node are taken from blue nodes, whereas white nodes from no contribution. In both figures,  $i$  and  $j$  denote vertical and horizontal indices of grid nodes,  $\phi_i$  are the FE basis functions and Roman letters denote indices of cells. Forming for example the laplacian for an FDM requires access to the vicinity of vertex  $i, j$  (red node) in all cartesian directions (blue-colored nodes). For both methods, gray-marked

nodes do not pose a contribution to the value of the central node. For a special case of FEM with quadrilateral elements, the same nodes form contributions to the global basis function  $\phi_5$ . However, the laplacian operator now is not evaluated directly instead contributions are gathered from weak form integrals. In the case of  $\phi_5$ , these contributions are from cells  $I$  to  $IV$ .

However, one may still show the equivalence of CGM and FDM by investigating the resulting global linear system. The second-order central stencil shall be used here as an example that is used to approximate a laplacian operator in two spatial dimensions

$$\Delta u \approx \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2}, \quad (5.8)$$

where  $h$  denotes the grid spacing that in this case is equal in each cartesian direction. Such a system in stencil notation will produce a global matrix with main diagonal values 4 and four off-diagonals with entries 1.

For a CG Finite Element scheme to be formally equivalent, the global system matrix is required to be exactly equivalent to the FD formulation. The required weak laplacian in CG formulation can be formulated as (Pinchover and Rubinstein, 2005): Find  $u_h \in V$  such that for all  $v_h \in V$

$$\int_{\Omega} v \Delta u \, dx = \int_{\partial\Omega} v (\nabla u \cdot \underline{n}) \, ds - \int_{\Omega} \nabla v \nabla u \, dx = - \int_{\Omega} \nabla v \nabla u \, dx. \quad (5.9)$$

In this case, the additional restriction that trial and test space be identical has been introduced. As outlined in section 2.2.4, this corresponds to a Bubnov Galerkin method. Now, let  $\Omega$  be an identical triangulation to the FD variant using quadrilateral  $\mathbb{Q}^1$  elements, that is linear Lagrange elements.

Then, the four basis functions spanning the reference element are (Arnold and Logg, 2014):

$$\phi_1(x, y) = xy - x - y + 1, \quad (5.10)$$

$$\phi_2(x, y) = x(1 - y), \quad (5.11)$$

$$\phi_3(x, y) = y(1 - x), \quad (5.12)$$

$$\phi_4(x, y) = xy. \quad (5.13)$$

The Finite Difference stencil given by Equation 5.8 only takes into account contributions from nodes that lie strictly horizontally or vertically from the node of interest. As a consequence, the node on the reference quadrilateral that is positioned diagonally from the center node must not have any contribution to the weak form integral, otherwise the resulting linear system cannot be equal. The weak form thus needs to be evaluated in a way such that the resulting matrix  $\phi_i \cdot \phi_j$  becomes sparse. It turns out that this can be achieved by choosing a collocation method for quadrature. This means that quadrature points are chosen to coincide with the node coordinates. As a consequence, the mass matrix  $\phi_i \cdot \phi_j$  becomes the identity matrix. From the family of Gaussian quadrature schemes, one can achieve this using a Gauss-Lobatto quadrature of equal order to the polynomial order of the Finite Element.

Now, the element-wise stiffness matrix  $-\int_{\Omega^{(e)}} \nabla_k \phi_i \nabla_k \phi_j \, dx$  is evaluated within the reference domain  $[0; 1] \times [0; 1]$  for the given first order Lagrange element using first-order Gauss-Lobatto quadrature. This results in

$$\underline{\underline{K}}^{(e)} = \begin{bmatrix} -1 & 1/2 & 1/2 & 0 \\ 1/2 & -1 & 0 & 1/2 \\ 1/2 & 0 & -1 & 1/2 \\ 0 & 1/2 & 1/2 & -1 \end{bmatrix} \quad (5.14)$$

As such,  $\underline{\underline{K}}^{(e)}$  does not yet equal Equation 5.8. The final step consists of assembling the linear system in the physical domain using the reference stiffness matrix. In a cartesian mesh in two dimensions, an interior node is owned by four quadrilateral elements and thus the global entry in  $\underline{\underline{K}}^{(e)}$  contains four accumulated matrix values. If one carries out



For the sake of achieving the same discretization, the use of the FDM over the CGM becomes apparent from the discussion above. Most strikingly, the process of producing a local stencil is vastly more straightforward than performing element-wise assembly and gathering the weak form integrals in a global, sparse linear system. Each element-wise operation in assembly would otherwise require the evaluation of the mesh jacobian for the requested element, that is the mapping from the reference to physical space. Furthermore, this constant stencil enables Finite Difference schemes to operate in a matrix-free manner easily. For larger systems, this can help to avoid a large amount of allocated memory, thus being suited well for modern hardware architectures that are typically memory-bound.

These advantages are however offset by some topological restrictions on the mesh. The simplicity of a constant stencil also implies that the mesh must not deviate from a cartesian geometry. Otherwise, additional complexity is introduced since Equation 5.8 becomes a stencil in the reference domain that has to be mapped to the physical domain. This would still save the computational effort to assemble the weak form. However, since this process only has to be carried out once for the reference element, the computational impact can be held low by pre-computing the integrand.

## 5.4 Finite Volume Method

In a similar vein to the FDM, the use of the FVM might appear distinctly different from the idea of Finite Elements. Here, extensive use of Stokes' theorem is made to replace volume with hull integrals in conservation laws (Eymard et al., 2000).

It can be shown however that the FVM can simply be considered a Bubnov DGM of polynomial order zero. To illustrate this, Equations 5.1 and 5.2 describing the strong and weak forms of the advection equation are considered again. A Finite Volume approximation in conservation form is:

$$\int_{\Omega} \partial_t \alpha \, dx + \int_{\Omega} \underline{u} \cdot \nabla \alpha \, dx = \int_{\Omega} \partial_t \alpha \, dx + \int_{\partial\Omega} \alpha (\underline{u} \cdot \underline{n}) \, ds \quad (5.16)$$

Apart from the presence of a test function  $v$  in Equation 5.2, the second integrand simply represents the net flux of the conserved quantity  $\underline{u} \alpha$  over the set of element boundaries.

For Equations 5.2 and 5.16 to be equivalent in this case, the third integrand resulting from partial integration has to vanish in addition. However, this can be shown trivially by setting the order of the polynomial space for the trial and test function to zero. Then, the derivative of the test function vanishes and thus the entire term does not contribute to the weak form.

After performing this step, the test function is still present in the remaining parts of Equation 5.2. For the remaining terms to be equivalent, it must vanish out of the equation as well. This can be accomplished straightforwardly by fixing the value of the test function to be unity. In the weak form, this step is admissible since it must hold for all instances of  $V$ . As  $1 \in V^0$  where  $V^0$  is the space of constant polynomials, this statement holds in particular for a Bubnov Galerkin scheme, as trial and test space must be identical. The qualitative similarity of both schemes is illustrated in Figure 5.3. For the FVM, one must first reconstruct the values of the DoFs at the mesh facets to then compute the numerical flux. For both FVM and DGM, DoFs are entirely local to the cell and coupling happens through the calculation of a numerical flux - or in more formal terms, through the evaluation of the hull integral in the corresponding weak form. However, the FVM only stores one DoF per cell which has notable implications for the calculation of the numerical flux. This means that as a first step, the cell values have to be reconstructed at the mesh facets. These reconstructed DoFs which depend on the cell values that they interpolate between can then be coupled to their counterparts at opposing mesh facets. These relationships are denoted by DoFs being colored

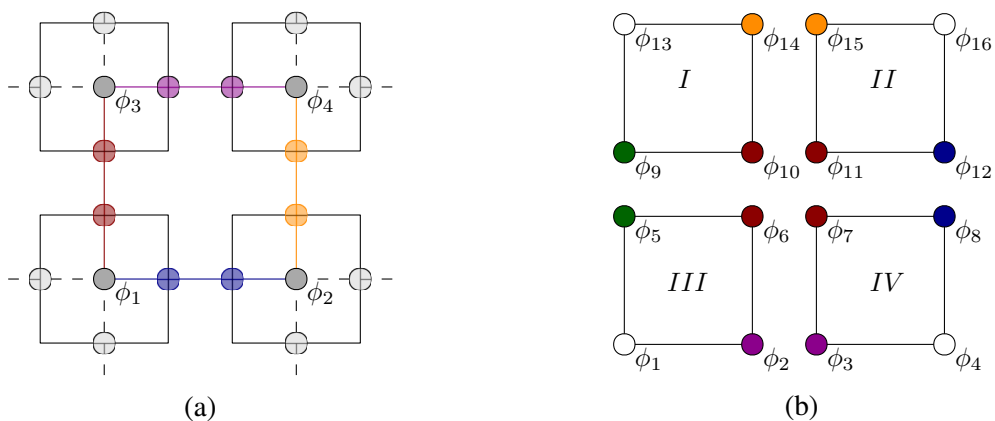


Fig. 5.3: Comparison of FVM (a) versus DGM (b) on an identical quadrilateral triangulation. Coupled DoFs are marked in identical colors.

identically (coupled) and being transparent (reconstructed) in Figure 5.3a. For a DGM scheme of first order or higher, this interpolation step oftentimes is not necessary if the quadrature scheme is chosen carefully. For a collocation method (see section 5.3 for a more thorough discussion), one does not need to tabulate the full list of DoF values at the set of facet quadrature points, but rather only a small subset of DoFs that are owned by the facet (Hesthaven and Warburton, 2008).

In summary, the FVM can again be considered as a special instance of the Bubnov type DGM, where the shared polynomial space  $V$  is taken to be of constant order and the test function  $v$  is set to be unity. Similarly to the discussion on the FDM, it should be noted that this simplification of Equation 5.2 brings with it some computational advantages that can be offset by sacrificing flexibility. The absence of a true weak form in a Finite Volume formulation again means that actual assembly is not needed. In addition, one may omit the transformation from the reference to physical space, as interpolating degrees of freedom to mesh facets and forming a finite sum of these contributions can be done on the mesh directly. The caveat of this approach is that an FVM in principle is bound to be at most first-order accurate. In practice, this does not hold as the FVM can be extended to higher orders by applying higher order flux reconstruction techniques (Zhou et al., 2001; Shu, 2003). Such techniques can however quickly become computationally expensive as well with increasing order. This is achieved in this case by widening the stencil for polynomial reconstruction, increasing memory and time complexity by a considerable amount (Liu and Zhang, 2013).

## 5.5 Findings

In this chapter, a common framework to formulate the most prevalent grid-based numerical schemes for the solution of PDEs has been established. It turns out that the DGM possesses enough flexibility to incorporate the CGM, FDM and FVM by imposing a set of restrictions. A summary of the results presented in this section on how the schemes compare overall is given in Table 5.1. This framework is not only of theoretical use. Rather, such a common formulation also enables the analyst to combine these schemes arbitrarily to solve larger problems. As each scheme possesses strengths and means to gain computational efficiency, this is an important result since it enables efficiently mixed discretizations of multiphysics problems. Establishing a practical method

to achieve exactly this will be the content of the next section.

Before concluding the discussion on relating the above numerical schemes, an important remark must be made. There exist several extensions to these methods that in general do not fit into the framework that has been established. For the sake of illustrations, some examples are given in the following.

There also exist formulations of the FDM that can capture domains with less regularity, see for example (Fornberg, 1988; Visbal and Gaitonde, 2002; Zhang et al., 2012). One can also find alternative discretization methods based on FDM in the literature that encompass the notion of missing structure in grids more naturally, such as understanding vertices as centroids of Voronoi cells (Sukumar, 2003)

As mentioned previously, there exist various formulations of the FVM that extend far beyond the original restriction of being first-order accurate. The cell-averaged flux is then determined in terms of reconstructing polynomials that in theory can be of arbitrary order. Such approaches per se do not fit well into the above-given DGM but do however achieve similar results.

In summary, research hypothesis H1 can be formally addressed as follows: There exists a common formulation for grid-based approximation methods in the form of the DGM, such that the schemes are comparable and can be combined. It has been shown

Table 5.1: Comparison of the individual restrictions that the presented schemes impose. Certain simplifications bring with them computational advantages, as discussed above.

Scheme	Geometry	Function Space	Weak Form	Quadrature
DGM	Arbitrary	Arbitrary	Full	Arbitrary
CGM	Arbitrary	Globally Continuous Space ( $H^1$ )	No hull integrals over interior facets	Arbitrary
FDM	Cartesian Geometry	Globally Continuous Space ( $H^1$ )	No hull integrals over interior facets	Collocation
FVM	Arbitrary	Discontinuous Polynomial Space $\mathbb{P}^0$ , Bubnov Galerkin	No volume integrals	None required

that the considered methods each represent simplifications of the DGM that enable efficient solutions to certain problems. Through careful tailoring of the methods to the multiphysics problem at hand, one can indeed construct an approximation method that generates a solution in a computationally efficient way. Therefore, H1 is considered to be formally verified.



# Chapter 6

## Deduction of the Unified Modeling Methodology

The overarching goal of this section is to identify a suitable combination of numerical schemes for a given multiphysics problem that is stable and accurate on the one hand, but also performant with regards to a specific choice of hardware on the other hand.

With the set relations between methods discussed in section 5, the simplifications and thus computational advantages that each scheme presents can now be utilized. That is, the general guideline in the following is to impose as many restrictions as possible whilst sustaining enough degrees of freedom to accurately capture the behavior of a given PDE.

### 6.1 Preliminary Assumptions

As a starting point, it has to be stated that encompassing the entire state of research on such schemes would be a daunting if not impossible task. The likewise formalization of a common framework is equally challenging as a consequence and thus not considered in this work.

Instead, the following restrictions are introduced that are on the one hand enough to construct a unifying scheme but on the other hand not too strict such that the efficient solution to real-world problems would be out of scope:

1. Only Bubnov Galerkin schemes are considered. Consequently, Petrov Galerkin methods are omitted from the scope. The former restricts the choice of test space

to be identical to the trial space. As such, schemes are left aside that for instance use weighted functions or stencils to account for flow fields. An example of such schemes would be the Streamline Upwind Petrov Galerkin (SUPG) method that is used to model flow problems, such as the passive advection equation (5.1) (Brooks and Hughes, 1982). This restriction is essential to obtain an unambiguous choice of method, as the notion of Petrov Galerkin methods does not imply any particular choice of function space. Naturally, omitting the use of arbitrary test function spaces contains a tradeoff as, in theory, the solution space is restricted by doing so. However, the majority of numerical methods developed so far indeed fall within the category of Bubnov Galerkin schemes with notable exceptions mainly being Petrov Galerkin methods used to stabilize saddle-point systems, such as the SUPG method outlined above (Fries and Matthies, 2004). Thus, mainly special instances of numerical methods are disregarded that are tailored to particular use cases.

2. Function spaces for approximation other than the  $L^2$  and  $H^1$  Sobolev spaces are omitted. There exists a vast variety of so-called Mixed Finite Element schemes that use Finite Elements based on different or composite function spaces with unique properties Ern and Guermond (2004). For example, one may construct function spaces that can exactly fulfill divergence-free properties ( $H(\text{div})$ ) or conditions based on the rotation of a field ( $H(\text{curl})$ ). The specific choice of Finite Element then would require a considerable amount of expertise and would warrant a complex decision process of its own. Furthermore, the amount of elements available for such spaces is vastly ambiguous, as outlined in section 1.2. Thus, regular scalar-, vector- and tensor-valued elements are solely focused on. They have been shown to encompass a similar solution space as well and perform comparably for fluid and electromagnetic problems (Cockburn et al., 2004; Hughes et al., 2006). There also exist mixed formulations for the poisson problem  $\Delta u = f$ , which can be regarded as a steady-state version of the heat equation (2.5), i.e. the temporal derivative in  $\partial_t u - \Delta u = f$  is set to zero. However, the ambiguity of choosing suitable mixed Finite Elements and resulting potential stability issues pose considerable disadvantages of such spaces, as already outlined for the poisson problem in section 1.2 and in particular Figure 1.4. In summary, some freedom in choosing possibly very well-suited function spaces is sacrificed for the benefit

- of attaining a decision metric that allows for an unambiguous recommendation.
3. Closely related to the previous statement, the solution space is further restricted by requiring that only Finite Elements utilizing Lagrange polynomials should be used. As the standard scalar- and vector-valued  $\mathbb{P}^k$  and  $\mathbb{Q}^k$  Finite Elements, being by far the most popular choices use exactly this family of polynomials, this requirement is weaker in practice than it might seem at first glance ([Arnold and Logg, 2014](#); [Cockburn and Fu, 2017](#)).
  4. The following method is based on a coarse taxonomy to classify the qualitative behavior of a given PDE. That is, limits regarding the leading coefficients of the differential operators are introduced. This should indicate whether the physical process described by the PDE is either more dissipative or more convective by nature. This leads to a more physical interpretation than the considerably stricter coercivity measures employed in functional analysis. The taxonomy closely follows the classes that were proposed by [Bitsadze \(1988\)](#). It should be noted that this classification is not claimed to be universally accurate. In practice, it has been shown however that having discrete cut-off values to disambiguate classes of PDE eases the choice of numerical scheme for application experts considerably.
  5. Only systems of PDEs with differential operators up to second order are considered. These are most common within physical processes and enable a wider range of numerical schemes to be used. For instance, equations of higher order such as the Cahn-Hilliard equation, would require the use of Finite Elements where up to third-order derivatives are defined. Such elements of high continuity are cumbersome to derive and are rarely used. Instead, a practical way to circumvent that is to reformulate the system as a mixed problem, where in the mentioned example one could represent the quantity of interest as two fields with second derivatives each. This technique is also regularly used in practice.

## 6.2 PDE Classification

To find a numerical scheme that produces stable results, knowing the qualitative behavior of the system oftentimes is a necessity. In particular, this means that the specific capabilities that a chosen numerical scheme possesses need to reflect the properties that the system presents.

This is illustrated by the following example. Once again, the simple advection equation (5.1) is investigated, which is known to be first-order hyperbolic. In general, hyperbolic systems are prone to either preserving or even amplifying discontinuities given in the initial condition. Thus, the capability of accurately representing these should be incorporated into the choice of numerical schemes (LeVeque, 1992). Suitable candidates would then be a Finite Volume or DGM. However, the FDM using a centered stencil or the CGM would give suboptimal results. The strong imposition of continuity in the domain would then yield spurious oscillations that affect stability (Ern and Guermond, 2004).

Hence, the system of PDEs is required to be classified firsthand. As mentioned previously, the popular taxonomy of second-order PDEs is used that can, for example, be found in the book by Bitsadze (1988), but as criteria for classification, a more general method for determining the appropriate class is used (Pinchover and Rubinstein, 2005). That is, a singular governing equation in the form of a PDE is defined to be either elliptic, parabolic or hyperbolic, depending on the shape that its characteristic quadric takes in space:

**Definition 6.2.1** (Classification of second order PDEs (Pinchover and Rubinstein, 2005)). Given a differential operator  $L$  of the form:

$$L(u) = \sum_{i,j=1}^n a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} \quad (6.1)$$

where  $x_i$  are the dependent variables and  $a_{ij}$  is the matrix forming the coefficients of the highest spatial derivatives. Considering the eigenvalues  $\lambda_i$  of  $a_{ij}$ ,  $L$  is called

- elliptic, if all  $\lambda_i$  are either positive or negative,
- parabolic, if at least one eigenvalue is zero and all others are either positive or negative,
- hyperbolic, if at least one eigenvalue is positive and at least one is negative.

The characterization of first-order differential operators is more straightforward, however. It can be shown that first-order PDEs with constant, real coefficients are always hyperbolic. This condition is met for most cases relevant to engineering or physical applications. More precisely, a first-order PDE is hyperbolic, if the resulting Cauchy problem is uniquely solvable. In the case of real, constant coefficients, the

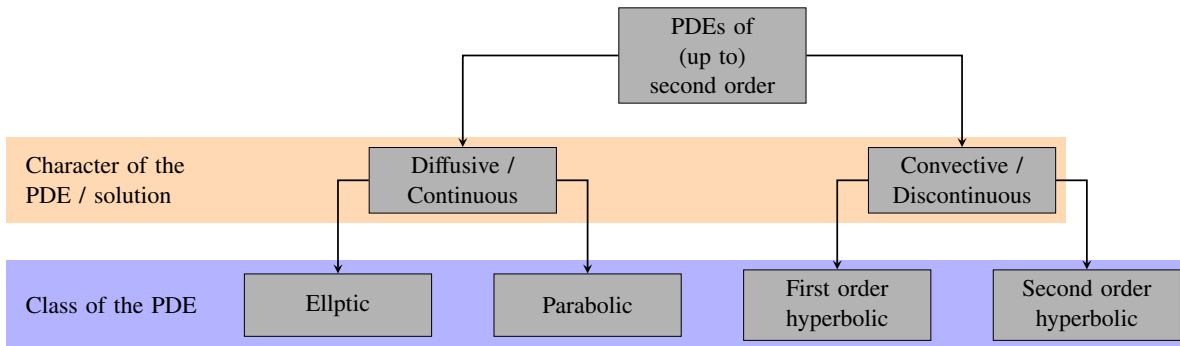


Fig. 6.1: Classification of PDEs up to second order by qualitative nature and types following Bitsadze (1988).

polynomial equation for each variable has to admit  $n$  solutions for an equation of order  $n$  while keeping all other variables constant (Hazewinkel, 2002). In the present case, this is trivially true.

This classification is applied for each governing equation of the independent variables for a given multiphysics problem. In practice, one may oftentimes identify the class by the differential operators that frequently appear in a given PDE. For example, a PDE that only has a laplacian as a spatial differential operator - such as the Laplace equation  $\Delta u = 0$  or the heat equation  $\partial_t u - \Delta u = 0$  exhibits dissipative behavior and is prototypical for elliptic and parabolic PDEs. Oftentimes, one can easily identify a differential operator as parabolic if it has an elliptic operator in its spatial derivatives and an additional temporal derivative, as is exactly the case for the heat equation.

Both the abovementioned classes of PDE are dissipative, the reason being that PDEs of second order can only have discontinuous derivatives along their characteristics. Since elliptic differential operators lack any characteristics, they strictly admit smooth solutions in that sense (Pinchover and Rubinstein, 2005). Thus, this qualitatively dissipative behavior is associated with elliptic and parabolic PDEs as defined above.

However, the advection equation (5.1) only has the gradient as a spatial differential operator, representing purely convective behavior. Exactly this behavior of transporting information through the domain with finite speed is associated with the wave-like character of hyperbolic equations. Figure 6.1 gives an overview of the classes of PDEs considered. In alignment with the postulate at the beginning of this section, a given class of PDEs should be solved with as few degrees of freedom as possible whilst not over-constraining the solution.

Most importantly, discontinuities that might appear in the solution should be properly accounted for and reflect the choice of numerical scheme. The direct consequence is that methods enforcing continuity should be used for problems that qualitatively exhibit high regularity and continuity. From the previous discussion, it becomes apparent that this is the case for the FDM and CGM. Problems that either conserve or even develop shocks however should be solved using methods that naturally allow for such. This means that either the FVM or DGM suit this requirement most naturally.

### 6.3 Domain Geometry

As discussed in section 5.3, the discretization using the FDM inherently assumes an even grid with uniform spacing between nodal points. The direct consequence of this simplification is that assembly can be done in the computational domain directly and in an equal manner for every node point.

In general, if the domain has a particularly simple shape, for example, a hypercube and does not contain any holes, it can be triangulated using a cartesian grid. Thus, if the discrete domain fulfills these conditions and the differential operators form an elliptic or parabolic PDE, using the FDM to efficiently assemble the global system is advisable.

For FVM, CGM and DGM, regularity of the computational domain, in general, does not pose any considerable advantages that may accelerate the assembly of the discretized system.

### 6.4 PDE Linearity

Another crucial property to assess is the linearity of a PDE. In this case, a PDE is defined to be strictly linear, semilinear, quasilinear and fully nonlinear:

**Definition 6.4.1** (Evans (2010)). A  $k$ -th order PDE of the form

$$F(D^k u(x), D^{k-1} u(x), \dots, Du(x), u(x), x) = 0$$

is called:

1. linear, if it has the form

$$\sum_{|\alpha| \leq k} a_\alpha(x) D^\alpha u = f(x)$$

for given functions  $a_\alpha(|\alpha| \leq k)$ ,  $f$ . The PDE is homogeneous if  $f \equiv 0$ .

2. semilinear, if it has the form

$$\sum_{|\alpha|=k} a_\alpha(x) D^\alpha u + a_0(D^{k-1}u, \dots, Du, u, x) = 0$$

3. quasilinear, if it has the form

$$\sum_{|\alpha|=k} a_\alpha D^\alpha u(D^{k-1}u, \dots, Du, u, x) + a_0(D^{k-1}u, \dots, Du, u, x) = 0$$

4. The PDE is fully nonlinear if it depends nonlinearly upon the highest-order derivatives.

While linearity does not pose much of a problem for elliptic or parabolic equations, it plays an important role in whether a discretization is stable for hyperbolic equations. The theory of nonlinear flux limiters is in general well researched for DGMs and largely profits from extensive developments that originally stem from the FVM. However, accurate computation and implementation remain to be a hurdle in practice. There have thus been several approaches to circumvent this issue, for example, by switching to an FV scheme in regions where the stability of the solution cannot be guaranteed ([Maltsev et al., 2023](#); [Sonntag and Munz, 2014](#)).

As the overarching goal of this method is to provide straightforward guidance for end users, such approaches will be omitted that must in most cases be implemented in a custom and rather particular fashion in favor of simplicity. The recommendation thus is that, for equations where the solution is not likely to require many nonlinear iterations per time step, one may safely use a DGM. In other cases where stability cannot be assured universally, one should rather switch to a Finite Volume formulation that may be overly diffusive, but on the upside is guaranteed to yield a stable solution.

## 6.5 Computing Environment

Within the last decade, a pronounced plateau in processor speed can be observed which is ultimately limiting further advancements in computing performance (Theis and Wong, 2017). Due to these slowly diminishing returns, this trend has pushed for other means of accelerating computing. One of the most established ones is the use of parallelism, i.e. distributing a computational task across multiple workers (Leiserson et al., 2020).

As a direct consequence, the advancement of computer hardware has been known to slowly approach the so-called memory wall (McKee and Wisniewski, 2011). This means that applications tend to be bound by the capability of the hardware to transfer shared information in memory instead of performing arithmetic operations. Since in parallel environments, many workers perform operations on the same data, information must frequently be transferred and updated. Such synchronization efforts can quickly become a bottleneck. This in particular holds for numerical simulations that are performed using many workers or problems that are large. In such cases, the evaluation of sparse matrix-vector products poses high loads regarding memory bandwidth (Arndt et al., 2020).

Regarding this specific, increasingly important requirement, some numerical schemes naturally lend themselves toward parallelism and others are more memory-bound by design (Weinan and Lu, 2011; Kirby et al., 2012). Thus, for a given computing hardware that puts enough emphasis on massive parallelism and two numerical schemes performing (nearly) identically, one should prefer the one that handles parallelism better. One thus naturally arrives at the question of where one should disambiguate between massively parallel and other, regular hardware.

There are essentially two factors that would affect such a classification. First, the hardware architecture itself plays an important role. One may on the one hand solve a PDE on the classic CPU architecture that is capable of performing arithmetic on many precision levels and use many specialized instruction sets, such as AVX or fused multiply-add (FMA). Another possibility is the use of highly parallel computing units, such as general-purpose graphics processing units. Those however have a memory layout and instruction set that is much more tailored toward one purpose. In the case of a GPU, this is medium to low precision operations with comparably low memory intensity but instead high arithmetic effort.

The other deciding factor is the amount of workers involved in the simulation process. The more workers exist, the more processor boundaries are present and thus more information has to be shared between processors. For some schemes, this overhead due to the exchange of memory between workers can become prohibitive. Within the FVM, for instance, parallel efficiency measured in GFlops/s starts to drop notably within the regime of 50 to 100 workers (Marshall et al., 1997). It should be noted that the quantitative drop-off also depends on the specific implementation. Fringer et al. for instance note a decline in parallel efficiency for a Finite Volume solver starting at 32 workers (Fringer et al., 2006). Thus, as a general guideline, it is recommended to employ methods that are suited for highly parallel environments at roughly 50 or more CPU workers. For execution on massively parallel architectures, such as GPUs, the switch to such algorithms is considered necessary to obtain good efficiency.

## 6.6 Problem Scale

Another deciding factor for whether adaptivity is needed or not is the presence of multiple length scales in a multiphysics model.

The underlying definition for a multiscale problem can be found in (Weinan, 2011; Weinan and Lu, 2011). A PDE-governed problem is thus characterized to have a multiscale nature if models of multiple spatial or temporal scales are used to describe a system. Oftentimes, this is the case if equations are used that originate from different branches of physics, such as continuum mechanics versus quantum mechanics or statistical thermodynamics.

This may on the one hand be a physical process with slow and fast dynamics, for example, in chemical reaction networks. Then, the multiscale nature shows itself in the time domain of the problem. Another example commonly encountered problem in alloy design is the evolution of the temperature field and phase kinetics during heating and solidification. In this case, various length scales can be involved, such as in processes involving laser heating. The temperature gradients then involve resolutions at a scale of around  $1 \times 10^{-5}$  m, whereas the width of a solidification front rather goes down to a sub-micrometer scale, that is, around  $1 \times 10^{-7}$  m (Zimbrod and Schilp, 2021). About the previous definition, there is one model that is governed by laws of macroscale thermodynamics, in this case specifically the heat equation (2.5). The other part of physics

present is typically described by the evolution of a phase field. The corresponding equations of this model are however derived from the formulation of a free energy functional from Landau theory (Landau, 1937).

Due to the wide variety of physical processes and combinations thereof, formulating general criteria for the presence of a multiscale problem from a mathematical point of view is challenging. As of yet, a corresponding metric is not present in the literature that would enable such a classification. Instead, knowledge of the application expert is required who is assumed to be familiar with the physics that should be captured, as stated in 4.3. For a rough disambiguation, however, if no other information is given, one may use the definition given above.

Such multiscale phenomena are prohibitively expensive to resolve on a uniform mesh due to the nonnegligible difference in the dynamics of the system. One option to efficiently resolve the physics at multiple scales is to employ different grids and solve the resulting problems in parallel. This has, for example, been done for the case of PBF-LB/M to concurrently resolve the macro- and mesoscale physics (Ghanbari et al., 2020).

A rather effective, alternative approach is the modification of the governing equations such that they become tailored to a specific numerical scheme. For instance, the well-known phase field model has been adapted using specialized stencils to the FDM such that spurious grid friction effects are eliminated (Fleck and Schleifer, 2023; Fleck et al., 2022). This approach however requires extensive knowledge about the numerics as well as the physical nature of a given problem, which has explicitly been excluded from the scope of the application expert as defined in section 4.3 and can thus not be considered as an option.

Another possibility that requires fewer algorithmic adaptations to the specific problem is to make use of grid adaptive algorithms. This approach for the problem presented is a popular alternative and has been implemented multiple times (Proell et al., 2023; Olleak et al., 2022; Olleak and Xi, 2020b,a). Thus, grid adaptivity plays a key role in creating solutions to such problems, if the domains are not to be resolved on different discretizations entirely. Numerical methods as a consequence need to reflect on this requirement and as such, FDMs are not suitable for such types of problems.

Whereas grid adaptivity is easily realizable within the FVM, there is little room for adaptivity regarding the order of approximation and can at best be achieved using

varying reconstruction stencils (Shu, 2003).

CG FEMs do enable grid as well as polynomial degree adaptivity. Yet, the imposition of hanging node constraints is oftentimes not trivial. Though there have been considerable strides toward easy and intuitive handling of hanging nodes for continuous elements (Šolín et al., 2008; Bangerth and Kayser-Herold, 2009), these methods naturally fall short of the inherently decoupled nature of DoFs present in discontinuous methods.

By far, the most naturally suited method for h- as well as p-adaptivity is the DG FEM. The locality of DoFs enables the splitting of cells without the need for hanging node constraints. The same argument applies to altering the degree of a Finite Element, as additional DoFs within the cell need not be attached to a counterpart on its neighbors. The natural splitting and refining of cells that the DGM allows in comparison to a CGM is illustrated in Figure 6.2. Unrefined DoFs are colored grey, refined, local DoFs blue and refined, global DoFs are colored red. In the case of the CG FEM, the red DoFs are considered hanging nodes, as they do not have a refined counterpart at the opposite side of the facet and are thus drawn as half circles. For the DG FEM, all DoFs are local and not being shared with other cells. Thus no hanging nodes are created

## 6.7 Resulting Method

The various aspects of choosing appropriate numerical schemes can now be condensed as follows into a unifying method, given the restrictions posed in section 6.1.

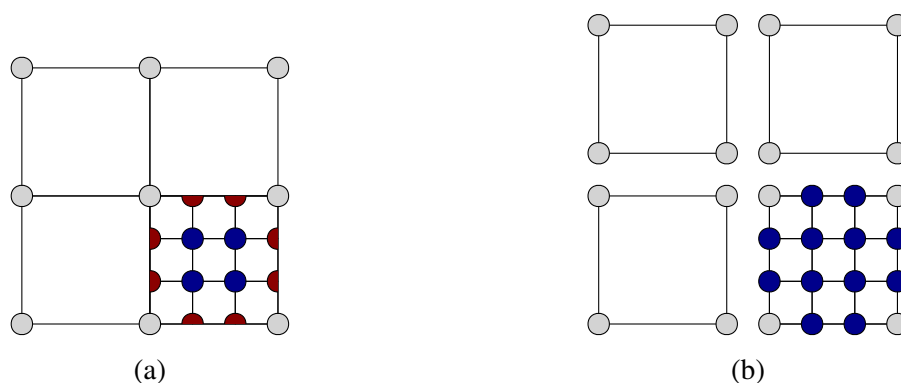


Fig. 6.2: Grid refinement within the CGM (a) versus DGM (b), both locally from first to third order.

First, three sets of inputs are taken that are of practical relevance: the mathematically formulated, continuous (i.e. not yet discretized) problem, the computational domain that one wishes to solve the former on, and the configuration of the target hardware.

To design the intended decision process, the decision metrics are first evaluated that impact the target scheme in the most general manner. The general question of whether the prescribed system of PDEs requires an efficient solution on a large scale fulfills this requirement here. The term large scale here denotes state-of-the-art computing hardware on massively parallel architectures. That decision in turn is influenced by two factors: One may directly intend to efficiently solve the system of PDEs on that hardware, or the multiscale nature of the problem demands such a computing environment. If either is the case, solving the entire system using the HDG method is advisable due to the resulting locality of the problem.

The remaining parts of the decision process depend on the class of PDE present. From here on, one operates in a field-wise manner and classifies the system of PDEs for each independent variable separately. If a PDE is convective in character, that is, hyperbolic, the use of numerical schemes that incorporate discontinuous approximations is recommended. But, if a problem is diffusive by nature, the solution will be continuous and thus the use of continuous approximations is more advisable.

In the case of the former, following the discussion in section 6.4, a final disambiguation must be made regarding linearity. If the PDE is linear or semilinear, a DGM can be applied due to the unlikelihood of stability issues. Otherwise, the use of a simple FV scheme is more advisable to obtain a stable solution without having to iterate through many different choices of flux limiters in a trial-and-error fashion.

Regarding the continuous schemes, as has been explained in sections 5.3 and 6.3, the configuration of the domain geometry plays an important role in the efficiency of the overall scheme. If the domain is cartesian, irrespective of dimensionality, the FDM can deliver accurate results with a considerably decreased amount of arithmetic operations. The conceptual flexibility of the FEM regarding the domain is then unnecessary. In the other case though where the domain is topologically more complex, relying on FEM algorithms that account for the necessary global mappings is more appropriate. It would of course be possible to identify a middle ground between both schemes, for example, when a simple and prescribed transformation can be applied to the entire domain. This would for example be the case for systems that can be described by polar

coordinates. However, few computer codes implement such functionality. As the focus of this method lies on practicality and usefulness, a method is rather chosen that can make use of widespread and established computer codes and thus these possibilities are omitted.

As a result, one obtains a process that guides the application expert through iteratively selecting the most appropriate combination of numerical schemes for a given, fixed and well-defined set of inputs. This method is summarised in a flow chart, which is depicted in Figure 6.3. Inputs (**I**) are given by purple trapezoids, decision points (**D**) by white diamonds and processes (**P**) by orange rectangles. Processes with additional vertical bars denote more complex processes and have references to their respective sections. Results are shown in green trapezoids. It should be noted at this point that, as can be seen in Fig. 6.3, this method will always output a combination of discretization schemes for a given Multiphysics problem that fits the requirements given in this chapter. Furthermore, the selected schemes are interoperable with each other and can thus be tightly coupled due to the theory established in the previous chapter.

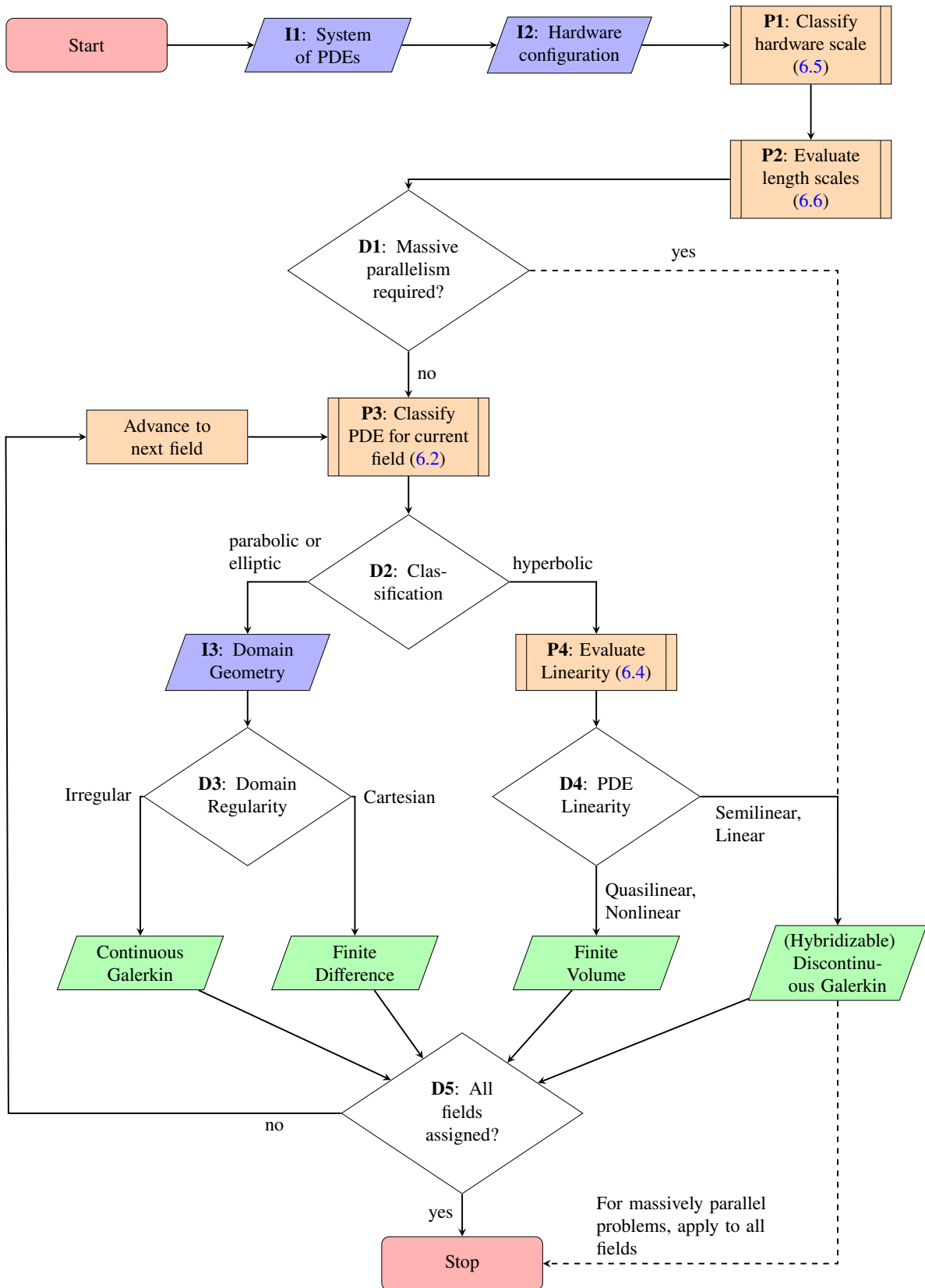


Fig. 6.3: Graphic summary of the proposed process for choosing appropriate numerical schemes.

## 6.8 Weak Form Derivation

As has been outlined in sections 2.2.4 and 2.2.5, working with FEMs on a general PDE level requires setting up the corresponding weak formulations of the desired equations. For larger or coupled multiphysics problems, this process can quickly become complex. It is thus assumed to be not trivial from the standpoint of an application expert. Therefore, the process of arriving at a valid weak form will be outlined below. Such weak forms can oftentimes then be entered almost as is into modern Finite Element codes, such as FEniCS (Logg et al., 2012b) or Firedrake (Ham et al., 2023).

For the vast majority of Finite Elements and in particular  $\mathbb{P}$  and  $\mathbb{Q}$  Lagrange elements, only  $H^1$  continuity holds at most, i.e. the second derivative across element boundaries is not well defined. There are few exceptions to this statement such that second derivatives are naturally incorporated into the Finite Element basis functions (Arnold and Logg, 2014). These, however, are outside the scope of this work as outlined in section 6.1. As such, second derivatives may not be computed as is, but require additional treatment.

If a given PDE containing only second-order derivatives or higher is to be instead discretized using the DG-FEM, algebraic splitting of the equation is required. This is due to the global space of DG solutions only having  $L^2$  continuity, i.e. first derivatives are not well defined at element boundaries. Thus, applying partial integration is generally not sufficient (Nguyen et al., 2011). This procedure has already been demonstrated for the Poisson problem in Equations 5.3 through 5.6.

Overall, an algebraic manipulation of the differential equation is required for both schemes if higher-order derivatives are present. This process is denoted as hybridization and can be executed as follows. For the quantity that is contained within a higher-order differential operator, one iteratively introduces additional field quantities. These each are defined as functions of the original quantity until only expressions are left of up to either second order in the case of the CGM or first order for a DGM. As an example, the Cahn-Hilliard equation is considered which is oftentimes used to model phase separation in binary mixtures (Cahn and Hilliard, 1958):

$$\partial_t u - \nabla \cdot M \left( \nabla \left( \partial_u f - \lambda \nabla^2 u \right) \right) = 0 \quad (6.2)$$

This effectively is a fourth order PDE in the variable  $u$ , whereas  $\partial_u f$  is a free energy

functional, i.e. a function of  $u$ , and  $M$  is a scalar parameter. In order to enable the solution using a CGM in combination with Lagrange elements, one has to split the PDE once to obtain a system of two coupled, second-order equations

$$\hat{u} - \partial_u f + \lambda \nabla^2 u = 0 \quad (6.3)$$

$$\partial_t u - \nabla \cdot M \nabla \hat{u} = 0, \quad (6.4)$$

where fields introduced by hybridization are denoted using hats ( $\hat{\cdot}$ ) (Feng and Prohl, 2004). Alternatively, it is also possible to solve this equation using a DGM. Further splitting is then required due to the reduced continuity of the function space. To solve this fourth-order system, one must apply hybridization until only expressions with first-order derivatives are left (Xia et al., 2007)

$$\underline{\hat{u}} - \nabla u = 0 \quad (6.5)$$

$$\hat{\underline{u}} - \lambda \nabla \cdot \underline{\hat{u}} = 0 \quad (6.6)$$

$$\hat{\underline{\hat{u}}} - M (\nabla (\partial_u f - \hat{\underline{u}})) = 0 \quad (6.7)$$

$$\partial_t u - \nabla \cdot \hat{\underline{\hat{u}}} = 0. \quad (6.8)$$

If the order of the differential operator  $L(u)$  does not exceed one for a DG or two for a CGM, this entire process may be skipped.

After this preprocessing step, the space of test functions  $v$  must be chosen such that the inner product  $\langle \cdot, \cdot \rangle$  with  $L(u)$  is well defined, i.e. assumes a scalar value. The inner product roughly corresponds to tensor multiplication where all occurring indices are summed over. By definition, this means that  $v$  and  $L$  must belong to the same function space. Otherwise, the inner product is not defined and a valid weak formulation cannot be obtained (Brokate et al., 2016). After that, the primary weak statement can be formed by forming the abovementioned inner product and integrating it over the problem domain  $\Omega$ . The following step usually consists of applying integration by parts to the resulting form. However, at this point, one must disambiguate between linear and nonlinear differential operators, as for some this step cannot be performed. In the case of linear differential operators, there is no issue, as well as for nonlinear operators where a differential term appears in the outermost position. For example, the following term

may be rearranged by integration by parts, although not being linear:

$$\int_{\Omega} \langle v, \nabla(u^2 \nabla u) \rangle dx = \int_{\partial\Omega} \langle v, u \nabla u \cdot \underline{n} \rangle d\tilde{s} - \int_{\Omega} \langle \nabla v, u \nabla u \rangle dx \quad (6.9)$$

Whereas the weak formulation

$$\int_{\Omega} \langle \underline{v}, (\underline{u} \cdot \nabla) \underline{u} \rangle dx \quad (6.10)$$

does not fulfill this condition and thus, according to this method, cannot be transformed in this way. After modifying the weak formulation through integration by parts, one is left with an additional surface integral. The necessary antiderivative applied to  $L(u)$  in this case reduces the order of the differential operator by one. The integration measure used for this integral  $d\tilde{s}$  for now includes all interior and exterior facets of the computational domain. The following steps depend on the choice of the numerical scheme and thus continuity of the approximation. If a continuous approximation is used, then all surface integrals over interior facets are strongly enforced to be zero and thus vanish. One may then replace this term with an integral over only the exterior facets, denoted by the integration measure  $dS$ . Otherwise, both exterior and interior contributions remain, and it is practical to split the integral accordingly. These are oftentimes denoted by using  $ds$  and  $dS$ , respectively as measures. The final step then consists of incorporating boundary conditions into the weak formulation. Although it is possible to enforce general boundary conditions weakly through additional terms by Nitsche's method, the following procedure is more common (Nitsche, 1971; Freund and Stenberg, 1995). In the case of Dirichlet conditions, i.e. there exists a prescribed function  $g(u)$  at the domain boundary, no modification of the weak form is necessary. This type of boundary condition is usually enforced strongly by applying a technique denoted as lifting and happens on the discrete, algebraic level (Ern and Guermond, 2004). For Neumann-type conditions, however, if the exterior surface integral contains the expression defined at the boundary, one may directly substitute that for the prescribed boundary function  $h(\partial_n u)$ . The same argument holds for boundary conditions that involve other differential expressions at the boundary. As an example, one may consider the boundary of a perfect

conductor in electromagnetics, where the condition

$$n \times H = 0 \quad \text{on } \partial\Omega \quad (6.11)$$

holds for the magnetic field  $H$  (Senior and Volakis, 1995). If such expressions appear within integrals over boundary facets, then, they can simply be replaced by the prescribed function.

Depending on whether the weak form is linear or not, a bilinear form  $a(u, v)$  or a semilinear weak residual  $F(u; v)$  is then attained. For arbitrary differential operators, acting on real-valued functions, the described process may be generalized and formalized in the form of a flow chart given by Figure 6.4. At this point, it should be noted that the presented process for forming the weak form of a nonlinear PDE is vastly simplified to provide a systematic way of derivation. More generally, weak formulations of nonlinear systems tend to be more tailored in an ad hoc fashion to the problem.

One of the most practically relevant steps in Figure 6.4 is modifying the form through integration by parts. For most of the linear forms encountered in practice, i.e. the gradient, divergence, curl and laplacian operators, this process however can be carried out systematically. A summary of the resulting weak forms for each differential operator satisfying the above-given conditions is listed in Table 6.1. Each tabulated weak form contains  $v$  as a test function, belonging to the same function space as  $u$  and the domain  $\Omega$  to be integrated over. The inner product of two tensors, i.e. contraction over all indices, is denoted by  $\langle \cdot, \cdot \rangle$ . The  $\cdot$  operation denotes the dot product, i.e. contraction over the last index. In addition, the facet normal vector  $\underline{n}$  is introduced in the hull integrals. For two vectors, this coincides with the regular dot product. The two integration measures  $d\tilde{s}$  and  $dx$  denote integrals over facets (exterior and interior) and the entire volume, respectively. For a CGM, the facet integral is only taken over boundary facets. The resulting integration measure is then oftentimes denoted as  $dS$  instead of  $ds$ . With this tableau of weak differential operators as baseline, constructing a computational version of this rule set using modern Finite Element libraries such as FEniCS Logg et al. (2012a) is straightforward. This is due to the common baseline that many of these packages share in the form of the so-called Unified Form Language (UFL) Alnaes et al. (2014). There, the residual forms of a PDE are constructed using an input that very closely resembles the weak differential operators given in Table 6.1. The remain-

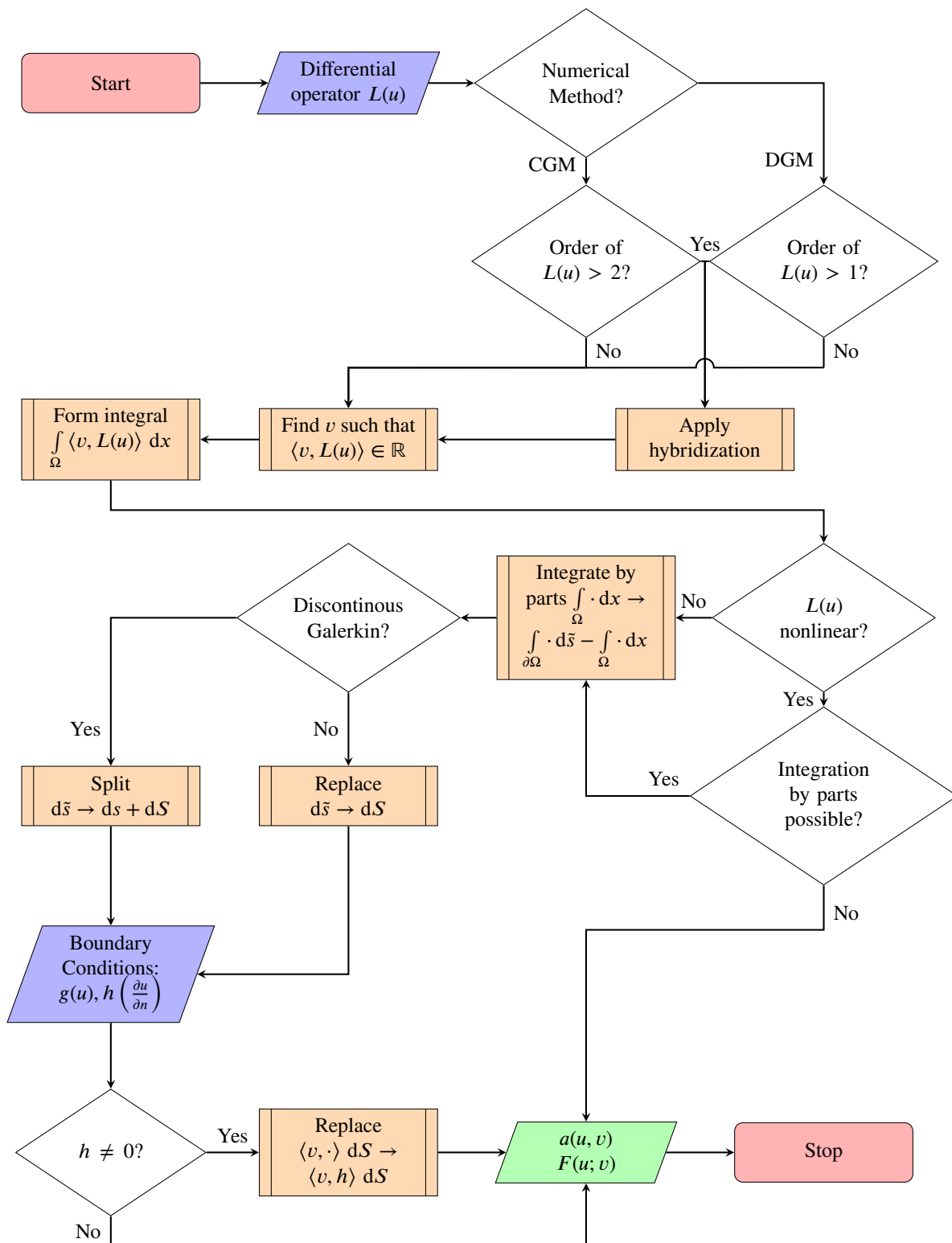


Fig. 6.4: Generation of a weak form expression for a differential operator  $L(u)$ . Inputs (I) are given by purple trapezoids, decision points (D) by white diamonds and processes (P) by orange rectangles.

Table 6.1: Tableau of differential operator weak forms, depending on the dimensionality of the target function  $u$ .

		<b>Tensor Rank</b>		
		0	1	2
Gradient	$\nabla$	$\int_{\partial\Omega} \langle \underline{v} \cdot \underline{n}, u \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla \cdot \underline{v}, u \rangle dx$	$\int_{\partial\Omega} \langle \underline{v} \cdot \underline{n}, \underline{u} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla \cdot \underline{v}, \underline{u} \rangle dx$	$\int_{\partial\Omega} \langle \underline{v} \cdot \underline{n}, \underline{\underline{u}} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla \cdot \underline{v}, \underline{\underline{u}} \rangle dx$
Divergence	$\nabla \cdot$	not defined	$\int_{\partial\Omega} \langle v, \underline{u} \cdot \underline{n} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla v, \underline{u} \rangle dx$	$\int_{\partial\Omega} \langle \underline{v}, \underline{\underline{u}} \cdot \underline{n} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla \underline{v}, \underline{\underline{u}} \rangle dx$
Curl	$\nabla \times$	not defined	$\int_{\partial\Omega} \langle \underline{v}, \underline{u} \times \underline{n} \rangle d\tilde{s}$ $+\int_{\Omega} \langle \nabla \times \underline{v}, \underline{u} \rangle dx$	not uniquely defined
Laplacian	$\nabla \cdot \nabla$	$\int_{\partial\Omega} \langle v, \nabla u \cdot \underline{n} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla v, \nabla u \rangle dx$	$\int_{\partial\Omega} \langle \underline{v}, \nabla \underline{u} \cdot \underline{n} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla \underline{v}, \nabla \underline{u} \rangle dx$	$\int_{\partial\Omega} \langle \underline{\underline{v}}, \nabla \underline{\underline{u}} \cdot \underline{n} \rangle d\tilde{s}$ $-\int_{\Omega} \langle \nabla \underline{\underline{v}}, \nabla \underline{\underline{u}} \rangle dx$

ing step thus lies in translating this tableau into the domain-specific pendants of (UFL). An exemplary implementation of the weak divergence and gradient that is compatible with the FEniCS package is given in Listing 1. As an example, the weak divergence of a vector quantity  $\underline{u}$  is considered using a CG formulation. According to Table 6.1, a scalar test function  $v$  is required. Then, by invoking the corresponding Python function via `FEOperators.divergence(type="Lagrange", test=v, u=u)` yields the expression `-inner(grad(v), u)*dx`.

## 6.9 Findings

In this section, a unified decision process for the selection of numerical schemes for multiphysics problems has been established. Its effectiveness has been demonstrated given two distinct examples.

With these results in mind, research hypothesis H2 is addressed as follows: Given

Listing 1: UFL implementation of the weak divergence, gradient and laplacian according with Table 6.1. The full source code implementing the remaining operators is available online on [GitHub](#).

---

```

1 class FEOperators:
2     def __init__(self, mesh: Mesh) -> None:
3         self.n = FacetNormal(mesh.dolfinx_mesh)
4
5         return
6
7     def divergence(self, type: str, test: TestFunction, u: Function,
8                   numerical_flux=None) -> Form:
9         F = - inner(grad(test), u) * dx # Partial integration
10        if type == "Lagrange":
11            pass # Nothing else to do
12        elif type == "Discontinuous Lagrange":
13            F += inner(jump(test), numerical_flux(u)) * dS # Hull integral
14        else:
15            raise NotImplementedError("Unknown type of discretization")
16        return F
17
18    def gradient(self, type: str, test: TestFunction, u: Function,
19                numerical_flux=None) -> Form:
20        F = - inner(div(test), u) * dx
21        if type == "Lagrange":
22            pass # Nothing else to do
23        elif type == "Discontinuous Lagrange":
24            F += (
25                dot(jump(test), numerical_flux(u)) * dS
26            )
27        else:
28            raise NotImplementedError("Unknown type of discretization")
29        return F
30
31    def laplacian(self, type: str, test: TestFunction, u: Function,
32                  coefficient=None, numerical_flux=None) -> Form:
33        if type == "Discontinuous Lagrange":
34            raise TypeError("Laplacian is not defined for DG discretizations.
35                ↪ The operator needs to be hybridized first.")
36        elif type == "Lagrange":
37            if coefficient == None:
38                F = -inner(grad(test), grad(u)) * dx
39            else:
40                F = -inner(grad(test), coefficient*grad(u)) * dx
41        else:
42            raise NotImplementedError("Unknown type of discretization")
43        return F

```

---

the set of restrictions introduced in section 6.1, one can indeed work out an unambiguous process to find suitable numerical methods in a reproducible way. This process is formalized using well-defined inputs as well as a set of steps to be conducted in a fixed order. Therefore, H2 is considered to be verified as well.

# Chapter 7

## Computational Examples

The purpose of this chapter is to walk through the proposed method using two simple example PDEs. Although these are not multiphysics problems, they may be combined in theory, as will be shown in the following chapter.

### 7.1 Allen Cahn Equation

First, the following scalar PDE is considered, together with zero flux boundary conditions to be imposed at the four borders of a rectangular domain  $\Omega : [0; L] \times [0; W]$ :

$$\frac{1}{K} \partial_t \phi - \Delta \phi = -\frac{2}{\xi^2} \partial_\phi g(\phi) - \frac{\mu_0}{3\gamma\xi} \partial_\phi h(\phi) \quad \forall \phi \in \Omega, \quad (7.1)$$

$$\frac{\partial \phi}{\partial n} = 0 \quad \forall \phi \in \partial\Omega, \quad (7.2)$$

This equation is called the Allen Cahn equation and describes the time-evolution of a scalar, non-conserved order-parameter field  $\phi$ , as is often called the phase field. If one only examines the left-hand side of Equation 7.1, it becomes clear that the Allen Cahn equation can be thought of as the heat equation (2.5) with some nonlinear terms added, as they both only contain first temporal and second spatial derivatives. Thus, the previously discussed difficulties in choosing appropriate discretizations for the heat equation (c.f. Section 1.2) apply here as well. However, as the equation itself is more complicated due to the additional nonlinear right-hand side, it can be stated that finding an efficient approximation is even less trivial than previously discussed. The Allen-Cahn equation is commonly used in the modeling of self-organised microstructure evolution

or complex pattern formation processes, as driven by local thermodynamics and/or mechanics. The phase field variable  $\phi$  can be understood as a coloring function that locally indicates the presence or absence of a certain phase or a certain material state within a given microstructure. For instance, in modeling of microstructure evolution during solidification,  $\phi = 1$  may denote the local presence of the solid and  $\phi = 0$  may denote the local presence of the liquid phase (Fleck and Schleifer, 2023; Fleck et al., 2022). If applied to the description of crack propagation, the order-parameter field  $\phi$  is understood as the local material state, which can be either broken  $\phi = 1$  or not  $\phi = 0$  (Pilipenko et al., 2011; Fleck et al., 2011).

The scalar quantities  $K$ ,  $\xi$ ,  $\mu_0$  and  $\gamma$  are model constants that determine the evolution of the scalar field  $\phi$ . Here, the notation of (Fleck et al., 2018) is adopted. The polynomials  $g$  and  $h$  on the right-hand side of Equation 7.1 pose a nonlinearity to the equation. Their derivatives are given by the following two source terms

$$\partial_\phi g(\phi) = 2\phi(1 - \phi)(1 - 2\phi), \quad (7.3)$$

$$\partial_\phi h(\phi) = 6\phi(1 - \phi). \quad (7.4)$$

In the following, those polynomial terms are gathered in the joint potential term

$$f(\phi) = \frac{2}{\xi^2} \partial_\phi g(\phi) + \frac{\mu_0}{3\gamma\xi} \partial_\phi h(\phi). \quad (7.5)$$

Further details on the parametrization of the model are given in Table 7.1. Concerning the Allen Cahn Equation, two different scenarios are considered, highlighting different aspects of the physics behind the equation. The first one is concerned with solving

Table 7.1: Parameters for the Allen Cahn model

	Unit	Value	Description
$\Gamma$	J/m <sup>2</sup>	1.0	Interface energy
$M$	m <sup>2</sup> /s	1.0	Kinetic coefficient
$\mu_0$	J/m <sup>3</sup>	0.1	Bulk energy gradient
$\xi$	m	1.5	Phase field interface width
$\tilde{x}_0$	m	20.0	Initial position of the interface
$\Delta x$	m	1.0	Grid spacing

Equations 7.1 and 7.2 in one spatial dimension, whereas in the second one, the problem is scaled up to 2D. For each of these two scenarios, quantitative measures are formulated to be able to precisely investigate the accuracy of the numerical solutions as well as compare the relative computational cost.

In the first scenario, the motion of a planar interface between two phases is considered at different energy density levels. The low energy phase is expected to grow at the expense of the high energy phase, which induces a motion of the interface between them at a velocity proportional to the constant energy density difference  $\mu_0$ . The scenario is realized as a quasi 1D problem  $[0; L = 100] \times [0; W = 1]$ , where the interface normal direction is pointing in the x-direction and the use of simple von Neumann boundary conditions with zero phase field fluxes at the borders of the rectangular domain is legitimate. The realization of this scenario with tilted interface orientations including the formulation of appropriate boundary conditions on the borders of the rectangular domain is discussed in detail in Fleck et al. (2022). In this highly symmetric quasi-1D case, the scenario can be quantitatively evaluated utilizing the existing analytic solution for the phase field

$$\phi(x, t) = \frac{1}{2} \left( 1 - \tanh \frac{x - \tilde{x}(t)}{\xi} \right), \quad (7.6)$$

where the time dependence of the central interface position  $\tilde{x}$  is given by,  $\tilde{x}(t) = x_0 + M\mu_0 t/\gamma$ , with the initial position at  $x_0 = 20$ . The initial condition of this problem is thus formed by evaluating Equation 7.6 at time zero. To investigate the impact of arithmetic complexity on computational efficiency, an approximation of a real-valued function  $\phi(x, t)$  is sought in one spatial dimension on a  $[0; 100]$  grid with equispaced vertices.

One may now start applying the proposed methodology, as described above in section 5 and following Figure 6.3. That is, the path of the flowchart should be followed from top to bottom. First, the hardware scale is classified according to P1 in the figure. The given hardware architecture, that is, an 8-core CPU system, falls well below the established recommendation for the threshold of partitioned problems which is at least 50 workers. Therefore, there is no need from a hardware side for massive parallelism.

Next, the problem scale is investigated with process P2. As the problem is governed by one scalar equation and there are no sub-models involved as defined in section 6.6.

Concerning the length scales, the presented system exhibits one extra physical length scale and that is the width  $\xi$  of the diffuse interface. This extra physical length scale originates from the nonlinearity of the Allen Cahn Equation and complements the other length scales such as the dimensions of the domain as well as the grid spacing, both being more natural in the numerical solution of PDEs. This poses the issue of numerical resolution of the systems length scales, that is, both the domain dimensions, as well as the width  $\xi$  of the diffuse interface, need to be properly represented on the discrete numerical grid (Fleck and Schleifer, 2023; Fleck et al., 2022). However, the fact that the problem is quasi-one-dimensional restricts the computational demands of the scenario. Thus, one arrives at the first decision point D1, where the necessity for massive parallelism may be negated.

The next process step P3 involves classifying the problem at hand, following the definition given in section 6.2. As dependent variables, the time  $t$ , as well as the spatial components  $x$  and  $y$ , appear in the governing equations. The coefficient matrix  $a_{ij}$ , summing up all leading coefficients of second derivatives then becomes for the 2D case:

$$a_{ij}^{AC} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (7.7)$$

In this case, enumerating the eigenvalues  $\lambda_i$  is trivial, since  $a_{ij}^{AC}$  is a diagonal matrix, and it follows that  $\lambda_0 = 0$ ,  $\lambda_1 = -1$ ,  $\lambda_2 = -1$ . Thus, one eigenvalue is zero since the temporal derivative is only of the first order and all other eigenvalues are of the same sign. Therefore, Equation 7.1 is a second-order PDE of parabolic type and one can proceed in D2 with the left branch.

Moving on in the decision process, one would next classify the problem domain in D3 given input I3. As an equispaced grid is used in 1D, the discretization is cartesian and thus solving the problem using Finite Difference would be the best choice. As there are no other fields to classify according to decision point D5, the decision process is concluded at this point. Within the unified methodological framework, both FD and CGMs are implemented and the scenario is comparatively solved using both schemes. This allows a comparison of the schemes regarding numerical resolution capabilities and an investigation of the differences in the mutual arithmetic complexity and their impact on efficiency.

Evaluating the CGM requires the re-formulation of Equation 7.1 in its weak form, though. Following the proposed procedure in section 6.8, one must first form the inner product with a scalar test function  $v$ . As a CG discretization is used and the highest order of derivative in the PDE is two, no hybridization is needed. Due to the zero Neumann boundary condition in combination with the CG discretization, the entirety of boundary terms vanishes to zero in the weak form. Thus, the finite-dimensional weak statement is: Find  $\phi_h \in V_h$ , such that

$$\int_{\Omega} \left\langle \frac{1}{K} v_h, \partial_t \phi_h \right\rangle dx + \int_{\Omega} \langle \nabla v_h, \nabla \phi_h \rangle dx - \int_{\Omega} \langle v_h, f(\phi_h) \rangle dx = 0 \quad \forall v_h \in V_h, \quad (7.8)$$

where it has already been assumed that both the solution and test function lie in the finite-dimensional subspace  $V_h$ .

Equation 7.1 requires the discretization of the Laplacian as its only differential operator. The temporal derivative will be treated using the Method of Lines approach, that is, a large system of spatially discretized ordinary differential equations is solved.

The Finite Difference discretization of the laplacian results in the well-known second-order central difference stencil:

$$\Delta \phi \approx \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \quad (7.9)$$

The nonlinear right-hand side  $f(\phi)$  must be updated every time step using the current value of  $\phi$ . As such, there is no need to perform any assembly and forming the global system of equations can even be avoided entirely. Instead, 7.9 is used to model the Laplacian, which can be handily vectorized. There is also no need to perform any mapping between the reference and physical domain as explained in section 5.3.

For the Finite Element discretization, all these steps need to be performed, resulting in a global nonlinear system of equations for each time step. The discrete form of Equation 7.8 then reads:

$$\underline{\underline{M}} \partial_t \underline{\phi} + \underline{\underline{K}} \underline{\phi} - \underline{F}(\underline{\phi}) = 0 \quad (7.10)$$

Where the mass matrix  $\underline{\underline{M}}$  and the stiffness matrix  $\underline{\underline{K}}$  for the Laplacian have been introduced. These represent the spatially discretized differential operators that act on the vector of degrees of freedom  $\underline{\phi}$ . The algebraic terms that are nonlinear in  $\underline{\phi}$  are gathered in the discrete vector  $\underline{F}(\underline{\phi})$ . For the sake of comparison regarding efficiency, the

resulting fields of both schemes are required to be (nearly) identical apart from floating point errors.

Given this requirement, it should be noted that the Finite Difference formulation lacks an analogous term to the Finite Element mass matrix. Consequently,  $M$  is required to be the identity matrix in an equivalent Finite Element formulation, given all other terms are equal. The latter can easily be verified for a stiffness matrix assembled with first-order Lagrange polynomials and a collocation method. The derivation of such an equivalent scheme has been covered in section 5.3. Using collocated Finite Elements is chosen here for the sake of comparison as well as for computational efficiency. The resulting mass matrix can then be inverted trivially by taking the element-wise inverse instead of computing the full inverse. Such an operation is considerably more expensive and should thus be avoided if possible.

To compare both schemes regarding efficiency, they are implemented from scratch within the Julia programming language (Bezanson et al., 2017). Due to its flexibility, high-level syntax and simultaneous, granular control over various performance aspects via its rich type system, Julia has gained considerable momentum in the past few years within the scientific community. Both schemes are carefully set up using analogous data structures to enable a side-to-side comparison of the computational complexity. The most high-level parts of the codes are given in Listing 2. Both implementations contain a function that solves the semidiscrete system at each time step which is passed to the ODE solver. The entire source code is available online at <https://github.com/pzimbrod/multiphysics-pde-methods/tree/main/code/Allen-Cahn>.

In the case of the FDM, one can avoid assembling a global linear system entirely, thus the top-level data structure only holds the solution and grid as large arrays. For the FEM, assembly on general grids in a matrix-free manner is far from trivial. Additionally, the triangulation data structure is more complex due to the necessary topological information. Furthermore, the reference FE needs to be stored and correctly mapped using Jacobian values.

Here, the functions that are called within each time step to solve the semidiscrete system, are also included to give a high-level view of which steps are necessary and how they are implemented in particular. Both semidiscrete systems use in-place operations to avoid memory allocations. For the CG-FEM code, a full mesh topology is implemented to solve the problem with a first-order method although both discretiza-

Listing 2: Overview of the necessary data structures and the functions to update the semi-discrete systems for the CG FEM (left) and FDM (right).

---

```

1 struct FETriangulation{V,C}<:Triangulation
2   vertices::V
3   connectivity::C
4   dim::Int
5 end
6
7 struct FiniteElement{E<:ElementType,
8   P<:Primitive,B<:AbstractMatrix,
9   Q<:AbstractVector,G<:AbstractArray}
10 primitive::P
11 element_type::E
12 order::Int
13 ndofs::Int
14 basis_coeffs::B
15 quadrature_nodes::B
16 quadrature_weights::Q
17 basis_at_quad::B
18 grad_basis_at_quad::G
19 grad_monomial_basis::G
20 end
21
22 struct AssemblyCache{T<:AbstractVecOrMat}
23   coeffs::T
24   loc::T
25   glob::T
26 end
27
28 struct CGProblem{T<:Triangulation,J,P,C
29   E<:FiniteElement,M<:AbstractMatrix,
30   V<:AbstractVector,F<:Function}
31 triangulation::T
32 referenceElements::E
33 detJ::J
34 bilinearForm::M
35 linearForm::V
36 u::V
37 massMatrix::Union{M,Nothing}
38 parameters::P
39 rhs::F
40 cache::C
41 end
42
43 function (a::CGProblem)(du,u,p,t)
44   mesh      = a.triangulation,
45   element   = a.referenceElements
46   K         = a.bilinearForm
47   F         = a.linearForm
48   detJ      = a.detJ
49   params    = a.parameters
50   cache,rhs = a.cache, a.rhs
51   M         = a.massMatrix
52
53   mul!(du,K,u,-1.,0.)
54   # f(u) changes, thus we reassemble
55   assemble_F!(F,cache,u,element,
56     mesh,detJ,params,rhs,M)
57   du .+= F
58
59 end

```

```

struct FDTriangulation{V,D}<:Triangulation
  vertices::D
  h::V      # dx in each dimension
  dim::Int
end

struct FDPProblem{T<:FDTriangulation,
  B<:Function,L<:Function
  V<:AbstractArray,P,BC<:Function}
  triangulation::T
  order::Int
  bilinearForm::B
  linearForm::L
  u::V
  parameters::P
  boundaryCondition::BC
end

function (a::FDPProblem)(du,u,p,t)
  apply_bilinear! = a.bilinearForm
  apply_linear!   = a.linearForm
  apply_bc!       = a.boundaryCondition
  h               = a.grid.h
  params          = a.parameters

  a.bilinearForm(du,u,h)

  a.linearForm(du,u,a.parameters)
  a.boundaryCondition(du)

end

```

---

tions consist of cartesian meshes. One could in this case assume a globally constant Jacobian and thus save a considerable amount of arithmetic complexity. However, this would skew the results regarding performance and would not make full use of the flexibility of the FEM.

It becomes immediately apparent from the comparison that solving the Allen Cahn equation using Finite Elements requires an assembly process that is noticeably more complex. The only arrays that need to be stored for the FD version are the grid coordinates and the solution array. Because the latter can be arranged in memory such that it represents the cartesian topology of the grid, one can simply point to the neighbors of a vertex in memory without having to look up the vertex-vertex connectivity. This is not the case for the FEM. Instead, there is an additional indirection through a cell-vertex list, where all DoFs associated with the currently visited cell are gathered.

Furthermore, it is not possible to construct the global linear system at once. Instead, one needs to go through the cell-wise assembly process which effectively leads to most of the non-zero matrix entries being visited multiple times. This is in sharp contrast to the FDM where the global system is only present implicitly through functions that apply the laplacian stencil. As a consequence, memory requirements are greatly reduced.

For transient problems, one needs to additionally make a suitable choice for the temporal discretization, that is, the choice of method as well as the time step. Here, the well-optimized Julia library `DifferentialEquations.jl` ([Rackauckas and Nie, 2017](#)) is utilized. As an exemplary implementation of modern, high-performance codes for the solution of ordinary differential equations (ODE), this package offers various algorithms that are capable of adaptive time stepping such that an application expert does not need to provide any input regarding temporal discretization. Here in particular, it is even possible to make use of built-in heuristics that automatically select a suitable integration scheme, based on the supplied ODE problem ([Rackauckas and Nie, 2019](#)). The resulting effort for the end user can be condensed to selecting a suitable numerical scheme for the spatial discretization as outlined by [Figure 6.3](#) and leave the problem of tuning the spatial discretization aside entirely. For this particular problem, the use of an adaptive, implicit, 4th-order Rosenbrock method is prescribed for the temporal evolution of both FD and CG-FE systems to achieve a fair comparison between both solutions. This solver is stable and third-order accurate when used on nonlinear parabolic problems ([Rackauckas and Nie, 2017](#)). The spatially discretized, nonlinear problems at

each time step are solved explicitly according to the timestep update functions given in Listing 2.

Before comparing both schemes regarding computational efficiency and to support the theory established in section 5.3, the FD and collocated CG-FE schemes will be shown to indeed produce identical results. Figure 7.1 shows the solutions of both schemes for solving the phase field evolution. The position of the interface over time is depicted by Figure 7.2. As can be observed, both schemes indeed produce visually identical results. The quantitative differences in the numerical results are minimal and can be attributed to floating point errors that accumulate over the process of time integration. However, there is a considerable difference between the analytical and the numerical interface velocity, as visible in Figure 7.2. The reason for this discrepancy is grid friction, which results from the limited numerical resolution of the diffuse interface profile and could be reduced by increasing the dimensionless ratio  $\xi/\Delta x = 1.5$ , where  $\Delta x$  denotes the grid spacing (Fleck and Schleifer, 2023; Fleck et al., 2022). Grid friction and pinning during stationary interface motion has been studied previously, for instance by Bösch et al. (1995); Karma and Rappel (1998). So far, this detrimental effect has been only studied using Finite Difference based schemes. It relates to metastabilities, that result from a broken translational invariance of the discrete numerical

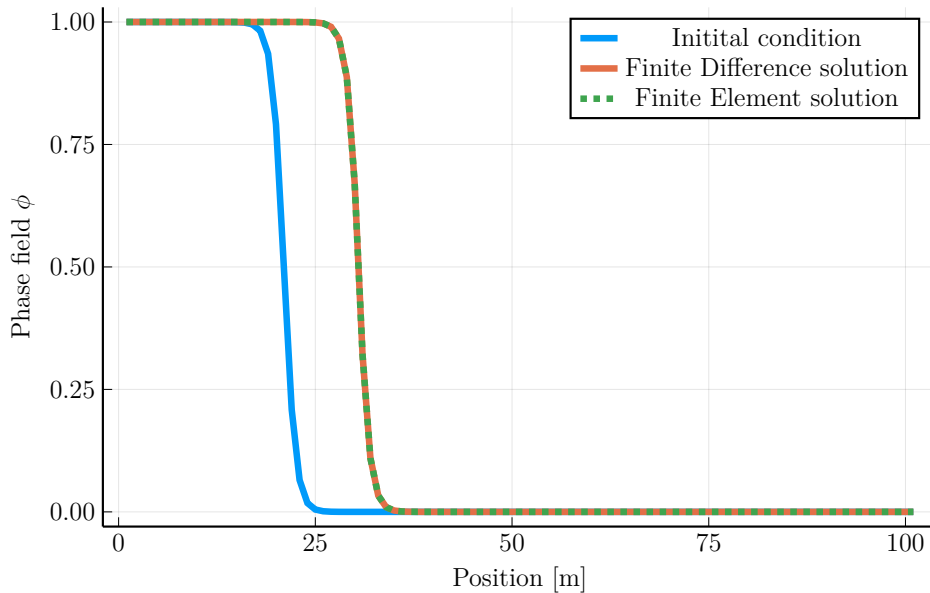


Fig. 7.1: Evolution of the phase front at  $t = 100s$  with respect to the initial condition.

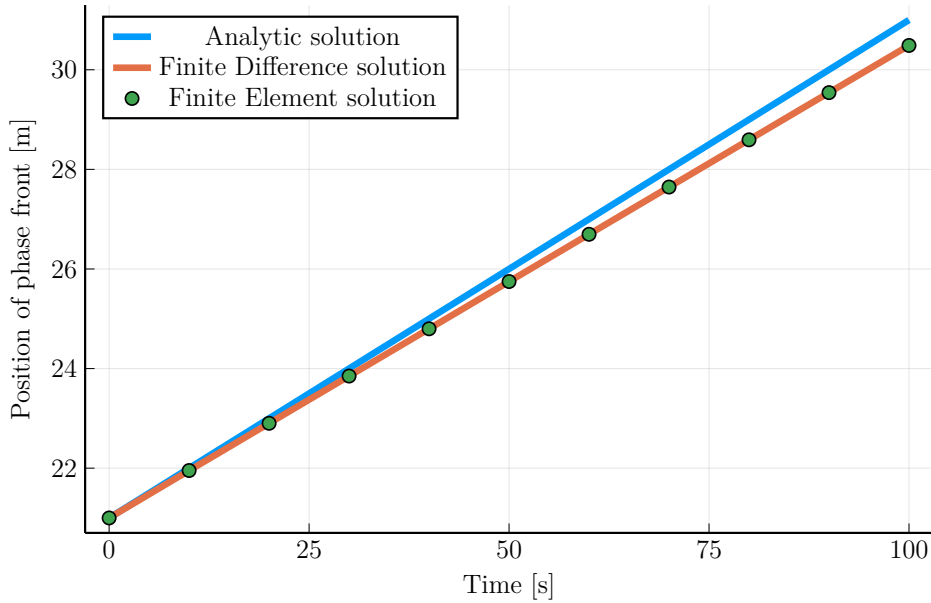


Fig. 7.2: Comparison of Finite Difference and Finite Element solutions to the analytical solution given by Equation 7.6

schemes (Fleck and Schleifer, 2023; Fleck et al., 2022), As a consequence, the current interface velocity oscillates as the center of the interface passes one grid point after the other. If further, the time discretization error is small enough, the average interface velocity turns out to be considerably below the expectation. With decreasing phase-field width, we obtain increasingly larger deviations of the average velocity as well as increasing larger oscillations. This culminates in a vanishing velocity, where the phase field is pinned to the computational grid. Interestingly, spurious grid friction and pinning can be eliminated in Fast Fourier- and Finite Difference implementation of the Allen-Chan equation, using the newly proposed sharp phase field method (Finel et al., 2018; Fleck et al., 2019). For phase field models, which use a double obstacle potential instead of the double well potential, a comparable technique has been proposed by Eiken (2012). It is interesting to note that this discretization error, which is very characteristic of the Allen Cahn equation, turns out to be so similar for the two different numerical schemes in this case. This again highlights the close relationship between the two different numerical schemes. Furthermore, it is noteworthy that computational resource usage differs considerably. Table 7.2 reports some descriptive statistics on the performance of both implementations. Both models were run on identical hardware and

Table 7.2: Run times of the Finite Element and Finite Difference model of the 1D Allen Cahn equation.

	<b>FDM</b>	<b>FEM</b>	<b>Relative</b>
Median run time	0.450 ms	9.503 ms	21.1x
Mean run time $\pm 1\sigma$	0.578 ms $\pm$ 0.759 ms	9.643 ms $\pm$ 0.741 ms	16.7x
Allocated Memory	1.446 kB	15.598 kB	10.8x

on Julia 1.8.5 with LLVM 13.0.1 (Bezanson et al., 2017). Time stepping was performed using the DifferentialEquations.jl library (Rackauckas and Nie, 2017). Linear Algebra operations are performed using OpenBLAS (Wang et al., 2013; Xianyi et al., 2012) on a single-threaded Apple M1Pro ARM processor. Fast evaluation of fused array expressions is provided by the Tullio.jl library (Abbott et al., 2022). Allocated Memory refers to the physical size of the problem-specific data structures given in Listing 2. The sample size for each scheme is  $n = 100$ . These differences in run times as well as memory consumption can be attributed to multiple factors. First, the nonlinear right-hand side changes each time step and thus assembly has to be performed dynamically for the FEM. The FDM in contrast can simply rely on point-wise evaluation of the strong form instead of numerically computing the weak form integrals. Secondly, the FDM does not need to perform any mapping during the time step as no assembly is required. During computation of the right-hand side integral, this is a necessity for the FEM.

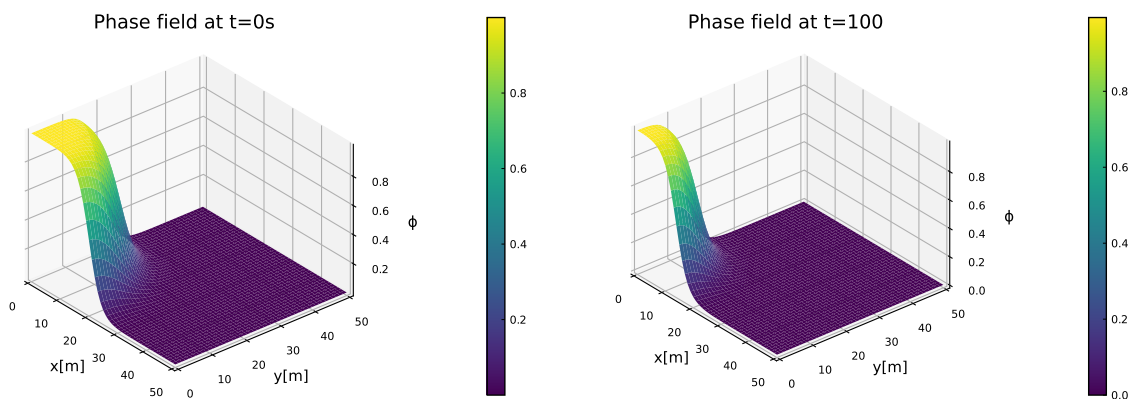
The largest discrepancy however can be attributed to the fact that the FDM can operate in a matrix-free manner due to the cartesian grid it is applied on. As all vertices are equispaced, there exists one global stencil that can be applied on each vertex independent of all other members of the grid. The FEM in contrast uses the grid topology to accumulate the weak form integrals into corresponding entries of the global system matrices and vectors. Thus, it always produces a typically very sparse global system that cannot be vectorized similarly. It should be noted that the discrepancy in results should not be expected to be as drastic as shown for linear PDEs, as then the FEM does not require the re-assembly of the right-hand side. The computational advantage then reduces to the matrix-free evaluation of the linear system.

In the second scenario, a more practically relevant benchmark in two dimensions is investigated in the form of the vanishing grain problem, leaving all other aspects of the problem as is. Here, the dissolution of a circular-shaped nucleus under the interface

energy density pressure under two-phase equilibrium condition  $\mu_0 = 0$  is simulated. These dynamics are also governed by the Allen Cahn equation and denote the complementary physical effects as compared to the above scenario. In a sharp interface picture, with a constant and isotropic interface energy density  $\gamma$ , the temporal evolution of the grain radius is expected to be given by:

$$r(t) = \sqrt{R_0^2 - 2Mt} \quad (7.11)$$

Where  $R_0$  indicates the initial radius and  $M$  is the phase field mobility. Snapshots of the phase field  $\phi$  at initial and terminal times are given in Figure 7.3. In addition, the temporal evolution of the radius function for solving this scenario using both numerical methods is shown in Figure 7.4. As in the one-dimensional simulation, the collocated FE and FD solutions behave identically to each other. Both exhibit a notable discrepancy towards the sharp interface behavior, which again relates to known issues of finite numerical resolution in the phase field simulation (Fleck and Schleifer, 2023). To assess the performance gap of both schemes for higher dimensions, both codes are again benchmarked against each other. The results are given in Table 7.3. Both models were run on identical hardware and on Julia 1.8.5 with LLVM 13.0.1 (Bezanson et al., 2017). Time stepping was performed using the DifferentialEquations.jl library (Rackauckas



(a) Initial configuration of the phase field. The domain shows a quarter slice of the nucleus.

(b) Phase field distribution within the quarter domain after 100 seconds. The nucleus has shrunk to a smaller radius whilst retaining the interface profile.

Fig. 7.3: Solution of the two-dimensional Allen Cahn equation using the FDM and FEM.

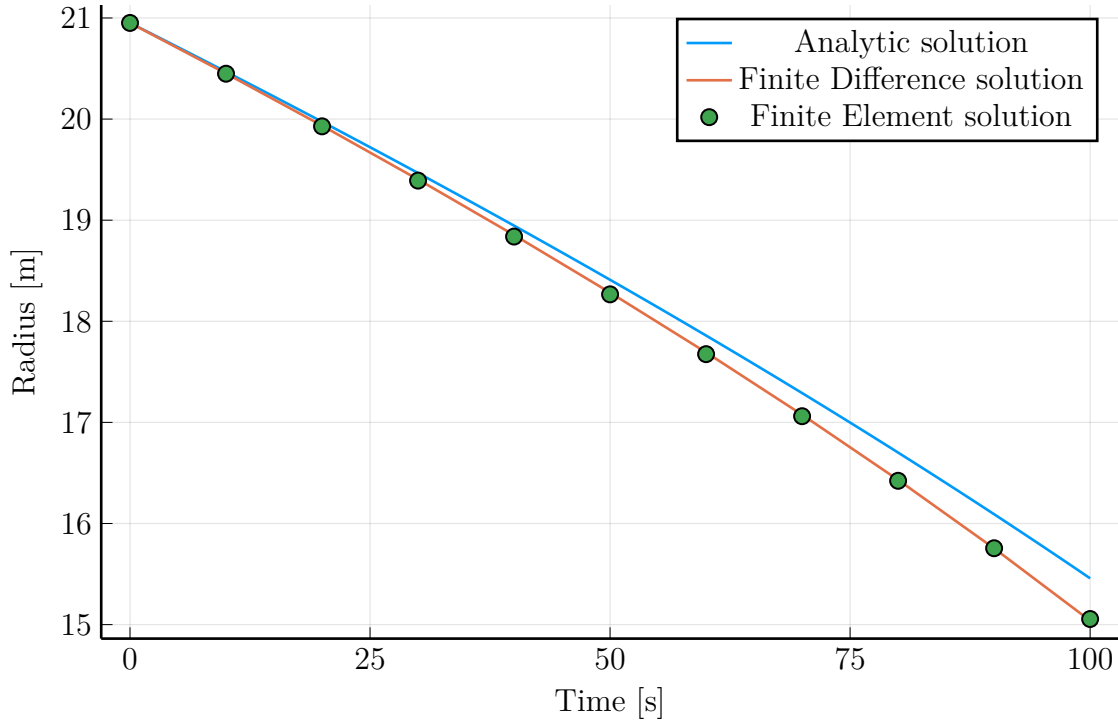


Fig. 7.4: Evolution of radius over time of the vanishing grain problem. Both Finite Difference and Finite Element solutions show a considerable, accumulating error toward the analytical solution.

and Nie, 2017). Linear Algebra operations are performed using OpenBLAS (Wang et al., 2013; Xianyi et al., 2012) on a single-threaded Apple M1Pro ARM processor. Fast evaluation of fused array expressions is provided by the Tullio.jl library (Abbott et al., 2022). Allocated Memory refers to the physical size of the problem-specific data structures given in Listing 2. The sample size for each scheme is  $n = 100$ . Comparing the results from the 2D simulation benchmark in Table 7.3 with its 1D counterpart (Table 7.2), it becomes apparent that the discrepancy in performance becomes noticeably

Table 7.3: Run times of the Finite Element and Finite Difference model of the 2D Allen Cahn equation.

	<b>FDM</b>	<b>FEM</b>	<b>Relative</b>
Median run time	4.743 ms	192.837 ms	40.7x
Mean run time $\pm 1\sigma$	4.731 ms $\pm$ 0.059 ms	192.787 ms $\pm$ 0.219 ms	40.7x
Allocated Memory	21.490 kB	494.730 kB	23.0x

more drastic with increasing dimensionality of the problem. This can be attributed to the increased scattering of DoFs in memory. Thus, memory access is less stridden, increasing the lookup time. For a hardware architecture that demands more parallelism and has a shared memory architecture, this could quickly evolve into a serious bottleneck.

This scenario has been discussed already in many different works. Quite often, it is used to highlight accuracy gains or performance improvements of advanced numerical techniques, that are particularly suggested to solve the Allen-Cahn problem. For instance, [Gräser et al. \(2013\)](#) discussed the scenario in the context of solving the anisotropic Allen-Cahn equation using fully implicit or linearized time discretization and semi-implicit time discretizations and globally convergent truncated nonsmooth Newton methods. They provide information on resulting differences in the achieved accuracies and concerning the complexity of the schemes ([Gräser et al., 2013](#)). Another example is the nonlinear preconditioning for diffuse interface models based on the Allen Cahn equation, as first suggested by [Glasner \(2001\)](#). Interestingly, this preconditioning technique seems to be related to the above-mentioned Sharp Phase Field Method suggested by [Finel et al. \(2018\)](#). Both methods provide a tremendous improvement potential, as demonstrated by a comparative study using this scenario.

## 7.2 Two-phase Advection

As a second model problem, the advection equation in two dimensions is investigated. This problem is well-studied in the literature and is known as challenging to solve accurately. Due to the absence of dissipative terms, numerical algorithms oftentimes struggle to converge towards the entropy solution and either produce spurious oscillations, rendering the solution unstable or yield overly diffusive approximations, where conservation laws are violated ([LeVeque, 1992](#)). Here, this problem is chosen in particular due to being simple yet challenging enough to study. In addition, the advection equation frequently arises in modeling multiple phases in an Eulerian framework and the motion of immersed immiscible fluids in general. It is thus of high relevance in a multitude of multiphysics problems.

In particular, a pure advection problem involving two phases with periodic boundary conditions is considered in the following. As a model assumption, the motion of two

fluids is described by tracking the volume fractions  $\alpha_i$ , as is common for the Volume-of-Fluid (VoF) formulation,

$$\partial_t \alpha + u \frac{\partial \alpha}{\partial x} = 0 \quad (7.12)$$

$$\alpha_1 + \alpha_2 = 1 \quad (7.13)$$

$$\alpha_{h, \tilde{\Gamma}_{b,l}} = \alpha_{h, \tilde{\Gamma}_{t,r}} \quad \forall x \in \partial\Omega \quad (7.14)$$

$$\Omega \in [0; 5] \times [0; 5] \quad (7.15)$$

$$t \in [0; 5], \quad (7.16)$$

where  $\Gamma$  denotes the union of all interior and exterior facets of the domain and  $\tilde{\Gamma}$  are the subsets of the domain boundary  $\partial\Omega$ . In this case specifically,  $\tilde{\Gamma}_{b,l}$  are the slave facets at the bottom and left boundary that the values of the slave facets from the top and right master facets  $\tilde{\Gamma}_{t,r}$  are mapped to. The initial condition to this problem is given as a rectangle function that is one in the interval  $x \in [2; 3] \times [2; 3]$  and zero everywhere else. One may alternatively track only the motion of the interface using a coloring function  $\phi$ . This is common for the level set method, the governing equation however is the same as Equation 7.12.

This problem is solved on three different architectures to showcase the effect of parallelism on the efficiency of numerical schemes. The choices of hardware along with important quantities are given in Table 7.4. Once again, the process summarized in Figure 6.3 is followed to identify the most suitable numerical scheme. Regarding the system of PDEs (I1), Equation 7.13 is simply an algebraic constraint and thus can be calculated in a simple postprocessing step. Thus, 7.13 is not a governing equation in the sense of a PDE and  $\alpha_2$  will consequently not be considered an independent variable, as detailed in section 6.2. Thus, one is left to solve a single scalar advection equation for  $\alpha_1$ .

Proceeding in the flow chart, the hardware scales within P1 are classified next. Here, one finds the last hardware configuration listed in Table 7.4 necessitates the use of schemes that are tailored for high parallelism, as the given amount of 128 processes is above the specified regime where the use of parallelizable algorithms is worthwhile using. Therefore, this configuration should be run using a DGM. For both other configurations using 8 and 18 processes, this does not apply. Continuing with process P2, it

Table 7.4: Hardware configurations for the advection equation model problem. The three setups mimic popular computing environments in applied settings: A mobile computer, a stationary workstation grade tower and a rack-mounted server tailored to numerical computing.

CPU Name	Number of Cores	Core Clock Speed [GHz]	Memory Size [GB]	Memory Bandwidth [GB/s]	Memory Speed [GHz]
Apple M1 Pro	8	3.2	16	200	6.4
Intel Xeon W-2295	18	3.0	128	94	2.9
2x AMD EPYC 7763	128	2.45	512	204	3.2

becomes clear that the given problem only exhibits one length scale and thus this criterion for parallelism can be omitted. Thus, one arrives at decision D1 and finds that for the problem statement involving the largest of the three computing architectures, the use of the DGM is advised.

For the remaining two configurations, one can proceed by classifying the PDE according to process P3. With the temporal derivative and gradient as the only differential operators, Equation 5.1 is a first-order PDE. The advection velocity vector  $\underline{u}$  has constant and real components. Thus, following section 6.2, it becomes apparent that the advection equation presented here is hyperbolic and one can proceed with the right branch of the flow chart after decision D2.

Consequently, the linearity of Equation 7.12 for process P4 needs to be evaluated as a next step. As the terms including the differential operators are linear and there is no right-hand side, it may be classified straightforwardly to be linear. Due to its linearity, the most efficient choice for the remaining configurations turns out to be the DGM as well. As stated previously, the original two-equation system only consists of one PDE, and thus the decision process can be concluded here, as all fields governed by a PDE have been assigned (decision D5).

It should be noted here that when the VoF method is used together with the FVM, one can expect the interface of two phases to diffuse considerably. Within the numer-

ics community, this problem is well-known and has led to the practice of introducing an additional interface compression term into the advection equation (Okagaki et al., 2021):

$$\partial_t \alpha + \underline{u} \cdot \nabla \alpha + \nabla \cdot [\underline{u}_c \alpha (1 - \alpha)] = 0, \quad (7.17)$$

$$\underline{u}_c = \min(c_\alpha |\underline{u}|, \max(|\underline{u}|)) \frac{\nabla \alpha}{|\nabla \alpha|}, \quad (7.18)$$

where the newly introduced, artificial interface compression velocity  $\underline{u}_c$  is used. In this case,  $c_\alpha$  is a positive, scalar constant somewhere around unity. As can easily be observed by combining Equations 7.17 and 7.18, this makes the resulting PDE fully nonlinear which can be regarded as a severe disadvantage in this case. This additional, artificial interface motion counteracts the diffusion that the FVM has shown to exhibit. As will be shown, such an additional term can be omitted if the problem is solved using the recommended DGM due to the considerably lowered numerical diffusion. In summary, choosing the FV over the recommended DGM, in this case, would not only require evaluating a fully nonlinear variant of the advection equation to gain accurate results, but it would also force the application expert to pick an appropriate value for an artificial model constant, which can be subject to tedious calibration work.

In the following, this particular choice of method is compared with the FVM, which would be the next alternative and is in principle also well-suited to tackle such problems. The CGM and FDM do not lend themselves well to solving such equations and will thus be omitted from this benchmark. In particular, the CGM is known to be unstable for first-order hyperbolic equations, as the stability of the scheme can be shown to be dependent on mesh size (Ern and Guermond, 2004).

One must add that in principle, the FDM could be applied here, where however two different limitations apply. First, the only choice of stencil that would be stable for this equation is the forward difference (or upwind) approximation. This choice however is not covered by the proposed decision process, as it is formally equivalent to a Petrov CGM. One can show that this scheme corresponds to a simplified Streamline Upwind Petrov Galerkin (SUPG) method (Brooks and Hughes, 1982). As the scope of this work for the sake of decidability is restricted to Bubnov Galerkin methods, this stencil is not admissible here. Secondly, using the FDM here implies the strict use of a cartesian grid.

Thus, the weak form for Equation 7.12 is required which can be obtained by multiplying with a discrete test function  $v_h$ , integrating over the whole domain  $\Omega$  and subsequently performing integration by parts. As a discontinuous discretization is sought, the hull integrals have to be split into contributions from internal and external facets. Additionally, Equation 7.14 prescribes periodic boundary conditions, i.e. a special instance of Dirichlet conditions. As no Neumann BC is specified, the integral over outer facets becomes zero. Thus, the weak form becomes: Find  $\alpha_h \in V_h$  such that

$$\int_{\Omega} \langle v_h, \partial_t \alpha_h \rangle dx + \int_{\partial\Omega} \langle v_h, \alpha_h (\underline{u} \cdot \underline{n}) \rangle ds - \int_{\Omega} \langle \nabla v_h, \underline{u} \alpha_h \rangle dx = 0 \quad (7.19)$$

for all discrete test functions  $v_h \in V_h$ . To implement the weak form of this equation however, most software packages implementing FEMs require a slightly different formulation of Equation 7.19 that however is straightforward to derive. Further details are given in the appendix section B.1.1. An excerpt of the actual implementation in Python is shown in Listing 3. The entire source code is available online at <https://github.com/pzimbrod/multiphysics-pde-methods/tree/main/code/Advection>. This PDE in combination with periodic boundaries possesses an analytical solution of the form:

$$\alpha(x, t) = \alpha(x - ut, 0) \quad (7.20)$$

That is, after traversing the quadratic domain with the given velocity  $\underline{u} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ , the solution field must exactly correspond to the initial condition. Verification of numerical results is thus very straightforward.

This problem is solved using the Firedrake problem-solving environment along with the popular libraries PETSc and Scotch for efficient parallel computing (Ham et al., 2023; McRae et al., 2016; Homolya and Ham, 2016; Hendrickson and Leland, 1995; Chevalier and Pellegrini, 2008; Rathgeber et al., 2016; Dalcin et al., 2011; Balay et al., 1997, 2023). As the FVM can simply be understood as a DGM of polynomial degree zero, the implementation is virtually the same for both schemes.

Note that for the FVM, the last term in Equation 7.19 becomes zero since the derivative of a constant vanishes. Thus, this term is omitted from the assembly to save computations and to more accurately represent the arithmetic intensity posed by the original formulation of this scheme.

Listing 3: Python code to form the Discontinuous Galerkin problem for the advection equation in Firedrake. Note that the definition of the problem variational form in lines 36 and 37 is very similar to the formal definition given in Equation 7.19.

---

```

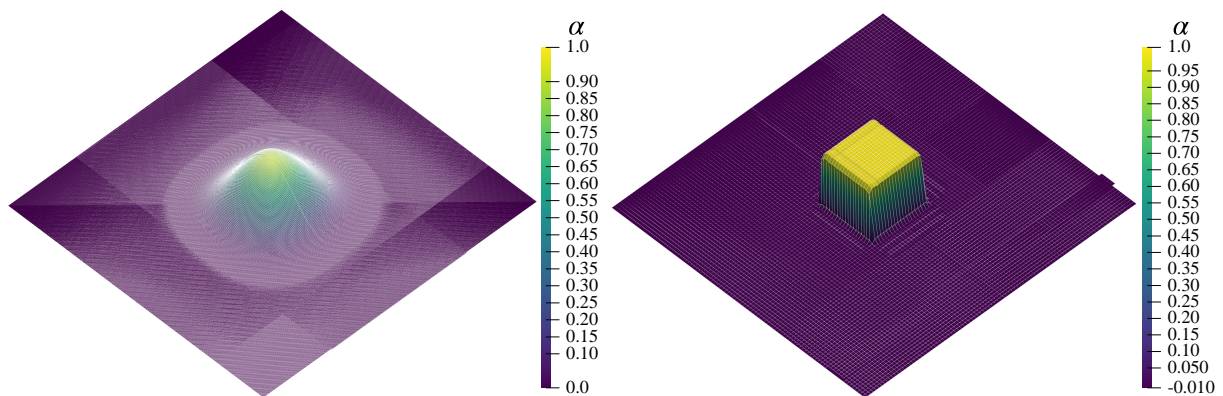
1 from firedrake import *
2 import numpy as np
3
4 n = 384          # Number of cells
5 L = 5.          # Domain length
6 mesh = PeriodicSquareMesh(n,n,L,quadrilateral=True)
7
8 degree = 4
9 fe = FiniteElement("DQ", quadrilateral, degree, variant="spectral")
10 V = FunctionSpace(mesh, fe)
11 W = VectorFunctionSpace(mesh, "DQ", 0) # Velocity
12
13 # Project constant velocity into function space
14 velocity = [1.,1.]
15 x, y = SpatialCoordinate(mesh)
16 vel = Function(W).interpolate(as_vector(velocity))
17
18 # Time domain and time stepping
19 T = 5.
20 # CFL = (u_max * dt) / (dx * (2p+1))  <=> dt = CFL * (dx*(2p+1))/vel_max
21 vel_max = np.sqrt(np.sum(velocity))
22 CFL = 0.2 * 1./(2*degree+1)
23 h = L / n
24 dt = CFL * h/vel_max
25 dt_ufl = Constant(dt)
26
27 test = TestFunction(V)
28 phi = Function(V)
29 dphi = TrialFunction(V)
30
31 # Define \vec{u} * \vec{n}
32 n = FacetNormal(mesh)
33 v_max, v_min = max(max(velocity),0), min(min(velocity),0)
34 flux = dot(avg(vel*phi),n('+')) + 0.5*v_max*jump(phi)
35
36 F = test * dphi * dx
37 F += dt_ufl *(phi*div(test*vel)*dx - jump(test)*flux*dS)
38 a, l = lhs(F), rhs(F)
39
40 phi_solution = Function(V)
41 prob = LinearVariationalProblem(a, l, phi_solution)
42 params = {"mat_type": "matfree", "ksp_type": "cg",
43          "ksp_monitor": None, "pc_type": "none"}
44 solv = LinearVariationalSolver(prob, solver_parameters=params)

```

---

For the sake of visualization, the solution is projected onto a first-degree space with  $H^1$  continuity. The equations are solved by a three-stage implicit Runge Kutta method. In both cases, careful attention has to be paid regarding the time step. For hyperbolic problems of such time, the time step where stability is given is strictly bounded by the Courant Friedrichs Lewy number  $CFL \leq \frac{1}{2k+1}$  for a scheme of degree  $k$  (Cockburn and Shu, 2001). One can easily verify that for a Finite Volume scheme, this corresponds to the well-known condition that the CFL number must stay at or below unity. Not only does that mean regarding arithmetic complexity that the FVM has a simplified assembly process, but also that the admissible time step is in general larger than for DGMs. This discrepancy drastically increases with the polynomial order taken for the DGM.

The corresponding results of the simulation are shown in Figure 7.5. The two numerical schemes produce approximations of obviously different quality. The DGM can capture the rectangular profile throughout the simulation with relatively good accuracy, while the FV simulation is strongly diffused. The latter is due to the missing artificial interface compression, given in Equation 7.17, resulting in a substantial numerical loss of the conserved quantity  $\alpha$ .



(a) Finite Volume solution at end time on a 384x384 grid. The solution is heavily diffused to a parabolic profile. (b) Discontinuous Galerkin solution at end time on a 96x96 grid using third order polynomials. One can observe the presence of spurious oscillations that remain stable in magnitude throughout the simulation.

Fig. 7.5: Solution of the two-dimensional advection equation using the Finite Volume and DGM. Both models were run using the identical amount of global degrees of freedom set to 147.456. Element facets are drawn in white to illustrate the difference in grid size.

At this point, it should be noted that there are formulations of the FVM that capture shocks much more accurately whilst controlling oscillations. Due to the maturity of this method, the field of constructing Weighted Essentially Non-Oscillatory schemes (WENO) is quite advanced and also, in this case, will yield better approximations. Such schemes are also applicable to considerably more complex systems of equations (Zimbrod et al., 2022). However, this argument also applies to the DGM since it is as has been shown an extension of the FVM. To make the comparison fair and to be able to rely on existing tools, both schemes are compared using the same, relatively simple reconstruction technique. For both cases, the use of WENO schemes would increase the computational load considerably, as these need to reconstruct polynomial approximations for each flux using relatively wide local stencils - similar to a high order Finite Difference scheme (Liu and Zhang, 2013). The choice of higher-order reconstruction techniques however should not affect the qualitative difference in approximation properties and performance. For example, Zhou et al. report very similar findings for higher-order WENO schemes for the advection equation in two dimensions (Zhou et al., 2001).

In addition to the previous comparison regarding accuracy, the capabilities for parallel computing are also benchmarked using the three machines given in Table 7.4 (c.f. page 96). To properly scale up this problem, the amount of Degrees of Freedom per core is kept constant for the three environments, that is, in this case around 18.400 DoFs/Core. This is in the vicinity of the 25.000 DoFs/Core regime, where beyond that point considerable drop-offs in performance are to be expected (Badia et al., 2020). According to the theory, the DGM should perform with a noticeably higher efficiency due to less communication overhead between processes. This is due to the reduced amount of cells for the same amount of DoFs and as such, the DGM has a much higher ratio of DoFs that lie inside the cell instead of at the boundary. As a consequence, there are fewer DoFs in relative terms that require the evaluation of a numerical flux and thus not as much overhead due to MPI efforts. This relationship is visualized for reference in Figure 7.6. As the FVM is restricted to one DoF per cell, there are no interior DoFs, increasing the necessary MPI effort. The DGM, on the other hand, has an increasing amount of interior DoFs that for a collocation method do not contribute to the numerical flux. For the present order four scheme, the ratio of interior to total DoFs is 36%. The benchmarking results for all three hardware configurations are given in Figure 7.7. The total run times are reported as well as the solution time for one singular time step.

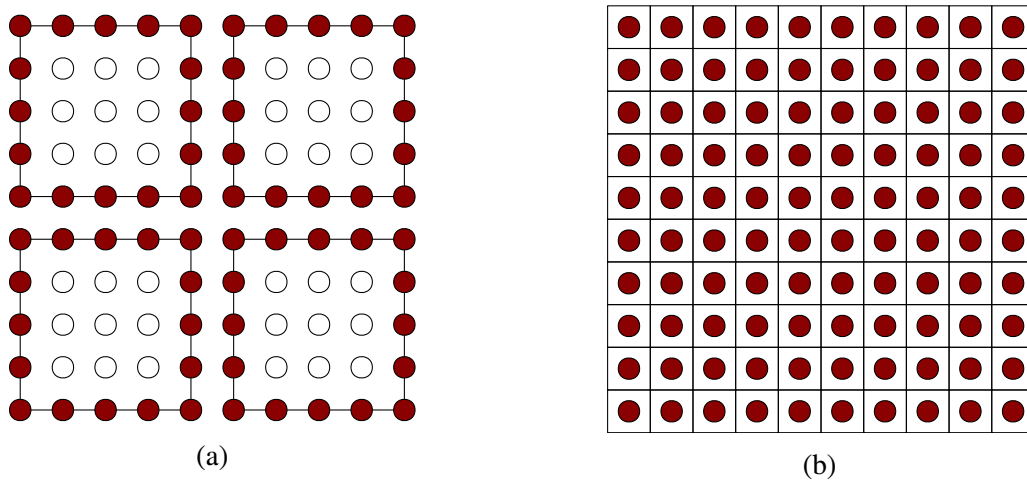
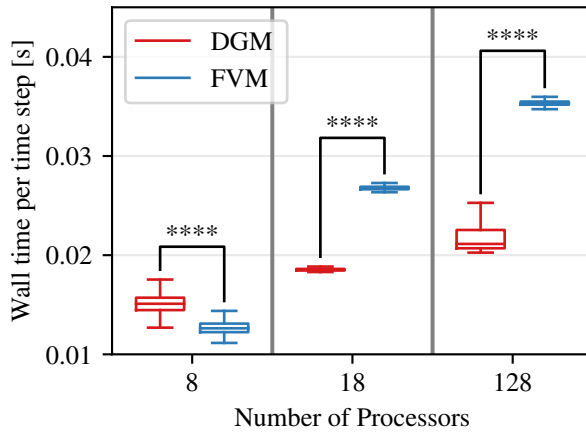


Fig. 7.6: Comparison of shared DoFs (colored dots) between cells for a fourth-order DGM with an FV method with an equal amount of total DoFs (100).



Cores	Wall time [s]	
	FVM	DGM
8	45.01	77.76
18	114.77	136.92
128	409.12	434.22

Fig. 7.7: Comparison of run times for the DGM and FVM advection benchmark case, run on the three configurations given in Table 7.4.

*Left:* Weak scaling of both models, measured as wall time per time step solve. The amount of DoFs/Core was fixed at 18.400. For each sample population,  $n \approx 3000$ . Horizontal lines: Median, Boxes: IQR, Whiskers: Quartiles  $\pm 1.5$  IQR, \*\*\*\*:  $p < 0.0001$ .

*Right:* Total Wall times for the solution over all time steps.

This is because as explained the DG model has a considerably smaller admissible time step. Thus, comparing overall run times is not suitable to empirically validate the above claims regarding parallel efficiency. The overall computation time is nonetheless an important factor in terms of practicality since this is the main quantity one is interested in when performing a simulation. It becomes evident from the reported wall times that the FV model only has an advantage on the small desktop machine concerning solution time. As predicted, scaling up the problem size and number of workers will yield an increasing advantage for the higher-order DGM. The differences in computation time which were found to be empirically significant grow with increasing problem size, thus supporting the claim that DGMs are more favorable for highly parallel architectures.

A similar trend is visible from the table reporting overall runtimes. The time step restrictions as discussed weaken the computational advantage gained in the solution of the semidiscrete system. However, the difference in run time still decreases noticeably with increasing problem and hardware size. For the largest machine with 128 cores, solution times are almost comparable, whereas in the case of the smallest machine, the FV model computes a solution about 42% faster.



## Chapter 8

# Application to Melt Flow Evolution in Metal AM

As a last example, a real multiphysics problem is investigated to demonstrate the proposed method for more complex settings. The results from the problem analysis summarised in Figure 6.3 are shown and an outline is given of how one may implement the resulting numerical scheme. The problem of interest is mesoscopic fluid flow behavior during Laser Powder Bed fusion, as introduced in section 3.2.3.

Having detailed knowledge about the fluid flow and temperature fields is crucial to obtain dense, that is, pore-free parts with a favorable microstructure. The former is governed by the morphology of the solidified melt pool, whereas the latter is heavily influenced by the spatial and temporal temperature gradients. A graphic representation of the computational problem is shown in Figure 3.3 on page 36.

The various numerical methods that have been used to tackle this problem have been discussed in section 3.2.3. Setting up the corresponding simulation problem requires knowledge of the intricate physics happening at that scale, which is governed by multiple, coupled PDEs. For an accurate representation of the multiple phases, some modeling of that has to be taken as preliminary, as there exist many different choices of PDE. In the following, the phase representation that is most often used in the FVM is employed, as this is the most widespread in section 3.2.3.

## 8.1 Physics and Governing Equations

Due to the presence of solid, liquid and gaseous phases, resolving the thermo fluid dynamics of the process involves a multitude of physics. Most prominently, research has shown that the accurate representation of surface tension forces due to temperature gradient plays a key role in obtaining realistic results regarding the morphology of the solidified tracks, temperature gradients and presence of pores - all of which have shown to be important indicators for process quality (DebRoy et al., 2018).

The multiphase problem that arises from this application is discretized using the Volume-of-Fluid method. The equations for this model have been introduced in section 7.2. In total, there are one solid metallic, one liquid metallic and two gaseous phases for vaporized metal and the shielding gas, yielding four phase fractions that need to be tracked. By the previous discussion in section 7.2, one may omit to model one of these phases using a conservation law as one can calculate it using the compatibility condition in Equation 7.13 as well.

With the phase fractions in place, one then obtains the mixed material constants at a given degree of freedom by a rule of mixture law. For thermal conductivity  $\kappa$ , for instance, the phase averaged value at DoF  $i$  is:

$$\kappa_{\text{VOF}} = \sum_{k=0}^{N_{\text{Phases}}} \kappa_k \alpha_k \quad (8.1)$$

The next step consists of enumerating the governing equations of this problem. For the melting of metallic materials, one finds that the flow field can be described by the incompressible Navier Stokes equations in the low Reynold's number regime, plus some source terms that account for the additional physics

$$\partial_t(\rho \underline{u}) + \nabla \cdot \left[ (p - p_{\text{recoil}}) \underline{\underline{I}} \right] - \rho \underline{k} - \eta \Delta \underline{u} + \nabla \cdot \underline{\underline{T}} = 0, \quad (8.2)$$

with the additional condition of incompressibility

$$\nabla \cdot \underline{u} = 0. \quad (8.3)$$

Here,  $\underline{k}$  is the vector of gravity and other volumetric forces.

One important driving force for melt flow is the temperature-dependent surface ten-

sion force, also called the Marangoni force. All capillary forces normal and tangential to the interface are summarised using the divergence of the so-called capillary stress tensor:

$$-\nabla \cdot \underline{\underline{T}} = -\nabla \cdot \left[ -\sigma(\underline{\underline{I}} - \underline{n} \otimes \underline{n}) \right], \quad (8.4)$$

where  $\sigma$  denotes the temperature-dependent coefficient of surface tension,  $\underline{\underline{I}}$  is the identity tensor and  $\underline{n}$  is the unit normal vector to the capillary surface. The divergence of this quantity represents the capillary force acting on interfaces (Lafaurie et al., 1994).

This in turn needs to be computed using the phase fractions and the surface gradient operator  $\nabla_s$  (Brackbill et al., 1992)

$$\underline{n} = \frac{\nabla_s \alpha}{|\nabla_s \alpha|}, \quad (8.5)$$

$$\nabla_s(\alpha_1, \alpha_2) = \alpha_1 \nabla \alpha_2 - \alpha_2 \nabla \alpha_1. \quad (8.6)$$

For the gaseous phase of vaporized material, the recoil pressure

$$p_{\text{recoil}}(T) = 0.53 p_0 \frac{L_v}{R} \exp\left(\frac{1}{T_v} - \frac{1}{T}\right) \quad (8.7)$$

is additionally considered. The only conserved quantity left is the total energy of the system. The corresponding balance equation that governs the evolution of temperature can be thought as a physically extended version of the heat equation (2.5). Adding convective transport as well as some source terms, the balance equation reads:

$$\partial_t(\rho c_p T) + \underline{u} \cdot \nabla(\rho c_p T) + \sum_i \rho L \partial_t \alpha_i - \nabla \cdot (\kappa \nabla T) + Q_{\text{Laser}} - Q_{\text{Vap}} - Q_{\text{Radiation}} = 0 \quad (8.8)$$

with the following (nonlinear) source terms for the laser heat input, vaporization loss and radiation loss, respectively (Bayat et al., 2019):

$$Q_{\text{Laser}}(x, y) = P_L \frac{2\tilde{\alpha}}{\pi R_L^2} \exp\left(-\frac{2(x^2 + y^2)}{R_L^2}\right), \quad (8.9)$$

$$Q_{\text{Vap}}(T) = 0.82 \frac{p_{\text{recoil}}}{\sqrt{2\pi M R T}}, \quad (8.10)$$

$$Q_{\text{Radiation}}(T) = \sigma \epsilon (T^4 - T_{\text{amb}}^4). \quad (8.11)$$

The laser that provides  $Q_{\text{Laser}}$  with total power  $P_L$  and focal radius  $R_L$  travels along a straight line throughout the domain with constant velocity. The absorbed power is dependent on the value for laser absorptivity  $\tilde{\alpha}$ . Thus, this source term represents a transient heat source where its position can be easily interpolated given the current time. Further details on the physical models can, for example, be found in [Zimbrod et al. \(2022\)](#).

## 8.2 Computational Domain

The given problem is solved on a grid that is overall three-dimensional and hexahedral in shape. The bounding box has dimensions 2.0 mm in length, 0.3 mm in width and 0.5 mm in height.

In this problem setting, steep temperature gradients are to be expected due to the concentrated heat input. As a result, it makes sense to employ a finer grid at locations where the melt pool dynamics take place. One possible way to account for that is to use a discretization technique that can incorporate adaptive grids.

However, one would instead like to make use of previous knowledge and only use a finer grid around the vicinity of the laser path - where the actual melting takes place and a finer grid is needed to appropriately resolve the flow field. As such, there is no need to use algorithms that flag candidate cells and employ re-meshing at every time step. The rest of the domain can be meshed using coarser cells, as they are primarily present in the simulation to properly account for heat conduction to surrounding solidified material.

A slice of the resulting discretization is depicted in [Figure 8.1](#). The actual mesh is box-shaped, and the visible regions are colored to illustrate the different phases present as well as the varying cell sizes. The grid consists of hexahedral elements with varying sizing. Colors denote different phases in the initial condition. That is, blue-marked cells contain the shielding gas, cells in red denote the already solidified metallic material of previous layers and green cells denote the freshly applied metallic powder of the current layer. One can observe that the cells become coarser with increasing depth about the powder bed.

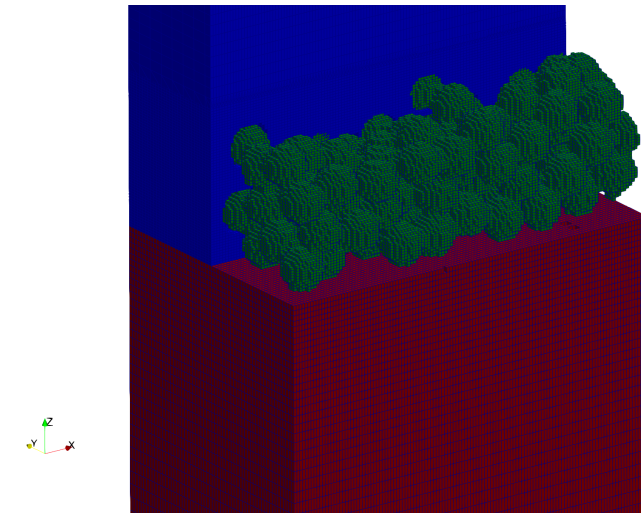


Fig. 8.1: Slice of the computational domain for the Laser Powder Bed Fusion test case.

### 8.3 Computing Resources

The selection process for suitable numerical schemes as done in the previous sections begins by investigating the computational hardware, as stated within process P1 in Figure 6.3. For this problem, the same hardware given in Table 7.4 is used. That is, this problem is to be solved on the 18-core CPU machine. As given in section 6.5, such a system of desktop scale does not per se necessitate the use of massively parallel hardware, as the amount of processes falls below the recommended amount of 50 to 100 workers.

### 8.4 Problem Scale

Using the system of PDEs as input, one can proceed with analyzing the problem length scales according to process P2 in Figure 6.3. Given the coefficients and material properties of the PDE system, it should be noted that all physics are expected to operate on a length scale of a few micrometers. One can thus conclude that this particular problem does not exhibit multiscale properties that would necessitate adaptivity. Therefore, in combination with the given hardware, the first decision point D1 in Figure 6.3 can be negated and one can proceed with field-wise analysis of the PDE system.

## 8.5 Classification

As a first step, the field-wise classification of the PDE system is done, that is, decisions D2 to D5 along with the corresponding processes as follows: All governing equations of the system (Equations 8.4-8.11) are considered individually for each field quantity. That is, the individual equations are extracted that one needs to solve. Afterward, the classification methods described by P3 and P4 are used, whereas the latter is only relevant for hyperbolic systems as illustrated in Figure 6.3.

First, the phase fraction fields are examined. As all variables are governed by the same set of equations and thus the resulting classification applies to every field, the discussion can be abbreviated by only classifying one phase fraction  $\alpha$ . The relevant governing equation then is exactly the passive advection equation that has been introduced in section 7.2

$$\partial_t \alpha + \nabla \cdot (\underline{u} \alpha) = 0. \quad (8.12)$$

It then immediately follows from the previous discussion that the phases are governed by first-order, linear, hyperbolic systems. One consequently arrives at the DGM for these variables.

Next, the velocity vector  $\underline{u}$  will be addressed. Extracting all components from Equation 8.2 that contain differentials of  $\underline{u}$  yields:

$$\rho \partial_t \underline{u} - \eta \Delta \underline{u} = 0 \quad (8.13)$$

This system of equations, for each component, possesses the exact structure of the heat diffusion equation, which is known to be a model parabolic equation. Following the more rigorous approach given in section 6.2, one would obtain for the coefficient matrix  $a_{ij}$

$$a_{ij}^u = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\eta & 0 \\ 0 & 0 & -\eta \end{bmatrix}. \quad (8.14)$$

The first zero diagonal element is due to the time component only appearing in a first, but not in any second derivative. Consequently, the same conclusion can be drawn, being that the governing equation for velocity in this case is parabolic. Thus, the left branch

after D2 applies and the next step involves classifying the regularity of the domain. The question of whether the grid is strictly regular and cartesian has already been answered in the previous section, and therefore one arrives at the CGM for the velocity field.

The pressure  $p$  only appears within the momentum balance in Equation 8.2 within the expression  $\nabla \cdot p\underline{I}$ , which is equal to the gradient  $\nabla p$ . That is,  $\partial_x p$  appears as a source term in the components in this equation and thus the evolution of  $p$  is not per se given by a separate governing equation. Therefore, one needs to fall back to the qualitative classification that was described in section 6.2 and choose an appropriate class based on the continuity requirements of the pressure field. The present problem consists of multiple phases, therefore steep jumps in pressure can be expected at interfaces. Consequently, the pressure field can be assumed to be discontinuous at some points during the simulation and thus follow the process in Figure 6.3 along the path of hyperbolic equations. As there are no additional source terms or nonlinearities in  $p$ , this constraint is classified as linear in D4 and therefore, one also arrives at the DGM.

For the remaining Temperature field  $T$ , the terms containing differential operators on  $T$  are gathered from the governing equation, Equation 8.8:

$$\rho c_p \partial_t T + \rho c_p \underline{u} \cdot \nabla T - \kappa \Delta T = \text{RHS}(T). \quad (8.15)$$

The remaining terms in Equation 8.8 that have been gathered in  $\text{RHS}(T)$  are either constant, or depend on  $T$  itself, but not in its derivatives. Therefore, one may omit them for the sake of classification. As such, the corresponding PDE is of second order and one may proceed analogously to the classification of the velocity  $\underline{u}$ . The coefficient matrix  $a_{ij}^T$  is then

$$a_{ij}^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\kappa & 0 \\ 0 & 0 & -\kappa \end{bmatrix}. \quad (8.16)$$

Similarly to the previous discussion, this resembles a parabolic system and the discretization of the temperature field should be done using the CGM, as the question of domain regularity (D3) has already been addressed and is the same for all fields. If the coefficient of thermal diffusion  $\kappa$  would be zero in this case, the governing equation would then collapse into a hyperbolic PDE and another classification would apply. However, the amount of heat conduction present in the model cannot be neglected, and

thus it follows that  $\kappa > 0$  everywhere in the domain.

## 8.6 Weak Formulation

As has been demonstrated in section 6.8, producing a valid weak formulation out of the strong form of a PDE is not always trivial. Thus, this process will be carried out for the mesoscale PBF-LB/M system exemplarily.

First, the set of advection equations for the Volume-of-Fluid method is considered that is described by Equation 8.12. The process of obtaining a weak formulation has largely been addressed in section 7.2. One important distinction, however, is that the advection velocity vector  $\underline{u}$  is no longer constant with a fixed, prescribed velocity field as it is separately solved for. For the three present phases, the weak form then reads according to Equation 7.19 and the notation introduced in Figure 6.4

$$\int_{\Omega} \langle v_{\alpha}, \partial_t \alpha \rangle dx + \int_{\partial\Omega} \langle v_{\alpha}, \alpha (\underline{u} \cdot \underline{n}) \rangle ds + \int_{\partial\Omega} \langle v_{\alpha}, \alpha (\underline{u} \cdot \underline{n}) \rangle dS - \int_{\Omega} \langle \nabla v_{\alpha}, \underline{u} \alpha \rangle dx = 0, \quad (8.17)$$

where  $v_{\alpha}$  denotes the test function belonging to the same function space as  $\alpha$ . As the fields are to be discretized using DGM, the hull integral resulting from partial integration does not vanish.

Next, the incompressibility constraint, given by Equation 8.3 is transformed. According to Table 8.1, the velocity field  $\underline{u}$  should be discretized using the CGM. Since the differential order of this equation does not exceed two, no hybridization is needed. The dimensionality of the corresponding test function has to be scalar in this case, since  $L(\underline{u}) = \nabla \cdot \underline{u}$  is scalar as well. Thus, the preliminary weak form reads

$$\int_{\Omega} \langle v_{\alpha}, \nabla \cdot \underline{u} \rangle dx = 0. \quad (8.18)$$

As this differential operator is linear according to Definition 6.4.1, one may proceed by integration by parts. The rule for performing integration by parts in case of the divergence of a vector quantity is given in Table 6.1. As the velocity vector is modeled using

a continuous approximation, only the boundary integral over exterior facets remains. Thus, the weak form reads

$$\int_{\partial\Omega} \langle v_u, \underline{u} \cdot \underline{n} \rangle dS - \int_{\Omega} \langle \nabla v_u, \underline{u} \rangle dx = 0. \quad (8.19)$$

The weak form of the momentum balance Equation 8.2 may be formed accordingly. The capillary stress term will be dealt with separately, as it does not involve the flow velocity  $\underline{u}$  and forms part of a more complicated expression requiring more in-depth analysis. As the remainder of Equation 8.2 is linear in  $\underline{u}$ , one can proceed as previously by writing down the integral form

$$\int_{\Omega} \langle \underline{v}_u, \rho \partial_t \underline{u} \rangle dx - \int_{\Omega} \langle \underline{v}_u, \nabla \cdot (\eta \nabla \underline{u}) \rangle dx. \quad (8.20)$$

Since this equation describes the evolution of the vector quantity  $\underline{u}$ , a corresponding test function  $\underline{v}_u$  is required to form the inner product. Partial integration of the viscous term in Equation 8.20 can also be performed using Table 6.1 and yields

$$\int_{\Omega} \langle \underline{v}_u, \rho \partial_t \underline{u} \rangle dx - \int_{\partial\Omega} \langle \underline{v}_u, \eta \nabla \underline{u} \cdot \underline{n} \rangle dS + \int_{\Omega} \langle \nabla \underline{v}_u, \eta \nabla \underline{u} \rangle dx. \quad (8.21)$$

The term containing the static pressure  $p$ , i.e.  $L(p) = \nabla \cdot (p \underline{I}) = \nabla p$  can be transformed straightforwardly. Using the same vector test function  $\underline{v}_u$ , one obtains

$$\int_{\Omega} \langle \underline{v}_u, \nabla p \rangle dx. \quad (8.22)$$

Again performing integration by parts for a DG discretisation yields

$$\int_{\partial\Omega} \langle \underline{v}_u, p \underline{n} \rangle ds + \int_{\partial\Omega} \langle \underline{v}_u, p \underline{n} \rangle dS - \int_{\Omega} \langle \nabla \cdot \underline{v}_u, p \rangle dx. \quad (8.23)$$

The second part of the pressure term has to be treated differently, however. The recoil pressure is a function of temperature according to Equation 8.7 in the form of a nonlinear equation. However, as the divergence encompasses the entire expression, one may still

perform integration by parts, yielding

$$\int_{\partial\Omega} \langle \underline{v}_u, p_{\text{recoil}} \underline{I} \cdot \underline{n} \rangle dS - \int_{\Omega} \langle \nabla \underline{v}_u, p_{\text{recoil}} \underline{I} \rangle dx. \quad (8.24)$$

The capillary stress term is effectively a function of the phase fractions, although through some indirections. Combining Equations 8.4 through 8.6, it becomes apparent that one needs to compute the second derivative of the phase fractions. As they are discretized using a DGM, hybridization is needed here. In case of an interface between phases  $a$  and  $b$ , the hybridization for the unit interface normal  $\hat{\underline{n}}_{ab}$  is already given by 8.5:

$$\hat{\underline{n}}_{ab} - \frac{\alpha_s \nabla \alpha_g - \alpha_g \nabla \alpha_s}{|\alpha_s \nabla \alpha_g - \alpha_g \nabla \alpha_s|} = 0. \quad (8.25)$$

However, by introducing the surface gradient given by Equation 8.6, there are still gradients of the discontinuous phase field remaining which are problematic to the DG discretization. Thus, the additional hybrid quantity

$$\hat{\underline{\alpha}} = \nabla \alpha \quad (8.26)$$

is introduced. Through this indirection, the expression for the unit interface normal  $\hat{\underline{n}}_{ab}$  between solid and liquid phase becomes

$$\hat{\underline{n}}_{ab} - \frac{\alpha_s \hat{\underline{\alpha}}_l - \alpha_l \hat{\underline{\alpha}}_s}{|\alpha_s \hat{\underline{\alpha}}_l - \alpha_l \hat{\underline{\alpha}}_s|} = 0. \quad (8.27)$$

The capillary stress tensor can then be expressed as

$$\underline{T} = -\nabla \cdot \left[ -\sigma \left( \underline{I} - \hat{\underline{n}}_{ab} \otimes \hat{\underline{n}}_{ab} \right) \right]. \quad (8.28)$$

This system can now be transformed into the corresponding weak form. As a starting point, the hybridized interface normal given by Equation 8.26 is considered. Forming the inner product with a vector test function, integrating over the domain  $\Omega$  and subsequently integrating by parts yields

$$\int_{\Omega} \langle \underline{v}, \hat{\underline{\alpha}} \rangle dx - \int_{\partial\Omega} \langle \underline{v}, \alpha \underline{n} \rangle ds - \int_{\partial\Omega} \langle \underline{v}, \alpha \underline{n} \rangle dS + \int_{\Omega} \langle \nabla \cdot \underline{v}, \alpha \rangle dx = 0. \quad (8.29)$$

Equation 8.27 is nonlinear, thus forming the weak form simply consists of forming the inner product with the test function and integrating over the domain. The corresponding weak form then reads

$$\int_{\Omega} \langle \underline{v}, \hat{\underline{n}}_{ab} \rangle dx - \int_{\Omega} \left\langle \underline{v}, \frac{\alpha_s \hat{\underline{\alpha}}_l - \alpha_l \hat{\underline{\alpha}}_s}{|\alpha_s \hat{\underline{\alpha}}_l - \alpha_l \hat{\underline{\alpha}}_s|} \right\rangle dx = 0, \quad (8.30)$$

where the absolute magnitude of a vector quantity  $|\cdot|$  can be computed by taking the square root of the inner product with itself  $\sqrt{\langle \cdot, \cdot \rangle}$ .

Finally, the capillary stress tensor can be expressed in its weak form. Although this expression is again nonlinear, it is still possible to integrate by parts due to the nabla operator encompassing the entire expression. The weak form is thus formulated as

$$\begin{aligned} \int_{\Omega} \left\langle \nabla \underline{v}, -\sigma \left( \underline{\underline{I}} - \hat{\underline{n}}_{ab} \otimes \hat{\underline{n}}_{ab} \right) \right\rangle dx - \int_{\partial\Omega} \left\langle \underline{v}, -\sigma \left( \underline{\underline{I}} - \hat{\underline{n}}_{ab} \otimes \hat{\underline{n}}_{ab} \right) \underline{n} \right\rangle ds \\ - \int_{\partial\Omega} \left\langle \underline{v}, -\sigma \left( \underline{\underline{I}} - \hat{\underline{n}}_{ab} \otimes \hat{\underline{n}}_{ab} \right) \underline{n} \right\rangle dS. \end{aligned} \quad (8.31)$$

This concludes the weak form of the momentum balance. As the temperature is approximated using a continuous discretization and Equation 8.8 is a second-order, semi-linear PDE, the weak formulation is considerably less tedious to derive. Here, one can proceed in a very similar manner to the Allen Cahn equation discussed in section 7.1. Both equations are primarily parabolic containing a nonlinear right-hand side. The weak temporal derivative of the temperature  $T$  thus reads

$$\int_{\Omega} \langle v_T, \rho c_p \partial_t T \rangle dx. \quad (8.32)$$

The diffusive term is again formed similarly

$$\int_{\partial\Omega} \langle v_T, \kappa \nabla T \cdot \underline{n} \rangle dS - \int_{\Omega} \langle \nabla v_T, \kappa \nabla T \rangle dx. \quad (8.33)$$

Table 8.1: Resulting schemes of the proposed decision method for the Laser Powder Bed Fusion example problem. Columns D1 to D4 refer to the corresponding decision points in Figure 6.3.

Variable	D1	D2	D3	D4	Result
$\alpha_{\text{solid}}$	no	hyperbolic	n.a.	linear	DGM
$\alpha_{\text{liquid}}$	no	hyperbolic	n.a.	linear	DGM
$\alpha_{\text{gas}}$	no	hyperbolic	n.a.	linear	DGM
$\underline{u}$	no	parabolic	irregular	n.a.	CGM
$p$	no	hyperbolic	n.a.	linear	DGM
$T$	no	parabolic	irregular	n.a.	CGM

One additional term compared to the Allen Cahn problem is the convective part

$$\int_{\Omega} \langle v_T, \rho c_p \underline{u} \cdot \nabla T \rangle dx - \int_{\partial\Omega} \langle v_T, \rho c_p T \underline{u} \cdot \underline{n} \rangle dS. \quad (8.34)$$

The remaining terms of Equation 8.8 do not contain any differential operators. Therefore, the weak formulation can easily be obtained by forming the inner product with the scalar test function  $v_T$  and integrating over  $\Omega$

$$\begin{aligned} \sum_i \int_{\Omega} \langle v_T, L \partial_i \alpha_i \rangle dx + \int_{\Omega} \langle v_T, P \rangle dx + \int_{\Omega} \langle v_T, \sigma \epsilon (T^4 - T_{\text{amb}}^4) \rangle dx \\ + \int_{\Omega} \left\langle v_T, 0.53 p_0 \frac{L_v}{R} \exp \left( \frac{1}{T_v} - \frac{1}{T} \right) \right\rangle dx. \end{aligned} \quad (8.35)$$

## 8.7 Findings

With all independent variables of the model addressed in the previous section, the chosen combination of numerical schemes is summarised in Table 8.1. This mixed discretization in combination with the outlined weak formulation for this problem can be implemented using any modern Finite Element code. In the first two examples, it has been shown that both creating a custom implementation using modern programming languages and implementing a model using popular Finite Element libraries are viable options to create a corresponding simulation.

In this case, a coupled, nonlinear problem is obtained. This means that instead of solving a discrete linear system of equations, one must solve a global, nonlinear root finding problem of the form  $F(u; v) = 0$  in every time step by appropriate means, such as Picard iteration or Newton's method. Such functionality however is readily implemented in Finite Element libraries such as Firedrake (Ham et al., 2023), which has been used in the previous section.

In summary, the developed method given in chapter 6 has been applied to the model problem of powder scale melting and solidification during PBF-LB/M. With the inputs of computational hardware, grid as well as a mathematical description of the problem, one can systematically and unambiguously derive a mixed discretization for the problem that differs from the most prevalent scheme in the literature, the FVM. For the given problem and given a fixed grid size, this model can reflect the qualitative behavior of each quantity of interest in a more accurate manner than the FVM. For the transport equations representing the phases, it has been shown in section 7.2 that the DGM delivers superior accuracy compared to the FVM. Modeling of diffusive quantities has been discussed in section 7.1 where it has been shown that the CGM does represent the continuous nature of the problem accurately. Although section 7.1 covers the Allen-Cahn equation, which is not present in the mesoscale PBF-LB/M model, its classification according to sections 6.2 and 6.4 is identical with the energy balance equation 8.8. That is, both equations are semilinear, parabolic equations with a scalar quantity of interest.

Due to the restrictions posed to the present method in section 6.1, this compound discretization will also yield a stable solution. This property can, e.g., not be guaranteed by applying a pure CGM or FDM to this problem as special care would have to be taken with upwinding for convective parts of the equations - for instance by using asymmetric stencils or introducing additional, artificial diffusion for sake of dampening oscillations (Langtangen and Linge, 2017). Using a pure FVM with unconditionally stable formulations, on the other hand, would yield a stable solution. However, this comes at the cost of accuracy in the form of pronounced artificial, numerical diffusion for convective quantities and low accuracy due to its discontinuous nature for diffusive quantities (Versteeg and Malalasekera, 2007).

Furthermore, the present model allows for additional flexibility regarding the desired resolution of each field. Although the computational grid is the same for each variable, the order of approximation can be set arbitrarily on a per-field basis. As a

result, large spatial gradients in, for example, the temperature field can be efficiently resolved using higher-order Finite Elements whereas slowly changing fields such as the phase fractions can be approximated with lower-order elements. Thus, the creation of computational bottlenecks can be effectively avoided in this way. This strategy in this form cannot be pursued using the FVM since it is formally limited to constant polynomial order in its approximation. The details of this argument have been elaborated in section 7.2. Furthermore, the use of a continuous FEM on the parabolic parts of the system, i.e. fluid velocity and temperature, will yield a lower overall error with fixed grid size since the real solution can be assumed to be continuous.

The proposed claim of having found an efficient spatial discretization is supported by the findings of several independent groups in recent years. In particular, similar models for resolving the melt pool dynamics in PBF-LB/M have been developed using mixed formulations of the FEM. For example, Meier et al. developed a custom, high-performance Finite Element simulation code called MeltPoolDG based on the DGM as well as a mixed FEM to resolve the melt pool on the powder bed scale (Meier et al., 2021; Kronbichler et al., 2018). Other recent developments include a Finite Element model on multiple grids to capture the governing physics (Caboussat et al., 2023), and a space-time FEM code that is based on a Petrov Galerkin approximation (Kopp et al., 2022). Furthermore, there also exist implementations of such melt pool models using commercially available software. For instance, the popular COMSOL software package which implements many variants of the FEM can resolve the relevant thermo fluid dynamics during laser melting (Liu et al., 2020; Mayi et al., 2021).

However, none of these works outline the systematic derivation of the weak formulation for the mesoscale PBF-LB/M governing equations as was done in this chapter. This underlines the overall contribution towards making such high-performance methods more accessible to application experts. Therefore, this work contributes to bridging the gap between well-established, widely used schemes and state-of-the-art, high-performance methods for PBF-LB/M that was outlined in Chapter 3.

It can thus be concluded that hypothesis H3 may be considered valid in the present work.

# Chapter 9

## Conclusions and Future Work

This work addresses an important and emerging research gap in the research community. Due to recent developments in high-performance numerical schemes to approximate PDEs, there is an increasing gap between well-established, widely used schemes and state-of-the-art, high-performance methods. This trend has been outlined in particular for the simulation problem of predicting the mesoscale physics during PBF-LB/M. Consequently, the question arises whether the abovementioned gap may be closed or narrowed by establishing a method for guiding an application expert through selecting suitable numerical schemes.

To answer this question, the research conducted is split into three main research hypotheses to systematically approach this topic. The key findings of this work can be summarised as follows. The Bubnov DGM can serve as a general mathematical baseline for the FVM, FDM and CGM. These form the majority of schemes used in practice, which is why the scope of this work is restricted to this specific subset. It has been shown that one can recover all these schemes by introducing certain restrictions to the DGM, which can either be on the algebraic or algorithmic level.

Furthermore, by using those simplified schemes where appropriate, one is to gain stability and performance in numerical simulations that scale with problem size and hardware. In particular, one may even avoid the assembly of a global linear system by leveraging domain restrictions in the case of the FDM. If a PDE exhibits strong nonlinearity, choosing a purely reconstruction-based approach via the FVM is then beneficial and assembly of the weak form can be avoided as well.

This common framework based on the DGM enables interoperability of the men-

tioned schemes. As such, one can combine them when solving multiphysics problems and thus assemble efficiently mixed discretizations. Therefore, manual coupling of different numerical solvers is not needed in theory. Instead, one may implement them in a monolithic way and thus avoid costly memory transfer operations, concluding the first research hypothesis H1.

One can systematically derive an efficient combination of numerical schemes, given only a few inputs about the problem, that is, hardware requirements, the mathematical formulation of the problem and the domain geometry. The method to identify these schemes is based on the field and delivers one-to-one recommendations on which method to use. The selected methods, regarding the spatial discretization of the problem, are shown to deliver stable and computationally efficient approximations.

As such, applying the developed method to the two model problems in this work, the Allen Cahn equation in 1D and 2D and the advection equation in 2D yields methods that notably outperform their respective alternatives. In the former case, this difference in computation time between FDM and FEM has been shown to exceed a factor of 40 times. In the latter case, it has been shown that as predicted by the theory, the differences in computational efficiency between DGM and FVM grow considerably with more powerful computational hardware and the necessary degrees of freedom for some fixed accuracy. The presented good parallelization capabilities of the DGM are essential to obtaining solutions in viable time for complex problems, where other numerical schemes would pose a severe bottleneck due to their algorithmic nature as outlined in section 7.2. This unified method forms the basis for verification of the second hypothesis H2.

It has also been shown that for a real multiphysics problem, even for coupled systems of PDEs, one can derive an efficient combination of schemes in a reproducible and for application experts accessible way. This highlights the practicality and usefulness of the established framework for end users. Especially researchers who are familiar with physical modeling, but not to the same degree with state-of-the-art high-performance numerical methods are expected to profit from this approach. The exemplary use of the developed method through the chosen example problem addresses the final hypothesis H3.

Some future work might be attributed to further breaking down the choices of numerical schemes based on a more granular problem classification. The classes of PDE

introduced in this work are rather coarse, albeit the most widely used.

One may also include mixed Finite Element discretizations into the method. These include the use of Finite Elements that use particular function spaces to exactly match the map of some differential operators. Including those will, however, make the selection process ambiguous and thus a key feature of this work would be sacrificed, i.e. simplicity and reproducibility would be traded for added flexibility and possibly efficiency. Thus, within the scope of this work, only scalar- and vector-valued Lagrange elements were included which would then have to be extended considerably. By further refining the established decision metric between numerical schemes, i.e. including a particular choice of function space, it is likely possible to account for mixed Finite Element methods within the established work.

Due to the clear recommendation and insight that this method can provide on a per-problem basis, it can be expected to have a substantial impact within various scientific communities that apply simulation techniques. For instance, the sharp phase field method which can describe frictionless interface motion is thus far only well researched using spectral and FDMs (Finel et al., 2018; Fleck and Schleifer, 2023). These, however, limit the applicability to relatively simple geometries as outlined above. It is thus desirable to obtain a formulation that allows for more flexibility, such as the FEM. In principle, the sharp phase field method is not straightforward to generalize, since the governing equations have to be modified on the discrete level (Fleck and Schleifer, 2023). The results outlined in section 5.3 and 7.1 may, however, provide a natural guideline to derive such a generalization using Finite Elements from the Finite Difference formulation given in (Fleck and Schleifer, 2023) by computing a reference stencil and applying the Finite Element specific assembly process afterward.

Thus, this work presents a possible way of establishing recent, high-performance FEMs in more application-driven research communities. This can hopefully form a contribution towards increasing the impact of numerical simulation in research and industry by shortening computational load and thus increasing development throughput.



# References

- Michael Abbott, Dilum Aluthge, N3N5, Simeon Schaub, Chris Elrod, Carlo Lucibello, and Johnny Chen. Mcabbott/Tullio.jl: V0.3.5. Zenodo, September 2022. URL <https://zenodo.org/record/7106192>.
- Ranadip Acharya, John A. Sharon, and Alexander Staroselsky. Prediction of microstructure in laser powder bed fusion process. *Acta Materialia*, 124:360–371, January 2017. ISSN 1359-6454. doi: 10.1016/j.actamat.2016.11.018.
- Mohamadreza Afrasiabi, Christof Lüthi, Markus Bambach, and Konrad Wegener. Multi-Resolution SPH Simulation of a Laser Powder Bed Fusion Additive Manufacturing Process. *Applied Sciences*, 11(7):2962, January 2021. doi: 10.3390/app11072962. URL <https://www.mdpi.com/2076-3417/11/7/2962>.
- Akash Aggarwal, Arvind Chouhan, Sushil Patel, D. K. Yadav, Arvind Kumar, A. R. Vinod, K. G. Prashanth, and N. P. Gurao. Role of impinging powder particles on melt pool hydrodynamics, thermal behaviour and microstructure in laser-assisted DED process: A particle-scale DEM – CFD – CA approach. *International Journal of Heat and Mass Transfer*, 158:119989, January 2020. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2020.119989. URL <http://www.sciencedirect.com/science/article/pii/S0017931019367997>.
- Faiyaz Ahsan, Jafar Razmi, and Leila Ladani. Global local modeling of melt pool dynamics and bead formation in laser bed powder fusion additive manufacturing using a multi-physics thermo-fluid simulation. *Progress in Additive Manufacturing*, 7(6): 1275–1285, December 2022. ISSN 2363-9520. doi: 10.1007/s40964-022-00302-w. URL <https://doi.org/10.1007/s40964-022-00302-w>.
- Vasiliki E. Alexopoulou, Emmanouil L. Papazoglou, Panagiotis Karmiris-Obratański, and Angelos P. Markopoulos. 3D finite element modeling of selective laser melting for conduction, transition and keyhole modes. *Journal of Manufacturing Processes*, 75:877–894, March 2022. ISSN 1526-6125. doi: 10.1016/j.jmapro.2022.01.054. URL <https://www.sciencedirect.com/science/article/pii/S1526612522000718>.
- Jeffrey Allen. Mesoscale modeling of metal-based selective laser melting: Evolution of the vapor capillary. *Engineering Research Express*, February 2021. ISSN 2631-8695. doi: 10.1088/2631-8695/abe9fc. URL <https://iopscience.iop.org/article/10.1088/2631-8695/abe9fc>.

- M. S. Alnaes, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40, 2014. doi: 10.1145/2566630.
- W E Alphonso, M Bayat, M Baier, S Carmignato, and J H Hattel. MULTI-PHYSICS NUMERICAL MODELLING OF 316L AUSTENITIC STAINLESS STEEL IN LASER POWDER BED FUSION PROCESS AT MESO-SCALE. page 6.
- N. Aminnia, A. Estupinan Donoso, and B. Peters. Developing a DEM-Coupled Open-FOAM solver for multiphysics simulation of additive manufacturing process. In *8th European Congress on Computational Methods in Applied Sciences and Engineering*. CIMNE, 2022a. doi: 10.23967/eccomas.2022.084. URL [https://www.scipedia.com/public/Aminnia\\_et\\_al\\_2022a](https://www.scipedia.com/public/Aminnia_et_al_2022a).
- Navid Aminnia, Alvaro Antonio Estupinan Donoso, and Bernhard Peters. CFD-DEM simulation of melt pool formation and evolution in powder bed fusion process. May 2022b. URL <https://orbilu.uni.lu/handle/10993/54513>.
- Larbi Arbaoui, Pierre Schrooyen, and Nicolas Poletz. Hydrodynamic model based on extended Discontinuous Galerkin method for powder-bed fusion numerical simulation.
- J. H. Argyris and D. W. Scharpf. Finite Elements in Time and Space. *The Aeronautical Journal*, 73(708):1041–1044, December 1969. ISSN 0001-9240, 2059-6464. doi: 10.1017/S0001924000051198. URL <https://www.cambridge.org/core/journals/aeronautical-journal/article/abs/finite-elements-in-time-and-space/58F4D00F45FC179682ECAA4F7A81F823>.
- Daniel Arndt, Niklas Fehn, Guido Kanschat, Katharina Kormann, Martin Kronbichler, Peter Munch, Wolfgang A. Wall, and Julius Witte. ExaDG: High-Order Discontinuous Galerkin for the Exa-Scale. In Hans-Joachim Bungartz, Severin Reiz, Benjamin Uekermann, Philipp Neumann, and Wolfgang E. Nagel, editors, *Software for Exascale Computing - SPPEXA 2016-2019*, volume 136, pages 189–224. Springer International Publishing, Cham, 2020. ISBN 978-3-030-47955-8 978-3-030-47956-5. doi: 10.1007/978-3-030-47956-5\_8. URL [http://link.springer.com/10.1007/978-3-030-47956-5\\_8](http://link.springer.com/10.1007/978-3-030-47956-5_8).
- Douglas N. Arnold and Anders Logg. Periodic table of the finite elements. *SIAM News*, 47, 2014.
- Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, January 2002. ISSN 0036-1429, 1095-7170. doi: 10.1137/S0036142901384162. URL <http://epubs.siam.org/doi/10.1137/S0036142901384162>.

- Santiago Badia, Alberto F. Martín, Eric Neiva, and Francesc Verdugo. A Generic Finite Element Framework on Parallel Tree-Based Adaptive Meshes. *SIAM Journal on Scientific Computing*, 42(6):C436–C468, January 2020. ISSN 1064-8275, 1095-7197. doi: 10.1137/20M1328786. URL <https://epubs.siam.org/doi/10.1137/20M1328786>.
- Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202, Basel, 1997. Birkhäuser Press.
- Satish Balay, Shrirang Abhyankar, Mark F. Adams, Steven Benson, Jed Brown, Peter Brune, Kris Buschelman, Emil Constantinescu, Lisandro Dalcin, Alp Dener, Victor Eijkhout, Jacob Faibussowitsch, William D. Gropp, V'aclav Hapla, Tobin Isaac, Pierre Jolivet, Dmitry Karpeev, Dinesh Kaushik, Matthew G. Knepley, Fande Kong, Scott Kruger, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Lawrence Mitchell, Todd Munson, Jose E. Roman, Karl Rupp, Patrick Sanan, Jason Sarich, Barry F. Smith, Stefano Zampini, Hong Zhang, Hong Zhang, and Junchao Zhang. PETSc/TAO users manual. Technical Report ANL-21/39 - Revision 3.19, Argonne National Laboratory, 2023.
- W. Bangerth and O. Kayser-Herold. Data structures and requirements for *hp* finite element software. *ACM Transactions on Mathematical Software*, 36(1):1–31, March 2009. ISSN 0098-3500, 1557-7295. doi: 10.1145/1486525.1486529. URL <https://dl.acm.org/doi/10.1145/1486525.1486529>.
- Jacques Baranger, Jean-François Maitre, and Fabienne Oudin. Connection between finite volume and mixed finite element methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 30(4):445–465, 1996. ISSN 0764-583X, 1290-3841. doi: 10.1051/m2an/1996300404451. URL <http://www.esaim-m2an.org/10.1051/m2an/1996300404451>.
- M. Bayat, V. K. Nadimpalli, S. Mohanty, and J. H. Hattel. Resolving the effects of local convective heat transfer via adjustment of thermo-physical properties in pure heat conduction simulation of Laser Powder Bed Fusion (L-PBF). *IOP Conference Series: Materials Science and Engineering*, 861(1):012006, May 2020. ISSN 1757-899X. doi: 10.1088/1757-899X/861/1/012006. URL <https://dx.doi.org/10.1088/1757-899X/861/1/012006>.
- Mohamad Bayat, Sankhya Mohanty, and Jesper Henri Hattel. A systematic investigation of the effects of process parameters on heat and fluid flow and metallurgical conditions during laser-based powder bed fusion of Ti6Al4V alloy. *International Journal of Heat and Mass Transfer*, 139:213–230, August 2019. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2019.05.017. URL <https://www.sciencedirect.com/science/article/pii/S0017931019300742>.

- Mohamad Bayat, Wen Dong, Jesper Thorborg, Albert C. To, and Jesper H. Hattel. A review of multi-scale and multi-physics simulations of metal additive manufacturing processes with focus on modelling strategies. *Additive Manufacturing*, page 102278, September 2021. ISSN 2214-8604. doi: 10.1016/j.addma.2021.102278. URL <https://www.sciencedirect.com/science/article/pii/S2214860421004383>.
- Lauren L. Beghini, Michael Stender, Daniel Moser, Bradley L. Trembacki, Michael G. Veilleux, and Kurtis R. Ford. A coupled fluid-mechanical workflow to simulate the directed energy deposition additive manufacturing process. *Computational Mechanics*, March 2021. ISSN 0178-7675, 1432-0924. doi: 10.1007/s00466-020-01960-9. URL <http://link.springer.com/10.1007/s00466-020-01960-9>.
- Joel Berry, Aurélien Perron, Jean-Luc Fattebert, John D. Roehling, Bey Vrancken, Tien T. Roehling, Debra L. Rosas, John A. Turner, Saad A. Khairallah, Joseph T. McKeown, and Manyalibo J. Matthews. Toward multiscale simulations of tailored microstructure formation in metal additive manufacturing. *Materials Today*, November 2021. ISSN 1369-7021. doi: 10.1016/j.mattod.2021.09.024. URL <https://www.sciencedirect.com/science/article/pii/S1369702121003370>.
- Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, January 2017. ISSN 0036-1445. doi: 10.1137/141000671. URL <https://epubs.siam.org/doi/abs/10.1137/141000671>.
- Andrei Vasilevich Bitsadze. *Some Classes of Partial Differential Equations*, volume 4. CRC Press, 1988.
- A. Bösch, H. Müller-Krumbhaar, and O. Shochet. Phase-field models for moving boundary problems: Controlling metastability and anisotropy. *Zeitschrift für Physik B*, 97:367–377, 1995. doi: 10.1007/BF01307490. URL <https://doi.org/10.1007/BF01307490>.
- Fani Boukouvala, Yijie Gao, Fernando Muzzio, and Marianthi G. Ierapetritou. Reduced-order discrete element method modeling. *Chemical Engineering Science*, 95:12–26, January 2013. ISSN 0009-2509. doi: 10.1016/j.ces.2013.01.053. URL <http://www.sciencedirect.com/science/article/pii/S0009250913000754>.
- J.U Brackbill, D.B Kothe, and C Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, June 1992. ISSN 0021-9991. doi: 10.1016/0021-9991(92)90240-Y. URL <http://www.sciencedirect.com/science/article/pii/002199919290240Y>.
- Brixle, Nicolas. Modellierung von Transportvorgängen für die pulverbettbasierte Additive Fertigung mittels Maschinellen Lernens. Master’s thesis, University of Augsburg, Faculty of Applied Computer Science, June 2022.

- Martin Brokate, Norbert Henze, Frank Hettlich, Andreas Meister, Gabriela Schranz-Kirlinger, and Thomas Sonar. *Grundwissen Mathematikstudium*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016. ISBN 978-3-642-45077-8 978-3-642-45078-5. doi: 10.1007/978-3-642-45078-5. URL <http://link.springer.com/10.1007/978-3-642-45078-5>.
- Alexander N. Brooks and Thomas J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, September 1982. ISSN 0045-7825. doi: 10.1016/0045-7825(82)90071-8. URL <https://www.sciencedirect.com/science/article/pii/0045782582900718>.
- Tan Bui-Thanh. From Godunov to a unified hybridized discontinuous Galerkin framework for partial differential equations. *Journal of Computational Physics*, 295:114–146, August 2015. ISSN 00219991. doi: 10.1016/j.jcp.2015.04.009. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999115002557>.
- Hans-Joachim Bungartz, Stefan Zimmer, Martin Buchholz, and Dirk Pflüger. *Modellbildung Und Simulation*. eXamen.Press. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-37655-9 978-3-642-37656-6. doi: 10.1007/978-3-642-37656-6. URL <http://link.springer.com/10.1007/978-3-642-37656-6>.
- Christian Burkhardt, Paul Steinmann, and Julia Mergheim. Thermo-mechanical simulations of powder bed fusion processes: Accuracy and efficiency. *Advanced Modeling and Simulation in Engineering Sciences*, 9(1):1–31, December 2022. ISSN 2213-7467. doi: 10.1186/s40323-022-00230-y. URL <https://amses-journal.springeropen.com/articles/10.1186/s40323-022-00230-y>.
- Alexandre Caboussat, Julien Hess, Alexandre Masserey, and Marco Picasso. Numerical simulation of temperature-driven free surface flows, with application to laser melting and polishing. *Journal of Computational Physics: X*, page 100127, June 2023. ISSN 25900552. doi: 10.1016/j.jcpx.2023.100127. URL <https://linkinghub.elsevier.com/retrieve/pii/S2590055223000057>.
- John W. Cahn and John E. Hilliard. Free Energy of a Nonuniform System. I. Interfacial Free Energy. *The Journal of Chemical Physics*, 28(2):258–267, February 1958. ISSN 0021-9606. doi: 10.1063/1.1744102. URL <https://aip.scitation.org/doi/10.1063/1.1744102>.
- Liu Cao. Mesoscopic-Scale Numerical Simulation Including the Influence of Process Parameters on SLM Single-Layer Multi-pass Formation. *Metallurgical and Materials Transactions A*, 51(8):4130–4145, January 2020a. ISSN 1073-5623. doi: 10.1007/s11661-020-05831-z.

- Liu Cao. Mesoscopic-scale simulation of pore evolution during laser powder bed fusion process. *Computational Materials Science*, 179:109686, January 2020b. ISSN 09270256. doi: 10.1016/j.commatsci.2020.109686.
- Liu Cao. Numerical Investigation on Molten Pool Dynamics During Multi-laser Array Powder Bed Fusion Process. *Metallurgical and Materials Transactions A*, November 2020c. ISSN 1543-1940. doi: 10.1007/s11661-020-06076-6. URL <https://doi.org/10.1007/s11661-020-06076-6>.
- Liu Cao. Workpiece-scale numerical simulations of SLM molten pool dynamic behavior of 316L stainless steel. *Computers & Mathematics with Applications*, January 2020d. ISSN 08981221. doi: 10.1016/j.camwa.2020.04.020.
- Liu Cao. Mesoscopic-scale numerical investigation including the influence of scanning strategy on selective laser melting process. *Computational Materials Science*, 189: 110263, March 2021. ISSN 0927-0256. doi: 10.1016/j.commatsci.2020.110263. URL <http://www.sciencedirect.com/science/article/pii/S0927025620307540>.
- Liu Cao and Wei Guan. Simulation and analysis of LPBF multi-layer single-track forming process under different particle size distributions. *The International Journal of Advanced Manufacturing Technology*, April 2021. ISSN 1433-3015. doi: 10.1007/s00170-021-06987-7. URL <https://doi.org/10.1007/s00170-021-06987-7>.
- Yang Cao, Xin Lin, Nan Kang, Liang Ma, Lei Wei, Min Zheng, Jun Yu, Dongjian Peng, and Weidong Huang. A novel high-efficient finite element analysis method of powder bed fusion additive manufacturing. *Additive Manufacturing*, 46:102187, October 2021. ISSN 2214-8604. doi: 10.1016/j.addma.2021.102187. URL <https://www.sciencedirect.com/science/article/pii/S221486042100350X>.
- Philip Cardiff, Zeljko Tukovic, Alojz Ivankovic, and P. Jaeger. *Development of an Arbitrary Lagrangian-Eulerian Finite Volume Method for Metal Forming Simulation in OpenFOAM*. January 2018.
- Dongju Chen, Gang Li, Peng Wang, Zhiqiang Zeng, and Yuhang Tang. Numerical simulation of melt pool size and flow evolution for laser powder bed fusion of powder grade Ti6Al4V. *Finite Elements in Analysis and Design*, 223:103971, October 2023a. ISSN 0168-874X. doi: 10.1016/j.finel.2023.103971. URL <https://www.sciencedirect.com/science/article/pii/S0168874X23000641>.
- Dongju Chen, Gang Li, Peng Wang, Zhiqiang Zeng, and Yuhang Tang. Numerical Simulation of Meltpool Evolution During Laser Powder Bed Fusion (LPBF) Processing of Ti6Al4V., June 2023b.
- Hui Chen and Wentao Yan. Spattering and denudation in laser powder bed fusion process: Multiphase flow modelling. *Acta Materialia*, January 2020. ISSN 1359-6454. doi: 10.1016/j.actamat.2020.06.033. URL <http://www.sciencedirect.com/science/article/pii/S1359645420304687>.

- Hui Chen, Yajing Sun, Weihao Yuan, Shengyong Pang, Wentao Yan, and Yusheng Shi. A Review on Discrete Element Method Simulation in Laser Powder Bed Fusion Additive Manufacturing. *Chinese Journal of Mechanical Engineering: Additive Manufacturing Frontiers*, 1(1):100017, March 2022. ISSN 2772-6657. doi: 10.1016/j.cjmeam.2022.100017. URL <https://www.sciencedirect.com/science/article/pii/S2772665722000071>.
- Qian Chen, Yunhao Zhao, Seth Strayer, Yufan Zhao, Kenta Aoyagi, Yuichiro Koizumi, Akihiko Chiba, Wei Xiong, and Albert C. To. Elucidating the effect of preheating temperature on melt pool morphology variation in Inconel 718 laser powder bed fusion via simulation and experiment. *Additive Manufacturing*, 37:101642, January 2021. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101642. URL <https://www.sciencedirect.com/science/article/pii/S2214860420310149>.
- Qiang Chen, Gildas Guillemot, Charles-André Gandin, and Michel Bellet. Three-dimensional finite element thermomechanical modeling of additive manufacturing by selective laser melting for ceramic materials. *Additive Manufacturing*, 16:124–137, August 2017. ISSN 22148604. doi: 10.1016/j.addma.2017.02.005. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860416302317>.
- Songtao Chen and Qinglin Duan. An adaptive second-order element-free Galerkin method for additive manufacturing process. *Computational Materials Science*, 183: 109911, January 2020. ISSN 09270256. doi: 10.1016/j.commatsci.2020.109911. URL <http://www.sciencedirect.com/science/article/pii/S092702562030402X>.
- C. Chevalier and F. Pellegrini. PT-SCOTCH: A tool for efficient parallel graph ordering. *Parallel Computing*, 34(6):318–331, 2008. doi: 10.1016/j.parco.2007.12.001.
- Cheng-Yen Chien, Trong-Nhan Le, Ze-Hong Lin, and Yu-Lung Lo. Numerical and experimental investigation into gas flow field and spattering phenomena in laser powder bed fusion processing of Inconel 718. *Materials & Design*, 210:110107, November 2021. ISSN 0264-1275. doi: 10.1016/j.matdes.2021.110107. URL <https://www.sciencedirect.com/science/article/pii/S0264127521006626>.
- Arvind Chouhan, Akash Aggarwal, and Arvind Kumar. Model Development in Open-FOAM for Laser Metal Deposition-based Additive Manufacturing Process. *Transactions of the Indian Institute of Metals*, 71(11):2833–2838, January 2018. ISSN 0972-2815. doi: 10.1007/s12666-018-1440-7.
- Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Number 40 in Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002. ISBN 978-0-89871-514-9.
- Bernardo Cockburn and Guosheng Fu. A systematic construction of finite element commuting exact sequences. *SIAM journal on numerical analysis*, 55:1650–1688, 2017. doi: 10.1137/16M1073352.

- Bernardo Cockburn and Chi-Wang Shu. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, December 1998. ISSN 0036-1429, 1095-7170. doi: 10.1137/S0036142997316712. URL <http://epubs.siam.org/doi/10.1137/S0036142997316712>.
- Bernardo Cockburn and Chi-Wang Shu. Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. *Journal of Scientific Computing*, 16(3): 173–261, September 2001. ISSN 1573-7691. doi: 10.1023/A:1012873910884. URL <https://doi.org/10.1023/A:1012873910884>.
- Bernardo Cockburn, Fengyan Li, and Chi-Wang Shu. Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *Journal of Computational Physics*, 194(2):588–610, March 2004. ISSN 0021-9991. doi: 10.1016/j.jcp.2003.09.007. URL <https://www.sciencedirect.com/science/article/pii/S0021999103004960>.
- Andrea Colagrossi and Maurizio Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2):448–475, November 2003. ISSN 0021-9991. doi: 10.1016/S0021-9991(03)00324-3. URL <https://www.sciencedirect.com/science/article/pii/S0021999103003243>.
- Lisandro D. Dalcin, Rodrigo R. Paz, Pablo A. Kler, and Alejandro Cosimo. Parallel distributed computing using Python. *Advances in Water Resources*, 34(9):1124–1139, September 2011. ISSN 03091708. doi: 10.1016/j.advwatres.2011.04.013. URL <https://linkinghub.elsevier.com/retrieve/pii/S0309170811000777>.
- My Ha Dao and Jing Lou. Simulations of Laser Assisted Additive Manufacturing by Smoothed Particle Hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 373:113491, January 2021. ISSN 0045-7825. doi: 10.1016/j.cma.2020.113491.
- David K. Kafui, Colin Thornton, and Jonathan P.K. Fully-3D DEM fluidization simulation of a shallow fine powder bed, June 2023. URL <https://aiche.confex.com/aiche/s06/techprogram/P35666.HTM>.
- T. DebRoy, H. L. Wei, J. S. Zuback, T. Mukherjee, J. W. Elmer, J. O. Milewski, A. M. Beese, A. Wilson-Heid, A. De, and W. Zhang. Additive manufacturing of metallic components – Process, structure and properties. *Progress in Materials Science*, 92: 112–224, March 2018. ISSN 0079-6425. doi: 10.1016/j.pmatsci.2017.10.001. URL <https://www.sciencedirect.com/science/article/pii/S0079642517301172>.
- Erik R. Denlinger, Michael Gouge, Jeff Irwin, and Pan Michaleris. Thermomechanical model development and in situ experimental validation of the Laser Powder-Bed Fusion process. *Additive Manufacturing*, 16:73–80, January 2017. ISSN 2214-8604. doi: 10.1016/j.addma.2017.05.001. URL <http://www.sciencedirect.com/science/article/pii/S2214860417300441>.

- Bastien Dietemann, Tim Najuch, Alexander Wessel, Alexander Butz, and Claas Bierwisch. Simulation of the laser powder bed melting process from powder application to mechanical component properties. 2022. URL <https://publica.fraunhofer.de/handle/publica/434030>.
- Michael Dumbser, Dinshaw S. Balsara, Eleuterio F. Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227 (18):8209–8253, September 2008. ISSN 00219991. doi: 10.1016/j.jcp.2008.05.025. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999108002829>.
- J. Eiken. Numerical solution of the phase-field equation with minimized discretization error. *IOP Conf. Ser.: Mater. Sci. Eng.*, 33:012105, 2012. doi: doi:10.1088/1757-899X/33/1/012105. URL <https://doi.org/10.1088/1757-899x/33/1/012105>.
- S. M. Elahi, R. Tavakoli, A. K. Boukellal, T. Isensee, I. Romero, and D. Tourret. Multiscale simulation of powder-bed fusion processing of metallic alloys. *arXiv:2203.10370 [cond-mat]*, March 2022. URL <http://arxiv.org/abs/2203.10370>.
- Emergen Research. Simulation Software Market, By Component (Software, Services), By Deployment (On-cloud, On-premise), By Application (Design & Consulting, Support & Maintenance, E-Learning, Training, and R&D), By End-use, and By Region Forecast to 2028. Technical Report ER\_00921, January 2022. URL <https://www.emergenresearch.com/industry-report/simulation-software-market>.
- Yekaterina Epshteyn and Béatrice Rivière. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *Journal of Computational and Applied Mathematics*, 206(2):843–872, September 2007. ISSN 0377-0427. doi: 10.1016/j.cam.2006.08.029. URL <https://www.sciencedirect.com/science/article/pii/S0377042706005279>.
- Alexandre Ern and Jean-Luc Guermond. *Theory and Practice of Finite Elements*. Applied Mathematical Sciences. Springer-Verlag, New York, 2004. ISBN 978-0-387-20574-8. doi: 10.1007/978-1-4757-4355-5. URL <https://www.springer.com/gp/book/9780387205748>.
- Alvaro Antonio Estupinan Donoso and Bernhard Peters. Exploring a Multiphysics Resolution Approach for Additive Manufacturing. *JOM*, 70(8):1604–1610, January 2018. ISSN 1047-4838. doi: 10.1007/s11837-018-2964-3.
- Lawrence C. Evans. *Partial Differential Equations*. Number v. 19 in Graduate Studies in Mathematics. American Mathematical Society, Providence, R.I, 2nd ed edition, 2010. ISBN 978-0-8218-4974-3.
- Robert Eymard, Thierry Gallouët, and Raphaële Herbin. Finite volume methods. In *Handbook of Numerical Analysis*, volume 7 of *Solution of Equation in  $\mathbb{R}$  (Part 3)*, *Techniques of Scientific Computing (Part 3)*, pages 713–1018. Elsevier, January

2000. doi: 10.1016/S1570-8659(00)07005-8. URL <https://www.sciencedirect.com/science/article/pii/S1570865900070058>.
- Chen Fan and Wentao Yan. High-fidelity Modelling of thermal stress for additive manufacturing by linking thermal-fluid and mechanical models. *Materials & Design*, page 109185, January 2020. ISSN 0264-1275. doi: 10.1016/j.matdes.2020.109185.
- Zongyue Fan, Hao Wang, Zhida Huang, Huming Liao, Jiang Fan, Jian Lu, Chong Liu, and Bo Li. A Lagrangian Meshfree Mesoscale Simulation of Powder Bed Fusion Additive Manufacturing of Metals. *International Journal for Numerical Methods in Engineering*, January 2020. ISSN 0029-5981. doi: 10.1002/nme.6546.
- Xiaobing Feng and Andreas Prohl. Error analysis of a mixed finite element method for the Cahn-Hilliard equation. *Numerische Mathematik*, 99(1):47–84, November 2004. ISSN 0029-599X, 0945-3245. doi: 10.1007/s00211-004-0546-5. URL <http://link.springer.com/10.1007/s00211-004-0546-5>.
- Alphonse Finel, Yann Le Bouar, Benoît Dabas, Benoît Appolaire, Yasunori Yamada, and Tetsuo Mohri. Sharp Phase Field Method. *Physical Review Letters*, 121(2): 025501, July 2018. doi: 10.1103/PhysRevLett.121.025501. URL <https://link.aps.org/doi/10.1103/PhysRevLett.121.025501>.
- M. Fleck, D. Pilipenko, R. Spatschek, and E. A. Brener. Brittle fracture in viscoelastic materials as a pattern-formation process. *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 83:046213, 2011. doi: 10.1103/PhysRevE.83.046213. URL <http://link.aps.org/doi/10.1103/PhysRevE.83.046213>.
- M. Fleck, F. Schleifer, M. Holzinger, and U. Glatzel. Improving the numerical solution of the phase-field equation by the systematic integration of analytic properties of the phase-field profile function. In T. Gleim and S. Lange, editors, *8th GACM Colloquium*, pages 445–450. Kassel University Press, 2019. doi: 10.17170/kobra-202007161457. URL <https://dx.doi.org/doi:10.17170/kobra-202007161457>.
- Michael Fleck and Felix Schleifer. Sharp phase-field modeling of isotropic solidification with a super efficient spatial resolution. *Engineering with Computers*, 39(3): 1699–1709, June 2023. ISSN 1435-5663. doi: 10.1007/s00366-022-01729-z. URL <https://doi.org/10.1007/s00366-022-01729-z>.
- Michael Fleck, Holger Federmann, and Evgeny Pogorelov. Phase-field modeling of Li-insertion kinetics in single LiFePO<sub>4</sub>-nano-particles for rechargeable Li-ion battery application. *Computational Materials Science*, 153:288–296, October 2018. ISSN 0927-0256. doi: 10.1016/j.commatsci.2018.06.049. URL <https://www.sciencedirect.com/science/article/pii/S0927025618304282>.
- Michael Fleck, Felix Schleifer, and Patrick Zimbrod. Frictionless Motion of Diffuse Interfaces by Sharp Phase-Field Modeling. *Crystals*, 12(10):1496, October 2022.

- ISSN 2073-4352. doi: 10.3390/cryst12101496. URL <https://www.mdpi.com/2073-4352/12/10/1496>.
- Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, 1988. ISSN 0025-5718, 1088-6842. doi: 10.1090/S0025-5718-1988-0935077-0. URL <https://www.ams.org/mcom/1988-51-184/S0025-5718-1988-0935077-0/>.
- Jouni Freund and Rolf Stenberg. On weakly imposed boundary conditions for second order problems. In *Proceedings of the Ninth Int. Conf. Finite Elements in Fluids*, pages 327–336. Venice, 1995.
- Thomas-Peter Fries and Hermann G Matthies. A Review of Petrov-Galerkin Stabilization Approaches and an Extension to Meshfree Methods. Informatikbericht 2004-01, Institut of Scientific Computing, Braunschweig, Germany, 2004.
- O.B. Fringer, M. Gerritsen, and R.L. Street. An unstructured-grid, finite-volume, non-hydrostatic, parallel coastal ocean simulator. *Ocean Modelling*, 14(3-4):139–173, January 2006. ISSN 14635003. doi: 10.1016/j.ocemod.2006.03.006. URL <https://linkinghub.elsevier.com/retrieve/pii/S1463500306000394>.
- Jan-Philipp Fürstenau, Henning Wessels, Christian Weißenfels, and Peter Wriggers. Generating virtual process maps of SLM using powder-scale SPH simulations. *Computational Particle Mechanics*, January 2019. ISSN 2196-4378. doi: 10.1007/s40571-019-00296-3.
- Jan-Philipp Fürstenau, Christian Weißenfels, and Peter Wriggers. *Incompressible Simulation of the Selective Laser Melting Process*. June 2021.
- Elena Gaburro. A Unified Framework for the Solution of Hyperbolic PDE Systems Using High Order Direct Arbitrary-Lagrangian–Eulerian Schemes on Moving Unstructured Meshes with Topology Change. *Archives of Computational Methods in Engineering*, 28(3):1249–1321, May 2021. ISSN 1886-1784. doi: 10.1007/s11831-020-09411-7. URL <https://doi.org/10.1007/s11831-020-09411-7>.
- Wenjun Ge, Jerry Y. H. Fuh, and Suck Joo Na. Numerical modelling of keyhole formation in selective laser melting of Ti6Al4V. *Journal of Manufacturing Processes*, 62:646–654, February 2021. ISSN 1526-6125. doi: 10.1016/j.jmapro.2021.01.005. URL <http://www.sciencedirect.com/science/article/pii/S1526612521000232>.
- S. Geer, M. L. Bernhardt-Barry, E. J. Garboczi, J. Whiting, and A. Donmez. A more efficient method for calibrating discrete element method parameters for simulations of metallic powder used in additive manufacturing. *Granular Matter*, 20(4):36, January 2018. ISSN 1434-5021. doi: 10.1007/s10035-018-0848-4.
- P. Gh Ghanbari, E. Mazza, and E. Hosseini. Adaptive local-global multiscale approach for thermal simulation of the selective laser melting process. *Additive Manufacturing*, page 101518, January 2020. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101518.

- Amanda Giam, Fan Chen, Jiaxiang Cai, and Wentao Yan. Factorial design analytics on effects of material parameter uncertainties in multiphysics modeling of additive manufacturing. *npj Computational Materials*, 9(1):1–18, April 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01004-9. URL <https://www.nature.com/articles/s41524-023-01004-9>.
- K. Glasner. Nonlinear preconditioning for diffuse interfaces. 174:695–711, 2001. doi: 10.1006/jcph.2001.6933. URL <https://doi.org/10.1006/jcph.2001.6933>.
- Grand View Research. Virtual Prototype Market Size, Share & Trends Analysis Report By Tool (CAD, CAE, CAM, CFD, FEA), By Deployment (Cloud-based, On Premise), By Vertical, By Region, and Segment Forecasts, 2018 - 2025. Technical Report GVR-2-68038-482-6, Grand View Research, 2017. URL <https://www.grandviewresearch.com/industry-analysis/virtual-prototype-market>.
- C. Gräser, R. Kornhuber, and U. Sack. Time discretizations of anisotropic Allen–Cahn equations. *Ima Journal of Numerical Analysis*, 33(4):1226–1244, 2013. doi: 10.1093/imanum/drs043. URL <https://doi.org/10.1093/imanum/drs043>.
- Heng Gu, Chao Wei, Lin Li, Quanquan Han, Rossitza Setchi, Michael Ryan, and Qian Li. Multi-physics modelling of molten pool development and track formation in multi-track, multi-layer and multi-material selective laser melting. *International Journal of Heat and Mass Transfer*, 151:119458, January 2020. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2020.119458. URL <http://www.sciencedirect.com/science/article/pii/S0017931019358302>.
- Heng Gu, Chao Wei, Lin Li, Michael Ryan, Rossitza Setchi, Quanquan Han, and Lili Qian. Numerical and experimental study of molten pool behaviour and defect formation in multi-material and functionally graded materials laser powder bed fusion. *Advanced Powder Technology*, October 2021. ISSN 0921-8831. doi: 10.1016/j.appt.2021.09.036. URL <https://www.sciencedirect.com/science/article/pii/S0921883121004635>.
- F.-J. Gürtler, M. Karg, K.-H. Leitz, and M. Schmidt. Simulation of Laser Beam Melting of Steel Powders using the Three-Dimensional Volume of Fluid Method. *Physics Procedia*, 41:881–886, January 2013. ISSN 18753892. doi: 10.1016/j.phpro.2013.03.162.
- Franz-Josef Gürtler, Michael Karg, Michael Dobler, Stefanie Kohl, Ilya Tzivilsky, and Michael Schmidt. Influence of powder distribution on process stability in laser beam melting: Analysis of melt pool dynamics by numerical simulations. *SFF Symposium 2014*, 25, January 2014.
- A. V. Gusarov, I. Yadroitsev, Ph. Bertrand, and I. Smurov. Model of Radiation and Heat Transfer in Laser-Powder Interaction Zone at Selective Laser Melting. *Journal of Heat Transfer*, 131(7), January 2009. ISSN 0022-1481. doi: 10.1115/1.3109245.

- David A. Ham, Paul H. J. Kelly, Lawrence Mitchell, Colin J. Cotter, Robert C. Kirby, Koki Sagiya, Nacime Bouziani, Sophia Vorderwuelbecke, Thomas J. Gregory, Jack Betteridge, Daniel R. Shapero, Reuben W. Nixon-Hill, Connor J. Ward, Patrick E. Farrell, Pablo D. Brubeck, India Marsden, Thomas H. Gibson, Miklós Homolya, Tianjiao Sun, Andrew T. T. McRae, Fabio Luporini, Alastair Gregory, Michael Lange, Simon W. Funke, Florian Rathgeber, Gheorghe-Teodor Bercea, and Graham R. Markall. *Firedrake User Manual*, first edition edition, May 2023.
- V. S. Hariharan, S. Pramod, D. Kesavan, B. S. Murty, and Gandham Phanikumar. ICME framework to simulate microstructure evolution during laser powder bed fusion of Haynes 282 nickel-based superalloy. *Journal of Materials Science*, April 2022. ISSN 1573-4803. doi: 10.1007/s10853-022-07170-3. URL <https://doi.org/10.1007/s10853-022-07170-3>.
- V. S. Hariharan, Baler Nithin, L. Ruban Raj, Surendra Kumar Makineni, B.S. Murty, and Gandham Phanikumar. Modelling Microsegregation during Metal Additive Manufacturing: Impact of Dendrite Tip Kinetics and Finite Solute Diffusion. Preprint, Engineering, April 2023. URL <https://www.preprints.org/manuscript/202304.0005/v1>.
- Michiel Hazewinkel. *Encyclopaedia of Mathematics*. Springer-Verlag, Berlin, 2002. ISBN 978-1-4020-0609-8.
- Qiyang He, Huanxiong Xia, Jianhua Liu, Xiaohui Ao, and Shengxiang Lin. Modeling and numerical studies of selective laser melting: Multiphase flow, solidification and heat transfer. *Materials & Design*, page 109115, January 2020a. ISSN 0264-1275. doi: 10.1016/j.matdes.2020.109115.
- Yi He, Jabbar Gardy, Ali Hassanpour, and Andrew E. Bayly. A digital-based approach for characterising spread powder layer in additive manufacturing. *Materials & Design*, page 109102, January 2020b. ISSN 0264-1275. doi: 10.1016/j.matdes.2020.109102.
- Yi He, Ali Hassanpour, and Andrew E. Bayly. Linking particle properties to layer characteristics: Discrete element modelling of cohesive fine powder spreading in additive manufacturing. *Additive Manufacturing*, page 101685, January 2020c. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101685.
- Mark Hedreen. *Simulation of Additive Manufacturing Process Physics and Properties in Powder Bed Electron-Beam Melting of Ti-6Al-4V*. Thesis, 2020. URL <https://digital.lib.washington.edu:443/researchworks/handle/1773/45526>.
- Bruce Hendrickson and Robert Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing '95: Proceedings of the 1995 ACM/IEEE Conference on Supercomputing (CDROM)*, page 28, New York, 1995. ACM Press. ISBN 0-89791-816-9. doi: <https://doi.acm.org/10.1145/224170.224228>.

- Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*, volume 54 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 2008. ISBN 978-0-387-72065-4 978-0-387-72067-8. doi: 10.1007/978-0-387-72067-8. URL <http://link.springer.com/10.1007/978-0-387-72067-8>.
- Charles Hirsch. *Numerical Computation of Internal and External Flows*. Elsevier, 2007. ISBN 978-0-7506-6594-0. doi: 10.1016/B978-0-7506-6594-0.X5037-1. URL <https://linkinghub.elsevier.com/retrieve/pii/B9780750665940X50371>.
- S. Mohammad H. Hojjatzadeh, Niranjana D. Parab, Wentao Yan, Qilin Guo, Lianghua Xiong, Cang Zhao, Minglei Qu, Luis I. Escano, Xianghui Xiao, Kamel Fezzaa, Wes Everhart, Tao Sun, and Lianyi Chen. Pore elimination mechanisms during 3D printing of metals. *Nature Communications*, 10(1):3088, January 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-10973-9. URL <https://doi.org/10.1038/s41467-019-10973-9>.
- Mikl'os Homolya and David A. Ham. A parallel edge orientation algorithm for quadrilateral meshes. *SIAM Journal on Scientific Computing*, 38(5):S48–S61, 2016. doi: 10.1137/15M1021325. URL <http://arxiv.org/abs/1505.03357>.
- Kyung-Min Hong, Corbin M. Grohol, and Yung C. Shin. Comparative Assessment of Physics-Based Computational Models on the NIST Benchmark Study of Molten Pool Dimensions and Microstructure for Selective Laser Melting of Inconel 625. *Integrating Materials and Manufacturing Innovation*, February 2021. ISSN 2193-9772. doi: 10.1007/s40192-021-00201-y. URL <https://doi.org/10.1007/s40192-021-00201-y>.
- Hamed Hosseinzadeh, Mohammadreza Nematollahi, Keyvan Safaei, Hossein Abedi, Parisa Bayati, Reza Javan, Behrang Poorganji, Lang Yuan, and Mohammad Elahinia. A Numerical Approach to Model Microstructure Evolution for NiTi Shape Memory Alloy in Laser Powder Bed Fusion Additive Manufacturing. *Integrating Materials and Manufacturing Innovation*, February 2022. ISSN 2193-9772. doi: 10.1007/s40192-022-00252-9. URL <https://doi.org/10.1007/s40192-022-00252-9>.
- X.Y. Hu and N.A. Adams. A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics*, 213(2):844–861, April 2006. ISSN 00219991. doi: 10.1016/j.jcp.2005.09.001. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999105004195>.
- Thomas J. R. Hughes, Arif Masud, and Jing Wan. A stabilized mixed discontinuous Galerkin method for Darcy flow. *Computer Methods in Applied Mechanics and Engineering*, 195(25):3347–3381, May 2006. ISSN 0045-7825. doi: 10.1016/j.cma.2005.06.018. URL <https://www.sciencedirect.com/science/article/pii/S0045782505002732>.
- S. R. Idelsohn and E. Oñate. Finite volumes and finite elements: Two ‘good friends’. *International Journal for Numerical Methods in Engineering*, 37(19):3323–3341,

- October 1994. ISSN 0029-5981, 1097-0207. doi: 10.1002/nme.1620371908. URL <https://onlinelibrary.wiley.com/doi/10.1002/nme.1620371908>.
- ISO/TC 261. Terminology for Additive Manufacturing - General Principles - Terminology. Technical Report ISO/ASTM 52900:2021, International Standards Organization, 2021. URL <https://www.iso.org/standard/74514.html>.
- Hrvoje Jasak. HPC Deployment of OpenFOAM in an Industrial Setting. In *PRACE Seminar: Industrial Usage of HPC*, Stockholm, Sweden, March 2011.
- Peng Jin, Qian Tang, Jun Song, Qixiang Feng, Fuyu Guo, Xiaojie Fan, Mengxia Jin, and Fuyou Wang. Numerical investigation of the mechanism of interfacial dynamics of the melt pool and defects during laser powder bed fusion. *Optics & Laser Technology*, 143:107289, November 2021. ISSN 0030-3992. doi: 10.1016/j.optlastec.2021.107289. URL <https://www.sciencedirect.com/science/article/pii/S0030399221003777>.
- D. J Jones, J. C South, and E. B Klunker. On the numerical solution of elliptic partial differential equations by the method of lines. *Journal of Computational Physics*, 9(3):496–527, June 1972. ISSN 0021-9991. doi: 10.1016/0021-9991(72)90007-1. URL <https://www.sciencedirect.com/science/article/pii/0021999172900071>.
- A. Karma and W.-J. Rappel. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 57:4323–4349, 1998. doi: 10.1103/PhysRevE.57.4323. URL <https://doi.org/10.1103/PhysRevE.57.4323>.
- Christopher Katinas and Yung C. Shin. Prediction of initial transient behavior with stationary heating during laser powder bed fusion processes. *International Journal of Heat and Mass Transfer*, 153:119663, January 2020. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2020.119663.
- Trevor Keller, Greta Lindwall, Supriyo Ghosh, Li Ma, Brandon M. Lane, Fan Zhang, Ursula R. Kattner, Eric A. Lass, Jarred C. Heigel, Yaakov Idell, Maureen E. Williams, Andrew J. Allen, Jonathan E. Guyer, and Lyle E. Levine. Application of Finite Element, Phase-field, and CALPHAD-based Methods to Additive Manufacturing of Ni-based Superalloys. *Acta Materialia*, 139:244–253, January 2017. ISSN 1359-6454. doi: 10.1016/j.actamat.2017.05.003.
- Samuel W. Key and Raymond D. Krieg. Comparison of Finite-Element and Finite-Difference Methods. In *Numerical and Computer Methods in Structural Mechanics*, pages 337–352. Elsevier, 1973. ISBN 978-0-12-253250-4. doi: 10.1016/B978-0-12-253250-4.50019-1. URL <https://linkinghub.elsevier.com/retrieve/pii/B9780122532504500191>.
- David E Keyes, Lois C McInnes, Carol Woodward, William Gropp, Eric Myra, Michael Pernice, John Bell, Jed Brown, Alain Clo, Jeffrey Connors, Emil Constantinescu, Don Estep, Kate Evans, Charbel Farhat, Ammar Hakim, Glenn Hammond, Glen

- Hansen, Judith Hill, Tobin Isaac, Xiangmin Jiao, Kirk Jordan, Dinesh Kaushik, Efthimios Kaxiras, Alice Koniges, Kihwan Lee, Aaron Lott, Qiming Lu, John Magerlein, Reed Maxwell, Michael McCourt, Miriam Mehl, Roger Pawlowski, Amanda P Randles, Daniel Reynolds, Beatrice Rivière, Ulrich Rüde, Tim Scheibe, John Shadid, Brendan Sheehan, Mark Shephard, Andrew Siegel, Barry Smith, Xianzhu Tang, Cian Wilson, and Barbara Wohlmuth. Multiphysics simulations: Challenges and opportunities. *The International Journal of High Performance Computing Applications*, 27(1):4–83, February 2013. ISSN 1094-3420, 1741-2846. doi: 10.1177/1094342012468181. URL <http://journals.sagepub.com/doi/10.1177/1094342012468181>.
- Saad Khairallah, Andrew Anderson, and Alexander Rubenchik. Laser powder-bed fusion additive manufacturing: Effects of main physical processes on dynamical melt flow and pore formation from mesoscopic powder simulation. December 2015.
- Saad A. Khairallah and Andy Anderson. Mesoscopic simulation model of selective laser melting of stainless steel powder. *Journal of Materials Processing Technology*, 214(11):2627–2636, January 2014. ISSN 09240136. doi: 10.1016/j.jmatprotec.2014.06.001.
- Saad A. Khairallah, Andrew T. Anderson, Alexander Rubenchik, and Wayne E. King. Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. *Acta Materialia*, 108: 36–45, January 2016. ISSN 1359-6454. doi: 10.1016/j.actamat.2016.02.014.
- Saad A. Khairallah, Aiden A. Martin, Jonathan R. I. Lee, Gabe Guss, Nicholas P. Calta, Joshua A. Hammons, Michael H. Nielsen, Kevin Chaput, Edwin Schwalbach, Megna N. Shah, Michael G. Chapman, Trevor M. Willey, Alexander M. Rubenchik, Andrew T. Anderson, Y. Morris Wang, Manyalibo J. Matthews, and Wayne E. King. Controlling interdependent meso-nanosecond dynamics and defect generation in metal 3D printing. *Science (New York, N.Y.)*, 368(6491):660–665, January 2020. ISSN 0036-8075. doi: 10.1126/science.aay7830.
- M. D. Khomenko, N. W. Makoana, F. Kh. Mirzade, and S. Pityana. Coupled heat transfer, fluid flow and solidification kinetics for laser additive manufacturing applications. *Journal of Manufacturing Processes*, 67:611–618, July 2021. ISSN 1526-6125. doi: 10.1016/j.jmapro.2021.05.019. URL <https://www.sciencedirect.com/science/article/pii/S1526612521003418>.
- W. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, and S. A. Khairallah. Overview of modelling and simulation of metal powder bed fusion process at Lawrence Livermore National Laboratory. *Materials Science and Technology*, 31(8):957–968, January 2015a. ISSN 0267-0836. doi: 10.1179/1743284714Y.000000728.
- W. E. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, and A. M. Rubenchik. Laser powder bed fusion additive manufacturing of metals;

- physics, computational, and materials challenges. *Applied Physics Reviews*, 2(4): 041304, January 2015b. doi: 10.1063/1.4937809. URL <https://aip.scitation.org/doi/pdf/10.1063/1.4937809>.
- Andrew C. Kirby and Dimitri J. Mavriplis. GPU-Accelerated Discontinuous Galerkin Methods: 30x Speedup on 345 Billion Unknowns. Technical report, June 2020. URL <http://arxiv.org/pdf/2006.15698v2>.
- Robert M. Kirby, Spencer J. Sherwin, and Bernardo Cockburn. To CG or to HDG: A Comparative Study. *Journal of Scientific Computing*, 51(1):183–212, April 2012. ISSN 1573-7691. doi: 10.1007/s10915-011-9501-7. URL <https://doi.org/10.1007/s10915-011-9501-7>.
- Knowledge Sourcing Intelligence. Virtual Prototype Market Size, Share, Opportunities, COVID-19 Impact, And Trends By Tool (Finite Element Analysis, Computational Fluid Dynamics, Computer-Aided Manufacturing, Computer-Aided Design, Computer Automated Design), By Deployment Model (On-Premise, Cloud), By Industry Vertical (Healthcare, Automotive, Consumer Electronics, Aerospace And Defense, Telecommunications, Others), And By Geography - Forecasts From 2021 To 2026. Technical Report KSI061610465, October 2021. URL <https://www.knowledge-sourcing.com/report/virtual-prototype-market>.
- Holger Koch, Karl-Heinz Leitz, Andreas Otto, and Michael Schmidt. Laser deep penetration welding simulation based on a wavelength dependent absorption model. *Physics Procedia*, 5:309–315, January 2010. ISSN 1875-3892. doi: 10.1016/j.phpro.2010.08.057. URL <https://www.sciencedirect.com/science/article/pii/S1875389210004839>.
- Philipp Kopp, Victor Calo, Ernst Rank, and Stefan Kollmannsberger. Space-time hp-finite elements for heat evolution in laser powder bed fusion additive manufacturing. *Engineering with Computers*, September 2022. ISSN 1435-5663. doi: 10.1007/s00366-022-01719-1. URL <https://doi.org/10.1007/s00366-022-01719-1>.
- Carolin Körner, Elham Attar, and Peter Heinl. Mesoscopic simulation of selective beam melting processes. *Journal of Materials Processing Technology*, 211(6):978–987, January 2011. ISSN 09240136. doi: 10.1016/j.jmatprotec.2010.12.016.
- Carolin Körner, Andreas Bauereiß, and Elham Attar. Fundamental consolidation mechanisms during selective beam melting of powders. *Modelling and Simulation in Materials Science and Engineering*, 21(8):085011, January 2013. ISSN 0965-0393. doi: 10.1088/0965-0393/21/8/085011.
- O. B. Kovalev, A. V. Gusarov, and V. V. Belyaev. Morphology of random packing of micro-particles and its effect on the absorption of laser radiation during selective melting of powders. *International Journal of Engineering Science*, 157:103378, January 2020. ISSN 0020-7225. doi: 10.1016/j.ijengsci.2020.103378. URL <http://www.sciencedirect.com/science/article/pii/S0020722520301658>.

- Toni A. Krol, Gregor Branner, and Johannes Schilp. Modelle zur thermomechanischen Simulation metallverarbeitender Strahlschmelzprozesse. In *Proceedings of the ANSYS Conference & 27th CADFEM Users' Meeting*. January 2009.
- Martin Kronbichler and Wolfgang A. Wall. A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers. *SIAM Journal on Scientific Computing*, 40(5):A3423–A3448, January 2018. ISSN 1064-8275, 1095-7197. doi: 10.1137/16M110455X. URL <http://arxiv.org/abs/1611.03029>.
- Martin Kronbichler, Ababacar Diagne, and Hanna Holmgren. A fast massively parallel two-phase flow solver for microfluidic chip simulation. *The International Journal of High Performance Computing Applications*, 32(2):266–287, 2018. doi: 10.1177/1094342016671790.
- Michal Krzyzanowski and Dmytro Svyetlichnyy. A multiphysics simulation approach to selective laser melting modelling based on cellular automata and lattice Boltzmann methods. *Computational Particle Mechanics*, March 2021. ISSN 2196-4378, 2196-4386. doi: 10.1007/s40571-021-00397-y. URL <http://link.springer.com/10.1007/s40571-021-00397-y>.
- Vera E. Küng, Robert Scherr, Matthias Markl, and Carolin Körner. Multi-material model for the simulation of powder bed fusion additive manufacturing. *Computational Materials Science*, 194:110415, June 2021. ISSN 0927-0256. doi: 10.1016/j.commatsci.2021.110415. URL <https://www.sciencedirect.com/science/article/pii/S0927025621001403>.
- Leila Ladani, John Romano, William Brindley, and Sergei Burlatsky. Effective liquid conductivity for improved simulation of thermal transport in laser beam melting powder bed technology. *Additive Manufacturing*, 14:13–23, March 2017. ISSN 2214-8604. doi: 10.1016/j.addma.2016.12.004. URL <https://www.sciencedirect.com/science/article/pii/S2214860416301439>.
- Bruno Lafaurie, Carlo Nardone, Ruben Scardovelli, Stéphane Zaleski, and Gianluigi Zanetti. Modelling Merging and Fragmentation in Multiphase Flows with SURFER. *Journal of Computational Physics*, 113(1):134–147, July 1994. ISSN 0021-9991. doi: 10.1006/jcph.1994.1123. URL <http://www.sciencedirect.com/science/article/pii/S0021999184711235>.
- Imre Lakatos. Falsification and the Methodology of Scientific Research Programmes. In Sandra G. Harding, editor, *Can Theories Be Refuted? Essays on the Duhem-Quine Thesis*, Synthese Library, pages 205–259. Springer Netherlands, Dordrecht, 1976. ISBN 978-94-010-1863-0. doi: 10.1007/978-94-010-1863-0\_14. URL [https://doi.org/10.1007/978-94-010-1863-0\\_14](https://doi.org/10.1007/978-94-010-1863-0_14).
- Lev Landau. On the Theory of Phase Transitions. *Zh. Eksp. Teor. Fiz.*, 7:19–32, 1937. ISSN 2071-019. URL <http://archive.ujp.bitp.kiev.ua/files/journals/53/si/53SI08p.pdf>.

- Hans Petter Langtangen and Svein Linge. *Finite Difference Computing with PDEs: A Modern Software Approach*, volume 16 of *Texts in Computational Science and Engineering*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-55455-6 978-3-319-55456-3. doi: 10.1007/978-3-319-55456-3. URL <http://link.springer.com/10.1007/978-3-319-55456-3>.
- Mats G. Larson and Fredrik Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*, volume 10 of *Texts in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-33286-9 978-3-642-33287-6. doi: 10.1007/978-3-642-33287-6. URL <https://link.springer.com/10.1007/978-3-642-33287-6>.
- K. Q. Le, C. H. Wong, K.H.G. Chua, C. Tang, and H. Du. Discontinuity of overhanging melt track in selective laser melting process. *International Journal of Heat and Mass Transfer*, 162:120284, January 2020a. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2020.120284. URL <http://www.sciencedirect.com/science/article/pii/S0017931020332208>.
- Trong-Nhan Le, Yu-Lung Lo, and Ze-Hong Lin. Numerical Simulation and Experimental Validation of Melting and Solidification Process in Selective Laser Melting of IN718 Alloy. *Additive Manufacturing*, page 101519, January 2020b. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101519. URL <http://www.sciencedirect.com/science/article/pii/S2214860420308915>.
- Weng-Hoh Lee, Yi Zhang, and Jing Zhang. Discrete element modeling of powder flow and laser heating in direct metal laser sintering process. *Powder Technology*, 315: 300–308, January 2017. ISSN 00325910. doi: 10.1016/j.powtec.2017.04.002.
- Yousub Lee and W. Zhang. Mesoscopic simulation of heat transfer and fluid flow in laser Powder bed additive manufacturing. In *International Solid Free Form Fabrication Symposium*, pages 1154–1165, 2015. URL <http://utw10945.utweb.utexas.edu/sites/default/files/2015/2015-94-Lee.pdf>.
- Y.S. Lee and W. Zhang. Modeling of heat transfer, fluid flow and solidification microstructure of nickel-base superalloy fabricated by laser powder bed fusion. *Additive Manufacturing*, 12:178188, January 2016. ISSN 2214-8604. doi: 10.1016/j.addma.2016.05.003.
- Charles E. Leiserson, Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez, and Tao B. Schardl. There’s plenty of room at the Top: What will drive computer performance after Moore’s law? *Science*, 368(6495): eaam9744, June 2020. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aam9744. URL <https://www.science.org/doi/10.1126/science.aam9744>.
- Randall J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Basel, Basel, 1992. ISBN 978-3-7643-2723-1 978-3-0348-8629-1. doi: 10.1007/978-3-0348-8629-1. URL <http://link.springer.com/10.1007/978-3-0348-8629-1>.

- E. L. Li, L. Wang, A. B. Yu, and Z. Y. Zhou. A three-phase model for simulation of heat transfer and melt pool behaviour in laser powder bed fusion process. *Powder Technology*, December 2020a. ISSN 0032-5910. doi: 10.1016/j.powtec.2020.11.061. URL <http://www.sciencedirect.com/science/article/pii/S0032591020311207>.
- Ming-Jian Li, Jiawei Chen, Yanping Lian, Feiyu Xiong, and Daining Fang. An efficient and high-fidelity local multi-mesh finite volume method for heat transfer and fluid flow problems in metal additive manufacturing. *Computer Methods in Applied Mechanics and Engineering*, 404:115828, February 2023. ISSN 0045-7825. doi: 10.1016/j.cma.2022.115828. URL <https://www.sciencedirect.com/science/article/pii/S0045782522007848>.
- Q. Li, K. H. Luo, Q. J. Kang, Y. L. He, Q. Chen, and Q. Liu. Lattice Boltzmann methods for multiphase flow and phase-change heat transfer. *Progress in Energy and Combustion Science*, 52:62–105, January 2016. ISSN 03601285. doi: 10.1016/j.pecs.2015.10.001.
- Xuxiao Li and Wenda Tan. Numerical Modeling of Powder Gas Interaction Relative to Laser Powder Bed Fusion Process. *Journal of Manufacturing Science and Engineering*, pages 1–26, January 2020. ISSN 1087-1357. doi: 10.1115/1.4048443.
- Xuxiao Li, Cang Zhao, Tao Sun, and Wenda Tan. Revealing Transient Powder-Gas Interaction in Laser Powder Bed Fusion Process through Multi-physics Modeling and High-speed Synchrotron X-ray Imaging. *Additive Manufacturing*, page 101362, January 2020b. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101362. URL <http://www.sciencedirect.com/science/article/pii/S221486042030734X>.
- Xuan Liang, Devlin Hayduke, and Albert C. To. An enhanced layer lumping method for accelerating simulation of metal components produced by laser powder bed fusion. *Additive Manufacturing*, page 101881, February 2021. ISSN 2214-8604. doi: 10.1016/j.addma.2021.101881. URL <https://www.sciencedirect.com/science/article/pii/S2214860421000464>.
- Guang Lin, Jianguo Liu, and Farrah Sadre-Marandi. A comparative study on the weak Galerkin, discontinuous Galerkin, and mixed finite element methods. *Journal of Computational and Applied Mathematics*, 273:346–362, January 2015. ISSN 03770427. doi: 10.1016/j.cam.2014.06.024. URL <https://linkinghub.elsevier.com/retrieve/pii/S0377042714003057>.
- Stephen Lin, Zhengtao Gan, Jinhui Yan, and Gregory J. Wagner. A conservative level set method on unstructured meshes for modeling multiphase thermo-fluid flow in additive manufacturing processes. *Computer Methods in Applied Mechanics and Engineering*, 372:113348, January 2020. ISSN 0045-7825. doi: 10.1016/j.cma.2020.113348.
- Steven J. Lind, Benedict D. Rogers, and Peter K. Stansby. Review of smoothed particle hydrodynamics: Towards converged Lagrangian flow modelling. *Proceedings of*

- the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2241): 20190801, September 2020. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2019.0801. URL <https://royalsocietypublishing.org/doi/10.1098/rspa.2019.0801>.
- O. A. Liskovets. The method of straight lines for one-dimensional mixed non-stationary problems and estimation of the mean square error. *USSR Computational Mathematics and Mathematical Physics*, 5(2):273–278, January 1965. ISSN 0041-5553. doi: 10.1016/0041-5553(65)90053-4. URL <https://www.sciencedirect.com/science/article/pii/0041555365900534>.
- L. Liu, M. Huang, Y. H. Ma, M. L. Qin, and T. T. Liu. Simulation of Powder Packing and Thermo-Fluid Dynamic of 316L Stainless Steel by Selective Laser Melting. *Journal of Materials Engineering and Performance*, November 2020. ISSN 1544-1024. doi: 10.1007/s11665-020-05230-w. URL <https://doi.org/10.1007/s11665-020-05230-w>.
- Wenqi Liu, Junhe Lian, Jiaojiao Wu, Napat Vajragupta, and Alexander Hartmaier. A numerical investigation on the effects of porosity on the plastic anisotropy of additive manufactured stainless steel with various crystallographic textures, April 2021. URL <https://popups.uliege.be/esaform21/index.php?id=4308>.
- Wing Kam Liu, Shaofan Li, and Harold S. Park. Eighty Years of the Finite Element Method: Birth, Evolution, and Future. *Archives of Computational Methods in Engineering*, 29(6):4431–4453, October 2022. ISSN 1134-3060, 1886-1784. doi: 10.1007/s11831-022-09740-9. URL <https://link.springer.com/10.1007/s11831-022-09740-9>.
- Yuan Liu and Yong-Tao Zhang. A Robust Reconstruction for Unstructured WENO Schemes. *Journal of Scientific Computing*, 54(2-3):603–621, February 2013. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-012-9598-3. URL <http://link.springer.com/10.1007/s10915-012-9598-3>.
- A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. FFC: The FEniCS form compiler. In A. Logg, K.-A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*, chapter 11. Springer, 2012a.
- A. Logg, G. N. Wells, and J. Hake. DOLFIN: A C++/Python finite element library. In A. Logg, K.-A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*, chapter 10. Springer, 2012b.
- Anders Logg, Kent-Andre Mardal, and Garth Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012c. ISBN 978-3-642-23098-1 978-3-642-23099-8. doi: 10.1007/978-3-642-23099-8. URL <http://link.springer.com/10.1007/978-3-642-23099-8>.

- Pan Lu, Liu Tong, Wang Wen-Hao, Gao Yu, Zhang Cheng-Lin, and Liu Jiang-Lin. Mesoscopic simulation of overlapping behavior in laser powder bed additive manufacturing. *Materials Research Express*, 8(12):125801, December 2021. ISSN 2053-1591. doi: 10.1088/2053-1591/ac40b4. URL <https://dx.doi.org/10.1088/2053-1591/ac40b4>.
- Sheng Luo, Hongze Wang, Zhenyang Gao, Yi Wu, and Haowei Wang. Interaction between high-velocity gas and liquid in gas atomization revealed by a new coupled simulation model. *Materials & Design*, 212:110264, December 2021. ISSN 0264-1275. doi: 10.1016/j.matdes.2021.110264. URL <https://www.sciencedirect.com/science/article/pii/S0264127521008194>.
- C. Lüthi, M. Afrasiabi, and M. Bambach. An adaptive smoothed particle hydrodynamics (SPH) scheme for efficient melt pool simulations in additive manufacturing. *Computers & Mathematics with Applications*, 139:7–27, June 2023. ISSN 0898-1221. doi: 10.1016/j.camwa.2023.03.003. URL <https://www.sciencedirect.com/science/article/pii/S0898122123000925>.
- Ju Ma, Xiaofeng Niu, Yan Zhou, Wenqi Li, Yan Liu, Mengqing Shen, Hongxia Wang, Weili Cheng, and Zhiyong You. Simulation of solidification microstructure evolution of 316L stainless steel fabricated by selective laser melting using a coupled model of smooth particle hydrodynamics and cellular automata. *Journal of Materials Research and Technology*, page S2238785423024225, October 2023. ISSN 22387854. doi: 10.1016/j.jmrt.2023.09.296. URL <https://linkinghub.elsevier.com/retrieve/pii/S2238785423024225>.
- Vadim Maltsev, Dean Yuan, Karl W. Jenkins, Martin Skote, and Panagiotis Tsoutsanis. Hybrid discontinuous Galerkin-finite volume techniques for compressible flows on unstructured meshes. *Journal of Computational Physics*, 473:111755, January 2023. ISSN 00219991. doi: 10.1016/j.jcp.2022.111755. URL <https://linkinghub.elsevier.com/retrieve/pii/S002199912200818X>.
- Vijay Mandal, Shashank Sharma, Sudhanshu S. Singh, and J. Ramkumar. Laser Surface Texturing in Powder Bed Fusion: Numerical Simulation and Experimental Characterization. *Metals and Materials International*, 28(1):181–196, January 2022. ISSN 2005-4149. doi: 10.1007/s12540-021-01072-w. URL <https://doi.org/10.1007/s12540-021-01072-w>.
- K. Marchais, J. Girardot, C. Metton, and I. Iordanoff. A 3D DEM simulation to study the influence of material and process parameters on spreading of metallic powder in additive manufacturing. *Computational Particle Mechanics*, January 2021. ISSN 2196-4386. doi: 10.1007/s40571-020-00380-z. URL <https://doi.org/10.1007/s40571-020-00380-z>.
- Matthias Markl, Alexander M. Rausch, Vera E. Küng, and Carolin Körner. SAMPLE: A Software Suite to Predict Consolidation and Microstructure for Powder Bed Fusion

- Additive Manufacturing. *Advanced Engineering Materials*, page 1901270, January 2019. ISSN 1438-1656. doi: 10.1002/adem.201901270.
- John Marshall, Alistair Adcroft, Chris Hill, Lev Perelman, and Curt Heisey. A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, 102(C3):5753–5766, March 1997. ISSN 01480227. doi: 10.1029/96JC02775. URL <http://doi.wiley.com/10.1029/96JC02775>.
- Aiden A. Martin, Nicholas P. Calta, Saad A. Khairallah, Jenny Wang, Phillip J. DePond, Anthony Y. Fong, Vivek Thampy, Gabe M. Guss, Andrew M. Kiss, Kevin H. Stone, Christopher J. Tassone, Johanna Nelson Weker, Michael F. Toney, Tony van Buuren, and Manyalibo J. Matthews. Dynamics of pore formation during laser powder bed fusion additive manufacturing. *Nature Communications*, 10(1):1987, January 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-10009-2. URL <https://doi.org/10.1038/s41467-019-10009-2>.
- Mayer, Dennis. Automatisierte Transformation von partiellen Differentialgleichungen in die schwache Form für die Finite-Elemente-Methode. Master’s thesis, University of Augsburg, Faculty of Applied Computer Science, October 2023.
- Yaasin A. Mayi, Morgan Dal, Patrice Peyre, Michel Bellet, Charlotte Metton, Clara Moriconi, and Remy Fabbro. Transient dynamics and stability of keyhole at threshold in laser powder bed fusion regime investigated by finite element modeling. *Journal of Laser Applications*, 33(1):012024, February 2021. ISSN 1042-346X, 1938-1387. doi: 10.2351/7.0000330. URL <http://lia.scitation.org/doi/10.2351/7.0000330>.
- Sally A. McKee and Robert W. Wisniewski. Memory Wall. In David Padua, editor, *Encyclopedia of Parallel Computing*, pages 1110–1116. Springer US, Boston, MA, 2011. ISBN 978-0-387-09766-4. doi: 10.1007/978-0-387-09766-4\_234. URL [https://doi.org/10.1007/978-0-387-09766-4\\_234](https://doi.org/10.1007/978-0-387-09766-4_234).
- Andrew T. T. McRae, Gheorghe-Teodor Bercea, Lawrence Mitchell, David A. Ham, and Colin J. Cotter. Automated generation and symbolic manipulation of tensor product finite elements. *SIAM Journal on Scientific Computing*, 38(5):S25–S47, 2016. doi: 10.1137/15M1021167. URL <http://arxiv.org/abs/1411.2940>.
- Tijan Mede, Andraž Kocjan, Irena Paulin, Godec, and Matjaž. Numerical Mesoscale Modelling of Microstructure Evolution during Selective Laser Melting. *Metals*, 10(6):800, January 2020. doi: 10.3390/met10060800.
- Christoph Meier, Sebastian L. Fuchs, A. John Hart, and Wolfgang A. Wall. A novel smoothed particle hydrodynamics formulation for thermo-capillary phase change problems with focus on metal additive manufacturing melt pool modeling. *arXiv:2012.08788 [cs]*, December 2020. URL <http://arxiv.org/abs/2012.08788>.

- Christoph Meier, Sebastian L. Fuchs, Nils Much, Jonas Nitzler, Ryan W. Penny, Patrick M. Praegla, Sebastian D. Pröll, Yushen Sun, Reimar Weissbach, Magdalena Schreter, Neil E. Hodge, A. John Hart, and Wolfgang A. Wall. Physics-Based Modeling and Predictive Simulation of Powder Bed Fusion Additive Manufacturing Across Length Scales. *arXiv:2103.16982 [cs]*, March 2021. URL <http://arxiv.org/abs/2103.16982>.
- Takafumi Mikami, Hidehiro Kamiya, and Masayuki Horio. Numerical simulation of cohesive powder behavior in a fluidized bed. *Chemical Engineering Science*, 53(10): 1927–1940, May 1998. ISSN 0009-2509. doi: 10.1016/S0009-2509(97)00325-4. URL <https://www.sciencedirect.com/science/article/pii/S0009250997003254>.
- Mojtaba Moatamedi, Thurai Rahulan, and Hassan Khawaja. *Multiphysics Simulations in Automotive and Aerospace Applications*. Elsevier, 2021. ISBN 978-0-12-817899-7. doi: 10.1016/C2018-0-02600-6. URL <https://linkinghub.elsevier.com/retrieve/pii/C20180026006>.
- K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, April 2005. ISBN 978-1-139-44320-3.
- N. C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable Discontinuous Galerkin Methods. In Jan S. Hesthaven and Einar M. Rønquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76, pages 63–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-15336-5 978-3-642-15337-2. doi: 10.1007/978-3-642-15337-2\_4. URL [http://link.springer.com/10.1007/978-3-642-15337-2\\_4](http://link.springer.com/10.1007/978-3-642-15337-2_4).
- Patiparn Ninpetch, Pruet Kowitwarangkul, Sitthipong Mahathanabodee, Prasert Chalermkarnnon, and Phadungsak Rattanadecho. Computational Investigation of Thermal Behavior and Molten Metal Flow with Moving Laser Heat Source for Selective Laser Melting Process. *Case Studies in Thermal Engineering*, page 100860, January 2021. ISSN 2214-157X. doi: 10.1016/j.csite.2021.100860. URL <http://www.sciencedirect.com/science/article/pii/S2214157X2100023X>.
- Patiparn Ninpetch, Prasert Chalermkarnnon, and Pruet Kowitwarangkul. Multiphysics Simulation of Thermal-Fluid Behavior in Laser Powder Bed Fusion of H13 Steel: Influence of Layer Thickness and Energy Input. *Metals and Materials International*, 29(2):536–551, February 2023. ISSN 2005-4149. doi: 10.1007/s12540-022-01239-z. URL <https://doi.org/10.1007/s12540-022-01239-z>.
- J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, July 1971. ISSN 1865-8784. doi: 10.1007/BF02995904. URL <https://doi.org/10.1007/BF02995904>.

- Aleksey Noskov, Torunn K. Ervik, Ilya Tsvil'skiy, Albert Gilmutdinov, and Yngvar Thomassen. Characterization of ultrafine particles emitted during laser-based additive manufacturing of metal parts. *Scientific Reports*, 10, December 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-78073-z. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7710759/>.
- Nothelfer, Tim. Recherche zu bestehenden Umsetzungen von numerischen Berechnungsverfahren für Multiphysik Probleme. Seminar paper, University of Augsburg, Faculty of Applied Computer Science, July 2023.
- Yuria Okagaki, Taisuke Yonomoto, Masahiro Ishigaki, and Yoshiyasu Hirose. Numerical Study on an Interface Compression Method for the Volume of Fluid Approach. *Fluids*, 6(2):80, February 2021. ISSN 2311-5521. doi: 10.3390/fluids6020080. URL <https://www.mdpi.com/2311-5521/6/2/80>.
- Alaa Olleak and Zhimin Xi. Scan-wise adaptive remeshing for efficient LPBF process simulation: The thermal problem. *Manufacturing Letters*, 23:75–78, January 2020a. ISSN 2213-8463. doi: 10.1016/j.mfglet.2020.01.003. URL <http://www.sciencedirect.com/science/article/pii/S2213846319301270>.
- Alaa Olleak, Florian Dugast, Prajwal Bharadwaj, Seth Strayer, Shawn Hinnebusch, Sneha Narra, and Albert C. To. Enabling Part-Scale Scanwise process simulation for predicting melt pool variation in LPBF by combining GPU-based Matrix-free FEM and adaptive Remeshing. *Additive Manufacturing Letters*, 3:100051, December 2022. ISSN 2772-3690. doi: 10.1016/j.addlet.2022.100051. URL <https://www.sciencedirect.com/science/article/pii/S2772369022000251>.
- Alaaeldin Olleak and Zhimin Xi. Part-scale Finite Element Modeling of the SLM Process with Layer-wise Adaptive Remeshing for Thermal History and Porosity Prediction. *Journal of Manufacturing Science and Engineering*, pages 1–26, January 2020b. ISSN 1087-1357. doi: 10.1115/1.4047733.
- Andreas Otto, Holger Koch, Karl-Heinz Leitz, and Michael Schmidt. Numerical Simulations - A Versatile Approach for Better Understanding Dynamics in Laser Material Processing. *Physics Procedia*, 12:11–20, January 2011. ISSN 1875-3892. doi: 10.1016/j.phpro.2011.03.003. URL <https://www.sciencedirect.com/science/article/pii/S1875389211000824>.
- Andreas Otto, Holger Koch, and Rodrigo Gomez Vazquez. Multiphysical Simulation of Laser Material Processing. *Physics Procedia*, 39:843–852, January 2012. ISSN 18753892. doi: 10.1016/j.phpro.2012.10.109.
- Pakhale, Vinit Vikas. Multiphysics Simulation of Melt Pool in Laser Powder Bed Fusion by Coupled Discrete Element Method and Computational Fluid Dynamics - ProQuest, June 2023. URL <https://www.proquest.com/openview/e439166b65e8c7b69e0836faf0237a62/1?cbl=18750&diss=y&pq-origsite=gscholar>.

- Mangesh V. Pantawane, Sriswaroop Dasari, Srinivas Aditya Mantri, Rajarshi Banerjee, and Narendra B. Dahotre. Rapid thermokinetics driven nanoscale vanadium clustering within martensite laths in laser powder bed fused additively manufactured Ti6Al4V. *Materials Research Letters*, 8(10):383–389, January 2020. doi: 10.1080/21663831.2020.1772396.
- Chinnapat Panwisawas, Chunlei Qiu, Magnus J. Anderson, Yogesh Sovani, Richard P. Turner, Moataz M. Attallah, Jeffery W. Brooks, and Hector C. Basoalto. Mesoscale modelling of selective laser melting: Thermal fluid dynamics and microstructural evolution. *Computational Materials Science*, 126:479–490, January 2017. ISSN 0927-0256. doi: 10.1016/j.commatsci.2016.10.011. URL <https://www.sciencedirect.com/science/article/pii/S0927025616305079>.
- Bernhard Peters. The extended discrete element method (XDEM) for multi-physics applications. *Scholarly Journal of Engineering Research*, 2013(2):1–20, January 2013.
- D. Pilipenko, M. Fleck, and H. Emmerich. On numerical aspects of phase field fracture modelling. *The European Physical Journal Plus*, 126(10):100, October 2011. ISSN 2190-5444. doi: 10.1140/epjp/i2011-11100-3. URL <https://doi.org/10.1140/epjp/i2011-11100-3>.
- Yehuda Pinchover and Jacob Rubinstein. *An Introduction to Partial Differential Equations*. Cambridge University Press, May 2005. ISBN 978-0-521-84886-2.
- Karl Raimund Popper. *Logik der Forschung*. 1989.
- Pikee Priya, Brian Mercer, Shenyan Huang, Mohamed Aboukhatwa, Lang Yuan, and Santanu Chaudhuri. Towards prediction of microstructure during laser based additive manufacturing process of co-Cr-Mo powder beds. *Materials & Design*, page 109117, January 2020. ISSN 0264-1275. doi: 10.1016/j.matdes.2020.109117.
- Sebastian D. Proell, Peter Munch, Wolfgang A. Wall, and Christoph Meier. A highly efficient computational framework for fast scan-resolved simulations of metal additive manufacturing processes on the scale of real parts, February 2023. URL <http://arxiv.org/abs/2302.05164>.
- Chunlei Qiu, Chinnapat Panwisawas, Mark Ward, Hector C. Basoalto, Jeffery W. Brooks, and Moataz M. Attallah. On the role of melt flow into the surface structure and porosity development during selective laser melting. *Acta Materialia*, 96: 72–79, January 2015. ISSN 1359-6454. doi: 10.1016/j.actamat.2015.06.004.
- Christopher Rackauckas and Qing Nie. Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

- Christopher Rackauckas and Qing Nie. Confederated modular differential equation APIs for accelerated algorithm development and benchmarking. *Advances in Engineering Software*, 132:1–6, 2019.
- Abha Rai, Harald Helmer, and Carolin Körner. Simulation of grain structure evolution during powder bed based additive manufacturing. *Additive Manufacturing*, 13:124–134, January 2017. ISSN 2214-8604. doi: 10.1016/j.addma.2016.10.007. URL <http://www.sciencedirect.com/science/article/pii/S2214860416302822>.
- M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving non-linear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- Vaishak Ramesh Sagar, Samuel Lorin, Johan Göhl, Johannes Quist, Christoffer Cromvik, Andreas Mark, Klas Jareteg, Fredrik Edelvik, Kristina Wärmefjord, and Rikard Söderberg. A SIMULATION STUDY ON THE EFFECT OF PARTICLE SIZE DISTRIBUTION ON THE PRINTED GEOMETRY IN SELECTIVE LASER MELTING. *Journal of Manufacturing Science and Engineering*, pages 1–14, October 2021. ISSN 1087-1357. doi: 10.1115/1.4052705. URL <https://doi.org/10.1115/1.4052705>.
- Florian Rathgeber, David A. Ham, Lawrence Mitchell, Michael Lange, Fabio Luporini, Andrew T. T. McRae, Gheorghe-Teodor Bercea, Graham R. Markall, and Paul H. J. Kelly. Firedrake: Automating the finite element method by composing abstractions. *ACM Trans. Math. Softw.*, 43(3):24:1–24:27, 2016. ISSN 0098-3500. doi: 10.1145/2998441. URL <http://arxiv.org/abs/1501.01809>.
- Santosh K. Rauniar and Kevin Chou. Melt Pool Analysis and Mesoscale Simulation of Laser Powder Bed Fusion Process (L-PBF) with Ti-6Al-4V Powder Particles. *JOM*, 71(3):938–945, March 2019. ISSN 1543-1851. doi: 10.1007/s11837-018-3208-2. URL <https://doi.org/10.1007/s11837-018-3208-2>.
- Alexander M. Rausch, Vera E. Küng, Christoph Pobel, Matthias Markl, and Carolin Körner. Predictive Simulation of Process Windows for Powder Bed Fusion Additive Manufacturing: Influence of the Powder Bulk Density. *Materials*, 10(10):1117, October 2017. ISSN 1996-1944. doi: 10.3390/ma10101117. URL <https://www.mdpi.com/1996-1944/10/10/1117>.
- W H Reed and T R Hill. Triangular mesh methods for the neutron transport equation. October 1973.
- Yong Ren and Qian Wang. A finite difference method for fast prediction and control of part-scale temperature evolution in laser powder bed fusion. *Journal of Manufacturing Processes*, 93:299–314, May 2023. ISSN 1526-6125. doi: 10.1016/j.jmapro

- .2023.03.020. URL <https://www.sciencedirect.com/science/article/pii/S1526612523002281>.
- Zhihao Ren, David Z. Zhang, Guang Fu, Junjie Jiang, and Miao Zhao. ofvHigh-fidelity modelling of selective laser melting copper alloy: Laser reflection behavior and thermal-fluid dynamics. *Materials & Design*, page 109857, June 2021. ISSN 0264-1275. doi: 10.1016/j.matdes.2021.109857. URL <https://www.sciencedirect.com/science/article/pii/S026412752100410X>.
- Lewis Fry Richardson and Richard Tetley Glazebrook. IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357, January 1911. doi: 10.1098/rsta.1911.0009. URL <https://royalsocietypublishing.org/doi/10.1098/rsta.1911.0009>.
- M. R. Ridolfi, P. Folgarait, and A. Di Schino. Modelling Selective Laser Melting of Metallic Powders. *Metallurgist*, January 2020a. ISSN 0026-0894. doi: 10.1007/s11015-020-01031-7.
- Maria Rita Ridolfi, Paolo Folgarait, and Andrea Di Schino. Laser Operating Windows Prediction in Selective Laser-Melting Processing of Metallic Powders: Development and Validation of a Computational Fluid Dynamics-Based Model. *Materials*, 13(6): 1424, January 2020b. doi: 10.3390/ma13061424.
- Theron M. Rodgers, Jonathan D. Madison, and Veena Tikare. Simulation of metal additive manufacturing microstructures using kinetic Monte Carlo. *Computational Materials Science*, 135:78–89, January 2017. ISSN 09270256. doi: 10.1016/j.commatsci.2017.03.053. URL <http://www.sciencedirect.com/science/article/pii/S0927025617301751>.
- Matthew Rolchigo, Robert Carson, and James Belak. Understanding Uncertainty in Microstructure Evolution and Constitutive Properties in Additive Process Modeling. *Metals*, 12(2):324, February 2022. ISSN 2075-4701. doi: 10.3390/met12020324. URL <https://www.mdpi.com/2075-4701/12/2/324>.
- Y. Saadlaoui, E. Feulvarch, J. Leblond, and J. Bergheau. Numerical Simulation of the Molten Pool of a Powder Bed. In *14th WCCM-ECCOMAS Congress*. CIMNE, 2021. doi: 10.23967/wccm-eccomas.2020.246. URL [https://www.scipedia.com/public/SAADLAOUI\\_et\\_al\\_2021a](https://www.scipedia.com/public/SAADLAOUI_et_al_2021a).
- Adrian S. Sabau, Lang Yuan, Narendran Raghavan, Matthew Bement, Srdjan Simunovic, John A. Turner, and Vipul K. Gupta. Fluid Dynamics Effects on Microstructure Prediction in Single-Laser Tracks for Additive Manufacturing of IN625. *Metallurgical and Materials Transactions B*, 143(4):172, January 2020. ISSN 1073-5615. doi: 10.1007/s11663-020-01808-w.

- A. A. Samarskii. Monotonic difference schemes for elliptic and parabolic equations in the case of a non-selfadjoint elliptic operator. *USSR Computational Mathematics and Mathematical Physics*, 5(3):212–217, January 1965. ISSN 0041-5553. doi: 10.1016/0041-5553(65)90158-8. URL <https://www.sciencedirect.com/science/article/pii/0041555365901588>.
- Abd Essamade Saufi, Olivier Desjardins, and Alberto Cuoci. An accurate methodology for surface tension modeling in OpenFOAM. *arXiv:2005.02865 [cs]*, May 2020. URL <http://arxiv.org/abs/2005.02865>.
- William E. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Elsevier, July 2012. ISBN 978-0-12-801551-3.
- Thomas B. A. Senior and John Leonidas Volakis. *Approximate Boundary Conditions in Electromagnetics*. IET, 1995. ISBN 978-0-85296-849-9.
- M Shademan, R M Barron, and R Balachandar. Evaluation of OpenFOAM in Academic Research and Industrial Applications. In *21st Conference of the CFD Society of Canada*, Canada, 2013.
- S. Shenouda and A. Hoff. Discrete Element Method Analysis for Metal Powders Used in Additive Manufacturing, and DEM Simulation Tutorial Using LIGGGHTS-PUBLIC. Technical report, January 2020.
- Yeonjong Shin, Jerome Darbon, and George Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. *Communications in Computational Physics*, 28(5):2042–2074, June 2020. ISSN 1815-2406, 1991-7120. doi: 10.4208/cicp.OA-2020-0193. URL <http://arxiv.org/abs/2004.01806>.
- Subin Shrestha and Kevin Chou. A build surface study of Powder-Bed Electron Beam Additive Manufacturing by 3D thermo-fluid simulation and white-light interferometry. *International Journal of Machine Tools and Manufacture*, 121:37–49, October 2017. ISSN 0890-6955. doi: 10.1016/j.ijmachtools.2017.04.005. URL <https://www.sciencedirect.com/science/article/pii/S0890695516305247>.
- Subin Shrestha and Kevin Chou. A study of transient and steady-state regions from single-track deposition in laser powder bed fusion. *Journal of Manufacturing Processes*, 61:226–235, January 2021. ISSN 15266125. doi: 10.1016/j.jmapro.2020.11.023. URL <https://linkinghub.elsevier.com/retrieve/pii/S1526612520308100>.
- Subin Shrestha, Santosh Rauniyar, and Kevin Chou. Thermo-Fluid Modeling of Selective Laser Melting: Single-Track Formation Incorporating Metallic Powder. *Journal of Materials Engineering and Performance*, 28(2):611–619, February 2019. ISSN 1544-1024. doi: 10.1007/s11665-018-3574-5. URL <https://doi.org/10.1007/s11665-018-3574-5>.

- Chi-Wang Shu. High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, March 2003. ISSN 1061-8562. doi: 10.1080/1061856031000104851. URL <https://doi.org/10.1080/1061856031000104851>.
- Liang Si, Tengfang Zhang, Mengyuan Zhou, Maoyuan Li, Yun Zhang, and Huamin Zhou. Numerical simulation of the flow behavior and powder spreading mechanism in powder bed-based additive manufacturing. *Powder Technology*, 394:1004–1016, December 2021. ISSN 0032-5910. doi: 10.1016/j.powtec.2021.09.010. URL <https://www.sciencedirect.com/science/article/pii/S0032591021007981>.
- Gordon D. Smith, Gordon D. Smith, and Smith Smith, Gordon Dennis. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press, 1985. ISBN 978-0-19-859650-9.
- Pavel Šolín, Jakub Červený, and Ivo Doležel. Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM. *Mathematics and Computers in Simulation*, 77(1): 117–132, February 2008. ISSN 0378-4754. doi: 10.1016/j.matcom.2007.02.011. URL <https://www.sciencedirect.com/science/article/pii/S0378475407001504>.
- Matthias Sonntag and Claus-Dieter Munz. Shock Capturing for Discontinuous Galerkin Methods using Finite Volume Subcells. In Jürgen Fuhrmann, Mario Ohlberger, and Christian Rohde, editors, *Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems*, Springer Proceedings in Mathematics & Statistics, pages 945–953, Cham, 2014. Springer International Publishing. ISBN 978-3-319-05591-6. doi: 10.1007/978-3-319-05591-6\_96.
- Alexander Staroselsky, Ranadip Acharya, and Brice Cassenti. Development of unified framework for microstructure, residual stress, and crack propensity prediction using phase-field simulations. *International Journal of Computational Methods and Experimental Measurements*, 8(2):111–122, January 2020. ISSN 2046-0546. doi: 10.2495/CMEM-V8-N2-111-122.
- Seth T. Strayer, William J. Frieden Templeton, Florian X. Dugast, Sneha P. Narra, and Albert C. To. Accelerating High-Fidelity Thermal Process Simulation of Laser Powder Bed Fusion via the Computational Fluid Dynamics Imposed Finite Element Method (CIFEM). *Additive Manufacturing Letters*, page 100081, July 2022. ISSN 2772-3690. doi: 10.1016/j.addlet.2022.100081. URL <https://www.sciencedirect.com/science/article/pii/S2772369022000500>.
- G. G. Stubblefield, K. Fraser, B. J. Phillips, J. B. Jordon, and P. G. Allison. A Mesh-free computational framework for the numerical simulation of the solid-state additive manufacturing process, additive friction stir-deposition (AFS-D). *Materials & Design*, page 109514, January 2021. ISSN 0264-1275. doi: 10.1016/j.matdes.2021.109514. URL <http://www.sciencedirect.com/science/article/pii/S0264127521000678>.

- N. Sukumar. Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids. *International Journal for Numerical Methods in Engineering*, 57(1):1–34, 2003. ISSN 1097-0207. doi: 10.1002/nme.664. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.664>.
- Zhe Sun, Yuan-Hui Chueh, and Lin Li. Multiphase mesoscopic simulation of multiple and functionally gradient materials laser powder bed fusion additive manufacturing processes. *Additive Manufacturing*, page 101448, January 2020. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101448. URL <http://www.sciencedirect.com/science/article/pii/S2214860420308204>.
- J. Svenungsson, I. Choquet, and A. Kaplan. Marangoni force and numerical instability when modelling keyhole laser welding. *Lasers in Manufacturing Conference 2017*, pages 1–10, January 2017. URL [https://www.wlt.de/lim/Proceedings2017/Data/PDF/Contribution245\\_final.pdf](https://www.wlt.de/lim/Proceedings2017/Data/PDF/Contribution245_final.pdf).
- J. L. Tan, C. Tang, and C. H. Wong. Study and modeling of melt pool evolution in selective laser melting process of SS316L. *MRS Communications*, 8(3):1178–1183, September 2018. ISSN 2159-6859, 2159-6867. doi: 10.1557/mrc.2018.180. URL <https://www.cambridge.org/core/journals/mrs-communications/article/abs/study-and-modeling-of-melt-pool-evolution-in-selective-laser-melting-process-of-ss316l/FA79572DA6C06372C8AC3C3ACBBD5007>.
- Chao Tang. Phase Field Modelling of Dendritic Solidification Under Additive Manufacturing Conditions. page 14.
- Chao Tang, Liming Yao, and Hejun Du. Computational framework for the simulation of multi material laser powder bed fusion. *International Journal of Heat and Mass Transfer*, 191:122855, August 2022. ISSN 0017-9310. doi: 10.1016/j.ijheatmasstransfer.2022.122855. URL <https://www.sciencedirect.com/science/article/pii/S0017931022003374>.
- Thomas N. Theis and H.-S. Philip Wong. The End of Moore’s Law: A New Beginning for Information Technology. *Computing in Science & Engineering*, 19(2):41–50, March 2017. ISSN 1521-9615. doi: 10.1109/MCSE.2017.29. URL <http://ieeexplore.ieee.org/document/7878935/>.
- Vidar Thomée. The finite difference versus the finite element method for the solution of boundary value problems. *Bulletin of the Australian Mathematical Society*, 29(2): 267–288, April 1984. ISSN 0004-9727, 1755-1633. doi: 10.1017/S000497270002150X. URL [https://www.cambridge.org/core/product/identifier/S000497270002150X/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S000497270002150X/type/journal_article).
- Thörner, Katharina. Analyse von Simulationsmethoden auf Mesoebene für das selektive Laser Schmelzen. Seminar paper, University of Augsburg, Faculty of Applied Computer Science, July 2023.

- Yunfu Tian, Lijun Yang, Dejin Zhao, Yiming Huang, and Jiajing Pan. Numerical analysis of powder bed generation and single track forming for selective laser melting of SS316L stainless steel. *Journal of Manufacturing Processes*, 58:964–974, January 2020. ISSN 1526-6125. doi: 10.1016/j.jmapro.2020.09.002. URL <http://www.sciencedirect.com/science/article/pii/S1526612520305879>.
- Asif Ur Rehman, Muhammad Arif Mahmood, Fatih Pitir, Metin Uymaz Salamci, Andrei C. Popescu, and Ion N. Mihailescu. Keyhole Formation by Laser Drilling in Laser Powder Bed Fusion of Ti6Al4V Biomedical Alloy: Mesoscopic Computational Fluid Dynamics Simulation versus Mathematical Modelling Using Empirical Validation. *Nanomaterials*, 11(12):3284, December 2021. ISSN 2079-4991. doi: 10.3390/nano11123284. URL <https://www.mdpi.com/2079-4991/11/12/3284>.
- Bram van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics*, 14(4):361–370, March 1974. ISSN 0021-9991. doi: 10.1016/0021-9991(74)90019-9. URL <http://www.sciencedirect.com/science/article/pii/0021999174900199>.
- Arturo Vargas, Thomas M. Stitt, Kenneth Weiss, Vladimir Z. Tomov, Jean-Sylvain Camier, Tzanio Kolev, and Robert N. Rieben. Matrix-free approaches for GPU acceleration of a high-order finite element hydrodynamics application using MFEM, Umpire, and RAJA. *arXiv:2112.07075 [cs]*, December 2021. URL <http://arxiv.org/abs/2112.07075>.
- G. Vastola, Q. X. Pei, and Y.-W. Zhang. Predictive model for porosity in powder-bed fusion additive manufacturing at high beam energy regime. *Additive Manufacturing*, 22:817–822, January 2018. ISSN 2214-8604. doi: 10.1016/j.addma.2018.05.042. URL <http://www.sciencedirect.com/science/article/pii/S221486041830174X>.
- Rodrigo Gómez Vázquez, Holger M. Koch, and Andreas Otto. Multi-physical Simulation of Laser Welding. *Physics Procedia*, 56:1334–1342, January 2014. ISSN 18753892. doi: 10.1016/j.phpro.2014.08.059.
- B. C. Vermeire, F. D. Witherden, and P. E. Vincent. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *Journal of Computational Physics*, 334:497–521, April 2017. ISSN 0021-9991. doi: 10.1016/j.jcp.2016.12.049. URL <https://www.sciencedirect.com/science/article/pii/S0021999116307136>.
- Henk K. Versteeg and Weeratunge Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson/Prentice Hall, Harlow, 2. ed., [nachdr.] edition, 2007. ISBN 978-0-13-127498-3.
- Miguel R Visbal and Datta V Gaitonde. On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes. *Journal of Computational Physics*,

- 181(1):155–185, September 2002. ISSN 0021-9991. doi: 10.1006/jcph.2002.7117. URL <https://www.sciencedirect.com/science/article/pii/S0021999102971172>.
- G. Gary Wang. Definition and Review of Virtual Prototyping. *Journal of Computing and Information Science in Engineering*, 2(3):232–236, September 2002. ISSN 1530-9827, 1944-7078. doi: 10.1115/1.1526508. URL <https://asmedigitalcollection.asme.org/computingengineering/article/2/3/232/450434/Definition-and-Review-of-Virtual-Prototyping>.
- Jin Wang, Yi Wang, and Jing Shi. A Novel Time Step Fusion Method with Finite Volume Formulation for Accelerated Thermal Analysis of Laser Additive Manufacturing. *International Journal of Precision Engineering and Manufacturing-Green Technology*, January 2020. ISSN 2288-6206. doi: 10.1007/s40684-020-00237-z.
- Qian Wang, Xianyi Zhang, Yunquan Zhang, and Qing Yi. AUGEM: Automatically generate high performance dense linear algebra kernels on x86 CPUs. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–12, Denver Colorado, November 2013. ACM. ISBN 978-1-4503-2378-9. doi: 10.1145/2503210.2503219. URL <https://dl.acm.org/doi/10.1145/2503210.2503219>.
- T.C Warburton and G.E Karniadakis. A Discontinuous Galerkin Method for the Viscous MHD Equations. *Journal of Computational Physics*, 152(2):608–641, July 1999. ISSN 00219991. doi: 10.1006/jcph.1999.6248. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999199962484>.
- Noriko Watari, Yuzuru Ogura, Noriko Yamazaki, Yukihiko Inoue, Keisuke Kamitani, Yasuyuki Fujiya, Masahiko Toyoda, Saneyuki Goya, and Toshiya Watanabe. Two-fluid model to simulate metal powder bed fusion additive manufacturing. *Journal of Fluid Science and Technology*, 13(2):JFST0010–JFST0010, 2018. doi: 10.1299/jfst.2018jfst0010.
- H. L. Wei, Y. Cao, W. H. Liao, and T. T. Liu. Mechanisms on inter-track void formation and phase transformation during laser Powder Bed Fusion of Ti-6Al-4V. *Additive Manufacturing*, 34:101221, January 2020. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101221.
- Pei Wei, Zhengying Wei, Zhne Chen, Jun Du, Yuyang He, and Junfeng Li. Fundamentals of radiation heat transfer in AlSi10Mg powder bed during selective laser melting. *Rapid Prototyping Journal*, 25(9):1506–1515, January 2018. ISSN 1355-2546. doi: 10.1108/RPJ-11-2016-0189. URL <https://doi.org/10.1108/RPJ-11-2016-0189>.
- E Weinan. *Principles of Multiscale Modeling*. Cambridge University Press, Cambridge ; New York, 2011. ISBN 978-1-107-09654-7.
- E Weinan and Jianfeng Lu. Multiscale modeling. *Scholarpedia*, 6(10):11527, 2011. ISSN 1941-6016. doi: 10.4249/scholarpedia.11527. URL [http://www.scholarpedia.org/article/Multiscale\\_modeling](http://www.scholarpedia.org/article/Multiscale_modeling).

- Johannes Weirather, Vladyslav Rozov, Mario Wille, Paul Schuler, Christian Seidel, Nikolaus A. Adams, and Michael F. Zaeh. A Smoothed Particle Hydrodynamics Model for Laser Beam Melting of Ni-based Alloy 718. *Computers & Mathematics with Applications*, 78(7):2377–2394, October 2019. ISSN 0898-1221. doi: 10.1016/j.camwa.2018.10.020. URL <https://www.sciencedirect.com/science/article/pii/S0898122118306175>.
- Scott Wells and Matthew John M. Krane. Uncertainty Quantification of Model Predictions Due to Fluid Flow in Laser Powder Bed Fusion of IN625. In *TMS 2022 151st Annual Meeting & Exhibition Supplemental Proceedings*, The Minerals, Metals & Materials Series, pages 1054–1063, Cham, 2022. Springer International Publishing. ISBN 978-3-030-92381-5. doi: 10.1007/978-3-030-92381-5\_101.
- Scott Wells, Alex Plotkowski, and Matthew John M. Krane. Propagation of Input Uncertainties in Numerical Simulations of Laser Powder Bed Fusion. *Metallurgical and Materials Transactions B*, June 2021. ISSN 1073-5615, 1543-1916. doi: 10.1007/s11663-021-02218-2. URL <https://link.springer.com/10.1007/s11663-021-02218-2>.
- Wiesner, Leonard. Multiphysics-Simulationen in der Prototypenentwicklung: Fortschritte beim Digital Prototyping und dem Digitalen Zwingling. Seminar paper, University of Augsburg, Faculty of Applied Computer Science, July 2023.
- Andreas Wimmer, Baturay Yalvac, Christopher Zoeller, Fabian Hofstaetter, Stefan Adami, Nikolaus A. Adams, and Michael F. Zaeh. Experimental and Numerical Investigations of In Situ Alloying during Powder Bed Fusion of Metals Using a Laser Beam. *Metals*, 11(11):1842, November 2021. doi: 10.3390/met11111842. URL <https://www.mdpi.com/2075-4701/11/11/1842>.
- Florian Wirth, Teresa Tonn, Markus Schöberl, Stefan Hermann, Hannes Birkhofer, and Vasily Ploshikhin. Implementation of the Marangoni effect in an open-source software environment and the influence of surface tension modeling in the mushy region in laser powder bed fusion (LPBF). *Modelling and Simulation in Materials Science and Engineering*, 30(3):034001, January 2022. ISSN 0965-0393. doi: 10.1088/1361-651X/ac4a26. URL <https://doi.org/10.1088/1361-651x/ac4a26>.
- D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997. ISSN 1089778X. doi: 10.1109/4235.585893. URL <http://ieeexplore.ieee.org/document/585893/>.
- Chaochao Wu, Muhammad Qasim Zafar, and Haiyan Zhao. Numerical investigation of consolidation mechanism in powder bed fusion considering layer characteristics during multilayer process. *The International Journal of Advanced Manufacturing Technology*, February 2021. ISSN 1433-3015. doi: 10.1007/s00170-021-06768-2. URL <https://doi.org/10.1007/s00170-021-06768-2>.

- Yu-Che Wu, Weng-Sing Hwang, Cheng-Hung San, Chih-Hsiang Chang, and Huey-Jiuan Lin. Parametric study of surface morphology for selective laser melting on Ti6Al4V powder bed with numerical and experimental methods. *International Journal of Material Forming*, 11(6):807–813, November 2018. ISSN 1960-6214. doi: 10.1007/s12289-017-1391-2. URL <https://doi.org/10.1007/s12289-017-1391-2>.
- Ziheng Wu, Guannan Tang, Samuel Clark, Andrey Meshkov, Subhrajit Roychowdhury, Benjamin Gould, Victor Ostroverkhov, Thomas Adcock, Steven Duclos, Kamel Fezzaa, Christopher Immer, and Anthony Rollett. A study of high frequency laser beam oscillation induced keyhole dynamics in laser powder bed fusion additive manufacturing process. Preprint, In Review, August 2022. URL <https://www.researchsquare.com/article/rs-1879321/v1>.
- Mujian Xia, Nianlian Li, Man Zhang, Donghua Dai, Yuebin Lin, Aihui Liu, Xiaogang Chen, and Hongyan Ding. Multi-material model for mesoscopic analysis of porosity evolution during laser powder-bed fusing TiNbTa powder mixture. *Computational Materials Science*, 198:110674, October 2021. ISSN 0927-0256. doi: 10.1016/j.commatsci.2021.110674. URL <https://www.sciencedirect.com/science/article/pii/S0927025621004018>.
- Yinhua Xia, Yan Xu, and Chi-Wang Shu. Local discontinuous Galerkin methods for the Cahn–Hilliard type equations. *Journal of Computational Physics*, 227(1):472–491, November 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.08.001. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999107003439>.
- Zhang Xianyi, Wang Qian, and Zhang Yunquan. Model-driven Level 3 BLAS Performance Optimization on Loongson 3A Processor. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pages 684–691, December 2012. doi: 10.1109/ICPADS.2012.97.
- Ziyu Xie, Wen Jiang, Congjian Wang, and Xu Wu. Bayesian Inverse Uncertainty Quantification of a MOOSE-based Melt Pool Model for Additive Manufacturing Using Experimental Data. *arXiv:2105.05370 [stat]*, May 2021. URL <http://arxiv.org/abs/2105.05370>.
- Sergey Yakovlev, David Moxey, Robert M. Kirby, and Spencer J. Sherwin. To CG or to HDG: A Comparative Study in 3D. *Journal of Scientific Computing*, 67(1): 192–220, April 2016. ISSN 1573-7691. doi: 10.1007/s10915-015-0076-6. URL <https://doi.org/10.1007/s10915-015-0076-6>.
- Jinhui Yan, Qiming Zhu, and Ze Zhao. Computational thermal multi-phase flow for metal additive manufacturing, June 2022. URL <http://arxiv.org/abs/2206.04799>.
- F. W. Yang, C. E. Goodyer, M. E. Hubbard, and P. K. Jimack. An optimally efficient technique for the solution of systems of nonlinear parabolic partial differential equations. *Advances in Engineering Software*, 103:65–84, January 2017. ISSN 0965-

9978. doi: 10.1016/j.advensoft.2016.06.003. URL <https://www.sciencedirect.com/science/article/pii/S0965997816301259>.
- Min Yang, Lu Wang, and Wentao Yan. Phase-field modeling of grain evolutions in additive manufacturing from nucleation, growth, to coarsening. *npj Computational Materials*, 7(1):56, December 2021. ISSN 2057-3960. doi: 10.1038/s41524-021-00524-6. URL <http://www.nature.com/articles/s41524-021-00524-6>.
- Yangyiwei Yang, Timileyin David Oyedeji, Xiandong Zhou, Karsten Albe, and Bai-Xiang Xu. Tailoring magnetic hysteresis of additive manufactured Fe-Ni permalloy via multiphysics-multiscale simulations of process-property relationships. *npj Computational Materials*, 9(1):1–19, June 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01058-9. URL <https://www.nature.com/articles/s41524-023-01058-9>.
- Zerong Yang, Matthias Markl, and Carolin Körner. Predictive simulation of bulk metallic glass crystallization during laser powder bed fusion. *Additive Manufacturing*, 59: 103121, November 2022. ISSN 2214-8604. doi: 10.1016/j.addma.2022.103121. URL <https://www.sciencedirect.com/science/article/pii/S2214860422005103>.
- Zhuo Yang, Yan Lu, H. Yeung, and Sundar Kirishnamurty. 3D Build Melt Pool Predictive Modeling for Powder Bed Fusion Additive Manufacturing. *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, January 2020.
- Dengzhi Yao, Xizhong An, Haitao Fu, Hao Zhang, Xiaohong Yang, Qingchuan Zou, and Kejun Dong. Dynamic investigation on the powder spreading during selective laser melting additive manufacturing. *Additive Manufacturing*, page 101707, November 2020. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101707. URL <http://www.sciencedirect.com/science/article/pii/S2214860420310794>.
- M. Reza Yavari, Kevin D. Cole, and Prahalada Rao. Thermal Modeling in Metal Additive Manufacturing Using Graph Theory. *Journal of Manufacturing Science and Engineering*, 141(7), January 2019. ISSN 1087-1357. doi: 10.1115/1.4043648.
- Wenlin Ye, Shanshan Zhang, Lazaro Lopez Mendez, Mathew Farias, Jianzhi Li, Ben Xu, Peisheng Li, and Ying Zhang. Numerical simulation of the melting and alloying processes of elemental titanium and boron powders using selective laser alloying. *Journal of Manufacturing Processes*, 64:1235–1247, April 2021. ISSN 1526-6125. doi: 10.1016/j.jmapro.2021.02.044. URL <https://www.sciencedirect.com/science/article/pii/S1526612521001304>.
- Xiu Ye. On the relationship between finite volume and finite element methods applied to the Stokes equations. *Numerical Methods for Partial Differential Equations*, 17(5):440–453, 2001. ISSN 1098-2426. doi: 10.1002/num.1021. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/num.1021>.

- Weihao Yuan, Hui Chen, Tan Cheng, and Qingsong Wei. Effects of laser scanning speeds on different states of the molten pool during selective laser melting: Simulation and experiment. *Materials & Design*, page 108542, January 2020. ISSN 0264-1275. doi: 10.1016/j.matdes.2020.108542. URL <http://www.sciencedirect.com/science/article/pii/S0264127520300757>.
- Fan Yurun. A comparative study of the discontinuous Galerkin and continuous SUPG finite element methods for computation of viscoelastic flows. *Computer Methods in Applied Mechanics and Engineering*, 141(1):47–65, February 1997. ISSN 0045-7825. doi: 10.1016/S0045-7825(96)01102-4. URL <https://www.sciencedirect.com/science/article/pii/S0045782596011024>.
- Jing Zhang, Yi Zhang, Weng Hoh Lee, Linmin Wu, Hyun-Hee Choi, and Yeon-Gil Jung. A multi-scale multi-physics modeling framework of laser powder bed fusion additive manufacturing process. *Metal Powder Report*, 73(3):151–157, January 2018a. ISSN 0026-0657. doi: 10.1016/j.mprp.2018.01.003. URL <http://www.sciencedirect.com/science/article/pii/S0026065718300031>.
- Wei Zhang, Zhenguo Zhang, and Xiaofei Chen. Three-dimensional elastic wave numerical modelling in the presence of surface topography by a collocated-grid finite-difference method on curvilinear grids. *Geophysical Journal International*, 190(1): 358–378, August 2012. ISSN 0956-540X. doi: 10.1111/j.1365-246X.2012.05472.x. URL <https://doi.org/10.1111/j.1365-246X.2012.05472.x>.
- Yancheng Zhang, Qiang Chen, Gildas Guillemot, Charles-André Gandin, and Michel Bellet. Numerical modelling of fluid and solid thermomechanics in additive manufacturing by powder-bed fusion: Continuum and level set formulation applied to track- and part-scale simulations. *Comptes Rendus Mécanique*, 346(11):1055–1071, January 2018b. ISSN 16310721. doi: 10.1016/j.crme.2018.08.008.
- Yaqi Zhang, Vadim Shapiro, and Paul Witherell. Towards Thermal Simulation of Powder Bed Fusion on Path Level. In *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, November 2019. doi: 10.1115/DETC2019-98046. URL <https://dx.doi.org/10.1115/DETC2019-98046>.
- Yi Zhang and Jing Zhang. Modeling of solidification microstructure evolution in laser powder bed fusion fabricated 316L stainless steel using combined computational fluid dynamics and cellular automata. *Additive Manufacturing*, 28:750–765, January 2019. ISSN 2214-8604. doi: 10.1016/j.addma.2019.06.024.
- Tie Zhou, Yinfan Li, and Chi-Wang Shu. Numerical Comparison of WENO Finite Volume and Runge–Kutta Discontinuous Galerkin Methods. *Journal of Scientific Computing*, 16(2):145–171, June 2001. ISSN 1573-7691. doi: 10.1023/A:1012282706985. URL <https://doi.org/10.1023/A:1012282706985>.

- Jonas Zielinski, Simon Vervoort, Hans-Wilfried Mindt, and Mustafa Megahed. Influence of Powder Bed Characteristics on Material Quality in Additive Manufacturing. *BHM Berg- und Hüttenmännische Monatshefte*, 162(5):192–198, January 2017. ISSN 1613-7531. doi: 10.1007/s00501-017-0592-9.
- Olgierd C. Zienkiewicz, Robert L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier, Amsterdam Heidelberg, 6. ed., reprint., transferred to digital print edition, 2010. ISBN 978-0-7506-6320-5.
- Patrick Zimbrod and Johannes Schilp. Modelling of microstructures during in-situ alloying in additive manufacturing for efficient material qualification processes. *Simulation in Produktion und Logistik 2021: 19. ASIM-Fachtagung Simulation in Produktion und Logistik, Erlangen, 15. – 17. September 2021*, pages 177–188, 2021. URL <https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/docId/87973>.
- Patrick Zimbrod, Magdalena Schreter, and Johannes Schilp. Efficient Simulation of Complex Capillary Effects in Advanced Manufacturing Processes using the Finite Volume Method. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6, November 2022. doi: 10.1109/ICECCME55909.2022.9988504.
- Patrick Zimbrod, Michael Fleck, and Johannes Schilp. An Application-Driven Method for Assembling Numerical Schemes for the Solution of Complex Multiphysics Problems. *Applied System Innovation*, 7(3):35, June 2024. ISSN 2571-5577. doi: 10.3390/asi7030035. URL <https://www.mdpi.com/2571-5577/7/3/35>.

# Appendix A

## Literature Research for Mesoscopic PBF-LB/M Simulation

Some works employ or compare multiple methods and are thus counted more than once. In total, 177 publications were considered that have been published mainly within the last 15 years. Daggers<sup>†</sup> denote mesh-based (eulerian) approaches, asterisks<sup>\*</sup> signify mesh-free (lagrangian) methods. An exception is given by the Arbitrary Lagrangian-Eulerian method, which implements both material and field viewpoints.

Table A.1: Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M.

Numerical Scheme	Number of Publications	
	Absolute	Relative [%]
Arbitrary Lagrangian Eulerian <sup>†</sup> (Berry et al., 2021; David K. Kafui, Colin Thornton, and Jonathan P.K, 2023; Fan et al., 2020; Khairallah et al., 2020, 2015, 2016; Khairallah and Anderson, 2014; King et al., 2015a; Li et al., 2023; Martin et al., 2019)	10	5.6

Continued on next page

Table A.1: Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M. (Continued)

Discrete Element Method <sup>*</sup> (Aggarwal et al., 2020; Aminnia et al., 2022b; Boukouvala et al., 2013; Chen et al., 2022; Chien et al., 2021; Estupinan Donoso and Peters, 2018; Geer et al., 2018; Gu et al., 2021; He et al., 2020b,c; Kovalev et al., 2020; Marchais et al., 2021; Mikami et al., 1998; Xia et al., 2021; Ninpetch et al., 2023; Peters, 2013; Ramesh Sagar et al., 2021; Shenouda and Hoff, 2020; Si et al., 2021; Tan et al., 2018; Tian et al., 2020; Ur Rehman et al., 2021; Wei et al., 2018; Wu et al., 2021; Lee and Zhang, 2016; Yao et al., 2020; Yuan et al., 2020; Zhang et al., 2019)	28	15.8
Finite Difference Method <sup>†</sup> (Gusarov et al., 2009; Hosseinzadeh et al., 2022; Wu et al., 2018)	3	1.7

Continued on next page

Table A.1: Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M. (Continued)

FEM <sup>†</sup>	32	18.1
<p>(Ahsan et al., 2022; Alexopoulou et al., 2022; Allen, 2021; Arbaoui et al.; Bayat et al., 2019; Beghini et al., 2021; Burkhardt et al., 2022; Caboussat et al., 2023; Cao et al., 2021; Chen et al., 2023b, 2017; Denlinger et al., 2017; Elahi et al., 2022; Fan and Yan, 2020; Jin et al., 2021; Keller et al., 2017; Kopp et al., 2022; Ladani et al., 2017; Lee et al., 2017; Liang et al., 2021; Lin et al., 2020; Liu et al., 2020; Mandal et al., 2022; Mayi et al., 2021; Mede et al., 2020; Meier et al., 2021; Pantawane et al., 2020; Saadlaoui et al., 2021; Strayer et al., 2022; Vastola et al., 2018; Xie et al., 2021; Yan et al., 2022; Yang et al., 2023; Ye et al., 2021; Zhang et al., 2018a,b)</p>		

Continued on next page

Table A.1: Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M. (Continued)

Finite Volume Method <sup>†</sup>	85	48.0
(Acharya et al., 2017; Chien et al., 2021; Hedreen, 2020; He et al., 2020a; Hong et al., 2021; Le et al., 2020b; Li and Tan, 2020; Luo et al., 2021; Noskov et al., 2020; Rauniyar and Chou, 2019; Ren et al., 2021; Ridolfi et al., 2020b,a; Shrestha and Chou, 2017; Staroselsky et al., 2020; Zhang and Zhang, 2019; Alphonso et al.; Bayat et al., 2020; Chen et al., 2021, 2023a; Chen and Yan, 2020; Fan and Yan, 2020; Ge et al., 2021; Giam et al., 2023; Gu et al., 2020; Lee and Zhang, 2015; Li et al., 2020a; Lu et al., 2021; Ninpetch et al., 2021; Shrestha and Chou, 2021; Shrestha et al., 2019; Tian et al., 2020; Wei et al., 2020; Lee and Zhang, 2016; Yuan et al., 2020; Zielinski et al., 2017; Watari et al., 2018; Aminnia et al., 2022a; Cao, 2021, 2020a,b,c; Cao and Guan, 2021; Cao, 2020d; Cardiff et al., 2018; Chouhan et al., 2018; Gürtler et al., 2014, 2013; Hojjatzadeh et al., 2019; Katinas and Shin, 2020; Khomenko et al., 2021; Koch et al., 2010; Le et al., 2020a; Li et al., 2020b; Otto et al., 2012, 2011; Pakhale, Vinit Vikas, 2023; Panwisawas et al., 2017; Priya et al., 2020; Qiu et al., 2015; Rolchigo et al., 2022; Saufi et al., 2020; Sun et al., 2020; Svenungsson et al., 2017; Tang et al., 2022; Tang; Tan et al., 2018; Vázquez et al., 2014; Wei et al., 2020; Wells et al., 2021; Wells and Krane, 2022; Wirth et al., 2022; Zimbrod et al., 2022; Sabau et al., 2020)		

Continued on next page

Table A.1: Quantitative summary of previous works on simulating melt pool evolution in PBF-LB/M. (Continued)

Lattice Boltzmann Method*	10	5.6
(Körner et al., 2013, 2011; Krzyzanowski and Svyetlichnyy, 2021; Küng et al., 2021; Li et al., 2016; Markl et al., 2019; Rai et al., 2017; Rausch et al., 2017; Wu et al., 2022; Yang et al., 2022)		
Smoothed Particle Hydrodynamics*	10	5.6
(Afrasiabi et al., 2021; Dao and Lou, 2021; Dietemann et al., 2022; Fan et al., 2020; Fürstenau et al., 2019, 2021; Lüthi et al., 2023; Ma et al., 2023; Meier et al., 2020, 2021; Stubblefield et al., 2021; Weirather et al., 2019; Wimmer et al., 2021)		

Commercial software is annotated with an asterisk <sup>\*</sup>, academic and open source software with a dagger <sup>†</sup>.

Table A.2: Quantitative summary of previous works that use the FVM to resolve melt pool dynamics in PBF-LB/M.

Software	Number of Publications	
	Absolute	Relative [%]
ANSYS <sup>*</sup> (Acharya et al., 2017; Chien et al., 2021; Hedreen, 2020; He et al., 2020a; Hong et al., 2021; Le et al., 2020b; Li and Tan, 2020; Luo et al., 2021; Noskov et al., 2020; Rauniyar and Chou, 2019; Ren et al., 2021; Ridolfi et al., 2020b,a; Shrestha and Chou, 2017; Staroselsky et al., 2020; Zhang and Zhang, 2019)	16	21.6
FLOW-3D <sup>*</sup> (Alphonso et al.; Bayat et al., 2020; Chen et al., 2021, 2023a; Chen and Yan, 2020; Fan and Yan, 2020; Ge et al., 2021; Giam et al., 2023; Gu et al., 2020; Lee and Zhang, 2015; Li et al., 2020a; Lu et al., 2021; Ninpetch et al., 2021; Shrestha and Chou, 2021; Shrestha et al., 2019; Tian et al., 2020; Wei et al., 2020; Lee and Zhang, 2016; Yuan et al., 2020; Zielinski et al., 2017)	20	26.3
FrontFlow <sup>†</sup> (Watari et al., 2018)	1	1.4

Continued on next page

Table A.2: Quantitative summary of previous works that use the FVM to resolve melt pool dynamics in PBF-LB/M. (Continued)

OpenFOAM <sup>†</sup> (Aminnia et al., 2022a; Cao, 2021, 2020a,b,c; Cao and Guan, 2021; Cao, 2020d; Cardiff et al., 2018; Chouhan et al., 2018; Gürtler et al., 2014, 2013; Hojjatzadeh et al., 2019; Katinas and Shin, 2020; Khomenko et al., 2021; Koch et al., 2010; Le et al., 2020a; Li et al., 2020b; Otto et al., 2012, 2011; Pakhale, Vinit Vikas, 2023; Panwisawas et al., 2017; Priya et al., 2020; Qiu et al., 2015; Rolchigo et al., 2022; Saufi et al., 2020; Sun et al., 2020; Svenungsson et al., 2017; Tang et al., 2022; Tang; Tan et al., 2018; Vázquez et al., 2014; Wei et al., 2020; Wells et al., 2021; Wells and Krane, 2022; Wirth et al., 2022; Zimbrod et al., 2022)	37	48.7
Truchas <sup>†</sup> (Sabau et al., 2020)	1	1.4



# Appendix B

## Implementation Details of the Presented Models

### B.1 Advection Equation

#### B.1.1 Derivation of the Weak Form

Equation 7.19 can be obtained by the strong form of the PDE as described in the article, by applying partial integration after multiplying with a test function and integrating over the domain  $\Omega$ . To arrive at a formulation that can be implemented, there are a few additional steps which are described below.

In most DG codes, the weak form given by Equation 7.19, more specifically the surface integral cannot be entered directly but needs to be reformulated using jump and average operators. This is also done in the advection equation example described in section 7.2. Such a modified weak form can be derived as follows.

The most straightforward consists of summarizing the convective term of the PDE as the divergence of some flux  $\nabla \cdot \underline{f}(\alpha)$ . In this case, the flux is simply  $\underline{f}(\alpha) = \underline{u}\phi$ .

Then, the surface integral of Equation 7.19 can be formulated as:

$$\int_{\partial\Omega} v\alpha(\underline{u} \cdot \underline{n}) \, dS = \int_{\partial\Omega} v\underline{f}(\alpha) \cdot \underline{n} \, dS \quad (\text{B.1})$$

One may now write down the terms of the surface integral of Equation 7.19 that contribute to an arbitrary internal facet, that is, there are always two integrals from the cells that the facet is owned by. The two cells are distinguished by introducing the notation

”+” and ”-”:

$$\int_{\partial\Omega} v_h^+ \alpha_h^+ (\underline{u}^+ \cdot \underline{n}^+) dS + \int_{\partial\Omega} v_h^- \alpha_h^- (\underline{u}^- \cdot \underline{n}^-) dS \quad (\text{B.2})$$

Where the yet-to-be-determined values for  $\alpha^+$  and  $\alpha^-$  are introduced. By introducing the common, yet unknown numerical flux  $\underline{f}(\alpha)^*$  and utilizing the fact that the facet normal vectors  $\underline{n}$  point opposite to each other, Equation B.2 becomes:

$$\int_{\partial\Omega} \left( \underline{f}(\alpha)^* \cdot \underline{n} \right) (v^+ - v^-) dS = \int_{\partial\Omega} \left( \underline{f}(\alpha)^* \cdot \underline{n} \right) \llbracket v \rrbracket dS \quad (\text{B.3})$$

The only issue that is left to deal with is now the choice of an appropriate expression for the numerical flux  $\underline{f}(\alpha)^* \cdot \underline{n}$ .

Both DG and FV implementations of the advection equation use the Lax-Friedrichs flux function:

$$f(\alpha)^* = \{ \alpha (\underline{u} \cdot \underline{n}^+) \} + \frac{1}{2} C \llbracket \alpha \rrbracket \quad (\text{B.4})$$

Where  $\{ \cdot \}$  and  $\llbracket \cdot \rrbracket$  denote the average and jump operators as defined above. In this case,  $C$  is the maximum velocity in the domain and takes the role of the maximum signal velocity, which must be computed separately for more complex problems. It is defined as the jacobian of the PDE flux  $\partial f(\alpha) / \partial \alpha$ . Since that  $f(\alpha) = \underline{u} \alpha$ , the jacobian simply becomes the prescribed advection velocity  $\underline{u}$ .

### B.1.2 Simulation Parameters

The resulting semi-discrete ODE systems are solved using a three-stage implicit Runge Kutta formulation using a matrix-free solver.

The entire simulation runs are timed from the command line and each is executed six times. Wall times per time step were recorded during the run using the PETSc distributed interface and logged to a text file.