Johannes Gutenberg University Mainz

Department 08 – Physics, Mathematics and Computer Science

Institute of Geosciences

Machine Learning for Predicting Flow Fields Around Settling Crystals

Generalization Capability of UNet/FNO Architectures for Multi-Crystal Sedimentation Systems

Master's Thesis

submitted in partial fulfillment of the requirements

for the degree of

Master of Science (M.Sc.)

in

Computational Sciences

submitted by

Paul Baselt

Matriculation Number: 2706608

born in Berlin

First Supervisor:
Univ.-Prof. Dr. Boris J. P. Kaus
Institute of Geosciences
Johannes Gutenberg-Universität Mainz

Second Supervisor:
Prof. Dr. Michael Wand
Institute of Computer Science
Johannes Gutenberg-Universität Mainz

Mainz, February 20, 2026

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Arbeiten entnommen sind, habe ich als solche kenntlich gemacht.

Die Arbeit ist in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt worden.

Ort, Datum	Unterschrift

Zusammenfassung

Abstract (English)

This master's thesis investigates the generalization capability of UNet architectures for predicting flow fields in multi-crystal sedimentation systems. While traditional computational fluid dynamics (CFD) methods are computationally intensive for complex multi-particle systems, machine learning approaches offer the potential for significant acceleration.

This exploratory study systematically evaluates whether UNet networks trained on a fixed number of crystals (e.g., 10 crystals) can successfully generalize to arbitrary other crystal configurations (1–15 crystals). Using LaMEM simulations as ground truth, controlled experiments are conducted where the model trained on N crystals is evaluated across the entire spectrum from 1 to N crystals. The results demonstrate [placeholder for actual findings], thereby characterizing both the possibilities and fundamental limitations of UNet generalization across different particle numbers.

This work contributes methodologically to physics-informed machine learning research and provides practical insights for geoscientific modeling applications.

Contents

Eidesstattliche Erklärung III

Zusammenfassung V

Abstract (English) VII

I List of Figures XIII

II List of Tables XV

III Listings XVII

1 Introduction and Research Motivation 1

1.1 Geoscientific Context 1
1.2 Computational Challenges 2
1.3 Machine Learning as a Surrogate Modeling Strategy 2
1.4 Research Question and Hypothesis 3
1.5 Objectives and Scope of the Study 4

2 State of the Art 7

2.1 Deep Learning for Fluid Dynamics 7
2.2 Physics-Informed and Hybrid Approaches 8
2.3 Neural Operators and Generalization Across Geometries 8
2.4 Machine Learning for Particle- and Crystal-Laden Flows 9
2.5 Comparison of Network Paradigms for Flow-Field Surrogates 10
2.5.1 Coordinate-Based Implicit Models 10
2.5.2 Linear Reduced-Order Models 10
2.5.3 Grid-Based Convolutional Surrogates 10
2.5.4 Spectral Neural Operators 11
2.5.5 Implications for This Work 11
2.6 Position of This Thesis in the Research Landscape 11

3 Theory and Physical Background 13

3.1 Incompressible Stokes Flow at Low Reynolds Number 13
3.2 Stream-Function Formulation in Two Dimensions 14

3.3 Vorticity and Poisson Reconstruction of the Stream Function 14
3.4 Learning Targets as Physically Motivated Inductive Biases 15
3.5 U-Net Architecture 16
3.5.1 Encoder Path 16
3.5.2 Decoder Path and Skip Connections 16
3.5.3 Multiscale Feature Extraction 16
3.6 Fourier Neural Operator 17
3.6.1 Fourier Layer 17
3.6.2 Architecture 17
3.6.3 Properties Relevant to Stokes Flow 17
3.7 Architectural Comparison: U-Net vs. FNO 18
3.8 Implications for Multi-Crystal Sedimentation Modeling 18

4 Methodology **21**
4.1 Learning Target: Stream-Function Prediction 21
4.2 LaMEM Simulations for Data Generation 22
4.2.1 Domain and Discretization 22
4.2.2 Material Properties and Governing Equations 22
4.2.3 Extraction of Flow Quantities 23
4.3 Computing and Normalizing the Stream Function 23
4.3.1 Discretization and Boundary Conditions 23
4.3.2 Motivation and Procedure for Normalization 24
4.4 Input Representation: Mask, Distance Field, and Coordinates 24
4.5 Surrogate Architectures 25
4.5.1 U-Net 25
4.5.2 Fourier Neural Operator 26
4.5.3 Regularization and Practical Considerations 27
4.6 Training Procedure 27
4.6.1 Data Splits and Benchmark Design 27
4.6.2 Batching and Device Handling 27
4.6.3 Loss Function 27
4.6.4 Optimization and Checkpointing 28
4.7 Evaluation Pipeline 28
4.8 Summary 28

5 Implementation **31**
5.1 Software Environment 31
5.2 Code Organization and Execution Flow 32
5.3 Model Implementations 32
5.4 Data Pipeline and Dataset Interfaces 33

5.5 Stream-Function Reconstruction and Normalization 34
5.6 Training Loop and Optimization 34
5.7 Evaluation Utilities 35
5.8 Summary 35

6 Experiments **37**
6.1 Experimental Setup 37
6.1.1 Learning task and reconstruction of velocity 37
6.1.2 Error metrics 38
6.2 Single–Crystal Hyperparameter Study 39
6.2.1 Overview of explored settings 39
6.2.2 Best configuration for one crystal 39
6.2.3 Trends and practical takeaways 40
6.3 Two–Crystal Evaluation 40
6.3.1 Aggregate performance for $N = 1$ vs. $N = 2$ 41
6.3.2 Spatial error dependence: distance to center and corners 41
6.4 Discussion 42
6.4.1 What works well 42
6.4.2 Where errors concentrate and why gradients are harder 42
6.4.3 Implications for the multi-crystal goal 42
6.5 Summary 42

7 Results **45**

8 Validation **47**

9 Conclusion and Outlook **49**

10 Discussion **51**

A Bibliography **53**

I List of Figures

II List of Tables

3.1 Structural comparison of U-Net and FNO architectures. 18

6.1 Representative single–crystal results (Approach 2). 40
6.2 Grouped test errors for one- and two-crystal configurations (100 samples each). 41

III Listings

1 Introduction and Research Motivation

1.1 Geoscientific Context

Crystal sedimentation in viscous magmatic fluids is a fundamental process governing the evolution, differentiation, and internal dynamics of magma reservoirs. During cooling, crystals nucleate and grow within the melt and begin to settle due to density contrasts between the solid and liquid phases. This settling process directly influences the thermal and chemical evolution of the system and contributes to the formation of cumulate layers that are preserved as macroscopic structures in plutonic bodies.

Early work by Daniel Martin and Roger Nokes (1988) demonstrated that crystal settling can occur even in vigorously convecting magma chambers, provided that the Stokes settling velocity exceeds characteristic convective flow velocities. This result challenged the classical assumption that convection necessarily suppresses sedimentation and emphasized the role of local particle–fluid interactions. A settling crystal perturbs the surrounding flow field by generating viscous boundary layers and wake structures that extend over several crystal diameters. These flow perturbations are not confined to the immediate vicinity of a single particle but can modify the trajectories and settling behavior of neighboring crystals.

Subsequent experimental and numerical investigations (D. Martin and R. Nokes 1989; Weinstein, Yuen, and Olson 1988; Uhlmann and Doychev 2014; Penlou et al. 2023; Nissanka, Ma, and Burton 2023) revealed that such interactions may lead to asymmetric flow patterns, recirculation zones, and clustering phenomena. As a consequence, even small variations in crystal geometry, spatial arrangement, or relative positioning can strongly affect collective settling dynamics and the development of mineralogical layering. These findings highlight that multi-crystal sedimentation is inherently a multiscale and interaction-dominated process, which places stringent demands on numerical modeling approaches intended to capture its essential physics.

1.2 Computational Challenges

Accurate numerical simulation of multi-crystal sedimentation remains computationally demanding due to the strongly multiscale nature of crystal fluid interactions. A physically faithful model must simultaneously resolve:

- **Microscale crystal boundaries**, where steep velocity gradients and localized shear layers develop,
- **Long-range hydrodynamic interactions** between multiple crystals mediated by the surrounding fluid, and
- **Domain-scale flow structures** that control the large-scale transport of crystals and melt.

The computational cost of resolving hydrodynamic interactions typically scales between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$ with the number of particles N (Ladd 1994). Numerical methods that explicitly couple particle and fluid dynamics—such as lattice Boltzmann discrete element methods (LBM-DEM) or immersed boundary approaches—generally require each particle to be resolved by tens to hundreds of grid cells. Even in two-dimensional configurations, this leads to simulations with 10^5-10^6 fluid nodes per particle (X. Li et al. 2022; Leonardi et al. 2014). For systems containing more than ten crystals, total grid sizes can easily exceed 10^7–10^8 degrees of freedom, resulting in runtimes ranging from several hours to multiple days per simulation (X. Li et al. 2022).

Such computational demands render systematic parameter studies infeasible. Investigating the influence of crystal size, viscosity contrast, particle number, or spatial arrangement would require thousands of high-fidelity simulations and thus exceed practical computational budgets. This limitation motivates the development of surrogate models capable of approximating the underlying flow fields at a substantially reduced computational cost while retaining sufficient physical fidelity.

1.3 Machine Learning as a Surrogate Modeling Strategy

In recent years, machine learning (ML) has emerged as a promising approach for accelerating the prediction of fluid-dynamical systems. Two architectural paradigms are of particular relevance to this work.

First, U-Net architectures (Ronneberger, Fischer, and Brox 2015) leverage an encoder–decoder structure with skip connections to capture multiscale spatial features. Their ability to encode complex geometric information at multiple resolutions has made them a widely adopted choice for structured flow-field prediction (Thuerey et al. 2020; Ribeiro et al. 2021; J. Chen, Viquerat, and Hachem 2019).

Second, Fourier Neural Operators (FNOs) (Z. Li et al. 2021) learn mappings between function spaces by performing convolutions in the spectral domain via the Fast Fourier Transform. By operating in frequency space, FNOs can efficiently capture global, long-range dependencies in the solution field—a property that is particularly attractive for problems governed by elliptic operators such as the Stokes equations.

Both architectures have demonstrated speed-ups of several orders of magnitude compared to classical solvers while maintaining acceptable accuracy in the context of computational fluid dynamics. Beyond purely data-driven approaches, physics-informed neural networks (PINNs) (Raissi, Perdikaris, and Karniadakis 2019; Jin et al. 2021) incorporate governing equations directly into the training objective, thereby encouraging physically consistent predictions even in data-scarce regimes.

A comprehensive overview of recent developments—including geometry-aware PINNs, neural operators, and scalable U-Net variants for complex flow problems—is provided in Chapter 2. These advances motivate the central premise of this thesis: How well can U-Net and FNO architectures, trained on a limited number of crystal configurations, generalize to unseen sedimentation scenarios with varying particle numbers and spatial arrangements? And which architecture is better suited for this task, given their fundamentally different approaches to capturing spatial structure?

1.4 Research Question and Hypothesis

The overarching research question guiding this thesis is:

> *To what extent can U-Net and Fourier Neural Operator architectures, trained on configurations containing up to a fixed maximal number of crystals (N), generalize to unseen sedimentation scenarios with varying particle numbers and spatial arrangements—and which factors fundamentally limit this generalization capability?*

This question is of direct relevance to geoscientific applications, as natural sedimentation processes rarely involve fixed particle counts or highly regular geometries. A practically useful surrogate model must therefore generalize across a wide range of spatial configurations, interaction regimes, and crystal numbers.

We hypothesize that an ML-based surrogate model can achieve such generalization provided that:

1. the training data sufficiently sample the space of possible geometric configurations;
2. the learning task incorporates a physically meaningful target representation—such as the stream function ψ—which enforces incompressibility by construction and reduces the effective degrees of freedom of the problem;

3. the network architecture captures the inherently multiscale character of crystal–fluid interactions—whether through spatially hierarchical feature extraction (U-Net) or through global spectral representations (FNO).

The controlled comparison of U-Net and FNO architectures presented in this thesis is designed to explicitly test these hypotheses, to quantify the impact of architectural choice on predictive performance, and to identify which paradigm—local spatial convolutions or global spectral operations—is better suited for surrogate modeling of multi-crystal Stokes flow.

1.5 Objectives and Scope of the Study

The objectives of this thesis are structured around three complementary goals.

Primary Objective

Quantitatively evaluate the generalization performance of U-Net and FNO based surrogate models across varying crystal numbers (1–100) and spatial arrangements on a fixed 256×256 computational grid.

Secondary Objective

Identify and analyze systematic failure modes encountered when predicting previously unseen geometries, including symmetry-breaking artifacts, nonphysical divergence, and reduced accuracy near crystal boundaries.

Tertiary Objective

Derive practical guidelines for the construction of surrogate models for multiphase Stokes flow in geophysical settings, with a focus on input representations, output variables, normalization strategies, and training methodologies.

Scope Limitations

To maintain computational feasibility while addressing the core research questions, this study is restricted to:

- two-dimensional incompressible Stokes flow formulated in stream-function-vorticity form,
- rigid circular crystal geometries without collision or lubrication forces,

- steady-state flow fields defined on a fixed 256×256 grid,
- synthetic training data generated using the LaMEM code (Popov and Kaus 2013),
- pre-generated training, validation, and evaluation datasets stored on disk and loaded during training to ensure reproducibility and consistent benchmarking across architectures.

These assumptions allow for a focused investigation of surrogate modeling strategies while preserving the essential physical characteristics of multi-crystal sedimentation dynamics.

Structure of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 reviews recent machine-learning-based surrogate models for fluid dynamics and situates the present work within the existing literature. Chapter 3 introduces the physical and mathematical foundations of Stokes flow, the stream-function formulation, and the U-Net and FNO architectures employed for structured flow-field prediction. Chapter 4 describes the computational workflow employed in this thesis, including LaMEM data generation, stream-function computation, input construction, network architecture, and training and evaluation procedures. Chapter 5 details the technical realization of the surrogate modeling framework, covering the software environment, model implementations, data pipeline, and training infrastructure. Chapter 6 presents the experimental setup, including the choice of hyperparameters, training configurations, and evaluation metrics used throughout the study. Chapter 7 reports the numerical results obtained from the U-Net and FNO surrogate models across varying crystal numbers and spatial arrangements. Chapter 8 analyzes the predictive quality of the trained models, identifies systematic failure modes, and examines where and why errors arise. Chapter 10 discusses the overall effectiveness of the surrogate modeling approach, its limitations, and potential improvements. Finally, Chapter 9 summarizes the key findings, assesses the extent to which the research questions have been answered, and outlines directions for future work.

2 State of the Art

This chapter reviews recent developments in machine-learning-based surrogate models for fluid dynamics, with a particular emphasis on incompressible Stokes flow, flow–structure interaction, and neural architectures operating on structured grids. While classical numerical methods remain the reference standard in terms of physical fidelity, their computational cost often renders large parameter studies infeasible. This limitation has motivated the growing interest in data-driven surrogate models that aim to approximate solution operators at a fraction of the computational expense.

The literature surveyed in this chapter provides the conceptual and methodological foundation for the modeling choices made in this thesis. In particular, it highlights both the opportunities and the limitations of existing surrogate approaches when applied to problems involving complex, geometry-dependent flow fields.

2.1 Deep Learning for Fluid Dynamics

Deep neural networks have emerged as powerful tools for approximating solutions of partial differential equations, especially in fluid dynamics. Convolutional neural networks (CNNs) and encoder–decoder architectures such as the U-Net have proven particularly effective for mapping geometric input fields to structured solution fields (**navab__u-net__2015**; Thuerey et al. 2020). Their success can be attributed to a combination of local feature extraction, multiscale representation, and translation-invariant convolutional operators.

Early studies on CNN-based surrogates for steady laminar flows demonstrated that these models can reproduce high-fidelity Navier–Stokes or Stokes solutions with speed-ups of several orders of magnitude compared to traditional solvers, while maintaining acceptable accuracy (Ribeiro et al. 2021; J. Chen, Viquerat, and Hachem 2019). These results established deep learning as a viable surrogate modeling strategy for computational fluid dynamics.

Subsequent work extended these approaches to more complex geometries and boundary conditions. In particular, scalable U-Net variants for flow prediction in multi-obstacle environments showed that multiscale encoder–decoder architectures with skip connections are well suited for problems that require simultaneous resolution of sharp boundary layers and global flow structures (Thuerey et al. 2020). This combination of local and global context is directly relevant for multi-crystal sedimentation, where localized wake structures

around individual crystals interact with domain-scale circulation patterns.

2.2 Physics-Informed and Hybrid Approaches

Physics-Informed Neural Networks (PINNs) aim to incorporate physical knowledge directly into the learning process by augmenting the loss function with residuals of the governing partial differential equations (Raissi, Perdikaris, and Karniadakis 2019). Rather than relying exclusively on paired input–output data, PINNs enforce approximate satisfaction of the underlying equations at collocation points within the computational domain. For incompressible flow, this typically involves penalizing the momentum equations together with the divergence-free constraint (Jin et al. 2021).

Hybrid extensions of this idea include geometry-aware PINNs, which embed geometric information through distance functions or latent representations to improve learning on irregular domains (Oldenburg et al. 2022). While these approaches can enhance physical consistency, they often introduce additional challenges. In practice, PINNs and hybrid formulations tend to be sensitive to the relative weighting of loss terms, exhibit slow convergence, and incur significant computational overhead due to repeated evaluation of spatial derivatives.

These difficulties become particularly pronounced for creeping Stokes flow with multiple rigid inclusions, where sharp gradients near particle boundaries coexist with long-range hydrodynamic interactions. For this reason, the present thesis does not adopt a full physics-informed formulation. Instead, physical structure is incorporated implicitly through the choice of target representation, most notably by predicting the stream function ψ, which enforces incompressibility by construction while retaining the efficiency of a purely data-driven training procedure.

2.3 Neural Operators and Generalization Across Geometries

Neural operators seek to learn mappings between function spaces rather than between individual solution instances. In contrast to classical surrogate models, which approximate a solution for a fixed discretization or geometry, neural operators aim to approximate the underlying solution operator itself.

A particularly prominent representative is the Fourier Neural Operator (FNO) proposed by Z. Li et al. (2021). The FNO parameterizes integral kernel operators in the spectral domain by applying the Fast Fourier Transform, learning a truncated set of Fourier coefficients at each layer. This architecture offers two key advantages: first, it captures global, long-range dependencies in a single layer—a property that is naturally suited for problems

governed by elliptic operators such as the Stokes equations; second, it achieves discretization invariance, meaning that a model trained on one grid resolution can, in principle, be evaluated on a different resolution without retraining. FNOs have demonstrated strong performance on benchmark PDE problems including Darcy flow, Burgers' equation, and two-dimensional Navier–Stokes turbulence (Z. Li et al. 2021).

More broadly, convolution-based neural operators on structured grids have demonstrated the ability to generalize across variations in material parameters, forcing terms, and boundary conditions (Thuerey et al. 2020). These properties make neural operators appealing candidates for surrogate modeling in multiphase flow problems.

However, most existing studies focus on flows in relatively regular domains or on smoothly varying geometries. Applications to systems with many interacting rigid inclusions—such as sedimenting crystals with strong hydrodynamic coupling—remain scarce. In particular, the question of whether learned operators can generalize across changes in topological complexity, such as varying particle numbers $(1 \dots N)$, has not been systematically addressed. This gap is especially relevant for geophysical sedimentation problems, where the number and spatial arrangement of crystals are inherently variable.

2.4 Machine Learning for Particle- and Crystal-Laden Flows

Fluid–particle interaction has long been an active area of research in classical computational fluid dynamics. Fully resolved numerical approaches, including lattice Boltzmann, immersed boundary, and fictitious-domain methods, have been used to study drafting–kissing–tumbling dynamics, clustering, and collective settling behavior in particulate flows (**fortes_nonlinear_1987**; **ladd_numerical_1994**; Leonardi et al. 2014). These methods provide detailed physical insight but are computationally expensive, particularly as the number of particles increases.

Machine-learning-based surrogates for particle- or crystal-laden flows are only beginning to emerge. Existing studies primarily focus on learning reduced quantities, such as drag corrections, closure terms, or accelerations within coupled CFD–DEM frameworks. In contrast, surrogate models that directly map multi-particle geometric configurations to full flow fields remain largely unexplored, especially in the low-Reynolds-number regime characteristic of magmatic systems.

This lack of full-field surrogates represents a significant limitation for applications where detailed spatial information is required, for example to analyze wake interactions, boundary-layer structures, or the emergence of collective sedimentation patterns.

2.5 Comparison of Network Paradigms for Flow-Field Surrogates

A wide range of machine-learning architectures has been proposed for approximating solutions of partial differential equations. For flow-field prediction problems, three paradigms are particularly relevant: coordinate-based implicit models, linear reduced-order models, and grid-based convolutional architectures.

2.5.1 Coordinate-Based Implicit Models

Coordinate-based neural networks represent physical fields as continuous functions of spatial coordinates. These models can achieve high accuracy and produce smooth reconstructions on fixed geometries. However, explicitly encoding complex and variable geometries within coordinate-based representations remains challenging. Extending such approaches to systems with multiple interacting inclusions typically requires substantial redesign of the input representation, limiting their scalability across varying particle numbers.

2.5.2 Linear Reduced-Order Models

Linear reduced-order models, such as Proper Orthogonal Decomposition or Principal Component Analysis, project high-dimensional flow fields onto a small number of dominant modes. While computationally efficient and interpretable, their linear nature fundamentally restricts their ability to capture nonlinear flow–geometry interactions. This limitation is particularly severe for multi-crystal Stokes flow, where wake interactions and boundary-layer effects play a central role.

2.5.3 Grid-Based Convolutional Surrogates

Grid-based convolutional neural networks operate directly on structured meshes and are therefore well suited for problems in which both inputs and outputs are defined on regular grids. U-Net architectures combine local feature extraction with global context through their encoder–decoder structure and skip connections (**navab_u-net_2015**). A key advantage of this paradigm is its geometry-agnostic input representation: masks and distance fields can encode an arbitrary number of inclusions without requiring architectural modifications.

Previous studies demonstrate that CNN-based surrogates can generalize effectively across variable geometries and boundary conditions (Thuerey et al. 2020). These properties make grid-based convolutional models particularly attractive for multi-crystal sedimentation problems.

2.5.4 Spectral Neural Operators

Fourier Neural Operators (Z. Li et al. 2021) constitute an alternative grid-based paradigm that operates in the spectral rather than the spatial domain. Instead of learning localized convolutional kernels, FNOs learn global kernel functions parameterized by their Fourier coefficients. This spectral perspective provides inherent access to long-range correlations within a single network layer, which is advantageous for elliptic problems where local perturbations propagate throughout the entire domain.

A potential limitation of the standard FNO architecture is its reliance on low-frequency Fourier modes: the truncation of high-frequency coefficients may reduce accuracy near sharp interfaces such as crystal boundaries, where steep velocity gradients require high-frequency content. How this trade-off affects predictive performance in multi-crystal Stokes flow is one of the central questions investigated in this thesis.

2.5.5 Implications for This Work

The comparison of these paradigms reveals a trade-off between accuracy, geometric flexibility, and scalability. Coordinate-based implicit models excel on fixed geometries, and linear reduced-order models offer interpretability, but neither scales naturally to variable multi-inclusion configurations. Among grid-based approaches, U-Net and FNO architectures represent two complementary strategies: U-Nets capture multiscale spatial features through hierarchical encoder–decoder processing, while FNOs capture global structure through spectral convolutions. Comparing these two paradigms on the same multi-crystal Stokes flow problem constitutes a core contribution of this thesis.

2.6 Position of This Thesis in the Research Landscape

Based on the reviewed literature, four key gaps can be identified:

1. The absence of surrogate models targeting multi-inclusion Stokes flow at the level of full flow fields.

2. A lack of systematic evaluation of surrogate-model generalization with respect to particle number.

3. Limited investigation of physically structured output representations—such as the stream function—as inductive biases in data-driven flow prediction.

4. No direct comparison of spatial (U-Net) and spectral (FNO) architectures for surrogate modeling of particle-laden Stokes flow with variable geometry.

This thesis addresses these gaps by constructing a dataset of multi-crystal Stokes simulations, predicting the stream function ψ as a physically structured learning target, and systematically comparing U-Net and FNO architectures with respect to generalization performance across varying crystal numbers and geometric configurations.

3 Theory and Physical Background

This chapter summarizes the physical, mathematical, and architectural foundations underlying the surrogate modeling approaches developed in this thesis. The first part covers low-Reynolds-number fluid mechanics and the stream-function formulation, which motivate the choice of learning target. The second part introduces the two neural network architectures compared in this work—the U-Net and the Fourier Neural Operator (FNO)—and discusses their structural properties in the context of flow-field prediction.

3.1 Incompressible Stokes Flow at Low Reynolds Number

Crystal settling in magmatic systems typically occurs at very low Reynolds numbers, where viscous forces dominate over inertial effects (**happel_low_1991**; **stokes_mathematical_2009**). In this creeping-flow regime, the governing equations reduce to the incompressible Stokes equations

$$\mathbf{0} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g}, \qquad \nabla \cdot \mathbf{u} = 0, \tag{3.1}$$

where $\mathbf{u}$ denotes the velocity field, p the pressure, μ the dynamic viscosity, ρ the density, and $\mathbf{g}$ the gravitational acceleration.

A defining property of Stokes flow is its linearity with respect to the velocity field for fixed material parameters and boundary conditions. This implies that hydrodynamic interactions between rigid inclusions are long-ranged and additive, such that the disturbance induced by one particle influences the flow field throughout the domain. These long-range interactions form the physical basis for collective sedimentation phenomena, including wake interaction, drafting–kissing–tumbling dynamics, and clustering effects (**fortes_nonlinear_1987**; **ladd_numerical_1994**; Uhlmann and Doychev 2014).

In geophysical contexts, such interactions have been shown to strongly affect crystal settling rates and spatial distributions in magma chambers (Daniel Martin and Roger Nokes 1988; Weinstein, Yuen, and Olson 1988; D. Martin and R. Nokes 1989). Accurately capturing these effects therefore requires resolving the full flow field around multiple interacting crystals rather than relying on isolated particle approximations.

From a surrogate modeling perspective, the Stokes regime offers two important advantages. First, the absence of inertia leads to smooth, steady-state flow fields for fixed crystal configurations. Second, incompressibility imposes a strong global constraint on admissible

velocity fields, which can be exploited through appropriate output representations.

3.2 Stream-Function Formulation in Two Dimensions

In two-dimensional incompressible flow, the divergence-free condition $\nabla \cdot \mathbf{u} = 0$ can be satisfied identically by introducing a scalar stream function $\psi(x, z)$ such that

$$u_x = \frac{\partial \psi}{\partial z}, \qquad u_z = -\frac{\partial \psi}{\partial x}. \tag{3.2}$$

This transformation reduces the vector-valued velocity field to a single scalar potential and guarantees incompressibility by construction (**happel_low_1991**).

The stream-function formulation has a long tradition in classical hydrodynamics, particularly for planar Stokes flow problems involving rigid inclusions and sedimentation (**stokes_mathematical_2009**). Streamlines coincide with level sets of ψ, allowing qualitative flow features such as recirculation zones, wakes, and stagnation points to be identified directly.

From a numerical and learning perspective, the stream function offers several advantages. As a scalar field, ψ typically exhibits smoother spatial structure than individual velocity components, as it integrates information over spatial derivatives. Moreover, representing the solution in terms of ψ eliminates the need to explicitly enforce the incompressibility constraint during learning, in contrast to direct velocity-based approaches.

These properties motivate the use of the stream function as a learning target in this thesis, particularly in comparison to direct prediction of velocity components.

3.3 Vorticity and Poisson Reconstruction of the Stream Function

In two-dimensional incompressible flow, the scalar vorticity is defined as

$$\omega = \frac{\partial u_z}{\partial x} - \frac{\partial u_x}{\partial z}. \tag{3.3}$$

Taking the curl of the Stokes momentum equation eliminates the pressure term and yields a Poisson equation for the stream function,

$$\Delta \psi = -\omega. \tag{3.4}$$

This relationship provides a natural link between velocity-based numerical solutions and the stream-function representation. Given a divergence-free velocity field, the corresponding vorticity can be computed, and the stream function can be reconstructed by

solving a Poisson problem with appropriate boundary conditions. In the present work, this procedure is applied to high-fidelity LaMEM simulations to construct ψ as a learning target.

It is important to emphasize that this Poisson solve does not introduce new physics. Instead, it constitutes an auxiliary transformation that maps the numerical reference solution into a representation that is more amenable to learning. Similar transformations are commonly used in classical fluid mechanics to analyze Stokes flow around particles and inclusions (**happel_low_1991**).

3.4 Learning Targets as Physically Motivated Inductive Biases

From a machine-learning perspective, the choice of output representation introduces an implicit *inductive bias*, i.e. a structural constraint on the space of functions that the model can represent (**belkin_reconciling_2019**). In the context of surrogate modeling for incompressible flow, this choice strongly influences the degree to which physical structure must be learned from data rather than enforced analytically.

Direct velocity prediction treats the learning task as an unconstrained regression problem on a vector field. Incompressibility and momentum balance are not enforced explicitly but must be inferred from the training data. While this approach has been successfully applied in many CNN-based flow surrogates (Ribeiro et al. 2021; J. Chen, Viquerat, and Hachem 2019; Thuerey et al. 2020), it can lead to small but systematic violations of physical constraints.

Predicting the stream function ψ introduces a stronger inductive bias by restricting the hypothesis space to divergence-free velocity fields by construction. This approach is conceptually related to divergence-free neural representations proposed in the context of physics-informed and structure-preserving learning (Richter-Powell, Lipman, and R. T. Q. Chen n.d.). By reducing the effective degrees of freedom of the output, the learning problem is simplified and numerical conditioning can be improved.

Residual learning further modifies the inductive bias by decomposing the solution into a physically motivated baseline and a learned correction. If the baseline captures the dominant large-scale behavior of the flow, the residual field is typically smoother and of lower amplitude. Such strategies have been shown to improve learning efficiency and generalization in physics-based surrogate models (Sun et al. 2020).

3.5 U-Net Architecture

The U-Net (**navab_u-net_2015**) is an encoder–decoder convolutional neural network originally developed for biomedical image segmentation. Its architecture consists of three principal components: a contracting encoder path, an expanding decoder path, and skip connections that bridge corresponding resolution levels.

3.5.1 Encoder Path

The encoder successively reduces the spatial resolution of the input through repeated application of convolutional layers followed by downsampling operations (typically max-pooling or strided convolutions). At each stage, the number of feature channels is increased, enabling the network to learn increasingly abstract and spatially coarse representations. For an input of spatial dimension $H \times W$, L encoding stages produce feature maps at resolutions $H/2^l \times W/2^l$ for $l = 1, \ldots, L$.

3.5.2 Decoder Path and Skip Connections

The decoder mirrors the encoder structure and progressively recovers spatial resolution through upsampling operations (transposed convolutions or interpolation) followed by convolutional layers. At each decoder stage, the upsampled feature map is concatenated with the corresponding encoder feature map via a skip connection. These skip connections are critical: they provide the decoder with high-resolution spatial information that would otherwise be lost during the encoding process, enabling the network to produce spatially precise outputs.

3.5.3 Multiscale Feature Extraction

The hierarchical structure of the U-Net endows it with an inherently multiscale receptive field. Shallow layers operate at fine spatial resolution and capture localized features such as boundary layers and sharp gradients near crystal surfaces. Deep layers operate at coarse resolution with large effective receptive fields and can represent domain-scale flow structures. The skip connections allow the final prediction to integrate information across all scales simultaneously.

For multi-crystal Stokes flow, this multiscale property is directly relevant: localized velocity gradients near crystal boundaries coexist with long-range hydrodynamic interactions that span the entire computational domain. The U-Net's ability to process both scales within a single forward pass makes it a natural candidate for this class of problems.

3.6 Fourier Neural Operator

The Fourier Neural Operator (FNO) (Z. Li et al. 2021) belongs to the family of neural operators that learn mappings between infinite-dimensional function spaces. Unlike conventional convolutional networks, which learn local spatial kernels, the FNO parameterizes integral kernel operators in the spectral domain.

3.6.1 Fourier Layer

The core building block of the FNO is the Fourier layer, which applies a linear transformation in frequency space. Given an input function $v(x) \in \mathbb{R}^{d_v}$ defined on a spatial domain, the Fourier layer computes

$$v_{l+1}(x) = \sigma\Big(W_l\, v_l(x) + \mathcal{F}^{-1}\Big(R_l \cdot \mathcal{F}(v_l)\Big)(x)\Big), \tag{3.5}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fast Fourier Transform and its inverse, $R_l \in \mathbb{C}^{k_{\max} \times d_v \times d_v}$ is a learnable weight tensor applied to the lowest $k_{\max}$ Fourier modes, W_l is a pointwise linear transformation (bias path), and σ is a nonlinear activation function.

The key operation is the multiplication $R_l \cdot \mathcal{F}(v_l)$ in Fourier space, which is equivalent to a global convolution in physical space. By retaining only $k_{\max}$ modes, the FNO implicitly applies a low-pass filter while keeping the number of learnable parameters independent of the spatial discretization.

3.6.2 Architecture

A complete FNO model consists of three stages:

1. A pointwise lifting layer $P : \mathbb{R}^{d_a} \to \mathbb{R}^{d_v}$ that projects the input channels into a higher-dimensional feature space.

2. A sequence of L Fourier layers as defined in Equation (3.5), each combining spectral and local processing.

3. A pointwise projection layer $Q : \mathbb{R}^{d_v} \to \mathbb{R}^{d_o}$ that maps the learned features to the desired output dimension.

3.6.3 Properties Relevant to Stokes Flow

The FNO architecture has several properties that are relevant for the surrogate modeling task considered in this work:

- **Global receptive field.** Each Fourier layer captures dependencies across the entire spatial domain in a single operation. For Stokes flow, where the elliptic character

of the governing equations implies that local perturbations propagate globally, this is a structurally natural property.

- **Spectral bias.** The truncation to $k_{\max}$ Fourier modes introduces an implicit smoothness prior. While this favors the representation of large-scale flow structures, it may limit accuracy near sharp interfaces such as crystal boundaries, where high-frequency content is physically significant.

- **Discretization invariance.** Because the FNO operates on continuous function representations via their Fourier coefficients, a model trained on one grid resolution can, in principle, be evaluated on a different resolution without retraining.

3.7 Architectural Comparison: U-Net vs. FNO

The U-Net and FNO embody fundamentally different inductive biases for structured prediction tasks. Table 3.1 summarizes the key distinctions.

Table 3.1: Structural comparison of U-Net and FNO architectures.

Property	**U-Net**	**FNO**
Domain of operation	Spatial	Spectral (Fourier)
Receptive field	Hierarchical, local-to-global	Global per layer
Multiscale mechanism	Encoder–decoder + skip connections	Mode truncation
High-frequency content	Preserved via skip connections	Limited by $k_{\max}$
Parameter scaling	Depends on depth and channels	Independent of resolution

For multi-crystal Stokes flow, these differences translate into complementary strengths: the U-Net is expected to excel at resolving sharp boundary layers and localized wake structures due to its hierarchical spatial processing, while the FNO may offer advantages in capturing the global, elliptic character of Stokes interactions through its spectral representation. The empirical comparison of these architectures on identical training data and evaluation metrics is a central contribution of this thesis.

3.8 Implications for Multi-Crystal Sedimentation Modeling

The theoretical considerations discussed above have direct implications for surrogate modeling of multi-crystal sedimentation. Long-range hydrodynamic interactions and incompressibility impose global structure on the flow field, which is difficult to capture through purely local regression of velocity components. Predicting the stream function ψ provides

a principled way to reduce the complexity of the learning task by enforcing incompressibility analytically.

The choice of network architecture interacts with this physical structure in complementary ways. The U-Net's hierarchical encoder–decoder processing naturally mirrors the multiscale character of the flow: sharp boundary layers near crystal surfaces are captured at fine resolution levels, while domain-scale circulation patterns are represented in the deep, coarse layers. The FNO's spectral approach, on the other hand, directly reflects the elliptic nature of the Stokes equations, where local perturbations propagate globally—a property that is inherently encoded in the Fourier representation.

These complementary architectural biases motivate the systematic comparison pursued in this thesis: by training both architectures on the same data and evaluating them under identical conditions, we can disentangle the effects of physical output representation (stream function) from those of architectural design (spatial vs. spectral processing).

The next chapter builds on this theoretical foundation by translating these considerations into a concrete computational methodology for data generation, network design, training, and evaluation.

4 Methodology

This chapter describes the complete computational workflow used to generate training data, construct machine-learning inputs, train surrogate models, and evaluate their generalization performance. The methodology builds on the surrogate modeling literature reviewed in Chapter 2 and implements a reproducible end-to-end pipeline for structured-grid flow-field prediction.

Both surrogate architectures considered in this thesis—U-Net and Fourier Neural Operator (FNO)—predict the stream function ψ on a fixed 256×256 grid. The velocity field is then recovered analytically from ψ via its spatial derivatives, guaranteeing incompressibility by construction. The simulation setup, input encoding, optimization procedure, and evaluation protocol are kept identical for both architectures, so that performance differences can be attributed to the architectural design—spatial hierarchical processing (U-Net) versus spectral global processing (FNO)—rather than to confounding factors in data or training.

4.1 Learning Target: Stream-Function Prediction

Both surrogate architectures predict the scalar stream function $\psi(x, z)$ on the computational grid. The velocity field is recovered analytically via

$$\mathbf{u} = \left(\frac{\partial \psi}{\partial z}, \; -\frac{\partial \psi}{\partial x} \right),$$

which enforces incompressibility by construction. The theoretical motivation for this formulation and its advantages over direct velocity prediction were discussed in Chapter 3.

From the perspective of surrogate modeling, predicting ψ instead of $\mathbf{u}$ directly offers two key benefits. First, it reduces the learning problem from a vector-valued to a scalar regression task, lowering the effective degrees of freedom. Second, incompressibility is guaranteed analytically rather than having to be learned from data, which eliminates a common source of systematic error in purely data-driven velocity surrogates (Ribeiro et al. 2021; Thuerey et al. 2020). This choice injects physical structure into the learning problem through the output representation rather than through penalty terms, consistent with divergence-free parameterizations advocated in related work (Richter-Powell, Lipman, and R. T. Q. Chen n.d.).

The central experimental factor in this thesis is therefore not the choice of learning target—which is fixed to ψ for all experiments—but the choice of network architecture. By training both U-Net and FNO on identical data with the same learning target, performance differences can be attributed to the architectural inductive bias: spatial hierarchical processing versus spectral global processing.

4.2 LaMEM Simulations for Data Generation

Training data are generated using the geodynamic finite-element code LaMEM, which solves the incompressible Stokes equations on a regular Cartesian grid. The simulation setup follows established geodynamic Stokes discretization practices, where robustness under viscosity and density contrasts is essential (Thieulot and Bangerth 2022).

Each simulation represents a steady (quasi-static) configuration of $1 \dots N_{\max}$ rigid crystals settling in a viscous matrix. Simulations are advanced until a quasi-steady velocity field is obtained; transient effects are not considered further, as the surrogate models are trained to predict steady-state fields.

4.2.1 Domain and Discretization

The computational domain is a rectangular box discretized on a fixed 256×256 grid with uniform spacing $\Delta x = \Delta z$. The surrogate models operate on fields represented on this grid. While physical units are retained internally during simulation, learning is performed on consistently scaled grid-based fields, and the fixed discretization provides a controlled benchmark for comparing output representations.

Crystals are modeled as rigid circular inclusions with prescribed radii and randomized positions. The number of crystals N is drawn uniformly from $\{1, \dots, N_{\max}\}$ to expose the surrogate to a broad range of interaction regimes, from isolated single-crystal wakes to strongly interacting clusters. Such multi-particle interaction effects are well documented in classical sedimentation studies (**fortes_nonlinear_1987**; **ladd_numerical_1994**) and motivate an explicit evaluation of generalization across particle number. Randomization is performed independently for each simulation run, such that the training distribution samples a diverse subset of the configuration space.

4.2.2 Material Properties and Governing Equations

The matrix phase is assigned viscosity μ and density ρ_f. Crystals are approximated as effectively rigid particles by assigning a viscosity several orders of magnitude larger than μ and a density $\rho_p > \rho_f$ to induce gravitational settling. Within LaMEM, this results in a two-phase Stokes problem with sharp viscosity and density contrasts.

The governing equations are the incompressible Stokes equations

$$\mathbf{0} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g}, \qquad \nabla \cdot \mathbf{u} = 0,$$

solved with free-slip boundary conditions on all walls. Free-slip is enforced by vanishing normal velocity and vanishing tangential shear stress at the boundaries, reducing artificial boundary-layer effects and providing a simple, well-posed benchmark setting.

4.2.3 Extraction of Flow Quantities

For each simulation, LaMEM outputs grid-based fields including the phase indicator and the velocity components as well as velocity gradients:

$$\{\text{phase field}, u_x, u_z, \nabla u_x, \nabla u_z\}.$$

The phase field indicates whether a cell belongs to the crystal or matrix.

From the velocity gradients, the scalar vorticity is computed as

$$\omega = \frac{\partial u_z}{\partial x} - \frac{\partial u_x}{\partial z},$$

consistent with the two-dimensional convention used in Chapter 3. The vorticity field serves as the right-hand side for reconstructing the stream function ψ via a Poisson solve, as described in Section 4.3.

4.3 Computing and Normalizing the Stream Function

The stream function is reconstructed from vorticity by solving the Poisson equation

$$\Delta \psi = -\omega$$

on the same 256×256 grid. Reconstructing ψ via a Poisson problem is standard in two-dimensional incompressible low-Reynolds-number flow theory (**happel_low_1991**). In the present workflow, this Poisson solve is used only to define a physically consistent scalar learning target; the underlying reference solution remains the LaMEM Stokes solution.

4.3.1 Discretization and Boundary Conditions

The Laplace operator is discretized with a second-order five-point stencil on interior grid points. Homogeneous Dirichlet conditions $\psi|_{\partial\Omega} = 0$ are imposed on the domain boundary to fix the stream-function gauge and yield a unique discrete solution. Since all evaluation comparisons are ultimately performed against LaMEM-derived quantities on the

same grid, the Poisson solve serves as an auxiliary reconstruction step rather than an independent physical model.

The resulting sparse linear system is solved iteratively. The specific solver is not critical for the learning setup, provided the reconstruction error is small compared to the surrogate prediction error.

4.3.2 Motivation and Procedure for Normalization

Raw values of ψ can span several orders of magnitude (typically 10^{-16} to 10^{-10}), depending on velocity magnitude, viscosity, and grid spacing. While such values are well resolved in the numerical solver, they are unfavorable for gradient-based learning because they lead to ill-conditioned losses and large sample-to-sample dynamic range.

To improve numerical conditioning, each sample is normalized individually. We compute a characteristic order of magnitude

$$p_{\text{mean}} = \text{round}(\text{mean}\left(\log_{10}(|\psi| + \epsilon)\right)), \qquad \epsilon = 10^{-30},$$

and scale the field as

$$\psi_{\text{norm}} = \psi \cdot 10^{-p_{\text{mean}}}.$$

Training is performed on ψ_{norm}, and predictions are rescaled after inference by multiplying with $10^{p_{\text{mean}}}$. This preserves the spatial structure and sign of ψ while shifting values into an $\mathcal{O}(1)$ range that yields stable optimization.

4.4 Input Representation: Mask, Distance Field, and Coordinates

All approaches use the same five-channel input representation on the 256×256 grid. This encoding combines sharp geometric information with smooth spatial context and is designed to be interpretable from both a physical and a machine-learning perspective. Encoding geometry through masks and distance-based fields is common in structured-grid PDE surrogates and geometry-aware learning frameworks (Thuerey et al. 2020; Oldenburg et al. 2022).

1. **Crystal mask.** A binary indicator field $M(x, z) \in \{0, 1\}$ specifying crystal locations:

$$M(x, z) = \begin{cases} 1, & \text{inside any crystal,} \\ 0, & \text{in the matrix.} \end{cases}$$

   This channel provides explicit information about the location of strong material contrasts and rigid inclusions.

2. **Signed distance field (SDF proxy).** For each grid point, the Euclidean distance to the nearest crystal center is computed and assigned a negative sign inside crystals and a positive sign outside. The resulting field is normalized to approximately $[-1, 1]$. While this is a center-distance proxy rather than an exact boundary SDF, it provides a smooth measure of proximity to inclusions, complementing the sharp mask with continuous geometric context.

3. **Distance to nearest crystal boundary.** For each grid point, the Euclidean distance to the nearest crystal boundary is computed. This field provides a direct measure of how close each point in the domain is to the nearest solid–fluid interface, which is where the steepest velocity gradients occur. Unlike the SDF proxy, which encodes distance to crystal centers, this channel captures boundary proximity explicitly and helps the network resolve thin boundary layers and interaction zones between closely spaced crystals.

4. **Normalized x-coordinate.** The horizontal coordinate is linearly mapped to $x_{\text{norm}} \in [-1, 1]$, providing the network with absolute positional information.

5. **Normalized z-coordinate.** Analogously, the vertical coordinate is mapped to $z_{\text{norm}} \in [-1, 1]$. Together, the coordinate channels reduce symmetry-induced and position-dependent artifacts and improve generalization across spatially varying configurations.

Each sample is therefore represented by an input tensor of shape $(256, 256, 5)$. In physical terms, the input encodes inclusion geometry, boundary proximity, and domain position; in ML terms, it provides both high-frequency (mask) and low-frequency (distance and coordinate) information to support multiscale feature learning.

4.5 Surrogate Architectures

Two architectures are compared in this thesis. Both receive the same five-channel input of shape $(256, 256, 5)$ and produce a single-channel output ψ_{norm} of shape $(256, 256, 1)$. The theoretical foundations of both architectures were introduced in Chapter 3; this section describes the specific configurations used in the experiments.

4.5.1 U-Net

The U-Net follows the encoder–decoder principle of the original architecture (**navab__u-net__2015**) and common adaptations for laminar flow-field prediction on Cartesian grids (Thuerey et al. 2020; J. Chen, Viquerat, and Hachem 2019).

Encoder. The encoder consists of four resolution levels. Spatial resolution is reduced by a factor of two at each level, while feature widths increase as $32 \to 64 \to 128 \to 256$. Each level contains two 3×3 convolutions with batch normalization and ReLU activations, followed by downsampling via a strided convolution (stride 2). Strided downsampling is used instead of max pooling to keep the operation learnable and to reduce grid-aligned artifacts in smooth Stokes-flow targets.

Bottleneck. At the coarsest resolution, a bottleneck block with two 3×3 convolutions encodes global flow structure conditioned on the entire crystal configuration. This compressed representation provides the long-range context required to model hydrodynamic interactions between distant inclusions.

Decoder. The decoder mirrors the encoder: transposed convolutions upsample the feature maps, which are then concatenated with encoder activations through skip connections. Each decoding stage applies a two-layer 3×3 convolutional block with batch normalization and ReLU activation. Skip connections preserve high-resolution geometric detail, which is essential to reconstruct sharp structures near crystal boundaries while maintaining global consistency.

Output layer. A final 1×1 convolution projects the decoder features onto a single output channel (ψ_{norm}). No activation is applied in the final layer, as the task is a continuous regression problem.

4.5.2 Fourier Neural Operator

The FNO follows the architecture proposed by Z. Li et al. (2021), consisting of a pointwise lifting layer, a sequence of Fourier layers, and a pointwise projection layer.

Lifting. A pointwise linear layer maps the five input channels into a higher-dimensional feature space of width d_v.

Fourier layers. Each Fourier layer applies the operation defined in Equation (3.5): the input is transformed into Fourier space via a two-dimensional FFT, multiplied with a learnable complex weight tensor R_l for the lowest $k_{\max}$ modes, and transformed back. This spectral path is combined with a pointwise linear transformation (bias path) and a nonlinear activation. A stack of L such layers forms the core of the model.

Projection. A pointwise linear layer maps the learned feature representation to the single output channel ψ_{norm}.

Hyperparameters. The specific values of d_v, $k_{\max}$, and L are reported together with the experimental results in Chapter 7, where they are chosen to yield a model with comparable parameter count to the U-Net to ensure a fair comparison.

4.5.3 Regularization and Practical Considerations

Both architectures are deliberately kept lightweight to enable repeated training runs on standard GPU hardware. For the U-Net, batch normalization stabilizes optimization. For the FNO, the implicit spectral smoothness prior imposed by mode truncation serves as an architectural regularizer. Generalization is primarily promoted through geometric diversity in the pre-generated training data and the physically structured output representation. Dropout is not used, as the large variety of randomized crystal configurations in the training set provides sufficient regularization.

4.6 Training Procedure

4.6.1 Data Splits and Benchmark Design

The dataset is divided into three fixed splits—training, validation, and test—each generated once from randomized crystal configurations and stored on disk prior to any model training. All three splits are identical for both the U-Net and FNO pipelines, ensuring that performance differences between architectures are not confounded by differences in training or evaluation data. The validation split is used for monitoring during training, while the test split provides a reproducible benchmark for the final evaluation.

4.6.2 Batching and Device Handling

Samples are stored in `.jld2` format to enable efficient loading of multi-channel arrays in Julia. Mini-batches are assembled on the fly, with batch size chosen as a compromise between gradient stability and GPU memory constraints. Data tensors and model parameters are explicitly moved to CPU or GPU to ensure consistent device placement and portable execution.

4.6.3 Loss Function

The primary loss is the mean-squared error

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N_{\mathrm{pix}}} \sum_i \left(y_i^{\mathrm{pred}} - y_i^{\mathrm{true}} \right)^2 .$$

In addition, the implementation supports Huber loss and weighted MSE to increase robustness in regions with sharp gradients or to emphasize accuracy near inclusion bound-

aries. The present study uses a consistent baseline loss for both architectures to ensure a controlled comparison.

4.6.4 Optimization and Checkpointing

Model parameters are optimized using the Adam optimizer with adaptive learning rates. A fixed learning rate is used within each run, without learning rate scheduling, to keep experiments comparable. After each epoch, model checkpoints are stored in a CPU-compatible format to enable reproducible evaluation on systems without GPU support. For each run, training configuration metadata (hyperparameters, data paths, and random seeds) are archived alongside the checkpoint.

4.7 Evaluation Pipeline

Model predictions are evaluated on the fixed test dataset containing crystal geometries not used during training. This separation is essential for assessing generalization beyond the specific configurations seen during optimization.

Primary evaluation metrics include:

- mean-squared error (MSE) of the predicted target field,
- relative L_2 errors for ψ, $\partial_x\psi$, and $\partial_z\psi$, which quantify errors in both amplitude and derived velocity gradients,
- pixelwise error thresholds $\epsilon_{0.01}$, $\epsilon_{0.05}$, and $\epsilon_{0.10}$, defined as the fraction of grid points exceeding relative error levels of 1%, 5%, and 10%.

Errors are grouped by the number of crystals in the test sample ($1 \dots N_{\max}$), enabling direct assessment of how prediction quality changes as hydrodynamic interactions become more complex. The use of MSE and relative norms follows common practice in benchmarking data-driven CFD surrogates (**luo_cfdbench_2023**). The same evaluation framework is applied identically to both U-Net and FNO predictions, ensuring that all reported differences reflect architectural properties rather than evaluation artifacts.

4.8 Summary

This chapter described the full surrogate modeling workflow: LaMEM-based data generation, reconstruction and normalization of the stream function, input encoding, U-Net and FNO architecture configurations, training procedure, and evaluation protocol. The modular design ensures that all pipeline components except the network architecture are identical between the two models, so that performance differences can be attributed to

the architectural inductive bias. The next chapter applies this methodology to present numerical results, with particular emphasis on generalization across crystal numbers and spatial configurations and on the comparative strengths and limitations of U-Net and FNO for stream-function prediction.

5 Implementation

This chapter describes the technical realization of the surrogate modeling framework introduced in Chapter 4. While the previous chapter motivated the modeling choices (learning targets, input encoding, and architecture), the focus here is on the concrete implementation of the full workflow in a reproducible and modular software stack.

All components are implemented in Julia and designed to be hardware-agnostic: the same code paths run on CPU and GPU without modification. Wherever possible, explicit implementations are preferred over high-level abstractions to retain full control over numerical behavior, memory usage, and device placement.

5.1 Software Environment

All experiments are implemented in **Julia 1.10.4**. The core dependencies are:

- **Flux.jl** (v0.16.5) for neural network layers, automatic differentiation, and parameter handling.
- **CUDA.jl** (v5.9.3) for GPU acceleration on CUDA-capable hardware.
- **NNlib.jl** for low-level neural network operations, in particular batched matrix multiplication used in the FNO spectral convolution layers.
- **FFTW.jl** for the Fast Fourier Transform operations required by the Fourier Neural Operator.
- **Optimisers.jl** for optimizer definitions, in particular Adam (U-Net) and Adam with gradient clipping (FNO).
- **JLD2.jl** for efficient storage and loading of multi-channel training samples.
- **BSON.jl** for serializing and restoring model checkpoints.
- **PyPlot.jl / CairoMakie.jl** for visualization of predicted and reference fields during evaluation.

At runtime, CUDA availability is detected automatically. If a compatible GPU is present, arrays and model parameters are moved to the GPU; otherwise, the implementation falls back transparently to CPU execution.

5.2 Code Organization and Execution Flow

The code base is organized into two parallel module trees that share common infrastructure but implement architecture-specific components independently:

- `UNET_Ansatz/` contains the U-Net model, training pipeline, and evaluation routines.
- `FNO_Ansatz/` contains the FNO model, spectral convolution layers, composite loss functions, and its own training and evaluation pipelines.

Both trees share the same LaMEM interface module for data generation and the same Poisson solver for stream-function reconstruction. Within each tree, a main execution script dispatches to dedicated modules depending on the selected workflow stage: data generation, preprocessing, training, or evaluation. Each stage can be executed independently, enabling flexible experimentation and reducing coupling between components.

5.3 Model Implementations

Both architectures are implemented explicitly using Flux primitives. They share the same input and output interface: a five-channel input tensor of shape $(256, 256, 5)$—consisting of the crystal mask, a signed distance proxy field, the distance to the nearest crystal boundary, and normalized x- and z-coordinate channels—and a single-channel output ψ_{norm} of shape $(256, 256, 1)$. This input representation is geometry-agnostic: configurations with varying numbers of crystals are encoded on the same fixed grid without architectural changes.

U-Net

The encoder–decoder structure follows a classical U-Net design with skip connections. Downsampling uses strided convolutions instead of max pooling to keep the operation learnable and reduce grid-alignment artifacts in smooth flow fields. Each resolution level applies two successive 3×3 convolutions, followed by batch normalization and ReLU activation.

Upsampling is implemented via transposed convolutions. Skip connections concatenate encoder feature maps with decoder feature maps at matching resolutions. A final 1×1 convolution maps the decoder output to the single output channel. No activation is applied in the output layer, as the network performs pure regression. The model is constructed via a `build_unet()` factory function with configurable base channel width (default: 32).

Fourier Neural Operator

The FNO implementation consists of two custom layer types and a model constructor:

SpectralConv2D. The spectral convolution layer implements the frequency-space multiplication described in Chapter 3. It uses a real-valued FFT (`rfft`) instead of a full complex FFT, reducing memory consumption by approximately 50%. Two sets of complex-valued weight matrices $W_1, W_2 \in \mathbb{C}^{k_{\max} \times d_v \times d_v}$ are learned for positive and negative frequency components along the vertical axis. The forward pass applies a batched matrix multiplication in Fourier space via `NNlib.batched_mul`, which is compatible with Zygote's automatic differentiation (no in-place array mutations). Weights are initialized with small magnitude ($\sigma = 0.02$) to provide implicit warmup during early training.

FNOBlock. Each FNO block combines the spectral convolution path with a pointwise 1×1 convolution (local bias path). Both outputs are added element-wise and passed through a GELU activation function. This parallel-path design allows each block to simultaneously capture global spectral structure and local spatial features.

Model assembly. The complete model chains a pointwise lifting convolution ($5 \to d_v$ channels), a stack of L FNO blocks, and a two-layer pointwise projection ($d_v \to d_v$, GELU, $d_v \to 1$). The default configuration uses $d_v = 64$, $k_{\max} = 16$ modes in each spatial direction, and $L = 4$ Fourier layers.

5.4 Data Pipeline and Dataset Interfaces

Training samples are stored as individual `.jld2` files. Each file contains:

- the input tensor,
- the normalized target field,
- metadata describing crystal configuration and grid parameters.

All arrays are stored and processed in single precision (`Float32`) to reduce memory usage and improve GPU throughput, without measurable loss of accuracy for the present regression task.

Dataset abstraction and batching

Custom dataset types provide fine-grained control over data loading and batch assembly. Rather than relying on high-level data loaders, batches are constructed explicitly to minimize memory overhead and to enable dynamic adjustment of batch size. The dataset interface supports:

- random shuffling of samples,

- variable batch sizes (typically between 2 and 16),
- transparent transfer of batches to CPU or GPU.

LaMEM-based data generation is executed serially. This avoids race conditions and ensures deterministic behavior in the presence of a solver that is not thread-safe.

5.5 Stream-Function Reconstruction and Normalization

For stream-function learning, the vorticity field produced by LaMEM is converted into a stream function by solving the Poisson equation

$$\Delta\psi = -\omega$$

on the same Cartesian grid using a finite-difference discretization with homogeneous Dirichlet boundary conditions. This reconstruction step is used exclusively to define the learning target and is not part of the inference pipeline.

The reconstructed stream function is normalized on a per-sample basis to shift values into an $\mathcal{O}(1)$ range. The network is trained only on the normalized field ψ_{norm}, while rescaling to physical units is applied during post-processing and evaluation. This prevents extremely small target magnitudes from approaching machine precision and improves numerical conditioning of the regression loss.

5.6 Training Loop and Optimization

Training is implemented using an explicit loop to retain full control over gradient computation, parameter updates, checkpointing, and device placement. Gradients are computed via Flux's automatic differentiation.

Optimizer configuration

The U-Net uses the Adam optimizer. The FNO uses Adam combined with gradient norm clipping (`ClipNorm`) to stabilize training in the presence of spectral operations that can produce large gradient magnitudes. Typical learning rates lie between 5×10^{-5} and 10^{-4}, and training runs span between 100 and 300 epochs, depending on batch size and dataset composition.

Loss functions

The U-Net is trained with a standard MSE loss on ψ_{norm}. The FNO uses a composite loss

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \alpha_{\text{grad}} \cdot \mathcal{L}_{\text{grad}} + \alpha_{\text{bnd}} \cdot \mathcal{L}_{\text{bnd}},$$

where $\mathcal{L}_{\text{grad}}$ penalizes errors in the spatial derivatives $\partial\psi/\partial x$ and $\partial\psi/\partial z$ (computed via finite differences), and $\mathcal{L}_{\text{bnd}}$ enforces the homogeneous Dirichlet boundary condition $\psi|_{\partial\Omega} = 0$. The gradient loss coefficient α_{grad} is ramped from zero over a configurable number of warmup epochs to avoid destabilizing early optimization. This composite loss encourages the FNO to produce predictions that are not only accurate in ψ itself but also yield physically consistent velocity fields.

Checkpointing

Model checkpoints are written at the end of each epoch using BSON. All checkpoints are stored in a CPU-compatible format, ensuring that trained models can be loaded and evaluated on systems without GPU support.

5.7 Evaluation Utilities

Both implementations include dedicated evaluation toolboxes that support quantitative benchmarking and qualitative diagnostics. Quantitative metrics include mean-squared error, mean absolute error, relative L_2 norms, and maximum absolute error, computed on the stream function ψ as well as on derived velocity fields obtained via $\mathbf{u} = (\partial\psi/\partial z,\ -\partial\psi/\partial x)$. The FNO evaluation additionally reports gradient MSE and boundary MSE to assess the quality of the composite loss components. Errors are aggregated by crystal count to assess generalization as a function of geometric complexity.

Qualitative diagnostics include side-by-side visualizations of predicted and reference fields, difference plots, and derived velocity representations such as quiver and streamline plots. All evaluation routines operate independently of the training code and can be applied to any stored checkpoint, enabling systematic comparison between U-Net and FNO across training runs.

5.8 Summary

This chapter described the concrete implementation of the surrogate modeling framework, including the software environment, code organization, U-Net and FNO realizations, dataset interfaces, training loop, and evaluation utilities. Key implementation decisions—such as the use of `rfft` and Zygote-compatible operations in the FNO, and

the composite loss with gradient and boundary terms—are motivated by the specific requirements of stream-function prediction for multi-crystal Stokes flow. Together with the methodology in Chapter 4, this implementation provides a reproducible basis for the numerical experiments presented in the following chapters.

6 Experiments

This chapter presents the empirical performance of the surrogate modelling framework for **Approach 2** (stream–function prediction). We first report a structured hyperparameter study for the **single–crystal** setting, identify a robust configuration, and then evaluate the same modelling approach on **two–crystal** configurations to assess initial generalization behaviour.

6.1 Experimental Setup

All experiments were run in Julia using Flux v0.16.5 and CUDA v5.9.3 on a GPU. The surrogate model is the U-Net architecture described in Chapters 4 and 5. Training follows the implementation in Chapter 5: gradients are computed with automatic differentiation and parameters are updated using the Adam optimizer with adaptive learning rates.

Across the runs reported in this chapter, training durations range from 100 to 800 epochs, batch sizes range from 2 to 16, and learning rates were varied between 10^{-6} and $3 \cdot 10^{-4}$. Unless stated otherwise, the loss function is the mean-squared error (MSE) on the normalized stream function ψ_{norm}.

6.1.1 Learning task and reconstruction of velocity

The network predicts the *normalized* stream function ψ_{norm} from a 5-channel input consisting of (i) the crystal mask, (ii) a signed distance field (SDF), (iii) the distance to the nearest crystal boundary, (iv) normalized x coordinates, and (v) normalized z coordinates. The physical stream function ψ is reconstructed using the sample-wise scaling factor stored with each dataset entry (see Chapter 5).

For two-dimensional incompressible flows, the velocity field can be recovered from ψ via

$$u_x = \frac{\partial \psi}{\partial z}, \qquad u_z = -\frac{\partial \psi}{\partial x}, \tag{6.1}$$

so that incompressibility is satisfied by construction. In practice, this means that the learning task focuses on reproducing a single scalar field, while the vector-valued velocity field is obtained deterministically by differentiation.

6.1.2 Error metrics

To quantify prediction quality, we compare the predicted field $\hat{\psi}$ to the reference solution ψ on the discrete 256×256 grid. Let Ω denote the set of all pixels and $|\Omega|$ its cardinality.

Mean-squared error (MSE). The mean-squared error measures the average squared deviation per pixel:

$$\mathrm{MSE}(\psi) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \left(\hat{\psi}_i - \psi_i\right)^2. \tag{6.2}$$

Because deviations are squared, MSE places a stronger emphasis on large local errors. It serves as a stable global indicator in absolute units, but it depends on the magnitude (scaling) of ψ.

Relative L_2 error. To obtain a scale-invariant metric, we report the relative L_2 error:

$$\mathrm{rel}L_2(\psi) = \frac{\|\hat{\psi} - \psi\|_2}{\|\psi\|_2}, \qquad \|a\|_2 = \left(\sum_{i \in \Omega} a_i^2\right)^{1/2}. \tag{6.3}$$

Unlike MSE, relL_2 normalizes by the energy of the reference field $\|\psi\|_2$, making it comparable across samples whose flow intensity differs (e.g. due to crystal position or interaction strength). For example, rel$L_2(\psi) = 0.02$ indicates that the L_2 magnitude of the error is about 2% of the L_2 magnitude of the reference field.

Threshold exceedance fractions ϵ_τ. Global norms can be small even when localised regions contain noticeable errors. Therefore, we also compute the fraction of pixels whose *pointwise relative error* exceeds a threshold τ:

$$\epsilon_\tau(\psi) = \frac{1}{|\Omega|} \left|\{i \in \Omega : e_i(\psi) > \tau\}\right|, \qquad e_i(\psi) = \frac{|\hat{\psi}_i - \psi_i|}{|\psi_i| + \delta}. \tag{6.4}$$

Here, δ is a small stabilizing constant to prevent division by values close to zero. We report $\epsilon_{0.01}$, $\epsilon_{0.05}$, and $\epsilon_{0.10}$, i.e. the fraction of pixels with local relative errors above 1%, 5%, and 10%. These metrics help distinguish whether errors are confined to localized regions (e.g. near boundaries or strong gradients) or spread across the full domain. Importantly, exceedance fractions can remain comparatively high in regions where ψ is small in magnitude, since the relative error is sensitive to small denominators.

Metrics for derived quantities (ψ_x, ψ_z). Since velocities are obtained from ψ via Equation (6.1), we also apply the same metrics to the spatial derivatives ψ_x and ψ_z. This provides a stricter assessment: numerical differentiation tends to amplify high-frequency noise, so small pixel-scale fluctuations in $\hat{\psi}$ can translate into larger discrepancies in ψ_x

and ψ_z. Consequently, gradient-based errors are expected to be larger than errors in ψ itself.

6.2 Single–Crystal Hyperparameter Study

A set of 16 training runs was performed for $N = 1$ crystal to study the influence of batch size, learning rate, dataset size, and training duration. All runs used MSE training loss on ψ_{norm}.

6.2.1 Overview of explored settings

The study covered:

- **training set size:** 2000 vs. 4000 samples,
- **batch size:** $2, 4, 8, 16$,
- **learning rate:** 10^{-6} to $3 \cdot 10^{-4}$,
- **epochs:** 100 to 800.

In addition, a small number of exploratory runs used a changing learning rate, but the core sweep results reported below are based on fixed learning rates to maintain comparability across runs.

6.2.2 Best configuration for one crystal

Across all single–crystal runs, the strongest performance was obtained by:

Run_15: 4000 samples, learning rate $5 \cdot 10^{-5}$, 300 epochs, batch size 8.

This configuration achieved

$$\text{MSE}(\psi) \approx 2.70 \cdot 10^{-26}, \qquad \text{rel}L_2(\psi) \approx 2.09 \cdot 10^{-2},$$

and also strong performance for gradient-derived quantities ($\text{rel}L_2(\psi_x) \approx 0.166$, $\text{rel}L_2(\psi_z) \approx 0.187$), indicating that the model not only matches the global stream-function structure but also reproduces velocity-relevant spatial variations comparatively well.

Table 6.1 summarizes representative runs and the best configuration.

Table 6.1: Representative single–crystal results (Approach 2).

Run	Samples	LR	Epochs	Batch	MSE(ψ)	rel$L_2(\psi)$
Run_1	2000	10^{-4}	500	2	$1.62 \cdot 10^{-23}$	0.300
Run_3	2000	10^{-4}	500	8	$2.30 \cdot 10^{-25}$	0.0500
Run_10	4000	10^{-4}	700	8	$1.26 \cdot 10^{-24}$	0.0969
Run_11	4000	$5 \cdot 10^{-5}$	800	8	$3.70 \cdot 10^{-25}$	0.0348
Run_15	**4000**	$\mathbf{5 \cdot 10^{-5}}$	**300**	**8**	$\mathbf{2.70 \cdot 10^{-26}}$	**0.0209**
Run_16	4000	$5 \cdot 10^{-5}$	500	8	$1.43 \cdot 10^{-25}$	0.0408

6.2.3 Trends and practical takeaways

Three practical patterns emerge:

- **Batch size has a strong effect.** For otherwise identical settings (2000 samples, LR=10^{-4}, 500 epochs), batch size 8 outperformed batch sizes 2, 4, and 16. This suggests that in this problem, an intermediate batch size provides a favourable trade-off between gradient noise (too small batches) and overly smooth updates (too large batches).
- **Moderate learning rates are most reliable.** Very small learning rates (e.g. 10^{-6}) underperformed in the available training budget, consistent with under-training. Conversely, larger learning rates ($\geq 2 \cdot 10^{-4}$) did not yield improvements and can lead to unstable or overly coarse optimisation steps for this architecture and loss.
- **More epochs are not automatically better.** Run_15 (300 epochs) outperformed longer trainings at similar learning rate (e.g. Run_16 at 500 epochs and Run_11 at 800 epochs). A plausible interpretation is that after a certain point, additional training can prioritise fitting sample-specific details that do not translate into better generalisation on the held-out test set. In this sense, the selected configuration can be viewed as a good "early-stopped" solution in terms of test performance.

Finally, threshold exceedance metrics (e.g. $\epsilon_{0.01}$) remain comparatively high even for the best run. This is consistent with the nature of relative pointwise errors: pixels with small $|\psi|$ can dominate exceedance counts, even if absolute deviations are tiny and the global norm errors are low.

6.3 Two–Crystal Evaluation

To assess an initial step towards multi-crystal capability, the stream-function model was evaluated on configurations containing $N = 2$ crystals. Compared to the single-crystal regime, two-crystal configurations introduce interaction effects that can alter streamline

Table 6.2: Grouped test errors for one- and two-crystal configurations (100 samples each).

N	MSE(ψ) mean	MSE(ψ) std	rel$L_2(\psi)$ mean	rel$L_2(\psi)$ std
1	$2.31 \cdot 10^{-25}$	$2.28 \cdot 10^{-25}$	0.0554	0.0315
2	$1.27 \cdot 10^{-24}$	$3.81 \cdot 10^{-24}$	0.0765	0.0429

topology and create sharper spatial variations, thereby providing a more stringent test of the surrogate.

6.3.1 Aggregate performance for $N = 1$ vs. $N = 2$

A grouped evaluation over 100 test samples per crystal count yields the statistics in Table 6.2.

Moving from $N = 1$ to $N = 2$ increases the mean error, as expected when evaluating outside the simpler regime. Notably, the two-crystal MSE exhibits a much larger standard deviation, indicating that prediction difficulty varies substantially across samples. This behaviour is consistent with the idea that geometry matters: the relative placement of crystals can create either weak interactions (easier cases) or strong interactions with sharper gradients and more complex streamline structures (harder cases).

6.3.2 Spatial error dependence: distance to center and corners

A diagnostic evaluation stores, per crystal instance, its distance to the domain center and to the four domain corners, together with the corresponding prediction error. While the present chapter does not yet include the corresponding figures, the recorded data supports two qualitative observations:

- For **single crystals**, errors tend to increase when the crystal is far from the domain center (equivalently, closer to boundaries/corners). A plausible explanation is that the flow structure in these cases is more strongly influenced by boundary conditions, and the effective fraction of "unstructured" fluid region changes as the crystal approaches the domain edges.

- For **two crystals**, the error distribution becomes broader and exhibits more outliers. This is consistent with interaction regimes in which the two crystals generate localised zones of stronger gradients and more complex streamline topology, which are harder to reproduce accurately with a purely data-driven surrogate.

A natural next step is to visualise these dependencies explicitly (scatter plots and binned averages) for $N = 1$ and $N = 2$ separately, and to relate high-error outliers to geometric descriptors such as minimum inter-crystal distance and boundary proximity.

6.4 Discussion

6.4.1 What works well

The results demonstrate that Approach 2 can learn the single-crystal mapping with high accuracy in global norms. In particular, the best configuration achieves rel$L_2(\psi) \approx 2\%$ for $N = 1$ without additional physics-informed loss terms. Because velocities are derived from a stream function, the approach preserves incompressibility by construction (Equation 6.1), which is a practical advantage: the model does not need to learn a divergence-free constraint explicitly.

6.4.2 Where errors concentrate and why gradients are harder

Two recurring patterns appear in the evaluation:

- **Gradient-derived quantities are more challenging.** Even when ψ is reconstructed accurately in a global sense, small pixel-scale deviations can translate into visibly larger discrepancies in ψ_x and ψ_z, because differentiation amplifies high-frequency noise. Consequently, gradient errors act as a stricter test for capturing fine-scale flow features that directly affect the velocity field.
- **Geometry-dependent difficulty.** Two-crystal configurations produce a wider spread of errors, indicating that the surrogate's performance depends on the interaction regime. Configurations with strong hydrodynamic coupling or pronounced local gradients are expected to be harder than cases where crystals are well separated or positioned in a more symmetric manner.

6.4.3 Implications for the multi-crystal goal

The increase in mean error from $N = 1$ to $N = 2$ provides a first estimate of the generalisation gap when moving to more complex geometries. The substantially higher variance for $N = 2$ suggests that "average" metrics alone are not sufficient to characterise performance; instead, evaluations should be stratified by geometric descriptors. Two particularly actionable directions are: (i) reporting errors as a function of boundary distance and minimum inter-crystal distance, and (ii) expanding the training distribution to include multi-crystal samples so that interaction patterns are learned directly rather than extrapolated.

6.5 Summary

This chapter evaluated the stream-function surrogate (Approach 2) on single- and two-crystal configurations. A hyperparameter study for $N = 1$ identified an effective con-

figuration (4000 samples, learning rate $5 \cdot 10^{-5}$, batch size 8, 300 epochs) achieving $\mathrm{rel}L_2(\psi) \approx 0.021$. When evaluating on $N = 2$ crystals, the mean error increased to $\mathrm{rel}L_2(\psi) \approx 0.076$ with a substantially larger variance across samples. Distance-based diagnostics suggest geometry-dependent difficulty, motivating a structured generalisation analysis for $N \geq 2$ and targeted dataset enrichment in subsequent work.

Das Kapitel muss umgeschrieben werden. Ich hier soll einmal kuz erklärt werden was genau ich untersuche und vergleiche etc. Kaptiel 7 wird gestrichen. Die Ergbnisse aus meinen Experiementen werden dann in Kapitel 8 präsentiert und diskutiert. In diesem Kapitel soll es also um die Beschreibung der Experimente gehen, die ich durchgeführt habe, um die Leistung meines Modells zu bewerten. Ich werde erklären, welche Metriken ich verwende, um die Vorhersagen meines Modells mit den Referenzlösungen zu vergleichen, und wie ich die Ergebnisse interpretiere. Es wird auch eine Diskussion darüber geben, welche Aspekte meiner Modellierung gut funktionieren und wo es noch Herausforderungen gibt.

7 Results

8 Validation

9 Conclusion and Outlook

10 Discussion

A Bibliography

[1] Chen, Junfeng et al. (2019). "U-net architectures for fast prediction in fluid mechanics". en. In.

[2] Jin, Xiaowei et al. (Feb. 2021). "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations". en. In: *Journal of Computational Physics* 426, p. 109951. ISSN: 00219991. DOI: 10.1016/j.jcp.2020.109951. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999120307257 (visited on 02/15/2026).

[3] Ladd, Anthony J C (1994). "Numerical Simulations of Particulate Suspensions via a Discretized Boltzmann Equation Part I. Theoretical Foundation". en. In.

[4] Leonardi, Alessandro et al. (May 2014). "Coupled DEM-LBM method for the free-surface simulation of heterogeneous suspensions". In: *Comp. Part. Mech.* 1.1, pp. 3–13. ISSN: 2196-4378, 2196-4386. DOI: 10.1007/s40571-014-0001-z. URL: http://link.springer.com/10.1007/s40571-014-0001-z (visited on 11/24/2025).

[5] Li, Xiaohui et al. (2022). "IBM-LBM-DEM Study of Two-Particle Sedimentation: Drafting-Kissing-Tumbling and Effects of Particle Reynolds Number and Initial Positions of Particles". In: *Energies* 15.9. ISSN: 1996-1073. DOI: 10.3390/en15093297. URL: https://www.mdpi.com/1996-1073/15/9/3297.

[6] Li, Zongyi et al. (May 2021). *Fourier Neural Operator for Parametric Partial Differential Equations*. en. arXiv:2010.08895 [cs]. DOI: 10.48550/arXiv.2010.08895. URL: http://arxiv.org/abs/2010.08895 (visited on 02/15/2026).

[7] Martin, D. and R. Nokes (Dec. 1989). "A Fluid-Dynamical Study of Crystal Settling in Convecting Magmas". en. In: *Journal of Petrology* 30.6, pp. 1471–1500. ISSN: 0022-3530, 1460-2415. DOI: 10.1093/petrology/30.6.1471. URL: https://academic.oup.com/petrology/article-lookup/doi/10.1093/petrology/30.6.1471 (visited on 02/15/2026).

[8] Martin, Daniel and Roger Nokes (Apr. 1988). "Crystal settling in a vigorously convecting magma chamber". en. In: *Nature* 332.6164, pp. 534–536. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/332534a0. URL: https://www.nature.com/articles/332534a0 (visited on 02/15/2026).

[9] Nissanka, Kavinda J. et al. (Feb. 2023). "Dynamics of Mass Polar Spheroids During Sedimentation". en. In: *Journal of Fluid Mechanics* 956. arXiv:2206.09505 [cond-

mat], A28. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2023.32. URL: http://arxiv.org/abs/2206.09505 (visited on 02/15/2026).

[10] Oldenburg, Jan et al. (June 2022). "Geometry aware physics informed neural network surrogate for solving Navier–Stokes equation (GAPINN)". en. In: *Advanced Modeling and Simulation in Engineering Sciences* 9.1, p. 8. ISSN: 2213-7467. DOI: 10.1186/s40323-022-00221-z. URL: https://amses-journal.springeropen.com/articles/10.1186/s40323-022-00221-z (visited on 02/15/2026).

[11] Penlou, Baptiste et al. (Mar. 2023). "Experimental Measurement of Enhanced and Hindered Particle Settling in Turbulent Gas-Particle Suspensions, and Geophysical Implications". en. In: *Journal of Geophysical Research: Solid Earth* 128.3, e2022JB025809. ISSN: 2169-9313, 2169-9356. DOI: 10.1029/2022JB025809. URL: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2022JB025809 (visited on 02/15/2026).

[12] Popov, Anton and Boris J. P. Kaus (2013). "LaMEM (Lithosphere and Mantle Evolution Model): advancing a staggered-grid finite difference version of the code". In: *EGU General Assembly 2013*. Vol. 15. Source code available at https://github.com/UniMainzGeo/LaMEM, EGU2013–7761.

[13] Raissi, M. et al. (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378, pp. 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[14] Ribeiro, Mateus Dias et al. (2021). "DeepCFD: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks". en. In.

[15] Richter-Powell, Jack et al. (n.d.). "Neural Conservation Laws: A Divergence-Free Perspective". en. In: ().

[16] Ronneberger, Olaf et al. (May 2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. en. arXiv:1505.04597 [cs]. DOI: 10.48550/arXiv.1505.04597. URL: http://arxiv.org/abs/1505.04597 (visited on 02/15/2026).

[17] Sun, Luning et al. (Apr. 2020). "Surrogate Modeling for Fluid Flows Based on Physics-Constrained Deep Learning Without Simulation Data". en. In: *Computer Methods in Applied Mechanics and Engineering* 361. arXiv:1906.02382 [physics], p. 112732. ISSN: 00457825. DOI: 10.1016/j.cma.2019.112732. URL: http://arxiv.org/abs/1906.02382 (visited on 02/15/2026).

[18] Thieulot, Cedric and Wolfgang Bangerth (Jan. 2022). "On the choice of finite element for applications in geodynamics". en. In: *Solid Earth* 13.1, pp. 229–249. ISSN: 1869-9529. DOI: 10.5194/se-13-229-2022. URL: https://se.copernicus.org/articles/13/229/2022/ (visited on 02/15/2026).

[19] Thuerey, Nils et al. (Jan. 2020). "Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows". en. In: *AIAA Journal* 58.1, pp. 25–36.

ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J058291. URL: https://arc.aiaa.org/doi/10.2514/1.J058291 (visited on 02/15/2026).

[20] Uhlmann, Markus and Todor Doychev (Aug. 2014). "Sedimentation of a dilute suspension of rigid spheres at intermediate Galileo numbers: the effect of clustering upon the particle motion". en. In: *Journal of Fluid Mechanics* 752. arXiv:1406.1667 [physics], pp. 310–348. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2014.330. URL: http://arxiv.org/abs/1406.1667 (visited on 02/15/2026).

[21] Weinstein, Stuart A. et al. (1988). "Evolution of crystal-settling in magma-chamber convection". In: *Earth and Planetary Science Letters* 87.1, pp. 237–248. ISSN: 0012-821X. DOI: https://doi.org/10.1016/0012-821X(88)90078-7. URL: https://www.sciencedirect.com/science/article/pii/0012821X88900787.